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

Luxshi Karunakaran[1], Chandrika Malkanthi[1], Senthan Prasanth[2], and R. M. K. T. Rathnayaka[1]

[1]Department of Physical Sciences and Technology, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka
klluxshi99@gmail.com, {chandrika, kapilar}@appsc.sab.ac.lk
[2]Faculty of Engineering and Applied Science, Memorial University of Newfoundland and Labrador, Canada
senthanprasanth007@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 utilizing hyperparameter tuning together with cross validation approaches. The XGBoost model achieved the highest accuracy of 0.9968 during cross-validation tests using the stratified k-fold (k=5) and leave-one-out methods. XGBoost and Gradient Boosting demonstrated optimal performance after hyperparameter tuning, as they achieved an accuracy of 0.9968 for both training and testing sets, although the total training time was relatively different. CNN demonstrated powerful performance throughout its training and testing stages, achieving the fastest training time among deep learning models with accuracy values of 0.9829 and 0.9872. Ensemble ML methods and optimized DL models were highly effective in lung cancer prediction. Future work will investigate the application of large-scale data platforms to improve the predictive performance of DL models.

Keywords: Cross validation · Deep learning · Hyperparameter tuning · Lung cancer · Machine learning.

1 Introduction

Medical science categorizes 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 Computed Tomography (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 hyperparameter 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, specificity, and ROC-AUC metrics, to determine the most suitable advanced 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 a generic model with model specific characteristics. An insufficient evaluation process may result in wasting diagnostic potential. 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.

2 Related work

The related work discusses recent developments in ML and DL for predicting lung cancer and the application of such approaches using numerical data. It also examines how the hyperparameter optimization strategy and cross-validation methods are utilized to ensure the models' robustness and reliability.

ML and DL methods have gained rapid development in lung cancer prediction over the last decade [3]. SVM alongside random forest, can surpass traditional

learning techniques in lung cancer prediction, when processed with advanced sets of features[4]. Ensemble models have generated accurate and stable results compared to traditional models in the literature. Ensemble models such as GNB, SVM, Logistic Regression, Random Forest, Gradient Boosting and XGBoost were accurate in classifying cancer patients using features such as age, smoking history and symptoms [5, 6]. DL models such as Convolutional Neural Networks (CNNs), MobileNet and Swin Transformer performed well in extracting complex patterns, whereby CNNs only capture hierarchical features, MobileNet is capable of capturing low-resource-efficient tasks, and Swin Transformer models long-range dependencies using an attention mechanism [1]. The transfer learning also enhances the performance, particularly where there is insufficient training data available [7]. Hyperparameter tuning techniques, including Bayesian optimization, have been used to optimize these models by sensitivity adjustment of parameters that improve on model generalization and avoid overfitting (e.g., learning rates, dropout rates). Besides, cross-validation procedures such as 5-fold, stratified 5-fold and Leave-One-Out (LOO) are needed to gain information on the robustness of a model to ensure reliable and precise clinical decision support systems [8].

Classification models such as Rule-Based, Decision Tree, Naive Bayes and Artificial Neural Network (ANN) are used to detect lung cancer using a massive volume of data [1]. In their study, the model was designed using features such as age, sex, wheezing, shortness of breath, and shoulder, chest, and arm. A comparative analysis revealed that SVM (95.56% accuracy) is accurate at cancer detection, outperforming CNN and K-Nearest Neighbor (KNN) (92.11% and 88.40% accuracy) models, for early lung cancer diagnosis using the UCI dataset, including patients who received a lung cancer diagnosis [5].

Radiomics implies automatic extraction of medical image-based quantitative features, which is widely used for lesion classification applications. Different imaging techniques like CT are being used in lesion investigation, where DL models are employed for automatic extraction of medical image-based features. A study reviewed the primary methods that classify nodules and predict lung cancer by analyzing CT imaging data [5]. The results revealed that CNNs trained with sufficient data performed best, with an Area Under Curve (AUC) of 0.90; after, it is required to pay careful attention to data limitations present in the validation and training datasets during system performance assessments. The ensemble model's prediction capability was compared with ResNet-50, VGG-16, and EfficientNet-B5 DL models with automated feature extraction of histopathological images using a U-Net model [6]. The results revealed that the ensemble model performed best with an accuracy of 0.99, followed by the EfficientNet-B5 with an accuracy of 0.97.

Lung cancer incidence rates of males and females across ten European countries were evaluated using support vector regression (SVR), backpropagation and Long-Short Term Memory networks (LSTM) before lung cancer prediction [7]. Effective assessment metrics, including mean square error (MSE), coefficient of determination (R2) and explained variance (EV) scores, were used for the results

evaluation, where SVR recorded the best performance and LSTM recorded the lowest performance.

