IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 3 November 2023, accepted 24 November 2023, date of publication 4 December 2023,
date of current version 13 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3339225

RESEARCH ARTICLE

HypGB: High Accuracy GB Classifier for Predicting Heart Disease With HyperOpt HPO Framework and LASSO FS Method

ABBAS JAFAR AND MYUNGHO LEE, (Member, IEEE)
Department of Computer Engineering, Myongji University, Yongin 17058, South Korea

Corresponding author: Myungho Lee (myunghol@mju.ac.kr)

This work was supported by the Supercomputer Development Leading Program of the National Research Foundation of Korea (NRF) funded by the Korean government (MSIT: Ministry of Science and ICT) (No. 2020M3H6A1084853). This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korean government (MSIT: Ministry of Science and ICT) (RS-2023-00321688).

ABSTRACT Early prediction of cardiovascular disease is crucial for medical experts to make informed decisions. Effective diagnosis of heart disease can help prevent heart failure, heart attacks, stroke, and coronary artery disease. This paper aims to build a high-accuracy heart disease prediction system using machine learning. For this purpose, an automatic machine learning system called HypGB was developed. HypGB uses a Gradient Boosting (GB) model to classify patients with heart disease. It also uses a standard LASSO feature selection technique to identify the most informative feature subset and remove noisy and redundant features from clinical data. The GB model was also optimized with the latest HyperOpt optimization framework to determine the best configuration of the hyperparameters. Experimental results using two open-source heart disease clinical datasets (Cleveland heart disease and Kaggle heart failure) indicate that HypGB identifies the most accurate features and obtains the optimal combinations of hyperparameters that efficiently predict heart disease. It achieved the highest classification accuracies of 97.32% and 97.72% using the Cleveland and Kaggle datasets, respectively, which outperformed the previous approaches. With the highest accuracy, the HypGB system shows its potential for implementation in the healthcare domain to help medical experts predict heart disease fast and accurately.

INDEX TERMS Cardiovascular disease, machine learning algorithms, hyperparameter optimization, redundant features, disease prediction, clinical data analysis.

I. INTRODUCTION

Cardiovascular diseases pose a serious health concern worldwide, affecting the lives of more than 17 million people each year. The World Health Organization (WHO) anticipates that this number could increase to over 23 million by 2030 [1]. The major factors for heart disease include poor diet, obesity, stress, high blood pressure, uncontrolled diabetes, and genetic predisposition [1]. An American study has identified several common symptoms of heart disease such as irregular heartbeats, swelling in the legs, chest pain, fatigue, and sleep disorders [2]. Early detection of heart disease can lead to better outcomes, increasing the chances of receiving effective treatment, and reducing the risk of death. However, there is a lack of technological tools for diagnosing the disease in its early stages [3].

Traditional heart disease diagnosis relies on medical history, reports, and the evaluation of symptoms by a medical expert. However, this approach is not always effective and takes a long time to complete. Recently, decision-making technologies such as Machine Learning (ML) have been developed to help practitioners diagnose patients effectively [4]. With ML algorithms, an automatic prediction system can be built as an analytical tool that can solve difficult health problems by learning from medical reports, making decisions, and predicting the disease [5]. In fact, heart disease diagnosis systems using ML algorithms have recently been developed [6], [7].

The associate editor coordinating the review of this manuscript and approving it for publication was Rongbo Zhu.

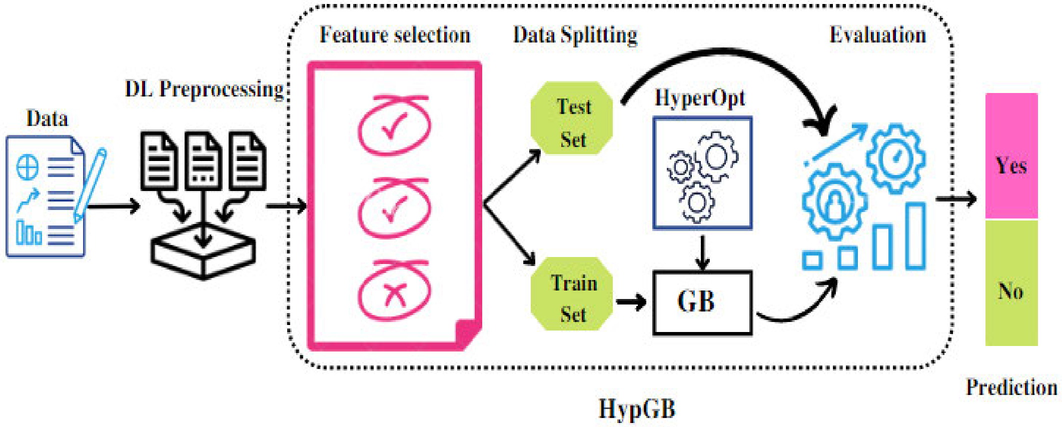

FIGURE 1. The procedure of HypGB automatic heart disease prediction system.

