Received: August 4, 2022. Revised: October 24, 2022.

INASS

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Improving Lung Cancer Relapse Prediction Using the Developed Optuna_XGB Classification Model

Rana Dhia'a Abdu-Aljabar[1]* **Osama A. Awad[1]**

[1]*Information Engineering College, Al-Nahrain University, Baghdad, Iraq*
* Corresponding author's Email: ranadhiaa1@nahrainuniv.edu.iq

Abstract: Lung cancer is more likely to relapse in the first five years following surgery; even though the operation may have been a complete success, there remains a chance that the lung cancer could return. This return may lead the patient to die after a successful surgery. Because there are no symptoms of lung cancer in its early stage, many researchers use intelligent systems to predict the relapse of lung cancer in its early stages. The outcome of previous works considering this issue still suffers from low prediction accuracy. This study proposed a method to predict lung cancer relapse more accurately. This method has multiple stages: 1st optimization system, feature selection stage, 2nd optimization stage, and extreme gradient boost (XGBoost) classifications stage. It used two datasets (GSE8894 and GSE68465) of a gene expression microarray for NSCLC with its clinical information on relapse state. We obtained three probes (3 genes) with clinical data combinations that can get good prediction results. These genes included 225389_at (BTBD6), 220239_at (KLHL7), and 204832_s_at (BMPR1A). A comparison between the proposed model and the original XGBoost with PSO and Hyperopt as hyperparameter optimization for the XGBoost classification model is performed. Extensive comparisons with four machine learning algorithms, including Deep Forest, K-nearest neighbor (KNN), Support Vector Machine (SVM), and Naive Bayes, are conducted. The proposed model accuracies are 0.93 for the GSE8894 dataset and 0.81 for the GSE68465 dataset.

Keywords: XGBoost classifier, Intelligent systems, Machine learning, Optuna, Optimization, Lung cancer, Gene expression, Microarray dataset.

1. Introduction

The term "lung cancer relapse" or "recurrence" refers to lung cancer that comes back after treatment. A relapse may be in a different type of previous cancer and may occur in the exact or other location as before. Even with early-stage cancers and new cancer treatments, lung cancer recurrence occurs rapidly, perhaps in three months or more often than one might assume [1]. Most lung cancers recur after two to five years from the first diagnosis, depending on the cancer type and stage. The relapse rate in stage 1 NSCLC patients is approximately three in 10 people, increasing in stage 4 to nearly seven in 10 [2]. Generally, early-stage tumor prediction has better clinical outcomes, and tumor staging aids treatment arrangement. However, there are cases where patients unexpectedly produce recurrent disease, exemplifying the limitations of current clinical staging techniques in precisely predicting tumor recurrence. The main benefit of early detecting the lung cacner relapse after success surgary is a lower chance of dying from lung cancer. Therefore, the prediction of lung cancer recurrence is crucial for categorizing patients to help doctors make decisions on therapeutic strategies. Many studies have tried to improve a method to predict lung cancer relapse early using gene expression profiles. They used different methods and had different results, such as R Alanni. et al. [3-8]. They proposed various studies in new optimization models to improve NSCLC detection using microarray datasets. Hasseeb A. et al. [9-12] improved multiclass using the Gene Expression Programming (GEP) algorithm to classify lung cancer. Y. Onish el al. [13] examined the use of a deep convolutional neural network (DCNN) for the automatic categorization of lung nodules in CT

images. They tested if the classification accuracy is enhanced by producing a large number of fresh pulmonary nodule pictures using generative adversarial networks (GANs), which is a typical challenge in medical research when only little quantities of data are available. Shu-long Li et al. [14] developed an incorporation technique that integrates handmade features (HF) into the features learned in the output layer of a 3D deep convolutional neural network to predict lung nodule malignancies They dealt with an imbalance dataset has 431 malignant nodules and 795 benign nodules extracted from the LIDC/IDRI database. The proposed model result has an accuracy value of 88.66 and an AUC value of 93.03. Lai, Y et al. [15] upgraded the deep neural classifier by utilizing clinical and gene expression datasets to predict the survival of lung cancer patients. It dealt with an imbalance dataset that has 512 patients; 355 survivals and 157 deaths. The results were AUC= 0.8163 and the accuracy is 0.7544. The previous three kinds of research [13-15] suffer from an imbalanced dataset which mostly occurs in bio datasets. This kind of dataset let the classification models divergent from their true values, as can see from the different values between the AUC metric (which takes the class imbalance into account) and the accuracy metric. Our proposed model handled this kind of dataset.

Wang, Q., et al. [16] presented a random forest with self-paced learning bootstrap that was demonstrated to enhance lung cancer classification and prognosis based on gene expression data. To be more precise, they suggested using ensemble learning and a random forest strategy to pick several classifiers, which would enhance the model's classification performance. Then, through self-paced learning, they gradually incorporate high to low-quality samples to evaluate the sampling technique. According to experimental findings based on five publicly available datasets on lung cancer, the accuracy values for the GSE4115, GSE33356, GSE3141, GSE8894, and GSE40419 models are 0.8261, 0.9472, 0.7059, 0.6905, and 0.9796, respectively. L. V. Povoa et al. [17], employed the Multi Learning Training (MuLT) algorithm, which combines supervised, unsupervised, and self-supervised learning techniques to identify cancer patients with a low and high risk of developing the disease. Through five-fold cross-validation trials, our method is assessed using three separate, publicly available cancer data sets while taking three different performance elements into account. MuLT outperforms alternative approaches, reaching AUCs of 0.6457. Mu Teng et al [18] analyzed the prognostic energy metabolism (EM) related gene signature using the Univariate Cox and LASSO (Least Absolute Shrinkage and Selection Operator) methods. To verify the prognostic value of the prognostic signatures, Kaplan-Meier and receiver operating characteristic (ROC) curves were plotted. Based on a risk model, a nomogram was developed to forecast the likelihood that LUAD would survive. 13 EM-related genes were compiled into a prognostic signature by the researchers. At one year, three years, and five years in the GSE31210 dataset, the AUC was 0.57, 0.67, and 0.73, respectively. At one year, three years, and five years in the GSE68465 dataset, the AUCs were 0.69, 0.63, and 0.63, respectively. Shahweli, Z. N [19] used an enhancer DBN classifier in predicted lung cancer. This classifier is related to the unsupervised phase using two restricted Boltzmann machines (RBMs) and a supervised phase when the deep belief network (DBN) is trained by a backpropagation neural network (BPNN). Essam H. Houssein et al. [20] proposed a hybrid algorithm from MRFO and SVM to select the most predictive and informative genes for cancer classification. Hui Jiang [21] used particle swarm optimization (PSO) to tune the hyperparameters of XGBoost to enhance the network intrusion detection system's accuracy. R. Dhia'a. et al. [22-24] compared multiple machine learnings on lung cancer prediction and found that XGBoost is the most accurate model when applied to balance and imbalance datasets. They used multiple XGBoost layers with different XGBoost hyperparameters to improve the prediction.

