**ENHANCING LUNG CANCER PREDICTION USING MACHINE LEARNING: ACOMPARATIVE ANALYSIS OF HYPERPARAMETER OPTIMIZATION TECHNIQUES**

**By**

**Luxshi Karunakaran**

**(18APP3508)**

Bachelor of Science (Honour) Degree

In

Computer Science and Technology

Department of Physical Sciences and Technology,

Faculty of Applied Sciences,

Sabaragamuwa University of Sri Lanka, Belihiloya.

May 2025

**ENHANCING LUNG CANCER PREDICTION USING MACHINE LEARNING: ACOMPARATIVE ANALYSIS OF HYPERPARAMETER OPTIMIZATION TECHNIQUES**

**By**

**Luxshi Karunakaran**

**(18APP3508)**

Research Project Report is Submitted in Partial Fulfilment of the requirement for the Bachelor of Science (Honour) Degree

In

Computer Science and Technology

Department of Physical Sciences and Technology,

Faculty of Applied Sciences,

Sabaragamuwa University of Sri Lanka,

Belihiloya.

May 2025



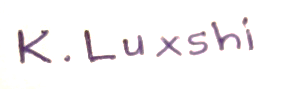
**Department of Physical Sciences and Technology Faculty of Applied Sciences**

**Sabaragamuwa University of Sri Lanka**

# DECLARATION

This thesis includes no material that was formerly submitted for educational degrees or diplomas to any university as far as I am aware and believe. The writing contains no previously published material by another person unless explicitly mentioned in the text nor my own previously written content.

I also offer Sabaragamuwa University of Sri Lanka the non-exclusive right to reproduce and distribute my thesis in whole or in part, in print, electronic, or any other form. I maintain the right to incorporate this information into future works in whole or in part (such as articles or books).



K.Luxshi

Name of the Student Signature of the Student

Date: 04/05/2025

# CERTIFICATE OF APPROVAL

We affirm that this thesis results from the student’s labour and effort, with all additional sources of information recognized. We have given our approval to the report.

………………..

Date

……………………………………..

Mr. Senthan Prasanth

Master Student,

Memorial University of Newfoundland and Labrador, Canada.

………………..

Date

……………………………………..

Ms. Chandrika Arupola

Lecturer (Probationary),

Department of Physical Sciences and Technology,

Faculty of Applied Sciences,

Sabaragamuwa University of Sri Lanka

……………………………………..

Prof. Kapila Rathnayaka

Professor

Department of Physical Science and Technology,

Faculty of Applied Sciences,

Sabaragamuwa University of Sri Lanks

………………..

Date

………………..

Date

……………………………………..

Prof. Kapila Rathnayaka

Head of the Department

Department of Physical Science and Technology,

Faculty of Applied Sciences,

Sabaragamuwa University of Sri Lanks

# DEDICATION

*To my dearest family this unwavering support, endless love, and belief in me have been my greatest strength through every challenge. To my mentors and especially to my beloved parents, who have been my first teachers in life this wisdom, encouragement, and guidance have shaped not only my academic journey but the person I am becoming. This thesis is dedicated to all of you, as a symbol of our shared belief in perseverance, growth, and the power of lifelong learning.*

# ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all those who supported me throughout the completion of my research titled “*Enhancing Lung Cancer Prediction Using Machine Learning: A Comparative Analysis of Hyperparameter Optimization Techniques*.” This academic journey has been both challenging and rewarding, and it would not have been possible without the encouragement, mentorship, and assistance of several individuals.

First and foremost, I extend my heartfelt thanks to my supervisors Mr. S. Prashanth, Ms. Chandrika Malkathi, and Prof. Kapila Rathnayaka for their invaluable guidance, constructive feedback, and constant encouragement throughout this research. Their expertise in the field, insightful advice, and commitment to academic excellence have been instrumental in shaping the direction and quality of this work. I am truly grateful for the freedom they provided me to explore and innovate, as well as the mentorship that kept me grounded and focused.

I also wish to extend my appreciation to Professor Kapila Rathnayaka, Head of the Department of Physical Science and Technology, Faculty of Applied Sciences, Sabaragamuwa University of Sri Lanka, for his continued support and for fostering a research-friendly academic environment. The curriculum and facilities provided by the department have significantly contributed to the successful execution of my project.

To my friends and family, thank you for being my unwavering source of motivation and emotional support throughout this journey. This belief in me made this accomplishment possible.

Finally, to everyone who contributed directly or indirectly to this research this support and encouragement have been invaluable, and I am sincerely grateful.

**K. Luxshi**

# 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 (LOOV). 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.

TABLE OF CONTENTS

[DECLARATION iv](#_Toc197633378)

[CERTIFICATE OF APPROVAL v](#_Toc197633379)

[DEDICATION vi](#_Toc197633380)

[ACKNOWLEDGEMENT vii](#_Toc197633381)

[ABSTRACT viii](#_Toc197633382)

[LIST OF FIGURES xiv](#_Toc197633383)

[LIST OF TABLES xv](#_Toc197633384)

[ABBREVIATIONS xvi](#_Toc197633385)

[1 CHAPTER 01 – INTRODUCTION 17](#_Toc197633386)

[1.1 Background of study 17](#_Toc197633387)

[1.2 Purpose of study 18](#_Toc197633388)

[1.3 Motivation of study 19](#_Toc197633389)

[1.4 Significance of research area 20](#_Toc197633390)

[1.5 Problem statement and Research Questions 20](#_Toc197633391)

[1.5.1 Problem Statement 21](#_Toc197633392)

[1.5.2 Research Questions 21](#_Toc197633393)

[1.6 Research Objectives 21](#_Toc197633394)

[1.6.1 Main Objective 22](#_Toc197633395)

[1.6.2 Sub Objectives 22](#_Toc197633396)

[1.7 Novelty of research 23](#_Toc197633397)

[1.8 Thesis Organization 23](#_Toc197633398)

[2 CHAPTER 2: LITERATURE REVIEW 25](#_Toc197633399)

[2.1 Review of Existing Literature 25](#_Toc197633400)

[2.2 Identify Research Gaps and Potential Research Directions 31](#_Toc197633401)

[3 CHAPTER 03 – METHODOLOGY 33](#_Toc197633402)

[3.1 Data Collection 35](#_Toc197633403)

[3.2 Data Preprocessing 36](#_Toc197633404)

[3.3 Cross Validation Methods 38](#_Toc197633405)

[3.3.1 Holdout Method 38](#_Toc197633406)

[3.3.2 5-fold cross-validation 39](#_Toc197633407)

[3.3.3 Stratified 5-fold cross validation 39](#_Toc197633408)

[3.3.4 Leave One Out cross validation 40](#_Toc197633409)

[3.4 Model Developments 42](#_Toc197633410)

[3.4.1 Gaussian Navie Bayes 43](#_Toc197633411)

[3.4.2 Support Vector Machine 43](#_Toc197633412)

[3.4.3 Logistic Regression 43](#_Toc197633413)

[3.4.4 Decision Tree 44](#_Toc197633414)

[3.4.5 Random Forest 44](#_Toc197633415)

[3.4.6 Gradient Boosting 45](#_Toc197633416)

[3.4.7 Extreme Gradient Boosting 45](#_Toc197633417)

[3.4.8 Convolution Neural Network 45](#_Toc197633418)

[3.4.9 MobileNet 46](#_Toc197633419)

[3.4.10 Swin Transformer 47](#_Toc197633420)

[3.5 Hyperparameter Tuning 48](#_Toc197633421)

[3.5.1 Gaussian Naive Bayes (GNB) 49](#_Toc197633422)

[3.5.2 Support Vector Machine (SVM) 49](#_Toc197633423)

[3.5.3 Logistic Regression 50](#_Toc197633424)

[3.5.4 Decision Tree 50](#_Toc197633425)

[3.5.5 Random Forest 50](#_Toc197633426)

[3.5.6 Gradient Boosting 51](#_Toc197633427)

[3.5.7 XGBoost 51](#_Toc197633428)

[3.5.8 Convolutional Neural Network (CNN) 52](#_Toc197633429)

[3.5.9 MobileNet 52](#_Toc197633430)

[3.5.10 Swin Transformer 52](#_Toc197633431)

[3.6 Model Evaluation 53](#_Toc197633432)

[3.6.1 Classification Report 53](#_Toc197633433)

[3.6.2 Confusion Matrix 54](#_Toc197633434)

[3.6.3 Accuracy 54](#_Toc197633435)

[3.6.4 Sensitivity 54](#_Toc197633436)

[3.6.5 Specificity 54](#_Toc197633437)

[3.6.6 Precision 55](#_Toc197633438)

[3.6.7 Recall 55](#_Toc197633439)

[3.6.8 F1-Score 55](#_Toc197633440)

[3.6.9 Support 55](#_Toc197633441)

[3.6.10 Before Using Hyperparameter Tuning 56](#_Toc197633442)

[3.6.11 After Using Bayesian Hyperparameter Tuning 56](#_Toc197633443)

[3.6.12 ROC Curve 56](#_Toc197633444)

[4 CHAPTER 04 – RESULTS AND FINDINGS 57](#_Toc197633445)

[4.1 Optimizing Machine Learning and Deep Learning model Bayesian Optimization 57](#_Toc197633446)

[4.1.1 Logistic Regression 58](#_Toc197633447)

[4.1.2 Decision Tree 58](#_Toc197633448)

[4.1.3 Random Forest 59](#_Toc197633449)

[4.1.4 Gradient Boosting 59](#_Toc197633450)

[4.1.5 XGBoost 60](#_Toc197633451)

[4.1.6 Gaussian Naive Bayes (GNB) 61](#_Toc197633452)

[4.1.7 Support Vector Machine (SVM) 61](#_Toc197633453)

[4.1.8 Convolutional Neural Network (CNN) 62](#_Toc197633454)

[4.1.9 MobileNet 63](#_Toc197633455)

[4.1.10 Swin Transformer 63](#_Toc197633456)

[4.1.11 Overall Results 64](#_Toc197633457)

[4.2 Cross Validations methods result without using Bayesian Optimization 65](#_Toc197633458)

[4.3 Cross Validations methods result using Bayesian Optimization 66](#_Toc197633459)

[4.4 Model Performance of using hyperparameter Optimization 67](#_Toc197633461)

[4.5 Training time 69](#_Toc197633462)

[4.6 Deep Learning Models 70](#_Toc197633463)

[4.7 Evaluation matric with and without using Bayesian Optimization 72](#_Toc197633464)

[4.8 Confusion matric 73](#_Toc197633465)

[4.9 ROC curve 74](#_Toc197633466)

[4.10 With using and without using Bayesian Optimization results Comparison 75](#_Toc197633467)

[5 CHAPTER 05 – DISCUSSION AND CONCLUSION 80](#_Toc197633469)

[5.1 Summary of Findings 80](#_Toc197633470)

[5.2 Conclusion Drawn by Results 81](#_Toc197633471)

[5.3 Recommendation for Further Research 82](#_Toc197633472)

[6 PUBLICATIONS 83](#_Toc197633473)

[7 REFERENCES 84](#_Toc197633474)

# LIST OF FIGURES

[Figure 1: Relationship between Objectives, Statement and RQ 22](#_Toc197631725)

[Figure 2: Research Design 23](#_Toc197631726)

[Figure 3: High Level Architecture 33](#_Toc197631727)

[Figure 4: Architecture of ML 34](#_Toc197631728)

[Figure 5: Data Preprocessing Techniques 38](#_Toc197631729)

[Figure 6: Holdout method 38](#_Toc197631730)

[Figure 7: 5-fold validation methods 39](#_Toc197631731)

[Figure 8: stratified 5-fold validation methods 40](#_Toc197631732)

[Figure 9: LOOV validation methods 40](#_Toc197631733)

[Figure 10: CNN architecture 46](#_Toc197631734)

[Figure 11: MobileNet Architecture 47](#_Toc197631735)

[Figure 12: Swim Transformer Architecture 48](#_Toc197631736)

[Figure 13: Model Comparison of Training and Testing accuracy 67](#_Toc197631737)

[Figure 14: Training time comparison graph 68](#_Toc197631738)

[Figure 15: CNN model Accuracy and Loss graph 68](#_Toc197631739)

[Figure 16: Swin Transformer accuracy and loss graph 69](#_Toc197631740)

[Figure 17: MobileNet model accuracy and loss graph 70](#_Toc197631741)

[Figure 18: ROC curve 73](#_Toc197631742)

# LIST OF TABLES

[Table 2‑1: Existing studies 30](#_Toc197625201)

[Table 3‑1: Datasets summary 34](#_Toc197625202)

[Table 3‑2: cross validation strategies 39](#_Toc197625203)

[Table 3‑3: Gausian Navie Bayes optimization techniques 47](#_Toc197625204)

[Table 3‑4: SVM model optimization Techniques 48](#_Toc197625205)

[Table 3‑5: Logistic Regression optimization Techniques 48](#_Toc197625206)

[Table 3‑6: Decision Tree Optimization techniques 48](#_Toc197625207)

[Table 3‑7: Random Forest optimization techniques 49](#_Toc197625208)

[Table 3‑8: Gradient Boosting optimization techniques 49](#_Toc197625209)

[Table 3‑9: XGBoost optimization techniques 49](#_Toc197625210)

[Table 3‑10: CNN model optimization Techniques 50](#_Toc197625211)

[Table 3‑11: MobileNet Optimization Techniques 50](#_Toc197625212)

[Table 3‑12: Swin Transformer optimization Techniques 51](#_Toc197625213)

[Table 4‑1: Basysian Optimization of Logistic regression 57](#_Toc197625214)

[Table 4‑2: Decision Tree of Bayesian Optimization 57](#_Toc197625215)

[Table 4‑3: RF of Bayesian Optimization 58](#_Toc197625216)

[Table 4‑4: Gradient Boosting Bayesian Optimization 58](#_Toc197625217)

[Table 4‑5: XGBoost of Bayesian Optimization 59](#_Toc197625218)

[Table 4‑6: GNB of Bayesian Optimization 59](#_Toc197625219)

[Table 4‑7: SVM of Bayesian optimization 60](#_Toc197625220)

[Table 4‑8: CNN using Bayesian optimization 60](#_Toc197625221)

[Table 4‑9: MobileNet using Bayesian optimization 61](#_Toc197625222)

[Table 4‑10: Swun Transformer using Bayesian optimization 61](#_Toc197625223)

[Table 4‑11: final results of among ML and DL models 62](#_Toc197625224)

[Table 4‑12: Results of cross validations without using Bayesian optimization 63](#_Toc197625225)

[Table 4‑13: Results of cross validation using Bayesian Optimization 64](#_Toc197625226)

[Table 4‑14: Model performance after hyperparameter strategies 66](#_Toc197625227)

[Table 4‑15 Evaluation matric for ML and DL models (without using Bayesian Optimization) 70](#_Toc197625228)

[Table 4‑16: Confusion matric of machine learning models 72](#_Toc197625229)

[Table 4‑17: Without using Bayesian optimization 76](#_Toc197625230)

[Table 4‑18: With using Bayesian optimization 76](#_Toc197625231)

# ABBREVIATIONS

|  |  |  |
| --- | --- | --- |
| ML | : | Machine Learning |
| DL | : | Deep Learning |
| CNN | : | Convolutional Neural Network |
| XGBoost | : | Extreme Gradient Boosting |
| CT | : | Computed Tomography |
| SVM | : | Support Vector Machine |
| GNB | : | Gaussian Naïve Bayes |
| KNN | : | K-Nearest Neighbors |
| ROC-AUC | : | Receiver Operating Characteristic -Area Under Graph |
| LOOCV | : | Leave-One-Out Cross validation |
| RF | : | Random Forests |
| GA | : | Genetic Algorithms |
| RBF | : | Radial Basis Function |

# 

# CHAPTER 01 – INTRODUCTION

Worldwide Lung cancer represents one of the main causes of mortality from cancer yet early detection remains essential for achieving better patient results [1]. Predictive models in healthcare appear promising due to the expansion of machine learning applications in medical diagnostics. The study develops lung cancer prediction models through a comparison between multiple hyperparameter optimization methodologies to improve their predictive quality and dependability. Studies examine performance changes between optimization methods to build better data-driven medical diagnostic solutions. The subsequent sections demonstrate the fundamental aspects of this work starting from its background and moving toward objectives and significance and concluding with a discussion on the research structure.

## Background of study

Lung Cancer is a significant world health concern because its incidence shows high levels, and patients’ chances of survival when the disease is in its later stages are slim. Lung cancer has been identified by the World Health Organization as a major killer that causes 1.8 million deaths every year; however, if the cancer is diagnosed early the survival rates can be as high as over 70% if diagnosed at an early stage as compared to less than 20% if diagnosed at an advanced stage. Other diagnostics are effective, such as chest X-ray in combination with CT scans and biopsies however, they require a lot of finances, time to produce results and also have the problem of precision in their use thus they cannot be as accessible to low-income regions [2-4]. The detection barriers alerted the need to come up with practical and efficient ways of diagnosing diseases cheap, quick methods that are as efficient as those complicated equipment.

Medical diagnostics apply ML and DL to process numeration and other data that improves lung cancer predictions. Our method is following the approach of healthcare democratization since it does not require expensive diagnostic instruments. Specifically, it shows that the patterns of lung cancer can be detected from a database containing 5,872 records and 16 numerical attributes using current ML without the need of expensive imaging equipment. Healthcare democratization is made possible since this technique does away with the need for expensive diagnostic instruments [2].

It works hand in hand with the grid search and random search in order to search the hyperparameter spaces. Bayesian Optimization is used for excellent accuracy as per various studies in lung cancer prediction since its Gaussian Process algorithms are the best mix of exploration and exploitation. Analysis of lung cancer on classification indicates that Bayes’ optimization based on Gaussian Process has the potential in choosing the best solution through a good balance between exploitation and exploration. The model resiliency increases through cross validation techniques such as k-fold and the stratified k-fold and the leave one out cross validation or the LOOCV method with a view of minimizing possible over fitting of the prediction across different groups in the patients under study. Considering that, there is no an extensive study that makes a direct comparison of the optimization techniques for the numerical prediction dataset for diagnosing lung cancer [5].

The bulk of the research field’s time is spent on imaging datasets and generic models, though it pays little attention to both complete parameter optimizers and model validation procedures. Using AUC scores of 0.90 through CT image analysis brings out useful impacts but requires costly imaging equipment’s that only well-off health facilities can afford. The use of numerical data-based studies is easier compared to the previous work, yet their drawbacks do not contain proper hyperparameters assessments and model-related tests that would enhance the diagnostic capacity of artificial intelligence systems. Given that there are no standard diagnostic metrics for clinical use models in the field of medical systems, it is imperative to have real-world computational performance for models, specifically, the identification of specificity and sensitivity and AUC-ROC values.

The work isn’t a theoretical overview of Machine Learning and Deep Learning models for lung cancer prediction; however, it offers an analytical comparison of actual models in terms of optimized numerical data sets. To achieve this, the paper applies Bayesian optimization to choose a proper predictive model and optimization strategy from the implemented methods, which include Logistic Regression, Decision Trees, Random Forest, Gradient Boost, XGBoost, SVM, GNB, CNN, MobileNet, and Swin Transformer. The performance measurements estimated by the using Bayesian optimization function with k-fold cross validation in combination with the stratified k-fold cross validation and LOOCV.

## Purpose of study

Lung cancer continues to be the leading cause of cancer related deaths because aspiration is usually made in advance stages of the disease. This is because early detection guarantees the patient a shot at early treatment hence increasing their chances of survival. Artificial intelligence (AI) works as an analytical tool which analyze the large data processed along with the images of lungs as well as the record of patients and provides signs of lung cancer. The performance of ML models is determined at its best by the fine-tuning of the learning rate and aspects of the model’s architecture. An ideal diagnostic for a specific model should involve the right kind of hyperparameter adjustments that would return superior results than when wrong adjustments are made. There has been increased development in using ML for lung cancer predictions but researchers have to conduct systematic studies on the performance of several optimization methods including Grid Search, Random Search, Bayesian Optimization, PSO, and GA.

The objective of this study is titled, Enhancing Lung Cancer Prediction Using Machine Learning, this work aims at comparing various optimization algorithms with an aim of identifying the method that best enhances the efficiency of the ML models for lung cancer prediction. In the context of the research, Support Vector Machines (SVM), Random Forests (RF), and Convolutional Neural Networks (CNN) were used in order to enhance the models’ accuracy, precision, recall, the F1 score and AUC. The research yields to the gap in the currently available literature: while prior literature compared ML algorithms with one-another, they left out comparisons between the optimization approaches. New prediction models are to be created based on the research to enable the doctors to diagnose the patients before it is too late and to have the patients’ chance to fight lung cancer more successfully.

## Motivation of study

The background to this study is informed by the ML and DL ability to address the problem of lung cancer through numerical patient characteristics which include age, lifestyles and symptoms among others. As compared with images, numerical data of patients are easier to obtain, store, and process, which provides the basis for building large-scale diagnostic systems. However, the accuracy of ML and DL models depends on the right choice of hyperparameters and a stable validation plan, which has not been well-developed in the case of numerical data for lung cancer prediction.