For an accurate heart disease diagnosis, we must efficiently deal with the heart disease data and improve the interpretability of the machine learning model. Medical disease data are variant, noisy, and unbalanced. In addition, it has a high dimensionality and redundant features. These can cause problems and affect the performance of the model [8]. For example, in heart disease data, the symptoms of patients vary depending on the individual, their medical tests may sometimes be inaccurate, and a lot more individuals have no disease than those suffering from heart disease [9], among many others. Furthermore, features in a high-dimensional dataset (patient age, sex, blood type, blood pressure, etc.) that are redundant and highly correlated make it difficult for the ML model to train and predict effectively.

For high interpretability of the performance outcomes of ML models, the selection of default hyperparameter values is crucial. The selection may lead to the overfitting, where the model performs well on the training data but poorly on the testing data. Furthermore, the use of default hyperparameters may lead to slow training and suboptimal performance which affects the model's accuracy and efficiency. In turn, this leads to the lack of model interpretability, thus making the early identification of heart disease more challenging [10]. Previous research on heart disease prediction [11], [12], [13], [14], [15] have attempted to overcome these issues. However, the performance of the previous models is still poor, and researchers are working on developing efficient methodologies for the timely prediction of heart disease.

In this paper, we present a high accuracy heart disease prediction system, HypGB, based on the Gradient Boosting (GB) ML model. It incorporates a feature selection approach to efficiently deal with the data and also incorporates a hyperparameter optimization approach to improve the accuracy and the interpretability of the ML model (see Fig.1). Our prediction system initially applies multiple data preprocessing techniques to solve the noise, missing values, and imbalanced data problems. We then use the Least Absolute Shrinkage Selection Operator (LASSO) feature selection (FS) method [7], [16] to extract the subset of the most important features from the dataset and rank the features based on their contributions toward the final prediction. LASSO effectively handles the problems of high dimensionality and redundant features by reducing the complexity of the ML model and preventing the overfitting. The selected subset of features is then used to train a Gradient Boosting (GB) model. As a decision tree-based ensemble learning method, GB combines weak learners to create a strong learner [17]. This makes the model more powerful and be capable of handling complex problems with larger features.

In order to further improve the generalization of the GB model, we conduct hyperparameter optimization using the HyperOpt framework, which explores a unique and best combination in the search space of the hyperparameters [18], [19]. Unlike other techniques such as grid search, random search, and Bayesian optimization, HyperOpt continuously gathers data from earlier iterations to learn from previous optimization and provides an ideal set of hyperparameters. The experimental results of the HypGB on the two publicly available datasets (the Cleveland dataset and the Kaggle dataset) show state-of-the-art accuracies: 97.32% on the Cleveland dataset and 97.72% on the Kaggle dataset. Thus, HypGB outperforms previous approaches for heart disease prediction using ML. The optimized hyperparameter combinations ensure effective performance of the ML model during training and testing, thus minimize the overfitting and also improve the interpretability of the model. This can potentially assist medical professionals in making accurate and timely disease predictions with limited knowledge of machine learning.

The remainder of this paper is organized as follows. Previous research relevant to heart disease prediction is provided in Section II. Some background information needed to design our HypGB methodology is provided in Section III. It includes the FS methods, GB classifier, and HyperOpt hyperparameter optimization framework. It also explains the

TABLE 1. Summary of the previous methods to predict the heart disease.

