Available online at www.ijournalse.org

Emerging Science Journal

(ISSN: 2610-9182)

Vol. 10, No. 1, February, 2026

Comparative Assessment of Machine Learning Approaches for Early Lung Cancer Diagnosis

Garvit Maheshwari [1], Babita Tiwari [2*], Domonkos Tinka [3], Satyanand Singh [4]

[1] *Department of Mechanical Engineering, Manipal University Jaipur, Rajasthan, 303007, India.*

[2] *Department of Computer Science and Engineering, Manipal University Jaipur, Jaipur, Rajasthan, India.*

[3] *Doctoral School of Economic and Regional Sciences, Széchenyi István University, 9026 Győr, Hungary.*

[4] *School of Electrical & Electronics Engineering, Fiji National University, Fiji Island.*

Abstract

Lung cancer, a leading cause of cancer-related mortality worldwide, often escapes early detection due to the absence of distinct symptoms in its initial stages. This work investigates how Machine Learning (ML) might improve early diagnosis by analyzing Electronic Health Records (EHR) data. Multiple ML models were developed and evaluated on a synthetic dataset created to replicate real-world patient characteristics, allowing controlled experimentation while safeguarding privacy. Model performance was tuned using both conventional optimization methods and nature-inspired approaches, with the aim of balancing predictive accuracy and computational efficiency. In our synthetic dataset experiments, ensemble learners optimized with metaheuristic techniques reached accuracy levels approaching 99 percent while maintaining computational efficiency and generally outperformed simpler baselines. The contribution of this work lies in exploring the integration of GFO and WOA for feature selection and hyperparameter tuning of XGBoost, together with a soft-voting ensemble. This approach provides an experimental pathway for enhancing predictive performance under computational constraints. However, as the dataset is synthetic, the conclusion remains experimental; validation against clinical records will be essential before translation into practice.

Keywords:

Electronic Health Records;
Ensemble Learning;
Hybrid Metaheuristics Algorithm;
Lung Cancer Diagnosis;
Machine Learning.

Article History:

Received:	29	August	2025
Revised:	16	January	2026
Accepted:	23	January	2026
Published:	01	February	2026

1- Introduction

1-1- Lung Cancer: A Global Health Threat

Lung cancer was hardly seen a century ago; today, for men and women, it ranks as the primary cause of cancer-related death globally [1]. GLOBOCAN 2020 forecasts indicate that lung cancer accounted for around 1.8 million fatalities (18%), globally, in 2020 alone [2]. The subtle development of the disease, sometimes free of obvious early signs, presents major difficulties for appropriate diagnosis and treatment [1]. Still, the main cause is tobacco use, particularly smoking, in over 80% of all cases. Further worsening the worldwide burden of this disease are exposures to harmful substances such as asbestos, radon, and air pollution [3].

1-2- Causes and Effects of Lung Cancer

A complex disease, lung cancer results from a confluence of several risk factors greatly raising the probability of its occurrence [4]. These factors can be categorized into the following:

* **CONTACT**: babita.tiwari@jaipur.manipal.edu

DOI: http://dx.doi.org/10.28991/ESJ-2026-010-01-02

© 2026 by the authors. Licensee ESJ, Italy. This is an open access article under the terms and conditions of the Creative Commons Attribution (CC-BY) license (https://creativecommons.org/licenses/by/4.0/).

Tobacco Smoke: At roughly 85% of all cases, tobacco smoke remains the main risk factor for lung cancer. Whether from cigarettes, cigars, or pipes, the harmful compounds in tobacco smoke—such as tar, nicotine, and carcinogens—damage lung cell DNA, hence initiating uncontrolled cell proliferation and cancer development [5].

Occupational Hazards: Lung cancer risk is raised by several jobs exposing people to carcinogens. Once extensively employed in building and insulation, asbestos is a proven carcinogen intimately associated with lung cancer, especially mesothelioma. Workers in construction, mining, and manufacturing sectors are at risk from other occupational hazards, including arsenic, chromium, nickel, radon, and diesel exhaust [4].

Environmental Pollutants: Another major risk factor is air pollution, whereby prolonged exposure to particulate matter—an amalgamation of solid particles and liquid droplets—causes inflammation and destruction of lung tissue. Sources include power plants, industrial pollutants, and car exhaust. Higher lung cancer risk has been associated particularly with more polluted urban regions [3].

Exposure to Everyday Items: A few often-used substances might raise lung cancer risk. For instance, arsenic-contaminated water can cause long-term health issues, including lung cancer. Moreover, factors advised to increase risk in certain research include high consumption of processed and red meats as well as alcohol [6].

Knowing these risk factors helps one to forecast their likelihood of lung cancer. Including information on smoking history, occupational exposures, environmental factors, and other relevant lifestyle and health data into electronic health records allows machine learning algorithms to use this comprehensive dataset to increase the accuracy of lung cancer projections. Early identification and fast response made possible by this can help to improve patient outcomes and maybe save lives by allowing prompt intervention.

1-3- Diagnosing Lung Cancer

Improving the survival rates in lung cancer cases depends on timely diagnosis [3]. Still, the complicated character of the illness sometimes causes delays in early diagnosis. Common diagnostic techniques include:

Physical Examination: Using a physical examination, doctors can evaluate the chest for anomalies, including tumors or inflammation [7].

Imaging: Potential lung tumors are found using diagnostics like computed tomography (CT), magnetic resonance imaging (MRI), and chest X-rays [1].

Bronchoscopy: This procedure directly examines the lungs by passing a thin, flexible tube containing a light and camera through the airways [7].

Biopsy: A definitive diagnosis calls for looking under a microscope at a tissue sample taken from a suspected abnormal area to find cancer cells [8].

Although these methods are widely used, they can be invasive, time-intensive, and subject to variability in interpretation, underscoring the need for complementary computational approaches.

1-4- The Challenges of Diagnosis and the Potential of Machine Learning

Lung cancer diagnosis is intrinsically difficult, and the sheer volume of data produced by contemporary medical technologies emphasizes the need for sophisticated algorithms able to identify tiny patterns that conventional studies could ignore. Machine learning offers great possibilities to address these issues. ML techniques, which also help to uncover trends buried in patient symptoms, genetic profiles, and electronic health records, fit large and sophisticated datasets.

ML application in lung cancer diagnostics offers promise in several important areas:

Improving Diagnostic Accuracy: By analyzing complex EHR datasets.

Enhancing Early Detection: ML can help with timely intervention by spotting trends suggestive of early-stage lung cancer that might escape conventional approaches.

Streamlining the Diagnostic Process: By means of automated analysis using ML, dependence on subjective interpretation is lessened, hence improving efficiency and reducing human error in diagnostic procedures.

Recent work has explored these possibilities through EHR-based pipelines. Inclusive risk models for both smokers and nonsmokers have demonstrated promising predictive accuracy but were affected by bias, missing data, and lack of external validation [9].

Other studies have targeted narrower challenges. For instance, smoking status has been extracted from free-text clinical notes using explainable AI methods, which achieved near-perfect accuracy but remained language- and system-specific [10]. Similarly, transformer-based approaches have shown that sequential care pathways in EHRs can be modeled to capture temporal dependencies and improve predictive value, yet these models are computationally demanding and not easily transferable across healthcare systems [11].

Large language models such as GPT-4 have also been tested for phenotype extraction, producing strong results but at high cost and within constrained datasets [12]. In parallel, gradient boosting methods have been applied to large real-world cohorts, achieving robust discrimination but relying heavily on imaging features and still lacking independent validation [13].

Finally, systematic reviews confirm that deep learning applied to imaging remains the dominant approach in lung cancer prediction. While these models offer high accuracy, they continue to face limitations of interpretability, reproducibility, and integration with multimodal data such as EHR [14]. While these capabilities are promising, their practical implementation requires careful validation in real-world settings to ensure accuracy, scalability, and clinician trust.

1-5- Research Gap

While deep learning methods applied to medical imaging have achieved high diagnostic accuracy in lung cancer detection [14], their dominance has left EHR-based research comparatively underdeveloped. Existing EHR studies remain fragmented: some address smoking status extraction from unstructured text [10], others design inclusive models for younger adults and nonsmokers [9], while transformer-based architectures capture sequential care pathways with high accuracy but substantial computational cost [11]. More recent work has applied GPT-4 to phenotype extraction [12] or gradient boosting on regional cohorts [13], yet these efforts remain limited either by data scope, lack of external validation, or reliance on imaging features.

This fragmentation highlights a persistent gap: the absence of lightweight, resource-conscious frameworks that can passively monitor routinely updated EHRs to raise early alerts. Such systems would not replace established diagnostic methods but could complement them by enabling scalable background surveillance, particularly valuable in populations who may not undergo screening despite high risk. Moreover, many advanced approaches—such as transformer-based models or large language models—are computationally intensive, creating barriers for deployment in smaller hospitals or resource-limited settings. Although federated learning offers potential for collaboration across institutions while preserving data privacy [15], lightweight hybrid optimization strategies for EHR-based prediction remain relatively unexplored.

1-6- Goal

Using electronic health records data to include the intricate interactions between the causes and consequences of lung cancer into prediction models has the potential to greatly enhance the detection and treatment of this disease. Healthcare professionals can create customized treatment plans, make more informed judgments, and gain insight into individual patient risk profiles by using machine learning and data analytics, thereby helping to reduce the burden of lung cancer. ML’s inclusion in the diagnostic process can extract meaningful information from EHR data, therefore enabling developments in early detection and management. Large datasets spanning symptoms, medical history, genetic information, and lifestyle factors—that may not be readily obvious to human doctors—showcase ML systems in finding subtle patterns and relationships. Improved early diagnosis and intervention help to increase this capability, hence improving lung cancer treatment; moreover, it streamlines the diagnostic process for better efficiency.

The aim of this work is to assess the ability of several ML techniques to detect diagnostic trends in EHR records. The database contains thorough information, including patient symptoms, characteristics, medical history, and medicines. For this exploratory purpose, we used a synthetic dataset generated through the Synthea Learning Health System [16], which provides realistic yet fully privacy-preserving patient records. In particular, we introduce a hybrid optimization pipeline combining Grey Wolf and Whale Optimization algorithms for feature selection and hyperparameter tuning, together with an XGBoost-centered ensemble designed to balance predictive performance with computational efficiency.

This work is presented as an exploratory step in applying ML for lung cancer detection. As the dataset is synthetic, the findings should be regarded as preliminary and require validation with real-world clinical data before practical implementation. In this way, the study contributes to ongoing discussions by comparing traditional and hybrid models and motivating further research to refine and test these methods in clinical environments.

2- Literature Review

One of the most common and deadly tumors in the world, lung cancer still kills a lot of people; early discovery greatly helps to improve patient outcomes. Particularly in the analysis of medical imaging, genetic markers, and patient records, researchers keep investigating many computational and data-driven approaches to improve diagnosis accuracy.

Electronic Health Records (EHRs) are a great tool for lung cancer research thanks to a thorough understanding of patient demographics, medical history, and test data. By meticulously reviewing these records, researchers have identified trends that might help individualised treatment planning and early diagnosis [17]. Investigating radiomics-based classifiers to distinguish between two main histologic subtypes of lung cancer, adenocarcinoma and squamous cell carcinoma, Wu et al. (2016) [18] undertook an important study. This work emphasizes the possibility of imaging-based analysis in guiding clinical care since histologic subtypes affect therapy options. Their work created multivariate classifiers evaluated on an independent patient cohort and found radiomic characteristics highly linked with histology. The study supports the importance of radiomics in improving precision medicine even if the prediction performance (AUC of 0.72) points to room for development.

Building on radiomics studies, Kumar et al. (2017) [19] presented a computational model meant to enhance CT imaging for data lung cancer diagnosis. Their efforts included fresh sequencing methods meant to improve categorization accuracy. Although the strategy improved over past techniques, the results highlight the need of more validation before clinical application. To address the integration of information from several imaging modalities (e.g., PET, CT, MRI) and the choice of suitable classifiers, Zhou et al. (2017) [20] tackled major hurdles in predictive modelling. Using evidential reasoning, their suggested classifier fusion approach merged modality-specific classifiers to produce optimal predictive models. Although more research is required to assess its resilience over several clinical environments, the study showed better performance than individual classifiers. Turning now to genomic analysis, Yuan et al. (2017) [21] examined variations in gene expression between squamous cell carcinoma and lung adenocarcinoma. The work found important genetic markers separating the two subtypes by use of support vector machines and feature selection approaches. These results advance knowledge of the molecular differences between lung cancer subtypes and might help to guide the creation of more focused treatment plans.

Apart from genetic analysis and imaging, scientists have investigated several diagnostic strategies. By means of exosome analysis in circulating blood, Shin et al. (2019) [22] devised a non-invasive early lung cancer detection technique. Their work classified molecules in exosomes using a computational model and used surface-enhanced Raman spectroscopy (SERS) to investigate chemical patterns, suggesting its possible use as a pre-screening tool. The method showed good performance in separating cancer-derived from normal exosomes. More evidence is needed, though, to evaluate its relevance in regular clinical practice. In the framework of treatment planning, Wang et al. (2019) [23] put up a multi-objective deep learning model meant to enhance radiotherapy decision-making. With consideration for sensitivity, specificity, and AUC during model selection, their approach used predictive modeling approaches to evaluate therapy outcomes. Although their results show the possibility for tailored therapy plans, more research is required to guarantee generalizability in several healthcare environments.

At the same time, Rieke et al. (2020) [15] provided a perspective on the future of digital health using federated learning (FL). Unlike conventional centralized approaches, FL enables collaborative model training across multiple institutions while keeping patient data local, thus directly addressing privacy and governance challenges. The authors reviewed applications in EHR-based event prediction, brain tumor segmentation, and mammography, showing that FL can match or outperform centralized training while maintaining privacy. They also highlighted critical challenges such as non-IID data, potential information leakage through gradients, and system-level requirements for traceability and secure computation. Their work positions FL as essential infrastructure for privacy-preserving AI in healthcare, laying groundwork for later EHR-focused ML studies.

Research has also explored feature selection methods to improve lung cancer diagnosis. Enhesari et al. (2021) [24] developed a technique meant to maximize the choice of informative features from patient datasets, hence lowering computing complexity while preserving diagnostic accuracy. Using just 11 features, their method—applied to a dataset of 32 patient records with 57 features—achieved an accuracy of 80.63%. This emphasises in clinical decision support the possibilities of effective feature selection techniques.

One more promising avenue for early identification is blood-based biomarkers. Investigating metabolic biomarkers in plasma samples, Xie et al. (2021) [25] found clear signals linked with early-stage lung cancer. Their work highlighted the promise of blood-based screening techniques by showing great accuracy in differentiating stage I cancer patients from healthy people. Still, adding other biological markers such as genetic data may improve diagnosis accuracy. With an eye towards routinely gathered medical data, Gould et al. (2022) [26] created a predictive model based on clinical and laboratory data. Using gradient boosting methods, their method beat conventional risk assessment models and showed the capacity to identify cancer risk months before clinical diagnosis. This study supports the possibility of data-driven methods in enabling earlier intervention.

Detection of lung cancer depends much on image segmentation. Senthil Kumar et al. (2019) [27] investigate how to raise segmentation accuracy using computational optimization methods. Their research showed that, especially when combined with preprocessing methods improving image clarity, the Guaranteed Convergence PSO (GCPSO) algorithm attained exceptional segmentation results.

Many recent investigations have investigated fresh computer approaches for cancer diagnosis. A hybrid strategy combining convolutional neural networks (CNNs) with feature optimization methods was proposed by Vijh et al. (2023) [28], thus improving accuracy while lowering computing costs. To improve CNN-based lung cancer classification, Priyadharshini & Zoraida (2023) [29] also presented a hybrid model including several optimization techniques. Their results imply that well-chosen optimization strategies can support more accurate diagnosis models.

Gupta et al. (2019) [30] investigated evolutionary algorithms for analysis of CT images of lung cancer, further enhancing feature selection techniques. Their work showed that an enhanced grey wolf method minimized computing cost and selected pertinent image features, hence producing high classification accuracy. Another area of interest has come from the evolution of ensemble classification techniques. Designed to improve diagnosis accuracy and efficiency, Alzubi et al. (2023) [31] suggested a neural network-based classification method using improved feature selection strategies; their system showed speed and accuracy gains over conventional classification approaches.

In recent years, alongside the well-established focus on imaging-based pipelines, researchers have also begun to explore EHR and text-driven approaches, often in the context of retrospective cohorts or pilot implementations. These studies remain fewer in number than imaging work but represent a growing strand of research, complemented by analyses on large real-world datasets and systematic reviews. Building on this trajectory, Chen et al. (2024) [9] developed inclusive ML models for lung cancer prediction using EHR data that cover both smokers and nonsmokers. The study reported strong predictive accuracy; however, it also noted bias, missing data, and the absence of external validation, highlighting practical barriers to wider deployment. Complementing this, Ebrahimi et al. (2024) [10] addressed unstructured clinical text by automatically identifying smoking status from Danish EHR notes using explainable AI. Performance was high across binary and multiclass tasks; nevertheless, the approach remained language- and system-specific, limiting straightforward transfer to other settings.

Wang et al. (2024) [11] advanced sequential modeling by proposing a transformer-based framework that captures temporal care pathways in large-scale EHR data. The method substantially outperformed traditional baselines; yet it required significant computational resources and lacked external validation, posing challenges for real-time clinical integration. In parallel, Bhattarai et al. (2024) [12] demonstrated that GPT-4 can extract complex phenotypes from clinical notes with strong performance in several categories. Despite these gains, the study's small patient cohort, single-institution scope, and computational expense temper the immediate generalizability of results. Moving to large cohort analyses, Su et al. (2025) [13] validated gradient-boosting models for lung cancer risk prediction in a high-risk population, achieving strong discrimination in training and validation cohorts. The approach relied heavily on CT imaging features and still lacked independent external validation, which constrains portability across health systems.