The prospect of incorporating various features and refined hyperparameters to achieve diagnostic precision in non-small cell lung cancer (NSCLC) precisely and small cell lung cancer (SCLC) is studied in literature [8]. Hybrid feature extraction of grey-level co-occurrence matrix (GLCM), Haralick and autoencoder features, and optimized machine learning models were used to develop accurate lung cancer detection models. The study results showed that SVM, radial basis functions (RBF) and SVM gaussian models with hybrid features and SVM polynomial with single Haralick features improved the accuracy of the models.

Boosting models like XGBoost and LightGBM are viable predictive models exhibiting superior performance when compared with AdaBoost, Logistic Regression and SVM [9]. The analysis revealed that XGBoost consistently outperformed the other models in terms of accuracy, sensitivity, specificity and F1 score, achieving 97.50% , 96.80% , 98% , and 97.50% . LightGBM also showed strong results, remaining as a potential alternative.

However, there are still limitations and challenges in the previous research to predict lung cancer. Homogeneous or small datasets are hardly representative of the diverse populations of patients that make the models less generalizable [1]. The excessive focus on accuracy as an essential indicator might ignore other vital clinical indicators such as sensitivity and specificity in the predisposition to unreliable forecasts. Transformers exhibit computational complexity, requiring large resources, an aspect that increases their limitation to low-resource environments [7, 10]. Features such as poor cross-validation strategies and inability to interpret the model weaken the trust placed in a given model by physicians, particularly due to the risk of overfitting [9]. Real-world applications are further complicated by poor integration into clinical workflows and adjustable data preprocessing, including processing of missing values or normalization [11].

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?

2.1 Significance of the study

In this study, ML models like logistic regression with decision tree along with random forest, gradient boosting, XGBoost, SVM, and GNB, as well as advanced DL models like CNN, MobileNet, 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 the hold-out

method, k-fold cross-validation, stratified k-fold and the LOO method. Diagnostic accuracy was improved from model comparison through measurements, including evaluation metrics like accuracy, sensitivity, specificity, confusion matrix and Area under Curve Receiver Operating Characteristics (AUC-ROC). There is no clear comparison to performance optimization in lung cancer prediction in existing studies, especially in addressing certain issues such as showing weaknesses in the utilization of small or confined datasets together with its restricted usage of DL 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, indicating their usefulness for healthcare implementations in real-world practice.

The study will be unique among other ML and DL studies on lung cancer prediction due to its ability to introduce an extensive system combining bayesian optimization with hyperparameter tuning of ML and DL models for numerical data. In contrast, most of the existing studies main focus is lung cancer pattern identification using images with straightforward validation schemes. This paper will discuss the effect of different cross-validation methods (hold-out, k-fold, stratified k-fold, LOO) on numerical data prediction with a strong emphasis on increasing both accuracy and efficiency of the model. The model's ability to accurately diagnose the cancer using numeric data aid to increases the accessibility of health diagnostics in the underprivileged regions, ensuring healthcare equity. Moreover, the paper addresses another limitation in the literature by providing a comprehensive performance assessment system with a positive trade-off between performance and computational demands, focusing accuracy, precision, recall, F1-score, AUC-ROC, confusion matrix, and training time. This has been substantiated by the fact that the prediction has significantly improved where its hyperparameters have been bayesian tuned; hence, it is now a scalable and efficient model which is ready to be integrated into clinical practice, therefore representing a reference point to predict lung cancer through numerical data.

3 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. Fig 1 shows the high-level architecture of ML techniques.

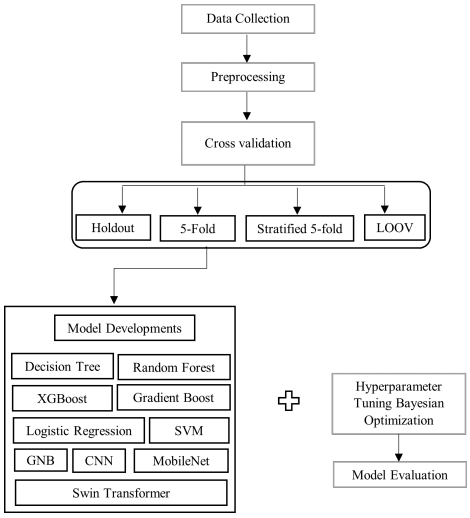

Fig. 1. High level architecture

3.1 Data gathering and data preprocessing

A set of numerical patient data (Table 1) contains demographic statistics in combination with smoking data and respiratory symptoms regarding cough and shortness of breath was acquired. The data collection contains 16 attribute features covering 5872 records [12]. As a part of data preparation, preprocessing deals with processing raw data through diverse methods to get it ready for subsequent data processing tasks [13]. 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 [13, 14]. 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.

The dataset was chosen for its focus on numerical attributes, enabling cost-effective lung cancer prediction without relying on expensive imaging techniques. In addition to mean imputation of missing values, the two important preprocessing methods are encoding categorical variables and feature scaling. Encoding transforms the categories, such as gender (M/F) and lung cancer (YES/NO) into numbers (label encoding) so that they can be compatible with algorithms.