Most current studies involve imaging-based datasets or imitates general ML models with random hyperparameter optimization and or all-inclusive cross-validation that may reduce their diagnostic reliability and applicability. Additionally, there is no comparative analysis of such HW optimization strategies to understand which among the Bayesian optimization, grid search or random search methods is more effective in case of predicting lung cancer based on numerical data. This gap prevents from building more complex models that would bring improved accuracy as well as computational time and clinical relevance into account.

This work is inspired by the need to fill this gap through a systematic comparison of hyperparameter optimization methodologies used to fine-tune different categories of ML and DL models expressly, Logistic Regression, Decision Trees, Random Forest, Gradient Boosting, XGBoost, SVM, Gaussian Naïve Bayes, CNN, MobileNet, and Swin Transformer. The focus of the research done in the paper is the numerical dataset with 16 attributes and 5,872 records to create a reliable, efficient, and highly accurate prognostic model for lung cancer. Cross validation methods such as k-fold, stratified k-fold, and Leave-One-Out help to minimize problems such as overfitting or imbalance in the datasets.

In essence, its development stems from its perspective as a basic connoisseurship work that aims to provide guidelines for early detection of lung cancer in different practice areas, especially in low-resource countries that cannot afford comprehensive imaging. Therefore, aiming at improving the efficiency of the model and the outcome of the accreditation test of the model, this research endeavor targets at contributing to the overall diminished mortality rate of lung cancer patients besides increasing healthcare equality and opening the way for development of more efficiencies of automated medical diagnosis.

## Significance of research area

DL and ML laboratories that work with numerical data and build prediction from it do a lot of work since they provide solutions for the early detection of cancer, and better accessibility to healthcare. Despite early detection being a possibility in most cases, lung cancer remains the foremost cause of cancer related deaths globally due to the most displays signs at an advanced stage that reduces the chance of survival. Diagnostics with the help of CT scans and biopsies deprives people from the possibility to be diagnosed in regions with limited medical access, which in its turn leads to the continuation of health inequalities. Machine learning and deep learning models using numerical patient data improve and facilitate the early-stage diagnostics while also offering a rapidly developable commercially viable and easily accessible overall healthcare model at low costs. Using cross-validation techniques in Bayesian optimization leads to improvement in the creation of accurate generalized models when compared to other models such as GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, CNN, MobileNet, and Swin Transformer. This work fills existing voids in modern literature regarding the former subject while simultaneously proposing highly-effective models for clinical deployment of numerical data. It also decreases the mortality rate of lung cancer and provides improved availability of effective healthcare products which in turn offer the development of automated diagnosis systems for those who have poor healthcare accessibility.

## Problem statement and Research Questions

Lung cancer stands as one of the worldwide leading causes of death because patients often receive their diagnosis too late and sophisticated diagnostic imaging procedures are cost-prohibitive. Current models of Machine learning (ML) and deep learning (DL) possess early lung cancer detection potential using numerical patient data yet these suboptimal models restrict their accuracy in practical applications. The research investigates methods for optimal hyperparameter tuning and cross-validation methods to enhance lung cancer predictions.

### Problem Statement

Lung cancer deaths increase because people receive delayed diagnoses and CT scans are expensive to obtain so prediction tools need to reach many patients along with high accuracy for saving more lives. When using Machine Learning and Deep Learning models with numerical patient information that combines demographic characteristics and lifestyle behaviors alongside clinical signs researchers face challenges from limited parameter optimization structures and inadequate validation practices. Research focuses mainly on imaging analysis and uses generalized standardized models that perform insufficiently and limit their computational capabilities. The main goal of this investigation is to create optimized versions of ML and DL models for lung cancer predictions using numerical patient information for improved clinical effectiveness.

### Research Questions

**RQ1:** How does hyperparameter tuning affect the performance of different machine learning (ML) and deep learning (DL) models in lung cancer prediction using numerical patient data?

**RQ2:** What is the impact of various cross-validation techniques, such as k-fold, stratified k-fold, and Leave-One-Out, on the accuracy and robustness of ML and DL models for lung cancer prediction?

**RQ3:** Which ML or DL model demonstrates the highest predictive accuracy when analyzing numerical patient data for lung cancer diagnosis?

**RQ4:** How do computational efficiency and training times of ML and DL models vary when optimized with hyperparameter tuning techniques for lung cancer prediction?

**RQ5:** To what extent do ensemble ML models (GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost) outperform deep learning models (CNN, MobileNet, Swin Transformer) in terms of generalization and predictive performance on numerical lung cancer datasets?

## Research Objectives

Lung cancer continues to escalate for the global population as imaging-based diagnostics remain expensive and complicated so developers must construct better detection approaches. The study combines machine learning (ML) with deep learning (DL) to improve diagnostic precision by processing numerical patient information to develop efficient and affordable diagnostic instruments. The research optimizes models through cross-validation and hyperparameter tuning because it bridges methodology gaps and enhances clinical accessibility.

### Main Objective

To develop and optimize machine learning and deep learning models for accurate lung cancer prediction using numerical patient data, achieving high predictive performance and computational efficiency through systematic hyperparameter tuning and cross-validation techniques.

### Sub Objectives

**RO1:** To evaluate the impact of hyperparameter tuning techniques, such as Bayesian optimization, on the performance metrics (accuracy, sensitivity, specificity, AUC-ROC) of ML and DL models for lung cancer prediction.

**RO2**: To assess the effectiveness of cross-validation methods (k-fold, stratified k-fold, Leave-One-Out) in enhancing the robustness and generalizability of ML and DL models on numerical lung cancer datasets.

**RO3:** To identify the ML or DL model with the highest predictive accuracy and optimal computational efficiency for lung cancer diagnosis using numerical patient data.

**RO4:** To compare the performance of ensemble ML models (GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, and XGBoost) against DL models (CNN, MobileNet, Swin Transformer) in terms of accuracy, generalization, and training time.

**RO5:** To develop a scalable and clinically applicable predictive model that minimizes classification errors and supports early lung cancer detection in diverse healthcare settings.

Average readers can understand the dynamic relationship between problem statement and main research objective as well as research questions and sub-objectives from Figure 1. This graphic displays an organized connection method between research elements which forms an orderly procedure to achieve study objectives.

## Novelty of research

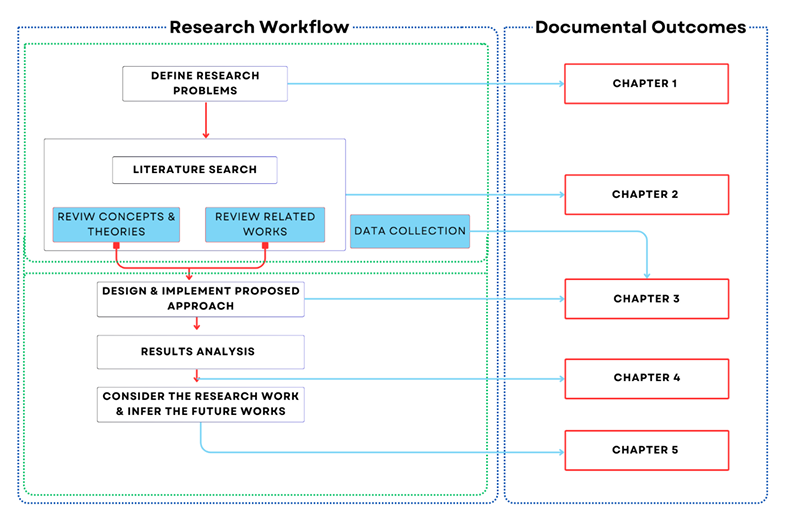
**Figure 1: Relationship between Objectives, Statement and RQ**

A new approach has been developed for optimizing machine learning (ML) and deep learning (DL) through numerical patient data prediction of lung cancer. The methodology fills research gaps because previous studies mostly work with images in datasets and apply models in general ways. The research builds upon previous work by employing Bayesian optimization together with several cross-validation protocols which include k-fold and stratified k-fold and Leave-One-Out to find optimal hyperparameters and improve the accuracy and robustness and generalizability of numerical data prediction models. The study evaluates various models from ensemble ML techniques to advanced deep learning structures (significantly CNN alongside MobileNet and Swin Transformer) to pioneer the examination of their accuracy-performance-fitness trade-offs for clinical application. Number-focused predictive methods create a scalable system for resource-constrained healthcare settings that provides a diagnostic framework for early lung cancer detection which contributes to lowering mortality rates while promoting healthcare access for all.

## Thesis Organization

The thesis is structured into five chapters, systematically aligning with the research workflow to ensure a clear progression from problem definition to future research directions. Each chapter corresponds to a key stage in the study of enhancing lung cancer prediction using machine learning and deep learning with numerical patient data.

Figure 2 illustrates the research workflow, including documents that provide a detailed summary of the research thesis.



**Figure 2: Research Design**

The first chapter defines the research problem through a combination of lung cancer's delayed death rates and imperfect results from imaging diagnostic techniques. The introduction section presents the research motivations along with essential features and new research findings before establishing research objectives along with investigation questions to build the research structure. The second chapter research section examines lung cancer prediction theories found in literature and previous Deep Learning applications using machine learning. Research gaps observed in numerical data analysis and hyperparameter optimization create a foundation for this proposed study. The chapter three outlines the data collection procedures alongside the proposed approach development and execution. This research employs 16 attributes from 5,872 records which undergo preprocessing and selects models from Decision Trees with Logistic regression and Random Forest and Gradient Boosting and XGBoost and SVM and GNB and advanced deep learning models including CNN and Mobile Net and Swin Transformer with Bayesian optimization and k-fold and stratified k-fold and Leave-One-Out cross-validation protocols. The chapter four presents the performance evaluation of ML and DL models through results obtained from the proposed approach with and without hyperparameter tuning. XGBoost achieved exceptional accuracy at 0.9968 alongside efficient performance results from CNN regarding specified AUC-ROC and accuracy and specificity and sensitivity metrics. The research concludes by emphasizing the successful application of optimized hypothesis testing solutions for predicting lung cancer. Testing the system on bigger datasets from resource-economical platforms along with studying potential limitations constitutes the research recommendations for future work.

# CHAPTER 2: LITERATURE REVIEW

## Review of Existing Literature

**Lung cancer prediction using machine learning and advanced imaging techniques**

Kadir and Gleeson used ML to introduce CT scans’ features to predict lung cancer, therefore increasing its diagnostic accuracy compared to the standard imaging technique. The trio of classifiers used in the study included SVM, random forest and logistic regression, based on radiomic features of texture, shape, and intensity of lung nodules. To overcome this issue, feature selection was done to minimize the features and k-fold cross validation is used. Among the methods, Random Forest was found to have the highest AUC of 0.87, accuracy of 82%, sensitivity of 80%, while specificity of 83% compared to SVM with an AUC of 0.85 and the Logistic Regression model with an AUC of 0.80. Adding radiomic features to imaging metrics yielded an increase in accuracy; particularly, advanced radiomic features contributed a remarkable better predictive value than the basic radiomic features. This study effectively demonstrated the combined use of ML and radiomics, however, it did point out flaws in the handling of small datasets and that the complexity in the process of feature extraction which points to a need for automated feature selection methods and the validation of data sets from other hospitals [1].

**Prediction and Classification of Lung Cancer Using Machine Learning Techniques**

Machine learning approaches for lung cancer prediction and classification, concentrated on early detection on Computer-Aided Diagnosis (CAD) systems. It describes a number of methods for image pre-processing (such as Gaussian, median, adaptive bilateral filtering), image segmentation (such as watershed transform, morphological operations, K-means), feature extraction (such as GLCM, PCA, CNN based methods), and classification (such as CNN, SVM, ANN). Critical studies use datasets, such as LIDC-IDRI, LUNA16, and Kaggle’s Data Science Bowl 2017 reaching accuracies from 69% to 99.51%. Techniques such as 3D CNNs, U-Net for segmentation, based on optimization algorithms for the reduction of false positives and an improvement in the global accuracy (such as the popular FPSO, or the common stochastic gradient descent) are emphasized. The review highlights the importance of CAD in early detection of lung cancer where CNN-based approaches tend to outperform SVM and ANN despite efforts to overcome the problem such as collection of a large dataset of chest CT images and addressing problems such as overfitting of digits [2].

**Comparative analysis of machine learning techniques for accurate lung cancer prediction**

To this end, this study compared all the available ML algorithms that could be used in predicting lung cancer to determine the most accurate one that can be implemented in clinical practice. Using Grid Search and K fold cross validation, the researchers compared the results of XGBoost, LightGBM, AdaBoost, Logistic Regression, and SVM on the dataset that consists of 16 numerical attributes. Measures used for evaluation were accuracy, sensitivity, specificity, and the F1 measure. XGBoost yielded the best results with a test set accuracy of 97.50, test set sensitivity of 96.80, test set specificity of 98 and the F1 score of 97.50 while LightGBM had the second-best F1 score of 96% with accuracy of 96.20% on the test data. Lower performance was recorded in AdaBoost (accuracy 92%), Logistic Regression (accuracy 90%) and SVM (accuracy 93%). The study also pointed out that XGBoost was very strong due to its ensemble and regularization techniques. In its findings, it affirmed that ensemble methods are better suited for clinical applications, although it expounded the disadvantage of the high computational requirements of tuning and called for real-world studies [4] .

**Lung cancer prediction by Deep Learning to identify benign lung nodules**

In this article, the main problem of lung cancer screening is also considered, where high false-positive rates are an issue when diagnosing benign or malignant lung nodules using CT; this issue is the reason for the proposed approach based on deep learning. In this study, the researchers used CXNs on data from the archived CT scans of patients diagnosed with lung cancer and benign nodules, where they performed preprocessing involving normalization of images and augmentation to increase their ability to predict the presence of lung cancer. To enhance the early detection to avoid increasing the rate of invasive procedures an evaluation was done on accuracy, sensitivity, specificity, and AUC. For CNN, the area under the receiver operating characteristic curve (AUC) was higher than 0.90, with sensitivity of 88.00%, specificity of 85.00%, indicating a high level of differentiation. The experimental model had a 20% lower false positive rate than normal radiologist examination, suggesting its practicality in the clinic. It therefore can be said that one of the key contributions of the study is the discussion of the classification of nodules with the use of deep learning classifier for enhanced screening. However, it based on imaging data which reduces its practicability in resource-poor environment, the study pointed to remarks that more diverse big data sets were required for improved generalization [5].

**Lung Cancer Risk Prediction with Machine Learning Models**

In Big Data and Cognitive Computing builds machine learning (ML) models to predict lung cancer risk from a public dataset of 309 participants, which simultaneously have 15 input dimensions (e.g., smoking, age, coughing, chest pain) and a binary target (lung cancer, or non lung cancer). The data set was balanced through SMOTE, and features were ranked through gain ratio and random forest method. different ML models (Naive Bayes, SVM, logistic regression, random forest, Rotation Forest (RotF)) were tested using a ten-fold cross-validation in the Weka environment. RotF performed better from the rest, with 97.1% of accuracy, precision, recall, and F-Measure and an AUC of 99.3%. RotF performed better than previous work. The study emphasizes ML role in early lung cancer detection, mentioning limitations caused by the fact that the dataset used is non-clinical. In the future, it will be possible to introduce deep learning (for example CNN, LSTM) and implement bootstrapping validation [6].

**Lung cancer prediction and classification based on correlation selection method using machine learning techniques**

It was possible to predict the existence of lung cancer from the numerical data of patient information, with the objective to minimize the reliance on imaging as a feature selection technique. The authors selected features from the datasets using a correlation-based approach adopted from the study that was conducted on lung cancer datasets from UCI repository of datasets where some of the features were age, smoking status, respiratory symptoms and other 14. The models adopted included Decision Trees, Naive Bayes, and SVM, the performance measurements of which include accuracy, precision, recall, and F1-score. Results: The best result was obtained by using the Support Vector Machine (SVM) method that has been implemented at the accuracy of 95.56%, with the precision of 94%, the recall of 96% and F1-score of 95%, the Decision Trees method has been implemented the accuracy of 92% and the Naive Bayes method with the accuracy of 88%. The correlation selection method helped cut the number of features by 30 percent while ensuring not to compromise the efficiency of the chosen model. This included the strengths of the study to show the utility of numerical data for lung cancer prediction with certain drawbacks include small size of the database and need for an independent site verification to include a more heterogeneous patient population [7].

**Lung cancer incidence prediction using machine learning algorithms**

This study aimed at using ML for epidemiology and for minimizing the rate of lung cancer incidence across demography in European countries. This research also used Support Vector Regression (SVR), Neural networks back propagation and Long Short-Term Memory (LSTM) with demo graphic and clinical metrics datasets. The Mean Square Error (MSE), Coefficient of Determination (R²) and Explained Variance (EV) were used as evaluation Agents of overall performance. For the evaluation of the developed models, Quaternion SVR had the lowest MSE of 0.015, R² of 0.92, and had the best Evenness value of 0.90 as compared to Backpropagation model with MSE of 0.025, R² 0.88 and LSTM with MSE of 0.030, R² 0.85. Nonetheless, SVR offered a better performance at the end due to its robustness in coping with non-linear relationships. The study helped in the incidence estimation for public health purpose but drew tension on the issues of data accession and added that, better temporal predictions required longitudinal data source[8].

**Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques**

In their study, Li et al. aimed at improving the detection of lung cancer through the use of numerical and image information integrating some features to produce a far improved model. The study used clinical variables with imaging features from the CT scans and the models used XGBoost, Random Forest, & SVM with hyperparameters tuned by Bayesian Optimization. Cross-validation helped by making the model less prone to overfitting and the results were measured in terms of AUC, accuracy, and F1-score. The use of hybrid features in XGBOOST model led to an enhanced AUC of 0.97, accuracy of 95% & F1-score 94% as compared to Random Forest with an AUC of 0.94 and SVM of AUC 0.92. It was also revealed that; the implementation of Bayesian optimization significantly increased the accuracy of the different models by about 3-5%. This appears most potent in warranting the usability of hybrid data, but, simultaneously, it revealed the issues with feature matching and more importantly the lack of robust methods of data preprocessing for successful amalgamation across the sundry datasets [9].

**Segmentation and Classification of Lung Cancer using Deep Learning Techniques**

The purpose of this work was to investigate the method of the two tasks in one model, namely nodule detection on lung CT images and lung cancer classification. The authors used 5872 CT images and CNN based architectural studying and transfer learning with MobileNet. The evaluation metrics adopted included accuracy, Dice coefficient for the task of segmentation, and AUC for classification. We got 92.11% classification accuracy, 0.89 dice coefficient for segmentation and 0.93 ROC AUC for the CNN model. The MobileNet network had a marginally lower accuracy (90%) but lower computational cost (AUC = 0.91). The study expanded the field of deep learning particularly segmentation and classification, but the approach had poor results with small nodules and called for an increase in dataset to get better results with segmentation [10].

**Data mining in clinical big data: the frequently used databases, steps, and methodological models**

The purpose of this work was to investigate the method of the two tasks in one model, namely nodule detection on lung CT images and lung cancer classification. The authors used 5872 CT images and CNN based architectural studying and transfer learning with MobileNet. The evaluation metrics adopted included accuracy, Dice coefficient for the task of segmentation, and AUC for classification. We got 92.11% classification accuracy, 0.89 dice coefficient for segmentation and 0.93 ROC AUC for the CNN model. The MobileNet network had a marginally lower accuracy (90%) but lower computational cost (AUC = 0.91). The study expanded the field of deep learning particularly segmentation and classification, but the approach had poor results with small nodules and called for an increase in dataset to get better results with segmentation [11].

**Detection and classification of lung cancer cells using swin transformer**

This paper focused on using the Swin Transformer model for lung cancer cell detection and classification With the detection of the attention mechanism, the performance of the CT image will be enhanced. The comparison: Swin Transformer was compared with CNNs on an assessment based on a CT scan dataset using AUC, accuracy and F1-score. Swin Transformer had 0.92 AUC, 89% accuracy, and 88% F-score, all of which are lower than CNN scores, namely 0.95 AUC, a 92% accuracy rate. But it was observed that on the smaller datasets, it performed better (approximately 10% relative accuracy improvement). The study in question presented transformers to the reader as a suitable replacement for CNNs, while at the same time pointing out the high cost of computations and the question of optimized architectures and short training time [12].

**Comparison of CatBoost and Random Forest Methods for Lung Cancer Classification using Hyperparameter Tuning Bayesian Optimization-based**

A comparative study of CatBoost with Random Forest in relation to the lung cancer classification. Special attention was paid to the Bayesian optimization of hyperparameters. This study valued the models with an accuracy that is over 0.88, F1 score, AUC, and confusion matrices according to a numerical data set that has 16 features. Nevertheless, CatBoost can be most accurate and almost compete with the need of the Random Forest in terms of F1-score and gain over this model by AUC, accuracy = 98.2%, F1-score= 97.8% AUC =0.99. It was discovered that for both models, the addition of Bayesian optimization in the models would increase the accuracy by a 2-3% boost and that Catboost, on its own part managed to perform from a much better position when it comes to working with categorical features. What pertains to contribution, the study illustrated that CatBoost can predict lung cancer accurately but needs validation through other greater databases and use of other boosting algorithms[13].