This study attempted to improve the lung cancer relapse-prediction probability after surgery by overcoming the bio datasets drawbacks such as the missing information, high dimension (a large number of features according to low samples), noise, and imbalance class. These drawbacks overcame in our proposed model by pre-processing the datasets at the beginning, selecting suitable genes involved in causing lung cancer, and improving XGBoost by applying the Optuna model to automatically tune the XGBoost hyperparameters to strengthen its construction.

The remaining paper is structured as follows. Section 2 provides lung cancer datasets. Section 3 presents the XGBoost classification works. Section 4 described the optimization part. Section 5 presented the proposed model. Section 6 describes experimental results and discussion, Finally, Section 7 concludes the discussion and provides future work.

2. Lung cancer datasets

Two microarray gene expression datasets, GSE8894 and GSE68465, were used. The data were

Table 1. Relapse dataset information

Datasets	Patients	Features	Relapse Class	Non_Relapse Class
GSE8894	135	54675	67	68
GSE68465	362	22283	205	157

gathered to represent the patient's clinical information and gene expression profiles. Gene expression displays alterations in the expression of several genes simultaneously due to lung cancer. The class of this work depends on the relapse state from the clinical information. It has two states; relapse/non-relapse. So, the system is binary classification, the 1 indicates the relapse and 0 for non-relapse. Both datasets were downloaded from the public Gene Expression Omnibus (GEO) database.

2.1 Dataset information

Two datasets are used in this study. Their types are gene expression microarray types with their clinical information. The first dataset (GSE8894) is an NCLC type for 138 cases; 67 cases have a lung cancer relapse state, and 68 patients have nonrelapse lung cancer. The second is the GSE68465 dataset. It has gene expression and clinical information for 442 cases; after removing the incomplete data, 362 cases remain, with 205 cases having a cancer relapse and 157 cases having nonrelapse cancer. It can summarize the relapse datasets in Table 1.

2.2 Data preprocessing

It is critical to clean biological data to increase their quality for searching and analyzing. It accomplishes this by removing the mess-up or incorrect records from the database. Every record with incomplete data must be eliminated because it is considered irrelevant and leads to incorrect learning outcomes. Furthermore, XGBoost is concerned with the numeric representation in the decision class, whereas classes in the lung cancer datasets, such as nonrecurrence and relapse, are nominal. As a result, they must be converted to numeric form (0/1).

3. XGBoost classification model

XGBoost is an ensemble machine learning method based on decision trees. It employs the gradient boosting method. Tianqi Chen and Carlos Guestrin created XGBoost in 2016. They presented their results at the SIGKDD conference [25]. It offers parallel tree boosting, which addresses many data science issues rapidly and correctly. It has several hyperparameters that allow one to fine-tune the model training process [24].

The greedy algorithm used in XGboost to evaluate the split candidates:

$$L_{split} = \frac{1}{2}\left[\frac{(\sum_{i\epsilon S_L} g_i)^2}{(\sum_{i\epsilon S_L} h_i+\lambda)} + \frac{(\sum_{i\epsilon S_L} g_i)^2}{(\sum_{i\epsilon S_R} h_i+\lambda)} - \frac{(\sum_{i\epsilon I} g_i)^2}{(\sum_{i\epsilon I} h_i+\lambda)}\right] - \gamma \quad (1)$$

$$g = yi - \hat{y}i \quad (2)$$

$$h = \hat{y}(1 - \hat{y}) \quad (3)$$

where;
L_{split}: the quality or the gain of each candidate split
S_L, and S_R: are the instance sets of left and right nodes after the split.
i: is the root instance sets
γ: the leaf weight penalty parameter.
λ: the tree size penalty parameter

If $L_{split} < 0$ it will be neglected or pruned. In the end, it will choose the largest value as a splitting point for that feature.

4. Optuna optimization model

Takuya Akiba et al. [26] introduced new design criteria for next-generation hyperparameter optimization software called Optuna, which has multiple features that give it the flexibility to deal with complex code, less time, and more accuracy in searching for optimal solutions. It is a software framework developed primarily for machine learning for automated hyperparameter tuning. It offers a new defining-by-run style API that allows the user to optimize hyperparameters while preserving greater flexibility than other frameworks, even if the user code is complicated. It can also optimize hyperparameters in a complex space as no other framework can previously represent. It can also stop unpromising testing before the training ends [27].

4.1 Optuna algorithm

Hyperparameter optimization has two parts in the Optuna model; they usually work together to quickly find the best hyperparameter values. The first is the sampling algorithm, which decides where to look. It uses a trial history record to select the next hyperparameter. It estimates and tests the best location and then calculates an even more promising area based on the new result. This method is repeated by utilizing the historical data of previous trials. It used multiple sampling methods; this study employed the Tree-structured Parzen Estimator (TPE) [28] as a sampling method. The second part is the pruning algorithm used when the particle trial is

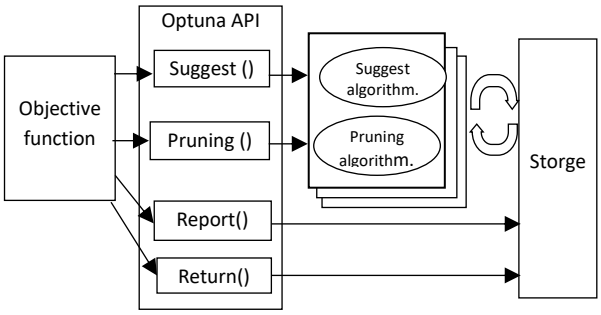

Figure. 1 Optuna optimization model design

not promising; it could terminate it early to provide a time for better trials. It is based on an asynchronous variant of the successive halving algorithm (ASHA) [29], which is a technique to parallelize SHA [30]. Fig. 1 shows the design of the Optuna model [26].