Table 1. Feature descriptions of the lung cancer dataset

Feature	Data description	Type
Gender	Patient's gender.	Categorical
Age	Patient's age in years.	Numerical
Smoking	Whether the patient smokes.	Categorical
Yellow Fingers	Presence of yellow fingers.	Categorical
Anxiety	Whether the patient experiences anxiety.	Categorical
Peer Pressure	Influence from peers affecting lifestyle.	Categorical
Chronic Disease	Presence of chronic disease(s).	Categorical
Fatigue	Whether the patient experiences fatigue.	Categorical
Allergy	Whether the patient has allergies.	Categorical
Wheezing	Wheezing sound during breathing.	Categorical
Alcohol	Whether the patient consumes alcohol.	Categorical
Coughing	Presence of persistent cough.	Categorical
Shortness of Breath	Breathing difficulties.	Categorical
Swallowing Difficulty	Difficulty swallowing.	Categorical
Chest Pain	Whether the patient reports chest pain.	Categorical
Lung Cancer	Diagnosis outcome for lung cancer (YES/NO).	Numerical

Numerical features such as age are normalized by feature scaling to balance model training. These procedures enhance the quality of data.

3.2 Data splitting methods

Table 2. Comparison of model validation techniques

Technology	Operation steps	Advantages	Ref
Holdout method	Data is split into training and testing sets using a fixed ratio (e.g., 80:20).	Simple and works well for large datasets.	[2]
K-Fold cross validation	Dataset is divided into K equal parts. Each part is used as validation once.	Uses all data for training and validation.	[15]
Stratified K-Fold cross validation	Similar to K-Fold, but preserves class proportions in each fold.	Better for imbalanced data; retains class distribution.	[4, 10]
Leave-One-Out cross validation	Uses one sample for validation and the rest for training. Repeated for each sample.	Utilizes all samples for both training and validation.	[16]

ML 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 [15]. The data splits into 'k' equivalent sections using K-Fold cross validation so each sub-set operates as validation data alongside training data that consists of remaining sections thus achieving more dependable results [2, 4]. The stratification of K-Fold cross-validation performs dataset stratification to maintain proportional class distribution between each fold, which benefits datasets with imbalanced classes [7, 15]. Inside LOO cross-validation, 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 2 shows the operations of each validation method [17, 18].

3.3 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 [16]. 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 [16]. The optimization mechanism in bayesian theory bases its foundation on bayes' Theorem as presented by Eq (1) [19].

$$P(A \mid B) = \frac{P(B \mid A) \cdot P(A)}{P(B)} \tag{1}$$

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).

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

3.4 Model development

This study considers binary classification of lung cancer data using ML and DL models including logistic regression, decision tree, random forest, gradient boosting, XGBoost, GNB, SVM, CNN, MobileNet and swin transformer 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) [20]. Decision tree creates a tree structure through repeated feature space splitting that

Table 3. Hyperparameter description of the ML/DL models

Model	**Hyperparameters tuned**
Logistic Regression	`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` (range: 3 to 10)
Gaussian Naive Bayes (GNB)	`var_smoothing` (Smoothing 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` (range: 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)

optimizes both maximum depth and minimum splitting data points. Random forest uses multiple decision trees to gather predictions and minimizes overfitting through number of trees and maximum depth optimization [8]. 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 [16, 21]. 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" [22]. 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 5-fold stratified, 5-fold and LOO cross-validation after applying bayesian optimization to find the best hyperparameters [10]. Table 3 shows that hyperparameter used in the tuning process.

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 a 0.5 rate before a sigmoid activation dense layer performs binary output [23]. Using bayesian optimization maintains the model structure intact yet allows the adjustable hyperparameters filters1 (16–64), filters2 (3(32–128), ense-units (64–256) and dropout-rate (0.3–0.7) to optimize validation accuracy. The model contains two iterations with the Adam optimiser (0.001 learning rate), implementing binary cross-entropy as a loss function until reaching 10 training epochs for accuracy evaluation [9, 6, 14].