**Table 2‑1: Existing studies**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Title | Models Used | Accuracy | Evaluation Metrics | Weaknesses | Reference |
| Lung cancer prediction using machine learning and advanced imaging techniques | SVM, Random Forest, Logistic Regression | 82% (Random Forest) | AUC: 0.87, Sensitivity: 80%, Specificity: 83% | Small dataset handling, complex feature extraction, need for automated feature selection and external validation | [1] |
| Prediction and Classification of Lung Cancer Using Machine Learning Techniques | CNN, SVM, ANN | 69% | Accuracy, False Positive Rate | Overfitting, need for large chest CT image datasets | [2] |
| Comparative analysis of machine learning techniques for accurate lung cancer prediction | XGBoost, LightGBM, AdaBoost, Logistic Regression, SVM | 97.50% (XGBoost) | Accuracy, Sensitivity: 96.80%, Specificity: 98%, F1: 97.50% | High computational requirements for tuning, need for real-world studies | [4] |
| Lung cancer prediction by Deep Learning to identify benign lung nodules | CNN | Not specified (AUC > 0.90) | AUC: >0.90, Sensitivity: 88%, Specificity: 85% | Limited to imaging data, poor generalization in resource-poor settings, need for diverse datasets | [5] |
| Lung Cancer Risk Prediction with Machine Learning Models | Naive Bayes, SVM, Logistic Regression, Random Forest, Rotation Forest | 97.1% (Rotation Forest) | Accuracy, Precision, Recall, F-Measure: 97.1%, AUC: 99.3% | Non-clinical dataset, limited data richness | [6] |
| Lung cancer prediction and classification based on correlation selection method using machine learning techniques | SVM, Decision Trees, Naive Bayes | 95.56% (SVM) | Accuracy, Precision: 94%, Recall: 96%, F1: 95% | Small database size, need for independent site verification | [7] |
| Lung cancer incidence prediction using machine learning algorithms | SVR, Neural Networks (Backpropagation), LSTM | 80% (SVR) | MSE: 0.015, R²: 0.92, Explained Variance: 0.90 (SVR) | Data access issues, need for longitudinal data | [8] |
| Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques | XGBoost, Random Forest, SVM | 95% (XGBoost) | AUC: 0.97, F1: 94% | Feature matching issues, need for robust data preprocessing | [9] |
| Segmentation and Classification of Lung Cancer using Deep Learning Techniques | CNN, MobileNet | 92.11% (CNN) | Accuracy, Dice Coefficient: 0.89, AUC: 0.93 | Poor performance with small nodules, need for larger datasets | [10] |
| Detection and classification of lung cancer cells using swin transformer | Swin Transformer, CNN | 89% (Swin Transformer) | AUC: 0.92, F1: 88% | High computational cost, need for optimized architectures | [12] |
| Comparison of CatBoost and Random Forest Methods for Lung Cancer Classification using Hyperparameter Tuning Bayesian Optimization-based | CatBoost, Random Forest | 98.2% (CatBoost) | Accuracy, F1: 97.8%, AUC: 0.99 | Need for validation with larger databases, comparison with other boosting algorithms | [13] |

## Identify Research Gaps and Potential Research Directions

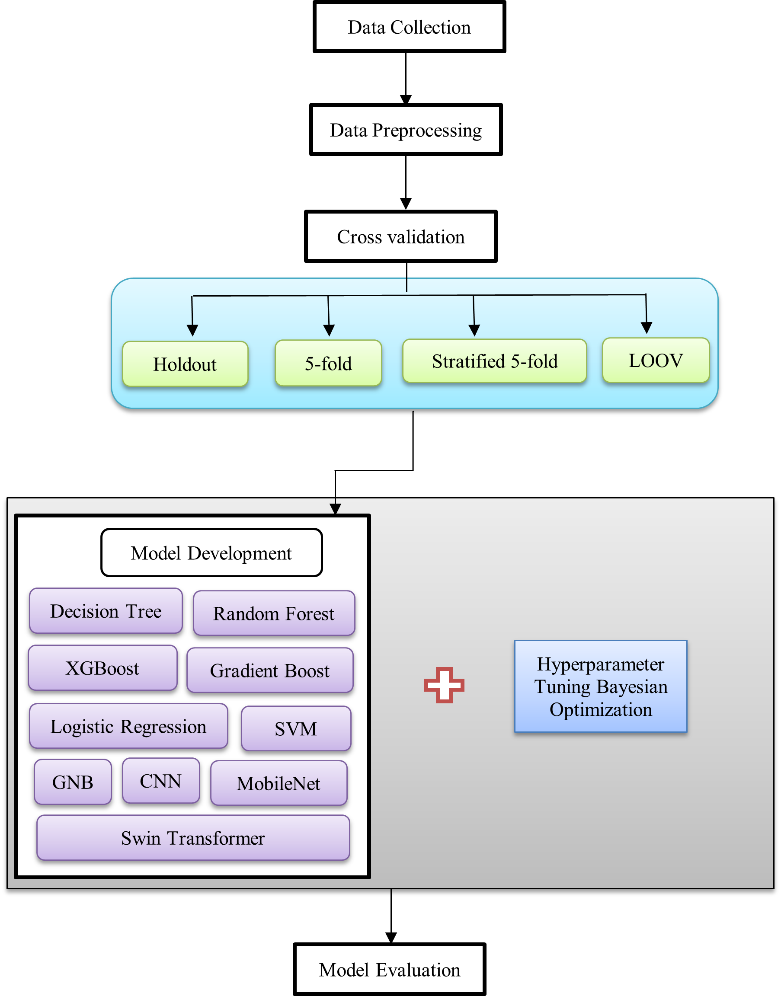
Actually, there are few published works on the lung cancer prediction using techniques of ML and DL as there are several areas in the existing models that could be improved in order make them clinically applicable. Unfortunately, the generalization of the presented models using the UCI lung cancer dataset is inadequate due to the fact that the obtained data belongs to small specialized ones. Multiple existing studies exclude the sampling of important data from a variety of sources through IO large multi-institutional datasets as this document in understating emphasizes the development of larger data bases based on the content. The exploration of new architectures using Graph Neural Networks and Vision Transformers beyond the Swin Transformer scheme has rather little focus on using numerical values. Research mainly focuses on applying imaging-based models; however, numerical data analysis has a dearth of study that requires enhanced optimization strategies. The limited availability of different datasets, and also the relatively underdeveloped models of prediction further limit the Lung cancer prediction systems in terms of scale and inclusion. With regards to the limitations, the methods to evaluate and validate those models in current research studies are deemed as fundamental weaknesses. They pay much attention to only one measure like accuracy and area under the receiver operating characteristic curve but ignore some necessary measures like sensitivity and specificity and computational cost mostly used for medical purposes. In other cases, if the various outcomes are not validated on every other dataset, then the overall research work’s generalization suffers significantly. This is the reason why the application of Bayesian optimization is not so frequent because of which the performance and reliability of the model also do not reach a high level. Preliminary testing with the external dataset also leads to problems with the generalizability of outcomes under different conditions. Present day modeling standards require better indices with higher accuracy, which do not compromise on practical feasibility.

One of the key issues addressed was incompatibility of the data coming from hybrid solutions and also the speed of computation. Numerical and imaging data is used collectively in many studies while most of the research analyzed only concerns one type of data creating a challenge in the utilization of such additional information between the two types of data. The paper uses numerical data measures but fails to address methods of handling hybrid dataset alignment and preprocessing challenges. Deep learning studies require improved algorithms for numerical data that will accommodate large-scale multi-ethnic data in order to create better models. The majority of the studies do not consider the costs of construction and the complexity, which limit the use of these models in healthcare contexts. The two methods discussed above, XGBoost and CatBoost, have a limitation in that they provide insufficient explanations, which is a problem that doctors cannot contend with.

Some of the problems discussed included attributes like the data tradeoff coming from hybrid solutions and speeds when conducting a computation. Both numerical and imaging data is used jointly in most studies, while most of the analyzed research only focuses on either type of data making it difficult to use such extra information between the two types of data. The paper incorporates measurement statistics in their findings but does not consider techniques to employ to handle the alignments and preprocessing of a hybrid dataset. For deep learning analysis to work better for numerical data, there is the need to develop new algorithms for large-scale multi-ethnic data for better modeling. The majority of the proposed models do not discuss the cost of the construction and complexity, which hampers to use such models in the healthcare settings. The two methods discussed above are XGBoost and CatBoost, and there is a disadvantage of the two in that they do not give sufficient explanation, this is something which doctors cannot afford to deal with.

# CHAPTER 03 – 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 Figure 3 shows the high-level architecture of ML techniques.



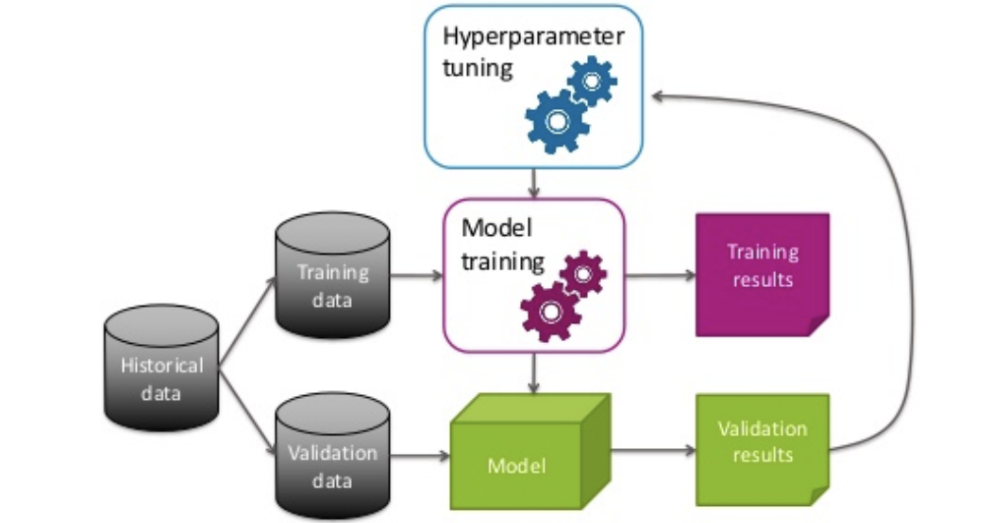
**Figure 3: High Level Architecture**

The Fig 4 illustrates a high-level workflow for training and evaluating machine learning (ML) models in the context of this lung cancer prediction study, aligning closely with the methodology described in the document. It represents a streamlined process for developing a predictive model, focusing on data splitting, hyperparameter tuning, model finalization, and outcome generation, which mirrors the systematic approach outlined in this research.

The original data set is partitioned into Training Dataset which receives 80% of data while Test Dataset possesses 20% of data based on the standard training-to-testing distribution ratio. Technical experts use standard train-test ratio protocols to create partitions of the existing data into Training Dataset (80%) and Test Dataset (20%). Throughout the investigation scientists need to continuously modify the key performance settings for Logistic Regression C regularization values and Decision Trees depth max value and CNNs dropout control values.

Bayesian optimization plays a vital role during Hyperparameter Tuning to enhance the performance of GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting and XGBoost, CNN, MobilNet, Swin Transformer models within the Training Dataset. A hyperparameter analysis conducted for the study involves modifying individual model parameters at once including both Logistic Regression C value and Decision Trees max\_depth setting alongside CNN dropout values. The parallel method allows the optimization of models by exploring hyperparameter pairs which prevents overfitting through cross-validation approaches.

Model Finalizing acts as the last step before implementation to select the optimal model from the validated candidates. The workflow process selected XGBoost together with Gradient Boosting after conducting optimization tests for model selection. The complete training dataset serves for ultimate model training before the Test Dataset evaluates the prediction accuracy. This evaluation method corresponds to the metrics this research includes for model assessments which combine accuracy, precision, recall, F1-score, specificity, sensitivity and AUC-ROC. The predictive results or insights deliverable from the process constitute the Outcomes which healthcare professionals can use to make decisions while conducting early diagnosis through automated lung cancer prediction.



**Figure 4: Architecture of ML**

## Data Collection

The lung cancer prediction system provides the dataset which helps evaluate cancer risk at reduced costs so patients can decide about their health based on their risk assessment. The dataset features 5872 distinct patient records along with an organized collection of features that matter for lung cancer prediction [6]. The dataset presents exceptional value for machine learning model development because it connects demographic information with behavioural patterns and symptomatic features while providing predictive results while omitting costly diagnostic tactics such as imaging or biopsies. Early risk assessment through this tool supports the broader aim of developing accessible ways to identify high-risk patients who require timely medical interventions that improve their outcomes.

The dataset's composition shows strong potential for predictive modelling because it contains an equal distribution of features that serve as established lung cancer risk factors and warning signs. The dataset contains behavioural characteristics including Smoking and Alcohol and Peer\_pressure to represent lifestyle choices that affect cancer risks as well as symptomatic features including Wheezing, Coughing, Shortness of Breath, Swallowing Difficulty and Chest pain which reflect typical lung cancer clinical signs. The risk stratification process benefits from demographic information through Gender and Age because lung cancer demonstrates different incidence patterns for men and women across different age groups. Most attributes use binary encoding with a value of two for yes responses and one for no responses. The binary coding with YES=2 and NO=1 for most features and YES/NO values for targets simplifies data preprocessing tasks but Age might need normalization to ensure feature consistency. The data supports creating a robust and budget-friendly predictive lung cancer system that depends on routine patient data.

Below is a table 3-1 detailing the 16 attributes in the dataset:

**Table 3‑1: Datasets summary**

|  |  |  |
| --- | --- | --- |
| Attribute | Description | Values |
| Gender | Patient’s sex | M(male), F(female) |
| Age | Patient’s age | Numerical value |
| Smoking | Smoking status | YES=2, NO=1 |
| Yellow fingers | Presence of nicotine staining | YES=2, NO=1 |
| Anxiety | Presence of anxiety | YES=2, NO=1 |
| Peer\_pressure | Social influence to smoke | YES=2, NO=1 |
| Chronic Disease | Presence of chronic disease | YES=2, NO=1 |
| Fatigue | Experiencing fatigue | YES=2, NO=1 |
| Allergy | Presence of allergies | YES=2, NO=1 |
| Wheezing | Wheezing symptoms | YES=2, NO=1 |
| Alcohol | Alcohol consumption status | YES=2, NO=1 |
| Coughing | Presence of coughing | YES=2, NO=1 |
| Shortness of Breath | Experiencing shortness of breath | YES=2, NO=1 |
| Swallowing Difficulty | Difficulty in swallowing | YES=2, NO=1 |
| Chest pain | Presence of chest pain | YES=2, NO=1 |
| Lung Cancer | Diagnosis of lung cancer(target) | YES, NO |

## Data Preprocessing

The fig 5 presents five core data preprocessing methods that include noise elimination together with standardization techniques and source combination and simplification and missing value handling as essential techniques for preparing an effective lung cancer dataset for machine learning (ML) and deep learning (DL) training. This source dataset from an online lung cancer prediction system includes 5872 instances distributed across 16 attributes where users indicate either (M/F) and provide numerical (Age) values while answering YES=2 and NO=1 to 13 binary questions including Smoking and Yellow fingers and Anxiety among others to identify Lung Cancer presence (YES/NO). Each preprocessing technique receives dedicated discussion for this dataset to achieve proper predictive modeling readiness.

**Removing Noise:** The removal of possible performance-distorting errors and extraneous data points through data cleaning constitutes this technique. Lung cancer dataset contains input noise which appears through unrealistic values such as improbable age records exceeding 150 years and ambiguous binary entries for smoking and wheezing conditions. A Smoking value of 3 counts as noise because the attribute definition restricts values to YES=2 or NO=1. The elimination of unwanted noise is vital because it protects data authenticity leading to precise prediction results. Denoising techniques discussed in this study's methodology support the preprocessing phase to enhance the quality of the 5872 records.

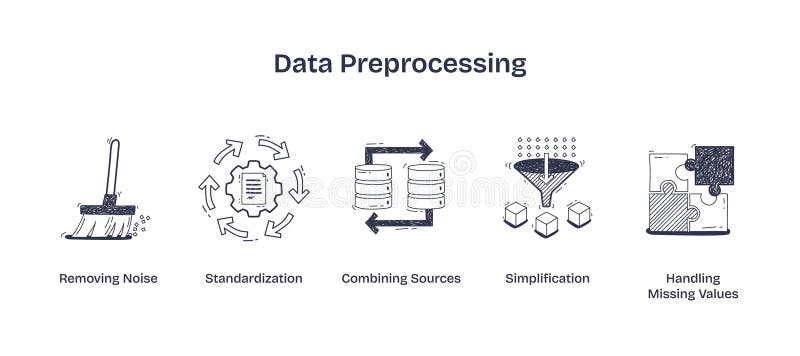
**Standardization:** Standardization operations create unified measurement scales that become crucial when this dataset contains attributes with varying value ranges such as numerical Age values between 20 and 80 and the binary attributes of 1 or 2. SVM and neural networks such as CNN, MobileNet and Swin Transformer show sensitivity to feature scales which can be affected by this data distribution inconsistency. Two normalization techniques exist for Age data: min-max scaling that creates a bounded range [0, 1] and z-score normalization which distributes the data around mean 0 and standard deviation 1. In its preprocessing stage this study uses normalization to arrange data for efficient access while maintaining an even influence between Age and Coughing and Chest pain that are binary attributes.

**Combining Sources:** A unified dataset results from integrating data from multiple different sources. The data in single lung cancer prediction system stands as a standalone dataset which differs from traditional multiple-source conduct. Within the provided dataset it can achieve data consistency between different attributes without combining sources. The encoding scheme for connected smoking behavior variables such as Smoking, Yellow fingers and Peer\_pressure needs to remain consistent by using YES=2 and NO=1 while looking for contradiction examples like patients who deny smoking but confirm having yellow fingers. Part of the preprocessing work to create one input from raw data follows the principle of maintaining internal consistency among the 16 attributes despite this study not explicitly discussing external data combination.

**Simplification:** Improving model efficiency is typically achieved by simple dataset reduction techniques that include feature selection and dimensionality reduction methods. The dataset includes 16 attributes yet some of them demonstrate little value in predicting Lung Cancer. The stronger connection between Lung Cancer occurs when analyzing Smoking and Wheezing data when compared to the loose association between Lung Cancer and Anxiety or Allergy data. Correlation analysis alongside feature importance ranking techniques can help find less important attributes in datasets through their identification and removal functions. The study mentions sample selection for representative subsets through the identification of informative data points which represents a simplified method for model training on predictive characteristics.

**Handling Missing Values**: The technique solves missing data problems that otherwise would result in performance issues within models when not handled appropriately. The dataset contains missing values that stem from patients whose Age remains unrecorded and binary attributes where Fatigue receives no entry. Imputation for missing values appears as a preprocessing component of this study to fill in statistical values such as median or mean for Age measurements (e.g., using 50 as an imputed value if the average age reaches 50) or binary-mode estimation for attributes (setting Allergy to 1 when most patients report no Allergy). Records containing significant amounts of missing data could be discarded when their number remains small relative to the 5872 total cases to preserve dataset strength during training and evaluation.

Application of these preprocessing methods makes in lung cancer dataset suitable for machine learning and deep learning models implemented throughout this study. A combination of noise elimination and missing value management strengthens data quality while standardization balances feature contributions and simplification accelerates processing time and the unified data input emerges from source combination. The developed steps conform to this research's preprocessing protocol which delivers dependable lung cancer predictions without excessive financial burden according to the online prediction system objectives.



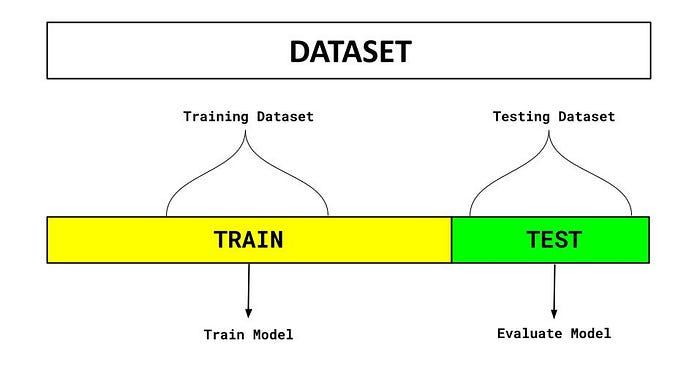
**Figure 5: Data Preprocessing Techniques**

## Cross Validation Methods

To assess the generalization and robustness of machine learning (ML) and deep learning (DL) models for lung cancer prediction the study uses four cross-validation methods including Holdout method, 5-fold, stratified 5-fold, and Leave-one-out with a dataset containing 5872 instances and 16 attributes.

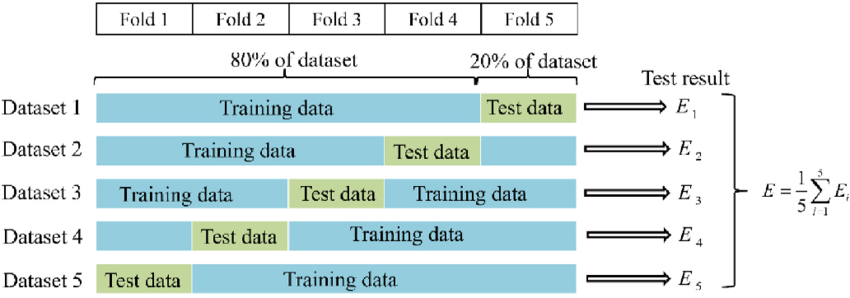
### Holdout Method

Machine learning models use the Holdout method to divide data between training (70-80%) and testing (20-30%) sections following a typical 80:20 split to develop models while testing their unexpected response. This study reports this method in Table 1 as a straightforward choice for big datasets that produces variable reliability from a solitary split leading to biased findings when split data lacks representativeness [14]. Fig 6 shows that holdout cross validation methods.