4.1.1. Tree-structured parzen estimator

The difference between random search and sequential model-based optimization (SMBO) is that random search is un-informed and requires more trials to maximize the objective function (accuracy). Tree Parzen Estimator (TPE), an algorithm used for SMBO, spends more time choosing the next values, but overall requires fewer evaluations of the objective function because it can reason about the next values to evaluate. Over many iterations, the TPE algorithm concentrates the search around the most promising values, yielding; higher scores on the objective function, and faster optimization.

4.1.2. Asynchronous successive halving algorithm

Modern learning models are characterized by large hyperparameter spaces and long training times. The asynchronous successive halving algorithm (ASHA) exploits parallelism and aggressive early-stopping to tackle large-scale hyperparameter optimization problems. In Optuna, the successive halving algorithm (SHA) has been slightly modified by using the TPE algorithm for sampling suggestions instead of the random search algorithm. Therefore, it behaves differently from the algorithm described in the paper [28]. The pruning in SHA depends on the median stopping rule. This rule means that the trial with a sample evaluation worse than the median value of the previous trial's intermediate values will be trimmed away.

5. Proposed model

It is impossible to educate a machine learning to suit all types of data. In our scenario, XGBoost successfully learned with high accuracy on specific datasets but had less accuracy on others [22]. These differences in the accuracy are because of its significant dependence on its hyperparameter setting. To overcome this problem, it used the Optuna model to select the best hyperparameters to tune the XGBoost to accommodate different types of datasets. The proposed methodology has four stages, as shown in Fig. 2.

Table 2. XGBoost hyperparameter information

Hyperparameter	The range	Default value	Suggested range
colsample_bytree	(0,1]	1	[0.6-1]
gamma	[0,∞]	0	[0-4]
eta (learning rate)	[0,1]	0.3	[0.2-0.6]
Max_depth	[0,∞]	6	[2-10]
Min_child_wieght	[0,∞]	1	[2-8]
n_estimators	[1,∞]	100	[1-200]
subsample	(0,1]	1	[0.5-1]

The optimization stage: Multiple types of lung cancer databases are required in our scenario. Because the bio dataset is noisy in general, the model must tune its hyperparameters with each dataset to avoid overfitting or underfitting and to be able to handle a variety of datasets. As a result, it auto-mated the tuning of XGBoost hyperparameters using the Optuna model.

Six common hyperparameters were used in this study's optimization: the learning_rate, max_depth, n_estimators, subsample, gamma, and colsample_bytree. First, it identified the search space for these hyperparameters. From experience, it can suggest a limit range for each hyperparameter in the search space to enhance the accuracy and decrease the optimization time, as seen in Table 2.

It shows how it can limit the hyperparameter search space range to obtain a more accurate result for the XGBoost model. The objective value in this study is the accuracy value, so this value needs to be maximized. The Optuna with XGBoost (Optuna_XGB) used the dataset for optimization. It repeats the optimization for 100 trials to obtain the best XGBoost hyperparameters that got the best accuracy. At the end of this stage, the best hyperparameters are chosen to tune the XGBoost.

The feature selection stage: Not all genes are involved in lung cancer recurrence. Therefore, we tried to select suitable genes for more accurate prediction. For that purpose, it used the XGBoost model to rank the importance of the features (genes) in making the prediction.

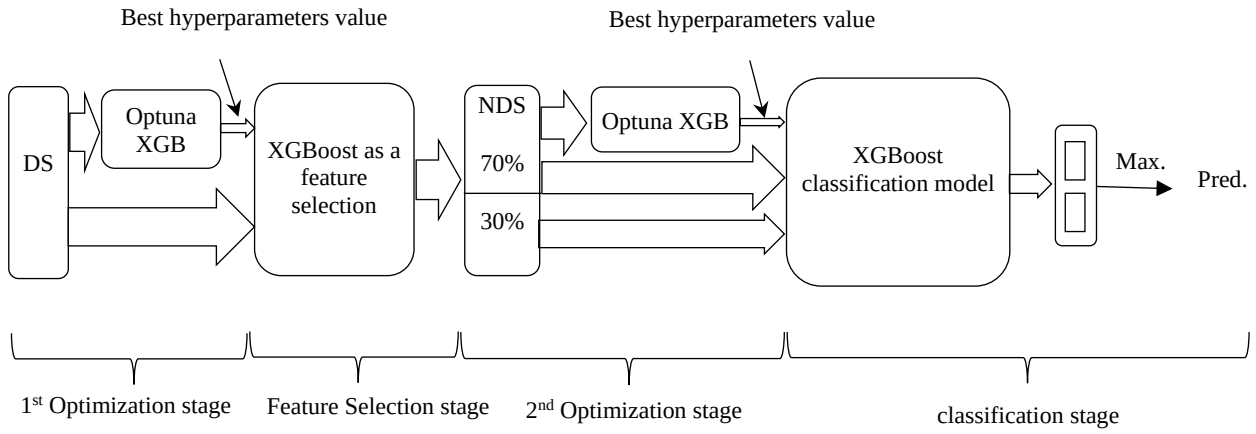

Figure. 2 The proposed model

In XGBoost, the importance of features is calculated after constructing the boosted trees. The calculation depends on how many each feature is used in that construction. Each time the feature is used in the construction, the higher the importance score. The importance score refers to how this feature is valuable or helpful in constructing the trees. The importance score algorithm calculated the importance of each decision tree by counting each feature, causing a splitting point, and improving the performance measure, weighted by the number of observations for which the node is responsible.

