Shahid Bahonar University of Kerman
Journal of Mahani Mathematical Research
Print ISSN: 2251-7952 Online ISSN: 2645-4505

A NEW MODEL FOR LUNG CANCER PREDICTION BASED ON DIFFERENTIAL EVOLUTION ALGORITHM AND EFFECTIVE FEATURE SELECTION

A. Khatibi Bardsiri ✉

Article type: Research Article
(Received: 23 March 2024, Received in revised form 03 August 2024)
(Accepted: 18 October 2024, Published Online: 24 October 2024)

Abstract. Lung cancer is one of the most dangerous and fatal diseases worldwide. By using advanced machine learning techniques and optimization algorithms, early prediction and diagnosis of this disease can be achieved. Early identification of lung cancer is an important approach that can increase the survival rate of patients. In this paper, a novel method for lung cancer prediction is proposed, which combines two important techniques: Support Vector Machine (SVM) and Differential Evolution (DE) algorithm. Firstly, using the differential evolution algorithm, important and suitable features for lung cancer prediction are extracted. Then, using the SVM classifier, a classification model is built for prediction. The proposed approach is implemented on two lung cancer databases and achieves a good level of accuracy, which is compared with four other methods: C4.5 decision tree, neural network, Naive Bayes classifier, and logistic regression. The proposed model, with high accuracy and generalization power, is a suitable model for lung cancer detection and can serve as a strong decision support system alongside medical professionals.

Keywords: Lung cancer, Support vector machine, Differential evolution algorithm, Feature selection.
2020 MSC: 37L05.

1. Introduction

Lung cancer is currently one of the major diseases worldwide, characterized by the uncontrolled growth of lung tissue cells. According to recent statistics, lung cancer is the leading cause of death among cancer diseases and accounts for a significant number of deaths globally [10]. It causes the death of approximately 1.61 million people annually. The majority of lung cancer cases (85%) are attributed to long-term tobacco use, while about 10 to 15% of cases occur in individuals who have never smoked [1]. Lung cancer is the second leading cause of death among men and the tenth among women. Early and accurate diagnosis of lung cancer is of great importance as it enables better and

✉ a.khatibi@srbiau.ac.ir, ORCID: 0000-0001-9640-498X
https://doi.org/10.22103/jmmr.2024.23134.1597
Publisher: Shahid Bahonar University of Kerman
How to cite: A. Khatibi Bardsiri, *A new model for lung cancer prediction based on differential evolution algorithm and effective feature selection*, J. Mahani Math. Res. 2025; 14(1): 345-367.

more effective treatment in the early stages of the disease, potentially saving the patient's life. Diagnosing the disease is a complex task and often requires conducting numerous tests on patients to reach an accurate result [3]. This can lead to the utilization of analytical devices designed to assist physicians in their decision-making processes. Lung cancer can be detected through chest radiography and computed tomography imaging. With the advancement of technology and the development of machine learning methods, the use of prediction algorithms and models for cancer detection and prediction has gained significant attention. To achieve high accuracy and optimal performance in lung cancer prediction, selecting appropriate features and extracting important information from the data is vital [2], we investigate the utilization of two advanced machine learning and optimization techniques: Support Vector Machine and Differential Evolution algorithm. SVM is a well-known algorithm for data classification and has the potential to detect and predict cancer diseases. The Differential Evolution algorithm is a popular optimization algorithm used for feature extraction from data. The main objective of this paper is to present a new and efficient method for lung cancer prediction that combines these two techniques and demonstrates a significant improvement in prediction accuracy and reliability. Finally, the performance of the proposed algorithm is evaluated using the lung cancer database and compared with four other methods: C4.5 decision tree, neural network, Naive Bayes classifier, and logistic regression. The fact is that the innovation of the proposed method is mainly focused on defining the fitness function (Section 3.3) rather than the algorithm itself. The proposed function can also be extended to advanced algorithms. For the first time, we simultaneously included both the number of features and the classification error rate. In summary, if the proposed objective function is integrated as the core of detection in new methods, it will significantly improve the accuracy of detection. Our goal has been to modify the objective function rather than the algorithm. In all similar works, evaluation has typically focused solely on a combination of metrics such as accuracy, precision, F-score, and recall. The subsequent sections of this paper are structured as follows. Section 2 provides an overview of the previous research and the algorithms used for lung cancer diagnosis. In Section 3, we provide a detailed explanation of the proposed method. Sections 4 and 5 analyze the performance of the proposed model through experimental design and results. Finally, Section 6 presents a general conclusion and outlines future research directions.

2. **Literature Review**

Within the domain of data mining, machine learning algorithms like SVM, Decision Trees, Neural Networks, Bayesian models, and k-Nearest Neighbors (k-NN) are prominently utilized. This section provides a comprehensive look at how these algorithms have been applied in cancer detection, organized chronologically. In 2013, Chen et al. proposed a fuzzy system utilizing the

k-Nearest Neighbors method for diagnosing Parkinson's disease [4]. They integrated principal component analysis, surpassing the Support Vector Machine with a 96.07% accuracy [4]. Odajima and Pawlovsky (2014) [11] delved into the impact of varying neighbor numbers on the k-Nearest Neighbors method's classification accuracy. They meticulously documented the fluctuations in accuracy concerning classifier sizes and neighbor variations. Lynch et al. (2017) conducted a comparative study of C4.5, Naive Bayes, and k-Nearest Neighbors for breast cancer detection [9]. Naive Bayes and k-Nearest Neighbors achieved a matching accuracy of 98.51%, while C4.5 lagged at 91.79%. Hashi et al. (2017) utilized Decision Tree and k-Nearest Neighbors for diabetes detection, obtaining accuracies of 90.43% and 76.96%, respectively, favoring the Decision Tree algorithm [7]. Alharbi (2018) implemented a fuzzy-genetic algorithm for lung cancer detection, achieving a commendable 97.5% accuracy [1]. Cherif (2018) introduced an accelerated k-Nearest Neighbors algorithm for breast cancer detection, showcases superior accuracy compared to alternative methods [5]. Vikas et al. (2019) scrutinized Support Vector Machines and Random Forest for lung cancer prediction, highlighting the superior performance of Support Vector Machines (SVM) with 98% accuracy and swift execution [25]. In 2020, Puneet and Chauhan focused on lung cancer prediction using various techniques such as Boosting models, Logistic Regression, SVM, Gaussian Naive Bayes, Decision Trees, and k-Nearest Neighbors. Their findings favored the Boosting model, achieving 92.16% accuracy [16]. Venkatesh and Raamesh (2022) explored ensemble learning methods for lung cancer prediction, identifying AdaBoost as the top performer with 98.2% accuracy [24]. Alsinglawi et al. (2022) introduced a framework for predicting lung cancer patient survival times, revealing that Random Forest with SMOTE class balancing achieved 98% accuracy [2]. In 2023, Varchagall, et al. proposed a novel approach using machine learning to identify tumorous lung characteristics on CT scans, showing promise in identifying emphysema (AUC = 0.78) [24]. Table 1 offers a comparative analysis of distinct machine learning methods for lung cancer prediction. These algorithms showcase varied performance and applicability, requiring careful consideration based on specific data attributes [8, 20, 27].

