ENHANCING LUNG CANCER PREDICTION USING MACHINE LEARNING: A COMPARATIVE ANALYSIS OF HYPERPARAMETER OPTIMIZATION TECHNIQUES

Luxshi K[1] Prasanth S[2] Chandrika Malkanthi [1] and R.M.K.T.Rathnayaka [1]

[1]Department of Physical Science and Technology, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka
[2] Memorial University of Newfoundland and Labrador, Canada
klluxshi99@gmail.com

Abstract. Lung cancers are identified as one of the lethal diseases by medical professionals due to delays in diagnosis leading to high mortality rates. Early detection of lung cancer improves survival probabilities but standard diagnosis methods entail high expenses and lengthy examination times with susceptibility to human errors. Thus, this study aims to automate lung cancer prediction using machine learning and deep learning models utilizing a dataset with 16 numerical attributes. GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost, and DL models like CNN, MobileNet and Swin Transformer were tested utilized hyperparameter tuning together with cross validation approaches. The XGBoost model reached the highest accuracy of 0.9968 during cross-validation tests for k-fold, stratified k-fold (k=5) and Leave one out methods. XGBoost and Gradient Boosting demonstrated optimal performance after hyperparameter tuning since they achieved 0.9968 accuracy for both training and testing sets although the total training time was different. CNN demonstrated powerful performance throughout its training and testing stages with the fastest training time in deep learning models and accuracy values of 0.9829 and 0.9872. Ensemble ML methods and optimized DL models effectively predict lung cancer. The researchers plan to incorporate large-scale data platforms in future research to enhance the predictive performance of the system.

Keywords: Cross validation, Deep learning models, Evaluation metrics, Hyperparameter tuning, Machine learning models

1 Introduction

Medical science categories lung cancer as a worldwide leading fatal illness because patients receive a delayed diagnosis leading to unacceptably high death rates [1]. The survival changes of patients improve dramatically when lung cancer exists at an early stage even though existing diagnostic approaches demand expensive and slow tests that include biopsies and x-ray examinations and CT scans but entail human error. Machine

Learning (ML) enables a promising predictive approach through its ability to use numerical patient data for assessing lung cancer probability [1, 2]. The research field lacks sufficient knowledge about how the hyperparameters tuning methods along with cross validation techniques impact the performance of existing ML models [2]. The research study to improve lung cancer prediction through a comparison of different ML and Deep Learning (DL) models following hyperparameter optimization and cross-validation strategies. The research minimizes its dependence on sophisticated imaging methods because it deals solely with numerical data which enhances accessibility and decreases expenditures for detecting lung cancer. Advance ML algorithm performance assessments lead to the selection of the most precise and dependable model for medical use.

Different ML and DL models used for lung cancer diagnosis assessment requires evaluation of their predictive capacities. The objectives of the study is to evaluate model accuracy along with generalization capabilities through changes in hyperparameter values. The goal is to study cross-validation methods when used to stop overfitting while developing reliable predictive models. A predictive model selection for lung cancer diagnosis requires assessment of multiple criteria including accuracy, sensitivity, and specificity and ROC-AUC metrics to determine the most suitable advance ML approach.

The current diagnostic system for lung cancer relies on two types of clinical data including numerical patient data consisting of demographics and lifestyle factors and clinical information to determine disease presence. Numerical patient data analysis for lung cancer diagnostic models requires additional examination and optimization despite the existing research with image-based data such as CT scans [2]. The published research mostly deposits generic model with model-specific characteristics. An insufficient evaluation process may result in wasting diagnostic potentials. The aim of this dissertation is to bridge the assessment gap through systematic algorithm evaluation on numerical patient information to determine an optimal diagnostic model for lung cancer.

Research Questions

1. How does hyperparameter tuning affect the performance of different ML and DL models in lung cancer prediction?
2. What is the impact of various cross-validation techniques on model accuracy and robustness?
3. Which ML or DL model demonstrates the highest predictive accuracy when analyzing numerical patient data?

Related Work

The paper examines multiple data mining methods used for lung cancer prediction in detail. The healthcare field uses data mining techniques to develop various useful applications that help discover valuable knowledge from medical information. In this case the use of classification-based data mining techniques such as Rule Based, Decision Tree, Navie Bayes and Artificial Neural Network (ANN) to massive volume of

healthcare data. Lung cancer prediction can be achieved by analyzing age, sex, wheezing, shortness of breath and shoulder, chest, arm pain symptoms. The proposed system aims to detect cancer early and verify its nature to help physicians save patient lives [1].

Medical practitioners use ML techniques because they achieve accurate results when monitoring the development of cancerous diseases. Lung cancer analysis and prognosis in healthcare utilize different types of ML including GNB, SVM, Logistic Regression (LR) and ANN [2].

Radiomics implies automatic extraction of medical image-based quantitative features which researchers widely investigate for lesion classification applications. This work reviewed the primary methods which classify nodules and predict lung cancer when analyzing Computed Tomography (CT) imaging data. Convolution Neural Network (CNNs) trained with deep learning approaches deliver the current best performance which reaches classification areas of 90s Area Under Curve (AUC) points after sufficient training data becomes available. During system performance assessment analysts should understand the data limitations present in the validation and training datasets. The research included patients who used cigarettes or not and what type of cancer treatment history these patients possessed. Avg. area under the curve results from SVMs alongside Random Forest when processing advanced sets of features which surpasses traditional learning techniques in producing enhanced outcomes [3].

This paper analyzes the accuracy ratios of SVM, KNN and CNN as classifiers for early lung cancer diagnosis to save numerous lives. This examination used informational indexes from the UCI dataset which includes patients who received lung cancer diagnosis. Weka Tool serves as the basis for this paper to evaluate the accuracy performance of classification methods. The experimental findings indicate SVM achieves the most outstanding results with 95.56% accuracy followed by CNN which reaches 92.11% and KNN performs with 88.40% accuracy [4].