The feature importance is averaged across all decision trees within the model [31]. In XGBoost it can calculate the importance by three methods; wights, gain and cover. This paper used the gain to calculate the importance which had the best accuracy in our case. At the end of XGBoost construction, it calculates the importance of each feature. During the construction, the gain calculates during the splitting choice. It finds the similarity for each root, right, and left node and then calculates the gain by following equations derived from Eq. (1):

$$S = \frac{\sum(y_i - p_{i-1})^2}{\sum(p_{i-1}(1 - p_{i-1})) + \lambda} \quad (4)$$

$$\text{Gain} = S_L + S_R - S_{root} \quad \text{-}\gamma \quad (5)$$

Where;
y_i: the Actual class
p_{i-1}: the class probability prediction of the previous tree
S_L : left similarity node.
S_R : right similarity node.
S_{root} : the root similarity
λ, γ: for regulation, to decrease the overfitting.

If Gain > 0 it accepts the splitting if not the node will prune. This procedure will be repeated to the number of offered splitting points till have the best gain value and this point will consider a splitting point. This calculation will be repeated for each tree in the XGBoost. In the end, it calculates the importance score for each feature by dividing the summation of its gain by the total features gain.

In Fig. 2, it is seen that in this stage, the whole dataset (DS) is used to construct the XGBoost to calculate the score importance of each feature and select only the features with the higher score, which are stored in a new dataset (NDS). The threshold taken in this study is all features above zero. Each feature with an importance score equal to zero will be neglected. After it repeats this stage five times it took the average of their important scores. As a result, it obtained three features which are called probes that have the highest important score and got the highest accuracy when they were used in the classification stage. These probes with their gene symbols are 225389_at (BTBD6), 220239_at (KLHL7), and 204832_s_at (BMPR1A).

The second optimization stage: The XGBoost is very sensitive to their hyperparameters value and because the dataset features are changed it must repeat its hyperparameters tuning. The Optuna is used again to find the best hyperparameters and it used the same six hyperparameters used in the first optimization stage with the same range's value.

The last stage is the classification stage: it constructs the XGBoost classification depending on 70% of the data and the chosen hyperparameters from the previous stage. After the learning phase, the testing phase will begin using the testing data (30% of the dataset) with the XGBoost built in the learning phase. At the end of this stage, the final prediction of the lung cancer relapse will be obtained.

6. Results and discussion

This study used different metrics for evaluating the proposed model. These metrics are sensitivity, specificity, precision, f1_score, AUC (area under the curve), accuracy, and the time consumed for learning and testing data for each machine learning used in this study. The results are the average of five runs of each model. At first, it showed how Optuna_XGB works, then compared their obtained results with other machine learning. Finally, discussed the work.

6.1 Analyzing optuna_XGB work

Optimization algorithms navigate the search space of input variables to locate the optimal. The shape of the objective function and the algorithm's behavior in the search space is opaque in real-world problems. To analyze the Optuna algorithms, it tries to visualize the behavior of its optimization in the search space. Once the Optuna process is completed, it can obtain the best set of hyperparameters values for the XGBoost model. The best hyperparameters of the Optuna_XGB model after 100 trials when it was applied to the GSE8894 dataset were 'colsample_bytree': 0.99, 'gamma': 2.51, 'eta': 0.25,'max_depth': 2, 'n_estimators': 14, and sub-sample: 0.61, and the best trial was 8, with a training accuracy value of 0.8125. In the GSE68465 dataset, the best trial was 32 with an accuracy value of 0.818, and the best hyperparameters were 'learning_rate': 0.31, 'max_depth': 2, 'n_estimators': 43, 'subsample': 0.76, 'gamma': 0.1, 'min_child_weight': 3, and 'colsample_bytree': 0.93.

The GSE8894 dataset results are used to represent the Optuna behavior in the search space. The optimization history of all trials is plotted in Fig. 3. That figure represented the accuracy of each trial and marked the best one from all trials that it currently had until it reached the best of all, which is the accuracy of 0.8925 in trial 76. Fig. 4 represents each hyperparameter value used in the optimization procedure in an independent slide. It shows the distribution value for each hyperparameter in each trial. The higher point in each slide has the best accuracy value, and the dark point represents the higher number of trials that have used the same hyperparameter value. It used this figure to update the limit range of optimization search space (see Table 2) by choosing a new range of each hyperparameter with a higher objective value. This updating improves the prediction accuracy and minimizes the time required to reach the best accuracy value.

6.2 Comparing Optuna_XGB with the original XGBoost

The use of the Optuna for XGBoost hyperparameter tuning enhanced the original XGBoost in predicting the lung cancer relapse probability when applied to both lung cancer datasets. The original XGBoost used the default tuning of the hyperparameters; see Table 2. In contrast, Optuna_XGB used the best hyperparameters chosen by the Optuna model.

The Optuna_XGB applied to both datasets improved the original XGBoost performance in all metrics except the time metric (see Table 3 and Figs. 5 and 6). The time spent in XGBoost is only 11 s and 17 s, while Optuna_XGB is 01:22 (82) and 01:33 (93 s) for GSE8894 and GSE68465, respectively. However, it is still an acceptable value

6.3 Optuna_XGB comparison with other optimization models

This comparison used two recent optimization models: the HyperOpt [32] optimization and the PSO optimization model used in the PSO_XGBoost model [21]. The results of both optimization models have accuracies of 0.83 and 0.86 in the GSE8894 dataset and 0.73 and 0.70 in the GSE68465 dataset for HyperOptXGB and PSOXGB, respectively.

They both enhanced the original XGBoost model performance in most states, but not as Optuna did, which had accuracies of 0.93 and 0.81 in the GSE8894 and GSE68465 datasets, respectively. Table 3 and Fig. 7 and 8 show the remaining metrics.