Table 2 provides a comprehensive comparison of the discussed machine learning methods in lung cancer prediction. This table systematically compares these methods across various criteria, ensuring a clearer understanding of their strengths and limitations in the context of lung cancer prediction. This addition will aim to provide readers with a more structured and informative overview of the current state-of-the-art methodologies in the field. This table provides a detailed comparison across key criteria, helping to understand the suitability of each method for lung cancer prediction based on specific requirements and challenges commonly encountered in healthcare applications.

*__Criteria Explanation:__

*Accuracy*__: Reflects the performance in terms of predictive accuracy on lung cancer datasets.__

TABLE 1. A review of different machine learning methods used in lung cancer prediction.

Authors (year)	Dataset (number of samples)	Methods used (*Proposed method)	Efficiency of the proposed method
Yuan et al. (2023)	LUNA16/ LIDC-IDR (1080)	3D ECA-ResNet*	Accuracy 94.89% sensitivity 94.91% F1-score 94.65%
Liu et al. (2023)	LIDC-IDRI (302)	PiaNet*	Sensitivity 93.6%
Siddiqui et al. (2023)	LUNA 16/ LIDC-IDRI/ TCIA(27816)	3D MLF-DCNN*	Accuracy 99.2% sensitivity 99.2% specificity 99.17%
Alsinglawi et al. (2022)	MIMIC-III (423)	Random forest* reinforcement logistic regression learning	Accuracy 95.3% Recall 98%
Venkatesh et al. (2022)	SEER(1000)	Begging AdaBoost* decision tree, neural network nearest neighbor	Accuracy 98.2%
Puneet et al. (2020)	Lanzhou University (277)	Begging Reinforcement learning* logistic regression decision tree support vector machine naive Bayesian gaussian decision tree k-nearest neighbor	Accuracy 92.16% Recall 96.97%
Sim et al. (2020)	HRQO(809)	AdaBoost* logistic regression decision tree random forest bagging	Accuracy 94.8%
Patra (2020)	UCI(32)	Radial network* support vector machine random forest artificial neural network k-nearest neighbor naive bayesian	Accuracy 81.25% Precision 81.3% Recall 81.1% F1 score 81.5%
Radhika et al. (2019)	Data World (1000)	Support vector machine* naive Bayes decision tree regression	Accuracy 99.2
Wu et al. (2019)	Lanzhou University (277)	Random forest*	Accuracy 95.7% Recall 96.3
Faisal et al. (2018)	UCI(32)	MLP neural network naive bayes support vector machine random forest majority voting gradient tree*	Accuracy 90% Precision 87.82% Recall 83.71% F1 score 85.71%
Safiyari et al. (2017)	SEER(924)	Bagging AdaBoost* Bayesian network random forest logistic regression C4.5	Accuracy 88.98%

TABLE 2. Comparing different machine learning methods used in lung cancer prediction*.

Method	Accuracy	Interpretability	Scalability	Handling Imbalanced Data	Training Time	Computational Complexity
Linear Support Vector Machines	High	Low	Moderate	Requires techniques such as class weighting or resampling (e.g., SMOTE)	Moderate	Training: $O(N^2.d)$ Prediction: $O(d)$
Non-Linear Support Vector Machines	High	Low	High	Requires techniques such as class weighting or resampling (e.g., SMOTE)	High	Training: $O(N^3)$ Prediction: $O(N_s.d)$
Random Forest	High	Moderate	High	Handles imbalanced data naturally through ensemble averaging	Moderate	Training: $O(T.m.N^2.logN)$ Prediction: $O(T.logN)$
Logistic Regression	Moderate to High	High	High	Requires techniques such as class weighting or regularization	Low to Moderate	Training: $O(I.N.d)$ Prediction: $O(N.d)$
Neural Networks	High	Low	High	Requires techniques such as dropout and regularization	High	Training: $O(E.N.L.n^2)$ Prediction: $O(N.L.n^2)$
Decision Trees	Moderate	Moderate	Moderate	Handles imbalanced data naturally through tree structure	Low to Moderate	Training: $O(d.N^2)$ Prediction: $O(N.logN)$
Naive Bayes	Moderate	High	High	Handles imbalanced data naturally through probabilistic framework	Low	Training: $O(N.d)$ Prediction: $O(N.C.d)$

Interpretability: **Indicates how easy it is to understand and interpret the model's predictions.**
Scalability: **Assesses the ability to handle large volumes of data efficiently.**

$HandlingImbalancedData$: Evaluate how well the method deals with datasets where one class (e.g., cancer cases) is significantly less prevalent than the other.
$TrainingTime$: Approximate time required to train the model on typical lung cancer datasets.
$ComputationalComplexity$: Reflects the overall complexity of the model architecture and its implementation.
N: Number of training samples
d: Number of features
N_s: Number of support vectors
T: Number of trees
m: Number of features considered for splitting at each node
I: Total number of iterations
E: Number of epochs
L: Number of layers
n: Average number of neurons per layer
C: Number of classes

3. Proposed Method

The proposed method in this paper is a combination of two algorithms: differential evolution algorithm and Support Vector Machine classifier. Initially, the DE algorithm will extract the effective features, and then the classifier will utilize these features to predict cancer. Both methods will be explained further. Figure 1 shows the diagram of the proposed method in summary. Figure 2 shows the pseudo code of the proposed method.