The analysis evaluates lung cancer incidence rates of males and females across ten European countries through Support Vector Regression (SVR), Backpropagation and Long-short Term Memory Network (LSTM) before performing predictions. The prediction results undergo evaluation through the most effective assessment metrics from previous literature which include Mean Square Error (MSE), Coefficient of Determination (R^2) and Explained Variance (EV) scores. The prediction outcomes achieved success for all used algorithms but SVR delivered the best performance through minimal error numbers along with superior results. The prediction performance analysis included Backpropagation as the second choice after SVR followed by LSTM [5].

We have explored and compared various ML algorithms for lung cancer prediction, including XGBoost, Light BGM, AdaBoost, Logistic Regression and SVM. The anaysis revealed that XGBoost consistently outperformed the order models in terms of accuracy, sensitivity, specificity and F1-score achieved 97.50%, 96.80%, 98% and

97.50%. While Light BGM also showed string results and remains a viable alternative, AdaBoost, Logistic Regression, and SVM exhibited relatively lower performance metrics, suggesting that XGBoost and Light BGM are the most suitable choices for clinical applications requiring accurate and reliable predictions [6].

Significant of this study

In this study, ML models like Logistic regression with Decision Tree along with Random Forest, Gradient Boosting, XGBoost, SVM, GNB as well as advanced deep learning models like CNN, Mobile Net and Swin Transformer were compared in order to identify the best approach for automating lung cancer prediction. We focus on hyperparameter tuning and cross validation methods such as hold-out method, k-fold cross validation, stratified k-fold and leave-one-out method. Diagnostic accuracy was improved from model comparison through measurements including evaluation metrics like accuracy, sensitivity, specificity, confusion matric and Area under Curve- Receiver Operating Characteristics (AUC-ROC). There is no clear comparison to performance optimization in lung cancer prediction existing studies, especially in addressing certain issues such as shows weaknesses utilization of small or confined datasets together with its restricted usage of deep learning models as well as its narrow observation of accuracy performance without sufficient evaluation metrics. Most studies failed to implement appropriate validation approaches as well as parameter optimization methods while neglecting computational system performance. Advanced medical applications receive better predictive capabilities when traditional methods along with modern DL and ML models are jointly used in analysis. The analytic methods show successful integration which indicates their usefulness for healthcare implementations in real-world practice.

2 Methodology

The research methodology used in this study involves the creation of a lung cancer prediction system using comprehensive analysis of hyperparameters and cross validation methods through ML and DL models. The methodology describes each stage of the study including the approach, data gathering, data preprocessing, and model development process with evaluation techniques. The systematic process includes activities for data acquisition followed by data preparation model creation before moving to performance evaluation and assessment of model results against other models. Below the Fig 1 shows the high-level architecture of ML techniques.

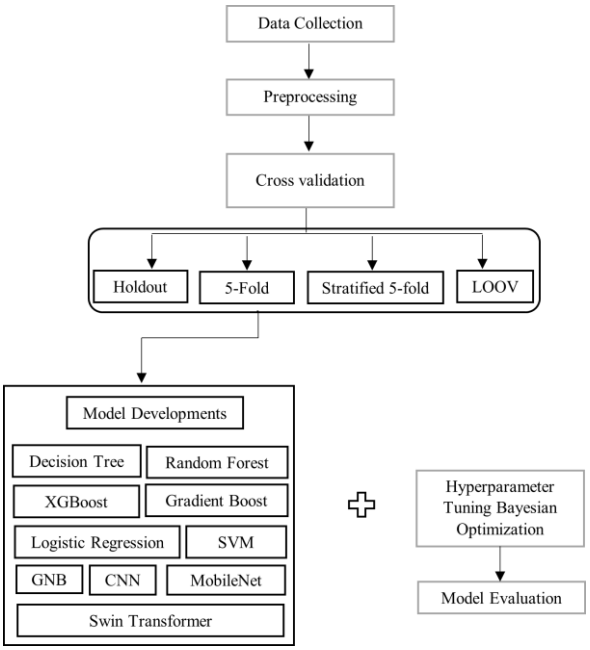

Fig. 1. High level Architecture

2.1 Data gathering and data preprocessing

A set of numerical patient data contains demographic statistic in combination with smoking data and respiratory symptoms regarding cough and shortness of breath was acquired. The data collection contains 16 attributes features covering 5872 records [7]. As a part of data preparation preprocessing deals with processing raw data through diverse methods to get it ready for subsequent data processing tasks [8]. Data preprocessing involves different methods that consist of extracting representative data samples from large populations and developing a single input from raw data while removing data noise. The preparation process requires data preprocessing to describe all the data processing mechanisms that run raw data ready for subsequent processing [5, 8]. The data preprocessing process depends on various methods and tools which consist of:

- Sampling: Selects the representative subset from a large population of data to transforms original raw information into one unified input stream.
- Denoising: Removes noise from data. The process of imputation generates statistical data estimates when values are missing from the information set.
- Normalization: Organizes data for more efficient access.

2.2 Data Splitting methods

Machine learning models need proper evaluation through cross-validation techniques which assess the extent to which they perform on new data. The simplicity of the Hold-out method comes from its single partitioning of data into training and test sets yet its unreliable performance remains its main drawback [9, 10]. The data splits into 'k' equivalent sections using K-Fold cross validation so each subset operates as validation data alongside training data that consists of remaining sections thus achieving more dependable results [2, 7]. The stratification of K-Fold cross validation performs dataset stratification to maintain proportional class distribution between each fold which benefits datasets with imbalanced classes [11, 12]. Inside Leave-One-Out cross validation (LOOCV) each data point serves as the test sample once because 'k' matches the sample count yet this method proves accurate with small amounts of data while being expensive to compute and yielding high variance results. Table 1 shows the operations of each validation methods [6, 13].