6.4 Optuna_XGB comparison with the other machine learning

This section compares Optuna_XGB with some standard and current machine learning methods, such as KNN (k-nearest neighbors), naive Bayes, SVM (support vector machines) [33], and gcForest (deep forest) [34], to evaluate the effectiveness of Optuna_XGB. They are set to their default setting. The comparison results tabulated in Table 3 represent the efficiencies of the Optuna_XGB model compared to other machine learning models. Although in the GSE68465 dataset, the SVM obtained a higher sensitivity (the rate of detecting lung cancer relapse cases) than Optuna_XGB, it failed in specificity (the rate of detecting nonrelapse cases). In contrast, the KNN model has a higher value than Optuna_XGB in specificity, but it fell in the sensitivity metric. The performance of Optuna_XGB, in general, is still better than those of the other metrics, as seen in Table 3 and Fig. 9 and 10.

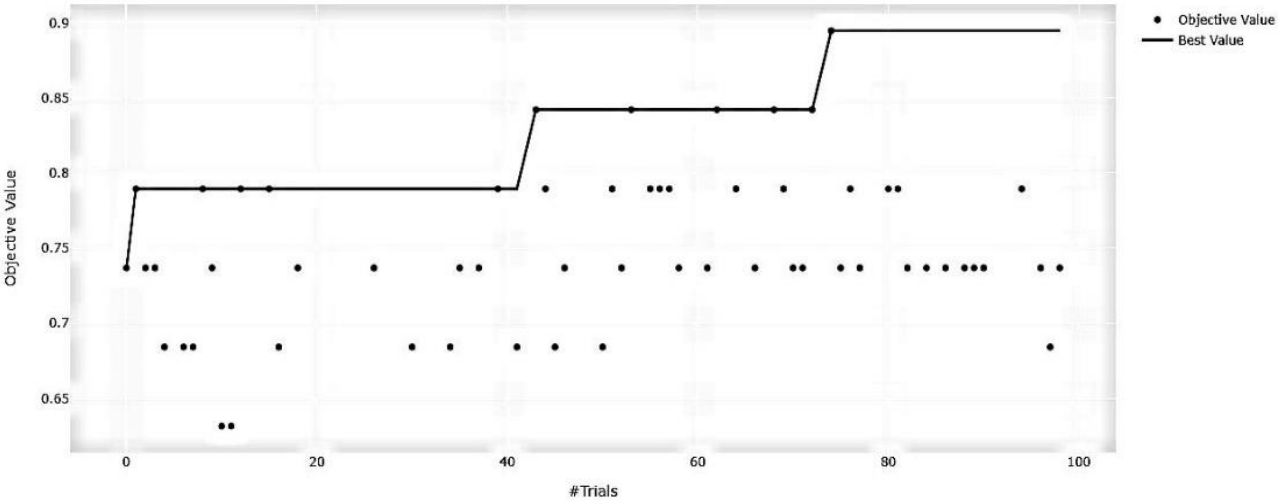

Figure. 3 The plot shows the accuracies (objective values) for all tries and marks the best values during the optimization stage of the Optuna_XGB model when applied to the GSE8894 dataset

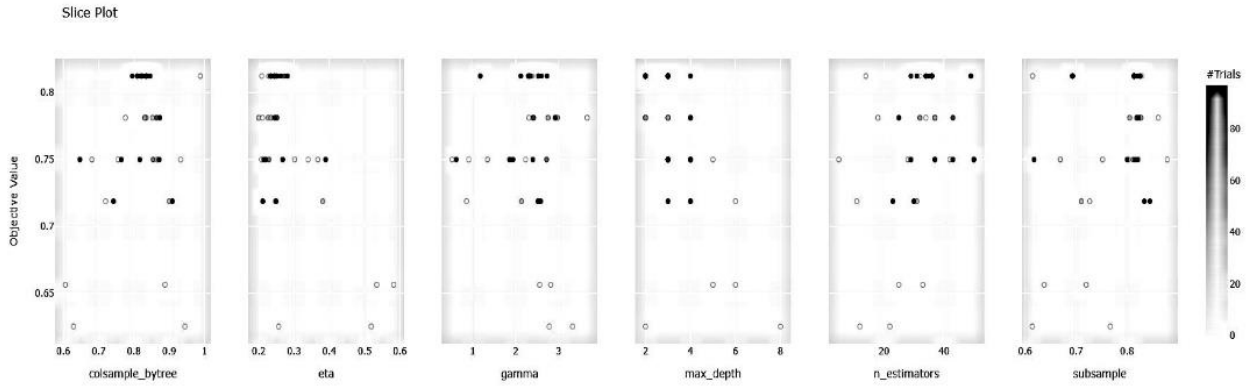

Figure. 4 Slice plot for each hyperparameter

Table 3. The comparison results for all models used in this study

Classifier Name	Sensitivity	Specificity	Precision	F1_score	AUC	Accuracy	Time (hour)
GSE8894 dataset							
Optuna_XGB	1.00	0.86	0.87	0.93	0.93	0.93	00:01:22
HyperOptXGB	0.80	0.86	0.84	0.82	0.83	0.83	00:41:37
PSOXGB	0.80	0.71	0.73	0.76	0.76	0.76	00:01:40
XGBoost	0.55	0.67	0.61	0.58	0.61	0.61	00:00:11
SVM	0.65	0.43	0.52	0.58	0.54	0.54	00:00:05
gcForest	0.40	0.67	0.53	0.46	0.53	0.54	00:02:47
KNN	0.45	0.76	0.64	0.53	0.61	0.61	00:00:02
Naive Bayes	0.50	0.62	0.56	0.53	0.56	0.56	00:00:01
GSE68465 dataset							
Optuna_XGB	0.90	0.68	0.79	0.84	0.79	0.81	00:01:33
HyperOptXGB	0.77	0.68	0.76	0.77	0.73	0.73	01:12:38
PSOXGB	0.84	0.51	0.69	0.76	0.67	0.70	00:01:26
XGBoost	0.81	0.57	0.71	0.76	0.69	0.71	00:00:14
SVM	1.0	0.02	0.57	0.73	0.51	0.58	00:00:35
gcForest	0.89	0.57	0.73	0.80	0.73	0.75	00:01:42
KNN	0.41	0.72	0.67	0.51	0.57	0.55	00:00:03
Naive Bayes	0.65	0.47	0.62	0.63	0.56	0.57	00:00:02