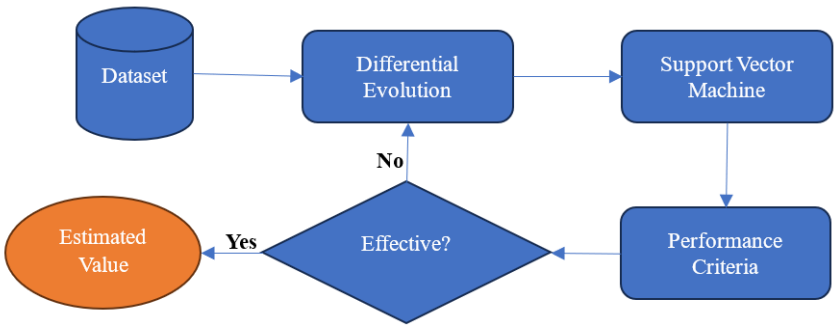

FIGURE 1. Diagram of the proposed method at a high level of abstraction.

```
1. **Define Objective Function for Differential Evolution**
  - **Function `objective_function(x, X_train, y_train)`**
    - **Input**: Feature selection vector `x`, Training data `X_train`, Labels `y_train`
    - **Process**:
      1. Convert the vector `x` to a set of feature indices (features where `x[i] >= 0.5`).
      2. Extract the selected features from `X_train` based on these indices.
      3. Train an SVM classifier using the extracted features.
      4. Compute the classifier's accuracy on the training data.
      5. Return the negative accuracy (to be minimized in the DE algorithm).
    - **Output**: Negative accuracy.

2. **Define Differential Evolution Optimization Function**
  - **Function `perform_differential_evolution(X_train, y_train)`**
    - **Input**: Training data `X_train`, Labels `y_train`
    - **Process**:
      1. Define bounds for each feature selection variable (0 to 1).
      2. Apply Differential Evolution to optimize the `objective_function`.
      3. Extract the best feature selection vector from the DE results.
    - **Output**: Boolean array indicating selected features.

3. **Main Procedure**
  - **Initialization**:
    - Load or generate training data `X_train` and labels `y_train`.
  - **Feature Selection**:
    - Call `perform_differential_evolution(X_train, y_train)` to determine the most relevant
features.
  - **Train SVM Classifier**:
    - Extract features from `X_train` based on the selected features.
    - Train an SVM classifier using these selected features.
  - **Model Evaluation**:
    - Load or generate test data `X_test`.
    - Predict labels for `X_test` using the trained SVM classifier.
    - Output the prediction results.

4. **Evaluation Procedure **
  **Calculate confusion matrix**
    - True Positives (TP): Number of correctly predicted positive cases
    - True Negatives (TN): Number of correctly predicted negative cases
    - False Positives (FP): Number of incorrectly predicted positive cases
    - False Negatives (FN): Number of incorrectly predicted negative cases
  **Evaluation metrics **
    - **Accuracy** = (TP + TN) / (TP + TN + FP + FN)
    - **Sensitivity (Recall)** = TP / (TP + FN)
    - **Specificity** = TN / (TN + FP)
```

FIGURE 2. Pseudo code of the proposed method.

3.1. **Support Vector Machine. Support Vector Machine is a machine learning algorithm used for classification and regression problems. This method is based on the idea of separating data using a specific hyperplane in a defined feature space. SVM aims to find an optimal**

hyperplane in the input space and divide different data points into distinct categories optimally. If the data points in the input space are separable, meaning that an exact hyperplane can be found to separate them, linear SVM is used. In this case, SVM is optimized based on support vectors (points from each class that are closest to the hyperplane) [13]. However, if the data points are not linearly separable, meaning they have some degree of overlap, nonlinear SVM is used. In this case, SVM transforms the data using kernel functions into higher-dimensional feature space and searches for a linear hyperplane to separate them in that space. The main formulations of the Support Vector Machine are as follows [22]:

$$w = \sum_{i=1}^{n} a_i y_i X_i \qquad Linear. \tag{1}$$

$$w = \sum_{i=1}^{n} a_i y_i K(X_i, X) \qquad Non - Linear. \tag{2}$$

$$f(x) = sign(w^T X + b) \qquad Kernel function. \tag{3}$$

(4)

$$Max\ w(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j K(X_i, X_j)\ Optimization\ objective\ function.$$

In these formulas, x represents the feature vectors, y represents the classification labels (1 or -1), α represents the Lagrange multipliers, b represents the bias of the hyperplane, w represents the support vector, and $K(x, y)$ represents the kernel function. The kernel function K is responsible for transforming the data into a higher-dimensional feature space. This function can be linear, polynomial, Gaussian, etc. The function $f(x)$ is the decision function for predicting the label of a new point X. Optimization in a Support Vector Machine involves finding the best separating hyperplane based on the data. Figure 3 illustrates the data classification process in the Support Vector Machine method.

3.2. **Differential Evolution algorithm. The Differential Evolution (DE) algorithm, conceived by Storn and Price in the mid-1990s [15], stands as a widely used population-based optimization technique for resolving various optimization problems. Specifically designed for continuous, nonlinear, and multi-modal optimization problems, DE operates through the following stages: Initialization: Establish the population size, NP, and initialize a group of NP candidate solutions randomly within the search space. Optionally, assign random velocity values to**

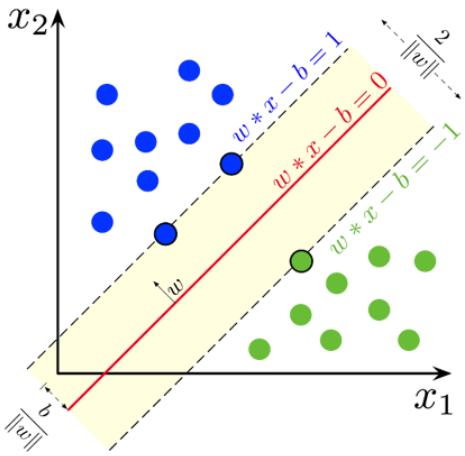

FIGURE 3. Data Classification in Support Vector Machine Method [22].

each candidate solution, applied in some variations. Mutation: For every candidate solution in the population, pick three distinct individuals labeled "target," "base," and "rand" from the population. Create a mutant vector by adding the weighted difference between the "base" and "rand" vectors to the "target" vector. over: Generate a trial vector for each candidate solution by combining the mutant vector with the original candidate solution using a crossover operator. This operator determines the inheritance of components from both vectors. Selection: Evaluate the fitness of the trial vector via an objective function for each candidate solution. If the trial vector demonstrates superior fitness compared to the original candidate solution, replace the latter with the former in the population. Termination: Repeat steps 2-4 until meeting a termination criterion, such as reaching the maximum number of iterations or fulfilling convergence criteria. The algorithm stops when the termination criterion is met, returning the best candidate solution obtained so far [12]. The crux of the Differential Evolution algorithm lies in its differential operators (mutation and crossover), enabling exploration of the search space and guiding populations toward optimal solutions. The mutation operator introduces diversity by modifying candidate solutions, while the crossover operator combines data from various solutions to form trial solutions for assessment. This capability allows DE to effectively navigate complex, multi-modal landscapes. Differential Evolution boasts several variants and extensions like $DE/rand/1$,

$DE/rand/2$, and $DE/best/1$, which differ in the number of individuals utilized for mutation and the crossover strategies employed. This algorithm serves as a robust and adaptable optimization technique, demonstrating successful applications across diverse fields and solving a multitude of optimization problems. The summary of the Differential Evolution algorithm is illustrated in Figure 4.