Finally, Liz-López et al. (2025) [14] provided a systematic review of deep learning in lung cancer detection. The review confirmed the dominance of imaging-based methods and high headline accuracy while underscoring persistent issues of interpretability, heterogeneous evaluation protocols, and limited multimodal integration with EHR data.

Taken together, these more recent efforts suggest an evolution toward incorporating EHR-derived signals and unstructured clinical text into lung cancer prediction research. At the same time, the literature shows fragmentation across tasks and methodologies. High-capacity models can report strong performance; however, they are often resource-intensive, context-specific, and rarely validated externally. Lightweight optimization and ensemble strategies—of the kind explored in this study—may offer one potential path toward scalable deployment, particularly for institutions with limited computational resources. Overall, the body of evidence points to the importance of approaches that balance predictive performance with efficiency while emphasizing privacy-preserving collaboration and, critically, validation on real-world clinical data before routine adoption.

3- Methodology and Implementation

3-1- Dataset Description

This work uses a synthetic dataset produced using Synthea, a Learning Health System (LHS) built to replicate realistic patient data for machine learning research in healthcare environments [16]. Comprising over 17,000 simulated cases, the dataset spans around 785 characteristics. These characteristics include a range of demographics, medical history, symptoms, drugs, and therapeutic results. Synthea enables exploration of machine learning algorithms for lung cancer prediction inside a regulated yet complicated healthcare environment by offering a complete and varied dataset.

For model development, four dataset partitions were merged to form the training/validation set, while a fifth independent partition was retained exclusively for final testing. The positive (lung cancer) and negative (non-lung cancer) classes were moderately imbalanced, with fewer positive cases. To address this, stratified splits were used to preserve class ratios during training/validation, and class weights were applied within the algorithms to mitigate bias. We did not apply oversampling or under sampling strategies, since the dataset itself as synthetic and already included diverse patient scenarios.

3-2- Data Preprocessing

To ensure the synthetic dataset Synthea generated was suited for machine learning applications, extensive feature engineering and data preparation were conducted. Medically relevant features were first identified using exploratory data analysis (EDA); then relationships were evaluated, and non-relevant data was deleted. Missing values were addressed by eliminating entries with notable missing information; minor gaps in the surviving dataset were filled in via median imputation. Categorical features were converted to binary values, and continuous variables were standardized using StandardScaler to ensure comparability across features.

Feature engineering entails categorizing EHR data into several groups, including diagnosis (e.g., respiratory and cardiovascular disorders), drugs (e.g., pain relievers and cardiovascular meds), and treatments (e.g., therapies and diagnostic procedures). By streamlining the data, this methodical approach made analysis more easily tractable. The dataset was scaled to guarantee that it matched machine learning models; a last review confirmed data completeness and quality. This careful preparation ensured the dataset was ready for developing prediction models to enhance lung cancer detection and control. Still, leaning too much on synthetic data emphasizes the need for confirming findings on real-world datasets. To contextualise these steps, Figure 1 presents a complete workflow of the study, spanning dataset preparation, model development, optimization, and evaluation.

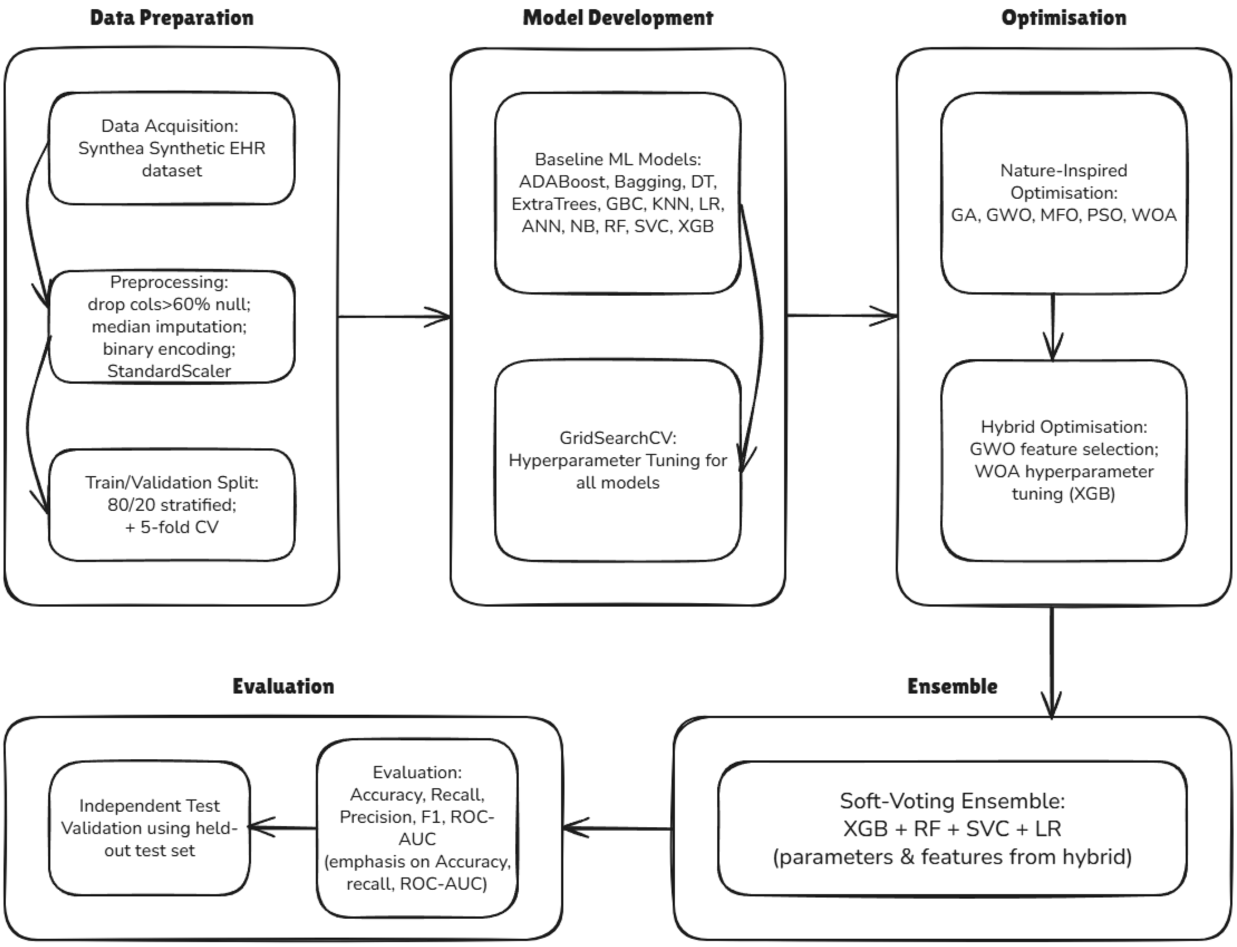
Data Preparation
Data Acquisition: Synthea Synthetic EHR dataset
Preprocessing: drop cols>60% null; median imputation; binary encoding; StandardScaler
Train/Validation Split: 80/20 stratified; + 5-fold CV
Model Development
Baseline ML Models: ADABoost, Bagging, DT, ExtraTrees, GBC, KNN, LR, ANN, NB, RF, SVC, XGB
GridSearchCV: Hyperparameter Tuning for all models
Optimisation
Nature-Inspired Optimisation: GA, GWO, MFO, PSO, WOA
Hybrid Optimisation: GWO feature selection; WOA hyperparameter tuning (XGB)
Ensemble
Soft-Voting Ensemble: XGB + RF + SVC + LR (parameters & features from hybrid)
Evaluation
Evaluation: Accuracy, Recall, Precision, F1, ROC-AUC (emphasis on Accuracy, recall, ROC-AUC)
Independent Test Validation using held-out test set

Figure 1. Workflow Diagram

3-3- Classifier Selection

Several classification models are evaluated in this paper to find the most appropriate one for the prediction of lung cancer. The selected algorithms are ensemble approaches, tree-based models, distance-based techniques, linear models, a neural network approach, and a probabilistic classifier. This wide spectrum enables one to assess performance comprehensively in several machine learning models.

Deep learning approaches, while effective in image-based cancer diagnosis, were not used here because this study focused on structured EHR data, where traditional ML algorithms can perform competitively at lower computational cost. A central goal was to design a lightweight, resource-conscious pipeline suitable for routine deployment, which is less feasible with transformer or CNN-based approaches. Hyperparameter optimization was performed for each model using GridSearchCV with stratified 5-fold cross-validation, ensuring fair and robust comparisons.

3-3-1- ADABOOST Classifier [32, 33]

One often utilized ensemble technique chosen for its capacity to improve the performance of weak learners is ADABOOST. The method runs repeatedly, changing the weights of samples depending on past misclassifications. ADABOOST trains a sequence of weak learners—such as decision trees—by concentrating progressively on harder-to-classify events and then aggregates them using weighted aggregation into a strong learner. For datasets with complicated trends especially, this iterative improvement procedure is quite helpful. ADABOOST may thus be less resilient in some situations since it can be sensitive to noisy data.

- ***Mathematical Formulation***

The ADABOOST ensemble model's final prediction is a weighted combination of the predictions of its weak learners:

$$f(x) = \mathrm{sign}(\textstyle\sum_{m=1}^{M} \alpha_m h_m(x)) \tag{1}$$

where: f(x) is the final ensemble model's prediction: $h_m(x)$ is the prediction of the (m)-th weak learner, α_m is the weight assigned to the (m)-th weak learner, determined by its accuracy on the training data; M is the total number of weak learners.

- ***Hyperparameters***

Tuning the following hyperparameters is crucial for optimal ADABOOST performance:

- Number of Estimators (M): This determines how many weak learners are included in the ensemble.
- Learning Rate (eta): This controls the contribution of each weak learner to the ensemble.
- Base Estimator: This specifies the type of weak learner used (e.g., decision tree, stump).
- Criterion (for Decision Tree Base Estimator): This is the criterion used to measure the quality of splits in the decision tree (e.g., Gini impurity, entropy).
- Max Depth (for Decision Tree Base Estimator): This limits the depth of the decision trees used as weak learners.

3-3-2- Bagging Classifier [34, 35]

Designed to lower overfitting and increase the stability of base learners, bagging—bootstrap aggregating—is an ensemble technique. Usually with replacement, the method generates several bootstrap samples by randomly choosing portions of the training set. Every sample is utilized to separately teach a basic learner, like a decision tree. All base learners' outputs are combined during prediction either by average (for regression tasks) or majority vote (for classification tasks). Although bagging is very good at lowering variance and enhancing generalization, for big datasets it may need computationally demanding training.

- ***Mathematical Formulation***

The ensemble prediction is the mode of individual base learner predictions:

$$\widehat{y_{\mathrm{bag}}}(x) = \mathrm{mode}\big(\widehat{y_1}(x), \widehat{y_2}(x), \dots, \widehat{y_B}(x)\big) \tag{2}$$

where: y^$_{\mathrm{bag}}$(x) is the bagging ensemble prediction for input x; y^i(x) is the prediction of the (i)-th base learner; *mode*(·) returns the most frequent class label among predictions.

- ***Hyperparameters***

Tuning the following hyperparameters is crucial for optimizing Bagging:

- Base Estimator: The choice of base learner (e.g., decision tree, linear model) affects the overall performance.
- Number of Estimators (B): Higher numbers can improve accuracy but increase computation time.
- Bootstrap Samples: Whether to use bootstrapping influences data diversity for each base learner.
- Max Samples: This determines the size of each bootstrap sample.

3-3-3- Decision Tree Classifier [36, 37]

Included for their simplicity, interpretability, and good baseline performance under the specified dataset conditions were Decision Tree (DT) methods. Targeting maximum homogeneity of target classes within each subset, decision trees build classification models by recursively partitioning the dataset depending on feature values. The model ends when some stopping criteria, such as uniform subgroups or predetermined depth, are reached; the splits are shown as branches. Decision trees are less robust without pruning or other regularizing methods even if they are transparent and easy to use. Their overfitting is especially prone when the tree gets too deep.

- ***Mathematical Formulation***

Decision trees partition the feature space by selecting the feature at each node that best splits the data into more homogeneous subgroups according to the target class. The selection criterion is often based on minimizing impurity measures. Two common choices:

○ Gini Impurity: For a given node *t*:

$$G_t = 1 - \sum_{k=1}^{n} p_{t,k}^2 \tag{3}$$

Where $p_{t,k}$ is the proportion of samples belonging to class *k* at node *t*.

○ Information Gain: Based on entropy, it measures the reduction in uncertainty about the target class after the split. For a node *t*:

$$IG(D_p, f) = H(D_p) - \sum_{j=1}^{m} \frac{N_j}{N_p} H(D_j) \tag{4}$$

where: D_p is the set of data points at the parent node; *f* is the feature used for the split; *m* is the number of branches created by the split; N_j is the number of data points in child node *j; N_p* is the number of data points in the parent node; *H*(*D*) is the entropy of dataset *D*

- ***Hyperparameters***

Key hyperparameters that guide decision tree construction and impact this study include:

○ Criterion: Impacts how the algorithm determines optimal feature splits. We will experiment with both ’gini’ (Gini impurity) and ’entropy’ (information gain) to compare performance.

○ Max Depth: Controls tree complexity and can mitigate overfitting. Different maximum depths will be explored during tuning.

○ Min Samples Split/Min Samples Leaf: These parameters govern the minimum data points allowed before further splitting or creation of terminal ‘leaf” nodes. Adjustments here can prevent overly specific tree structures.

3-3-4- Gradient Boosting Classifier [37, 38]

Gradient Boosting is an ensemble technique used to progressively integrate weak learners—usually decision trees—to raise predicting performance. Gradient Boosting, unlike Bagging, stresses on matching each new learner to the errors of the previous ensemble, hence lowering residual errors. Particularly suited for datasets with complicated interactions, this iterative technique lets the model progressively improve accuracy. Without proper hyperparameter tweaking, Gradient Boosting can, however, be computationally taxing and prone to overfit.

- ***Mathematical Formulation***

Gradient boosting iteratively reduces a loss function through sequential addition of weak learners. The algorithm fits weak learners to the negative gradient of the loss function with respect to the predictions of the current ensemble. Let us denote:

○ F(*x*) as the ensemble model

○ L(*y*, *F*(*x*)) as the loss function

○ $h_m(x)$ as the *m*-th weak learner

In each iteration *m*, gradient boosting approximately solves:

$$h_m = \arg\min_h \sum_{i=1}^{N} - \left[\frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}\right] h(x_i) \tag{5}$$

Then the ensemble is updated:

$$F_m(x) = F_{m-1}(x) + \nu h_m(x) \tag{6}$$

where v is the learning rate.

- ***Hyperparameters***

The following hyperparameters will be significantly evaluated for their impact on gradient boosting performance:

- Learning Rate: Regulates the degree to which each weak learner alters the ensemble. Smaller learning rates demand a higher number of weak learners but might enhance generalization.
- Number of Trees: Dictates how many weak learners comprise the ensemble. Increasing this can boost performance but elevates computational demands.
- Tree-Specific Parameters: These govern attributes of individual decision trees (e.g., max depth, min samples split, max features).
- Sub-sample: This controls the portion of data points used to train each tree. Introducing randomness through sub-sampling can mitigate overfitting.

3-3-5- K-Nearest Neighbors [39, 40]

Simplicity and an easy approach to categorization defined the K-Nearest Neighbors (KNN) algorithm as the choice. KNN works under the presumption that, depending on feature traits, similar data points most certainly belong to the same class. KNN, unlike other methods, does not include an explicit training phase. Rather, it determines, from a query point to all other points in the dataset, distances—e.g., Euclidean distance. The query is categorized using the majority class among the K nearest neighbors. Although KNN is sensitive to the choice of K and the distance metric and straightforward to build and successful for small datasets, it can be computationally expensive for bigger datasets.

- ***Mathematical Formulation***

KNN relies on distance calculations. The most common metric is Euclidean Distance:

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{n} \left(x_{ik} - x_{jk}\right)^2} \tag{7}$$

where, x_i and x_j are two data points, each with n features.

- ***Hyperparameters***

The following hyperparameters within KNN are crucial to optimize:

- K: The number of neighbors that 'vote' on the class label. A careful selection is vital – too low K risks overfitting, while too high a K might lead to underfitting.
- Distance Metric: A choice (Euclidean, Manhattan, etc.) impacts how similarity between data points is measured.
- Weight Function: Optionally, the influence of neighbors can be weighted by their distance to the query point, prioritizing closer neighbors.

3-3-6- Logistic Regression Classifier [41, 42]

Included for its simplicity, interpretability, and efficiency in binary classification tasks was logistic regression—a well-known statistical model. Though its name suggests otherwise, logistic regression is a linear model that maps predicted values to probabilities using the logistic (sigmoid) function. This method uses thresholding of the probabilities to allow classification. Particularly useful for linearly separable data, logistic regression may have difficulty with complicated, non-linear relationships unless expanded with feature transformations or regularization methods.

- ***Mathematical Formulation***

The core of logistic regression lies in the logistic (or sigmoid) function, which maps any real-valued number into a probability between 0 and 1. The logistic function is given by:

$$p(y = 1|x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x_1+\beta_2 x_2+\cdots+\beta_n x_n)}} \tag{8}$$

where: $p(y = 1|x)$ is the probability of the target variable y being 1 (positive class) given the input features x; $\beta_0, \beta_1, \ldots, \beta_n$ are the model's coefficients (parameters) learned during training; $x_1, x_2, \ldots, x_n$ are the input features.