Table 1. Cross Validation Methods

Technology	Operation Steps	Advantages	References
Holdout Method	Researchers allocate data into training and testing sections with a preset value ratio. They usually use 80:20 or 70:20 ratios.	It is suitable for large dataset and also easy to implement.	[2]
K-Fold cross validation	The original dataset becomes separated into 'K' equal parts for analysis.	The entire dataset functions as the training set as well as the validation set.	[13]
Stratified K-Fold cross validation	The approach functions identically to K-fold cross validation where every split maintains equal class proportion distributions across the whole data structure.	It works better when dealing with imbalanced datasets. The distribution of each classes fold is similar.	[5, 14]
Leave-One-Out cross validation	The research employs one sample for validation purposes before using the n-1 remaining samples for training.	All data samples are used both training and validation samples.	[8]

2.3 Hyperparameter Tuning

Bayesian optimization operates as a powerful technique for hyperparameter tuning because it successfully identifies optimal values through efficient exploration. The algorithm uses Gaussian Processes as a basis to represent prior understanding and forecast how system performance will change in different input regions [15]. The search process receives guidance from a posterior distribution which Bayes' theorem calculates for its operations. This strategy combines exploration of areas with high uncertainty with exploitation of areas with high expected accuracy which changes from early exploration to late exploitation in different iterations [15].

The optimization mechanism in Bayesian theory bases its foundation on Bayes' Theorem as presented by Eq (1) [4].

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)} \tag{1}$$

The prior probability P (A|B) can be described by the product of likelihood P (B|A), prior probability P (A), and evidence P (B). In this equation the term P (A) denotes our prior belief regarding model A together with P (B) which represents the probability distribution of observation B. The observation affects model probabilities through P (A|B) combined with P (B|A) describing mutual influence between observation and model. In a simplified form the normalization factor P (B) becomes unnecessary so the statement becomes according to Eq (2).

$$P(A|B) = P(B|A)P(A) \tag{2}$$

2.4 Model Developments

This study considers binary classification of lung cancer data using seven machine learning models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, GNB, and SVM with and without Bayesian optimization for tuning hyperparameters. The Logistic Regression approximates lung cancer probability through a linear separation boundary that optimizes its ability to adjust the regularization parameter (C) [16]. Decision Tree creates a tree structure through repeated feature space splitting that optimizes both maximum depth and minimum splitting data points. Random Forest uses multiple decision trees to gather predictions and minimizes overfitting through number of tree and maximum depth optimization [17]. Gradient Boosting constructs trees in series where subsequent models repair errors in preceding models through three main parameter adjustments that include maximum depth and learner rate as well as number of estimators. The optimized gradient boosting system XGBoost offers better performance through its addition of regularization techniques and parallel processing mechanisms which need similar optimizations [17, 18]. The GNB model implements a statistical technique that makes independence assumptions between features while using Gaussian distributions for probability prediction through variance smoothing optimization. The SVM algorithm detects the most suitable hyperplane

boundary between class distinctions through its radial basis function kernel while it requires parameter adjustments of "C" together with "gamma" [18]. The training and validation of each model occurred with scaled features while the evaluation used various metrics such as accuracy and precision in addition to recall and F1-score and AUC. Models were validated with k-fold stratified k-fold and leave-one-out cross-validation after applying Bayesian optimization to find the best hyperparameters [19]. Table 2 shows that Hyperparameter used in tuning process.

Table 2. Hyperparamter Used in the Tuning process

Model	Hyperparameters Tuned
Logistic Regresion	C (Regularization parameter, range: 10^{-3} to 10^{2})
Decision Tree	Max_depth (Maximum Tree depth, range: 3 to 1), min_samples_split(range: 2 to 10)
Random Forest	N_estimators (Number of trees, range: 50 to 200), max_depth (range: 5 to 20)
Gradient Boosting	N_estimators (range:50 to 200), learning rate (range: 0.01 to 0.2), max_depth (range: 3 to 10)
XGBoost	N_estimators (range: 50 to 200), learning rate (range: 0.01 to 0.2), max_depth (3 to 10)
GNB	Var_smoothing (Smooting parameter, range: 10^{-9} to 1)
SVM	C (Regularization parameter, , range: 10^{-2} to 10^{2}), gamma (Kernel coefficient, range: 10^{-3} to 10^{1})
CNN	Filters1 (range: 64 to 256), filters2 (32 to 128), dense units (range: 64 to 256), dropout_rate (range: 0.3 to 0.7)
MobileNet	Dense_units (range: 64 to 256), dropout_rate (range: 0.3 to 0.7)
Swin Transformer	Dense_units (range: 64 to 256), dropout_rate (range: 0.3 to 0.7)

Fig 2 shows that, a compact neural network designed for binary classification exists as the CNN structure and its variant with Bayesian Optimization. The model without Bayesian Optimization includes two convolutional layers equipped with fixed filters at 32 and 64 strength and 3x3 kernels and ReLU activation and same padding then max-pools using 2x2 layers. After flattening the output the model utilizes 128 units with ReLU activation followed by a dropout layer with 0.5 rate before a sigmoid activation dense layer performs binary output [18]. Using Bayesian Optimization maintains the model structure intact yet allows the adjustable hyperparameters filters1 (16–64), filters2 (32–128) and dense_units (64–256) and dropout_rate (0.3–0.7) to optimize validation accuracy. The model contains two iterations with Adam optimizer (0.001 learning rate) implementing binary crossentropy as loss function until reaching 10 training epochs for accuracy evaluation [17, 20].

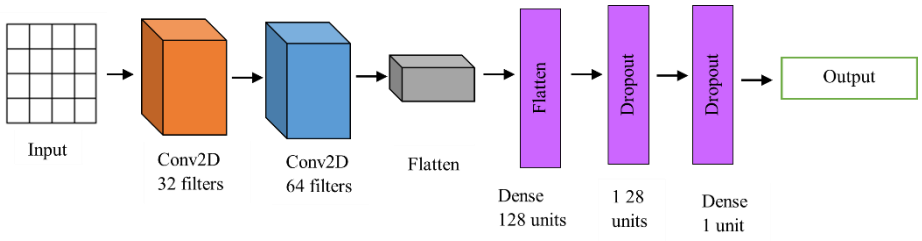

Fig. 2. CNN model Architecture

As shown Fig 3 MobileNet serves as a lightweight transfer learning model for binary classification which employs Bayesian Optimization or functions without it. Without Bayesian Optimization the MobileNetV2 base (frozen ImageNet weights) operates on a 32x32x3 input which is followed by global average pooling and three sequential layers: 128 units with ReLU activation and 0.5 dropout and sigmoid output [13, 20]. Bayesian Optimization optimizes the dense layer units between 64 and 256 units and dropout rates ranging from 0.3 to 0.7 to achieve maximum validation accuracy when applied to the identical model structure. The training process includes 10 epochs with Adam optimizer (learning rate set at 0.001) and binary crossentropy loss to reach an evaluation based on accuracy.