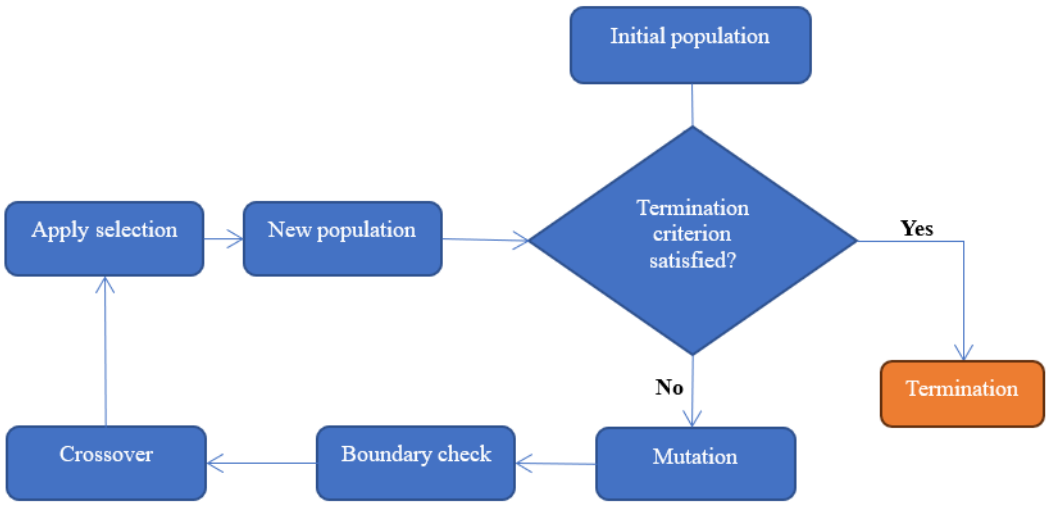

FIGURE 4. A flowchart of a differential evolution algorithm.

3.3. **Problem Objective Function. The main objective in feature selection is to achieve the minimum possible accuracy where all the features of the samples contribute to it, and the secondary objective is to improve the accuracy. Collecting extensive information about the features of the samples incurs significant costs in terms of time and money, and it also leads to wasted time in the classification and detection process. Therefore, it is better to reduce the dimensions, meaning the number of features, in order to obtain better results and also achieve a better correlation between the features and the outcomes. The particle swarm optimization algorithm is a suitable technique for selecting the best features. In this algorithm, a binary random vector, called the Vector, which includes the features, is generated using the formula (5).**

$$Vector = \begin{cases} 1 & i-th\ feature\ included \\ 0 & otherwise \end{cases}. \tag{5}$$

Then, an objective function is defined based on the sum of the error rate and the number of features for each selected combination of features. This objective function acts as a penalty function that needs to be minimized to find the best feature combination. Here, the Misclassification Rate (MCR) can be easily obtained using formula (6).

$$mcr = \frac{\sum a_{ij} - [\sum a_{ij} \ ; (i = j)]}{\sum a_{ij}}; \qquad i = j = 1, 2, ..., m. \tag{6}$$

In this case, the variable m represents the number of classes, and a_{ij} represents the number of cases where sample i is classified as class j using the classification method. Now, the objective function that needs to be minimized is a weighted sum of MCR (Misclassification Rate) and NF (number of selected features), defined as follows:

$$MinZ = w_1 * mcr + w_2 * nf. \tag{7}$$

The variables $w1$ and $w2$ can be defined as weighted penalties for misclassification and having an extra feature, respectively. Using this objective function, the Differential Evolution algorithm aims to find the best combination of features with the minimum number of features, simultaneously minimizing both the cost and the misclassification rate. The termination condition for the algorithm can be a predefined number of iterations.

4. Experimental Design

In this section, we will describe the data used, evaluation metrics, and the execution system for the codes. The goal of the experimental design is to prepare a proper and accurate comparison framework. The results of all experiments are calculated using 10-fold cross-validation to prevent artificial bias in the responses.

4.1. **Data Description. The importance of the dataset is undeniable as it significantly influences the final outcome of the research. In this study, the Lung Cancer dataset from Data World is used (https://data.world/cancerdatahp/lu cancer-data). This dataset consists of 1000 samples, each with 23 features, which are shown in Table 3. After obtaining the dataset, it is examined whether any preprocessing is needed to remove missing values or replace them with appropriate data. Rows in the dataset that were incomplete were automatically filled using the mean of other values. The classes in this dataset represent the risk levels of lung cancer, classified into three levels: low, medium, and high.**

Another suitable dataset for laryngeal cancer that can be used to test data mining methods is the Laryngeal Data from the UCI Machine Learning Repository. This dataset contains information about

TABLE 3. Features available in the Lung Cancer dataset 1.

Age	**gender**	**occupational risk**	**genetic risk**	**blood group**
Weight	**smoking**	**shortness of breath**	**chest pain**	**coughing**
Snoring	**dry cough**	**frequent colds**	**alcohol consumption**	**smoking cessation**
Air pollution	**allergy**	**chronic disease**	**diet**	**nail color change**
Weight loss	**excessive fatigue**	**difficulty in swallowinge**		

patients with laryngeal cancer and related features. Laryngeal Cancer Dataset Specifications are as follows:
Dataset Name: Laryngeal Data 2
Source: UCI Machine Learning
Description: This dataset includes medical information of patients with laryngeal cancer, making it suitable for data mining and machine learning analyses.
Number of Records (Instances): 213
Number of Features (Attributes): 17
Attributes: Includes demographic and clinical information about patients, such as age, gender, type of cancer, metastasis status, and other relevant features.
Instances: Contains data on various patients, which can be used as input for data mining models.
UCI Laryngeal Data (https://archive.ics.uci.edu/ml/datasets/Laryngeal1)
This dataset can help us test various data mining methods and obtain meaningful results in the diagnosis and analysis of laryngeal cancer.