- ***Hyperparameters***

Tuning of the following hyperparameters is essential for effective logistic regression:

- Regularization Parameter (C): This controls the strength of regularization, which helps prevent overfitting by penalizing large model coefficients.

- ○ Penalty: The type of regularization (L1 or L2) determines how coefficients are penalized.
- ○ Solver Algorithm: This is the optimization algorithm used to find the best model parameters. Different solvers have varying efficiency and scalability.
- ○ Class Weight: This parameter allows for adjusting weights for different classes to mitigate issues arising from class imbalance.

3-3-7- Neural Network Classifier [43, 44]

Because they could replicate intricate, non-linear patterns in data, artificial neural networks (ANNs) were chosen. Connected nodes arranged into layers make up ANNs; each node uses an activation function to apply a weighted transformation to its inputs. The network learns by progressively changing these weights to reduce the prediction-actual value error. Although ANNs are quite versatile and good at capturing complicated interactions, their complexity sometimes requires major processing resources and careful tweaking to prevent overfitting, especially for small datasets.

- ***Mathematical Formulation***

The fundamental operation in an ANN involves a weighted sum of inputs, followed by a non-linear activation function:

$$z = \sum_{i=1}^{n} w_i \cdot x_i + b \tag{9}$$

$$a = \sigma(z) \tag{10}$$

where; z represents the weighted sum of inputs (x_i) and a bias term (b); w_i: weights associated with each input; σ: activation function (e.g., sigmoid, ReLU); a: output of the neurons.

- ***Hyperparameters***

Optimizing ANN performance requires tuning of the following hyperparameters:

- ○ Number of Layers: The depth and intricacy of a network depend on its layer count. A balance must be struck between representational power and overfitting.
- ○ Neurones per Layer: Each hidden layer's count of neurons affects the capacity of the model to learn.
- ○ Activation Function: Choices in sigmoid, tanh, or ReLU influence non-linear transformations in the network.
- ○ Learning Rate: Important for convergence and stability, the learning rate affects the weight updating step size.
- ○ Regularization: Techniques such as dropout, batch normalization, and L1/L2 regularization help to reduce overfitting.

3-3-8- Naïve Bayes [45, 46]

Included for its simplicity and computing economy was the Naive Bayes classifier, a probabilistic model produced from Bayes' theorem. The method simplifies often difficult but necessary predictions in practice by assuming that features are conditionally independent given the class label. Naive Bayes performs well in many situations, particularly with text or categorical data, although its performance may suffer if the independence condition is much broken.

- ***Mathematical Formulation***

Naive Bayes is built on Bayes' theorem:

$$P(y|x_1, x_2, \dots, x_n) = \frac{P(y) \cdot P(x_1, x_2, \dots, x_n|y)}{P(x_1, x_2, \dots, x_n)} \tag{11}$$

where: $P(y|x_1,x_2,\dots,x_n)$ is the posterior probability of class y considering features x_1 through x_n; $P(y)$ is the previous probability of class y; $P(x_1,x_2,\dots,x_n|y)$ is the probability of detecting features x_1 through x_n given class y; $P(x_1,x_2,\dots,x_n)$ is the proof (probability of seeing the features).

Conditional independence's "naive" assumption streamlines the probability:

$$P(x_1, x_2, \dots, x_n|y) = P(x_1|y) \cdot P(x_2|y) \cdots P(x_n|y) \tag{12}$$

- ***Hyperparameters***

Naive Bayes requires tuning of several elements.

- ○ Distribution Assumption: We will discuss several Naive Bayes versions (Gaussian, Bernoulli) depending on the type of the features.

- Smoothing Parameter (alpha): This value adds a small constant to prevent overfitting and handles zero probabilities in feature likelihoods.
- Prior Probabilities: Should past understanding of class distribution exist, it can be included here.

3-3-9- Random Forest & Extra Trees Classifier [35, 47]

Selected as an ensemble approach for its capacity to manage high-dimensional data and its resistance against overfitting, Random Forest, along with its version Extra Trees (Extremely Randomized Trees), was used with variations in their method of randomness during tree generation, both techniques aggregate forecasts from many decision trees to increase accuracy.

- Random Forest: Each trained on a bootstrapped subset of the data, Random Forest assembles an ensemble of decision trees. Furthermore, at every split, just a random selection of features is taken into account. Features randomness combined with data helps to reduce overfitting and improve generalization. For classification problems, the last prediction is obtained by means of majority vote among all trees.
- Extra Trees: Like Random Forest, Extra Trees creates several decision trees but adds more unpredictability by choosing split points totally at random, without considering feature relevance. Though it can result in lower interpretability than Random Forest, this method also helps to lower variance and might improve generalization in particular datasets.

- ***Mathematical Formulation***

Random Forest and Extra Trees are ensemble learning methods based on decision trees. Both models aggregate predictions from many decision trees to improve generalization and reduce overfitting.

For a classification problem, let:

- $X=\{x_1,x_2,\ldots,x_n\}$ be the feature vector.
- Y is the target class.

Each individual decision tree Tt in the ensemble predicts a class $\hat{}yt$, and the final class prediction is obtained via majority voting:

$$\hat{y} = \arg\max_{y} \sum_{t=1}^{T} 1\,!\,(T_t(X) = y) \tag{13}$$

where: $T_t(X)$ represents the output of the t^th decision tree; T is the total number of trees; $\nVdash(T_t(X) = y)$ is an indicator function that counts the votes for class y.

The key difference between Random Forest and Extra Trees lies in how splits are chosen:

- Random Forest: Split points are selected by searching for the best feature threshold based on an impurity measure (e.g., Gini impurity or entropy for classification, variance reduction for regression).
- Extra Trees: Split points are chosen randomly, without considering optimal splits, further reducing variance.

- ***Hyperparameters***

Tuning the following hyperparameters greatly affects model performance:

- Number of Trees (T): Determines the number of decision trees in the ensemble. Increasing T improves stability but increases computation.
- Max Features (m): Defines how many features are randomly selected at each split. Lower values increase randomness, reducing overfitting.
- Split Criterion:

Gini Impurity:

$$G = 1 - \sum_i p_i^2 \tag{14}$$

Entropy:

$$H = -\sum_i p_i \log p_i \tag{15}$$

- Minimum Samples per Split (min samples split): Controls the minimum number of samples required to create a split, preventing overfitting.

- ○ Maximum Depth (max depth): Limits the depth of trees to prevent excessive complexity.
- ○ Randomness in Split Selection: *Random Forest:* Searches for the best split; *Extra Trees:* Chooses splits randomly.

3-3-10- Support Vector Classifier [48, 49]

The choice of the Support Vector Classifier resulted from its ability to handle both linear and non-linear classification issues. SVC maximizes the margin between data points of different classes by means of an ideal hyperplane in the feature space. SVC uses kernel functions—e.g., RBF or polynomial—for non-linearly separable data to translate the data into a higher-dimensional space where separation becomes possible. SVC is sensitive to hyperparameter decisions and computationally demanding for big datasets, even if it shines in accuracy and adaptability.

- ***Mathematical Formulation***

○ Linear SVC: The decision function of linearly separable data is:

$$f(x) = \text{sign}(w \cdot x + b) \tag{16}$$

where: The weight vector is *w*; The input feature vector is *x*; The Bias term is *b*.

○ Non-linear SVC: In non-linearly separable data, a kernel function K(x_i,x_j) converts data into a higher-dimensional space:

$$f(x) = \text{sign}(\textstyle\sum_i \alpha_i y_i K(x_i, x) + b) \tag{17}$$

where: α_i are Lagrange multipliers; y_i are support vector class labels.

- ***Hyperparameters***

Effective SVC performance depends on the hyperparameters' tuning:

- ○ C (Regularization Parameter): This balances classification error with margin maximizing.
- ○ Kernel: The choice of kernel function (linear, polynomial, RBF, or sigmoid) depends on the data's characteristics.
- ○ Gamma (for RBF kernel): Affects the non-linear case decision boundary's form.
- ○ Class Weight: Adjusts for class imbalance, if present.

3-3-11- XGBoost Classifier [38, 50]

Strong performance in machine learning challenges and real-world applications made XGBoost, a highly efficient and scalable implementation of gradient boosting, a top choice. The system enhances standard boosting methods with significant developments in regularization to reduce overfitting, a tree-learning algorithm tuned for sparse data, and support for parallel processing. Like other boosting techniques, XGBoost generates an ensemble of successive decision trees, each trained to correct the errors of the previous ensemble. But XGBoost advances conventional gradient boosting by adding:

- ○ Regularization: Penalties for too complex models are imposed using L1 and L2 regularization, hence enhancing generalization.
- ○ Handling Sparse Data: The algorithm features a split-finding approach meant especially to efficiently address sparse or missing data.
- ○ Efficiency: For big datasets XGBoost is quicker and more scalable due to optimizations including parallel computing and out-of-core processing.

These characteristics balance predictive performance with computational economy, therefore making XGBoost especially appropriate for high-dimensional data.

- ***Mathematical Formulation***

XGBoost maximizes an objective function with a regularization component and a component of loss:

$$\text{Objective} = \textstyle\sum_{i=1}^{n} L(y_i, \hat{y_i}) + \sum_{k=1}^{K} \Omega(f_k) \tag{18}$$

where: L(y_i, $\hat{y}_i$) is the loss function measuring the variation between actual label y_i and predicted label $\hat{y}_i$; Ω(f_k) shows the regularization term for the *k*-th tree, aiming to prevent overfitting.

- ***Hyperparameters***

Optimizing XGBoost depends on tuning of the following hyperparameters:

- ○ Learning Rate (eta): Calculates the step size for every boosting iteration.
- ○ Max Depth: Control complexity by limiting tree depth, hence preventing overfitting.
- ○ Sub-sample: Fraction of training instances used for each tree.
- ○ Column Sub-sample (col_sample_bytree): Fraction of features used for each tree.
- ○ Number of Trees (num_boost_round): Total number of trees in the ensemble.

3-4- Nature Inspired Algorithm

This work explores the possibilities of nature-inspired algorithms (NIAs), outside of conventional machine learning models. Inspired by natural and biological processes, NIAs are optimization techniques defined by their imaginative search and approach. This paper addresses five NIAs: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Grey Wolf Optimizer (GWO), Whale Optimization Algorithm (WOA), and Moth-Flame Optimization (MFO). These algorithms were selected because they could investigate several solution environments and find trends possibly missed by more conventional approaches. The hypothesis is that the special mechanisms of NIAs could improve hyperparameter tuning and feature selection for models of lung cancer prediction.

3-4-1- Genetic Algorithm [51, 52]

Inspired by ideas of biological evolution, genetic algorithms were applied to maximize machine learning model hyperparameters. GAs run by preserving a population of candidate solutions, sometimes known as “chromosomes”, which change over consecutive generations. GAs steadily raises the population’s fitness by means of genetic operations, including selection, crossover, and mutation. Especially appropriate for this adaptable and flexible approach is investigating large and complex hyperparameter ranges. Still, GAs can be computationally expensive—especially for high-dimensional datasets (see Figure 2).

- ***Steps of the Genetic Algorithm:***

- ○ Initialization: An arbitrary starting population of solutions is created.
- ○ Evaluation: Every solution’s fitness (i.e., how well the corresponding model performs) is evaluated.
- ○ Selection: Parents preferably choose fitting solutions to become parents.
- ○ Crossover: New offspring solutions are created by combining elements of parent solutions.
- ○ Mutation: Offspring are given random modifications to preserve variation.
- ○ Replacement: The younger generation replaces the present population.
- ○ Termination: Until a stopping criteria is satisfied (i.e., maximum generations or convergence), steps 2-6 are repeated.

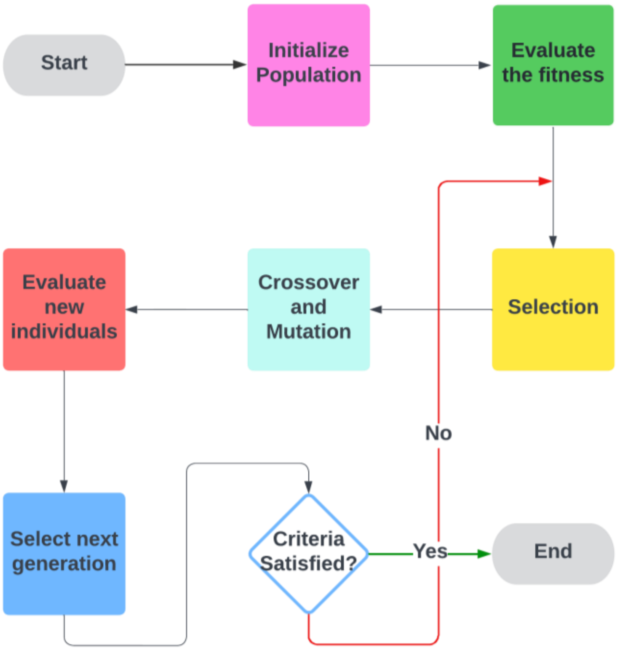

Figure 2. Genetic Algorithm Flowchart

- ***Hyperparameter Considerations:***
  - Population Size: The count of possible answers in every generation. While they raise computational cost, larger populations provide more diversity.
  - Crossover Rate: The chance of doing a crossover between parent solutions.
  - Mutation Rate: The chance of random mutations in offspring.
  - Selection Method: The method of selecting parents (e.g., tournament or roulette wheel choice).

- ***GA for Model Optimization:***

The hyperparameters of every machine learning model investigated in this work will be optimized using the genetic algorithm. The fitness function will be determined by the model's performance, mainly accuracy, on a validation set derived via cross validation. Through the iterative refinement of hyperparameter setups, we seek to ascertain parameters that enhance model performance while minimizing training time.

3-4-2- Grey Wolf Optimization [53, 54]

Inspired by the social hierarchy and hunting techniques of grey wolves, the Grey Wolf Optimizer was utilized for hyperparameter optimization in this work. GWO models the cooperative behavior of wolves in a pack using three best solutions—alpha, beta, and delta wolves. Based on these leaders, the surviving wolves change their stances to come to the best answer. Although its effectiveness may rely on suitable parameter choices, GWO's hierarchical search mechanism presents different benefits for hyperparameter optimization.

- ***Steps of Grey Wolf Optimization:***
  - Initialization: Random initialization of a population of grey wolves (potential solutions) inside the hyperparameter search range.
  - Hierarchy Establishment: Based on their fitness (model performance on the validation set), the three best solutions, alpha, beta, and delta, are found.
  - Update Positions: Using equations like distance computations and random vectors, other wolves in the pack change their places depending on the positions of the leader wolves.
  - Fitness Evaluation: Every wolf's (i.e., the performance of the corresponding model) fitness is evaluated.
  - Update Leadership: A wolf replaces one of the present leaders if its new position increases its fitness more than that of the others.
  - Termination: Until a stopping criterion is satisfied (that is, maximum iterations attained), steps 2–5 are repeated.

- ***Mathematical Formulation:***

The position update equations in GWO are:

$$D_\alpha = |C_1 \cdot X_\alpha - X| \tag{19}$$

$$D_\beta = |C_2 \cdot X_\beta - X| \tag{20}$$

$$D_\delta = |C_3 \cdot X_\delta - X| \tag{21}$$

$$X_{new} = X_\alpha - A \cdot D_\alpha + B \cdot D_\beta + C \cdot D_\delta \tag{22}$$

where: Distance vectors from the current wolf's position ((X)) to the alpha, beta, and delta wolves accordingly are (D_α), (D_β), and (D_δ); The alpha, beta, and delta wolves have positions (X_α), (X_β), and (X_δ); Coefficient vectors (C1), (C2), and (C3) progressively drop over iterations; (A), (B), and (C) are random vectors.

- ***Hyperparameters Considerations:***
  - Population Size: The amount of wolves in the pack.
  - Coefficients ((C_1), (C_2), (C_3)): Controls the exploration and exploitation balance.
  - Maximum Number of Iterations: Manages the length of the optimization process.

- ***GWO for Model Optimization:***

Each machine learning model under consideration in the study will have hyperparameter optimization using GWO. The fitness of a wolf is determined by the performance (accuracy) of its corresponding model on the validation set. Through the iterative process of hunting and leadership updates, GWO seeks to discover hyperparameter configurations that lead to superior model performance (see Figure 3).