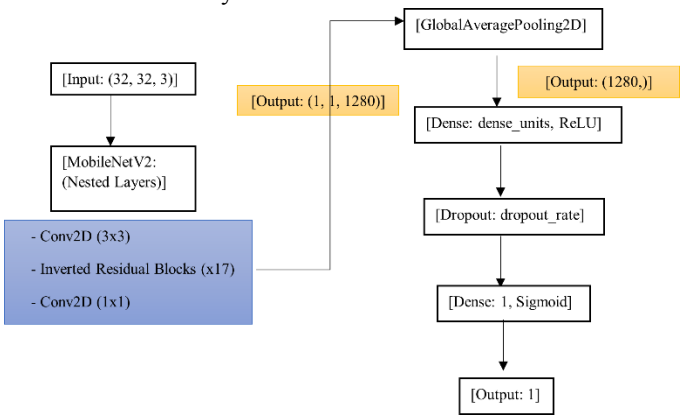

Fig. 3. Mobile Net model Architecture

As shown Fig 4 shown that, the Swin Transformer architecture for binary classification uses Bayesian Optimization to run either with or without its implementation of convolutional and transformer elements. The network architecture begins with 32x32x3 inputs treated by a 32-filter Conv2D with ReLU activation and same padding and then applies a 2x2 max-pooling layer before reshaping for sequence input followed by a custom Swin Transformer Layer with 32 dimensions, 4 heads and then executes global average pooling, dense layer with 128 units using ReLU activation followed by 0.5 dropout and a sigmoid output layer. In Bayesian Optimization the network uses the identical design yet the dense layer units' fall within 64-256 units and the dropout rate

ranges between 0.3 and 0.7 for optimizing validation accuracy. Two versions of the network use the Adam optimizer with a learning rate of 0.001 and binary crossentropy loss during 10 epochs of training before they evaluate models based on accuracy [21]. Fig 4 shows that Swin Transformer architecture.

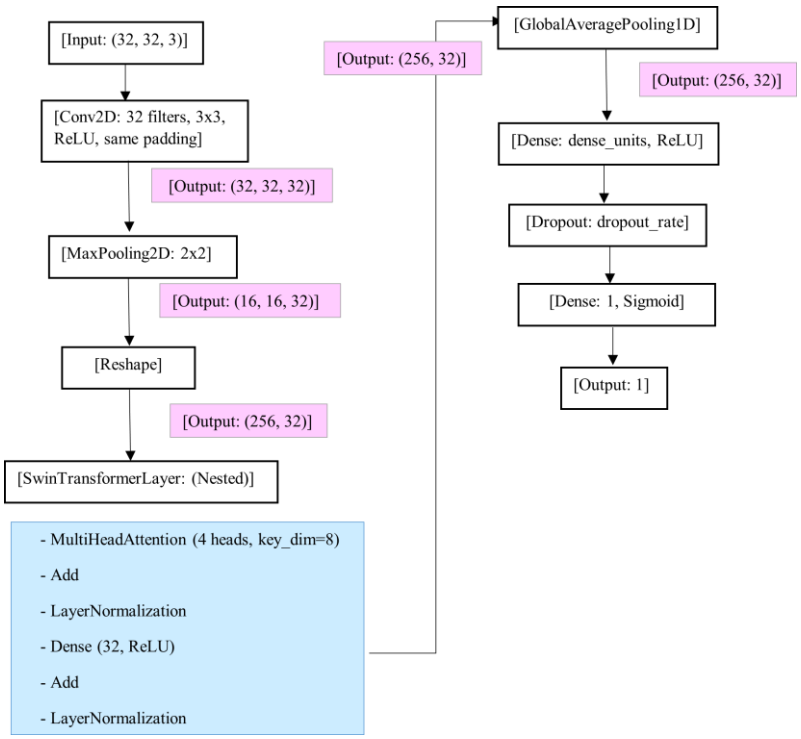

Fig. 4. Swin Transformer model Architecture

2.5 Model Evaluation

Different performance metrics exist to evaluate systems which perform either classifying tasks or tasks that require regression. Performance evaluation metrics including Accuracy, Precision and recall, F1-score with AUC-ROC must be used to assess Logistic regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVM, GNB along with CNN, Mobile Net and Swin Transformer classification models. The evaluation of model accuracy assists in assessing total performance yet precision indicates successful matching between predicted and actual positive outcomes alongside recall measurement that identifies detected actual positive results and the F1-score combines precision and recall evaluation and AUC-ROC measures across thresholds that proves fundamental for lung cancer diagnosis.

The AUC-ROC represents a performance assessment method for binary classification models which determines their capacity to differentiate between positive and negative outcomes [11]. An AUC-ROC exceeding 0.9 demonstrates an excellent model which accurately recognizes different classes amid low levels of classification mistakes. Model performance is good when the score lies between 0.8 and 0.9 even though there are some prediction errors. A model with scores between 0.7 to 0.8 demonstrates fair performance because it separates classes yet makes multiple misclassification errors. A predictive model shows poor performance when the AUC-ROC reaches values below 0.7 since it demonstrates weak abilities to detect class differences and performs at a level similar to basic random guessing. The model demonstrates superior performance based on its higher AUC-ROC value which indicates its ability to correctly label different classes regardless of threshold settings.

- AUC-ROC $>$ 0.9: Excellent Model
- 0.8 $\leq$AUC-ROC$<$0.9: Good Model
- 0.7$\leq$ AUC-ROC$<$0.8: Fair Model
- AUC-ROC$<$0.7: Poor model

3 Results and Discussion

The results from Tables 3 and 4 demonstrate that Bayesian Optimization enhances model performance since its implementation yields superior accuracy measurements. The accuracy of GNB, SVM, Gradient Boosting, XGBoost, CNN, and Mobile Net models increases substantially through the application of K-fold, Stratified K-fold, and Leave-one-out methods when Bayesian Optimization is implemented. The accuracy levels for SVM improved from 0.9772 to 0.9961 along with Gradient Boosting increasing from 0.9833 to 0.9889 and Mobile Net achieving 0.9778 to 0.9902. However, some models like Logistic Regression and Swin Transformer exhibit minimal or no improvement. The Bayesian Optimization approach dramatically improves model generalization quality along with resulting in reliable performance outcomes mainly in complex modeling scenarios.