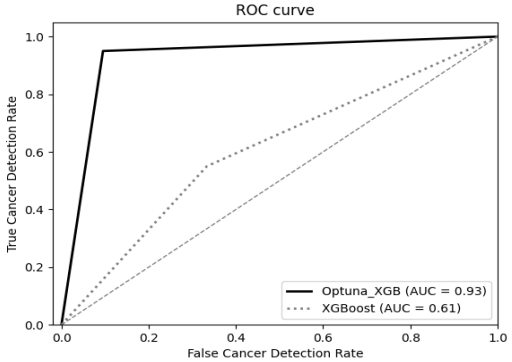

Figure. 5 The ROC and AUC values of the Optuna_XGB and the original XGBoost model when they were applied to the GES8894 dataset

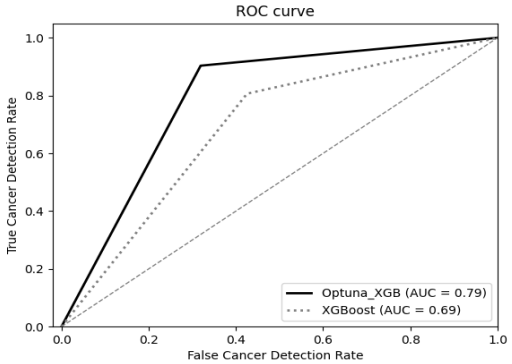

Figure. 6 The ROC and AUC values of the Optuna_XGB and the original XGBoost models when they were applied to the GES68465 dataset

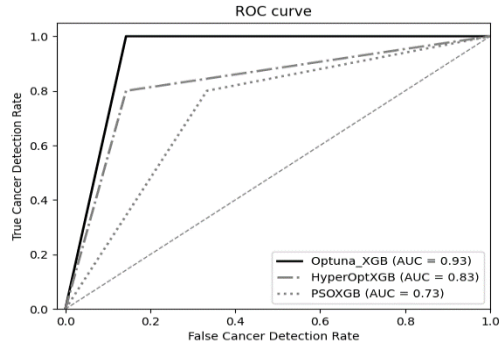

Figure. 7 The ROC and AUC values of the Optuna_XGB model and the representative optimization models when applied to the GES8894 dataset

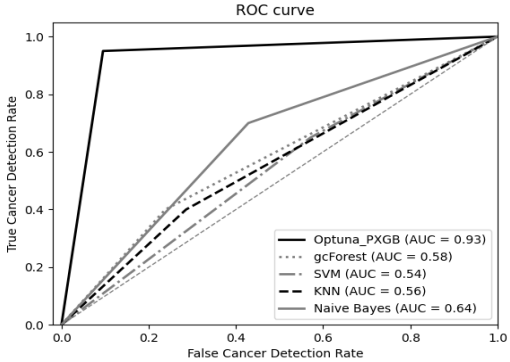

Figure. 8 The ROC and AUC values of the Optuna_XGB model and the representative optimization models when applied to the GSE68465 dataset

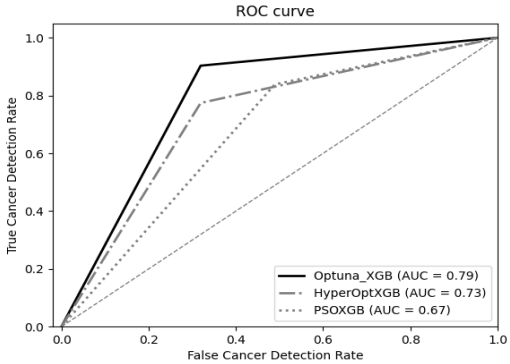

Figure. 9 The ROC and AUC values of Optuna_XGB and the other machine models on the GES8894 dataset

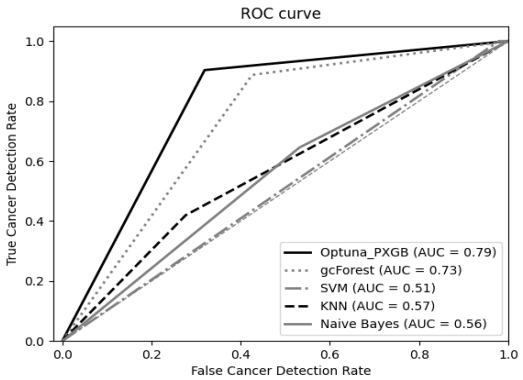

Figure. 10 The ROC and AUC values of Optuna_XGB and the other machine models on the GES68465 dataset

The time consumed in Optuna_XGB is not as good as most machine learning, but it is still acceptable.

6.5 Optuna_XGB comparison with the other published papers

To have a full view of Optuna_XGB performance its accuracy and AUC values compared with other recently published papers which are discussed in the introduction section. This comparison used only the works that used the same datasets as illustrated in Table 4. Mu Teng et al. tried to use gene signatures related to EM prognostic to predict lung cancer relapse. They use the AUC matric to evaluate their work. The results for prediction of lung cancer relapse in the 1st, 3rd, and 5th stage were o,69, 0.63, and 0.63 respectively. Additionally, L. V. Povoa et al. used multi-learning training and then take the average result of all of them. The AUC was 0.6457.

Table 4. Comparison of Optuna_XGB with other published works have the same relapse datasets used in this study

Dataset	Author name	Date	Classification type	Result
GSE68465	The current work	2022	Optuna_XGB	AUC = 0.79., Acc.=0.81
	Mu Teng [16]	2022	Univariate Cox and LASSO + nomogram	AUCs=0.69 in 1st year, 0.63 in 3rd year, and 0.63 in 5th years
	L. V. Povoa et al. [17]	2021	Multi Learning Training (MuLT) algorithm	AUCs= 0.6457
GSE8894	The current work	2022	Optuna_XGB	Acc.=0.93
	Wang, Q. et al. [18]	2020	random forest with self-paced learning (RFSPL)	Acc.= 0.6905
	Russul A. et al. [7]	2019	Deep genetic selection (DGS)	Acc.= 0.8714

Both of the previous works used the GSE68456. The same dataset is used by the Optuna_XGB and the results were AUC=0.86 and acc.= 0.87. The following two works used the GSE8894 dataset. The first article by R. Alanni et al. used Deep genetic selection to select the best genes. The AUC was 0.8714. In the second article Q. Wang used the developed RFSPL. The acc. was 0.6905. while the proposed work accuracy in the same dataset was 0.93.