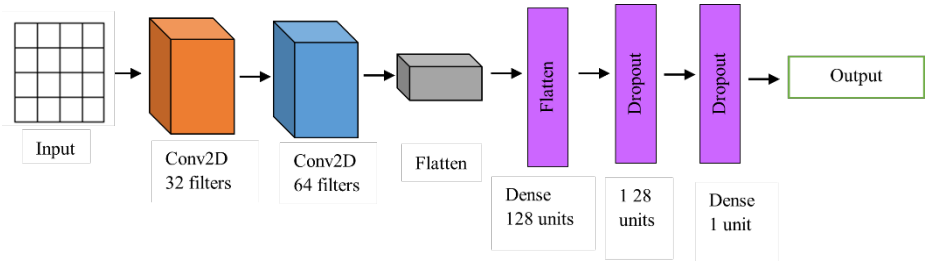

Fig. 2. CNN architecture

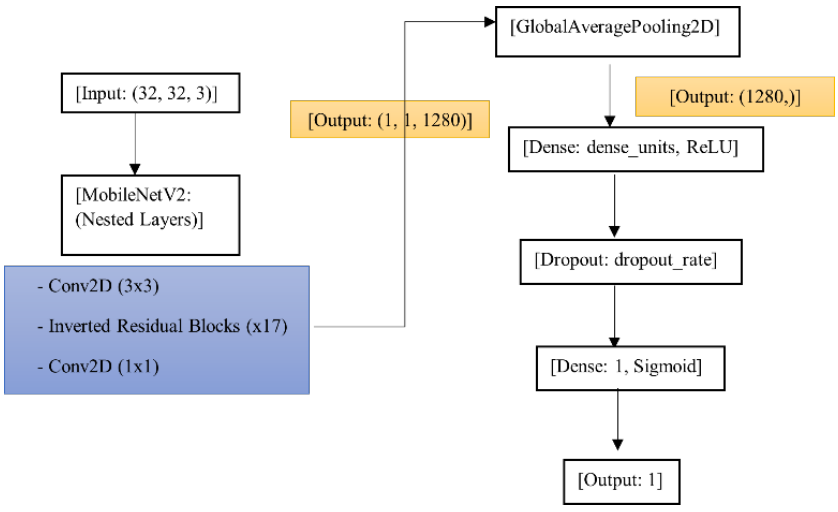

Fig. 3. MobileNet architecture

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 MobileNet 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 [10, 17]. 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 the Adam optimizer (learning rate set at 0.001) and binary cross-entropy loss to reach an evaluation based on accuracy.

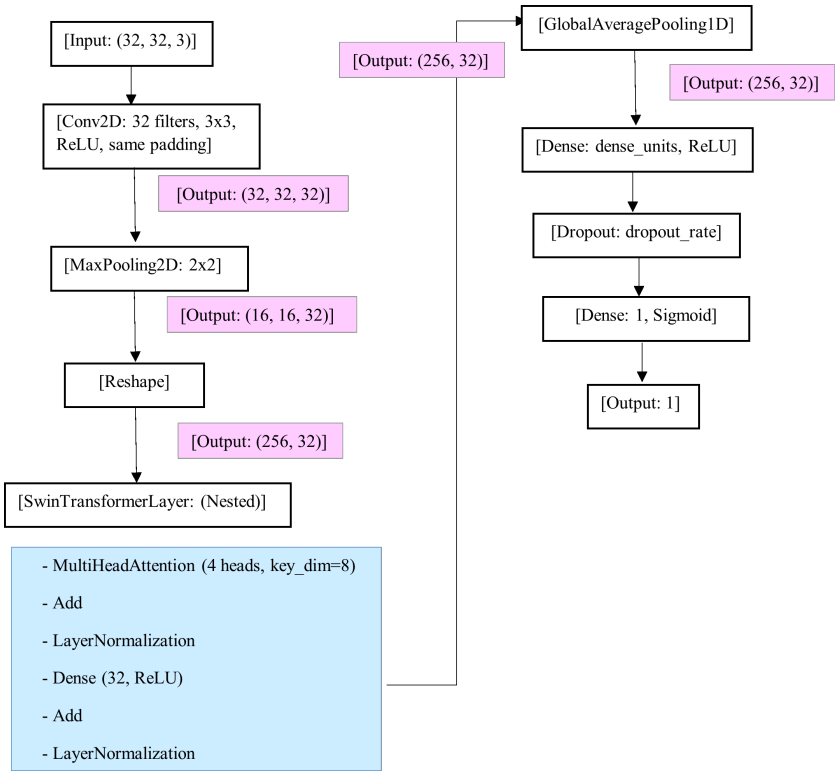

Fig. 4. Swin Transformer architecture

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 dimen-

sions and 4 heads and then executes global average pooling, a 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 cross-entropy loss during 10 epochs of training before they evaluate models based on accuracy [24].

3.5 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, and GNB, along with CNN, MobileNet 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.

Both hyperparameter optimization and evaluation of the final model were done using stratified 5-fold cross-validation with k=5. The dataset was split into stratified into five folds (YES/NO lung cancer) while mitigating the imbalance issue. Each fold was used as a validation set once, and the remaining four as training to tune the hyperparameters using Bayesian optimization to maximize the accuracy. Stratified 5-fold cross-validation with optimized hyperparameters was used to evaluate the final models .

The AUC-ROC represents a performance assessment method for binary classification models which determines their capacity to differentiate between positive and negative outcomes [7]. 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
- 0.8AUC-ROC<0.9: Good Model
- 0.7AUC-ROC<0.8: Fair Model
- AUC-ROC<0.7: Poor model

4 Results and Discussion

The results from Tables 4 and 5 demonstrate that bayesian optimization enhances model performance since its implementation yields superior accuracy measurements. The accuracy of GNB, SVM, gradient boosting, XGBoost, CNN, and MobileNet models increases substantially through the application of K-fold, stratified k-fold, and LOO 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 MobileNet 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 modelling scenarios.