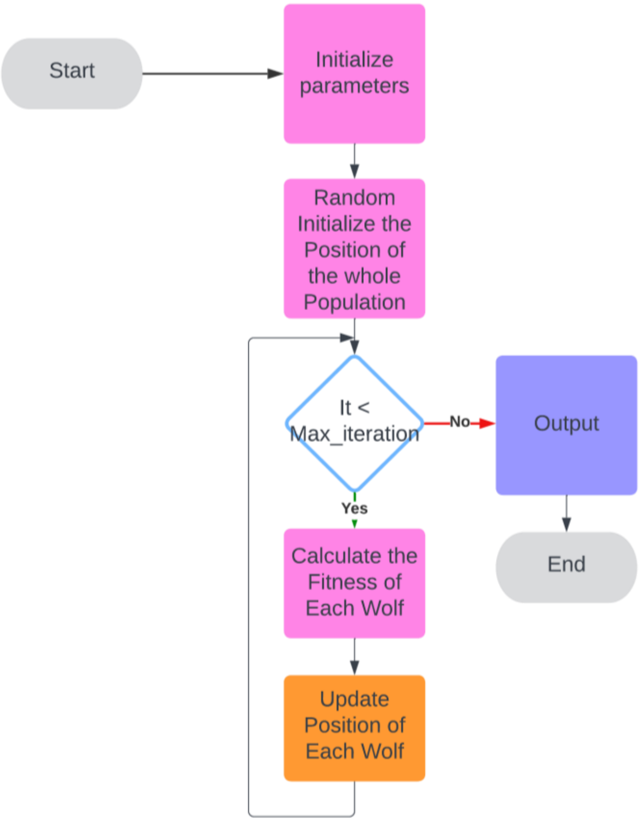
Start
Initialize parameters
Random Initialize the Position of the whole Population
It < Max_iteration
No
Output
End
Yes
Calculate the Fitness of Each Wolf
Update Position of Each Wolf

Figure 3. Grey Wolf Optimization Flowchart

3-4-3- Moth Flame Optimization [55]

Inspired by moth navigation patterns, Moth-Flame Optimization was added to vary hyperparameter optimization techniques. With "moths" standing in for possible solutions and "flames" acting as reference points, MFO replicates the moths' spiraling towards light source behavior. MFO strikes a mix of exploration and exploitation in the search space by dynamically varying the flames throughout repetitions. MFO is good at avoiding local optima, but its performance could be sensitive to the complexity of the optimization issue and the control parameter selection (see Figure 4).

- ***Steps of Moth-Flame Optimization:***
  - Initialization: Random initialization of a population of moths (potential solutions) within the hyperparameter search space. Additionally, starting with the same locations is a set of flames equal in count to the moths starting out.
  - Fitness Evaluation: On the validation set, assess every month's (that is, the matching model's) fitness.
  - Flame Sorting: Sort the flames in ascending order according to their fitness values.
  - Moth Updates: Every moth uses the logarithmic spiral equation to change its location in relation to a matching flame.
  - Flame Update: Change the flames, keeping in mind the best solutions discovered in the current iteration by moths.
  - Flame Reduction: Based on the current iteration and maximum number of iterations, lower the flames' count.
  - Termination: Continue steps 2–6 until a stopping criterion is satisfied—e.g., maximum number of iterations.

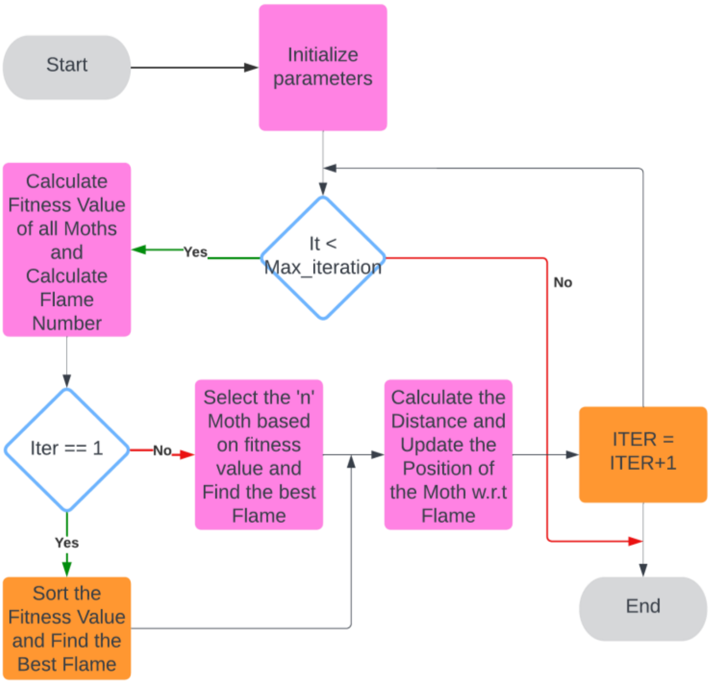
Start
Initialize parameters
It < Max_iteration
Yes
No
Calculate Fitness Value of all Moths and Calculate Flame Number
Iter == 1
No
Yes
Select the 'n' Moth based on fitness value and Find the best Flame
Calculate the Distance and Update the Position of the Moth w.r.t Flame
ITER = ITER+1
Sort the Fitness Value and Find the Best Flame
End

Figure 4. **Moth-Flame Optimization Flowchart**

- ***Mathematical Formulation:***
  - Moth Position Update: The position of a moth is updated based on the position of a corresponding flame using a logarithmic spiral equation:

$$S(\overrightarrow{M_\iota}, \overrightarrow{F_j}) = D_i \cdot e^{bt} \cdot cos(2\pi t) + \overrightarrow{F_j} \quad (23)$$

where: *S* is the updated position of moth *I*; $\vec{M}_i$ is the current position of moth *I*; $\vec{F}_j$ is the position of flame *j*; D_i is the distance between moth $_i$ and flame *j*; *b* is a constant for defining the shape of the logarithmic spiral; *t* is a random number in [-1, 1].

  - Flame Number Adaptation: The number of flames decreases over iterations according to the following equation:

$$\text{Flame No.} = \text{round}\left(N - l \cdot \frac{N-1}{T}\right) \quad (24)$$

Where: (N) is the initial number of flames; (l) is the current iteration; (T) is the maximum number of iterations.

- ***Hyperparameter Considerations:***
  - Population Size: Number of moths in the population.
  - Logarithmic Spiral Constant (b): Controls the shape of the spiral movement.
  - Maximum Number of Iterations: Controls the duration of the optimization process.

- ***MFO for Model Optimization:***

MFO will be used to maximize the hyperparameter values of every machine learning model under evaluation in the research. The performance (accuracy) of the associated model on the validation set defines the fitness of a moth. MFO seeks hyperparameter configurations that result in improved model performance by means of the iterative moth migration towards flames and adaptive reduction of flames.

3-4-4- Particle Swarm Optimization [56, 57]

Inspired by swarms' collective behavior, particle swarm optimization was used to maximize hyperparameters in machine learning systems. As "particles" that negotiate the search space under the impact of their personal best position and the global best position of the swarm, PSO offers possible solutions. Particularly in high-dimensional environments, PSO is efficient for hyperparameter optimization because of this dynamic balance between exploration and exploitation. On the other hand, PSO could converge too soon if it is not adequately calibrated (see Figure 5).

- ***Steps of Particle Swarm Optimization:***
  - Initialisation: Initialise a swarm of particles randomly inside the hyperparameter space with velocities and starting places.
  - Fitness Evaluation: Analyse every particle's fitness in relation to the performance of the matching model (e.g., accuracy on the validation set).
  - Update Personal Best: Update the pbest of a particle if its current position improves its fit over its best past position.
  - Update Global Best: Update the gbest if any particle settles better than the present best.
  - Update Velocity and Position: Update the velocity and position of every particle depending on random elements, its distance from pbest and gbest, and its present speed.
  - Termination: Until a stopping requirement is satisfied (e.g., maximum iterations reached), iterate stages 2–5.

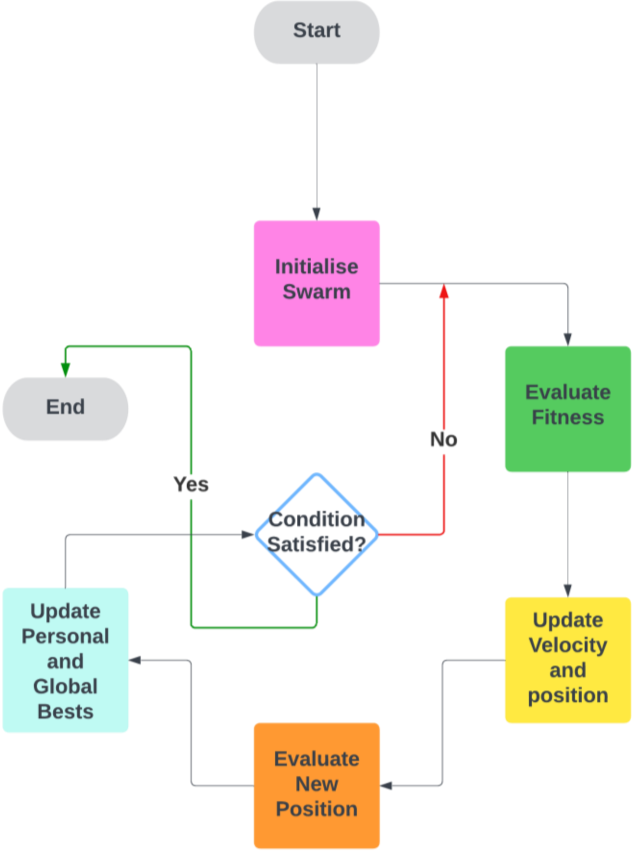

Figure 5. Particle Swarm Optimization Flowchart

- ***Mathematical Formulation:***

  The fundamental equations guiding particle velocity and position are

$$velocity_{ij}^{t+1} = \omega \cdot velocity_{ij}^{t} + c_1 \cdot r_1 \cdot \left(pbest_{ij} - position_{ij}^{t}\right) + c_2 \cdot r_2 \cdot \left(gbest_j - position_{ij}^{t}\right) \tag{23}$$

$$position_{ij}^{t+1} = position_{ij}^{t} + velocity_{ij}^{t+1} \tag{24}$$

where: ($velocity^t_{ij}$) and ($position^t_{ij}$) are the velocity and position of the particle (i) in dimension (j) at iteration (t); (ω) is the inertia weight, which controls the influence of the previous velocity; Acceleration coefficients (c_1) and (c_2) balance the effect of global best positions and personal ones; (r_1) and (r_2) are random numbers between 0 and 1.

- ***Hyperparameter Considerations:***
  - Inertia Weight (ω): Controls the balance between exploration and exploitation.
  - Acceleration Coefficients ((c_1) and (c_2)): Balance the influence of personal best and global best positions.
  - Swarm Size: The number of particles in the swarm.
  - Maximum Velocity Limits: Limitations on particle movement help to prevent overshoot of the ideal solution.

- ***PSO for Model Optimization:***

Each machine learning model's hyperparameters will be optimised using PSO. The performance of the relevant model on the validation set determines the fitness of a particle; mostly, accuracy is the metric used in this regard. PSO seeks hyperparameter settings that improve model performance by means of iterative swarm behavior.

3-4-5- Whale Optimization Algorithm [58]

Inspired by humpback whale hunting behavior, the whale optimization algorithm was applied to maximize hyperparameters for models of lung cancer prediction. By balancing exploration and exploitation, WOA replicates the bubble-net feeding method of the whales. Represented as "whales," candidate solutions change their locations in the search space depending on mathematical models of encircling prey and spiral motions. Since WOA can adaptively enhance solutions, it is a strong choice for hyperparameter tuning; but its convergence may depend on careful parameter change.

- ***Steps of Whale Optimization Algorithm:***
    - Initialisation: Randomly start a population of whales (potential answers) inside the hyperparameter search range.
    - Fitness Evaluation: Based on the performance of the related model—e.g., validation set accuracy—evaluate each whale's degree of fitness.
    - Update Best Position: Identify the whale having the highest fitness value (global best).
    - Exploration Phase (Shrinking Encircling Mechanism): Using a shrinking encircling mechanism, where the distance between the whale and the prey (global best solution) lowers over iterations, update the position of every whale. This stimulates whales' bubble-net attacking action.
    - Exploitation Phase (Spiral Updating Position): Using a spiral equation that mimics the helix-shaped motion of humpback whales as they approach their prey, update the location of every whale.
    - Termination: Continue steps 2-5 until a stopping criterion is met (e.g., maximum number of iterations).

- ***Mathematical Formulation:***
    - Shrinking Encircling Mechanism:

$$\vec{D} = \left|\vec{C} \cdot \overrightarrow{X^*} - \vec{X}\right| \tag{27}$$

$$\vec{X}(t+1) = \overrightarrow{X^*} - \vec{A} \cdot \vec{D} \tag{28}$$

where: The distance vector ($\vec{D}$) between the best whale position ($\overrightarrow{X^*}$) and the current whale position ($\vec{X}$); ($\vec{C}$) and ($\vec{A}$) are coefficient vectors that are updated in each iteration.

- Spiral Updating Position:

$$\overrightarrow{D'} = \left|\overrightarrow{X^*} - \vec{X}\right| \tag{29}$$

$$\vec{X}(t+1) = \overrightarrow{D'} \cdot e^{bl} \cdot cos(2\pi l) + \overrightarrow{X^*} \tag{30}$$

where: ($\overrightarrow{D'}$) is the distance vector between the current whale position and the best whale position; (b) is a constant for defining the shape of the logarithmic spiral; (l) is a random number in [-1, 1].

- ***Hyperparameter Considerations:***
    - Population Size: Number of whales in the population.
    - Coefficients (($\vec{A}$) and ($\vec{B}$)): Influence the exploration and exploitation balance.
    - Maximum Number of Iterations: Controls the duration of the optimization cycle.

- ***WOA for Model Optimization:***

WOA will be used to maximize the hyperparameter values of every machine learning model under evaluation in the research. Whale fitness is found by the performance (accuracy) of its matching model on the validation set. WOA seeks to find hyper parameter combinations that result in improved model performance by iteratively changing whale locations via exploration and exploitation.

3-5- Novel Hybrid Algorithm

The GWO-WOA hybrid algorithm was developed to maximize the XGBoost model for lung cancer prediction. This method combines hyperparameter adjustment with feature selection with a final ensemble voting system to improve recall and accuracy while preserving computational economy.

3-5-1- Feature Selection using GWO

Features were selected using the Grey Wolf Optimizer by simulating grey wolf hierarchical hunting behavior. Four hierarchical roles define this process: alpha, beta, delta, and omega.

- ***Steps:***
  - Initialisation: Randomly generate a population of candidate solutions (grey wolves), where every solution is expressed as a binary string denoting either selected (1) or non-selected (0) features.
  - Fitness Evaluation: Evaluate every option according to the classification performance that is, accuracy of the XGBoost model trained with the chosen features.
  - Update Positions: Simulating the cooperative hunting approach and guiding the optimization process with the positions of the alpha, beta, and delta wolves helps to update candidate positions.
  - Iterate: Over a certain number of cycles or until convergence conditions are satisfied, repeat the evaluation and position update process.
  - Select Features: The best-performing solution (the binary string of the alpha wolf) represents the optimal set of selected features.

3-5-2- Hyperparameter Tuning Using WOA

Inspired by humpback whale bubble-net hunting, the Whale Optimization Algorithm was applied to maximize XGBoost model hyperparameters. Phases of this process involve prey seeking, spiral position updates, and encircling prey.

- ***Steps:***
  - Initialisation: Create a population of candidate solutions (whales) randomly, where every solution corresponds to an XGBoost model set of hyperparameters.
  - Fitness Evaluation: Using the relevant hyperparameter setup, evaluate every candidate using a fitness function such as classification accuracy.
  - Update Positions: Using circular and spiral movement techniques, change potential locations in the search space to investigate and take advantage of best areas.
  - Iterate: Until the convergence conditions are met, repeat the procedure of evaluation and position updating.
  - Select Hyperparameters: The hyperparameters of the best-performing whale are chosen as the optimal configuration.

3-5-3- Training and Ensemble

Following optimization of features and hyperparameters using GWO and WOA, the XGBoost model was trained. Additionally, an ensemble model was constructed to further improve predictive performance.

- ***Steps:***
  - Train XGBoost Model: Use the selected features from GWO and optimised hyperparameters from WOA to train the XGBoost model.
  - Train Other Classifiers: With the same chosen features, train Random Forest, SVC, and Logistic Regression.
  - Ensemble Voting: Use a soft voting mechanism to enhance robustness by combining predictions from the XGBoost model and other classifiers.
  - Evaluate Ensemble Model: On the validation set, evaluate the ensemble model with a focus towards accuracy, recall, and computational efficiency.

While the ensemble voting system ensures strong and accurate lung cancer prediction, our hybrid method uses the exploring and exploiting powers of GWO and WOA to maximize feature selection and hyperparameter tweaking.

3-6- Performance Evaluation

Several criteria were applied to assess the models comprehensively, each offering information on multiple aspects of their predictive power.

3-6-1- Stratified Cross-Validation [59]

Stratified five-fold cross-validation assured consistent and exhaustive performance evaluation. This method divides the dataset into five folds while preserving the original class distribution. Training used four folds; each fold acted as the validation set once. The process was carried out five times, and the final performance evaluations were averaged over all folds to ensure a robust and objective assessment.

3-6-2- Metrics

The following evaluation metrics were used to comprehensively assess the models' performance:

1. ***Accuracy:*** Represents the proportion of correct predictions relative to the total number of predictions. It offers a general measure of model performance but might not fairly represent performance in imbalanced data [60].

$$Accuracy = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{31}$$

2. ***Recall (Sensitivity, True Positive Rate):*** Calculates the true percentage of actual positive cases the model accurately detected. Recall is particularly important in minimising false negatives, which is critical in medical diagnoses [61].

$$Recall = \frac{\text{True Positives}}{\text{True Positives+False Negatives}} \tag{32}$$

3. ***Precision (Positive Predictive Value):*** Reflects the proportion of true positive predictions. High precision is essential for ensuring the reliability of predicted lung cancer cases [61].

$$Precision = \frac{\text{True Positives}}{\text{True Positives+False Positives}} \tag{33}$$

4. ***F1-Score:*** Combines precision and recall into a single metric, representing their harmonic mean. It is particularly useful when false positives and false negatives carry similar significance [62].

$$F1 - Score = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision+Recall}} \tag{34}$$

5. ***ROC-AUC (Receiver Operating Characteristic-Area under Curve):*** Evaluates the model's ability to distinguish between classes at varying classification thresholds. A higher AUC indicates better overall performance in differentiating cancer and non-cancer cases [63].

