Performance Analysis of Various Machine Learning and Deep Learning Approaches for the Detection of Lung Cancer

K. Ramkumar[1], M. Natarajan[2]

[1]Research Scholar, Department of Computer and Information Science, Faculty of Science, Annamalai University, Annamalainagar, Tamil Nadu, India.
[2]Department of Computer and Information Science, Faculty of Science, Annamalai University, Annamalainagar, Tamil Nadu, India.

Email: [1]ramau13@gmail.com, [2]mind.2004@gmail.com

KEYWORDS
Lung Cancer; Classification; Machine Learning; Artificial Neural Networks; Support Vector Machines; Decision Tree; Naïve Bayes; Deep Learning.

ABSTRACT:
Lung cancer is one of the deadliest forms of cancer worldwide, making early detection essential for improving survival outcomes. The use of machine learning (ML) and deep learning (DL) has brought significant advancements to medical diagnostics, providing exceptional accuracy in detecting lung cancer. This text examines the use of ML and DL methods for the detection and classification of lung cancer, emphasizing their effectiveness in analyzing complex medical imaging data, such as CT scans. Sophisticated models, like Principal Component Analysis, K-Nearest Neighbors, Support Vector Machines, Naïve Bayes, Decision Trees, Artificial Neural Networks, and Deep learning techniques, have shown outstanding performance in identifying lung nodules and differentiating between benign and malignant tumors. These approaches not only improve diagnostic accuracy but also minimize false positives, enabling timely and appropriate medical intervention.

1. Introduction

Lung cancer continues to be one of the leading causes of cancer-related deaths globally, underscoring the critical importance of early detection and precise classification. Advances in computational methods have made machine learning (ML) and deep learning (DL) approaches increasingly vital for enhancing diagnostic accuracy. This article offers an in-depth performance analysis of various ML and DL techniques applied to lung cancer detection and classification. By comparing these methods, the study seeks to identify the most effective solutions, shedding light on their advantages, limitations, and potential clinical applications. The evaluation delivers valuable insights into the rapidly advancing field of AI-driven cancer diagnostics.

Data mining is a key process within data science focused on extracting valuable patterns and insights from large datasets. Integrating techniques from statistics, machine learning, database systems, and artificial intelligence, it reveals hidden patterns, correlations, and trends that are not immediately obvious. This process is essential for transforming raw data into actionable information, supporting decision-making, prediction, and strategic planning across various

fields such as healthcare, finance, and marketing. It involves multiple stages, including data preprocessing, model building, pattern discovery, and interpretation, making it indispensable in today's data-centric environment.

According to Han, Pei, and Kamber (2011), data mining has grown significantly in importance due to the exponential increase in data generated across industries. The ability to analyze and interpret large volumes of data is essential for organizations seeking to remain competitive in a rapidly evolving market landscape.

Machine learning, a branch of artificial intelligence (AI), centers on creating algorithms and statistical models that allow computers to learn from data and make predictions or decisions. Unlike traditional programming, where humans explicitly define rules, machine learning systems enhance their performance over time by recognizing patterns and relationships within the data. This adaptive capability makes machine learning a versatile tool across various fields, including image recognition, natural language processing, predictive analytics, and autonomous systems.

Machine learning is typically divided into three main categories: supervised learning, where models are trained on labeled data; unsupervised learning, where models detect patterns in unlabeled data; and reinforcement learning, where models learn through trial and error within an environment. As Mitchell (1997) explains, machine learning involves "the study of computer algorithms that improve automatically through experience," underscoring its importance in the evolution of AI and data-driven decision-making.

2. Review of the Literature

At the heart of deep learning lies the artificial neural network, which typically comprises numerous layers that incrementally extract more abstract features from the input data. As noted by Goodfellow, Bengio, and Courville (2016), the depth of these networks enables them to capture intricate patterns, driving state-of-the-art performance in various complex tasks and significantly advancing the field of artificial intelligence. CNNs have been the dominant model for such tasks, the introduction of capsule networks offers a new avenue for improving the reliability and accuracy of automated diagnostic tools. The findings of Afshar et al. (2020) align with other studies that seek to enhance model interpretability and reduce false positives/negatives in medical diagnoses.

Ardila et. al. (2019) conducted a pioneering study on lung cancer screening using three-dimensional deep learning, as detailed in Nature Medicine. Their research introduced a deep learning model for end-to-end lung cancer detection from low-dose chest computed tomography (CT) scans. This model was designed to enhance both the accuracy and efficiency of screening processes. Balagurunathan et. al. (2019) explored the reproducibility and prognostic value of quantitative features extracted from CT images in their study published in Translational Oncology.