Table 4. Results of cross validations without using Bayesian Optimization

No	Model	5-fold	Stratified 5-fold	LOOCV
1	Gaussian Naive Bayes (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	MobileNet	0.9778	0.9778	0.9778
10	Swin Transformer	0.9418	0.9418	0.9418

Table 5. Results of cross validation using Bayesian Optimization

No	Model	5-fold	Stratified 5-fold	LOOCV
1	Gaussian Naive Bayes (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	MobileNet	0.9902	0.9902	0.9902
10	Swin Transformer	0.9369	0.9369	0.9369

The Table 6 and Fig 5 shows an assessment of different models that focuses on training accuracy along with testing accuracy and training duration. The

Table 6. Model training and testing accuracy with training Time

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 Transformer	0.9369	0.9421	400.94

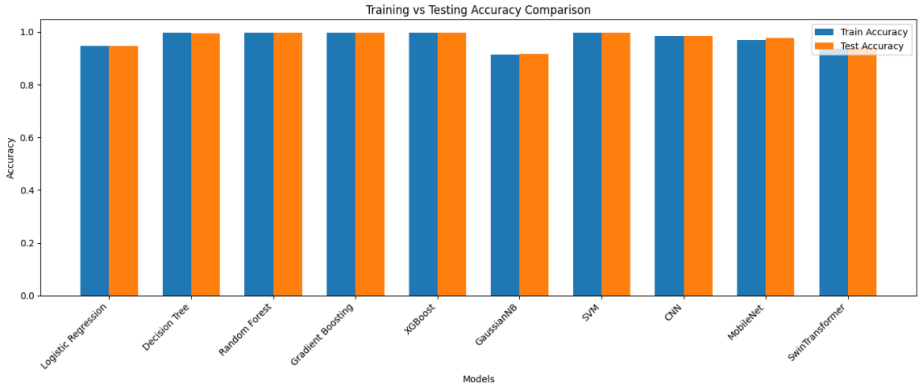

Fig. 5. Model comparison of training and testing accuracy

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 DL models the CNN outperforms MobileNet in terms of accuracy though it demands a 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.

The Fig 6 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 de-

mand 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.

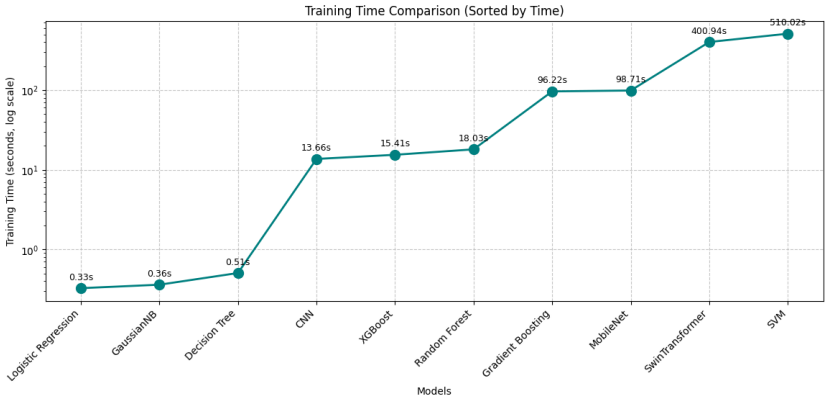

Fig. 6. Training time of ML/DL models

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 7, Fig 8 and Fig 9 show that training and testing results of DL models.