Years	Objective	Technique	Accuracy (%)
2010 [20]	Build an accurate and fast AI-based system to predict cardiovascular disease using SVM and ANN.	SVM+ANN	80.41
2007 [21]	Proposed a new hybrid neural network that combines the ANN with the fuzzy neural network to predict heart and diabetes diseases.	Hybrid ANN	87.41
2008 [22]	Developed an IHDPS prototype to address healthcare datasets by making intelligent decisions using multiple machine learning algorithms.	DT, NB, ANN	88.12
2020 [11]	Developed an automatic diagnostic methodology from the heart disease dataset by extracting the key features using feature selection methods.	BPNN+SVM	85.12
2019 [12]	Use of hybrid automatic approaches to diagnose cardiovascular disease by identifying the important features and improve the performance of models.	HRFLM	88.07
2017 [13]	Enhancing the prediction of heart diseases by building a hybrid system using ReliefF and Rough Set feature selection approaches.	RFRS	92.32
2020 [14]	Build an effective automated system to predict disease and improve the performance of models using multiple feature selection approaches.	FCMIM	92.37
2019 [15]	Develop an intelligent diagnostic system to overcome the problem of the overfitting by extracting the key features using a random search algorithm.	RSA+RF	93.33
2021 [25]	Development of medical diagnosis method to predict cardiovascular disease. It detects heart disease at an early stage by preventing critical cases and reducing treatment expenses.	MDSS	94

information on the heart disease datasets and performance evaluation indices used for the experiments in a later section. Section IV explains the methodology for the proposed HypGB system. Section V analyzes the experimental results, along with a comparison with the previous methods. Finally, Section VI concludes the paper.

II. PREVIOUS RESEARCH

Machine Learning has been widely adopted to diagnose heart disease. Various techniques are being developed to predict heart disease at an early stage so that it can be treated more effectively. Existing techniques are based on multiple machine learning models, artificial neural networks (ANNs), and hybrid methods.

Gudadhe et al. [20] proposed a perceptron NN trained using the backpropagation method and obtained 80.41% accuracy in identifying heart disease. Kahramanli and Allahverdi [21] proposed an automatic system that integrated fuzzy logic into an ANN to detect heart disease. They achieved 87.4% accuracy using the Cleveland dataset. Palaniappan and Awang [22] developed the IHDPS, a disease diagnostic system that predicts heart disease using various symptoms-based datasets. It is a web-based system that includes Navies Bays (NB), DT, and ANN classifiers. The ANN classifier achieved better accuracy (88.12%) than the NB (86.12%) and the DT (80.4%) classifiers.

Jabbar et al. [23] proposed an AI method that uses ANN and feature selection methods to predict the disease. They preprocessed the data using PCA to eliminate the irrelevant features. Using the Andhra Pradesh heart failure dataset, their approach outperformed the previous methods. Shah et al. [11] developed a prediction method using the FS and extraction techniques. They used the mean Fisher and accuracy-based feature selection algorithms to select important features, and PCA as a feature extraction algorithm. They trained a SVM classifier across multiple datasets and achieved 85.12% accuracy. Mohan et al. [16] developed an effective HRFLM hybrid ML approach for predicting cardiovascular disease by extracting the important features using FS methods. They achieved 88.07% accuracy. Liu et al. [13] proposed an automatic prediction system using ReliefF and Rough-Set feature selection methods. ReliefF was used to select the key features, and the irrelevant features were reduced using the Rough Set algorithm. This system recorded a high classification accuracy of 92.32%.

Saboor et al. [40] proposed an optimized machine learning solution to predict the heart disease using clinical data. ML classifiers are optimized using GridSearchCV method and the SVM achieved an accuracy of 96.72%. Li et al. [14] presented an automated learning study to predict heart patients. Different classifiers were used to classify the diseases using the Cleveland dataset. Multiple FS techniques, including LASSO, Relief, FCMIM, and MRMR, have been developed to select the relevant features. The selected features were validated using various classifiers. Additionally, the Leave-One-Subject-Out (LASO) optimization approach was used to obtain the optimal combinations of hyperparameters. They achieved a higher accuracy of 92.37% with the FCMIM and SVM classifiers, outperforming all the existing methods. Javeed et al. [15] developed an AI-based system for detecting heart failure. This diagnostic system uses the Random Forest (RF) classifier for the prediction, and the feature is selected using a random search. It uses the grid search optimization algorithm to obtain the best set of hyperparameters. RF with GS improved the accuracy by 3.3% compared with no optimization methods. Another AI-based Swarm-ANN system was proposed by Nandy et al. [24] to predict cardiovascular diseases. The ANN was optimized using Swarm optimization algorithms to update the weights of the neurons and achieved higher performance. Table 1 summarizes the previous research described above.

III. BACKGROUND

In this section, the background information needed to build our HypGB system is explained. It includes the LASSO FS, GB classifier, and HyperOpt HPO framework. These components serve as the foundation of the HypGB system and play key roles in its development and functionality. We also explain the Cleveland heart disease and Kaggle heart failure datasets, along with the performance evaluation indices used for our experiments.