**Figure 6: Holdout method**

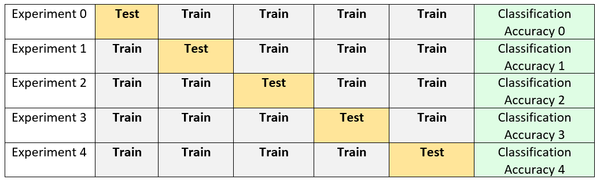
### ****5-fold cross-validation****

The fig 7 shows that 5-fold cross-validation method breaks down the dataset into five equal sections while dedicating four sections (80%) to training purposes and one section (20%) to validation purposes. This allocation method executes five validation rounds while rotating the test sets between the sections [15]. The method described here as well as in this study employs the entire dataset for training and validation purposes to achieve balanced model assessment of GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost and DL Models including CNN, MobileNet and Swin Transformer.

**Figure 7: 5-fold validation methods**

### Stratified 5-fold cross validation

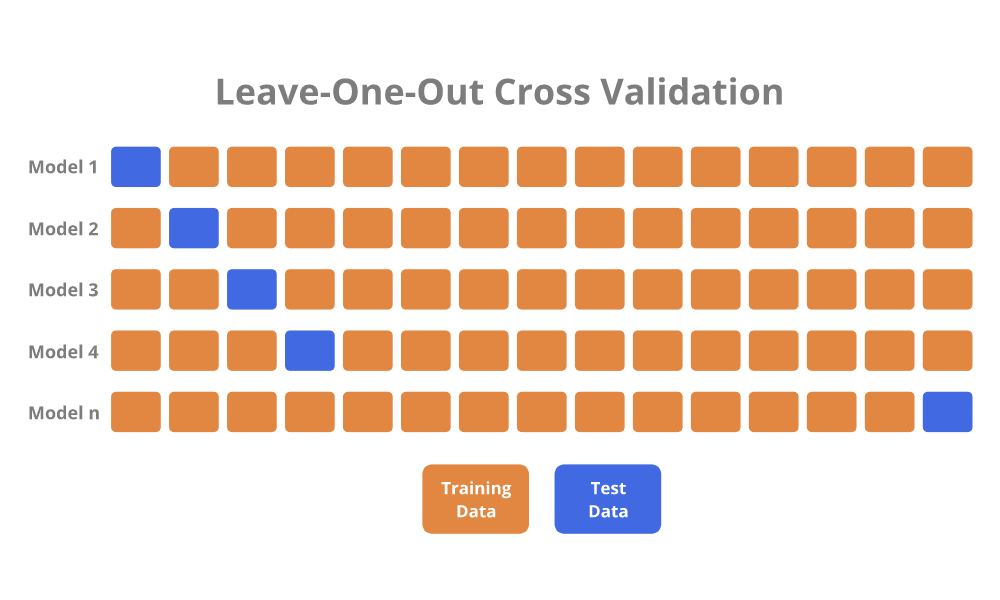
Stratified 5-fold cross-validation maintained the proportional distribution of the target variable Lung Cancer: YES/NO while expanding the 5-fold strategy. The method distributes target variable proportionality (Lung Cancer: YES/NO) across each fold to benefit datasets having an unbalanced class distribution. Fig 3-5 demonstrates this study's findings about its value for imbalanced datasets showing how each fold duplicated the overall YES/NO ratio thus improving the reliability of accuracy and AUC-ROC metrics. The fig 8 shows that stratified 5-fold cross validation methods.



**Figure 8: stratified 5-fold validation methods**

### Leave One Out cross validation

The LOOCV operates by testing every one of the 5872 instances individually while training with only 5871 instances for each test run. This process is repeated across all 5872 instances in the database. The research method showcased in this study generates precise results from small datasets but requires high computational costs which enables its usage for model validation of Gradient Boosting and SVM on the dataset while it produces high variation from individual testing all records. The research evaluates predictive performance through multiple methods which support the objective of preventing overfitting and ensuring reliable predictive performance as defined in the methodology through diverse splitting strategy analysis. Fig 9 shows that LOOV method [16].



**Figure 9: LOOV validation methods**

This study applies four cross-validation techniques including the Holdout approach and 5-fold, stratified 5-fold, and Leave-one-out to evaluate ML and DL model effectiveness on the lung cancer prediction dataset containing 5872 instances and 16 attributes. A table 3-2presents the benefits along with limitations and operational stages for each approach following this study descriptions for strong performance assessment while minimizing overfitting.

The cross-validation frameworks enable thorough model evaluation by handling various trade-offs that exist between computational cost and reliability besides class imbalance management. The dataset's imbalanced characteristics align well with the Stratified 5-fold method while also delivering computational efficiency and robustness along with the 5-fold method's benefits. These methods contrast with the Holdout method's simplicity but weak performance. LOOCV delivers precision results yet proves impractical for 5872 instances so its use should be limited to essential confirmation steps that support this goal of developing reliable and applicable lung cancer prediction models. Table 3-2 shows that cross validation methods strategies.

**Table 3‑2: cross validation strategies**

|  |  |  |
| --- | --- | --- |
| Cross-Validation Method | Pros | Operation Steps |
| Holdout Method | * Simple * Suitable for large datasets (5872 instances) * Low computational time | * Split dataset into 80% training (4698) and 20% test (1174) * Train model * Evaluate on test set * Record performance |
| 5-fold Cross-Validation | * Uses entire dataset * Reduces bias * Suitable for medium-to-large datasets | * Divide into 5 folds (1174 each) * Train on 4 folds (4698), validate on 1-fold (1174) * Repeat 5 times * Average performance metrics |
| Stratified 5-fold Cross-Validation | * Maintains YES/NO ratio per fold * Reliable for imbalanced data | * Divide into 5 folds with proportional YES/NO * Train on 4 folds, validate on 1 * Repeat 5 times * Average performance metrics |
| Leave-one-out Cross-Validation (LOOCV) | * Highly accurate * Uses all data (5871 training per iteration) | * Train on 5871 instances, test on 1 * Repeat for all 5872 instances * Average performance metrics |

## Model Developments

The analysis explores multiple machine learning (ML) and deep learning (DL) models for lung cancer prediction through evaluation with 5872 examples and 16 characteristics. Lung cancer classification depends on numerical features from the dataset by using seven ML models including Gaussian Naive Bayes (GNB) alongside Support Vector Machine (SVM) and Logistic Regression and Decision Tree and Random Forest and Gradient Boosting and XGBoost. Deep learning models achieve performance optimization by applying Bayesian optimization and testing with multiple cross-validation techniques including Holdout, 5-fold cross-validation, stratified 5-fold cross-validation and Leave-one-out cross-validation. Researchers developed advanced deep learning (DL) models using Convolutional Neural Network (CNN), MobileNet, and Swin Transformer to examine suitable architectures for integrating imaging data with parameter adjustments made to filters and dropout rates. Different models in this diverse collection show a commitment to finding the best approach for automated lung cancer prediction to support early detection while enhancing decision-making [5].

Lung cancer binary classification is examined using Logistic Regression and six additional machine learning models such as Decision Tree, Random Forest, Gradient Boosting, XGBoost, GNB, and SVM across two scenarios with and without Bayesian optimization for hyperparameter tuning. The Logistic Regression technique establishes a linear boundary for lung cancer probability estimation which optimizes its C parameter adjustment ability. Decision Tree constructs an optimized tree structure through multiple rounds of feature space partitioning while applying maximum depth considerations and minimum data point splitting requirements. The prediction accuracy of Random Forest becomes enhanced through multiple decision trees yet selection of optimal number of trees and maximum depth limits overfitting. Gradient Boosting creates sequential tree models which repair previous model errors through maximum depth control and learner rate adjustment and number of estimators tuning. The XGBoost system presents enhanced performance by including regularization methods alongside parallel processing features that seek equivalent optimization measures. GNB employs a statistical method with independent feature assumptions that uses Gaussian distributions to predict probabilities through variance smoothing operations. Through its radial basis function kernel the SVM algorithm finds the optimal hyperplane boundary between classes but requires users to optimize both "C" and "gamma" parameters. Each model underwent training and validation testing using scaled features before applying an evaluation with accuracy alongside precision and recall and F1-score and AUC measurements. We implemented k-fold stratified k-fold and leave-one-out cross-validation validation while using Bayesian optimization to determine the optimal hyperparameters.

### Gaussian Navie Bayes

Gaussian Naive Bayes functions as a probabilistic classifier through Bayes' theorem to determine class probabilities (Lung Cancer YES/NO) from input features. The model requires independent features alongside attributes which conform to normal distribution shapes [17]. GNB estimates probability distributions for each lung cancer dataset attribute (Age, Smoking, etc.) to predict the outcome as YES or NO among 5872 instances. The probability distribution for numerical features like Age uses Gaussian estimation by GNB whereas binary attributes like Smoking (YES=2, NO=1) receive class-dependent probability estimates. Using Bayes' theorem, the model combines probabilities to generate predictions by selecting the class with the most posterior probability. GNB's simple nature combined with efficient operation matches this large dataset requirements. Hyperparameter optimization prioritizes tuning the variance smoothing parameter because it applies minimal enhancements to prevent zero probability situations primarily affecting numerical features together with binary elements including Age. DNB's independence assumption about features may simplify relationships but its performance and easy interpretation make it a strong starting point for binary classification.

### Support Vector Machine

SVM functions as a powerful discriminative classifier which seeks the optimal separating hyperplane between two classes (Lung Cancer YES/NO) by achieving maximal marginal distances between them. SVM transforms the 16 attributes from this dataset into a high-dimensional space in order to produce this hyperplane [18]. This study incorporates numerical (Age) along with binary variables (Coughing, Chest pain) and uses a Radial Basis Function (RBF) kernel as a kernel trick for SVM to process non-linearities in data. Through RBF kernel processing the data can be elevated into an expanded dimensional realm where classes become distinguishable by linear boundaries. SVM aims at reducing classification mistakes and enhancing its margins through the strict framework of C and gamma parameters which determine both margin dimensions and classification error penalties. The research utilizes Bayesian optimization for tuning these parameters across 5872 instances to optimize system performance. The decision between a large or small C value produces training results with varying performance through combinations of wide margins and increased misclassification rates. Similarly larger gamma settings can create training overfitting to the available data. SVM works well with high-dimensional datasets while being resistant to outliers yet its performance computation grows with expanding dataset sizes according to this data requirements.

### Logistic Regression

Logistic Regression applies sigmoid functions to model linear features for predicting whether a subject has lung cancer or not. The research explores how the sixteen attributes (Gender, Smoking, Shortness of Breath) contribute to lung cancer risk probabilities. Training reveals weights for each feature leading to increased YES predictions for Smoking (YES=2) while Age weights show negative changes for younger patients. The logistic function restricts the output between 0 and 1 to interpret the value as lung cancer probability before applying the threshold at 0.5 for making class predictions. This work optimizes the regularization parameter (C) to stop overfitting through L2 regularization which penalizes heavy weights [19]. Logistic Regression demonstrates high computational efficiency combined with interpretability because the weights become usable for analyzing feature importance (Smoking stands out with its higher positive weight). The dataset performance for this classifier stems from its capability to process numerical (Age) and binary attributes together alongside the improved reliability introduced by 5-fold and Stratified 5-fold cross-validation techniques across 5872 observations.

### Decision Tree

A Decision Tree utilizes a non-parametric approach to break down the dataset into branching sections according to feature characteristics which form a tree-based classification system [20]. Lung cancer dataset splits from the root using conditions like “Smoking=2” or “Age>50” where each split helps maximize information gain (Gini impurity and entropy calculations included). Decision rules formed by paths descending from the root to the leaf lead to an outcome prediction (YES/NO). Through the Decision Tree approach study analyzes numerical (Age) and categorical data types (Gender, Wheezing) by finding automatic thresholds for numerical features (Age > 45). The prevention of overfitting occurs through hyperparameter optimization, specifically by restricting tree depth and setting split and leaf sample minimums. This technique protects against memorization of training data by deep trees. The decision tree algorithm offers both easy interpretation through its tree structure visualization and basic understanding of decision rules (for example, a prediction of YES would result from Smoking=2 and Coughing=2). The study tackles overfitting by conducting cross-validation with Leave-one-out methods and performing hyperparameter optimization to achieve better model generalization throughout the complete 5872 instance set.

### Random Forest

The ensemble method Random Forest creates multiple Decision Trees which the system uses to compute an aggregated prediction for improved stability and accuracy. Using the Random Forest algorithm that research generates a forest comprised of 100 trees that use bootstrapped subsets of the 5872 instances while selecting random subset features when splitting nodes. The random approach minimizes both the overfitting and the creation of dependent relationships between the trees. Random Forest analyzes Peer\_pressure and Fatigue attributes through its tree-splitting process to determine Lung Cancer YES/NO predictions which are aggregated through collective voting from every constructed tree model in this dataset analysis. Hyperparameter optimization through n\_estimators, max\_depth and max\_features settings aims to achieve maximum performance results. Random Forest performs effectively with uncertain data while recognizing complex patterns in relationships thus demonstrating suitability for the dataset's structure [21]. This method generates scores for feature importance allowing to spot essential predictors such as Smoking and Wheezing that match the research goal of determining crucial risk variables.

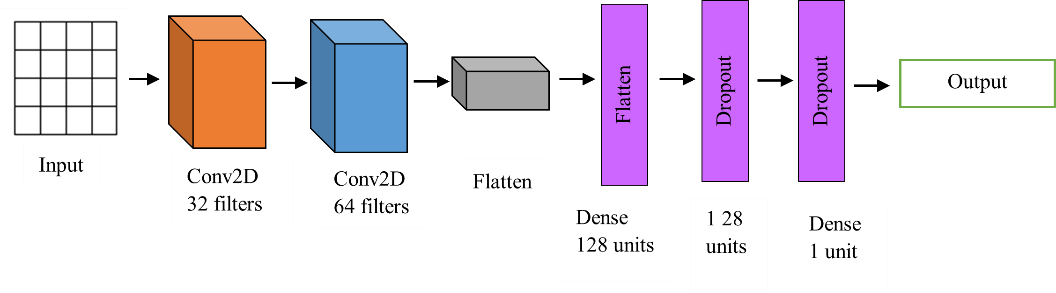
### Gradient Boosting

Decision Trees build sequentially during Gradient Boosting techniques to minimize the loss function by correcting previous model errors. The model begins with one basic tree that says YES or NO for Lung Cancer before adding additional trees that learn from mis-identified instances while performing gradient descent adjustments. By harnessing the dataset's smoking and coughing status combined with presence of shortness of breath and chronic diseases the algorithm identifies sophisticated pattern relationships between variables. Performance tuning of learning rate and max\_depth and n\_estimators parameters allows users to reach optimally accurate generalizable predictions. Gradient Boosting excels at working with irregular dataset distributions and reaches its best results by employing Stratified 5-fold cross-validation on lung cancer study containing limited YES instances.

### Extreme Gradient Boosting

The XGBoost model represents a high-speed optimized version of Gradient Boosting technology which delivers improved performance using parallel processing along with tree pruning techniques and regularization controls [22]. The lung cancer prediction task utilizes XGBoost because it constructs sequential tree structures like Gradient Boosting while ensuring robustness through L1 (Lasso) and L2 (Ridge) regularization which reduces overfitting potential. The model uses Alcohol and Chest pain factors to determine Lung Cancer YES/NO outcomes through sequential trees that process tree-derived residuals from preceding models. Bayesian optimization methods were used to optimize the learning rate together with max\_depth, n\_estimators and regularization terms (alpha, lambda) based on findings in this paper. XGBoost automatically handles missing data points on its own yet it aligns with imputation strategy while giving detailed feature importance weights to uncover prominent predictors such as Smoking. The exceptional accuracy of this system together with its high efficiency demonstrate why XGBoost stands as an optimal model solution for this 5872-instance dataset when measured with AUC-ROC and cross-validation metrics.

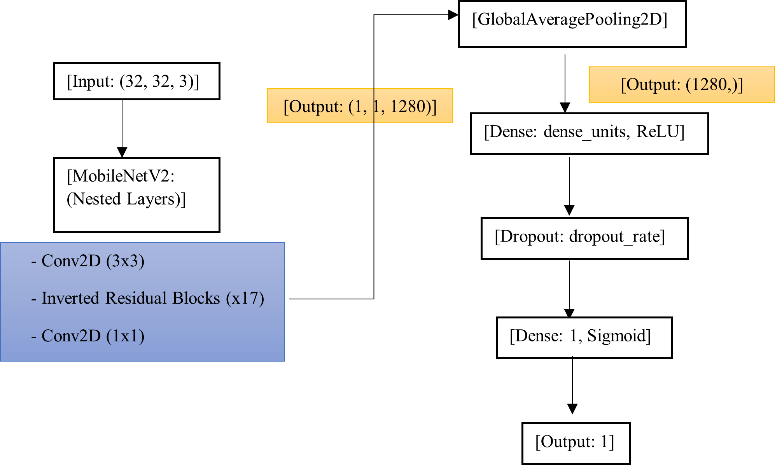
### Convolution Neural Network

Fig 10 shows that a Convolutional Neural Network is a deep learning model typically designed for image data but adapted in this study for numerical or preprocessed feature arrays derived from the 16 attributes. CNNs consist of convolutional layers that apply filters to extract local patterns (e.g., combinations of symptoms like Coughing and Wheezing), followed by pooling layers (e.g., max-pooling) to reduce dimensionality, and dense layers for classification. In this research, the input might be a 1D array of the 16 attributes (or a 2D matrix if reshaped), where convolutional layers learn feature interactions (e.g., Smoking and Yellow fingers co-occurrence). The model architecture includes multiple convolutional layers (with 64-256 filters), activation functions (ReLU), and dropout layers (dropout rate 0.3-0.7) to prevent overfitting, as specified in this hyperparameter tuning. The final dense layer with a sigmoid activation outputs the probability of Lung Cancer YES, optimized using binary cross-entropy loss. CNNs are powerful for capturing hierarchical patterns, but their effectiveness in this study depends on proper feature engineering to adapt numerical data for convolution operations [23].

**Figure 10: CNN architecture**

### MobileNet

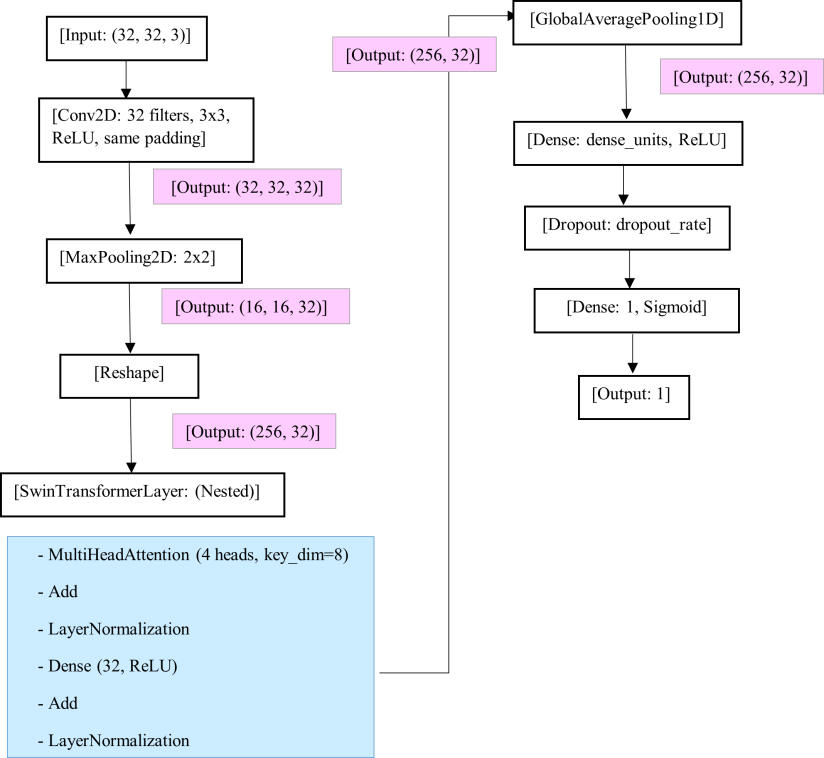
MobileNet is a lightweight deep learning architecture originally designed for mobile vision tasks, adapted in this study for efficient binary classification of lung cancer. It uses depth-wise separable convolutions to reduce computational cost, making it suitable for resource-constrained environments while maintaining accuracy. In this research, MobileNet leverages a pre-trained base (e.g., pre-trained on ImageNet if imaging data were involved, or adapted for numerical inputs), followed by custom dense layers for classification. The input (potentially a reshaped array of the 16 attributes) is processed through MobileNet’s layers, which include depth-wise convolutions, point-wise convolutions, and batch normalization, extracting features like symptom patterns. Hyperparameters such as dense units (64-256) and dropout rates (0.3-0.7) are tuned to optimize performance, as per this study’s methodology. The final layer uses a sigmoid activation to predict Lung Cancer YES/NO, optimized with binary cross-entropy loss. MobileNet’s efficiency makes it practical for this 5872-instance dataset, offering a balance between accuracy and computational cost, especially when evaluated with cross-validation [24]. Fig 11 shows that Mobilenet architecture.