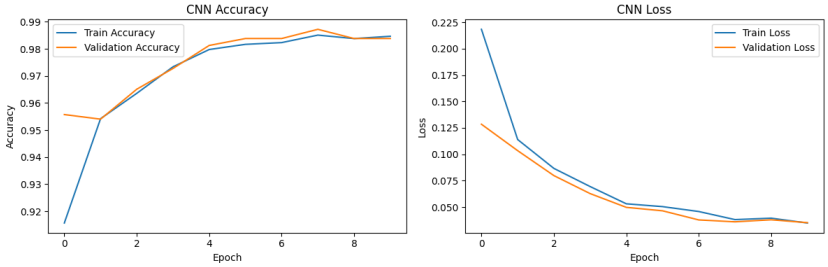

Fig. 7. CNN model accuracy and loss graph

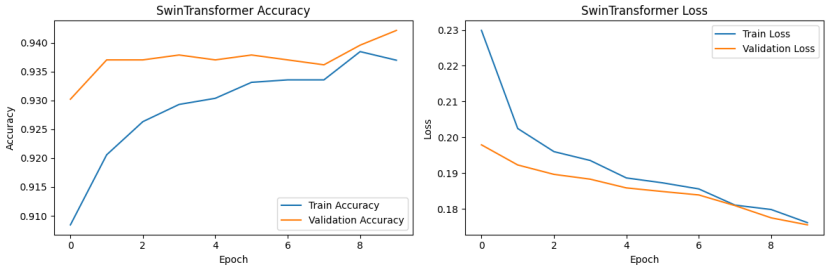

Fig. 8. Swin Transformer model accuracy and loss graph

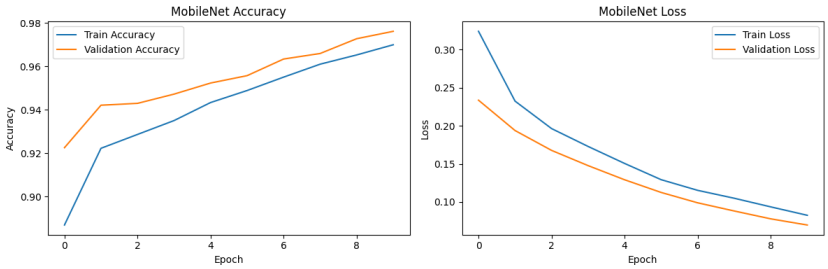

Fig. 9. MobileNet model accuracy and loss graph

The traditional ensemble models random forest along with gradient boosting, XGBoost, SVM and decision tree, generated top performance through their perfect sensitivity (1.000) and near-perfect F1 score (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 GNB 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 DL architecture when compared to simpler or less specialized methodologies. Table 7 shows that results of the evaluation metric of ML and DL models.

Fig 10 evaluates the performance of ML 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.

As shown Table 8, the confusion matrix information supports previous metrics, as it shows the exact accuracy of model classifications. All models, including

Table 7. Evaluation metric for ML and DL models (without using Bayesian Optimization)

No	Model	Specificity	Sensitivity	Precision	Recall	F1
1	Gaussian Naive Bayes (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	MobileNet	0.8750	0.9942	0.9827	0.9942	0.9884
10	Swin Transformer	0.7361	0.9670	0.9633	0.9670	0.9652

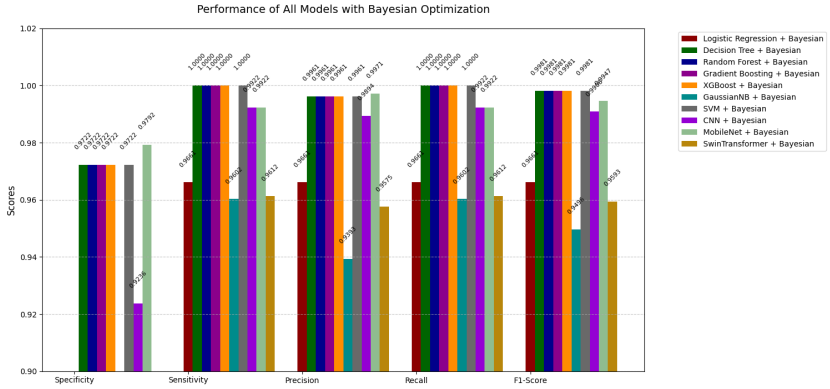

Fig. 10. Comparison of ML and DL evaluation metrics with Bayesian Optimization

SVM and decision tree alongside random forest, gradient boosting and XGBoost, reached the maximum true positive (TP) score (1031) with zero false negative (FN) while producing only 4 false positive (FP) instances, thus demonstrating their top-level predictive capabilities. CNN showed similar strong performance by recording 8 FN and 7 FP together with MobileNet. GNB experienced significant limitations in its ability to distinguish between categories since it misidentified 65 negative examples (FN) and 36 positive examples (FP), thus demonstrating very weak specificity measures. Swin transformer along with logistic regression, achieved average performance by producing elevated numbers of FP and FN compared to standard models. The ensemble methods together with CNN successfully minimize classification errors above simpler and transformer-based methods.

The AUC-ROC curve proves that different ML and DL models effectively identify cancerous and non-cancerous patterns in lung cancer diagnoses. The AUC-ROC reached perfection at 1.00 for classification results produced by SVM

Table 8. Confusion matric of ML models

No	Model	TN	FP	FN	TP
1	Gaussian Naive Bayes (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	MobileNet	126	18	6	1025
10	Swin Transformer	106	38	34	997

alongside decision tree, random forest, gradient boosting, XGBoost, CNN and MobileNet, ensuring flawless detection of lung cancer cases along with no false positives positive with healthy patients. Both logistic regression and swin transformer delivered accurate predictions yet their AUC-ROC reached 0.96 and 0.95, respectively, while showing slightly less accuracy. GNB demonstrated the lowest performance in terms of AUC-ROC value, reaching 0.94 while providing less reliable results. The AUC-ROC curve demonstrates ensemble models together with DL systems possess remarkable accuracy in lung cancer prediction which makes them appropriate for clinical use in early diagnosis procedures. Fig 11 shows the ML model's result of the AUC-ROC curve.