Table 3. Results of cross validations without using Bayesian optimization

No	Model	K-fold method	Stratified K-fold	Leave-one-out
1	GNB	0.9097	0.9097	0.9069
2	SVM	0.9772	0.9772	0.9821
3	Logistic Regression	0.9456	0.9456	0.9461
4	Decision Tree	0.9957	0.9957	0.9968
5	Random Forest	0.9961	0.9957	0.9967
6	Gradient Boosting	0.9833	0.9833	0.9842
7	XGBoost	0.9961	0.9961	0.9968
8	CNN	0.9870	0.9870	0.9870

9	Mobile Net	0.9778	0.9778	0.9778
10	Swin Transformer	0.9418	0.9418	0.9418

Table 4. Results of cross validation using Bayesian Optimization

No	Model	K-fold method	Stratified K-fold	Leave-one-out
1	GNB	0.9172	0.9172	0.9171
2	SVM	0.9961	0.9961	0.9961
3	Logistic Regression	0.9463	0.9463	0.9461
4	Decision Tree	0.9957	0.9957	0.9968
5	Random Forest	0.9961	0.9961	0.9968
6	Gradient Boosting	0.9889	0.9909	0.9870
7	XGBoost	0.9968	0.9968	0.9968
8	CNN	0.9838	0.9838	0.9838
9	Mobile Net	0.9902	0.9902	0.9902
10	Swin Transformer	0.9369	0.9369	0.9369

The Table 5 and Fig 6 shows an assessment of different models that focuses on training accuracy along with testing accuracy and training duration. The traditional ML models including Decision Tree, Random Forest, Gradient Boosting, and XGBoost deliver outstanding performance that results in almost identical training and testing accuracies at 99.68% and requires quick training periods with XGBoost needing only 15.41 seconds for completion. The combination of Logistic Regression and GNB produces efficient results with good accuracy ratings along with minimal training duration requirements. The high accuracy delivered by SVM comes with lengthy training sessions of 510.02 seconds which may present challenges for time critical purposes. Among deep learning models the CNN model outperforms MobileNet in terms of accuracy though it demands less training duration. The Swin Transformer presents a high training duration (400.94 seconds) as well as good accuracy levels (93.69%) because transformer-based architectural designs are computationally intensive. Alongside their high accuracy and efficient performance Random Forest and XGBoost ensemble models present an optimal blend which other models achieve when resources become available.

Table 5. Model performance after hyperparameter strategies

No	Model	Training Accuracy	Testing Accuracy	Training Time (s)
1	GNB	0.9148	0.9150	0.36
2	SVM	0.9968	0.9961	510.02
3	Logistic Regression	0.9469	0.9463	0.33
4	Decision Tree	0.9968	0.9957	0.51
5	Random Forest	0.9968	0.9961	18.03
6	Gradient Boosting	0.9968	0.9968	96.22

7	XGBoost	0.9968	0.9968	15.41
8	CNN	0.9847	0.9838	13.66
9	MobileNet	0.9699	0.9762	98.71
10	Swin Transfomer	0.9369	0.9421	400.94

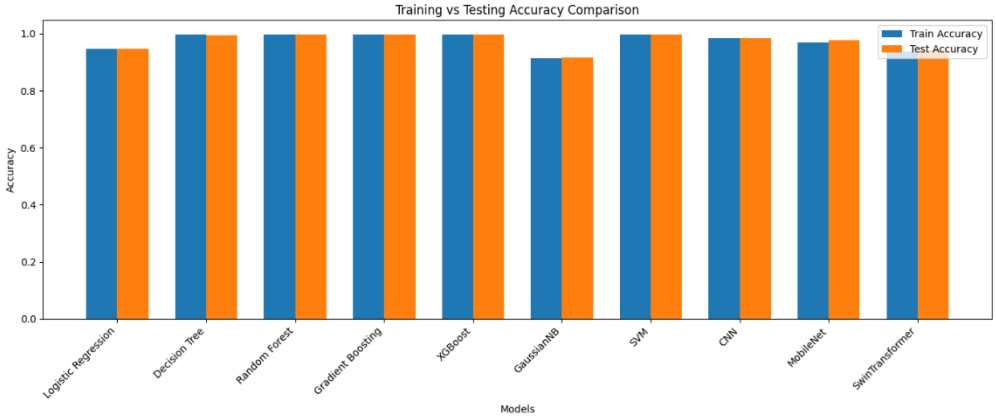

Fig.5. Model Comparison of Training and Testing accuracy

The Fig 7 compares training times of various models on a log scale. Fast training occurs within under 1 second for the Logistic Regression and Naïve Bayes models but the complex models such as CNN, XGBoost and transformers demand considerably longer training times. Among all models SVM possesses the longest training duration of 510 seconds which represents an established relationship between model complexity and training time.