A. LASSO FS APPROACH

The Least Absolute Shrinking Selection Operator (LASSO) is an efficient FS approach for data fitting that helps reduce the overfitting and enhances the prediction performance. LASSO removes redundant features without losing key information from the data. It uses the absolute coefficient to determine the importance of features, where non-important features are set to zero coefficients and removed [26]. Zhou and Wieser et. al. [27] developed a randomized LASSO in which the features with the largest coefficient values were selected as a subset of the key features. This technique was repeated multiple times to enhance the reliability of the features and select the most important features.

B. GRADIENT BOOSTING MACHINE LEARNING CLASSIFIER

We used a Gradient Boosting (GB) classifier to predict heart disease. GB is a widely used tree-boosting machine learning classifier that efficiently solves complex problems. The GB model aims to improve the performance in terms of accuracy and speed. The algorithm builds models sequentially and reduces errors by building a new model based on the errors of the previous models. It calculates the negative gradient in each iteration with respect to the values predicted from the previous models and minimizes the loss function. This negative gradient is known as residuals, which adjust the next weak learner to minimize the overall loss.

C. HyperOpt HYPERPARAMETER OPTIMIZATION

In machine learning, HPO is an approach for selecting the optimal set of hyperparameters that are not properly learned during model training. HyperOpt is an advanced open-source framework [18] that automates the optimization process. It is designed to select the optimal configuration of hyperparameters by tuning the given search space using optimization algorithms [28]. Other techniques such as grid search, random search, and Bayesian optimization require too much effort to assess the unproductive search space and do not take previous optimizations into account. The HyperOpt, on the other hand, learns from previous optimizations by continuously gathering data from earlier iterations and providing an ideal set of hyperparameters.

HyperOpt has three key components: objective function, search space of hyperparameters, and search algorithm for choosing the optimal combination of hyperparameters. The objective function is a performance metric that must be optimized. The goal of the objective function is to minimize the loss. It uses search-space hyperparameter values as input for a loss function. The search space includes all the possible hyperparameters within a range of values. HyperOpt supports a range of search algorithms to optimize the given search space and find the optimal combination of hyperparameters that results in performance improvements [28]. In this work, we use the Tree-structured Parzen Estimators (TPE) search algorithms to optimize the GB model. Fig. 2 illustrates a systematic approach to use HyperOpt with the Sklearn library to enhance the performance of the GB with TPE algorithms and configure the optimal combinations of hyperparameters.

D. CLEVELAND AND KAGGLE DATASETS

Our approach used two heart disease datasets: Cleveland heart disease [29] and Kaggle heart failure [30]. The Cleveland dataset was used for classification purposes and is available from the UCI repository [29]. The original dataset included 303 instances with 76 raw features. However, researchers processed the data and created a subset of 14 important features to accurately predict heart disease. We also used a subset of 14 features, one of which was the output label that classified whether the patient had heart disease or not. Information on the Cleveland dataset is summarized in Table 2.

The Kaggle heart failure dataset is available at the Kaggle site [30]. The dataset consists of 299 instances with 13 features. The dataset includes both genders of patients with an age range between 40 and 95 years. Out of the 13 features, there is one output target feature that has two classes (yes or no). The information on the Kaggle dataset is summarized in Table 3.

Fig. 3 and Fig. 4 show the visual representation of input features with respect to the target feature T of the Cleveland and the Kaggle heart disease datasets, respectively. The vertical axis represents the value count and the horizontal axis represents the distribution of each feature by the target feature. The histogram in each plot represents the frequency distribution of the values for each feature, whereas the kernel density estimation represents a smooth estimation of the data distribution.

E. PERFORMANCE EVALUATION INDICES

The performance in predicting heart disease can be measured using various metrics. These metrics can be calculated using