6.6 Discussion

The proposed methodology tried to improve the XGBoost model to enhance the accuracy of diagnosing the patient's probability of relapse after successful surgery to support the doctor's assessment to have earlier decision-making for better patient treatment. These enhancements are dependent on the main tools;

First, prepossessing the datasets to delete the whole samples that have the missing data. Doing that decreases the data noise.

Second, feature selection selected the suitable genes to decrease the data noise and their high dimension and made the system faster and more accurate.

Second, it can be seen from the results that the use of Optuna as an optimization model gives the XGBoost model two specifications; the first one is a good sampling. This sampling makes the model select the best hyperparameters sets for multi XGBoost, giving it the best classification metrics The second specification that Optuna provided is the pruning method, which made Optuna_XGB finish the learning time for all datasets within a short time compared to the other representative optimization method; PSO_XGB and Hyperopt_XGB models.

Third, from the result, it can see that a lot of machine learning failed in the imbalance dataset, while the Optuna_XGB is slightly affected by this situation. That is because of the combination that let it suitable for a wide range of data.

7. Conclusion

This work aims to improve the opportunity to handle lung cancer earlier in the case of relapse prediction to enhance patient treatment. They used the gene expression data in the microarray dataset to have more accurate results. The bio datasets are severe from different drawbacks such as missing information, high dimension, noise, and imbalance class. This paper proposed a multistage method to overcome these drawbacks. This methodology consists of an optimization stage using the Optuna model to optimize the XGBoost hyperparameters that use as a feature selection for two lung cancer relapse datasets. It selects three features (probes), these probes with their gene symbols are 225389_at (BTBD6), 220239_at (KLHL7), and 204832_s_at (BMPR1A). In the third stage, it used the Optuna again to tune the XGBoost hyperparameters for perfect tree construction compatible with the new changes in datasets. In the end, it used the best hyperparameters in constructing the XGBoost classification. The experimental results show the Optuna_XGB improvement in both accuracy and AUC values. Where the proposed model has accuracies of 93% and 81% for the GSE8894 and GSE68465 datasets respectively. Which they are the highest lung cancer relapse prediction accuracy compared to other compared models. Additionally, it has the highest AUC values for the GSE8894 and GSE68465 datasets; 93% and 0.79% respectively. Which is means the better handling the imbalance dataset (GSE68465) than the other compared models (HyperOptXGB, PSOXGB, original XGBoost, SVM, Random Forest, KNN, and Naive Bayes).

For future work, it will take n top sets of hyperparameters from Optuna and run multiple XGBoost in parallel each one tuned with one of these

n top sets of hyperparameters to have different XGBoost constructions. Then take the average outputs. This improvement will increase the model diversity, so it can handle a wider range of data.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Rana Dhia'a Abdu_aljabar conceived this research, designed and performed experiments, interpretation of the data; and wrote the manuscript; Dr. Osama A. Awad supervised all the processing and revised the manuscript critically for important intellectual content. All authors read and approved the final manuscript.

References

[1] P. Choi, S. Jeong, and S. Yoon, "Prognosis of recurrence after complete resection in early-stage non-small cell lung cancer", *The Korean Journal of Thoracic and Cardiovascular Surgery*, Vol. 46, No. 6, pp. 449-456, 2013.

[2] H. Sasaki, A. Suzuki, T. Tatematsu, M. Shitara, Y. Hikosaka, and K. Okuda, "Prognosis of recurrent non-small cell lung cancer following complete resection", *Oncology Letters*, Vol. 7, No. 4, pp. 1300-1304, 2014.

[3] R. A. Anni, J. Hou, R. A. Aljabar, and Y. Xiang, "Prediction of NSCLC recurrence from microarray data with GEP", *IET Systems Biology*, Vol. 11, No. 3, pp. 77-85, 2017.

[4] R. A. Anni, J. Hou, H. Azzawi, and Y. Xiang, "Cancer adjuvant chemotherapy prediction model for non-small-cell lung cancer", *IET Systems Biology*, Vol. 13, No. 3, 2018.

[5] R. A. Anni, J. Hou, H. Azzawi, and Y. Xiang, "A novel gene selection algorithm for cancer classification using microarray datasets", *BMC Medical Genomics*, Vol. 12, No. 10, 2018.

[6] R. A. Anni, J. Hou, H. Azzawi, and Y. Xiang, "Risk classification for NSCLC survival using microarray and clinical data", *International Journal of Advances in Electronics and Computer Science*, Vol. 6, No. 5, 2019.

[7] R. A. Anni, J. Hou, H. Azzawi, and Y. Xiang, "deep gene selection method to select genes from microarray datasets for cancer classification", *BMC Bioinformatics*, Vol. 20, 2019.

[8] R. A. Anni, J. Hou, H. Azzawi, and Y. Xiang, "New Gene Selection Method Using Gene Expression Programming Approach on Microarray Data Sets", *Lee R. (eds) Computer and Information Science, Studies in Computational Intelligence*, Vol. 791, pp. 17-31, 2018.

[9] H. Azzawi, J. Hou, Y. Xiang, and R. A. Anni, "Lung cancer prediction from microarray data by gene expression programming", *IET Systems Biology*, Vol. 10, No. 5, pp. 168-178, 2016.

[10] H. Azzawi, J. Hou, R. A. Anni, H. Azzawi, Y. Xiang, R. A. Aljabar, and A. Azzawi, "Multiclass lung cancer diagnosis by gene expression programming and microarray datasets", In: *Proc. of 13th Int. Conf. on Advanced Data Mining and Applications*, pp. 541-553, 2017.