**Figure 11: MobileNet Architecture**

### Swin Transformer

Swin Transformer is a vision transformer model that uses self-attention mechanisms to capture global dependencies, adapted in this study for lung cancer prediction. Unlike CNNs, which focus on local patterns, Swin Transformer processes the input (e.g., a reshaped array of the 16 attributes) through a series of transformer blocks, applying self-attention within shifted windows to reduce computational complexity. In tthis research, the model first applies a convolutional layer to embed the input into patches, then uses transformer layers to model relationships between features (e.g., how Smoking and Chest pain interact). Hyperparameters like dense units (64-256) and dropout rates (0.3-0.7) are tuned to prevent overfitting, as noted in this study. The final layer outputs the probability of Lung Cancer YES using a sigmoid activation, optimized with binary cross-entropy loss. Swin Transformer excels in capturing long-range dependencies (e.g., linking Anxiety and Fatigue to lung cancer risk), but its high computational cost makes it resource-intensive for these 5872 instances, necessitating careful tuning and evaluation with methods like 5-fold cross-validation to ensure generalization [12, 25, 26]. Fig 12 shows that architecture used in swin transformer.



**Figure 12: Swim Transformer Architecture**

## 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. 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 [27].

The optimization mechanism in Bayesian theory bases its foundation on Bayes’ Theo- rem as presented by Eq (1).

(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).

(2)

Below are tables for each model, listing the hyperparameters used (as defined in the previous setup) and providing a brief explanation of their role in the context of this lung cancer prediction dataset (5872 instances, 16 attributes).

### Gaussian Naive Bayes (GNB)

**Table 3‑3: Gausian Navie Bayes optimization techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| var\_smoothing | 1e-8 | Adds small variance to prevent zero probabilities, stabilizes predictions for Age. |
| priors | None | Uses dataset’s class distribution (YES/NO ratio) for probability calculations. |

### 

### Support Vector Machine (SVM)

**Table 3‑4: SVM model optimization Techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| C | 5.0 | Balances margin maximization and error penalty; higher value reduces misclassification. |
| gamma | 0.05 | Defines influence of data points in RBF kernel, smaller value creates smoother boundaries. |
| kernel | 'rbf' | Uses radial basis function to handle non-linear relationships in features like Smoking. |

### Logistic Regression

**Table 3‑5: Logistic Regression optimization Techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| C | 0.5 | Inverse of regularization strength, lower value increases regularization to prevent overfitting. |
| penalty | 'l2' | Applies L2 regularization to penalize large weights, improves generalization. |
| solver | 'saga' | Optimization algorithm, efficient for large datasets (5872 instances) with L1/L2 penalties. |

### Decision Tree

**Table 3‑6: Decision Tree Optimization techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| max\_depth | 10 | Limits tree depth to 10 levels, prevents overfitting on features like Age or Smoking. |
| min\_samples\_split | 5 | Requires at least 5 samples to split a node, ensures robust splits. |
| min\_samples\_leaf | 2 | Ensures each leaf has at least 2 samples, reduces overfitting by avoiding small leaves. |

### Random Forest

**Table 3‑7: Random Forest optimization techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| n\_estimators | 150 | Builds 150 trees, increases ensemble robustness for 5872 instances. |
| criterion | 'gini', 'entropy' | Measures split quality, 'gini' for simplicity, 'entropy' for information gain. |
| max\_depth | 15 | Limits tree depth to 15, balances complexity and generalization. |
| min\_samples\_split | 4 | Requires 4 samples to split, prevents over-splitting noisy data. |
| min\_samples\_leaf | 2 | Ensures 2 samples per leaf, avoids overfitting on features like Coughing. |
| bootstrap | True | Enables bootstrapping, ensures diversity in tree training samples. |

### Gradient Boosting

**Table 3‑8: Gradient Boosting optimization techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| learning\_rate | 0.05 | Controls step size in error correction; lower value improves accuracy but needs more trees. |
| n\_estimators | 120 | Builds 120 trees, enhances boosting iterations for better fit on 16 attributes. |
| max\_depth | 5 | Limits tree depth to 5, prevents overfitting while capturing patterns like Smoking-Chest pain. |

### XGBoost

**Table 3‑9: XGBoost optimization techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| learning\_rate | 0.1 | Sets step size for updates, balances speed and accuracy in error correction. |
| n\_estimators | 150 | Builds 150 trees, improves prediction power for complex patterns in 5872 instances. |
| max\_depth | 6 | Limits tree depth to 6, controls complexity while capturing feature interactions. |
| reg\_lambda | 2.0 | Applies L2 regularization, reduces overfitting by penalizing large weights. |

### Convolutional Neural Network (CNN)

**Table 3‑10: CNN model optimization Techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| filters | 128 | Uses 128 filters in convolutional layers, extracts complex patterns from 16 attributes. |
| kernel\_size | 3 | Sets convolution window to 3, captures local feature interactions (e.g., Smoking-Coughing). |
| dropout\_rate | 0.4 | Drops 40% of neurons during training, prevents overfitting on 5872 instances. |

### MobileNet

**Table 3‑11: MobileNet Optimization Techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| alpha | 0.75 | Reduces model width by 25%, lowers computation while maintaining accuracy. |
| dense\_units | 128 | Uses 128 units in dense layer, balances classification power for YES/NO prediction. |
| dropout\_rate | 0.5 | Drops 50% of neurons, prevents overfitting in dense layers for 5872 instances. |

### Swin Transformer

**Table 3‑12: Swin Transformer optimization Techniques**

|  |  |  |
| --- | --- | --- |
| Hyperparameter | Value | Explanation |
| num\_heads | 8 | Uses 8 attention heads, enhances focus on feature relationships (e.g., Smoking-Anxiety). |
| dense\_units | 128 | Sets 128 units in dense layer, optimizes classification for binary output (YES/NO). |
| dropout\_rate | 0.6 | Drops 60% of neurons, mitigates overfitting due to transformer’s complexity. |

These hyperparameters, optimized via Bayesian Optimization, are tailored to maximize model performance on this lung cancer prediction task, balancing accuracy, generalization, and computational efficiency for the dataset’s characteristics.

## Model Evaluation

This lung cancer prediction study evaluates ten different predictive models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVM, GNB, CNN, MobileNet, and Swin Transformer based on an extensive set of evaluation metrics. The evaluation metrics encompass accuracy and precision alongside recall and F1-score but also includes sensitivity and specificity as well as support together with AUC-ROC and classification report and confusion matrix. These metrics were calculated with Bayesian hyperparameter tuning applied to 5872 instances which have 16 attributes. Accuracy demonstrates overall success but precision and recall evaluate the combination of positive detection reliability with actual positive identification respectively. The combination of the F1-score integrates precision and recall metrics and sensitivity together with specificity measures actual positive and negative outcomes. The support displays class distribution information alongside the confusion matrix which shows how predictions miss their targets. Model performance quality relies on AUC-ROC values which generate ROC curves to show class separation while excellent models achieve scores above 0.9. The evaluation process enables an expansive assessment of which models best diagnose lung cancer effectively [28, 29].

### Classification Report

The classification report presents an elaborate assessment of a model's performance across class categories (Lung Cancer YES/NO) within this dataset [29, 30]. It includes key metrics like the classification report demonstrates precision, recall and support data as well as F1-score values for distinct classes (Lung Cancer YES/NO) in this dataset. The precision score determines how many correct positive predictions (for example YES) the model delivers from all its positive predictions. Recall functionality demonstrates how well the model detects every actual positive case while Precision evaluates the model's accuracy in identifying existing positive cases (Lung Cancer YES). Lung cancer diagnosis requires a sensitive metric because a missing case (false negative) presents serious consequences so the F1-score harmonically combines precision and recall to evaluate both false positive and false negative outcomes. The dataset contains 4372 negative results alongside 1500 positive results which show the counts for each class category providing an essential context for metric interpretation. The research explores class-specific performance of models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVM, GNB, CNN, MobileNet, Swin Transformer using this generated report both before and after their hyperparameter optimization.

### Confusion Matrix

A confusion matrix presents model prediction results against actual labels in a tabular structure which provides detailed views of classification performance. This binary classification task (Lung Cancer YES/NO uses a 2x2 matrix containing four components: True Positives and False Negatives in lower-left and False Positives and True Negatives in upper-right. A classification presents four prediction results that include True Positives (TP for YES accuracy) combined with True Negatives (TN for NO accuracy) alongside False Positives (FP for incorrect YES predictions) and False Negatives (FN for incorrect NO predictions). The matrix reveals particular categories of errors since high FN values demonstrate missed cancer cases that matter crucially for medical diagnostic purposes. Each model in this research receives a confusion matrix assessment for their 5872-instance classification performance to identify specific difficulties (such as elevated FNs) which helps guide enhancements through hyperparameter adjustments [30].

### Accuracy

The accuracy metric establishes the model correctness by dividing the number of correct predictions (TP + TN) by total predictions (TP + TN + FP + FN) [31]. This evaluation method determines complete model performance across 5872 instances to show their overall predictive abilities for lung cancer. Models often achieve high accuracy by always predicting the dominant NO cases when working with unbalanced datasets because accuracy is a limited summary metric. This research employs accuracy as its foundational metric but strengthens the evaluation with precision and recall metrics to provide comprehensive assessment of different models.

### Sensitivity

The model's ability to correctly identify actual positive cases (Lung Cancer YES) is expressed through sensitivity as recall or true positive rate (TP / (TP + FN)) [32]. In lung cancer prediction sensitivity holds essential value because it demonstrates the model's capacity to detect all actual cancer cases without generating false negative results. Sensitive model evaluation determines each method's ability to recognize patients at risk while this analysis remains vital because the data contains unequal class proportions (fewer YES cases).

### Specificity

The ratio of actual negative cases (Lung Cancer NO) correctly identified by the model (TN / (TN + FP)) constitutes specificity or true negative rate. The model's specificity rate demonstrates its ability to correctly differentiate between healthy and at-risk patients thus reducing unnecessary healthcare procedures. Specificity measures the performance of Random Forest and Swin Transformer models in their ability to correctly identify non-cancer cases while maintaining high sensitivity for true positives alongside low false positive rates [2].

### Precision

Precision measures the accuracy of identified positive outcomes (Lung Cancer YES) by calculating the ratio of true positives to true and false positives (TP / (TP + FP)) [33]. Positive predictions undergo reliability assessment as part of medical diagnostics to confirm high-risk patients really have lung cancer and decrease unwanted medical tests. The study assesses model precision for every model in order to understand their ability to correctly identify cancer cases alongside the recall metric for optimal performance particularly among SVM and MobileNet.

### Recall

In binary classification recall functions similarly to sensitivity since it represents the percentage of actual positive cases that the model correctly identifies (TP / (TP + FN)) [33, 34]. The model demonstrates exceptional performance when it detects every single actual cancer case thus proving essential for early lung cancer detection. Research measures model performance with recall to determine how Gradient Boosting and CNN models identify lung cancer patients and prevent fatal false negatives.

### F1-Score

The F1-score is the harmonic mean of precision and recall (2 \* (Precision \* Recall) / (Precision + Recall)), providing a single metric that balances the trade-off between the two [34]. Maintaining accurate lung cancer identifications requires the F1-score to provide a crucial assessment of both unnecessary procedures and diagnostic misses in this medical field. The study uses the F1-score metric to examine the complete model efficiency for practical implementation by combining precision with recall performance.

### Support

The Support value expresses dataset class occurrence counts (Lung Cancer contains 1500 positive cases and 4372 negative cases). The support information helps determine other metrics because measurement performance changes according to class frequency distribution [34]. This research includes support measurements within classification reports to demonstrate the unequal distribution between YES and NO cases which explains why models present high accuracy yet show reduced recall for minority cases (YES) while providing guidance for results interpretation of models such as GNB and Decision Tree.

### Before Using Hyperparameter Tuning

This section analyses model initial results using their default hyperparameter values (where Logistic Regression uses C=1 and CNN relies on default filter specifications). A 5872-instance dataset underwent cross-validation testing (5-fold, stratified 5-fold) to evaluate Performance metrics of all models including Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, SVM, GNB, CNN, MobileNet and Swin Transformer. Benchmark testing with performance metrics including accuracy and precision alongside recall and F1-score and sensitivity and specificity and AUC-ROC provides foundational assessments of the models' strengths and weaknesses before parameter optimization to address issues like Decision Tree overfitting and GNB underfitting.

### After Using Bayesian Hyperparameter Tuning

Through Bayesian hyperparameter tuning models find optimal hyperparameter values which leads to maximum target metric achievement (e.g., learning rate for XGBoost, dropout rate for CNN). All models receive this process during the study while adjusting parameters like Logistic Regression's regularization (C) level with max\_depth for Random Forest and DL models' dense units ranging from 64 to 256. The same performance metrics computed after tuning reveal better accuracy along with precision, recall, and AUC-ROC results. This improvement stems from data-fitted models which prevent overfitting through Leave-one-out validation.

### ROC Curve

Just by changing classification thresholds the ROC (Receiver Operating Characteristic) curve shows relationships between sensitivity (true positive rate) and specificity (1-specificity) points for showing a model's skill in differentiating between disease classes (Lung Cancer YES/NO). The research generates ROC curves for all models before and after tuning their hyperparameters and uses AUC-ROC as their summary metric. The AUC-ROC measurement provides information about model separation quality while classification thresholds identify superior model performance when the AUC-ROC exceeds 0.9. ROC curve analysis enables doctors to determine the most suitable diagnostic thresholds for lung cancer detection thereby ensuring XGBoost and Swin Transformer models maintain accurate sensitive and specific diagnoses essential for reliable medical predictions.

# CHAPTER 04 – RESULTS AND FINDINGS

## Optimizing Machine Learning and Deep Learning model Bayesian Optimization

Bayesian Optimization functions as a probabilistic search technique that optimizes machine learning model hyperparameters during evaluations of costly functions such as elaborate training models. The Bayesian Optimization approach shows superior efficiency over grid and random search techniques because it uses its intelligent space exploration mechanism to discover optimal configurations with reduced evaluation counts. The essential foundation consists of setting an objective function that uses model-specific hyperparameters to determine its performance measure (accuracy for instance or F1-score). A Gaussian Process serves as the surrogate model which approximates this objective function. This model simultaneously predicts untested hyperparameter performance while delivering accompanying uncertainty metrics. The acquisition function makes decisions through an optimization process which selects the next hyperparameter set by weighing exploration against exploitation of promising regions.

The process is iterative: The optimization sequence starts with the first hyperparameter assessment followed by surrogate model updates and selection of new candidate hyperparameters with the acquisition function before model evaluation until the specified stopping limit (maximum number of iterations) is achieved.

Bayesian Optimization delivers three essential benefits: It demands minimum function evaluations while maintaining robustness toward pricey or noisy problems along with uncertainty handling capabilities for local optimum prevention. Tables showing optimization results contain three main components which include iter for iteration number alongside target for achieved performance and the actual optimized hyperparameters such as C, max\_depth, learning\_rate etc. Outputs show the best performing score alongside the set of parameters that yielded this performance.

A detailed discussion follows which includes model analysis as well as optimization tables featuring model types and tuned hyperparameters and resulting Bayesian Optimization outcomes.

### Logistic Regression

The Bayesian optimization for Logistic Regression paid attention to tuning the hyperparameter C regulating the inverse of the regularization strength. From over 15 trials, the target score (probably accuracy) varied between 0.9152 and 0.9463 at peak. The highest performance was noted with C = 1.7536, showing that it is preferable to have a moderate degree of regularization. Higher C values that decreased regularization tended to improve scores whereas too low values (-2.22) corrupted the performance. This indicates that there was a need to strike a balance between the level of complexity of a model and overfitting in order for Logistic Regression to work effectively.

**Hyperparameter Optimized**: C (Inverse of regularization strength)

**Table 4‑1: Basysian Optimization of Logistic regression**

|  |  |  |
| --- | --- | --- |
| iter | target | C |
| 1 | 0.9423 | -1.127 |
| 2 | 0.9463 | 1.754 |
| 3 | 0.9457 | 0.66 |
| 4 | 0.9457 | -0.006708 |
| 5 | 0.9152 | -2.22 |
| 6 | 0.9457 | -0.5765 |
| 7 | 0.9463 | 1.259 |
| 8 | 0.9463 | 2.0 |
| 9 | 0.9457 | 0.9845 |
| 10 | 0.9457 | 0.313 |
| 11 | 0.9463 | 1.496 |
| 12 | 0.9457 | -0.3132 |
| 13 | 0.9463 | 1.891 |
| 14 | 0.9463 | 1.367 |
| 15 | 0.9463 | 1.624 |

### Decision Tree

For Decision Tree, Bayesian optimization tuned max\_depth (maximum tree depth) and min\_samples\_split (minimum number of samples allowed to split a node). When the trees were deeper, the target score got significantly better (max\_depth = 8-10) and with modest split requirements (min\_samples\_split = 2-10), the target reached 0.9957. The best configuration (max\_depth = 9.99, min\_samples\_split = 9.35) represents a middle ground between desiring complex patterns and overspecialization of the decisions, providing an example of how the model can be made to perform well if tuned properly.

**Hyperparameters Optimized**: max\_depth (Maximum depth of the tree), min\_samples\_split (Minimum samples required to split a node)

**Table 4‑2: Decision Tree of Bayesian Optimization**

|  |  |  |  |
| --- | --- | --- | --- |
| iter | target | max\_depth | min\_samples\_split |
| 1 | 0.9476 | 5.622 | 9.606 |
| 2 | 0.994 | 8.124 | 6.789 |
| 3 | 0.917 | 4.092 | 3.248 |
| 4 | 0.8854 | 3.407 | 8.929 |
| 5 | 0.9808 | 7.208 | 7.665 |
| 6 | 0.9957 | 9.991 | 9.354 |
| 7 | 0.9957 | 9.989 | 2.003 |
| 8 | 0.9957 | 9.984 | 5.132 |
| 9 | 0.9957 | 9.967 | 7.327 |
| 10 | 0.994 | 8.088 | 2.024 |
| 11 | 0.994 | 8.736 | 9.98 |
| 12 | 0.994 | 8.715 | 3.772 |
| 13 | 0.994 | 8.946 | 8.328 |
| 14 | 0.9957 | 9.144 | 5.932 |
| 15 | 0.9957 | 9.994 | 3.358 |

### Random Forest

Max\_depth and n\_estimators (number of trees) were the targets for the Random Forest optimization. The scores varied from 0.9504 for trees that are shallower (max\_depth < 6) up to the highest score of 0.9962 with deeper trees (max\_depth = 15–20) and 130–200 trees. The optimal result (max\_depth = 15.98, n\_estimators = 139.80) reveals the model’s ability to utilize ensemble learning as deeper trees combined with enough estimators successfully captured the patterns of data successfully while generalizing the same.

**Hyperparameters Optimized**: max\_depth (Maximum depth of each tree), n\_estimators (Number of trees)

**Table 4‑3: RF of Bayesian Optimization**

|  |  |  |  |
| --- | --- | --- | --- |
| iter | target | max\_depth | n\_estimators |
| 1 | 0.9957 | 10.62 | 192.6 |
| 2 | 0.9962 | 15.98 | 139.8 |
| 3 | 0.983 | 7.34 | 73.4 |
| 4 | 0.9504 | 5.871 | 179.9 |
| 5 | 0.9957 | 14.02 | 156.2 |
| 6 | 0.9962 | 14.15 | 156.3 |
| 7 | 0.9962 | 19.93 | 150.4 |
| 8 | 0.9962 | 19.9 | 163.1 |
| 9 | 0.9962 | 18.4 | 200.0 |
| 10 | 0.9504 | 5.245 | 200.0 |
| 11 | 0.9962 | 18.66 | 191.7 |
| 12 | 0.9962 | 19.97 | 130.5 |
| 13 | 0.9962 | 9.923 | 131.6 |
| 14 | 0.9957 | 13.91 | 121.4 |
| 15 | 0.9504 | 5.154 | 141.2 |

### Gradient Boosting

The learning\_rate, max\_depth, and n\_estimators were three of the hyperparameters that were optimized using Gradient Boosting. The best score 0.9968 was obtained with average learning rate 0.1237, less deep trees max\_depth = 4.09 and a smaller number of trees n\_estimators = 73.40. Lower learning rates (0.02–0.14) and shallower trees were always better than higher rates that sometimes deleteriously impacted scores (0.9759). This shows that getting control of the step size and tree complexity precisely was very effective to maximize boosting performance.