4.2. **Performance criteria. Confusion matrix is an important tool in data mining used for evaluating the performance of models and machine learning algorithms. This matrix represents the number of correct and incorrect predictions for each class, using the actual values (labeled samples) and the predictions made by the model. A confusion matrix is defined as an $n * n$ table, where n is the number of classes in the problem. The actual and predicted classes are placed on the horizontal and vertical axes of the table, respectively. Generally, a confusion matrix is defined as shown in Figure 5.**

The values that are placed in the cells of the confusion matrix are as follows:

• $TruePositive(TP)$**: The number of samples that are correctly identified and belong to the positive class.**

		Prediction	
		Positive	Negative
Actual	Positive	TP	FN
	Negative	FP	TN

FIGURE 5. Confusion matrix.

• $TrueNegative(TN)$**: The number of samples that are correctly identified and belong to the negative class.**
• $FalsePositive(FP)$**: The number of samples that are incorrectly identified and actually belong to the negative class.**
• $FalseNegative(FN)$**: The number of samples that are incorrectly assigned to the negative class and actually belong to the positive class. Using the confusion matrix, we can calculate metrics such as accuracy, sensitivity, and specificity of the model as follows:**

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{9}$$

$$Specificity = \frac{TP}{TN + FP} \tag{10}$$

These metrics provide insights into different aspects of the model's performance and help evaluate its effectiveness in classification tasks. Accuracy is not always the best metric for imbalanced datasets so we use metrics such as sensitivity, specificity to better evaluate the model performance. Additionally, we used the 10-fold cross-validation method for model evaluation, which prevents biased results.

4.3. **Execution System. All simulations and experiments were performed on a system with an Intel 2.7 GHz 7-core processor, 12 GB RAM, and Windows 10 operating system. The algorithms and methods were implemented using MATLAB software version 2018.**

4.4. **Handling Class Imbalance. Certainly, addressing class imbalance is crucial for ensuring the robustness and fairness of our proposed method. Here's how we handle class imbalance and mitigate bias in our model training process. We analyze the distribution of classes in**

the dataset to identify imbalance. We assign higher weights to minority class samples during model training to ensure they contribute more to the overall loss function, thereby giving them proportionate importance. We choose algorithms that inherently handle class imbalance well, such as algorithms like SVM with class weighting. Also, we utilize evaluation metrics that are robust to class imbalance, such as sensitivity, Precision-Recall curve, rather than relying solely on accuracy. We focus on selecting informative features that can help the model distinguish between classes effectively, reducing the impact of imbalance. We employ stratified 10-fold cross-validation to ensure that each fold retains the proportion of classes similar to that in the original dataset, preventing overfitting and bias. By implementing these strategies, our proposed method aims to ensure fair representation of each class and mitigate bias introduced by imbalanced datasets, thereby enhancing the reliability and applicability of our findings.

4.5. **Initial Parameters. In the paper, the initial parameters of different methods have been carefully configured to optimize their performance in data analysis and predictions. This includes selecting the algorithm type, specifying relevant parameters, and conducting necessary experiments to fine-tune the model's performance. Table 4 illustrates these settings.**

These initial parameter settings are adjusted based on experimenta-

TABLE 4. Initial parameters and variables.

Method	Parameter	Initial value
Differential evolution	Population Size	50
	Mutation Rate	0.1
	Mutation Rate	0.9
SVM	Kernel Type	RBF
	Regularization Parameter (C)	1.0
	Gamma(for RBF kernel)	0.1
Neural network	Number of Layers	3
	Neurons per Layer	128, 64, 32
	Activation Function	ReLU
C4.5	Pruning Method	Reduced Error Pruning
	Confidence Threshold	0.25
Logistic regression	Regularization Parameter (C)	1.0
	Solver	'lbfgs'
	Maximum Iterations	100

tion and empirical validation to achieve optimal results in the context

of the study's dataset. Naive Bayes classifiers do not have complex parameters like regularization or specific solver options as seen in Logistic Regression or Support Vector Machines. Instead, they rely on the assumption of independence among features given the class label, making them simple yet effective for many classification tasks. Therefore, no specific parameters need to be set initially for Naive Bayes beyond the basic implementation of the algorithm itself.

5. Experimental Results

In this section, we will compare the performance of the proposed method with other approaches. Firstly, the convergence behavior of the Differential Evolution algorithm can be observed in Figure 6. According to the information, the rate of decrease in the objective function error is initially high and then slows down after the sixth iteration. At this point, the algorithm reaches an almost optimal combination of problem features. The penalty weights for misclassification and the number of features are assumed to be equal in this experiment. Next, Table 5 presents the confusion matrix for

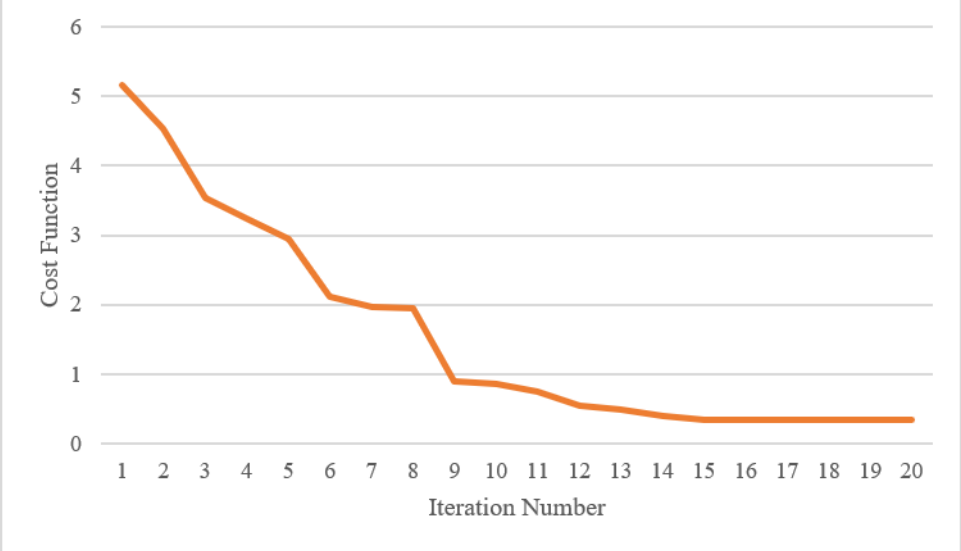

FIGURE 6. The best value of the objective function during different iterations.

the C4.5 method. The total number of samples is 1000 patients. This method achieved the worst performance for the average class, reaching an accuracy of 86.79%. The best performance is observed

for the high-risk class with an accuracy rate of 98.51%. The overall performance of the method is also shown in each case.