Bhatia, A., & Verma, K. (2021) conducted a comprehensive survey on deep learning techniques for lung cancer detection from CT scans, systematically evaluates various deep learning methodologies applied to CT imaging for lung cancer detection. The study provides an in-depth analysis of the strengths, limitations, and advancements in this field, offering valuable insights into the effectiveness and future directions of deep learning approaches in improving lung cancer diagnosis. Coudray, et. al. (2018) developed a deep learning approach for the classification and mutation prediction of non-small cell lung cancer (NSCLC) histopathology images. Focused on leveraging deep learning techniques to analyze histopathological images for accurate cancer classification and prediction of genetic mutations.

Dhaware, B. U., & Pise, A. C. (2020) investigated lung cancer detection utilizing a Bayesian classifier combined with Fuzzy C-Means (FCM) segmentation, as presented at the IEEE International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT). Giger, M. L., & MacMahon, H. (2019) explores how deep learning technologies are transforming clinical imaging and interpretation practices for lung cancer detection. The authors discuss the advancements and potential of deep learning methods in enhancing the accuracy and efficiency of lung cancer screening, marking a significant shift in the field.
Huang et. al. (2021) investigated densely connected convolutional networks for lung cancer detection, they introduced a novel network architecture designed to enhance the detection of lung cancer from medical images. Their approach leverages dense connections to improve feature extraction and network performance, demonstrating promising results in advancing the accuracy of lung cancer detection. Liu, H., et. al. (2021) provided a detailed review of deep learning techniques in medical image analysis, with a focus on their applications in lung cancer detection, discusses various deep learning methodologies and their implementation in analyzing medical images for lung cancer diagnosis.
Makaju, S., et. al., S. (2019) investigated lung cancer detection using CT scan images in their study published in Procedia Computer Science. Their research focuses on various methods and techniques for identifying lung cancer from CT imaging. The study evaluates different approaches to enhance detection accuracy and improve diagnostic performance, contributing valuable insights into the application of image analysis for lung cancer detection. Shen, W., et. al. (2019) proposed a novel approach for classifying lung nodule malignancy using multi-crop convolutional neural networks (CNNs), detailed in Pattern Recognition.
Tan, C., et. al., (2020) introduced an innovative attention-based aggregation convolutional neural network (CNN) for tumor segmentation and classification in lung cancer CT images. Their study, presents a novel approach that enhances tumor identification by integrating attention mechanisms into the CNN framework. This method aims to improve the precision of tumor segmentation and classification, demonstrating significant advancements in the application of deep learning techniques for analyzing lung cancer CT scans. Wang, H., et. al. (2021) developed an interpretable CNN-based model for lung cancer prediction using 3D chest CT scans, as detailed in Nature Communications. Their study presents a convolutional neural network (CNN) designed to enhance the interpretability of lung cancer predictions.
Xie, Y., et. al. (2019) investigated the integration of texture, shape, and deep model-learned information at multiple scales for diagnosing COVID-19 from chest CT images and explores how combining these diverse types of information can enhance diagnostic accuracy. Zhang, H., et. al., (2021) provided a thorough review of deep learning-based methods for lung cancer diagnosis in their article, systematically examining various deep learning techniques and their applications in lung cancer detection. The study covers advancements, challenges, and the overall effectiveness of these methods, highlighting their impact on improving diagnostic accuracy and efficiency.

3. Backgrounds and Methods

A. Dataset

In this study, we utilized the Standard Digital Image Database from the Japanese Society of Radiological Technology (JSRT) discussed by Shiraishi et al. in 2000. Gunaydin et. al. in 2019, the dataset comprises 247 computed tomography (CT) images, as illustrated in Fig. 1. Each image is labeled to indicate the presence or absence of a lung nodule, with 154 images identified as containing a nodule and 93 as not containing one. The images are stored as

2048x2048 matrices, with each data point occupying 2 bytes. The dataset is divided into training and testing sets for classification purposes, with 70% of the data allocated to the training set and 30% to the testing set, selected randomly.

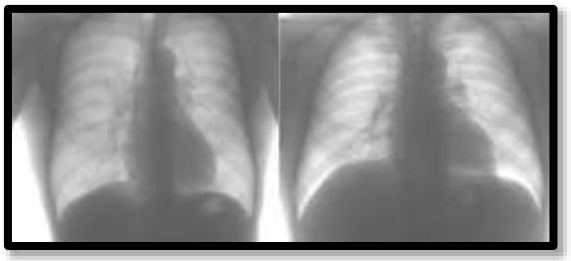

Fig. 1. Lung Nodule CT Images from Dataset

B. Principal Component Analysis – Eigenvectors

Principal Component Analysis (PCA) is a technique for dimensionality reduction that reorients data into a new coordinate system where the primary variance is captured by the initial principal components. The process involves several essential steps related to eigenvectors and eigenvalues:

1. **Standardize the Data:** Ensure equal contribution from each feature by normalizing the data.
2. **Compute the Covariance Matrix:** Measure how features vary together. Calculate the covariance matrix C of the standardized data. For a dataset with 'n' features, this matrix is n x n
3. **Calculate Eigenvalues and Eigenvectors:** Determine the directions (eigenvectors) of maximum variance and their magnitude (eigenvalues). Find eigenvalues λ and corresponding eigenvectors v such that: $Cv = \lambda v$
4. **Sort Eigenvalues and Eigenvectors:** Rank principal components by importance. Arrange eigenvalues in descending order and reorder the eigenvectors accordingly.
5. **Select Principal Components:** Choose the most significant components for dimensionality reduction.
6. **Transform the Data:** Reduce dimensionality while preserving maximum variance.