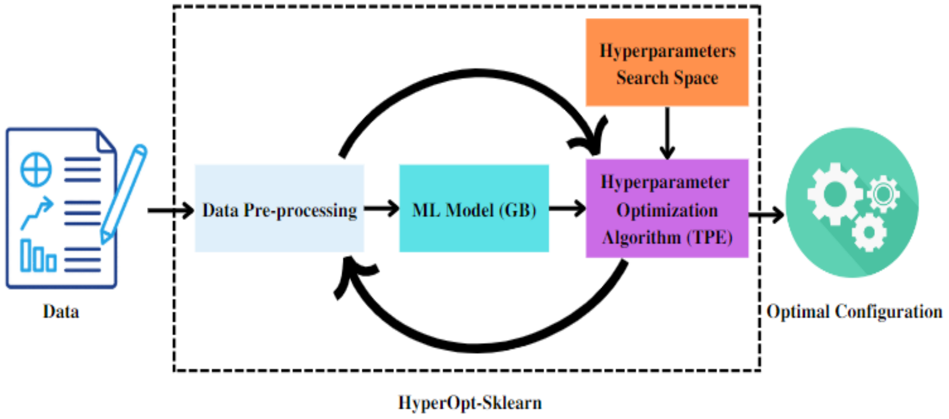

FIGURE 2. Process of the HyperOpt optimization framework.

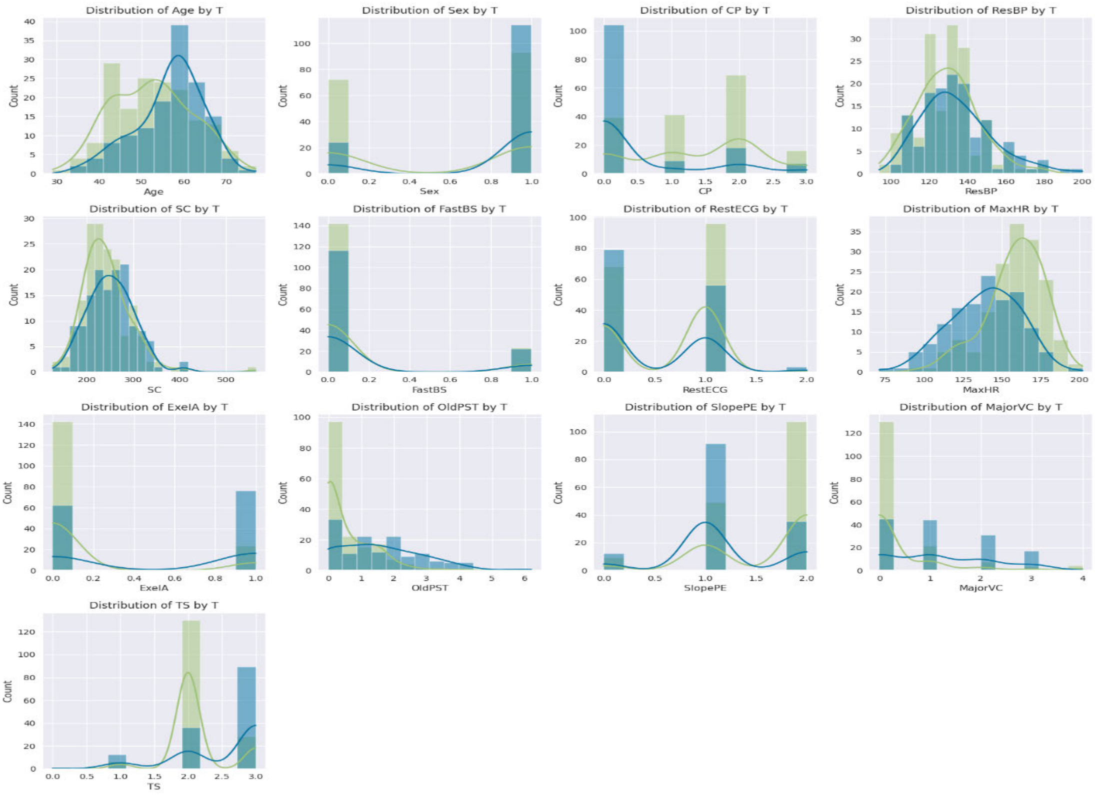

FIGURE 3. The cleveland data distributions of input features by target feature T.

confusion matrices. A confusion matrix measures the model outputs based on true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). TP refers to correctly diagnosed heart patients, whereas TN is correctly diagnosed without heart disease. FP is a patient diagnosed with heart disease but does not have it, while FN is a patient without heart disease diagnosed with it and it is the most dangerous as it negatively impacts the performance of the model. The mathematical notation for these evaluation indices is provided in Equations (1)-(4) below respectively.