TABLE 5. Confusion matrix for C4.5.

	High	Low	Mid	Total	Accuracy
High	**330**	**0**	**5**	**335**	**98.51**
Low	**0**	**375**	**25**	**400**	**93.75**
Mid	**20**	**15**	**230**	**265**	**86.79**
Total	**350**	**390**	**260**	**1000**	**93.50**

Similar results for the Neural Network method are provided in Table 6. Generally, the performance of the Neural Network method is worse than the C4.5 decision tree, with an overall accuracy of 91.30% in cancer diagnosis. Similarly, for the average class, it had the most challenging classification task, correctly identifying only 84.91% of the disease cases.

TABLE 6. Confusion matrix for Neural network.

	High	Low	Mid	Total	Accuracy
High	**334**	**8**	**2**	**344**	**97.09**
Low	**17**	**382**	**25**	**424**	**90.09**
Mid	**20**	**15**	**197**	**232**	**84.91**
Total	**371**	**405**	**224**	**1000**	**91.30**

The next method is the Naive Bayes algorithm, and its confusion matrix is shown in Table 7. The performance of the Naive Bayes method is close to the C4.5 method and has created a more uniform distribution among the different data classes. This method has achieved almost similar accuracy percentages in the high, low, and average classes, indicating its stability and robustness on different and imbalanced data.

TABLE 7. Confusion matrix for Naive bayesian.

	High	Low	Mid	Total	Accuracy
High	**315**	**10**	**8**	**333**	**94.59**
Low	**27**	**371**	**10**	**408**	**90.93**
Mid	**8**	**5**	**246**	**259**	**94.98**
Total	**350**	**386**	**264**	**1000**	**93.20**

Table 8 illustrates the performance of the Logistic Regression method. On average, this method had the worst performance. Its accuracy of 89.7% is significantly lower compared to similar methods. Finally, Table 9 presents the performance of the proposed method

TABLE 8. Confusion matrix for logistic regression.

	High	Low	Mid	Total	Accuracy
High	**332**	**15**	**15**	**362**	**91.71**
Low	**7**	**340**	**25**	**372**	**91.40**
Mid	**6**	**35**	**225**	**266**	**84.59**
Total	**345**	**390**	**265**	**1000**	**89.70**

in the form of a confusion matrix. As evident, the proposed method effectively increased the accuracy rate, reaching 99.4% overall. Here, the reliability and stability of the results are comparable to the Naive Bayes method. The overall improvement of the proposed method compared to the C4.5, Neural Network, Naive Bayes, and Logistic Regression methods is 4.7%, 7.2%, 5%, and 11% respectively. The results indicate that the proposed method has achieved its primary goal of improving the accuracy of lung cancer prediction. Regarding the interpretation of results, on dataset 1, which consists of real and meaningful data (Table 3), the proposed model evaluated three features age, smoking, and polluted air, as highly important and impactful. The findings of the proposed model align with medical theories in this area. Figure 7 introduces the specificity and sensi-

TABLE 9. Confusion matrix for proposed method.

	High	Low	Mid	Total	Accuracy
High	**354**	**0**	**1**	**355**	**99.72**
Low	**3**	**345**	**0**	**348**	**99.14**
Mid	**0**	**2**	**295**	**297**	**99.33**
Total	**357**	**347**	**296**	**1000**	**99.40**

tivity measures to the comparison process on dataset 1. The results demonstrate that the proposed method outperforms in other measures as well [17]. The simultaneous increase in both specificity and sensitivity measures indicates that the proposed method has accurately distinguished between true positive and true negative diagnoses, without errors. Therefore, this method can be generalized to other sensitive diseases (e.g., infectious diseases, acute diseases requiring quarantine) as well. Figure 8 depicts the accuracy results

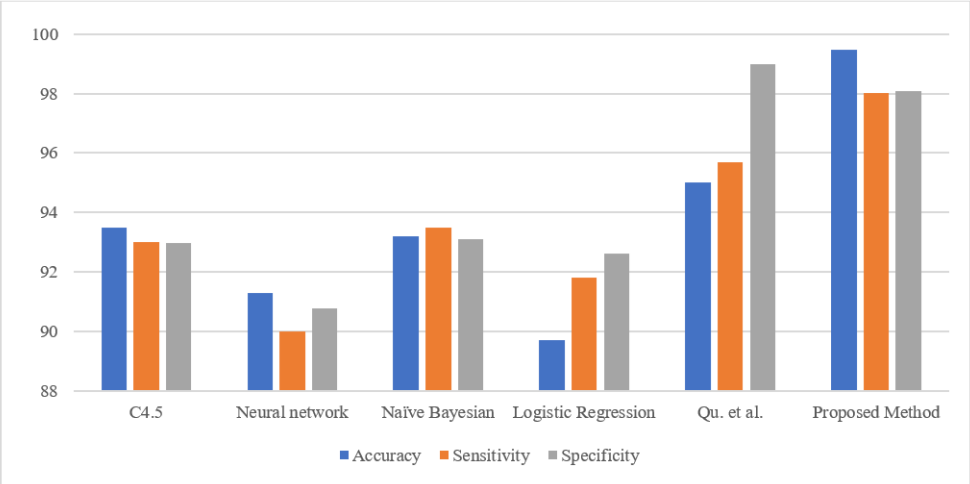

FIGURE 7. Efficiency criteria for different methods on dataset 1.

of various models on Dataset 2. As evident, the proposed method outperforms other methods on this dataset as well. However, an important point to note is the decrease in accuracy across all methods in Dataset 2 compared to Dataset 1. This decrease in accuracy is due to greater diversity and heterogeneity of data in Dataset 2 compared to Dataset 1. Table 10 presents the results for examining the achievement of the second objective of the research. The results demonstrate that the proposed method not only improves the diagnostic accuracy but also enhances the speed of the process. The performance of the feature selection process is clearly evident here. The reduction in the number of features has made the proposed method faster compared to the baseline support vector machine, without compromising accuracy. It is worth noting that performing initial parameter settings significantly affects the response time of each method.