[11] H. Azzawi, J. Hou, Y. Xiang, and R. A. Anni, "SBC: A new strategy for multiclass Lung cancer classification based on tumour structural information and microarray data", In: *Proc. of 17th IEEE/ACIS International Conf. on Computer and Information Science*, pp. 68-73, 2018.

[12] H. Azzawi, J. Hou, R. A. Anni, and Y. Xiang, "A hybrid neural network approach for lung cancer classification with gene expression dataset and prior biological knowledge", In: *Proc. of International Conf. on Machine Learning for Networking, Paris France Springer, Cham, Lecture Notes in Computer Science*, 11407, pp. 279-293, 2019.

[13] Y. Onishi, A. Teramoto, M. Tsujimoto, T. Tsukamoto, K. Saito, H. Toyama, K. Imaizumi, and H. Fujita, "Automated pulmonary nodule classification in computed tomography images using a deep convolutional neural network trained by generative adversarial networks", *BioMed Research International*, Vol. 2019, No. 6051939, 2019.

[14] S. Li, P. Xu; B. Li, L. Chen, Z. Zhou, H. Hao, Y. Duanl, M. Folkert, J. Mal, S. Huang, S. Jiang, and J. Wang, "Predicting lung nodule malignancies by combining deep convolutional neural network and handcrafted features", *Physics in Medicine & Biology*, Vol. 64, No. 17, 2019.

[15] Y. Lai, W. Chen, T. Hsu, C. Lin, Y. Tsao, and S. Wu, "Overall survival prediction of non-small-cell lung cancer by integrating

microarray and clinical data with deep learning", *Scientific Reports*, Vol. 10, No. 4679, 2020.

[16] M. Teng, L. Haoran, and L. Xiangnan, "Prognostic Implication of Energy Metabolism-Related Gene Signatures in Lung Adenocarcinoma", *Frontiers in Oncology*, Vol. 12, 2022.

[17] L. Povoa, U. Calvi, A. Lorena, C. Ribeiro, and I. Silva, "A Multi-Learning Training Approach for Distinguishing Low and High Risk Cancer Patients", *IEEE Access*, Vol. 9, pp. 115453-115465, 2021.

[18] Q. Wang, Y. Zhou, W. Ding, Z. Zhang, K. Muhammad, and Z. Cao, "Random Forest with self-paced bootstrap learning in lung cancer prognosis", *ACM Transactions on Multimedia Computing, Communications, and Applications*, Vol. 16, No. 1, pp. 1-12, 2020.

[19] N. Shahweli, "Deep Belief Network for Predicting the Predisposition to Lung Cancer in TP53 Gene", *Iraqi Journal of Science*, Vol. 61, No. 1, pp. 171-177, 2020.

[20] E. Houssein, H. Hassan, and M. A. Sayed, "Gene Selection for Microarray Cancer Classification based on Manta Rays Foraging Optimization and Support Vector Machines", *Arabian Journal for Science and Engineering Volume*, 2021.

[21] H. Jiang, Z. He, G. Ye, and H. Zhang, "Network Intrusion Detection Based on PSO-Xgboost Model", *IEEE Access*, Vol. 8, pp. 58392-58401, 2020.

[22] R. A. Aljabar and O. Awad, "A Comparative analysis study of lung cancer detection and relapse prediction using XGBoost classifier", In: *Proc. of 2nd Int. Sci. Conf. of Eng. Sci. (ISCES2020), Col. of Eng., Univ. of Diyala & IOP Pub.*, 1076, 2021.

[23] R. A. Aljabar and O. Awad, "Parallel extreme gradient boosting classifier for lung cancer detection", *Indonesian Journal of Electrical Engineering and Computer Science*, Vol. 24, No. 3, pp. 1610-1617, 2021.

[24] R. A. Aljabar and O. Awad, "Lung Cancer Relapse Prediction Using Parallel XGBoost", *Iraqi Journal of Information and Communication Technology*, Vol. 5, No. 2, 2022.

[25] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", In: *Proc. of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining*, pp. 785-794, 2016.

[26] Y. Li, D. Umbach, B. Adrienna, Q. Li, Y. Zhuang, and L. Li, "Putative biomarkers for predicting tumor sample purity based on gene expression data", *BMC Genomics*, Vol. 20, No. 1021, 2019.

[27] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework", In: *Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 2623-2631, 2019.

[28] J. Bergstra, R Bardenet, Y. Bengio, and K. Balázs, "Algorithms for Hyper-Parameter Optimization", In: *Proc. of the 25th Annual Conf. on Neural Inf". Processing Sys. NIPS*, 2011.

[29] L. Li, K. Jamieson, A. Rostamizadeh, E. Gonina, M. Hardt, B. Recht, and A. Talwalkar, "A System for Massively Parallel Hyperparameter Tuning", In: *Proc. of the 3rd MLSys Conference*, 2020.

[30] T. Kraska, A. Talwalkar, J. Duchi, R Griffith, M. Franklin, and M. Jordan, "MLbase: Distributed Machine-learning System", In: *Proc. of the 6th Biennial Conference on Innovative Data Systems Research (CIDR'13)*, 2013.

[31] X. Ji, W. Tong, Z. Liu, and T. Shi, "Five-Feature Model for Developing the Classifier for Synergistic vs. Antagonistic Drug Combinations Built by XGBoost", *Front Genet*., Vol. 10, No. 600, 2019.

[32] J. Bergstra, D. Yamins, B. Komer, E. Chris, D. Yamins, and D. David, "Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms", In: *Proc. of the 12th Pyhton in Science Conf. (SCIPY 2013)*, 2013.

[33] L. Wang, "Support Vector Machines: Theory and Applications", *USA: Springer*, 2005.

[34] Z. Zhou and J. Feng, "Deep Forest: towards an alternative to deep neural networks", In: *Proc. of the 26th Int. Joint Conf. on AI. (IJCAI-17)*, pp. 3553-3559, 2017.