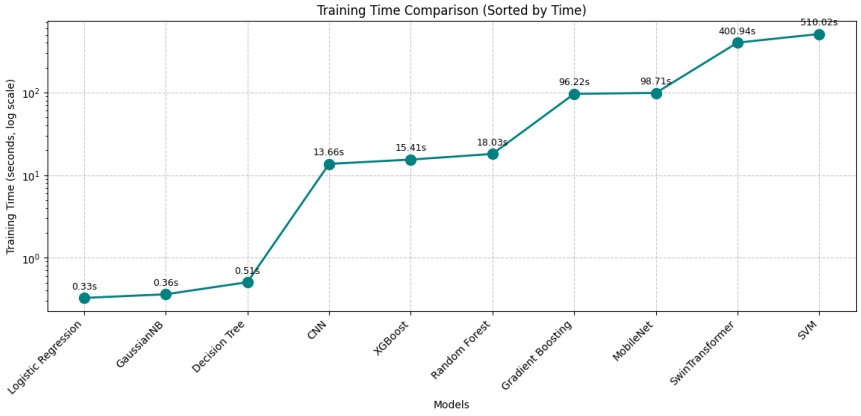

Fig. 6. Training time of ML models

A performance comparison of MobileNet, Swin Transformer and CNN models exists across ten training epochs through graphical display. The training models demonstrate superior accuracy performance and decreased loss values during the learning process. The evaluation shows CNN delivers superior results because it obtains high accuracy and minimal overfitting alongside the lowest loss. MobileNet exhibits good performance through a steady improvement process while maintaining strong generalization capabilities. During training Swin Transformer exhibits steady convergence yet it reaches accuracy levels which are slightly lower than the other approaches. Among these three models CNN demonstrates both the highest efficiency and accuracy levels. Fig 8, Fig 9 and Fig 10 show that Training and Testing results of DL models.

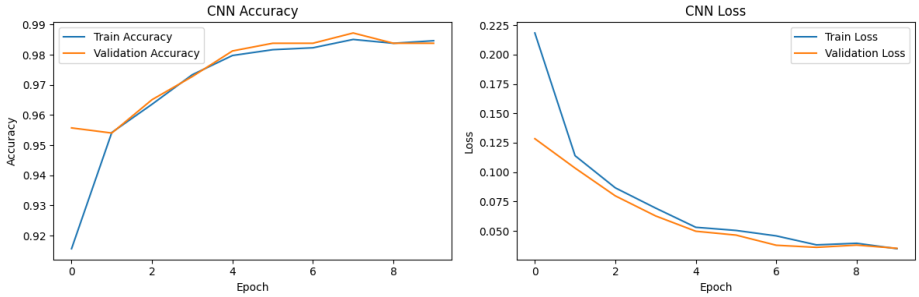

Fig. 7. CNN model Accuracy and Loss graph

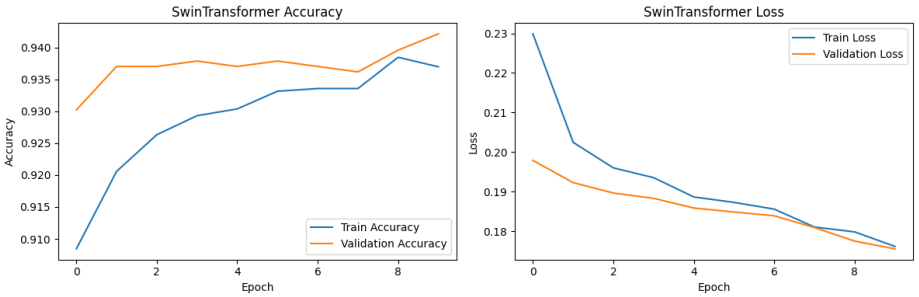

Fig. 8. Swin Transformer model Accuracy and Loss Graph

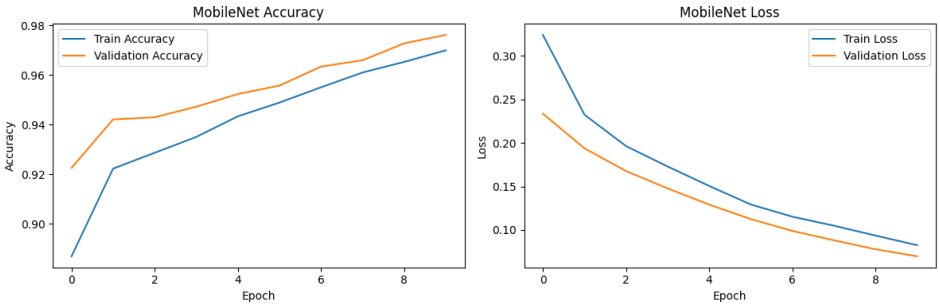

Fig.9. MobileNet model Accuracy and Loss graph

The traditional ensemble models Random Forest along with Gradient Boosting and XGBoost and SVM and Decision Tree generated top performance through their perfect sensitivity (1.000) and near-perfect F1-scores (0.9981) measurements. CNN established its superiority in the performance metrics by attaining an F1-score of 0.9927 while surpassing MobileNet (0.9884) and demonstrating much better performance than Swin Transformer (0.9652) and Logistic Regression (0.9661). The high sensitivity rate (0.9651) from Gaussian Naïve Bayes produced limited outcomes because it was matched with weak specificity (0.5486). The current analysis demonstrates better classification performance achieved by ensemble learning models together with CNN deep learning architecture when compared to simpler or less specialized methodologies. Table 4 shows that results of Evaluation matric of ML and DL models. Table 6 shows that Evaluation matirc of ML and DL models without using Bayesian optimization.

Table 6. Evaluation matric for ML and DL models (without using Bayesian Optimization)

No	Model	Specificity	Sensitivity	Precision	Recall	F1-Score
1	GNB	0.5486	0.9651	0.9387	0.9651	0.9517
2	SVM	0.9722	1.000	0.9961	1.000	0.9981
3	Logistic Regression	0.7569	0.9661	0.9661	0.9661	0.9661
4	Decision Tree	0.9722	1.000	0.9961	1.000	0.9981
5	Random Forest	0.9722	1.000	0.9961	1.000	0.9981
6	Gradient Boosting	0.9722	1.000	0.9961	1.000	0.9981
7	XGBoost	0.9722	1.000	0.9961	1.000	0.9981
8	CNN	0.9514	0.9922	0.9932	0.9922	0.9927
9	Mobile Net	0.8750	0.9942	0.9827	0.9942	0.9884
10	Swin Transformer	0.7361	0.9670	0.9633	0.9670	0.9652

Fig 11 evaluates the performance of machine learning models which include Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVM, CNN, MobileNet and Swin Transformer under Bayesian optimization testing across Specificity, Sensitivity, Precision, Recall, and F1-Score. The models show success rates between 0.92 and 1.0 which gather mostly in the 0.98-1.0 range to demonstrate superior performance metrics.