C. K-Nearest Neighbors

The K-Nearest Neighbors (K-NN) algorithm is a simple, non-parametric method used for classification and regression tasks. It operates on the principle of identifying the closest data points (neighbors) to a given query point and using their information to make predictions. K-Nearest Neighbors (KNN) algorithm is the best known, old, simple and effective pattern classification methods proposed by Thomas. M. Cover and Peter. E. Hart, where the sample data point is located and the nearest neighbor is determined by the k value.

1. **Choose the Number of Neighbors (K):** Select a positive integer K. The choice of K influences the algorithm's performance and generalization.
2. **Select a Distance Metric:** Define how to measure the similarity between data points.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

3. **Compute the Distance between Data Points:** Measure the distance between the query point and all other points in the training set.
4. **Identify the K Nearest Neighbors:** Find the K closest data points to the query point. Sort the training points by their distance to the query point and select the K nearest ones.

5. **Determine the Prediction:**
   Classification: Assign a class label to the query point based on its neighbors.
   Regression: Predict a continuous value for the query point.
6. **Evaluate the Model:** Assess the effectiveness of the K-NN model. Evaluate performance using metrics such as accuracy, precision, recall (for classification), or mean squared error (for regression) on a validation or test set.

D. Support Vector Machine

The Support Vector Machine (SVM) is a robust supervised learning algorithm used for both classification and regression tasks. It operates by identifying the optimal hyperplane that best separates different classes within the feature space. Support Vector Machines (SVM) is a non-parametric classification method based on statistical learning theory presented in Vapnik, (1999). The key steps involved in the SVM algorithm are as follows:

1. **Prepare the Data:** Organize and preprocess the data for effective training.
2. **Choose a Kernel Function:** Define a function to transform the data into a higher-dimensional space.
3. **Construct the Optimization Problem:** Formulate the problem to find the optimal separating hyperplane.

$$\text{Minimize } \frac{1}{2}\|\mathbf{w}\|^2$$

$$y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1 \text{ for all } i$$

where w is the weight vector, b is the bias, and y_i is the class label for the training sample x_i.

4. **Solve the Optimization Problem:** Determine the optimal parameters for defining the hyperplane.
5. **Construct the Decision Function:** Define the function to classify new data points. Formulate the decision function based on the learned parameters:

$$f(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + b)$$

6. **Classify New Data Points:** Predict the class of unseen data points. Apply the decision function to new data points to assign them to one of the classes.
7. **Evaluate the Model: Measure** the performance of the SVM model. Use **metrics** such as accuracy, precision, recall, F1 score, or area under the ROC curve (AUC) to evaluate the model's performance on a validation or test set.

E. Decision Tree

Building a decision tree involves a series of steps that enable the model to make decisions based on the input data. Below is a detailed, step-by-step guide to constructing a decision tree:

- **Begin with the Entire Dataset**: **Input:** The complete dataset DDD with features and target labels. **Output:** The root node of the tree.
- **Identify the Best Feature to Split:** Calculate the impurity of the dataset. **For each feature:** Evaluate all possible splits. Calculate the impurity reduction for each potential split.
- **Create a Decision Node:** Based on the optimal split, create a decision node. **Split the dataset** into two subsets: (i) Left child: Records that satisfy the split condition. Right child: Records that do not satisfy the split condition.

1. **Repeat the Process Recursively: For each child node:** If the node is pure or meets a stopping criterion, make it a leaf node.

2. **Apply Stopping Criteria**
   - **Leaf Node:** Stop splitting if any of the following conditions are met: All records in the node belong to the same class (for classification).
3. **Assign Class Labels (or Prediction Values)**

For Classification Trees: Assign the most common class label in the node as the prediction.
For Regression Trees: Assign the mean (or median) value of the target in the node as the prediction.

4. **Pruning (Optional)**
   Post-Pruning: After the tree is fully grown, remove branches that add little value **Pre-Pruning:** Limit tree growth during construction based on predefined conditions
5. **Finalize the Model**: After the dataset is fully split according to these rules, use the final tree to make predictions on new data points by traversing the tree based on feature values.