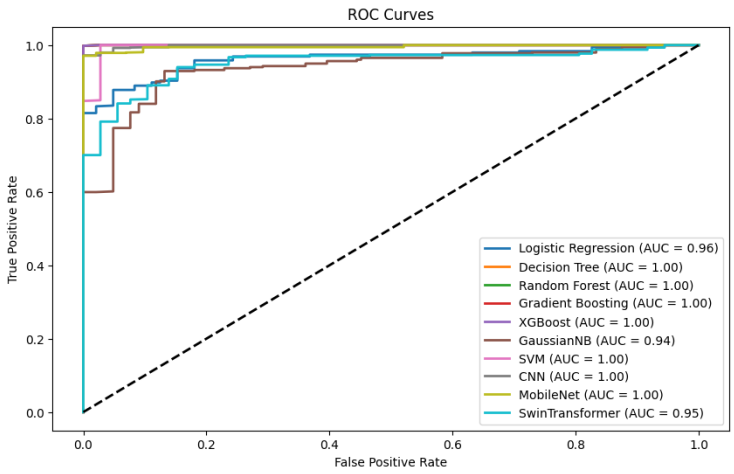

Fig. 11. AUC-ROC curve for ML models

5 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 DL algorithms (CNN, MobileNet, 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, LOO). 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 and 1.000 for precision, sensitivity, F1 score, recall, and specificity. The swin transformer model achieves a 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 MobileNet model demonstrates robust generalization across cross-validation, maintaining the result 0.9902, which highlights its effectiveness with the optimized numerical dataset. AUC results from ROC curves demonstrate that ensemble strategies together with DL models except GNB (AUC = 0.94) perform well in class differentiation.

Hyperparameter tuning enhances lung cancer prediction by optimizing model configurations to improve accuracy and generalization. Using techniques like bayesian optimization, models fine-tune parameters such as learning rate, tree depth, and regularization, enabling better capture of complex patterns in numerical patient data. This process balances exploration of new parameter combinations with exploitation of known effective settings, reducing overfitting and boosting predictive reliability for more robust clinical diagnoses.

The choice of k=5 in stratified 5-Fold cross-validation is due to its compromise between reliability and efficiency. It allows dividing the data into 80% training and 20% validation per fold, which is enough to train on, and it gives stable performance estimates with low variance compared to k=3 and Holdout [7] . k=5 is computationally efficient in comparison to k=10 and LOOCV [8]. In the case of models such as SVM (510.02s training time) and appropriate to the size of the dataset, as demonstrated by XGBoost rapid 15.41s training time and stable accuracy of 0.9968. Stratification preserves the proportion of classes, which is best in terms of reliability with imbalanced data thus, it is the best option in this study.

The superior performance of ensemble models like random forest, gradient boosting, and XGBoost over DL models such as CNN, MobileNet and swin transformer can be attributed to their ability to aggregate multiple weak learners, which enhances generalization and reduces overfitting by enhancing generalization. The models produced almost perfect metrics, having an accuracy of 0.9968, an F1 score of 0.9981 and an AUC-ROC of 1.00 with stratified 5-fold cross validation. These models are capable of capturing the complex patterns with the use of

different decision trees and iterative error correction than simple models such as logistic regression (accuracy of 0.9463) and GNB (accuracy of 0.9150), the CNN model (0.9838 accuracy, 0.9927 F1-score) slightly outperformed the ensembles, while MobileNet (0.9762 accuracy) and swin transformer (0.9421 accuracy) were computationally intense and overfitting.

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 DL models need precise optimization to avoid overfitting and control their cost requirements.

The varying training times of ML models for lung cancer prediction significantly impact their practical deployment. Models like XGBoost and CNN, with training times of around 15.00 sseconds are more feasible for real-time clinical applications due to their efficiency, whereas SVM's extensive training time (SVM 510.02 second) may hinder its use in ttime-ensitive settings. Balancing high accuracy with shorter training durations is crucial for integrating these models into resource-constrained healthcare environments.

Future work should exert efforts to enhance the efficiency of DL platforms including swin transformer and MobileNet, through the implementation of techniques such as weight decay and dropout regulation alongside data augmentation or model simplification methods. The generalization capabilities of CNN models during LOO 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] M.A. Heuvelmans et al. "Lung cancer prediction by Deep Learning to identify benign lung nodules". In: *Lung Cancer* 154 (2021), pp. 1–4.

[2] T. Kadir and F. Gleeson. "Lung cancer prediction using machine learning and advanced imaging techniques". In: *Translational Lung Cancer Research* 7.3 (2018), p. 304.

[3] CS Anita et al. "Lung cancer prediction model using machine learning techniques". In: *International Journal of Health Sciences* II (2022), pp. 12533–12539.

[4] S.G. Kanakaraddi, V.S. Handur, A. Jalannavar, et al. "Segmentation and classification of lung cancer using deep learning techniques". In: *Procedia Computer Science* 235 (2024), pp. 3226–3235.