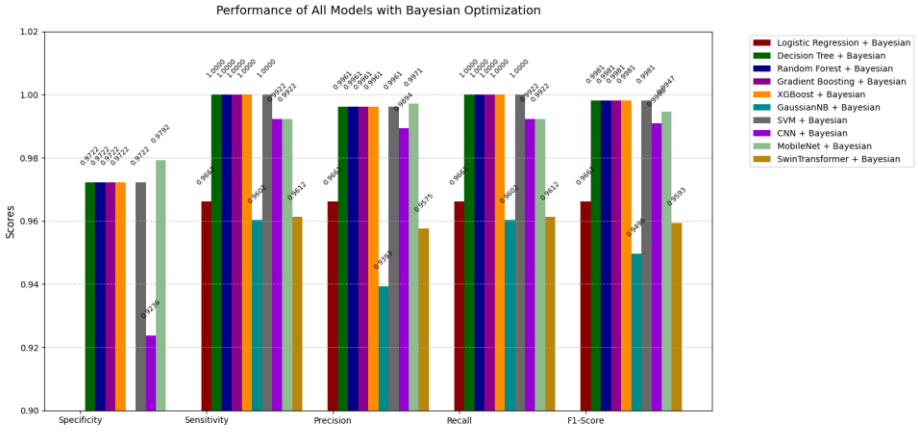

Fig. 10. Comparison of ML and DL evaluation matric with Bayesian optimization

As shown Table 7, the confusion matrix information supports previous metrics as it shows the exact accuracy of model classifications. All models including SVM and Decision Tree alongside Random Forest and Gradient Boosting and XGBoost reached the maximum TP score (1031) with zero FN while producing only 4 FP instances thus demonstrating their top-level predictive capabilities. CNN showed similar strong performance by recording 8 false negatives and 7 false positives together with MobileNet. Gaussian Naïve Bayes experienced significant limitations in its ability to distinguish between categories since it misidentified 65 negative examples (FP) and 36 positive examples (FN) thus demonstrating very weak specificity measures. Swin Transformer along with Logistic Regression achieved average performance by producing elevated numbers of false positives and negatives compared to standard models. The ensemble methods together with CNN successfully minimize classification errors above simpler and transformer-based methods.

Table 7. Confusion matric of machine learning models

No	Model	TN	FP	FN	TP
1	GNB	79	65	36	995
2	SVM	140	4	0	1031
3	Logistic Regression	109	35	35	996
4	Decision Tree	140	4	0	1031
5	Random Forest	140	4	0	1031
6	Gradient Boosting	140	4	0	1031
7	XGBoost	140	4	0	1031
8	CNN	137	7	8	1023
9	Mobile Net	126	18	6	1025
10	Swin Transformer	106	38	34	997

The ROC curve proves that different ML and DL models effectively identify cancerous and non-cancerous patterns in lung cancer diagnoses. The AUC reached perfection at

1.00 for classification results produced by SVM alongside Decision Tree, Random Forest, Gradient Boosting, XGBoost, CNN and MobileNet ensuring flawless detection of lung cancer cases along with no false positives with healthy patients. Both Logistic Regression and Swin Transformer delivered accurate predictions yet their AUC reached 0.96 and 0.95 respectively while showing slightly less accuracy. Gaussian Naïve Bayes demonstrated the lowest performance in terms of AUC value reaching 0.94 while providing less reliable results. The ROC curve demonstrates ensemble models together with deep learning systems possess remarkable accuracy in lung cancer prediction which makes them appropriate for clinical use in early diagnosis procedures. Fig 11 shows that ML models result of ROC Curve.

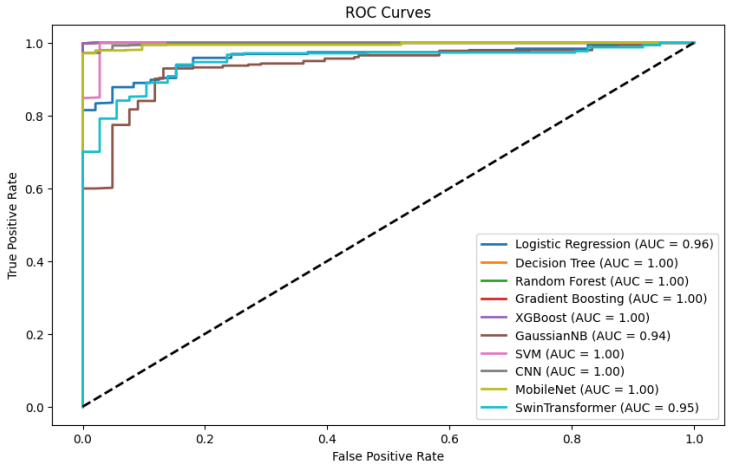

Fig. 11. ROC curve for machine learning models

4 Conclusion and Future Work

In the research multiple ML models were evaluated for classification work with traditional (Logistic Regression, Decision Tree, Random Forest, XGBoost, Gradient Boosting, SVM, GNB) and deep learning algorithms (CNN, Mobile Net, Swin Transformer). The analysis included assessment of accuracy, specificity, sensitivity, precision, recall, F1-score and AUC from ROC curves as well as training and testing times and cross-validation techniques (K-fold, Stratified K-fold, Leave-one-out). Ensemble approaches comprising Random Forest, Gradient Boosting, and XGBoost prove to be the most effective classifiers according to the results since these algorithms consistently achieve 0.9961–0.9968 training and testing accuracies as well as 1.00 AUC scores and equal metric values between 0.9722–1.000 for precision, sensitivity, F1-score, recall, and specificity. The Swin Transformer model achieves testing accuracy of 0.9421 but requires long training at 400.94 seconds alongside substantial computational expense whereas Mobile Net demonstrates better performance at 0.9761 testing accuracy in

98.71 seconds training time although both methods show slight overfitting through accuracy and loss curve comparisons. The CNN model achieves poor generalization when performing Leave-one-out cross-validation because its accuracy falls to 0.4000. AUC results from ROC curves demonstrate that ensemble strategies together with deep learning models except GNB (AUC = 0.94) perform well in class differentiation.