F. Artificial Neural Network

Artificial Neural Network (ANN) is a supervised machine learning method that a well-known solution for image classification problems. An ANN consists of input, hidden and output layers respectively. An Artificial Neural Network (ANN) is a computational framework inspired by the human brain, comprising layers of interconnected neurons that process input data to generate an output. Below is a step-by-step guide for constructing and training an ANN:

1. Initialize the Network

Input: Define the network architecture, including: Number of layers (input, hidden, and output layers). Number of neurons in each layer. **Weights and Biases:** Randomly initialize weights and biases for each neuron in the network.

2. Forward Propagation

Input: Feed input data into the input layer. **For each layer:** Compute the weighted sum of inputs and add the bias. Apply an activation function to the weighted sum to produce the neuron's output. Pass the output A to the next layer as input.
Output: The final output is the result of the output layer's activation.

3. Compute the Loss Function

Calculate the error between the predicted output and the actual target using a loss function (e.g., Mean Squared Error for regression, Cross-Entropy for classification).

4. Backpropagation

Minimize the loss by adjusting weights and biases.
Compute the gradient of the loss with respect to each weight and bias using the chain rule. **Output Layer:** Calculate the gradient of the loss with respect to the output layer's weights and biases. **Hidden Layers:** Backpropagate the error through the network, calculating gradients for each layer. **Update Weights and Biases:** Adjust weights and biases using the gradients

5. Iterate (Training Process)

Repeat steps 2-4 for multiple epochs until the model converges or a stopping criterion is met.

6. Evaluation: Validation

Evaluate the trained network's performance on a validation dataset.
Metrics: Use metrics like accuracy, precision, recall, F1-score for classification, or R-squared for regression to assess the model.

7. Testing
After training, test the network on a separate test dataset to evaluate its generalization ability.

8. Deployment
Once the model's performance is satisfactory, deploy the ANN to make predictions on new, unseen data.

G. Generative Adversarial Networks (GANs)
Generative Adversarial Networks (GANs), first introduced by Ian Goodfellow and his team in 2014, are a type of deep learning model that consists of two neural networks: the generator and the discriminator, which are trained simultaneously through an adversarial process. This technique has garnered considerable attention for its capacity to produce high-quality synthetic data, such as images, text, and audio. The following provides an in-depth examination of GANs, covering their architecture, training methodology, applications, challenges, and potential future developments.
Goodfellow et al. (2014): The seminal paper "Generative Adversarial Nets" by Ian Goodfellow and his colleagues introduced the GAN framework, describing the basic adversarial model where a generator and a discriminator are trained simultaneously. The generator's goal is to create realistic data samples, while the discriminator's role is to distinguish between real and generated data. This work established GANs as a foundational and powerful tool for generative modeling discussed by Sampath et. al. in 2021.

1. Initialize the Networks
- Generator (G)
- Discriminator (D)

2. Prepare the Training Data: Define a random noise distribution, often Gaussian or uniform, from which the generator will draw its input.
3. Training Process: GAN training involves alternating between training the discriminator and the generator.
Step 3.1: Train the Discriminator
- Real Data
- Fake Data
- Update the Discriminator

Step 3.2: Train the Generator
- Generator Objective
- Update the Generator

4. Iterative Training: Repeat the training steps (3.1 and 3.2) for many iterations (epochs).

5. Monitor the Training Process
- **Loss Values**: Track the loss values of both the generator and discriminator over time. Ideally, the discriminator's accuracy should be around 50% when the generator is producing highly realistic data.
- **Generated Samples**: Periodically generate samples from the generator and visually inspect them (if applicable, like with images) to monitor progress.

- **Convergence**: Training continues until the generator's outputs are indistinguishable from real data, or until the loss values stabilize, indicating that the networks have reached a state of equilibrium.

6. **Evaluation:** After training, evaluate the quality of the generated data using metrics such as the Inception Score (IS) or Fréchet Inception Distance (FID).
7. **Final Model Usage:** Once training is complete, the generator can be used independently to produce new, realistic data samples.

H. Evolutionary Multiobjective Optimization (EMO)

Miettinen, K. (1999) and Deb, K., et al. (2013): Evolutionary Multiobjective Optimization (EMO) is a research area that deals with solving optimization problems involving multiple conflicting objectives using evolutionary algorithms. Evolutionary Multiobjective Optimization (EMO) is a specialized area within evolutionary computation that addresses optimization problems with multiple, often conflicting, objectives. Unlike single-objective optimization, which aims to find the best solution based on a single criterion, multiobjective optimization aims to identify a set of optimal trade-offs, known as the Pareto-optimal front. In this context, no objective can be improved without causing a deterioration in at least one other objective. Various steps involved in this algorithm were discussed Wang, et. al. in 2023.