In this study, Accuracy, Recall, Precision, F1-Score, and ROC-AUC were prioritised for model comparison because they jointly capture both overall correctness and the model's diagnostic balance. Recall (also known as Sensitivity) directly measures the model's ability to detect true lung-cancer cases, while Precision quantifies the reliability of those detection. F1-Score balances these two competing aspects, and ROC-AUC summarizes discrimination performance across decision thresholds. Specificity was not reported separately, as it is mathematically the complement of false-positive rate in binary classification and therefore redundant once ROC-AUC is included. Emphasising Recall and ROC-AUC is particularly important in medical screening, where minimising false negatives carries greater clinical value than marginal improvements in overall accuracy.

Although accuracy gives a broad picture, recall, precision, and F1-score give a better knowledge of the model's ability in spotting and separating between cancer and non-cancer instances. As a balanced indicator, the F1-score is especially useful in situations where recall and precision both matters. By evaluating the model's capacity to discriminate between classes across a spectrum of thresholds, ROC-AUC complements these measures and provides a more complete assessment of its diagnostic power.

4- Results

This section presents the outcomes of a systematic, staged approach for predicting lung cancer using a synthetic dataset. Three distinct strategies were employed—progressing from baseline machine learning models to nature-inspired algorithm optimizations and culminating in a hybrid ensemble. Each stage's results are summarized below, illustrating how incremental refinements improved predictive accuracy and maintained computational efficiency.

4-1- Base Machine Learning Models

By use of nature-inspired algorithms and possible integration into a hybrid NIA framework, the evaluation of baseline machine learning models sought to discover a solid fundamental model for later optimization. With hyperparameter tuning using GridSearchCV to guarantee competitive performance on the synthetic dataset, the study evaluated several conventional classifiers, including decision trees, ensemble approaches, and neural networks. These studies found models that balance predicted accuracy and computational efficiency, therefore establishing a strong basis for lung cancer diagnosis. The results guided the choice of candidate models for next optimization and ensemble construction.

4-1-1- Performance Analysis of Baseline Models

The performance of the baseline models was evaluated using key classification metrics, including accuracy, precision, recall, F1 score, and ROC-AUC. Table 1 provides a summary of these metrics for each classifier:

Table 1. Comparison of Models Performance

Model	Accuracy	Precision	Recall	F1 Score	ROC AUC
AdaBoost	98.95%	98.95%	98.95%	98.95%	99.82%
Bagging Classifier	99.06%	99.07%	99.06%	99.06%	99.40%
BernoulliNB	88.23%	90.65%	88.23%	88.73%	97.27%
Decision Tree	98.33%	98.33%	98.33%	98.33%	97.84%
ExtraTrees	98.97%	98.98%	98.97%	98.97%	99.83%
GradientBoosting	98.55%	98.57%	98.55%	98.55%	99.62%
KNeighbors	98.01%	98.03%	98.01%	98.01%	99.56%
Logistic Regression	97.63%	97.66%	97.63%	97.63%	99.12%
MLPClassifier	97.40%	97.43%	97.40%	97.40%	99.18%
Random Forest	98.75%	98.76%	98.75%	98.75%	99.68%
SVC	97.95%	97.97%	97.95%	97.95%	99.28%
XGBoost	99.15%	99.16%	99.15%	99.15%	99.85%

Most baseline classifiers showed strong performance over the evaluated parameters, as compiled in Table 1 and shown in Figure 6. XGBoost was recognised as the most effective model, reaching an accuracy of 99.15% and consistently high metrics in precision, recall, F1 score, and ROC-AUC. XGBoost is a strong candidate for additional improvement using nature-inspired algorithms and a hybrid framework since it can harmonise prediction accuracy with reasonable computing capacity.

Some models, such as Bernoulli Naive Bayes, demonstrated fast training times but lower accuracy (88.23%), highlighting the trade-off between computational efficiency and predictive performance. Other ensemble models, including Bagging, Random Forest, and Extra Trees, delivered near-top accuracy (≥98.75%) but required higher computational resources. As illustrated in Figure 7, XGBoost strikes a favourable balance between computational cost and reliability, making it suitable for applications requiring both high precision and efficiency.

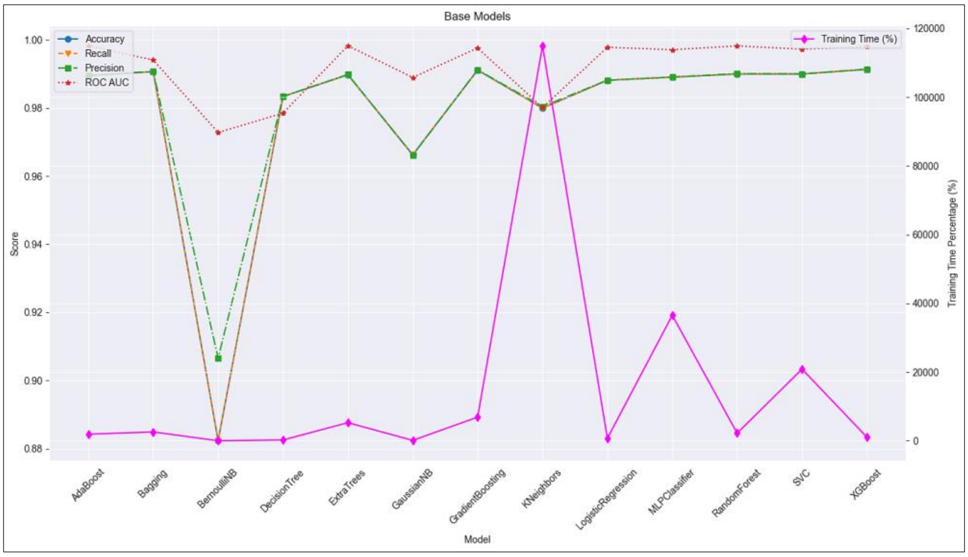

Figure 6. Base Model Score Plot

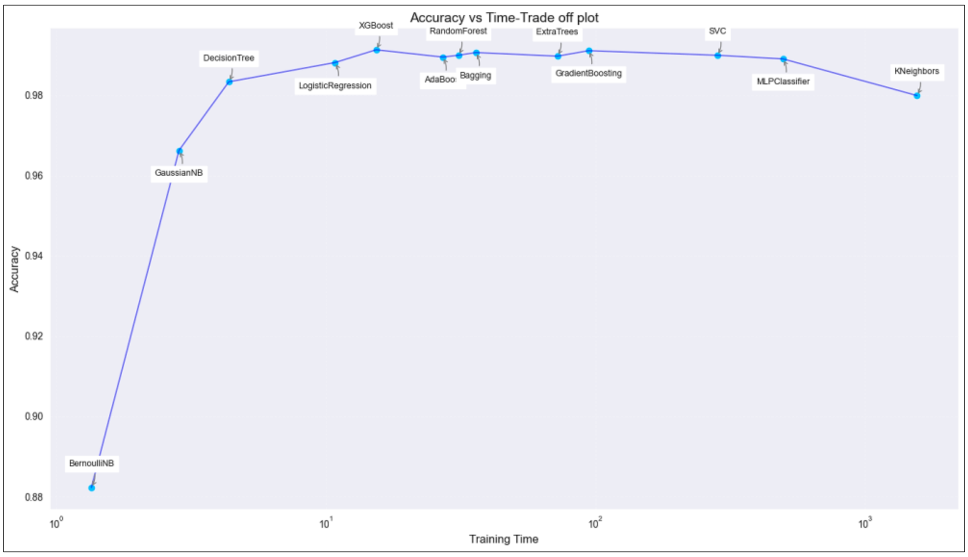

Figure 7. **Accuracy vs. Time-Trade Off (Base Models)**

Compared with findings from recent structured-data studies, these baselines results are consistent with previously observed trends. Chen (2024) [9] reported XGBoost-based inclusive models with AUC ≈ 0.89 and accuracy ≈ 0.89, while Su (2025) [13] achieved AUC ≈ 0.86 in large-scale clinical dataset. The baseline XGBoost model here achieved similar relative behavior, showing strong discriminative capacity under synthetic conditions. The comparatively higher accuracy reflects the cleaner, fully structured nature of synthetic data and should therefore be viewed as indicative rather than definitive.

These results overall support the effectiveness of ensemble-based approaches and show a clear road for later improvements. Especially XGBoost, models that mix great accuracy with reasonable training times will form the basis for the following phases of this effort, involving advanced hyperparameter adjustment using NIAs and the creation of a hybrid (ensemble) solution.

4-2- Comparison with NIA Models

The baseline evaluation demonstrates that ensemble and boosting algorithms perform robustly on structured synthetic EHR data. Their superior recall and AUC values suggest these models can reliably identify risk patterns when data are well-structured and noise-free. However, this should not be interpreted as guaranteed clinical reliability. In real-world datasets, inconsistencies in coding, incomplete medical histories, and diverse patient demographics often reduce performance. These results therefore illustrate the algorithmic potential under idealised conditions rather than a validated diagnostic capability.

This work uses several nature-inspired algorithms in the second phase to maximize the evaluated machine learning models from the baseline phase. These NIAs comprised the Genetic Algorithm, Grey Wolf Optimizer, Moth-Flame Optimization, Particle Swarm Optimization, and Whale Optimization Algorithm. Every method was applied to adjust hyperparameters in order to maximize prediction performance under control of computational overhead. Key ideas from every method are summarized here, with Figures 8 to 17 visualising their outcomes.

4-2-1- Genetic Algorithm Optimization

Consistent improvements in ensemble-based models like Extra Trees, XGBoost, and Random Forest across metrics including accuracy, precision, and recall were shown by using Genetic Algorithm (Figure 8). Minimal increases were shown by simpler models, including BernoulliNB, which emphasises the trade-off between prediction accuracy and processing economy. Training time studies (Figure 9) showed that although GA increases computing cost, it typically performs better than baseline hyperparameter tuning method.

4-2-2- Grey Wolf Optimization

With Random Forest, Extra Trees, and XGBoost obtaining accuracy rates in the 98–99% range (Figure 10), Grey Wolf optimizer adjustment strengthened the superiority of ensemble classifiers. Although other models, like KNeighbors and MLPClassifier, showed similar accuracy, they sometimes needed more computational resources. Top-performing models' ROC-AUC repeatedly topped 99%, highlighting GWO's efficiency in hyperparameter optimization. Training time analysis (Figure 11) showed that for some classifiers GWO could lower computational overhead while maintaining strong performance.

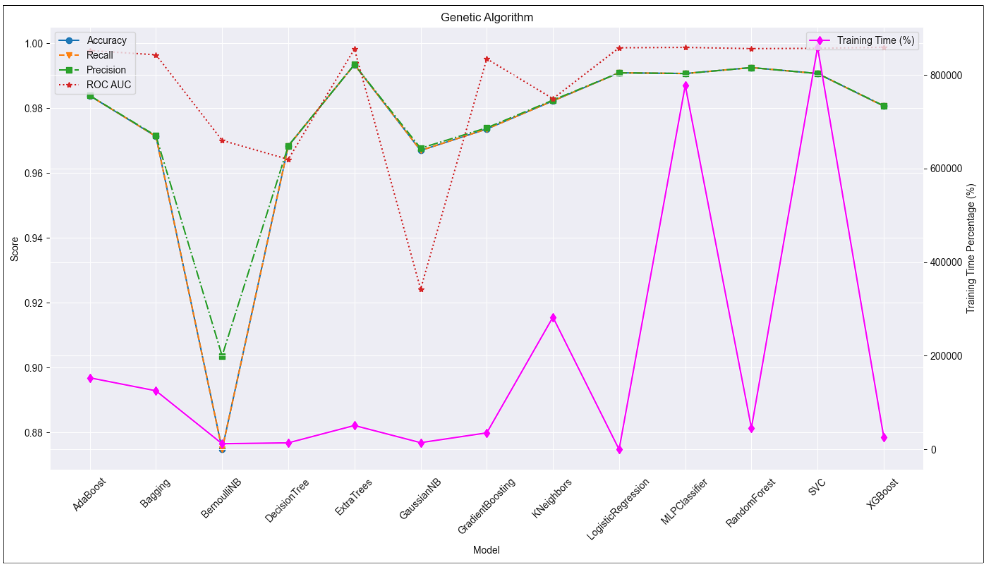

Figure 8. Genetic Performance Plot

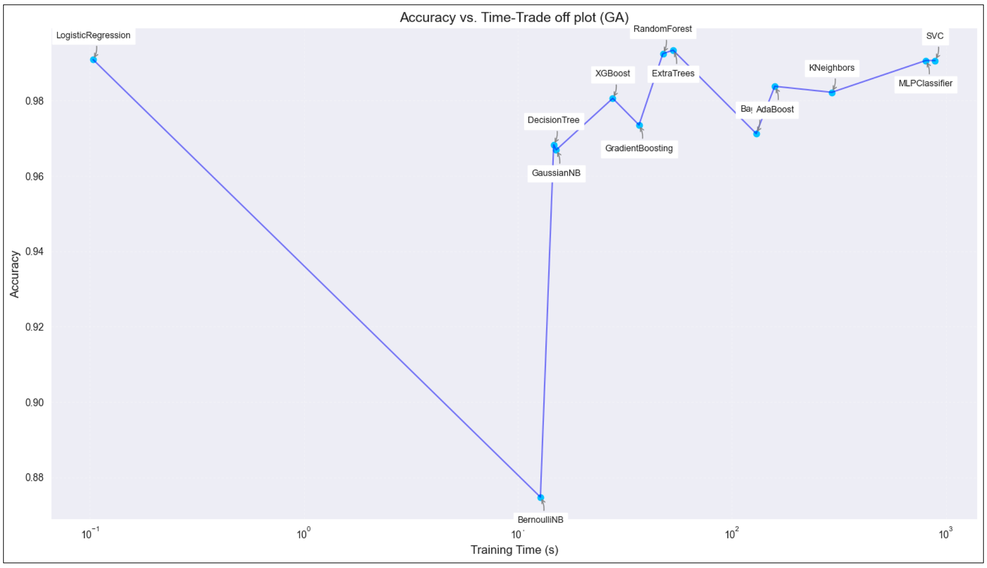

Figure 9. Accuracy vs Time-Trade Off (GA)

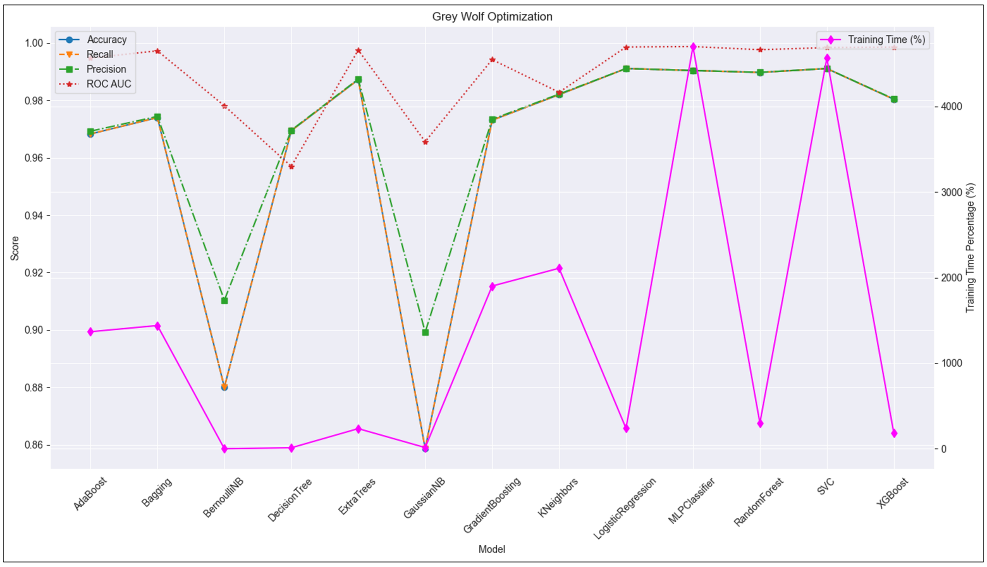

Figure 10. Grey Wolf Optimization Performance Plot

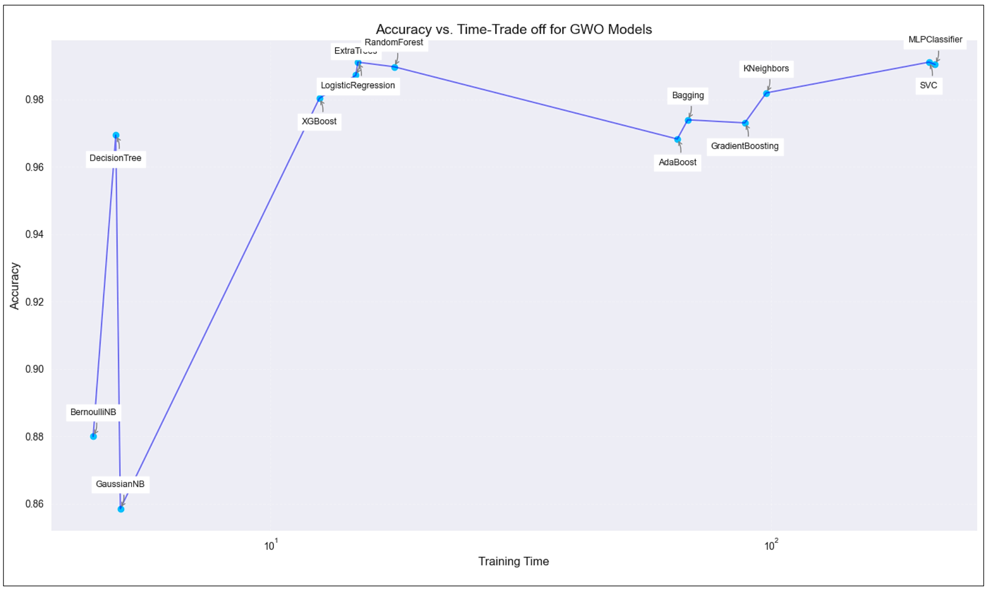

Figure 11. Accuracy vs Time-Trade Off (GWO)

4-2-3- Moth Flame Optimization

With both models attaining accuracy rates above 98% and strong ROC-AUC values, moth-flame optimization routinely improved the performance of XGBoost and Random Forest (Figure 12). MFO also greatly helped logistic regression since it provides a good mix between computational efficiency and accuracy. Although simpler models like BernoulliNB and GaussianNB stayed computationally efficient, their sub-90% accuracy made them less suitable for precision-critical uses like lung cancer diagnosis (Figure 13). Especially for ensemble techniques, these results show MFO's capacity to precisely adjust hyperparameters.