Most models demonstrate small differences between training accuracy and testing accuracy while ensemble methods specifically maintain stable accuracy between these measures. Swin Transformer demonstrates a minor difference between its training accuracy of 0.9369 and testing accuracy of 0.9421 suggesting overfitting potential which is confirmed through observation of slower validation loss reduction compared to training loss. Logistic Regression and GNB maintain efficient computation times (0.32s and 0.36s respectively) but produce lower performance accuracies (0.9463 and 0.9150) along with specificities of 0.7569 and 0.5486. The ensemble approaches strike an optimal combination between model performance and generalization ability yet deep learning models need precise optimization to avoid overfitting and control their cost requirements.

Future work should exert efforts to enhance the efficiency of deep learning platforms including Swin Transformer and Mobile Net through implementation of techniques such as weight decay and dropout regulation alongside data augmentation or model simplification methods. The generalization capabilities of CNN models during Leave-one-out cross-validation need improvement which might be achieved through testing with expanded datasets and transfer learning approaches. A thorough assessment involving performance-testing on resource-limited platforms (such as edge devices) would determine how to maximize real-world usage of these models.

References

1. Heuvelmans, M.A., van Ooijen, P.M., Ather, S., Silva, C.F., Han, D., Heussel, C.P., Hickes, W., Kauczor, H.-U., Novotny, P., and Peschl, H.: 'Lung cancer prediction by Deep Learning to identify benign lung nodules', Lung cancer, 2021, 154, pp. 1-4
2. Kadir, T., and Gleeson, F.: 'Lung cancer prediction using machine learning and advanced imaging techniques', Translational lung cancer research, 2018, 7, (3), pp. 304
3. Abdullah, D.M., Abdulazeez, A.M., and Sallow, A.B.: 'Lung cancer prediction and classification based on correlation selection method using machine learning techniques', Qubahan Academic Journal, 2021, 1, (2), pp. 141-149
4. Tuncal, K., Sekeroglu, B., and Ozkan, C.: 'Lung cancer incidence prediction using machine learning algorithms', Journal of advances in information technology, 2020, 11, (2)
5. Sweet, M.M.R., Ahmed, M.P., Mozumder, M.A.S., Arif, M., Chowdhury, M.S., Bhuiyan, R.J., Rahman, T., Ahmmed, M.J., Ahmed, E., and Mamun, M.A.I.: 'COMPARATIVE ANALYSIS OF MACHINE LEARNING TECHNIQUES FOR ACCURATE LUNG CANCER PREDICTION', The American Journal of Engineering and Technology, 2024, 6, (09), pp. 92-103

6. Li, L., Yang, J., Por, L.Y., Khan, M.S., Hamdaoui, R., Hussain, L., Iqbal, Z., Rotaru, I.M., Dobrotă, D., and Aldrdery, M.: 'Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques', Heliyon, 2024, 10, (4)
7. Kanakaraddi, S.G., Handur, V.S., Jalannavar, A., Chikaraddi, A., and Giraddi, S.: 'Segmentation and Classification of Lung Cancer using Deep Learning Techniques', Procedia Computer Science, 2024, 235, pp. 3226-3235
8. Wu, W.-T., Li, Y.-J., Feng, A.-Z., Li, L., Huang, T., Xu, A.-D., and Lyu, J.: 'Data mining in clinical big data: the frequently used databases, steps, and methodological models', Military Medical Research, 2021, 8, pp. 1-12
9. Han, J., Kamber, M., and Mining, D.: 'Concepts and techniques', Morgan kaufmann, 2006, 340, (1), pp. 94104-103205
10. Berrar, D.: 'Cross-validation', in Editor (Ed.)^(Eds.): 'Book Cross-validation' (2019, edn.), pp.
11. Chen, Y., Feng, J., Liu, J., Pang, B., Cao, D., and Li, C.: 'Detection and classification of lung cancer cells using swin transformer', Journal of Cancer Therapy, 2022, 13, (7), pp. 464-475
12. Qiu, J.: 'An analysis of model evaluation with cross-validation: techniques, applications, and recent advances', Advances in Economics, Management and Political Sciences, 2024, 99, pp. 69-72
13. Sachdeva, R.K., Uppal, P., Bathla, P., and Solanki, V.: 'Employing Machine Learning for Effective Lung Cancer Diagnosis', in Editor (Ed.)^(Eds.): 'Book Employing Machine Learning for Effective Lung Cancer Diagnosis' (IEEE, 2024, edn.), pp. 1-6
14. Wani, N.A., Kumar, R., and Bedi, J.: 'DeepXplainer: An interpretable deep learning based approach for lung cancer detection using explainable artificial intelligence', Computer Methods and Programs in Biomedicine, 2024, 243, pp. 107879
15. Falana, A.O., Osinuga, A., Dabira Ogunbiyi, A.I., Odezuligbo, I.E., and Oluwagbotemi, E.: 'Hyperparameter Tuning in Machine Learning: A Comprehensive Review'
16. Miller, C., Portlock, T., Nyaga, D.M., and O'Sullivan, J.M.: 'A review of model evaluation metrics for machine learning in genetics and genomics', Frontiers in Bioinformatics, 2024, 4, pp. 1457619
17. Parker, W.S.: 'Model evaluation': 'The Routledge Handbook of Philosophy of Scientific Modeling' (Routledge, 2024), pp. 208-219
18. Zamzam, Y.F., Saragih, T.H., Herteno, R., Nugrahadi, D.T., and Huynh, P.-H.: 'Comparison of CatBoost and Random Forest Methods for Lung Cancer Classification using Hyperparameter Tuning Bayesian Optimization-based', Journal of Electronics, Electromedical Engineering, and Medical Informatics, 2024, 6, (2), pp. 125-136
19. Aurélien, G.: 'Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow' (o'reilly, 2019. 2019)
20. Rybczak, M., and Kozakiewicz, K.: 'Deep machine learning of MobileNet, Efficient, and Inception models', Algorithms, 2024, 17, (3), pp. 96
21. Sun, R., Pang, Y., and Li, W.: 'Efficient lung cancer image classification and segmentation algorithm based on an improved swin transformer', Electronics, 2023, 12, (4), pp. 1024