1. Problem Formulation
2. Initialization
3. Selection
4. Variation Operators
5. Evaluation
6. Replacement
7. Termination
8. Analysis and Interpretation

5. Experimental Results and Analysis

A confusion matrix is the most widely used tool for assessing the performance of models trained on datasets with predefined target labels in machine learning.

True Positive (TP) Indicates that the predicted class of the test data matches the actual class. **False Negative (FN)** Indicates that the predicted class of the test data does not match the actual class. **False Positive (FP)** Indicates that the data was incorrectly classified as positive when the actual value is negative. **True Negative (TN)** Indicates that the data was correctly classified as negative when the actual value is negative.

Accuracy refers to the proportion of correctly diagnosed nodules out of the total number of nodules. Precision is the proportion of correctly classified positive nodules out of all positive predictions. Recall measures the proportion of correctly classified positive nodules out of all actual positive cases. F-measure is the harmonic mean of precision and recall. We created a confusion matrix for each technique and evaluated accuracy, precision, recall, and F-measure to compare the performance of different machine learning techniques, as presented in Table 1 and Fig. 3 to Fig.

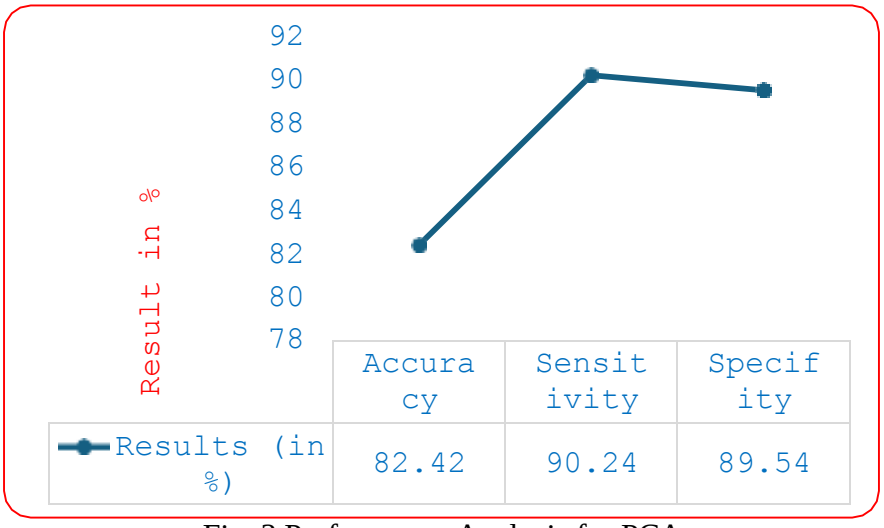

Fig. 3 Performance Analysis for PCA

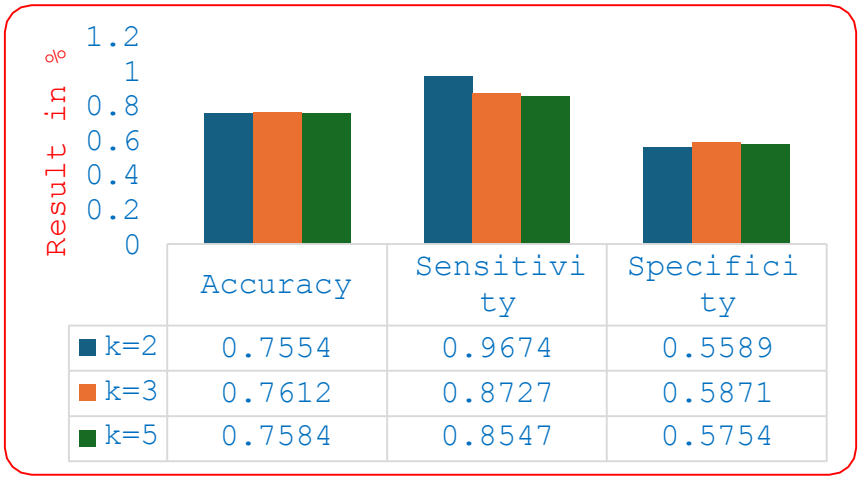

Fig. 4. Performance Analysis for KNN

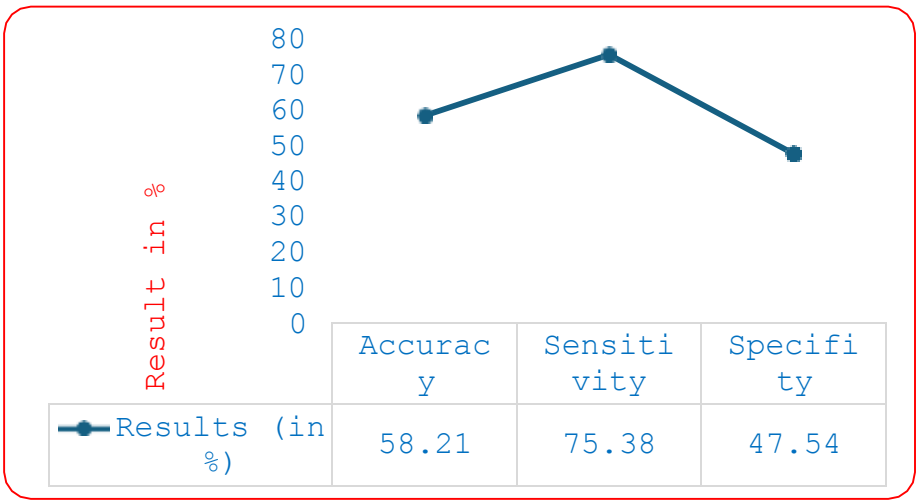

Fig. 5 Performance Measurements of SVM

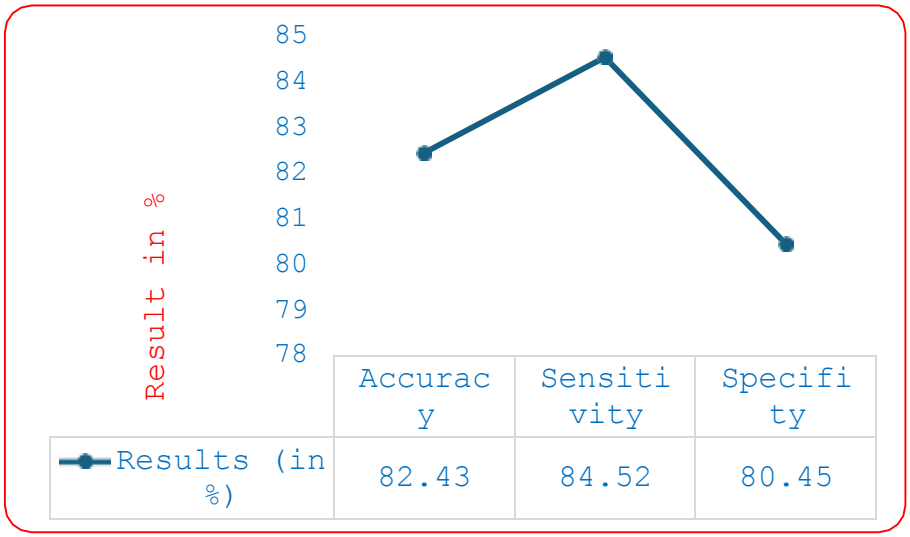

Fig. 6 Performance Measurements of DT

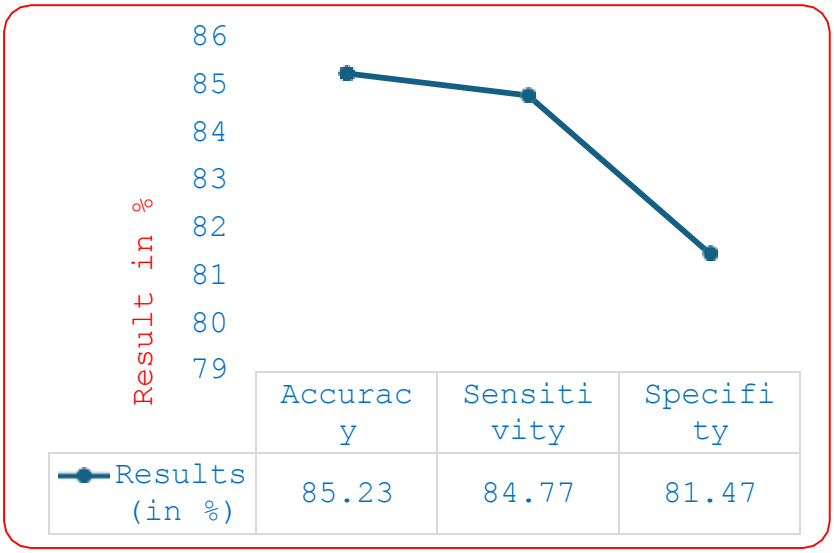

Fig. 7 Performance Measurements of ANN

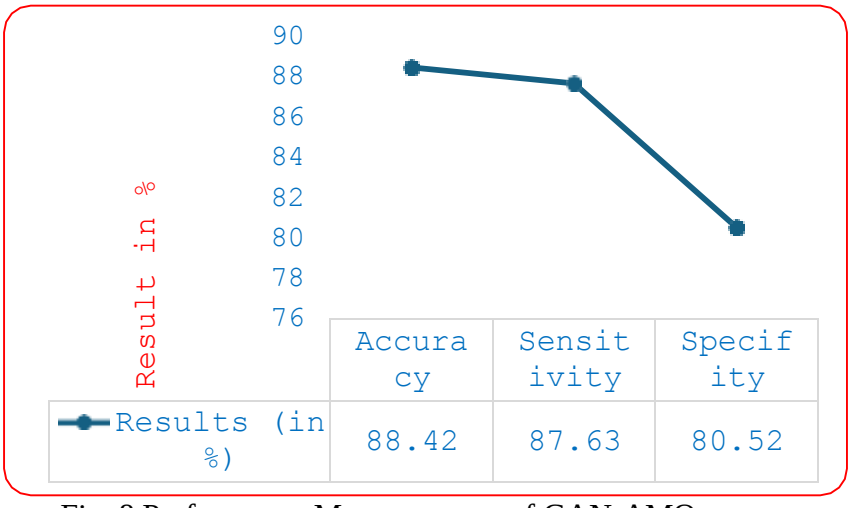

Fig. 8 Performance Measurements of GAN-AMO

Table 1. Classification Methods with Performance (in %)

Performance	KNN	SVM	DT	ANN	Eigen Vectors	GAN-EMO
Accuracy	78.34	58.21	82.43	85.23	82.42	88.42
Precision	84.21	48.32	73.42	94.20	85.10	95.10
Recall	81.74	75.88	90.84	85.24	90.24	92.24
f-measure	83.45	59.44	81.54	90.87	88.54	92.54

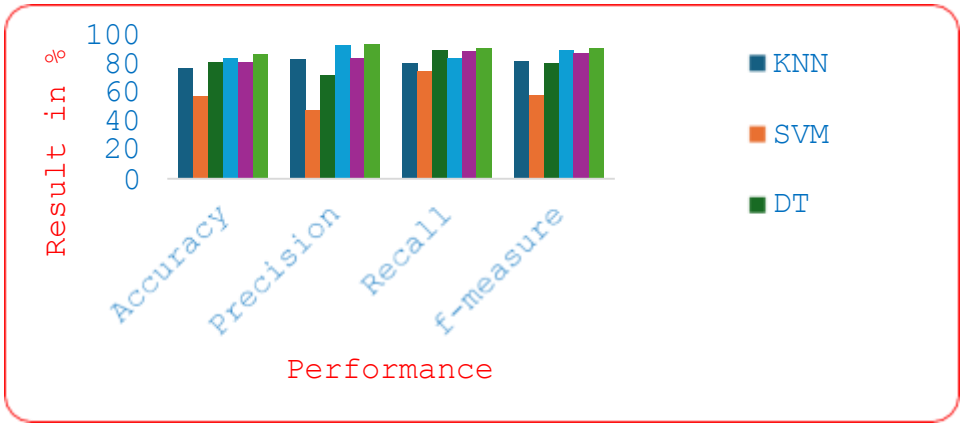

Fig. 8 Classification Methods with Performance (In %)

6. Conclusion

In this study, we employed various machine learning and deep learning methods to detect lung cancer from CT images. We also utilized ML techniques, such as PCA, to reduce the dimensionality of the images, which can lead to some feature loss. However, reducing dimensions also helped improve the accuracy when using KNN (with k=2 and k=3) and SVM, with minor accuracy trade-offs that were acceptable for reducing time complexity and saving storage. Due to the large data volume, we were unable to apply Naïve Bayes and a 10-layer Feed Forward Neural Network without PCA, despite neural networks generally achieving higher accuracy than other machine learning methods. The Decision Tree method yielded the highest performance metrics on the original data, although some performance measures decreased after pre-processing. This research demonstrates that using different machine and deep learning approaches, GAN provided the best results for lung cancer detection with better performance. Future research could explore applying similar deep learning approaches with different optimization techniques to enhance detection performance further.

7. References

1. Han, J., Pei, J., & Kamber, M. (2011). Data Mining: Concepts and Techniques (3rd ed.). Elsevier.
2. Mitchell, T. M. (1997). Machine Learning. McGraw-Hill.
3. Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
4. Afshar, P., Plataniotis, K. N., & Mohammadi, A. (2020). Capsule networks for lung cancer screening and diagnosis. Scientific Reports, 10(1), 1-11. https://doi.org/10.1038/s41598-020-69049-9

5. Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., & Tse, D. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. Nature Medicine, 25(6), 954-961. https://doi.org/10.1038/s41591-019-0447-x
6. Balagurunathan, Y., Wang, H., Kumar, V., Gu, Y., Aerts, H. J. W. L., & Gillies, R. J. (2019). Reproducibility and prognosis of quantitative features extracted from CT images. Translational Oncology, 12(9), 1086-1096.
7. https://doi.org/10.1016/j.tranon.2019.06.012
8. Bhatia, A., & Verma, K. (2021). A survey of deep learning techniques for lung cancer detection from CT scans. Artificial Intelligence Review, 54(3), 1923-1955. https://doi.org/10.1007/s10462-020-09915-2
9. Coudray, N., Ocampo, P. S., Sakellaropoulos, T., Narula, N., Snuderl, M., Fenyo, D., & Tsirigos, A. (2018). Classification and mutation prediction from non–small cell lung cancer histopathology images using deep learning. Nature Medicine, 24(10), 1559-1567. https://doi.org/10.1038/s41591-018-0177-5
10. Dhaware, B. U., & Pise, A. C. (2020). Lung cancer detection using Bayesian classifier and FCM segmentation. IEEE International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT), 2020, 170–174. https://doi.org/10.1109/ICACDOT.2019.9029382
11. Giger, M. L., & MacMahon, H. (2019). Deep learning in lung cancer screening: A new era of clinical imaging and interpretation. Journal of Thoracic Imaging, 34(5), 1-8. https://doi.org/10.1097/RTI.0000000000000406
12. Huang, G., Liu, Z., Maaten, L., & Weinberger, K. Q. (2021). Densely connected convolutional networks for lung cancer detection. IEEE Transactions on Medical Imaging, 40(5), 1061-1073. https://doi.org/10.1109/TMI.2021.3068546
13. Liu, H., Liu, J., Zheng, C., Zhang, C., & Xia, Y. (2021). Deep learning in medical image analysis: Techniques and applications in lung cancer detection. IEEE Access, 9, 155736-155747. https://doi.org/10.1109/ACCESS.2021.3131897
14. Makaju, S., Prasad, A. A., & Elchouemi, S. (2019). Lung cancer detection using CT scan images. Procedia Computer Science, 125, 107–114. https://doi.org/10.1016/j.procs.2018.12.017
15. Shen, W., Zhou, M., Yang, F., Yu, D., Dong, D., Yang, C., & Tian, J. (2019). Multi-crop convolutional neural networks for lung nodule malignancy suspiciousness classification. Pattern Recognition, 61, 663-673. https://doi.org/10.1016/j.patcog.2016.07.031
16. Tan, C., Li, Z., Hu, J., & Xu, Y. (2020). A novel attention-based aggregation CNN for tumor segmentation and classification in lung cancer CT images. IEEE Transactions on Medical Imaging, 39(6), 2049-2060. https://doi.org/10.1109/TMI.2020.2971189
17. Wang, H., Zheng, B., Li, Z., & Yuan, M. (2021). An interpretable CNN-based model for lung cancer prediction using 3D chest CT scans. Nature Communications, 12(1), 1-10. https://doi.org/10.1038/s41467-021-21030-z
18. Xie, Y., Zhang, J., Xia, Y., Fulham, M., & Zhang, Y. (2019). Fusing texture, shape, and deep model-learned information at multi-scale for COVID-19 diagnosis of chest CTimages. Pattern Recognition, 106, 107311. https://doi.org/10.1016/j.patcog.2020 107311
19. Zhang, H., Chang, Z., Xie, J., Jin, L., & Liu, D. (2021). A comprehensive review of deep learning-based methods for the diagnosis of lung cancer. Journal of Biomedical Informatics, 120, 103863. https://doi.org/10.1016/j.jbi.2021.103863

20. Shiraishi, S., Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi. (2000). Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules. Am. J. Roentgenol., 174, 71-74.
21. Vapnik, V., (1999). An overview of statistical learning theory. IEEE Transactions on Neural Networks, 10(5), 988-999.
22. Goodfellow, I., et al. (2014). Generative Adversarial Nets. NeurIPS.
23. Günaydin, Özge, Melike Günay, and Öznur Şengel. "Comparison of lung cancer detection algorithms." 2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT). IEEE, 2019.
24. J. Shiraishi, S. Katsuragawa, J. Ikezoe, T. Matsumoto, T. Kobayashi, K. Komatsu, M. Matsui, H. Fujita, Y. Kodera, and K. Doi, "Development of a digital image database for chest radiographs with and without a lung nodule: Receiver operating characteristic analysis of radiologists' detection of pulmonary nodules," Am. J. Roentgenol., vol. 174, pp. 71– 74, 2000.
25. Sampath, V., Maurtua, I., Aguilar Martin, J. J., & Gutierrez, A. (2021). A survey on generative adversarial networks for imbalance problems in computer vision tasks. Journal of big Data, 8, 1-59.
26. Miettinen, K. (1999). Nonlinear Multiobjective Optimization. Springer.
27. Deb, K., et al. (2013). "Evolutionary Multi-Objective Optimization: Past, Present, and Future." IEEE Computational Intelligence Magazine, 8(1), 4-17.
28. Wang, Z., Pei, Y., & Li, J. (2023). A survey on search strategy of evolutionary multi-objective optimization algorithms. Applied Sciences, 13(7), 4643.