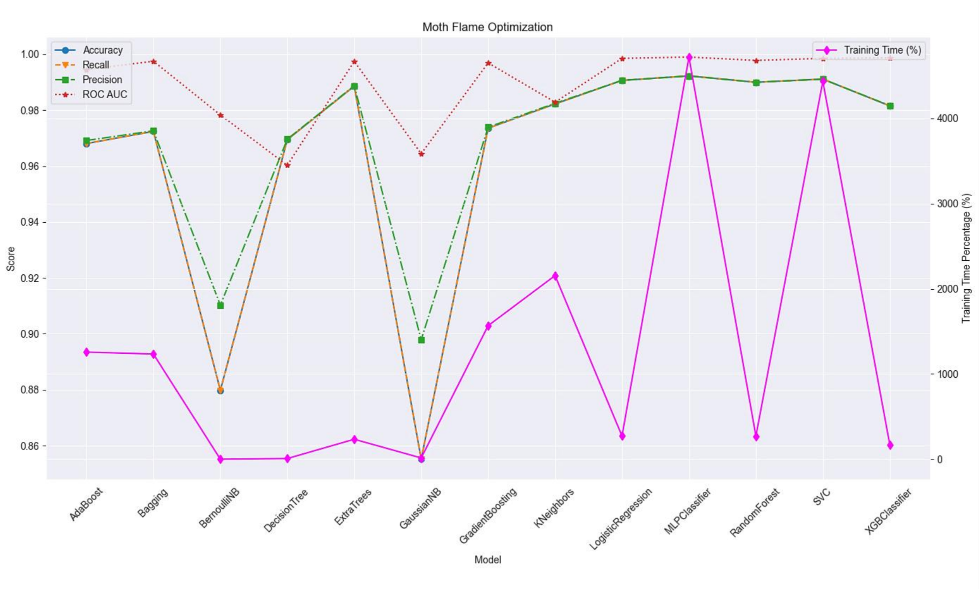

Figure 12. Moth-Flame Optimization Performance Plot

4-2-4- Particle Swarm Optimization

For high-performance models including Logistic Regression, XGBoost, Random Forest, and Gradient Boosting—all of which attained accuracy rates above 97%—Particle Swarm Optimization produced notable gains (Figure 14). Though their complicated hyperparameter setups caused more computational effort, bagging and AdaBoost performed equally. Though it needed more processing resources, MLPClassifier displayed competitive metrics (Figure 15). Although PSO sometimes gave accuracy top priority at the expense of efficiency, its capacity to investigate several parameter spaces proved useful.

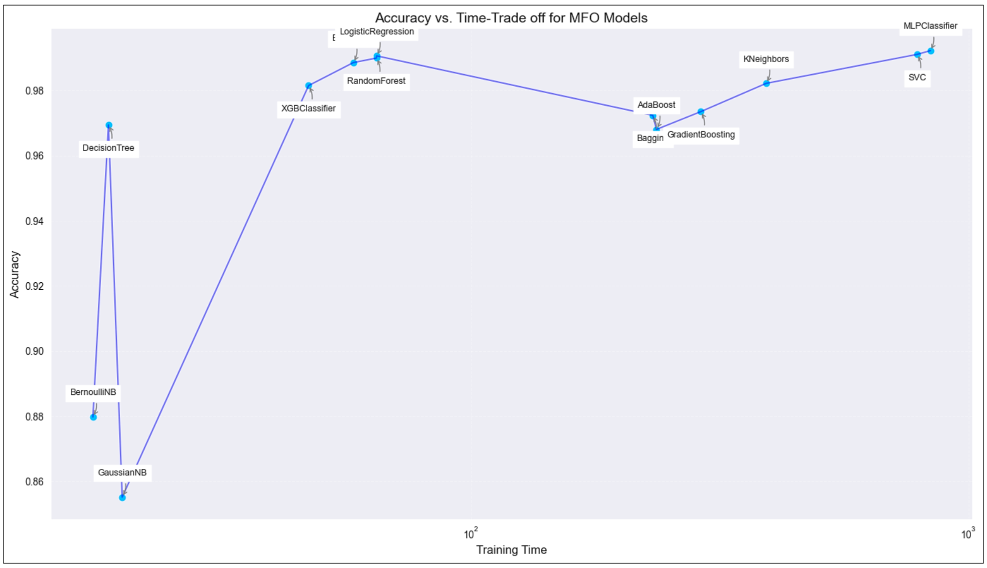

Figure 13. **Accuracy vs Time-Trade Off (MFO)**

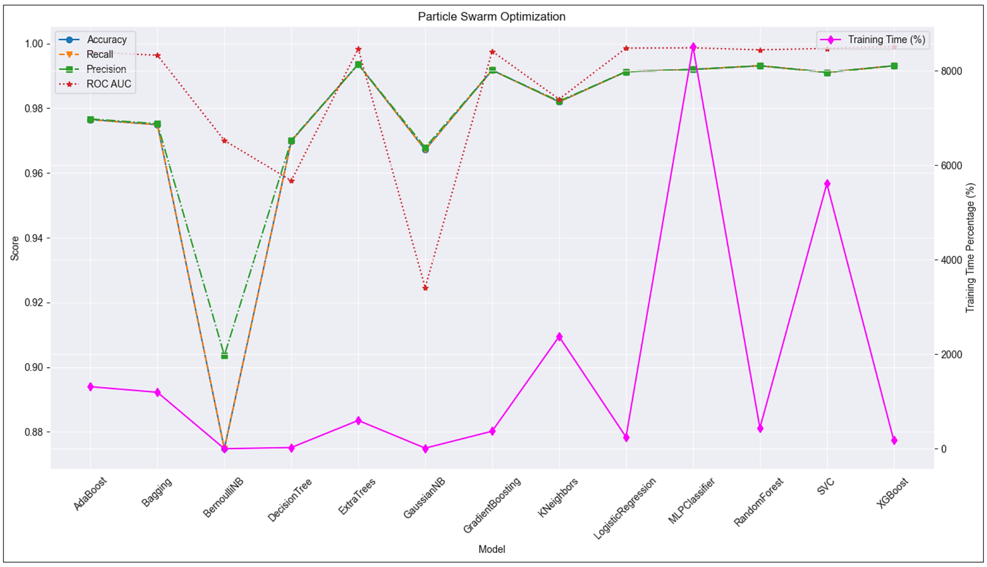

Figure 14. **Particle Swarm Optimization Performance Plot**

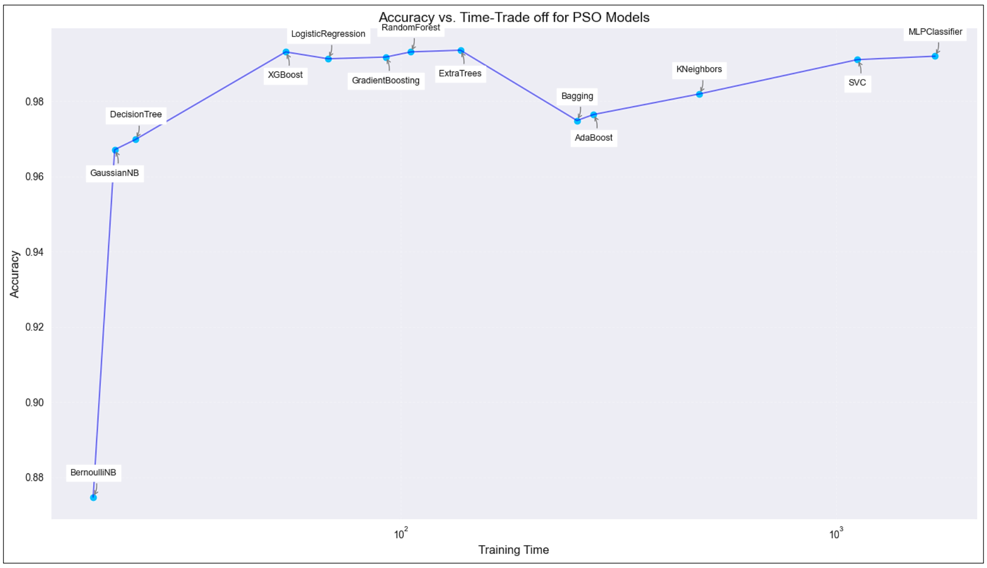

Figure 15. **Accuracy vs Time-Trade Off (PSO)**

4-2-5- Whale Optimization Algorithm

With accuracy rates exceeding 98–99% (Figure 16), The whale optimization algorithm routinely improved top-performing models, including Random Forest, XGBoost, and LogisticRegression. Training-time analysis showed that WOA sometimes preferred simpler ensemble techniques such as AdaBoost and Bagging, which are competitive in both accuracy and speed (Figure 17). More complicated models like MLPClassifier and SVC, on the other hand, albeit having excellent accuracy, paid more computational costs because of their hyperparameter spaces.

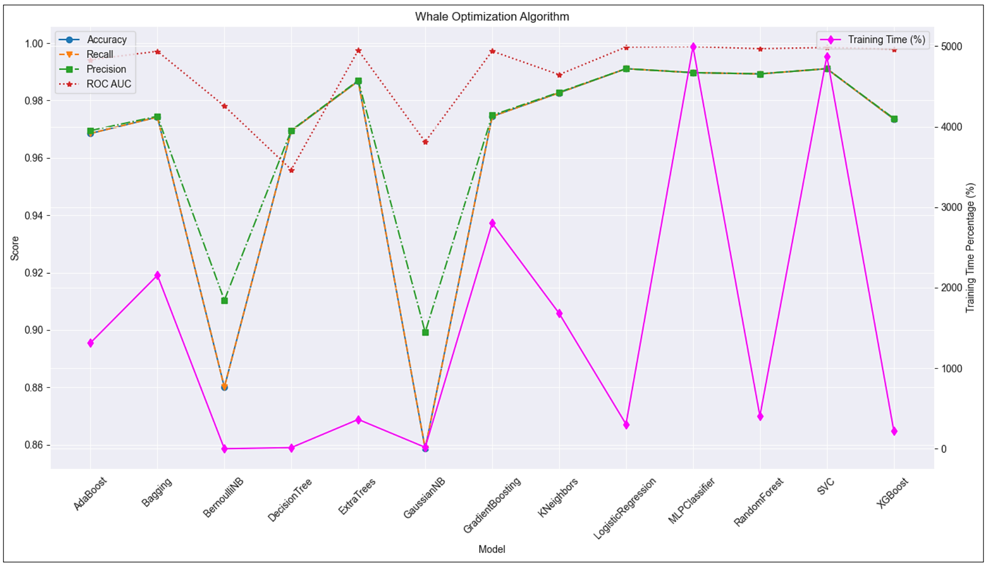

Figure 16. Whale Optimization Algorithm Performance Plot

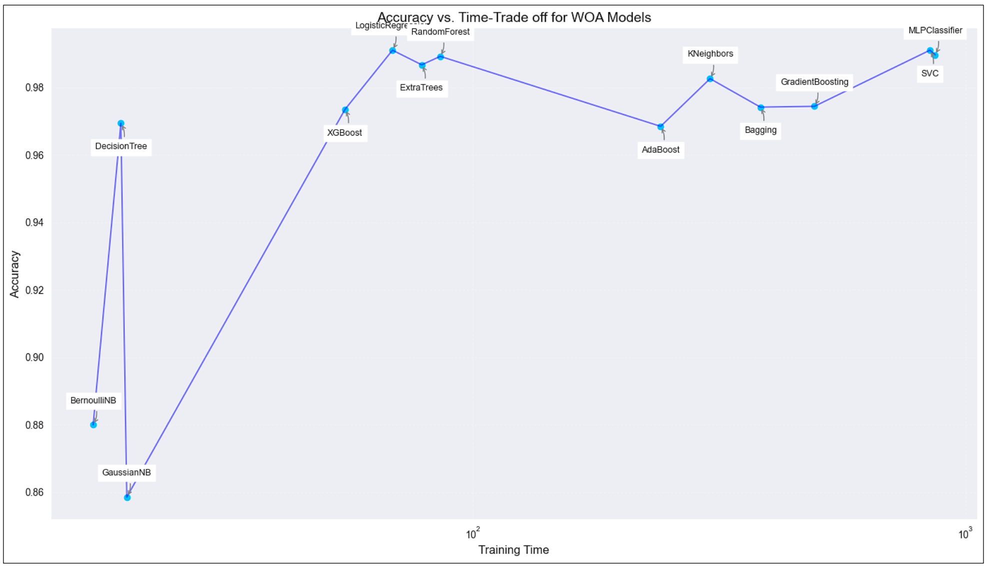

Figure 17. Accuracy vs Time-Trade Off (WOA)

4-2-6- Overall Observation

- Ensemble Models Excel: Random Forest, Extra Trees, and XGBoost are among the ensemble-based approaches that consistently produce good classification metrics across all NIAs, therefore supporting the idea that aggregating several weak learners can produce strong decision boundaries and low variation.
- Trade-Off Between Performance and Efficiency: Although other simpler models (e.g., BernoulliNB, GaussianNB) have low training times, their accuracy stays below 90%, which is less desired for important medical applications. Conversely, classifiers like SVC or MLPClassifier occasionally show almost-top performance but at a larger computational cost.
- Tuneable Gains from NIAs: Sometimes by also lowering training times, each metaheuristic effectively finds configurations that exceed the baseline in terms of accuracy or other measures. The particular increases depend on the method and classifier; hence, it emphasises the need of trying several NIAs to find the optimal hyperparameter values.

The application of metaheuristic optimization produced incremental gains in accuracy and recall, reflecting the capacity of nature-inspired search strategies to fine-tune parameter spaces efficiently. These results also underscore a trade-off between computational complexity and performance: optimization improved metrics modestly but increased processing time. This mirrors observations from other optimization-based studies, where metaheuristics are most beneficial when marginal improvements can translate into better stability or generalization. From a clinical standpoint, the modest numerical improvements are less critical than the demonstration that feature selection and tuning can enhance model reliability without altering interpretability.

The observed performance gains from metaheuristic optimization align with reports from recent evolutionary-algorithm research. Gupta et al. (2019) [30] demonstrated that Grey Wolf and related hybrid optimization methods could enhance feature selection while reducing computational load, achieving accuracies near 99 % on small imaging datasets. Similar though more moderate improvements are seen here, with accuracy and recall rising modestly after optimization but at higher runtime cost. This indicates that metaheuristics are valuable primarily for fine-tuning ensemble and tree-based learners when computational constraints allow. Across Studies, improvements of 0.3-0.6 % in accuracy or recall are typical and emphasize refinement rather than transformation of model behavior.

Overall, these nature-inspired optimization strategies yield improvements over conventional grid-based searches, especially for ensemble methods. The following section will delve into a hybrid approach, which combines the top-performing NIA-optimised models to further enhance accuracy while keeping computational demands in check.

4-3- Hybrid-Nature Inspired Ensemble Algorithm

Following analysis of high-performance models under both baseline and NIA-optimised layouts, a last hybrid ensemble was created to aggregate the complementary strengths of the top-performing models. Maintaining reasonable computing overhead, the aim was to reach or surpass the best classification metrics recorded.

4-3-1- Ensemble Model Performance

With an ROC-AUC of 0.9984, the hybrid ensemble produced accuracy, recall, precision, and F1 scores of 0.9925, as shown in Figure 18. These measures show how strong it is in separating negative from positive cases—lung cancer. Combining several learning techniques shows the advantages since the performance of the ensemble matched or exceeded the top individual models from both baseline and NIA-optimised configurations.