6. Adaptability to New Data

Here, we discuss the mechanisms that can be implemented for updating or retraining the model in response to new data or evolving patterns: Incremental Learning: Implementing incremental learning techniques allows the model to learn continuously from new data without retraining from scratch. This can be particularly useful in

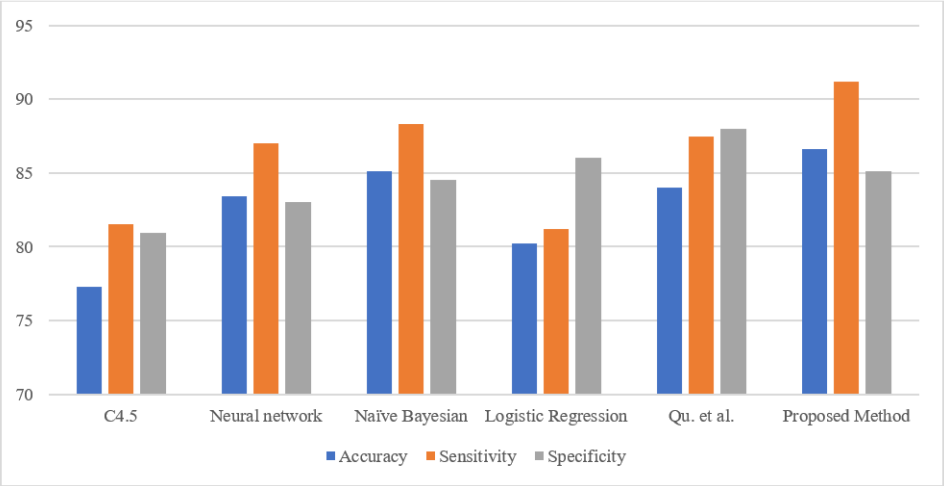

FIGURE 8. Efficiency criteria for different methods on dataset 2.

TABLE 10. Comparison of accuracy and time of different methods.

	Dataset 1		Dataset 2	
	Accuracy	**Time**	**Accuracy**	**Time**
C4.5	**93.5**	**0.017**	**77.3**	**0.005**
Neural Network	**91.3**	**0.026**	**83.4**	**0.013**
Naive Bayesian	**93.2**	**0.029**	**85.12**	**0.011**
Vector Machine	**92.5**	**0.031**	**80.2**	**0.021**
Logistic Regression	**89.7**	**0.012**	**86.6**	**0.007**
Proposed Method	**99.4**	**0.16**	**77.3**	**0.09**

adapting to new patterns while preserving previously learned knowledge. Online Learning Algorithms: These algorithms can update the model in real-time as new data points arrive, making the model adaptable to changes in data distribution. Scheduled Retraining: Establishing a schedule for periodic retraining of the model ensures that it incorporates the latest data and adapts to any changes in the data distribution. This can be done weekly, monthly, or based on the volume of new data accumulated. Data Drift Detection: Implementing data drift detection mechanisms can alert when the statistical

properties of the input data change. Techniques such as monitoring feature distributions or using drift detection algorithms can trigger retraining when significant drift is detected. Ensemble Learning: Using ensemble methods such as stacking or boosting can enhance the model's adaptability. New models can be trained on the latest data and combined with existing models to improve overall performance and adapt to new trends. By incorporating these mechanisms, we aim to ensure that the proposed model remains adaptable and continues to provide accurate predictions in the face of new data and evolving data distributions. Future work will focus on implementing and testing these strategies to maintain the model's efficacy over time.

7. Conclusion and Future Work

Lung cancer is one of the most important and dangerous diseases worldwide. Providing a low-cost and highly accurate method can significantly contribute to the early and cost-effective diagnosis of this disease, in addition to medical and pathological approaches. In the literature review, it was observed that there are various machine learning methods whose performance depends on different aspects, including the dataset used for their application. In this paper, a combined method of support vector machine classifier with particle swarm optimization algorithm was introduced for more accurate prediction of lung cancer. The experimental results demonstrated that the proposed method achieves satisfactory results compared to five other methods: C4.5 decision tree, neural network, Naive Bayes classifier, logistic regression, and Qu et al. method. Furthermore, empirical results showed that by selecting an optimal number of features and adjusting appropriate parameters, both the accuracy and speed can be effectively improved. Proposed model can become computationally expensive and memory-intensive when dealing with large datasets, which is often the case in medical applications where extensive patient data is involved. Also, it can be sensitive to noisy data and outliers, potentially affecting its performance, especially in datasets where noise is prevalent or data quality is varied. Future work may involve the use of other classification algorithms in the field of machine learning or the utilization of different optimization algorithms for further enhancement of the proposed method.

8. Author Contributions

Amid Khatibi Bardsiri was responsible for all aspects of the research and manuscript preparation, including conceptualization, methodology, software, validation, formal analysis, investigation, resources, data curation, writing—review and editing, and visualization.

9. Data Availability Statement

The primary dataset utilized is the Lung Cancer dataset from Data World, which can be accessed via the following link: https://data.world/cancerdatahp/lung-cancer-data. Additionally, another relevant dataset for testing data mining methods is the Laryngeal Data available from the UCI Machine Learning Repository.

10. Aknowledgement

We would like to thank the reviewers for their thoughtful comments and efforts towards improving our manuscript.

11. Ethical considerations

The authors avoided from data fabrication and falsification.

12. Funding

This research received no external funding.

13. Conflict of interest

The author declare no conflict of interest.

References

[1] Alharbi, A. (2018). An automated computer system based on genetic algorithm and fuzzy systems for lung cancer diagnosis. International Journal of Nonlinear Sciences and Numerical Simulation, 19(6), 583-594. https://doi.org/10.1515/ijnsns-2017-0048

[2] Alsinglawi, B., Alshari, O., Alorjani, M., Mubin, O., Alnajjar, F., Novoa, M., Darwish, O. (2022). An explainable machine learning framework for lung cancer hospital length of stay prediction. Scientific reports, 12(1), 1-10. https://doi.org/10.1038/s41598-021-04608-7

[3] Chauhan, A. (2020). Detection of lung cancer using machine learning techniques based on routine blood indices. Paper presented at the 2020 IEEE international conference for innovation in technology (INOCON). https://doi.org/10.1109/INOCON50539.2020.9298407

[4] Chen, H.-L., Huang, C.-C., Yu, X.-G., Xu, X., Sun, X., Wang, G., & Wang, S.-J. (2013). An efficient diagnosis system for detection of Parkinson's disease using fuzzy k-nearest neighbor approach. Expert Systems with Applications, 40(1), 263-271. https://doi.org/10.1016/j.eswa.2012.07.014

[5] Cherif, W. (2018). Optimization of K-NN algorithm by clustering and reliability coefficients: application to breast-cancer diagnosis. Procedia Computer Science, 127, 293-299. https://doi.org/10.1016/j.procs.2018.01.125