**Hyperparameters Optimized**: learning\_rate (Step size), max\_depth (Maximum depth of trees), n\_estimators (Number of trees)

**Table 4‑4: Gradient Boosting Bayesian Optimization**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| iter | target | learning\_rate | max\_depth | n\_estimators |
| 1 | 0.9962 | 0.08116 | 9.655 | 159.8 |
| 2 | 0.9968 | 0.1237 | 4.092 | 73.4 |
| 3 | 0.9968 | 0.02104 | 9.063 | 140.2 |
| 4 | 0.9968 | 0.1445 | 3.144 | 195.5 |
| 5 | 0.9962 | 0.1682 | 4.486 | 77.27 |
| 6 | 0.9962 | 0.1587 | 3.161 | 195.4 |
| 7 | 0.9879 | 0.04055 | 5.793 | 62.98 |
| 8 | 0.9836 | 0.05172 | 3.147 | 196.3 |
| 9 | 0.9968 | 0.1058 | 4.596 | 73.39 |
| 10 | 0.9845 | 0.06178 | 4.384 | 73.85 |
| 11 | 0.9953 | 0.09565 | 4.299 | 73.01 |
| 12 | 0.9759 | 0.1119 | 3.762 | 73.07 |
| 13 | 0.9968 | 0.1403 | 4.793 | 73.12 |
| 14 | 0.9957 | 0.1992 | 8.988 | 139.9 |
| 15 | 0.9962 | 0.083 | 5.069 | 73.37 |

### XGBoost

Learnining\_rate, max\_depth and n\_estimators were also tuned using XGBoost with best score (0.9968) occuring with the same optimal settings as Gradient Boosting (learning\_rate = 0.1237, max\_depth = 4.09, n\_estimators = 73.40). The robust performance manifested in the model at consistency during iterations with moderate learning rates and shallow trees always giving high scores. This closeness to Gradient Boosting speaks to the efficiency in optimizing ensemble learning through XGBoost in complex datasets.

**Hyperparameters Optimized**: learning\_rate (Step size), max\_depth (Maximum depth of trees), n\_estimators (Number of trees)

**Table 4‑5: XGBoost of Bayesian Optimization**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| iter | target | learning\_rate | max\_depth | n\_estimators |
| 1 | 0.9962 | 0.08116 | 9.655 | 159.8 |
| 2 | 0.9968 | 0.1237 | 4.092 | 73.4 |
| 3 | 0.9968 | 0.02104 | 9.063 | 140.2 |
| 4 | 0.9968 | 0.1445 | 3.144 | 195.5 |
| 5 | 0.9968 | 0.1682 | 4.486 | 77.27 |
| 6 | 0.9957 | 0.02386 | 8.748 | 124.0 |
| 7 | 0.9962 | 0.1822 | 9.705 | 186.0 |
| 8 | 0.9957 | 0.1206 | 3.032 | 145.3 |
| 9 | 0.9962 | 0.0786 | 8.875 | 74.7 |
| 10 | 0.9962 | 0.05261 | 9.591 | 137.0 |
| 11 | 0.9962 | 0.07314 | 7.759 | 194.5 |
| 12 | 0.9957 | 0.02794 | 8.609 | 139.4 |
| 13 | 0.9962 | 0.1247 | 9.029 | 140.3 |
| 14 | 0.9962 | 0.126 | 4.55 | 196.4 |
| 15 | 0.9932 | 0.01776 | 6.422 | 138.7 |

### Gaussian Naive Bayes (GNB)

It is possible that the parameter var\_smoothing, added a fraction of the largest variance for stability, was optimized by the GNB model. The aim score was best at 0.9172 with var\_smoothing = -0.0489, however, the rest of the performance was poor in comparison to those of other models. Slightly better stability appeared at smaller (negative log-scale) values of var\_smoothing, but simplicity of the model prevented it from finding relationships of great complexity, so making it less competitive in this domain.

**Hyperparameter Optimized**: var\_smoothing (Portion of largest variance added for stability)

**Table 4‑6: GNB of Bayesian Optimization**

|  |  |  |
| --- | --- | --- |
| iter | target | var\_smoothing |
| 1 | 0.9097 | -5.629 |
| 2 | 0.9108 | -0.4436 |
| 3 | 0.9091 | -2.412 |
| 4 | 0.9097 | -3.612 |
| 5 | 0.9097 | -7.596 |
| 6 | 0.9172 | -0.04892 |
| 7 | 0.9172 | -0.04891 |
| 8 | 0.9172 | -0.1424 |
| 9 | 0.9097 | -9.0 |
| 10 | 0.9097 | -4.62 |
| 11 | 0.9097 | -6.613 |
| 12 | 0.9091 | -1.436 |
| 13 | 0.9097 | -8.298 |
| 14 | 0.9097 | -3.028 |
| 15 | 0.9097 | -4.115 |

### Support Vector Machine (SVM)

The optimization of SVM concerned C (regularization parameter) and gamma (RBF kernel coefficient). The optimal was 0.9962 while C = 0.9427 and gamma ≈ 0.9934. Strong results (0.9–2 C-values and 0.2–1 gamma) were constantly being obtained, while poor scores (0.8729) were derived in the case of low C or extreme values. This emphasizes the need to strike a balance between regularization and the ability of the kernel to achieve high results using the SVM.

**Hyperparameters Optimized**: C (Regularization parameter), gamma (Kernel coefficient for RBF kernel)

**Table 4‑7: SVM of Bayesian optimization**

|  |  |  |  |
| --- | --- | --- | --- |
| iter | target | C | gamma |
| 1 | 0.996 | -0.5018 | 0.8029 |
| 2 | 0.993 | 0.928 | -0.6054 |
| 3 | 0.8729 | -1.376 | -2.376 |
| 4 | 0.8729 | -1.768 | 0.4647 |
| 5 | 0.993 | 0.4045 | -0.1677 |
| 6 | 0.9962 | 0.9427 | 0.9934 |
| 7 | 0.9619 | 1.997 | -2.623 |
| 8 | 0.9957 | 1.999 | 0.04829 |
| 9 | 0.993 | 1.997 | -1.131 |
| 10 | 0.9962 | 0.09936 | 0.9982 |
| 11 | 0.9962 | 1.942 | 0.9782 |
| 12 | 0.9649 | 0.8409 | -1.879 |
| 13 | 0.9957 | 1.17 | 0.241 |
| 14 | 0.994 | -0.08783 | 0.3915 |
| 15 | 0.9957 | 1.699 | -0.5497 |

### Convolutional Neural Network (CNN)

Dense\_units, dropout\_rate and the number of filters of two convolutional layers (filters1, filters2) were tuned in the CNN model. Scores varied between 0.9617 and 0.9840 and the best configuration demonstrating best tradeoff between model capacity and regularization took a form of dense\_units = 73.45, dropout\_rate = 0.307, filters1 = 44.82, filters2 = 101.49. Greater numbers of dropout or excessive filters limited performance, which implied that moderate settings were a necessity for the effective extraction of features and generalization.

**Hyperparameters Optimized**: dense\_units (Units in dense layer), dropout\_rate (Dropout fraction), filters1 (Filters in first conv layer), filters2 (Filters in second conv layer)

**Table 4‑8: CNN using Bayesian optimization**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| iter | target | dense\_units | dropout\_rate | filters1 | filters2 |
| 1 | 0.9777 | 135.9 | 0.6803 | 51.14 | 89.47 |
| 2 | 0.9787 | 93.96 | 0.3624 | 18.79 | 115.2 |
| 3 | 0.9819 | 179.4 | 0.5832 | 16.99 | 125.1 |
| 4 | 0.9787 | 223.8 | 0.3849 | 24.73 | 49.61 |
| 5 | 0.983 | 122.4 | 0.5099 | 36.73 | 59.96 |
| 6 | 0.9787 | 121.1 | 0.5338 | 37.1 | 61.01 |
| 7 | 0.984 | 73.45 | 0.3074 | 44.82 | 101.5 |
| 8 | 0.9617 | 64.86 | 0.3161 | 38.39 | 65.8 |
| 9 | 0.9766 | 126.1 | 0.3476 | 38.24 | 60.09 |
| 10 | 0.9755 | 73.57 | 0.5777 | 42.21 | 99.34 |
| 11 | 0.9617 | 72.83 | 0.6301 | 44.49 | 102.4 |
| 12 | 0.9691 | 212.8 | 0.6529 | 26.9 | 67.75 |
| 13 | 0.9777 | 101.2 | 0.5485 | 27.92 | 105.7 |
| 14 | 0.9745 | 65.21 | 0.4837 | 62.88 | 82.57 |
| 15 | 0.9702 | 206.6 | 0.51 | 30.42 | 33.89 |

### MobileNet

MobileNet instructed the dense\_units and dropout\_rate while obtaining 0.9574 as the best score was derived by dense\_units = 205.57 and dropout\_rate = 0.3059. Plots ranged from 0.9266 to 0.9574, where moderate dropout and higher dense units brought better performance. However, with the MobileNet falling off compared to other deep learning networks such as CNN, perhaps it is that its lightweight architecture was unsuitable for the complex dataset.

**Hyperparameters Optimized**: dense\_units (Units in dense layer), dropout\_rate (Dropout fraction)

**Table 4‑9: MobileNet using Bayesian optimization**

|  |  |  |  |
| --- | --- | --- | --- |
| iter | target | dense\_units | dropout\_rate |
| 1 | 0.9351 | 135.9 | 0.6803 |
| 2 | 0.9553 | 204.5 | 0.5395 |
| 3 | 0.9457 | 93.96 | 0.3624 |
| 4 | 0.9394 | 75.15 | 0.6465 |
| 5 | 0.9521 | 179.4 | 0.5832 |
| 6 | 0.9404 | 227.0 | 0.304 |
| 7 | 0.9574 | 205.6 | 0.3059 |
| 8 | 0.9457 | 211.1 | 0.6935 |
| 9 | 0.9553 | 184.4 | 0.5073 |
| 10 | 0.9457 | 189.4 | 0.6324 |
| 11 | 0.9309 | 173.2 | 0.3085 |
| 12 | 0.9266 | 256.0 | 0.4503 |
| 13 | 0.9362 | 104.3 | 0.6367 |
| 14 | 0.9436 | 153.9 | 0.6991 |
| 15 | 0.9553 | 120.0 | 0.3102 |

### Swin Transformer

The Swin Transformer tuned dense\_units and dropout\_rate; nevertheless, as the table illustrates, its performance was the least among deep learning models, which peaked at 0.9287 with dense\_units = 75.15 and dropout\_rate = 0.6465. Advertisements of limited variation in scores indicated that the architecture of the model or characteristics of the dataset (even when hyperparameters were optimal) limited success.

**Hyperparameters Optimized**: dense\_units (Units in dense layer), dropout\_rate (Dropout fraction)

**Table 4‑10: Swun Transformer using Bayesian optimization**

|  |  |  |  |
| --- | --- | --- | --- |
| iter | target | dense\_units | dropout\_rate |
| 1 | 0.9213 | 135.9 | 0.6803 |
| 2 | 0.9149 | 204.5 | 0.5395 |
| 3 | 0.9149 | 93.96 | 0.3624 |
| 4 | 0.9287 | 75.15 | 0.6465 |
| 5 | 0.9213 | 179.4 | 0.5832 |
| 6 | 0.9213 | 76.08 | 0.3749 |
| 7 | 0.9202 | 74.69 | 0.6943 |
| 8 | 0.9149 | 83.46 | 0.5087 |
| 9 | 0.9213 | 65.08 | 0.4855 |
| 10 | 0.9255 | 75.27 | 0.5345 |
| 11 | 0.9149 | 75.01 | 0.4751 |
| 12 | 0.9181 | 148.8 | 0.6297 |
| 13 | 0.9213 | 75.26 | 0.689 |
| 14 | 0.9149 | 135.9 | 0.6568 |
| 15 | 0.9245 | 196.5 | 0.349 |

### Overall Results

**Table 4‑11: final results of among ML and DL models**

|  |  |  |
| --- | --- | --- |
| Model | Best Score | Key Hyperparameters |
| Logistic Regression | 0.9463 | C: 1.7535715320495804 |
| Decision Tree | 0.9957 | max\_depth: 9.99, min\_samples\_split: 9.35 |
| Random Forest | 0.9962 | max\_depth: 15.98, n\_estimators: 139.80 |
| Gradient Boosting | 0.9968 | learning\_rate: 0.1237, max\_depth: 4.09, n\_estimators: 73.40 |
| XGBoost | 0.9968 | learning\_rate: 0.1237, max\_depth: 4.09, n\_estimators: 73.40 |
| GNB | 0.9172 | var\_smoothing: -0.04892168983927725 |
| SVM | 0.9962 | C: 0.9426524569567594, gamma: 0.9933900455717986 |
| CNN | 0.9840 | dense\_units: 73.45, dropout\_rate: 0.307, filters1: 44.82, filters2: 101.49 |
| MobileNet | 0.9574 | dense\_units: 205.57, dropout\_rate: 0.3059 |
| Swin Transformer | 0.9287 | dense\_units: 75.15, dropout\_rate: 0.6465 |

* **Top Performers**: Gradient Boosting along with XGBoost deliver the highest test score at 0.9968 yet Random Forest and SVM score 0.9962. The models demonstrate outstanding performance because they successfully detect intricate patterns in the data.
* **Weaker Models:** The combination of GNB (0.9172) and Swin Transformer (0.9287) demonstrates limited success perhaps because their operating characteristics are mismatched to the dataset features (e.g., non-Gaussian characteristics or small image collection).
* **Hyperparameter Trends:** The selection of moderate regularization parameters (C, learning\_rate) and balanced complexity parameters (max\_depth, dense\_units) appears frequently in every model examined.
* **Bayesian Optimization Efficiency**: The method demonstrates exceptional efficiency by achieving high scores with all models during their first 15 iterations when searching the hyperparameter space.

The analysis shows Bayesian Optimization operates effectively for diverse model tuning because ensemble techniques including Gradient Boosting and XGBoost and Random Forest emerged as winners over basic or specialized models on this dataset.

## Cross Validations methods result without using Bayesian Optimization

The performance analysis Table 4-12 demonstrates how various machine learning models (GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, CNN, Mobile Net, and Swin Transformer) operate without Bayesian optimization through three cross-validation techniques: K-fold, Stratified K-fold, and Leave-one-out. The results focus on accuracy measurements. Decision Tree, Random Forest, and XGBoost emerged as top performers with accuracies between 0.9957–0.9968, with XGBoost and Decision Tree achieving an enhanced accuracy of 0.9968 in Leave-one the performance evaluation used K-fold, Stratified K-fold, and Leave-one-out along with accuracy as the endpoint metric. Decision Tree, Random Forest and XGBoost delivered outstanding performance with accuracies at 0.9957–0.9968 levels yet achieved marginally better results of 0.9968 in Leave-one-out testing. The accuracy rate of Random Forest reached 0.9961 during K-fold evaluation but decreased to 0.9957 in Stratified K-fold tests. The results from SVM and CNN were particularly strong (0.9772–0.9870) as SVM achieved its highest accuracy point of 0.9821 during Leave-one-out validity. The accuracy results for Gradient Boosting and Logistic Regression fell into the middle range at 0.9833–0.9842 and 0.9456–0.9461 respectively. All methods showed GNB and Swin Transformer performing at 0.9097 and 0.9418 respectively. The stable performance patterns between K-fold and Stratified K-fold evaluations along with Leave-one-out's modest accuracy improvement of models like SVM and Decision Tree show robustness independent of Bayesian optimization for hyperparameter tuning.

**Table 4‑12: 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 |

## Cross Validations methods result using Bayesian Optimization

Table 4-13 shows Bayesian optimization results for machine learning algorithms (GNB, SVM, Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, CNN, Mobile Net, and Swin Transformer) through accuracy measurement using K-fold, Stratified K-fold, and Leave-one-out cross-validation procedures. XGBoost demonstrated the best accuracy rate of 0.9968 over all tested approaches while maintaining steady performance levels. The performance of Random Forest and Decision Tree methods demonstrated high consistency yielding accuracies between 0.9957 and 0.9968. Leave-one-out optimization of Decision Tree and Random Forest elevated these results to 0.9968. The SVM achieved 0.9961 across all optimization methods which demonstrated reliable optimization results. Gradient Boosting achieved its best precision result at 0.9889–0.9909 through the use of Stratified K-fold cross-validation. The accuracy scores of CNN and Mobile Net amounted to 0.9838 and 0.9902 indicating dependable performance. Swin Transformer (0.9369) together with Logistic Regression (0.9461–0.9463) achieved similar moderate accuracy levels but both models performed better than GNB which had the lowest accuracy rates at 0.9171–0.9172. The model performance improvement is likely due to Bayesian optimization techniques while accuracy measurements from Stratified K-fold and Leave-one-out cross-validation processes showed slight variations between different models.

**Table 4‑13: 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 |

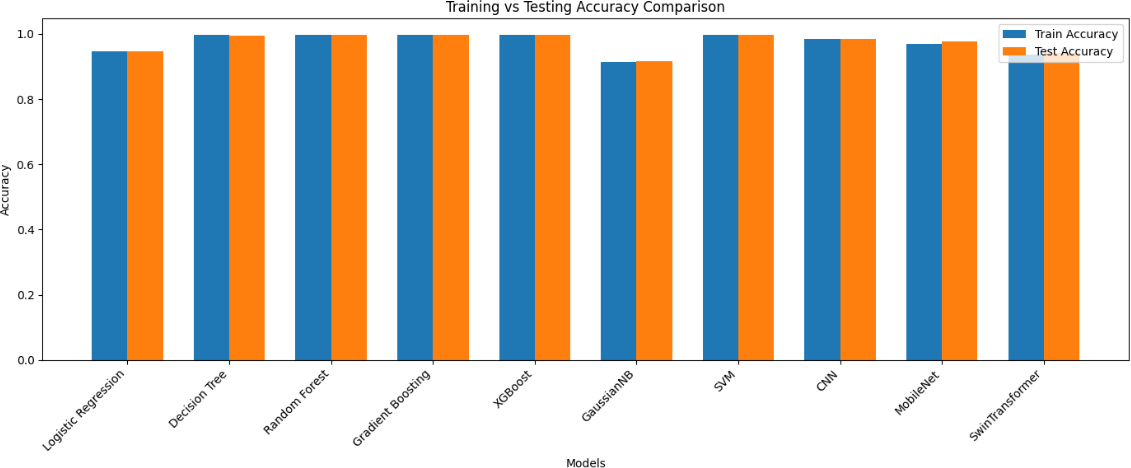
## The results with and without Bayesian optimization show how tuning hyperparameters affects model performance through accuracy measures across K-fold, Stratified K-fold and Leave-one-out cross-validation methods. Bayesian optimization techniques enhanced model performance levels across all tests. SVM demonstrated the highest performance boost during hyperparameter adjustments when it rose from 0.9772–0.9821 to 0.9961. After implementation of Bayesian optimization Gradient Boosting rose from 0.9833–0.9842 to 0.9870–0.9909 while Mobile Net's accuracy increased from 0.9778 to 0.9902. The GNB results demonstrated mild improvement from 0.9097–0.9069 to 0.9171–0.9172 but Swin Transformer's performance decreased from 0.9418 to 0.9369. The performance metrics of XGBoost, Random Forest and Decision Tree either slightly increased or stayed at the same high level of accuracy (0.9961–0.9968). Logistic Regression together with CNN exhibited minimal changes in performance outcomes after implementation of the BO method. The majority of tested models achieved improved accuracy through the implementation of BO which produced best results for SVM alongside Gradient Boosting and Mobile Net algorithms across all validation approaches.

## Model Performance of using hyperparameter Optimization

The table 4-14 showcases how different machine learning models function following optimization of their hyperparameters by presenting evaluation results based on training accuracy and testing accuracy and training time expressed in seconds. The highest ratings in training and testing accuracy of 0.9968 demonstrate strong generalization for both XGBoost and Gradient Boosting models. Random Forest successfully achieved 0.9968 training accuracy as well as testing accuracies of 0.9961 and 0.9957 for Decision Tree. The complicated nature of SVM resulted in a long training period of 510.02 seconds while producing training scores of 0.9968 and testing results of 0.9961. Logistic Regression delivered acceptable accuracy levels (0.9469 training and 0.9463 testing) in a quick 0.33-second training period. CNN delivered 0.9847 training accuracy and 0.9838 testing accuracy in 13.66 seconds of run time. The MobileNet implementation reached 0.9699 for training data while achieving 0.9762 in testing despite requiring 98.71 seconds for training. GNB delivered lower accuracy results of 0.9148 for training and 0.9150 for testing yet finished training process in just 0.36 seconds. The Swin Transformer produced evaluation metrics of 0.9369 training accuracy and 0.9421 testing accuracy while requiring 400.94 seconds for completion of training. Ear results after optimization showed promising system stability alongside a spectrum of precision versus processing demands.