$$Accuracy = \frac{TP + TN}{(TP + TN + FP + FN)} \tag{1}$$

$$Precision = \frac{TP}{(TP + FP)} \tag{2}$$

$$Recall = \frac{TP}{(TP + FN)} \tag{3}$$

$$F1Score = \frac{2\,(Precision \times Recall)}{(Precission + Recall)} \tag{4}$$

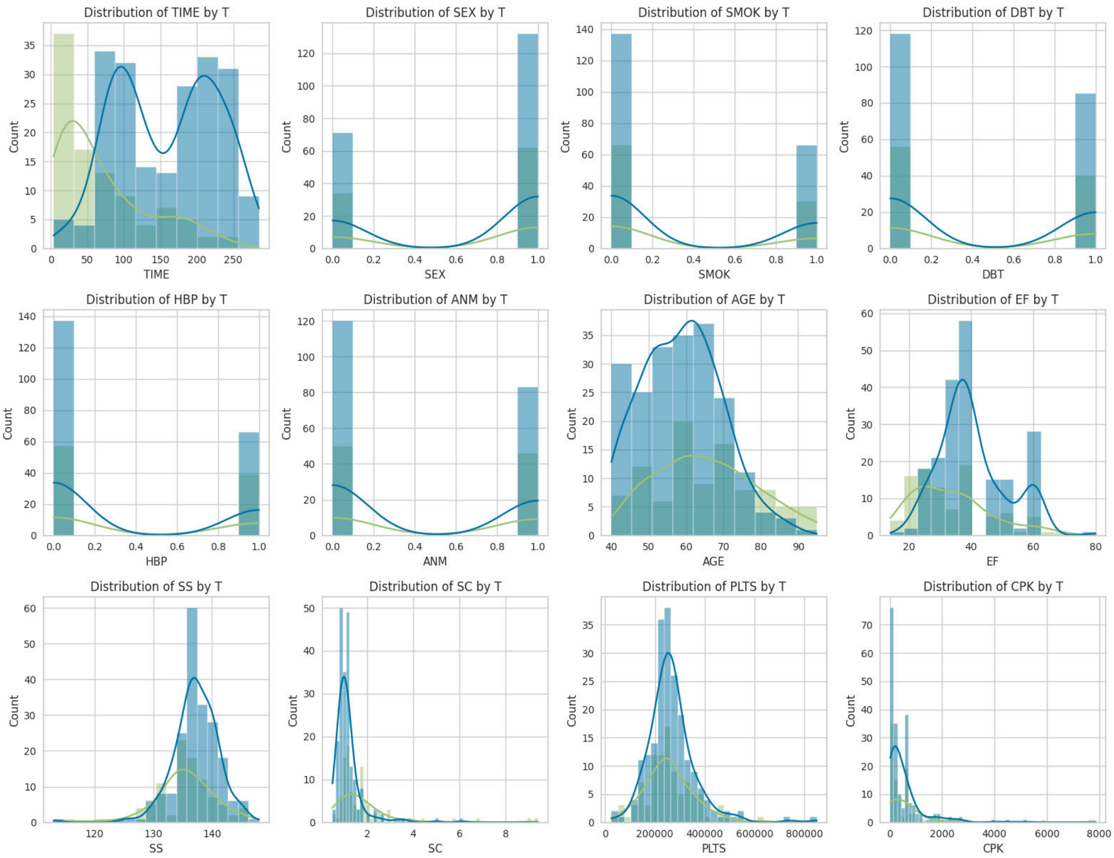

FIGURE 4. The Kaggle heart failure data distribution of input features by target feature T.

IV. METHODOLOGY

In this section, the proposed HypGB methodology for predicting heart diseases is presented. We first present an overview of the methodology in subsection A. The detailed procedure is presented in Section B.

A. OVERVIEW

The proposed methodology is a comprehensive approach for designing an automatic heart disease prediction system. It integrates the LASSO FS, Gradient Boosting (GB), and the HyperOpt optimization framework. The overall goal is to predict heart disease efficiently while increasing the generalizability so that it can be easily used in the healthcare system. Fig. 5 shows an overview of our proposed methodology which includes the following steps:

- Initially, the data preprocessing was conducted with respect to the selected Cleveland heart disease and Kaggle heart failure datasets to improve their representation for the classification. In order to handle the missing values, we used a *simpleimputer* from the scikit-learn library with a mean strategy. We used one-hot encoding, which transforms the categorical data into the numerical data, allowing the classifier to learn more effectively. Similarly, we used Min-Max scalar normalization which normalized the feature values into a consistent range between 0 and 1.
- The LASSO FS technique was then used to identify and select the most relevant features during the training. This helped prevent the model from the overfitting and could lead to improved performance on unseen data. The datasets were then split into training and testing sets with ratios of 80% and 20%, respectively.
- The GB model was trained, and its performance was analyzed using the full features and the selected feature subsets. The HyperOpt optimization framework was then used to optimize the hyperparameters of the GB using the training set. This process obtained the optimal set of hyperparameters of the GB model, and was evaluated to measure its performance using the testing set. This process is named HypGB.
- The output of the HypGB system predicts the heart disease as a binary yes or no.