[6] Faisal, M. I., Bashir, S., Khan, Z. S., Khan, F. H. (2018). An evaluation of machine learning classifiers and ensembles for early stage prediction of lung cancer. Paper presented at the 2018 3rd international conference on emerging trends in engineering, sciences and technology (ICEEST). https://doi.org/10.1109/ICEEST.2018.8643311

[7] Hashi, E. K., Zaman, M. S. U., & Hasan, M. R. (2017). An expert clinical decision support system to predict disease using classification techniques. Paper presented at the 2017 International conference on electrical, computer and communication engineering (ECCE). https://doi.org/10.1109/ECACE.2017.7912937

[8] Liu, W., Liu, X., Luo, X., Wang, M., Han, G., Zhao, X., Zhu, Z. (2023). A pyramid input augmented multi-scale CNN for GGO detection in 3D lung CT images. Pattern Recognition, 136, 109261. https://doi.org/10.1016/j.patcog.2022.109261

[9] Lynch, C. M., Abdollahi, B., Fuqua, J. D., de Carlo, A. R., Bartholomai, J. A., Balgemann, R. N., . . . Frieboes, H. B. (2017). Prediction of lung cancer patient survival via supervised machine learning classification techniques. International Journal of Medical Informatics, 108, 1-8. https://doi.org/10.1016/j.ijmedinf.2017.09.013

[10] Maleki, N., Zeinali, Y., Niaki, S. T. A. (2021). A k-NN method for lung cancer prognosis with the use of a genetic algorithm for feature selection. Expert Systems with Applications, 164, 113-981. https://doi.org/10.1016/j.eswa.2020.113981

[11] Odajima, K., & Pawlovsky, A. P. (2014). A detailed description of the use of the kNN method for breast cancer diagnosis. Paper presented at the 2014 7th International Conference on Biomedical Engineering and Informatics. https://doi.org/10.1109/BMEI.2014.7002861

[12] Opara, K. R., & Arabas, J. (2019). Differential Evolution: A survey of theoretical analyses. Swarm and evolutionary computation, 44, 546-558. https://doi.org/10.1016/j.swevo.2018.06.010

[13] Pathoee, K., Rawat, D., Mishra, A., Arya, V., Rafsanjani, M. K., Gupta, A. K. (2022). A cloud-based predictive model for the detection of breast cancer. International Journal of Cloud Applications and Computing (IJCAC), 12(1), 1-12. https://doi.org/10.4018/IJCAC.310041

[14] Patra, R. (2020). Prediction of lung cancer using machine learning classifier. Paper presented at the Computing Science, Communication and Security: First International Conference, COMS2 2020, Gujarat, India, March 26–27, 2020, Revised Selected Papers 1. https://doi.org/10.1155

[15] Price, K. V., Storn, R. M., & Lampinen, J. A. (2005). The differential evolution algorithm. Differential evolution: a practical approach to global optimization, 37-134. https://doi.org/10.1007/3-540-31306-0

[16] Puneet, & Chauhan, A. (2020, 6-8 Nov. 2020). Detection of Lung Cancer using Machine Learning Techniques Based on Routine Blood Indices. Paper presented at the 2020 IEEE International Conference for Innovation in Technology (INOCON). https://doi.org/10.1109/INOCON50539.2020.9298407

[17] Quanyang, W., Yao, H., Sicong, W., Linlin, Q., Zewei, Z., Donghui, H., . . . Shijun, Z. (2024). Artificial intelligence in lung cancer screening: Detection, classification, prediction, and prognosis. Cancer Medicine, 13(7), e7140. https://doi.org/10.1002/cam4.7140

[18] Radhika, P., Nair, R. A., Veena, G. (2019). A comparative study of lung cancer detection using machine learning algorithms. Paper presented at the 2019 IEEE International Conference on Electrical, Computer and Communication Technologies (ICECCT). https://doi.org/10.1109/ICECCT.2019.8869001

[19] Safiyari, A., Javidan, R. (2017). Predicting lung cancer survivability using ensemble learning methods. Paper presented at the 2017 intelligent systems conference (IntelliSys). https://doi.org/10.1109/IntelliSys.2017.8324368

[20] Siddiqui, E. A., Chaurasia, V., Shandilya, M. (2023). Classification of lung cancer computed tomography images using a 3-dimensional deep convolutional neural network with multi-layer filter. Journal of Cancer Research and Clinical Oncology, 149(13), 11279-11294. https://doi.org/10.1007/s00432-023-04992-9

[21] Sim, J.-a., Kim, Y., Kim, J. H., Lee, J. M., Kim, M. S., Shim, Y. M., . . . Yun, Y. H. (2020). The major effects of health-related quality of life on 5-year survival prediction among lung cancer survivors: applications of machine learning. Scientific reports, 10(1), 1-12. https://doi.org/10.1038/s41598-020-67604-3

[22] Suthaharan, S. (2016). Support vector machine. Machine learning models and algorithms for big data classification: thinking with examples for effective learning, 207-235. https://doi.org/10.1007/978-1-4899-7641-3

[23] Varchagall, M., Nethravathi, N. P., Chandramma, R., Nagashree, N., & Athreya, S. M. (2023). Using Deep Learning Techniques to Evaluate Lung Cancer Using CT Images. SN Computer Science, 4(2), 173. https://doi.org/10.1007/s42979-022-01587-y

[24] Venkatesh, S. P., & Raamesh, L. (2022). Predicting Lung Cancer Survivability: a Machine Learning Ensemble Method on Seer Data. https://doi.org/10.21203/rs.3.rs-1490914/v1

[25] Vikas, P. K., & Kaur, P. (2021). Lung cancer detection using chi-square feature selection and support vector machine algorithm. International Journal of Advanced Trends in Computer Science and Engineering. https://doi.org/10.30534/ijatcse/2021/801032021

[26] Wu, J., Zan, X., Gao, L., Zhao, J., Fan, J., Shi, H., . . . Xie, X. (2019). A machine learning method for identifying lung cancer based on routine blood indices: qualitative feasibility study. JMIR medical informatics, 7(3), e13476. https://doi.org/10.2196

[27] Yuan, H., Wu, Y., Dai, M. (2023). Multi-Modal Feature Fusion-Based Multi-Branch Classification Network for Pulmonary Nodule Malignancy Suspiciousness Diagnosis. Journal of Digital Imaging, 36(2), 617-626. https://doi.org/10.1007/s10278-022-00747-z

Amid Khatibi Bardsiri
Orcid number: 0000-0001-9640-498X
Computer Engineering Department
Bardsir Branch, Islamic Azad University
Bardsir, Iran
***Email address*: a.khatibi@srbiau.ac.ir**