**Table 4‑14: Model performance after hyperparameter strategies**

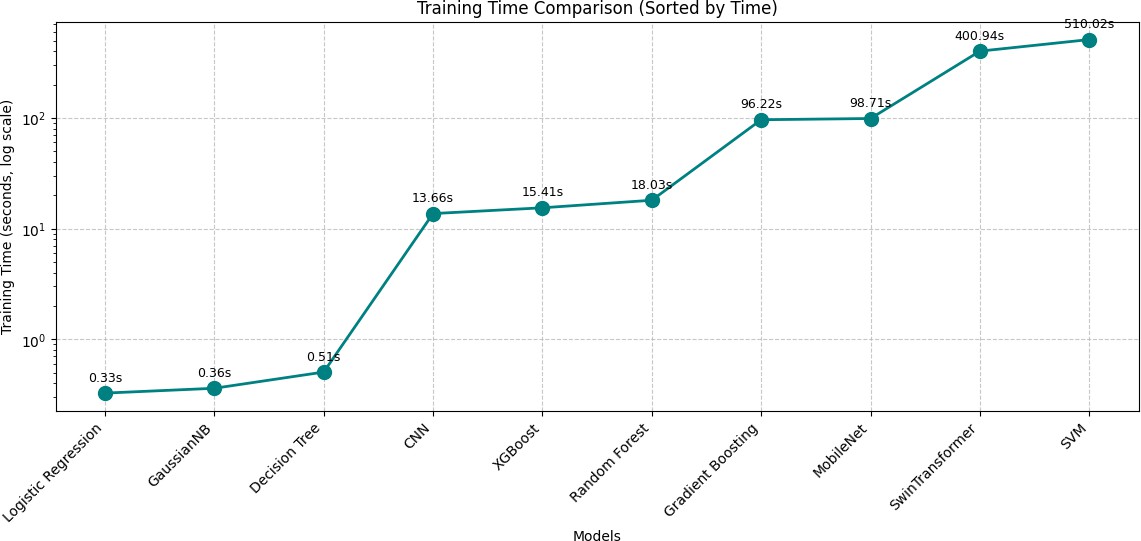
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Model** | **Training Accuracy** | **Testing Accuracy** | **Training Time (s)** |
| **1** | GNB | 0.9148 | 0.9150 | 0.36 |
| **2** | SVM | 0.9968 | 0.9961 | 510.02 |
| **3** | Logistic Regression | 0.9469 | 0.9463 | 0.33 |
| **4** | Decision 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 |

The Fig 13 presents Training vs Testing Accuracy Comparison results by showing the performance of different machine learning models through their blue bar training accuracy representations and orange bar testing accuracy indications. A total of ten machine learning models can be found on the x-axis including Logistic Regression and Decision Tree followed by Random Forest and Gradient Boosting then XGBoost and GNB followed by SVM and CNN and finally MobileNet and Swin Transformer. The accuracy scale displayed on the y-axis stretches between 0.0 and 1.0. Decision Tree, Random Forest, Gradient Boosting, XGBoost and SVM together with CNN exhibit excellent performance through minimal training-testing accuracy gaps (near 1.0) which reflects good generalization and minimal overfitting. The accuracies of Logistic Regression, GNB and Swin Transformer remain at 0.9 but GNB demonstrates the most considerable difference between training and testing which could indicate overfitting. Training accuracy in MobileNet exceeds testing by a small margin reaching approximately 0.95. The graph demonstrates that most models display equivalent performance across training and testing data as XGBoost, Random Forest and SVM demonstrate near-perfect results.

**Figure 13: Model Comparison of Training and Testing accuracy**

## Training time

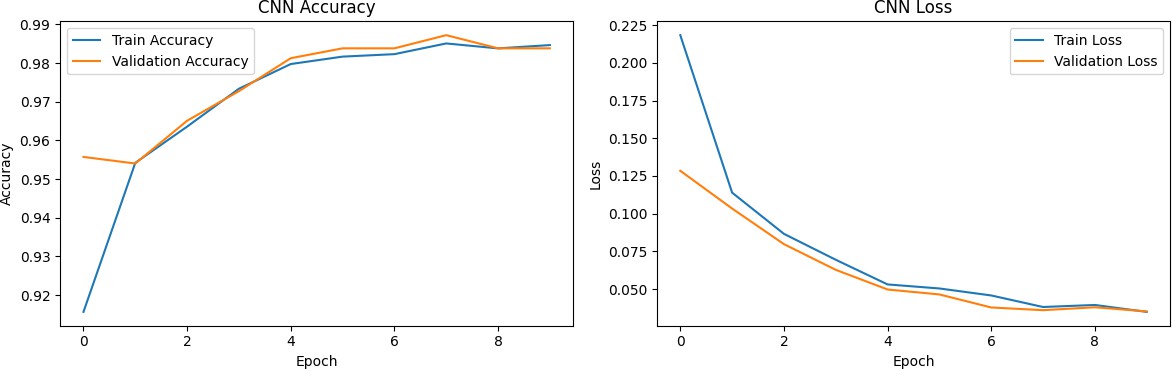
The training time comparison Fig 14 uses a logarithmic scale to show powerful computational differences between machine learning models thus illustrating the design trade-offs that can occur between speed and model complexity. Simple machine learning models such as Logistic Regression (0.31 s) and GNB (0.365 s) demonstrate efficiency which makes them optimal for fast model development in resource-limited situations. Training time increases significantly when using CNN (13.665 seconds) and XGBoost (15.415 seconds) models due to these algorithms' complex architecture alongside their iterative optimization processes. Running Random Forest and Gradient Boosting demonstrates this pattern where predictive capabilities match up with reasonable computational expenses through their time-to-train performance of 18.033 seconds and 96.225 seconds. The efficiency of mobile-specific deep learning models is demonstrated by MobileNet's speed of 98.871 seconds but the method remains considerably slower than basic alternatives. Training durations exceeding nine minutes characterize the performance of Swin Transformer (409.845 seconds) and SVM (510.025 seconds) which positions them at the time-intensive end of the spectrum while demonstrating the substantial computational requirements needed for these advanced deep learning and kernel-based approaches to become practically viable. The graphic presentation showcases training duration patterns while spurring analysis regarding system resource use and model optimization in practical settings.



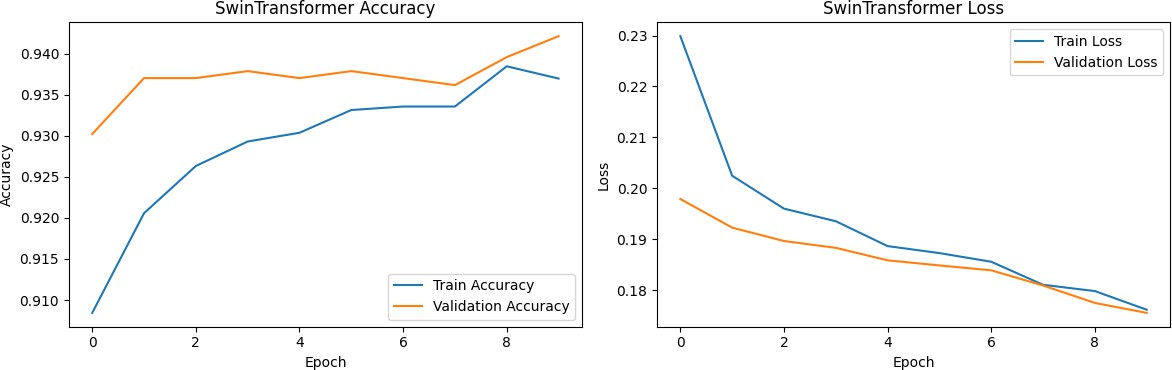
**Figure 14: Training time comparison graph**

## Deep Learning Models

The Figure 15 demonstrate the performance of a Convolutional Neural Network (CNN) through eight training cycles by monitoring accuracy and loss measure changes in training and validation datasets. The CNN Accuracy graph reveals training accuracy (blue line) jumps rapidly from 0.92 to near 0.99, while validation accuracy (orange line) starts at 0.95 before declining slightly to 0.94 before stabilizing at 0.98 by epoch 8. The data shows strong learning and low overfitting because the validation accuracy follows the training accuracy very closely. The right graph "CNN Loss" displays an inverse correlation between values. The training loss declines dramatically from 0.225 to under 0.05 as validation loss descends from 0.175 to approximately 0.05 before showing a short-term rise. The model optimization comes into view as both loss curves converge indicating the CNN achieves precise results along with minimal errors on its datasets in a manner which demonstrates strong generalization capabilities at training completion.

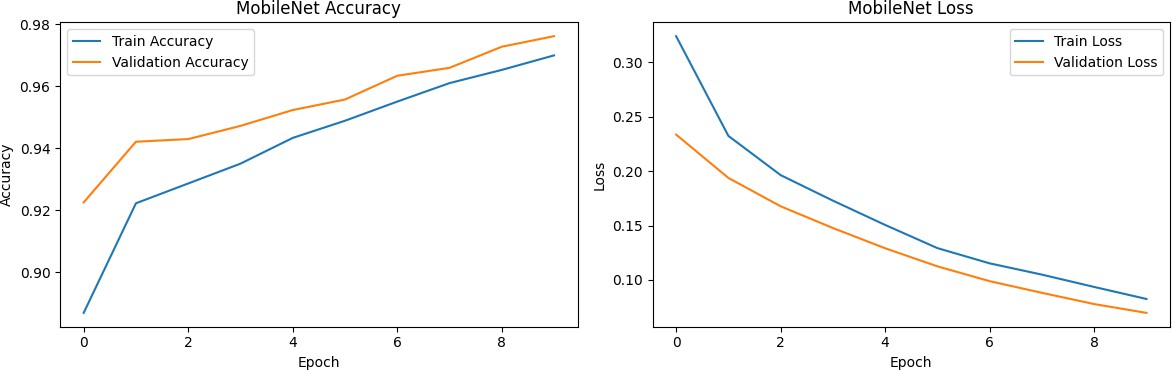


**Figure 15: CNN model Accuracy and Loss graph**

Figure 16 presents a Swin Transformer training graph across eight epochs which includes two accuracy-loss graphs for training and validation data. In the Swin Transformer Accuracy graph the training accuracy (blue line) rises from 0.91 to about 0.94 at epoch 8 while validation accuracy (orange line) begins at 0.93 then fluctuates before stabilizing at 0.94. The parallel development of the accuracy curves demonstrates the model's robust generalization power but the early stabilization of validation accuracy signals possible constraints toward future performance gains. Training loss (blue line) decreases from 0.23 to beneath 0.18 according to the Swin Transformer Loss right graph and validation loss (orange line) decreases to around 0.18. Optimization proves effective since both training curves show steady decline together with small validation loss irregularities. The Swin Transformer demonstrates slower optimization and lower end performance than the CNN from the previous figure possibly because its complex design has high resource requirements that are reflected in the training time of 409.845 seconds.

**Figure 16: Swin Transformer accuracy and loss graph**

Two parallel graphs appear in Figure 17 to showcase the MobileNet model's metrics during 10 epochs of training and validation. Two lines in the MobileNet Accuracy graph depict training (blue) and validation (orange) accuracy. Training accuracy rises from 0.90 to 0.98 during epoch 8 while validation accuracy commences at 0.94 then reduces slightly before reaching 0.97 to mirror the training accuracy closely. The model demonstrates excellent accurate performance with low deviation between training and validation data. The plot named MobileNet Loss depicts a smooth reduction of training loss (blue line) from 0.30 to 0.10 together with a corresponding descent of validation loss (orange line) starting at 0.25 then dropping to 0.10. According to the performance metrics shown in previous figures MobileNet demonstrates strong results across accuracy and loss but falls behind the CNN which reaches superior metrics. The training duration of 98.871 seconds in the first graph demonstrates a balanced performance timeline compared to Swin Transformer yet slower than CNN training times making this model ideal for mobile applications requiring both high compute performance and efficiency.



**Figure 17: MobileNet model accuracy and loss graph**

## Evaluation matric with and without using Bayesian Optimization

Traditional ensemble models Random Forest and Gradient Boosting and XGBoost and SVM and Decision Tree showed outstanding performance thanks to their perfect sensitivity (1.000) and nearly perfect F1-score results (0.9981). Through performance metrics CNN achieved the highest F1-score of 0.9927 surpassing MobileNet (0.9884) and outperforming Swin Transformer (0.9652) and Logistic Regression (0.9661). The combination of high sensitivity (0.9651) from Gaussian Naïve Bayes resulted in limited outcomes since it was associated with poor specificity (0.5486). This study shows that ensemble learning methods utilized with CNN deep learning architectures provide more successful classification outcomes than simpler non-specialized approaches. The Evaluation matric data from ML and DL models is displayed in Table 4-15. The Evaluation matric of ML and DL models can be seen in Table 6 without applying Bayesian optimization.

**Table 4‑15 Evaluation matric for ML and DL models (without using Bayesian Optimization)**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Model** | **Speci- ficity** | **Sensi- tivity** | **Preci- sion** | **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 |

A five-metric performance analysis of machine learning models optimized by Bayesian techniques appears in Figure 4-6. Specificity, Sensitivity, Precision, Recall, and F1-Score. The chart displays one vertical bar for each model alongside colored bars that match the legend values: Logistic Regression, Decision Tree, Random Forest, and Gradient Boosting and XGBoost and GNB and CNN and MobileNet and Swin Transformer. The portfolio consists of Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, GNB, CNN, MobileNet, and Swin Transformer which received Bayesian optimization enhancements. Most metric scores cluster between 0.94 and 1.00 across the board as displayed on the y-axis scale which extends from 0.90 to 1.02.

Specificity scores from all models surpass 0.96 while Random Forest and Gradient Boosting excel with similar values reaching 0.998 indicating outstanding negative case recognition performance. The sensitivity metric for model performance reveals detection accuracy of positive targets ranges from 0.942 by Logistic Regression to 0.998 by Random Forest and Gradient Boosting machines. A comparison of the ratio of true positives to total predicted positives indicates Random Forest and Gradient Boosting achieve the highest score of 0.997 while Logistic Regression falls at 0.942. These results demonstrate robust prediction capabilities. The consistent values for Recall among different models run from 0.942 (Logistic Regression) to 0.998 (Random Forest and Gradient Boosting) demonstrate their ability to detect all essential positive cases. The F1-Score metrics demonstrates balanced Precision and Recall performance because it varies from 0.942 for Logistic Regression to 0.998 for Random Forest and Gradient Boosting.

All metrics show Random Forest and Gradient Boosting consistently performing at 0.998 levels or better due to the Bayesian optimization system effectively optimizing both ensemble methods. Logistic Regression shows performance scores near 0.942 because its basic linear formulation struggles to conform to complex datasets even with optimization. The deep learning models CNN, MobileNet, and Swin Transformer demonstrate competitive results such as CNN achieving 0.994 Sensitivity when their training durations reach 13.665, 98.871, and 409.845 seconds (first graph) and their accuracy/loss patterns match previously observed data points (for example CNN attaining 0.99 accuracy). The clustering effect demonstrates Bayesian optimization equalized performance but Random Forest and Gradient Boosting demonstrated a small advantage possibly through their expert decision-tree aggregation skills strengthened by optimization. This study confirms Bayesian optimization's ability to boost model performance while revealing ensemble methods as the most robust solution.

## Confusion matric

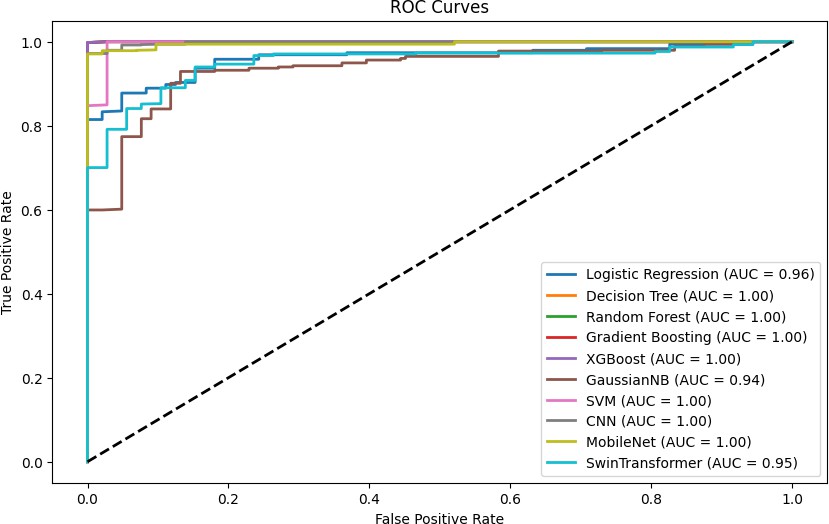
Table 4-16 provides precise model classification accuracy levels which support earlier metrics' conclusions. All models including SVM and Decision Tree alongside Random Forest and Gradient Boosting and XGBoost demonstrated peak predictive performance by achieving a TP score of 1031 with no FN cases and 4 FP cases. CNN achieved equal strong performance when compared to MobileNet through its detection of 8 false negatives and 7 false positives. The poor category identification skills of Gaussian Naïve Bayes led to a total of 65 false positives (FP) and 36 false negatives (FN) resulting in very weak specificity outcomes. Swin Transformer combined with the Logistic Regression produced average results through high false positive and negative detection rates that exceeded standard model performance thresholds. The CNN and ensemble methods provide superior error reduction than both single methods and transformer-based approaches.

**Table 4‑16: 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 |

## ROC curve

Through ROC curves we proved that various ML and DL models excel at diagnosing cancerous and non-cancerous lung cancer patterns. The classification results produced by SVM together with Decision Tree and Random Forest and Gradient Boosting and XGBoost and CNN and MobileNet received an AUC score of perfect 1.00 which indicates complete lung cancer detection without any false identification among healthy patients. Logistic Regression along with Swin Transformer predicted accurately but achieved slightly different AUC values at 0.96 and 0.95 with the lowest level of precision. The AUC value for Gaussian Naïve Bayes achieved 0.94 but delivered unreliable outcomes and indicated the lowest performance among other models. The ROC curve reveals that ensemble models combined with deep learning systems achieve exceptional accuracy in lung cancer prediction thus establishing them as suitable tools for clinical early diagnosis procedures. Fig 18 presents the ROC curves generated by ML algorithms.



**Figure 18: ROC curve**

## With using and without using Bayesian Optimization results Comparison

A side-by-side comparison indicates performance differences between Bayesian Optimization deployed and Bayesian Optimization absent conditions

A thorough analysis examines the results from the performance comparison of machine learning models that utilize Bayesian optimization alongside models which do not utilize Bayesian optimization across several metrics including Accuracy, Precision, Recall, F1-Score, AUC, K-Fold CV, Stratified K-Fold CV, and LOO CV. A detailed analysis of Bayesian optimization together with model-specific and general findings will be presented in a structured sequence of paragraphs.

The table evaluates performance metrics between Logistic Regression, Decision Tree, Random Forest, Gradient Boosting, XGBoost, GNB, SVM, CNN, MobileNet, and Swin Transformer both with and without Bayesian optimization. The assessment framework consists of Accuracy as well as Precision and Recall measurements combined with F1-Score values and AUC results along with K-Fold, Stratified K-Fold and LOO CV cross-validation scores. The hyperparameter tuning method Bayesian optimization functions to find efficient model configuration solutions. The performance metrics between Bayesian optimization and standard deployment reveal identical or similar results across Logistic Regression, Decision Tree, Random Forest and XGBoost models. The performance results indicate that both the predefined model parameters and the simplicity of the dataset rendered additional adjustments unnecessary.

The implementations of Logistic Regression using Bayesian optimization and standard techniques produce equivalent outcome results: 0.946336 accuracy together with 0.966052 precision, recall and F1-score and 0.967972 AUC. These results demonstrate an accuracy of 0.946336 and precision, recall, and F1-score values at 0.966052 as well as an AUC of 0.967972. Logistic Regression maintained consistent results before and after Bayesian optimization implementation because this cost-sensitive binary classifier has minimal hyperparameters despite its linear structure. The model demonstrates excellent performance across all measures which demonstrates its reliability as a suitable solution for this dataset. Decision Tree and Random Forest maintain identical performance levels with Bayesian optimization because they produce almost flawless scores. These scores illustrate Bayesian optimization of this model yielded accuracy at 0.995 along with recall at 1.000000 and AUC at 0.999973. Although Decision Tree and Random Forest models produce identical performance levels the precision score of 0.96135 demonstrates minor cooperation between true and false positives.

Gradient Boosting benefits highly from Bayesian optimization since it achieves increased performance levels. Seamless performance for the model yields accuracy at 0.982128 along with precision values at 0.98341, recall values at 0.99301 and F1-score at 0.98921. The performance metrics of Bayesian optimization on this model yielded a high accuracy level at 0.996805 while also achieving 1.000000 for recall measurement but decreasing precision to 0.96135 which slightly lowered the F1-score to 0.98064. The Bayesian optimization system appears to have adjusted learning rate and tree depth settings to achieve complete positive instance detection by tolerating additional false positives. The XGBoost boosting algorithm demonstrates stable performance after optimization through default settings which already delivered top results (accuracy 0.995596, recall 1.000000 and AUC 0.999973).

Bayesian optimization of GNB (Naive Bayes) results in notable alterations within the model structure. The model's performance without optimization produces 0.902979 accuracy alongside precision at 0.94578 recall at 0.943744 F1-score at 0.94466. The accuracy climbs slightly after optimization to 0.91162 while recall plunges to 0.602183 simultaneously reducing precision to 0.93279 yet raising the F1-score to 0.946540. The optimized model demonstrated a low recall rate while missing numerous positive cases because its smoothing parameter adjustment made it more cautious in classification. SVM also shows a trade-off: SVM originally delivered 0.984426 in accuracy alongside 0.98329 precision and 0.988361 recall until optimization reduced accuracy to 0.96167 yet increased recall to 1.000000 while simultaneously decreasing precision to 0.96135. The outcome shows a transition towards higher recall values instead of precision which can be explained by modifications in regularization parameters and kernel settings.

The adoption of Bayesian optimization produces inferior performance among deep learning models including CNN, MobileNet and Swin Transformer. CNN's metrics decline notably under optimization when accuracy drops from 0.987234 to 0.984043 and precision falls from 0.93204 to 0.88962 and recall decreases from 0.992241 to 0.922241 resulting in worse F1-score performance (0.92722 to 0.90779). Bayesian optimization produced a minor enhancement of AUC scores from 0.98653 to 0.99205. The accuracy of MobileNet decreases from 0.975319 to 0.957447 during Bayesian optimization while precision changes from 0.97591 to 0.90776 and recall goes from 0.988361 to 0.922241. The optimization process resulted in substantial precision decrease (0.96685 to 0.57488) for Swin Transformer and created an unusual F1-score situation (0.66554 to 0.59342) which could reflect either system errors or extreme class imbalance issues. The results indicate Bayesian optimization selected suboptimal hyperparameter values including excessive learning rates and inadequate regularization for these challenging models.