B. DETAILED PROCEDURE

The procedure for the HypGB system to obtain the optimized GB model with the best hyperparameters involves the following steps:

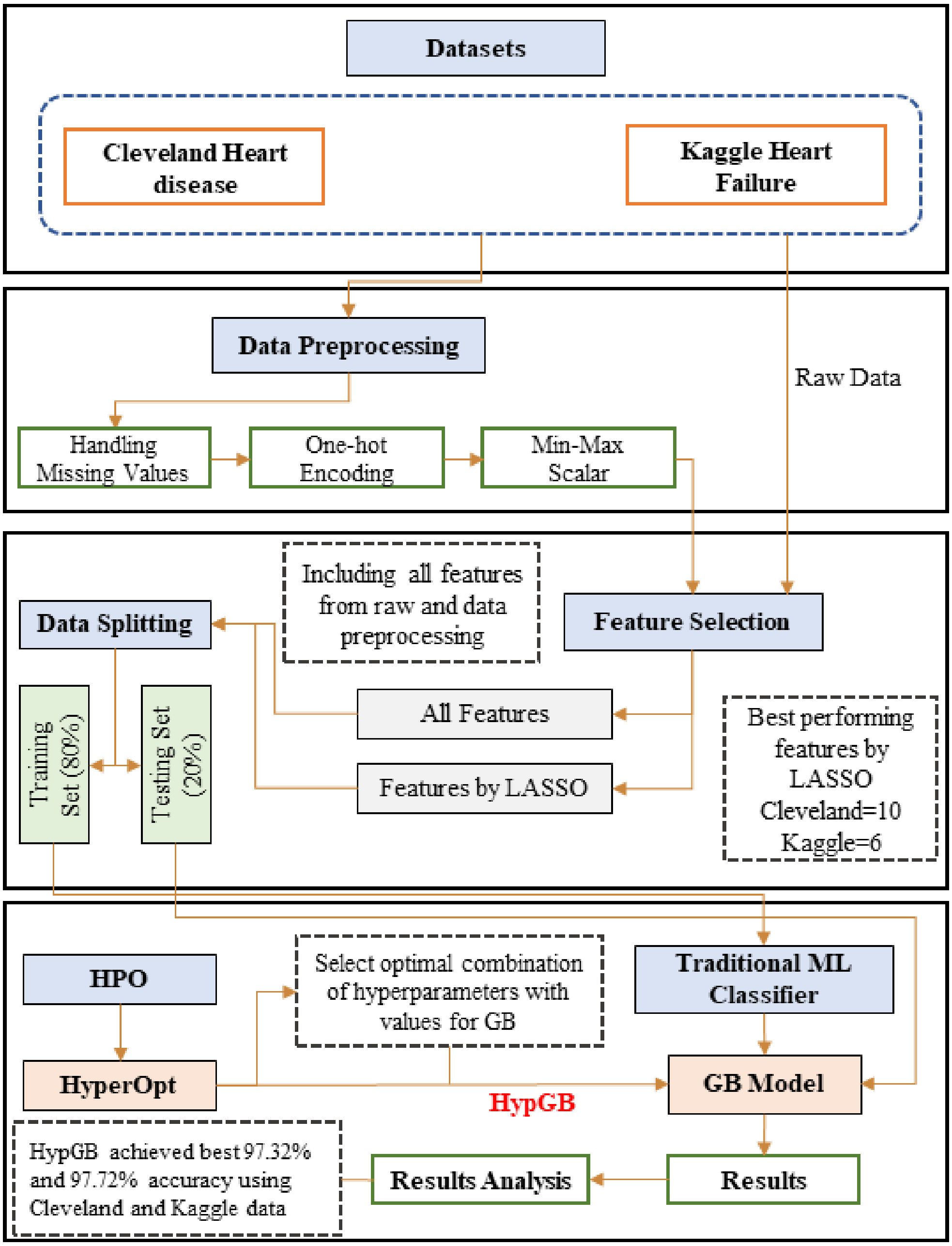
Datasets
Cleveland Heart disease
Kaggle Heart Failure
Data Preprocessing
Raw Data
Handling Missing Values
One-hot Encoding
Min-Max Scalar
Including all features from raw and data preprocessing
Data Splitting
Feature Selection
All Features
Features by LASSO
Best performing features by LASSO Cleveland=10 Kaggle=6
Training Set (80%)
Testing Set (20%)
HPO
Select optimal combination of hyperparameters with values for GB
Traditional ML Classifier
HyperOpt
HypGB
GB Model
HypGB achieved best 97.32% and 97.72% accuracy using Cleveland and Kaggle data
Results Analysis
Results