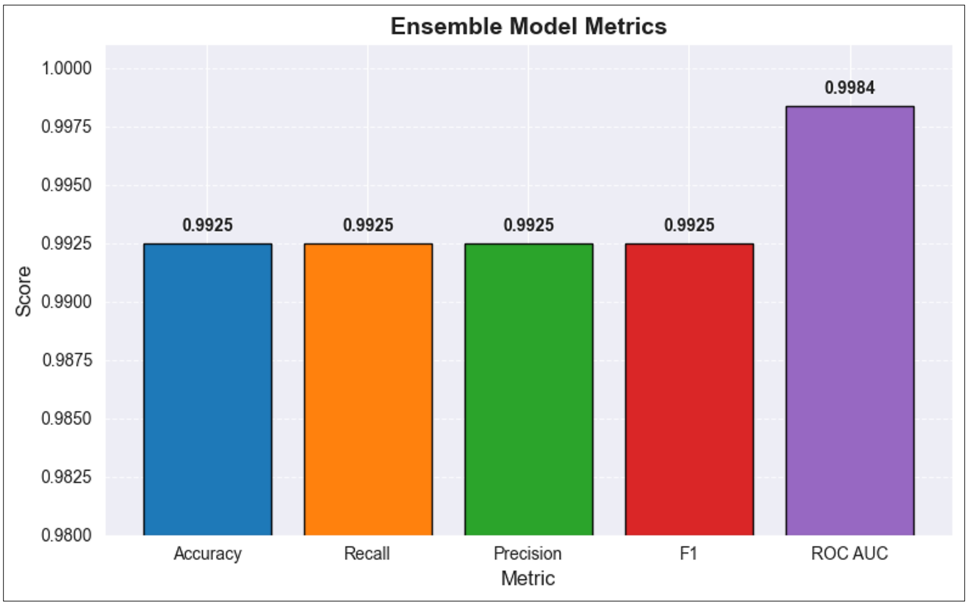

Figure 18. Ensemble Score Plot

4-3-2- Comparison with NIA-Optimised XGBoost

Figure 19 evaluates the performance of XGBoost both inside the hybrid ensemble and among several optimization approaches (Base, GA, GWO, MFO, PSO, WOA). Especially for recall and precision, the hybrid ensemble routinely outperformed several NIA-optimised variations by stabilising measures near or over the 0.99 threshold, even though some variants attained accuracy and ROC-AUC values exceeding 0.98. The group also showed a similar or better F1 score, therefore verifying its success in tackling class imbalance.

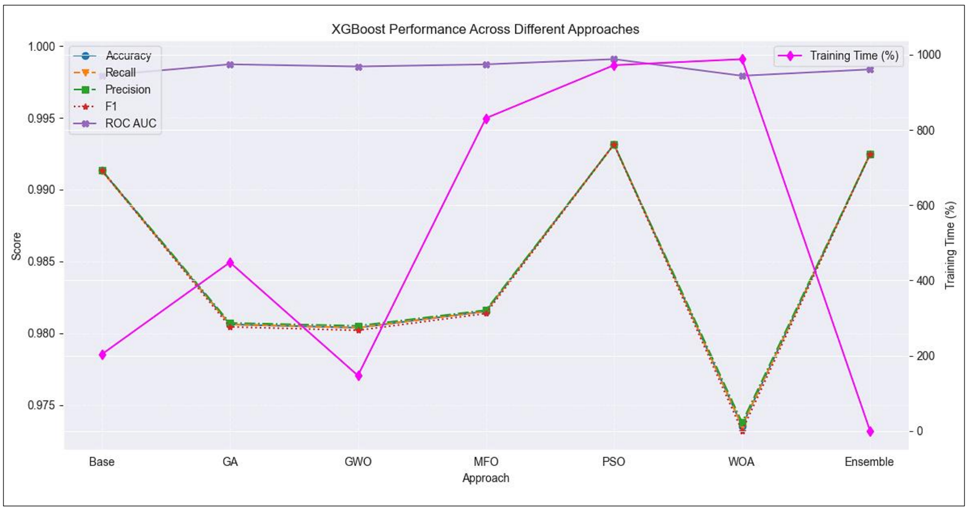

Figure 19. XGBoost Performance Across Different Approaches

4-3-3- Training Time Consideration

As seen in Figure 20, NIAs typically extend training timeframes even when they efficiently optimise hyperparameters to improve predictive performance. Even with great accuracy, techniques like PSO and WOA can inflate training overhead to more than three times that of base XGBoost. By contrast, the hybrid ensemble obtained a relative training time of 1.0—much below any single NIA-optimised model. By carefully choosing models that paired good accuracy with reasonable training costs, this efficiency was attained, hence reducing computing complexity and redundancy.

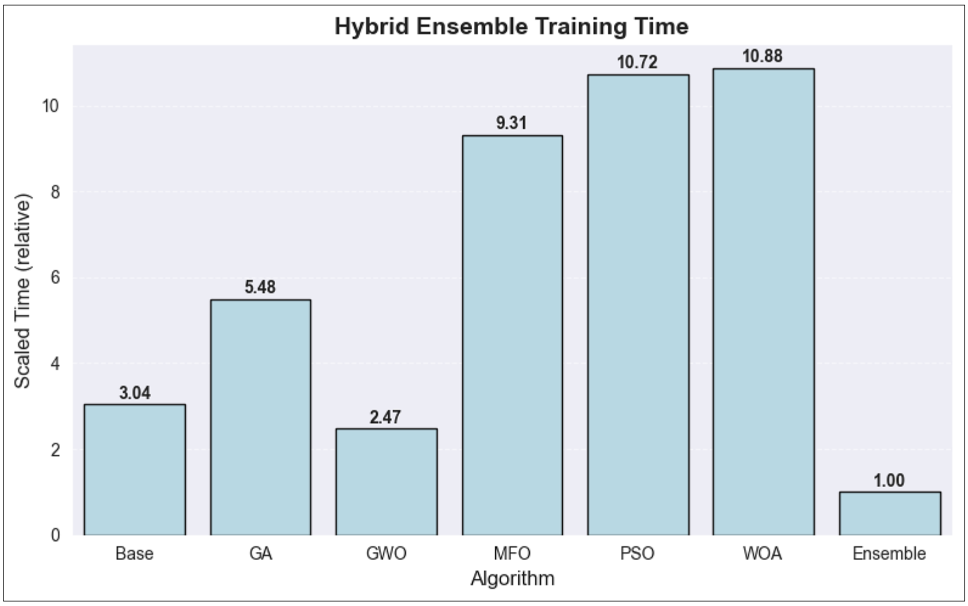

Figure 20. Hybrid-Ensemble Training Time Comparison

When compared with current EHR- and imaging-based lung-cancer studies, the proposed hybrid ensemble achieved performance levels consistent with contemporary benchmarks. Wang et al. (2024) [11] obtained an AUROC of 0.924 using transformer-based temporal model on UK primary-care data, and Su et al. (2025) [13] reported 0.860 with XGBoost on real-world Chinese EHRs. Chen et al. (2024) [9] achieved 0.89 AUC with an inclusive screening model, whereas imaging-centred deep-learning surveys such as Liz-López et al. (2025) [14] cite AUCs between 0.90 and 0.98 but with far greater computational expense. The present ensemble reached AUC ≈ 0.998 on synthetic data, demonstrating that under controlled conditions, lightweight optimization can approximate deep-learning-level discrimination. Nonetheless, these findings are exploratory: validation on clinical data is essential before any practical comparison can be confirmed.

4-3-4- Summary of Hybrid Ensemble Advantages

- High Predictive Power: Achieves near-uniform improvements across accuracy, recall, precision, F1, and ROC AUC.

- Efficient Training: Outperforms many NIA-optimised models in terms of speed, ensuring feasibility for large-scale or near-real-time applications.
- Robustness: Maintains top-tier performance despite dataset variations and parameter perturbations, indicating strong generalization potential.

By effectively leveraging the capabilities of baseline and NIA-optimised classifiers, the hybrid ensemble produces a final model with outstanding prediction accuracy and reasonable training costs. For medical chores like lung cancer screening, where both dependability and efficiency are critical, this method shows particularly great promise.

The hybrid GWO-WOA-XGBoost ensemble exhibited the highest overall performance, achieving near-perfect discrimination on synthetic data. This outcome supports the principle that layered optimization can capture subtle nonlinear relationships among health variables more effectively than single-stage tuning. Nevertheless, the performance must be viewed as an experimental ceiling: the absence of real-world variability, human input, and incomplete EHR entries likely inflate these metrics. The practical implication is that the framework could serve as an auxiliary analytical tool to flag high-risk profiles, prompting early screening, rather than as an autonomous diagnostic system.

4-4- Result Interpretation and Comparative Analysis

Overall, the progression from baseline to metaheuristic-optimised and hybrid models produced incremental but consistent improvements in predictive metrics. Ensemble and boosting architectures benefited most from parameter tuning while simpler learners plateaued early. Comparisons with recent literature indicate that structured-data approaches achieve AUCs in the 0.98-0.89 range, whereas deep-learning pipelines or imaging-based methods can exceed 0.90 AUC. The present study’s hybrid ensemble reached ≈ 0.998 on synthetic data, illustrating the discriminative potential of optimised traditional models when data noise is minimal. Importantly, the system is designed as an assistive framework that can operate in the background of EHR systems, flagging potential cases for clinical follow-up rather than replacing established diagnostic procedures. Future validation on federated or multi-institutional datasets, as envisioned by Rieke et al. (2020) [15], will be necessary to confirm generalizability and reliability in real-world settings. Overall, these interpretations emphasise that while performance values appear high, the real contribution lies in demonstrating how lightweight, interpretable optimization frameworks can complement—rather than replace—traditional screening workflows.

4-5- Feature Importance, Explainability, and Statistical Confidence

4-5-1- Feature Importance and Interpretability Analysis

SHAP analysis of the hybrid GWO-WOA-XGBoost ensemble (Figure 21) showed that only a small subset of variables accounted for most of the predictive variance.

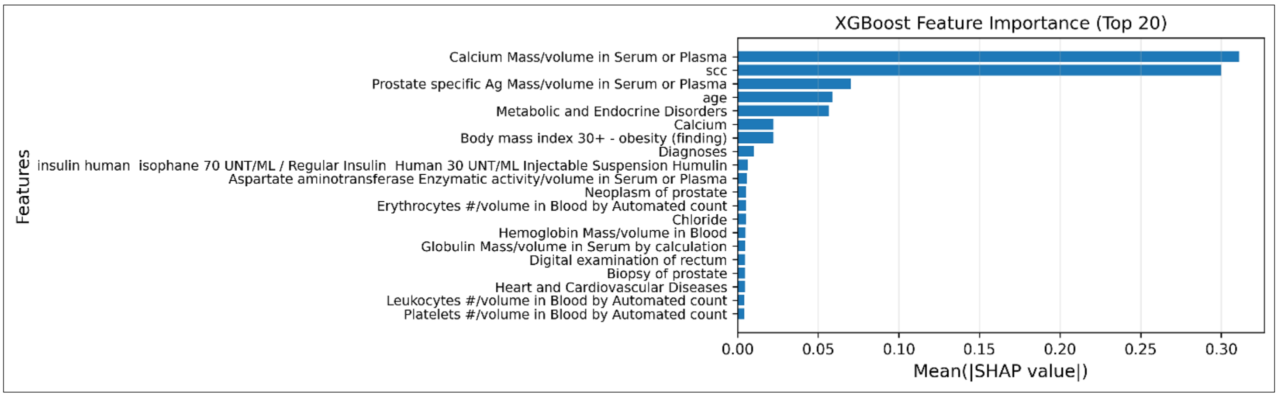

Figure 21. Feature Importance Plot

The SCC tumor marker exhibited the highest mean absolute SHAP value, emerging as the strongest positive contributors to predicted lung-cancer risk. Moderate contributions arose from metabolic and endocrine disorders, advanced age, and obesity (BMI ≥ 30), confirming that systemic metabolic dysfunction and ageing were key background risks within the model. Secondary but consistent effects were observed for liver-function enzymes (AST, ALT, ALP) and lipid measures (LDL, HDL, total cholesterol), reflecting systemic inflammation and metabolic stress frequently linked to cancer development.

Several variables with unexpectedly high importance—such as prostate-specific antigen (PSA) and digital-rectal-exam codes—were traced to overlapping diagnostic templates within the synthetic Synthea records. These behave as generic “screening-marker” surrogates rather than organ-specific findings and therefore illustrate an artefact of synthetic

EHR generation rather than biological signal. Conversely, smoking status and other classical exposure factors appeared with only small positive SHAP values (~ 0.20). This under-representation likely stems from incomplete or randomized encoding of behavioral data in the synthetic dataset, which limits the model's ability to learn population-level exposure effects.

Features with mean |SHAP| < 0.18 (≈ 5 % of the SCC magnitude) had negligible influence and were excluded from the interpretive summary. The resulting panel of ≈ 15 variables therefore provides a compact, transparent feature set dominated by tumor, metabolic, and demographic indicators that remain clinically plausible.

4-5-2- Directionality, Noise, and Clinical Coherence

Positive SHAP values for SCC, metabolic disorders, and older age increased the predicted probability of malignancy, while normal metabolic profiles or absence of such markers exerted neutral or weakly negative effects. The appearance of non-specific laboratory or procedural codes (PSA, calcium, general blood counts) likely reflects synthetic co-correlation noise. Such artefacts highlight an expected limitation of simulation-based data generation and do not diminish the model's methodological validity. Overall, the SHAP landscape remains biologically consistent with recognised lung-cancer pathways in which chronic inflammation and metabolic imbalance precede malignant transformation. Nevertheless, because the data are synthetic, these relationships should be interpreted as plausible associations requiring confirmation on real clinical datasets.

4-5-3- Statistical Confidence and Model Robustness

Bootstrapped 95% confidence intervals (B 2000, stratified by class) were computed on independent test set to quantify performance stability. Point estimates are reported alongside their intervals in Table 2. The hybrid ensemble achieved ROC-AUC = 0.9925 (95 % CI 0.989 – 0.997) and F1 = 0.93 (95% CI 0.90-0.95), indicating both high accuracy and low variance. The narrow confidence widths (± 0.02-0.03) confirm reproducibility rather than overfitting. Simpler learners such as Logistic Regression or KNN displayed wider intervals and lower mean scores, underscoring the ensemble's efficiency-performance balance.

Table 2. Bootstrapped 95 % Confidence Intervals for Model Performance Metrics

Model	Accuracy (95 % CI)	Recall (95 % CI)	F1-Score (95 % CI)	ROC-AUC (95 % CI)
Hybrid Ensemble (GWO-WOA-XGB)	0.976 (0.970-0.982)	0.945 (0.932–0.957)	0.930 (0.900–0.950)	0.993 (0.989–0.997)
XGBoost (Base)	0.958 (0.951-0.965)	0.912 (0.897–0.926)	0.902 (0.887–0.916)	0.975 (0.969–0.981)
Random Forest	0.939 (0.930–0.948)	0.888 (0.870–0.905)	0.873 (0.857–0.890)	0.955 (0.946–0.964)
Logistic Regression	0.904 (0.893–0.915)	0.852 (0.832–0.871)	0.842 (0.823–0.862)	0.919 (0.907–0.931)
SVC	0.887 (0.874–0.900)	0.834 (0.812–0.856)	0.821 (0.800–0.841)	0.900 (0.886–0.914)
KNN	0.865 (0.850–0.880)	0.809 (0.784–0.833)	0.796 (0.772–0.819)	0.878 (0.862–0.894)

5- Conclusions

5-1- Conclusion

Starting with baseline models and progressively improving performance using nature-inspired optimization techniques and a hybrid ensemble framework, this work displayed a methodical approach for lung cancer diagnosis with machine learning. The findings are summarized as follows:

- Baseline Benchmarking: A strong performance baseline was given by conventional classifiers like Random Forest, AdaBoost, Bagging, and Decision Trees. For accuracy, recall, and general robustness, ensemble-based methods usually exceeded simpler algorithms.
- NIA-Based Optimizations: By hyperparameter optimization, metaheuristic algorithms—such as Genetic Algorithm, Grey Wolf Optimizer, Moth-Flame Optimization, Particle Swarm Optimization, and Whale Optimization Algorithm—improved predictive metrics. Often achieving around 99% accuracy, ensemble models such Random Forest, Extra Trees, and XGBoost routinely drew advantages from these advanced optimization methods.
- Hybrid Ensemble: Preserving reasonable training times, the hybrid ensemble—which combined the most successful models from baseline and NIA-optimised configurations—achieved the best accuracy—about 99.25%. For large-scale or time-sensitive applications, this approach effectively balanced computational efficiency with prediction accuracy.
- Clinical Relevance: Strong precision-recall measures and the higher ROC-AUC (0.9984) show how well the system can detect lung cancer cases. Still, more validation with real-world data is necessary to confirm the therapeutic relevance of the study since it uses synthetic data.

Although it is only preliminary research, this paper highlights the potential of machine learning in improving lung cancer diagnosis. The methods and results presented here provide a basis for further projects, therefore encouraging further improvement and validation of the suggested models in many different and useful medical environments.

5-2- Limitation

- Synthetic Dataset: Although it provides a good framework for study, the synthetic dataset falls short in capturing the variety and complexity of real-world clinical settings. This limitation might affect the generalizability of the findings.
- Scalability: The computational requirements of some hybrid models could make resource-limited healthcare environments problematic; hence, extra optimization is necessary to ensure accessibility and efficiency.
- Model Interpretability: The complexity of ensemble models could make them difficult to interpret, which is absolutely vital for therapeutic use. Building the confidence of medical experts depends on establishing explainable models.
- Data Bias: Under control by set criteria, the engineering process and feature selection could accidentally bring bias. Reduction of this risk depends on validation using diverse datasets.

5-3- Future Scope

- Validation with Real-World Data: Real-world clinical data should be included in future research to confirm the generalizability and strength of the framework. Cooperation with medical establishments would give access to varied and representative data samples.
- Multi-Modal Integration: Combining information from several sources—medical imaging, genetic profiles, and electronic health records—could increase predictive accuracy and provide a thorough understanding of lung cancer diagnosis.
- Explainability in Clinical Use: Building trust among doctors and patients depends on improving model interpretability by means of SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-Agnostic Explanations).
- Adaptation to Resource-Limited Settings: Developing lightweight models or incremental learning solutions catered for situations with limited resources, such as rural clinics or edge devices, will guarantee more general accessibility.
- Real-Time Applications: Investigating online and streaming-based categorization methods could help to detect lung cancer in real time, especially in telemedicine and emergency care situations where quick feedback is vital.