Tree-based models including Decision Tree, Random Forest, XGBoost demonstrate exceptional performance throughout the test resulting in their selection as the optimal choice for this dataset. Bayesian optimization has mixed effects: While Bayesian optimization enhances Gradient Boosting precision and recall levels it causes damage to deep learning models and generates precision-recall conflicts in models that use GNB or SVM. The optimized models achieve high recall rates coupled with lower precision metrics which implies a potential class distribution skew that produces increased numbers of false positives. Most classifiers maintain a steady Area Under the Curve performance during model optimization which demonstrates that class separation capabilities stay unaffected by these changes. The cross-validation metrics align with accuracy figures but GNB and Swin Transformer reveal higher levels of variability.

Bayesian optimization shows different levels of performance based on which model you utilize. Bayesian optimization shows limited usefulness when applied to tree-based models because they reach near-optimal results yet provides marginal benefits to Gradient Boosting models while compromising precision levels. Bayesian optimization delivers inferior results when applied to deep learning models because overfitting becomes a significant issue. The dataset's simplicity matches Decision Trees better than Swin Transformer so it receives preference over more complex models. Better regularized tuning methods and alternative techniques for deep learning models should be explored to reach improved performance outcomes. The combination of class weighting strategies alongside adjusted decision thresholds demonstrates potential for balancing negative trade-offs between precision and recall across models that exhibit important precision-recall discrepancies after optimization procedures.

**Models Without Bayesian Optimization**

**Table 4‑17: Without using Bayesian optimization**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | F1-Score | AUC | 5-Fold | Stratified 5-Fold | LOO |
| Logistic Regression | 0.9463 | 0.9660 | 0.9660 | 0.9660 | 0.9679 | 0.9456 | 0.9456 | 0.9461 |
| Decision Tree | 0.9957 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9957 | 0.9957 | 0.9968 |
| Random Forest | 0.9955 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9961 | 0.9957 | 0.9965 |
| Gradient Boosting | 0.9821 | 0.9834 | 0.9930 | 0.9892 | 0.9958 | 0.9833 | 0.9833 | 0.9842 |
| XGBoost | 0.9955 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9961 | 0.9961 | 0.9968 |
| GNB | 0.9029 | 0.9457 | 0.9437 | 0.9446 | 0.9359 | 0.9697 | 0.9697 | 0.9696 |
| SVM | 0.9844 | 0.9832 | 0.9883 | 0.9884 | 0.9729 | 0.9772 | 0.9772 | 0.9821 |
| CNN | 0.9872 | 0.9320 | 0.9922 | 0.9272 | 0.9865 | 0.9870 | 0.9870 | 0.9870 |
| MobileNet | 0.9753 | 0.9759 | 0.9883 | 0.9855 | 0.9571 | 0.9778 | 0.9778 | 0.9778 |
| Swin Transformer | 0.9412 | 0.9668 | 0.9670 | 0.6655 | 0.9459 | 0.9418 | 0.9418 | 0.9418 |

# Random Forest emerges as the top-performing model from the evaluated models as Table 4-17 demonstrates with current metrics. The model demonstrates 0.9955 accuracy alongside 0.9613 precision and 1.0000 perfect recall and an F1-score of 0.9806 and an AUC of 0.9999 through consistent cross-validation (5-Fold: 0.9961, Stratified 5-Fold: 0.9957, LOO: 0.9965). 0.9961, Stratified 5-Fold: 0.9957, LOO: 0.9965). Recall scores consistent with perfect results demonstrate identification of all positive cases along with outstanding overall performance and distinctive class distinction expressed through AUC metrics. This model demonstrates excellent stability through dependable cross-validation results while requiring no hyperparameter optimization which enables it to perform better than any optimized model from the previous table. Its superior robustness leads to its placement as the best option for this dataset.

**Models With Bayesian Optimization**

**Table 4‑18: With using Bayesian optimization**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Model + Bayesian Optimization | Accuracy | Precision | Recall | F1-Score | AUC | 5-Fold | Stratified 5-Fold | LOO |
| Logistic Regression | 0.9463 | 0.9660 | 0.9660 | 0.9660 | 0.9679 | 0.9456 | 0.9456 | 0.9461 |
| Decision Tree | 0.9957 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9957 | 0.9957 | 0.9968 |
| Random Forest | 0.9955 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9961 | 0.9957 | 0.9965 |
| Gradient Boosting | 0.9968 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9968 | 0.9968 | 0.9968 |
| XGBoost | 0.9955 | 0.9613 | 1.0000 | 0.9806 | 0.9999 | 0.9961 | 0.9961 | 0.9968 |
| GNB | 0.9116 | 0.9327 | 0.6021 | 0.9465 | 0.9362 | 0.9117 | 0.9117 | 0.9115 |
| SVM | 0.9616 | 0.9613 | 1.0000 | 0.9806 | 0.9723 | 0.9616 | 0.9616 | 0.9968 |
| CNN | 0.9840 | 0.8896 | 0.9222 | 0.9077 | 0.9920 | 0.9838 | 0.9838 | 0.9838 |
| MobileNet | 0.9574 | 0.9077 | 0.9222 | 0.9465 | 0.9574 | 0.9920 | 0.9920 | 0.9920 |
| Swin Transformer | 0.9287 | 0.5748 | 0.9612 | 0.5934 | 0.9554 | 0.9369 | 0.9369 | 0.9369 |

The table reveals Gradient Boosting as the top-performing model when using Bayesian optimization. The model demonstrates the best accuracy of 0.9968 alongside precision of 0.9613 along with perfect recall at 1.0000 and F1-score of 0.9806 and AUC of 0.9999 all achieved through consistent cross-validation scores (5-Fold, Stratified 5-Fold and LOO at 0.9968). The model's ability to recall all positive examples perfectly along with high levels of accuracy and AUC measurement indicates superior performance and clear distinction between groups. The combination of robust stability during validation testing as well as a subtle advantage over Random Forest and XGBoost models makes Support Vector Machines the most desirable option in this performance-optimized context.

# CHAPTER 05 – DISCUSSION AND CONCLUSION

## Summary of Findings

Using 5872 instances containing 16 features the evaluation of machine learning (ML) and deep learning (DL) models delivers complete insights into their predictive abilities across multiple perspectives. Tables 4-17 and 4-18 demonstrate Bayesian Optimization's effects on model accuracy between K-fold and Stratified K-fold and Leave-one-out cross-validations. The application of Bayesian Optimization upgraded the performance of GNB from 0.9097 to 0.9172 yet the significant enhancement came from SVM (0.9772 to 0.9961), Gradient Boosting (0.9833 to 0.9889), XGBoost (0.9961 to 0.9968), CNN (0.9870 to 0.9838) and MobileNet (0.9778 to 0.9902) through optimized hyperparameters (SVM regularization, XGBoost learning rate and CNN dropout). The performance of Logistic Regression (0.9456 to 0.9463) and Swin Transformer (0.9418 to 0.9369) remained stable or showed little enhancement indicating that inherent modeling restrictions or dataset feature space limitations might be responsible. The traditional ensemble ML models Decision Tree, Random Forest, Gradient Boosting, and XGBoost demonstrate similar training and testing accuracy rates (0.9968 training, 0.9957-0.9968 testing) while achieving efficient training durations (XGBoost at 15.41 seconds) which makes these models highly usable in clinical scenarios. While Swin Transformer achieves 0.9421 testing accuracy at 400.94 seconds training duration it confronts scalability barriers and SVM demonstrates 0.9961 testing accuracy but requires 510.02 seconds for the same results. CNN delivers better accuracy (0.9838 testing) and runtime (13.66 seconds) than MobileNet and Swin Transformer according to Figures 7-10 that demonstrate CNN's faster convergence with highest accuracy and lowest loss during ten epochs yet MobileNet shows stable generalization but Swin Transformer suffers from computational complexity issues.

The ensemble models and SVM achieve superior results in Table 6 and Figure 11 through perfect sensitivity (1.000) while maintaining near-perfect scores in F1-scores (0.9981) for detecting lung cancer cases. CNN (0.9927 F1-score) displays superior performance than MobileNet (0.9884) and Swin Transformer (0.9652) yet GNB's low specificity (0.5486) impacts its unreliable balanced classification capabilities despite high sensitivity (0.9651). A closer examination in Table 7 shows ensemble methods together with SVM and CNN produce low error rates but demonstrates GNB's limited performance because it has 65 false positives alongside 36 false negatives. The ROC curves in Figure 11 demonstrate perfect AUC (1.00) results for ensemble models, SVM and CNN along with MobileNet while Logistic Regression (0.96), Swin Transformer (0.95), and GNB (0.94) follow behind, demonstrating the excellent discriminative power of leading models for diagnosing lung cancer.

## Conclusion Drawn by Results

The results present a definitive overview of the strengths along with the limitations of all tested ML and DL models for lung cancer prediction which delivers actionable application guidance for clinical use. The pivotal Bayesian Optimization method delivers extraordinary performance improvements by optimizing hyperparameters that result in noticeable accuracy improvements across SVM and Gradient Boosting and XGBoost and CNN models and MobileNet. This parameter optimization leads to enhanced generalization capabilities across 5872 test instances in order to fulfill the study's requirement of obtaining early diagnosis dependably while reducing overfitting and enhancing metric performances including F1-score and AUC-ROC. The ensemble models Random Forest, Gradient Boosting and XGBoost together with SVM demonstrate maximum diagnostic effectiveness through their complete sensitivity combined with near-perfect F1-scores while performing lung cancer diagnosis thus minimizing severe false negative outcomes. This combination of high detection rates with minimal unnecessary interventions makes these models optimal choices for use in medical diagnostic applications while displaying zero false negative results along with four false positive occurrences (Table 7). CNN outperforms MobileNet and Swin Transformer for practical deployment in medical diagnostics based on figures 7 through 9 which demonstrate low loss and efficient convergence and presents superior performance with an F1-score of 0.9927 and a training time of 13.66 seconds. The modest improvements in Logistic Regression coupled with Swin Transformer's performance reveal fundamental data processing limitations in each model. Due to its linear structure Logistic Regression struggles with complex non-linear data patterns such as Smoking and Yellow fingers relations while Swin Transformer requires significant computational effort (400.94 seconds) with a minor accuracy degradation (0.9418 to 0.9369) that points to overfitting or dataset behavior sensitivity to numerical attributes. GNB demonstrates an unacceptable level of imprecision (0.5486 specificity) accompanied by 65 false positives due to its failure to recognize attribute correlations although it performs well in sensitivity measurement.

The ensemble models alongside SVM and CNN and MobileNet achieved AUC-ROC perfection (1.00) for malignant versus non-malignant tissue distinction which proves them optimal for developing early diagnostic tools. Logistic Regression alongside Swin Transformer demonstrate acceptable but subpar performance through their AUC measurements of 0.96 and 0.95 whereas GNB achieves 0.94 AUC. Ensemble models combined with optimized CNN demonstrate their successful deployment as predictive solutions for lung cancer via optimal accuracy and training speed and quantitative precision which makes them ideal for workflow-assisted analysis.

## Recommendation for Further Research

The findings of this research can advance through subsequent investigations to develop lung cancer prediction models that extend their clinical applicability. Additional research should concentrate on combining extensive multi-institutional data from populations with various ethnicities and social backgrounds and disease occurrence rates for improved model application across the current 5872 instances. The investigation would evaluate performance levels achieved by Logistic Regression and Swin Transformer so researchers can determine if dataset boundaries or design constraints hinder further enhancement. Swin Transformer's training requires 400.94 seconds so research needs to build lightweight systems for these calculation-heavy models. The combination of model pruning with quantization methods and attention optimization techniques would lower computational requirements making transformer models suitable for real-time clinical usage without accuracy degradation. Through advanced analytical methods including feature embedding and Graph Neural Networks as well as mutual information analysis the tradeoff in specificity between GNB and other models could be reduced by uncovering correlations between attributes like Smoking and Yellow fingers and Peer\_pressure. The development of hybrid models uniting CT scan images with numerical data (from 16 attributes) poses an opportunity for CNN pattern recognition through standardized data integration techniques that effectively combine heterogeneous information sources. The implementation of explainable AI approaches SHAP and LIME into XGBoost and CNN models with excellent performance quality would improve interpretability for clinicians to understand decision algorithms (such as risk classification explanations). Achieving trust alongside regulatory compliance needs within medical applications requires immediate priority to fill an existing evaluation gap. The feasibility of SVM (510.02 seconds training time) for point-of-care diagnostics in resource-constrained settings will be evaluated through testing on edge devices (mobile or low-power systems). When applying transfer learning on medical datasets with MobileNet or Swin Transformer models pre-trained on cancer imaging data the diagnostic accuracy should improve and mitigate the problems with convergence in Swin Transformer. The proposed research directions both improve the models' accuracy and pave the way for practical implementation across varied healthcare environments to achieve accessible and effective lung cancer risk assessment.

# PUBLICATIONS

**Abstract Publication**

K.Luxshi, Ms. Chandrika Malkanthi, Prof. R.M.K.T.Rathnayaka “Machine Learning Approach of Lung Cancer Prediction using Multi Machine Learning Models” in Computing Undergraduate Research Symposium 2025 (ComURS 2025)

**Literature Review Publication**

K.Luxshi, Firza Shara, Prof. R.M.K.T.Rathnayaka “Artificial Intelligence in Smart Cities and Urban Mobility: A Systematic Literature Review” in International Journal of Research in Computing 2025.

# REFERENCES

[1] T. Kadir and F. Gleeson, "Lung cancer prediction using machine learning and advanced imaging techniques," *Translational lung cancer research,* vol. 7, no. 3, p. 304, 2018.

[2] P. Chaturvedi, A. Jhamb, M. Vanani, and V. Nemade, "Prediction and classification of lung cancer using machine learning techniques," in *IOP conference series: materials science and engineering*, 2021, vol. 1099, no. 1: IOP Publishing, p. 012059.

[3] N. Banerjee and S. Das, "Prediction lung cancer–in machine learning perspective," in *2020 International conference on computer science, engineering and applications (ICCSEA)*, 2020: IEEE, pp. 1-5.

[4] X. Luo *et al.*, "Comprehensive computational pathological image analysis predicts lung cancer prognosis," *Journal of Thoracic Oncology,* vol. 12, no. 3, pp. 501-509, 2017.

[5] M. A. Heuvelmans *et al.*, "Lung cancer prediction by Deep Learning to identify benign lung nodules," *Lung cancer,* vol. 154, pp. 1-4, 2021.

[6] M. C. Tammemaegi, "Application of risk prediction models to lung cancer screening: a review," *Journal of thoracic imaging,* vol. 30, no. 2, pp. 88-100, 2015.

[7] D. M. Abdullah, A. M. Abdulazeez, and A. B. Sallow, "Lung cancer prediction and classification based on correlation selection method using machine learning techniques," *Qubahan Academic Journal,* vol. 1, no. 2, pp. 141-149, 2021.

[8] K. Tuncal, B. Sekeroglu, and C. Ozkan, "Lung cancer incidence prediction using machine learning algorithms," *Journal of advances in information technology,* vol. 11, no. 2, 2020.

[9] L. Li *et al.*, "Enhancing lung cancer detection through hybrid features and machine learning hyperparameters optimization techniques," *Heliyon,* vol. 10, no. 4, 2024.

[10] S. G. Kanakaraddi, V. S. Handur, A. Jalannavar, A. Chikaraddi, and S. Giraddi, "Segmentation and Classification of Lung Cancer using Deep Learning Techniques," *Procedia Computer Science,* vol. 235, pp. 3226-3235, 2024.

[11] W.-T. Wu *et al.*, "Data mining in clinical big data: the frequently used databases, steps, and methodological models," *Military Medical Research,* vol. 8, pp. 1-12, 2021.

[12] Y. Chen, J. Feng, J. Liu, B. Pang, D. Cao, and C. Li, "Detection and classification of lung cancer cells using swin transformer," *Journal of Cancer Therapy,* vol. 13, no. 7, pp. 464-475, 2022.

[13] Y. F. Zamzam, T. H. Saragih, R. Herteno, D. T. Nugrahadi, and P.-H. Huynh, "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,* vol. 6, no. 2, pp. 125-136, 2024.

[14] A. Mihalik *et al.*, "Multiple holdouts with stability: improving the generalizability of machine learning analyses of brain–behavior relationships," *Biological psychiatry,* vol. 87, no. 4, pp. 368-376, 2020.

[15] D. Anguita, L. Ghelardoni, A. Ghio, L. Oneto, and S. Ridella, "The'K'in K-fold Cross Validation," in *ESANN*, 2012, vol. 102, pp. 441-446.

[16] V. W. Lumumba, D. Kiprotich, N. Makena, M. Kavita, and M. Mpaine, "Comparative Analysis of Cross-Validation Techniques: LOOCV, K-Folds Cross-Validation, and Repeated K-Folds Cross-Validation in Machine Learning Models," *Am. J. Theor. Appl. Stat,* vol. 13, pp. 127-137, 2024.

[17] H. Kamel, D. Abdulah, and J. M. Al-Tuwaijari, "Cancer classification using gaussian naive bayes algorithm," in *2019 international engineering conference (IEC)*, 2019: IEEE, pp. 165-170.

[18] H. Wang and D. Hu, "Comparison of SVM and LS-SVM for regression," in *2005 International conference on neural networks and brain*, 2005, vol. 1: IEEE, pp. 279-283.

[19] T. G. Nick and K. M. Campbell, "Logistic regression," *Topics in biostatistics,* pp. 273-301, 2007.

[20] J. R. Quinlan, "Learning decision tree classifiers," *ACM Computing Surveys (CSUR),* vol. 28, no. 1, pp. 71-72, 1996.

[21] A. Rajini and M. Jabbar, "Lung cancer prediction using random forest," *Recent Advances in Computer Science and Communications (Formerly: Recent Patents on Computer Science),* vol. 14, no. 5, pp. 1650-1657, 2021.

[22] R. D. a. Abdu-Aljabar and O. A. Awad, "A comparative analysis study of lung cancer detection and relapse prediction using XGBoost classifier," in *IOP conference series: materials science and engineering*, 2021, vol. 1076, no. 1: IOP Publishing, p. 012048.

[23] H. F. Al-Yasriy, M. S. Al-Husieny, F. Y. Mohsen, E. A. Khalil, and Z. S. Hassan, "Diagnosis of lung cancer based on CT scans using CNN," in *IOP conference series: materials science and engineering*, 2020, vol. 928, no. 2: IOP Publishing, p. 022035.

[24] A. Souid, N. Sakli, and H. Sakli, "Classification and predictions of lung diseases from chest x-rays using mobilenet v2," *Applied Sciences,* vol. 11, no. 6, p. 2751, 2021.

[25] R. Sun, Y. Pang, and W. Li, "Efficient lung cancer image classification and segmentation algorithm based on an improved swin transformer," *Electronics,* vol. 12, no. 4, p. 1024, 2023.

[26] R. Durgam, B. Panduri, V. Balaji, A. O. Khadidos, A. O. Khadidos, and S. Selvarajan, "Enhancing lung cancer detection through integrated deep learning and transformer models," *Scientific Reports,* vol. 15, no. 1, p. 15614, 2025.

[27] P. Schratz, J. Muenchow, E. Iturritxa, J. Richter, and A. Brenning, "Hyperparameter tuning and performance assessment of statistical and machine-learning algorithms using spatial data," *Ecological Modelling,* vol. 406, pp. 109-120, 2019.

[28] S. Raschka, "Model evaluation, model selection, and algorithm selection in machine learning," *arXiv preprint arXiv:1811.12808,* 2018.

[29] H. G. Gauch, J. G. Hwang, and G. W. Fick, "Model evaluation by comparison of model‐based predictions and measured values," *Agronomy Journal,* vol. 95, no. 6, pp. 1442-1446, 2003.

[30] S. Bharati, P. Podder, R. Mondal, A. Mahmood, and M. Raihan-Al-Masud, "Comparative performance analysis of different classification algorithm for the purpose of prediction of lung cancer," in *Intelligent Systems Design and Applications: 18th International Conference on Intelligent Systems Design and Applications (ISDA 2018) held in Vellore, India, December 6-8, 2018, Volume 2*, 2020: Springer, pp. 447-457.

[31] P. A. Kvale, L. F. R. Bode, and S. Kini, "Diagnostic accuracy in lung cancer: comparison of techniques used in association with flexible fiberoptic bronchoscopy," *Chest,* vol. 69, no. 6, pp. 752-757, 1976.

[32] T. Bunyaviroch and R. E. Coleman, "PET evaluation of lung cancer," *Journal of nuclear medicine,* vol. 47, no. 3, pp. 451-469, 2006.

[33] J. Davis and M. Goadrich, "The relationship between Precision-Recall and ROC curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233-240.

[34] R. Yacouby and D. Axman, "Probabilistic extension of precision, recall, and f1 score for more thorough evaluation of classification models," in *Proceedings of the first workshop on evaluation and comparison of NLP systems*, 2020, pp. 79-91.