[5] D.M. Abdullah, A.M. Abdulazeez, and A.B. Sallow. "Lung cancer prediction and classification based on correlation selection method using machine learning techniques". In: *Qubahan Academic Journal* 1.2 (2021), pp. 141–149.

[6] D. Berrar. "Cross-validation". In: *Cross-validation*. Editor, Eds. 2019.

[7] Y. Chen et al. "Detection and classification of lung cancer cells using Swin Transformer". In: *Journal of Cancer Therapy* 13.7 (2022), pp. 464–475.

[8] A.O. Falana, A. Osinuga, A.I.D. Ogunbiyi, et al. *Hyperparameter Tuning in Machine Learning: A Comprehensive Review*. n.d.

[9] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2006.

[10] Aurélien Géron. *Hands-on Machine Learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly, 2019.

[11] Kubra Tuncal, Boran Sekeroglu, and Cagri Ozkan. "Lung Cancer Incidence Prediction Using Machine Learning Algorithms". In: *Journal of Advances in Information Technology* 11.2 (2020), pp. 91–96. DOI: 10.12720/jait.11.2.91-96. URL: https://www.researchgate.net/publication/343855359.

[12] Kaggle. *Lung Cancer*. https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer. Accessed: June 18, 2025. n.d.

[13] *Lung Cancer*. In Book *Lung Cancer* (Editor, Eds.) n.d.

[14] W.-T. Wu, Y.-J. Li, A.-Z. Feng, et al. "Data mining in clinical big data: the frequently used databases, steps, and methodological models". In: *Military Medical Research* 8 (2021), pp. 1–12.

[15] J. Qiu. "An analysis of model evaluation with cross-validation: techniques, applications, and recent advances". In: *Advances in Economics, Management and Political Sciences* 99 (2024), pp. 69–72.

[16] C. Miller et al. "A review of model evaluation metrics for machine learning in genetics and genomics". In: *Frontiers in Bioinformatics* 4 (2024), p. 1457619.

[17] L. Li, J. Yang, L.Y. Por, et al. "Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques". In: *Heliyon* 10.4 (2024).

[18] R.K. Sachdeva et al. "Employing Machine Learning for Effective Lung Cancer Diagnosis". In: *IEEE Conference Proceedings*. 2024, pp. 1–6.

[19] K. Tuncal, B. Sekeroglu, and C. Ozkan. "Lung cancer incidence prediction using machine learning algorithms". In: *Journal of Advances in Information Technology* 11.2 (2020).

[20] N.A. Wani, R. Kumar, and J. Bedi. "DeepXplainer: An interpretable deep learning-based approach for lung cancer detection using explainable artificial intelligence". In: *Computer Methods and Programs in Biomedicine* 243 (2024), p. 107879.

[21] W.S. Parker. "Model Evaluation". In: *The Routledge Handbook of Philosophy of Scientific Modeling*. Routledge, 2024, pp. 208–219.

[22] Y.F. Zamzam, T.H. Saragih, R. Herteno, et al. "Comparison of CatBoost and Random Forest methods for lung cancer classification using hyperparameter tuning Bayesian optimization-based". In: *Journal of Electronics, Electromedical Engineering, and Medical Informatics* 6.2 (2024), pp. 125–136.

[23] M. Rybczak and K. Kozakiewicz. "Deep machine learning of MobileNet, Efficient, and Inception models". In: *Algorithms* 17.3 (2024), p. 96.

[24] R. Sun, Y. Pang, and W. Li. "Efficient lung cancer image classification and segmentation algorithm based on an improved Swin Transformer". In: *Electronics* 12.4 (2023), p. 1024.

[25] Fatimah Abdulazim Altuhaifa, Khin Than Win, and Guoxin Su. "Predicting lung cancer survival based on clinical data using machine learning: A review". In: *Computers in biology and medicine* 165 (2023), p. 107338.

[26] Liangyu Li et al. "Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques". In: *Heliyon* 10.4 (2024).

[27] Amoakoh Gyasi-Agyei. "A comparative assessment of machine learning models and algorithms for osteosarcoma cancer detection and classification". In: *Healthcare Analytics* 7 (2025), p. 100380.

[28] Mohammad Q Shatnawi, Qusai Abuein, and Romesaa Al-Quraan. "Deep learning-based approach to diagnose lung cancer using CT-scan images". In: *Intelligence-Based Medicine* 11 (2025), p. 100188.

[29] Kamta Nath Mishra et al. "Enhancing cancer detection and prevention mechanisms using advanced machine learning approaches". In: *Informatics in Medicine Unlocked* 50 (2024), p. 101579.

[30] Taisheng Zeng et al. "AI diagnostics in bone oncology for predicting bone metastasis in lung cancer patients using DenseNet-264 deep learning model and radiomics". In: *Journal of Bone Oncology* 48 (2024), p. 100640.

[31] V Sreeprada and K Vedavathi. "Lung Cancer Detection from X-Ray Images using Hybrid Deep Learning Technique". In: *Procedia Computer Science* 230 (2023), pp. 467–474.