5-4- Translational and Collaborative Implications

- Collaborative Potential: Cooperation with clinical researchers will help future projects to test the relevance of these models using actual patient data. Such collaborations could close the distance between computer developments and useful use in the healthcare sector.
- Integration with Existing Systems: Though experimental, the suggested approaches have great potential for inclusion into diagnostic processes, especially for early-stage lung cancer identification. To guarantee congruence with clinical criteria and procedures, nonetheless, more research is needed.
- Practical Limitations: Although the outcomes are positive, some algorithms' cost and training time could create challenges for acceptance in real-time environments, especially in low-resource settings.

The translational potential of this framework lies in its adaptability to real-world healthcare infrastructure. By leveraging routinely collected EHR data, the proposed pipeline can be extended to diverse healthcare systems without demanding costly imaging workflows or bespoke computational resources. Future collaborations with clinicians and data scientists will be essential to align algorithmic performance with the operational realities of screening programs and hospital information systems. Such collaboration will also ensure that model outputs are clinically interpretable and usable within standard decision-support interfaces.

Beyond technical translation, practical adoption requires careful consideration of ethical, regulatory, and workflow integration issues which are discussed in the following subsection.

5-5- Ethical, Regulatory, and Clinical Integration Considerations

5-5-1- Integration into Clinical Workflow

The framework developed in this study is envisioned as a supportive analytical layer rather than a replacement for clinical judgment. The model could be integrated into existing EHR systems, operating passively in the background to flag potential high-risk profiles for further screening. Each flag would prompt physician review rather than automatic action, preserving clinical oversight. Such integration would allow scalable, low-cost support for screening in primary and secondary care—especially in institutions lacking radiology-based infrastructure. While this work did not involve direct collaboration with clinicians, future research will include multidisciplinary validation with medical professionals to assess usability, interpretability, and workflow compatibility.

5-5-2- Ethical and Regulatory Dimensions

Responsible deployment of AI in healthcare requires strict adherence to data privacy, transparency, and fairness. Although the present study employs synthetic data, any real-world implementation must comply with privacy frameworks such as GDPR and HIPAA, ensuring de-identification and secure model governance. Bias mitigation remains essential—algorithmic behavior should be monitored across demographic subgroups to prevent disparities in screening outcomes. Explainability is another ethical necessity: methods such as SHAP and permutation importance help make model reasoning traceable to clinicians, reinforcing trust and accountability.

Finally, any deployment in clinical settings would require regulatory clearance (e.g., under FDA's software as a Medical Device framework or EU MDR) and institutional ethical review, underscoring that models like this are exploratory decision-support tools, not autonomous diagnostic systems.

6- Declarations

6-1- Author Contributions

Conceptualization, G.M. and B.T.; methodology, G.M. and B.T.; software, G.M. and B.T.; validation, G.M., B.T., D.T., and S.S.; formal analysis, G.M. and B.T.; investigation, G.M. and B.T.; resources, G.M. and B.T.; data curation, G.M., B.T., and D.T.; writing—original draft preparation, G.M. and B.T.; writing—review and editing, B.T., D.T., and S.T.; visualization, G.M. and B.T.; supervision, B.T., D.T., and S.S.; project administration, D.T. and S.S.; funding acquisition, G.M. and B.T. All authors have read and agreed to the published version of the manuscript.

6-2- Data Availability Statement

The data presented in this study are available on request from the corresponding author.

6-3- Funding

The authors received no financial support for the research, authorship, and/or publication of this article.

6-4- Institutional Review Board Statement

Not applicable.

6-5- Informed Consent Statement

Not applicable.

6-6- Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this manuscript. In addition, the ethical issues, including plagiarism, informed consent, misconduct, data fabrication and/or falsification, double publication and/or submission, and redundancies have been completely observed by the authors.

7- References

[1] Spiro, S. C., & Silvestri, G. A. (2005). One hundred years of lung cancer. American Journal of Respiratory and Critical Care Medicine, 172(5), 523–529. doi:10.1164/rccm.200504-531OE.

[2] Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. CA: A Cancer Journal for Clinicians, 71(3), 209–249. doi:10.3322/caac.21660.

[3] CDC (2026). Lung Cancer. Centers for Disease Control and Prevention (CDC), Georgia, United States. Available online: https://www.cdc.gov/lung-cancer/index.html (accessed on January 2026).

[4] Ridge, C., McErlean, A. M., & Ginsberg, M. S. (2013). Epidemiology of lung cancer. Seminars in Interventional Radiology, 30(2), 93–98. doi:10.1055/s-0033-1342949.

[5] WHO (2026). Lung Cancer. World Health Organization (WHO), Geneva, Switzerland. Available online: https://www.who.int/news-room/fact-sheets/detail/lung-cancer (accessed on January 2026).

[6] WCRF (2022). Liver Cancer Statistics. World Cancer Research Fund International (WCRF), London, United Kingdom. Available online: https://www.wcrf.org/preventing-cancer/cancer-statistics/lung-cancer-statistics/ (accessed on January 2026).

[7] Collins, L., Haines, C., Perkel, R., & Enck, R. (2007). Lung Cancer: Diagnosis and Management. American Family Physician, 75, 56–63.

[8] Nooreldeen, R., & Bach, H. (2021). Current and future development in lung cancer diagnosis. International Journal of Molecular Sciences, 22(16), 8661. doi:10.3390/ijms22168661.

[9] Chen, A., Wu, E., Huang, R., Shen, B., Han, R., Wen, J., Zhang, Z., & Li, Q. (2024). Development of Lung Cancer Risk Prediction Machine Learning Models for Equitable Learning Health System: Retrospective Study. JMIR Publications (Preprint), 1-29. doi:10.2196/preprints.56590.

[10] Ebrahimi, A., Henriksen, M. B. H., Brasen, C. L., Hilberg, O., Hansen, T. F., Jensen, L. H., Peimankar, A., & Wiil, U. K. (2024). Identification of patients' smoking status using an explainable AI approach: a Danish electronic health records case study. BMC Medical Research Methodology, 24(1), 114. doi:10.1186/s12874-024-02231-4.

[11] Wang, L., Yin, Y., Glampson, B., Peach, R., Barahona, M., Delaney, B. C., & Mayer, E. K. (2024). Transformer-based deep learning model for the diagnosis of suspected lung cancer in primary care based on electronic health record data. eBioMedicine, 110, 105442. doi:10.1016/j.ebiom.2024.105442.

[12] Bhattarai, K., Oh, I. Y., Sierra, J. M., Tang, J., Payne, P. R. O., Abrams, Z., & Lai, A. M. (2024). Leveraging GPT-4 for identifying cancer phenotypes in electronic health records: A performance comparison between GPT-4, GPT-3.5-turbo, Flan-T5, Llama-3-8B, and spaCy's rule-based and machine learning-based methods. JAMIA Open, 7(3), ooae060. doi:10.1093/jamiaopen/ooae060.

[13] Su, Y., Zhan, H., Li, S., Lu, Y., Ma, R., Fang, H., Xu, T., & Tian, Y. (2025). Development and Validation of Machine Learning Models for Lung Cancer Risk Prediction in High-Risk Population: A Retrospective Cohort Study. Biomedical and Environmental Sciences, 38(4), 501–505. doi:10.3967/bes2025.038.

[14] Liz-López, H., de Sojo-Hernández, Á. A., D'Antonio-Maceiras, S., Díaz-Martínez, M. A., & Camacho, D. (2025). Deep Learning Innovations in the Detection of Lung Cancer: Advances, Trends, and Open Challenges. Cognitive Computation, 17(2), 67. doi:10.1007/s12559-025-10408-2.

[15] Rieke, N., Hancox, J., Li, W., Milletarì, F., Roth, H. R., Albarqouni, S., Bakas, S., Galtier, M. N., Landman, B. A., Maier-Hein, K., Ourselin, S., Sheller, M., Summers, R. M., Trask, A., Xu, D., Baust, M., & Cardoso, M. J. (2020). The future of digital health with federated learning. NPJ Digital Medicine, 3(1), 119. doi:10.1038/s41746-020-00323-1.

[16] Harvard Dataverse (2024). Synthetic patient data for lung cancer risk prediction machine learning – Synthetic Patient Data ML Dataverse. Harvard Dataverse, Massachusetts, United States. Available online: https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/GD5XWE (accessed on January 2026).

[17] Post, A. R., Burningham, Z., & Halwani, A. S. (2022). Electronic Health Record Data in Cancer Learning Health Systems: Challenges and Opportunities. JCO Clinical Cancer Informatics, 6. doi:10.1200/cci.21.00158.

[18] Wu, W., Parmar, C., Grossmann, P., Quackenbush, J., Lambin, P., Bussink, J., Mak, R., & Aerts, H. J. W. L. (2016). Exploratory study to identify radiomics classifiers for lung cancer histology. Frontiers in Oncology, 6, 71. doi:10.3389/fonc.2016.00071.

[19] Kumar, D., Chung, A. G., Shaifee, M. J., Khalvati, F., Haider, M. A., & Wong, A. (2017). Discovery radiomics for pathologically-proven computed tomography lung cancer prediction. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 10317 LNCS, 54–62. doi:10.1007/978-3-319-59876-5_7.

[20] Zhou, Z., Zhou, Z. J., Hao, H., Li, S., Chen, X., Zhang, Y., ... & Wang, J. (2017). Constructing multi-modality and multi-classifier radiomics predictive models through reliable classifier fusion. arXiv preprint, arXiv:1710.01614. doi:10.48550/arXiv.1710.01614.

[21] Yuan, F., Lu, L., & Zou, Q. (2020). Analysis of gene expression profiles of lung cancer subtypes with machine learning algorithms. Biochimica et Biophysica Acta - Molecular Basis of Disease, 1866(8), 165822. doi:10.1016/j.bbadis.2020.165822.

[22] Shin, H., Oh, S., Hong, S., Kang, M., Kang, D., Ji, Y. G., Choi, B. H., Kang, K. W., Jeong, H., Park, Y., Kim, H. K., & Choi, Y. (2020). Early-Stage Lung Cancer Diagnosis by Deep Learning-Based Spectroscopic Analysis of Circulating Exosomes. ACS Nano, 14(5), 5435–5444. doi:10.1021/acsnano.9b09119.

[23] Wang, R., Weng, Y., Zhou, Z., Chen, L., Hao, H., & Wang, J. (2019). Multi-objective ensemble deep learning using electronic health records to predict outcomes after lung cancer radiotherapy. Physics in Medicine and Biology, 64(24), ab555e. doi:10.1088/1361-6560/ab555e.

[24] Enhesari, A., Montazeri, M., & Baghshah, M. S. (2013). Hyper-Heuristic Algorithm for Finding Efficient Features in Diagnose of Lung Cancer Disease. Journal of Basic and Applied Scientific Research, 3(10), 134–140.

[25] Xie, Y., Meng, W. Y., Li, R. Z., Wang, Y. W., Qian, X., Chan, C., Yu, Z. F., Fan, X. X., Pan, H. D., Xie, C., Wu, Q. B., Yan, P. Y., Liu, L., Tang, Y. J., Yao, X. J., Wang, M. F., & Leung, E. L. H. (2021). Early lung cancer diagnostic biomarker discovery by machine learning methods. Translational Oncology, 14(1), 100907. doi:10.1016/j.tranon.2020.100907.

[26] Gould, M. K., Huang, B. Z., Tammemagi, M. C., Kinar, Y., & Shiff, R. (2021). Machine learning for early lung cancer identification using routine clinical and laboratory data. American Journal of Respiratory and Critical Care Medicine, 204(4), 445–453. doi:10.1164/rccm.202007-2791OC.

[27] Senthil Kumar, K., Venkatalakshmi, K., & Karthikeyan, K. (2019). Lung Cancer Detection Using Image Segmentation by means of Various Evolutionary Algorithms. Computational and Mathematical Methods in Medicine, 2019(1), 4909846. doi:10.1155/2019/4909846.

[28] Vijh, S., Gaurav, P., & Pandey, H. M. (2023). Hybrid bio-inspired algorithm and convolutional neural network for automatic lung tumor detection. Neural Computing and Applications, 35(33), 23711–23724. doi:10.1007/s00521-020-05362-z.

[29] Priyadharshini, P., & Zoraida, B. S. E. (2021). Bat-inspired metaheuristic convolutional neural network algorithms for CAD-based lung cancer prediction. Journal of Applied Science and Engineering, 24(1), 65–71. doi:10.6180/jase.202102_24(1).0008.

[30] Gupta, N., Gupta, D., Khanna, A., Rebouças Filho, P. P., & de Albuquerque, V. H. C. (2019). Evolutionary algorithms for automatic lung disease detection. Measurement: Journal of the International Measurement Confederation, 140, 590–608. doi:10.1016/j.measurement.2019.02.042.

[31] ALzubi, J. A., Bharathikannan, B., Tanwar, S., Manikandan, R., Khanna, A., & Thaventhiran, C. (2019). Boosted neural network ensemble classification for lung cancer disease diagnosis. Applied Soft Computing Journal, 80, 579–591. doi:10.1016/j.asoc.2019.04.031.

[32] Freund, Y., & Schapire, R. E. (1996). Experiments with a New Boosting Algorithm. Proceedings of the 13th International Conference on Machine Learning, 148–156. doi:10.1.1.133.1040.

[33] Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197–227. doi:10.1007/bf00116037.

[34] Breiman, L. (1996). Bagging predictors. Machine Learning, 24(2), 123–140. doi:10.1023/A:1018054314350.

[35] Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5–32. doi:10.1023/A:1010933404324.

[36] Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (2017). Classification and Regression Trees. Routledge, New York, United States. doi:10.1201/9781315139470.

[37] Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. Mathematical Intelligencer, 27(2), 83-85. doi:10.1007/BF02985802.

[38] Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189–1232. doi:10.1214/aos/1013203451.

[39] Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21–27. doi:10.1109/tit.1967.1053964.

[40] Dasarathy, B.V. (1991) Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos, United States.

[41] Hosmer, D. W., Lemeshow, S., & Sturdivant, R. X. (2013). Applied Logistic Regression. Wiley Series in Probability and Statistics. Wiley, New Jersey, United States. doi:10.1002/9781118548387.

[42] Agresti, A. (2002). Categorical Data Analysis. Wiley Series in Probability and Statistics. Wiley, New Jersey, United States. doi:10.1002/0471249688.

[43] Heaton, J. (2017). Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning. Genetic Programming and Evolvable Machines, 19(1–2), 305–307. doi:10.1007/s10710-017-9314-z.

[44] Nielsen, M. (2026). Neural Networks and Deep Learning. Available online: http://neuralnetworksanddeeplearning.com/ (accessed on January 2026).

[45] Rish, I. (2001) An Empirical Study of the Naive Bayes Classifier. IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence, Seattle, 4 August 2001, 41-46.

[46] Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to Information Retrieval. Cambridge University Press, Massachusetts, United States. doi:10.1017/cbo9780511809071.

[47] Geurts, P., Ernst, D., & Wehenkel, L. (2006). Extremely randomized trees. Machine Learning, 63(1), 3–42. doi:10.1007/s10994-006-6226-1.

[48] Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273–297. doi:10.1007/bf00994018.

[49] Shawe-Taylor, J., & Cristianini, N. (2004). Kernel Methods for Pattern Analysis. Cambridge University Press, Massachusetts, United States. doi:10.1017/cbo9780511809682

[50] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 13-17-August-2016, 785–794. doi:10.1145/2939672.2939785.

[51] Holland, J. H. (1992). Adaptation in Natural and Artificial Systems. MIT Press, Massachusetts, United States. doi:10.7551/mitpress/1090.001.0001.

[52] Goldberg, D. E. (1989). Genetic algorithms in search, optimization, and machine learning. Choice Reviews Online, 27(02), 27-0936. doi:10.5860/choice.27-0936.

[53] Mirjalili, S., Mirjalili, S. M., & Lewis, A. (2014). Grey Wolf Optimizer. Advances in Engineering Software, 69, 46–61. doi:10.1016/j.advengsoft.2013.12.007.

[54] Mirjalili, S. (2015). How effective is the Grey Wolf optimizer in training multi-layer perceptrons. Applied Intelligence, 43(1), 150–161. doi:10.1007/s10489-014-0645-7.

[55] Mirjalili, S. (2015). Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm. Knowledge-Based Systems, 89, 228–249. doi:10.1016/j.knosys.2015.07.006.

[56] Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. Proceedings of ICNN'95 - International Conference on Neural Networks, 4, 1942–1948. doi:10.1109/icnn.1995.488968.

[57] Shi, Y., & Eberhart, R. (1998). Modified particle swarm optimizer. Proceedings of the IEEE Conference on Evolutionary Computation, ICEC, 69–73. doi:10.1109/icec.1998.699146.

[58] Mirjalili, S., & Lewis, A. (2016). The Whale Optimization Algorithm. Advances in Engineering Software, 95, 51–67. doi:10.1016/j.advengsoft.2016.01.008.

[59] Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. IJCAI International Joint Conference on Artificial Intelligence, 2, 1137–1143.

[60] Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. arXiv Preprint, arXiv:1504.00854. doi:10.48550/arXiv.2010.16061.

[61] Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). Modern Information Retrieval. ACM Press, New York, United States.

[62] Blair, D. C. (1979). Information Retrieval, 2nd ed. C.J. Van Rijsbergen. London: Butterworths; 1979: 208 pp. Price: $32.50. Journal of the American Society for Information Science, 30(6), 374–375. doi:10.1002/asi.4630300621.

[63] Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters, 27(8), 861–874. doi:10.1016/j.patrec.2005.10.010.