FIGURE 5. Overview of our HypGB methodology to predict the heart disease.

i) We have dataset $D(X, Y)$, where X is the input features and Y is a target variable (see Equations (5) and (6)):

$$D(X, Y) = \{(X_i, Y_i) \mid X_i \varepsilon M_n, Y_i \varepsilon \{0, 1\}\} k_{i=1} \tag{5}$$

where X_i can be

$$X_i = \{X_1, X_2, X_3, \ldots, X_n\} \tag{6}$$

ii) We apply Min-Max normalization to the $D(X, Y)$ to ensure consistent scaling across the features. This normalization is represented by Equation (7) below [18]. This helps in preparing the dataset for the subsequent analysis.

$$V' = V - \frac{min_{(a)}}{max_{(a)} - min_{(a)}} \left(new_{max(a)} - new_{min(a)}\right) + new_{min(a)} \tag{7}$$

In Equation 7, V is the original value of a feature and V' represents the normalized value. Similarly, $min_{(a)}$ and $max_{(a)}$ are the minimum and maximum values of the feature in the dataset, whereas $new_{\min(a)}$ and $new_{\max(a)}$ exhibit the desired minimum and maximum values for the normalized range respectively.

TABLE 2. Cleveland heart disease dataset [29].

No.	Feature expression	Description	Domain	Data type
1	Age	Age in years	[29 – 77]	Real
2	Sex	Gender	0,1	Binary
3	CP	CP types	1,2,3,4	Nominal
4	ResBP	ResBP in mmHg	[94-200]	Real
5	SC	SC in mg/dl	[126 – 564]	-
6	FastBP	FastBP>120mg/dl	0,1	Binary
7	RestECG	RestECG results	0,1,2	Nominal
8	MaxHR	-	[71 – 202]	Integer
9	ExeIA	-	0,1	Binary
10	OldPST	Depression	[0 - 6.2]	Real
11	SlopePE	Peak slope ways	1,2,3	Nominal
12	MajorVC	Major vessels	0,1,3,4	Real
13	TS	Types of defects	3,6,7	Nominal
14	T	Heart disease patient, healthy	0,1	Binary

TABLE 3. Kaggle heart failure dataset [30].

No.	Feature expression	Description	Domain	Data type
1	AGE	Years	[40 – 95]	Real
2	SEX	Gender	0,1	Binary
3	ANM	Deficiency of hemoglobin	0,1	-
4	SMOK	Patient with a smoking habit	0,1	-
5	SS	SS in mEq/L	[114 - 148]	Real
6	PLTS	PLTS count (kilo platelets/mL)	[25.01- 850.0]	-
7	TIME	-	[4 - 285]	-
8	DBT	Patient with diabetes or not	0,1	Binary
9	HBP	Patients with high HBP	0,1	-
10	EF	EF in percentage	[14 - 0.80]	Real
11	CPK	CPK enzyme	[23 - 0.786]	-
12	SC	SC in blood	[0.50 - 0.9]	-
13	T	Heart disease presence, absence	0,1	Binary

iii) The preprocessed dataset $D(X, Y)$ is split into the training set (D_{train}) and testing set (D_{tes}). We perform the LASSO FS approach to select an important feature subset. LASSO computes the relevance score of each feature and ranks them based on their relevancy to the target class.

iv) The GB model is trained using D_{train} with selected feature subsets and its performance is evaluated on D_{tes} to obtain the initial score, denoted as S_{init}.

v) To keep track of the hyperparameters and their corresponding scores, history (h) is initialized. For each evaluation, a sample set of hyperparameters (x) from the search space (S) using a search algorithm (A) is evaluated. If x is not yet evaluated, train the GB model *(GB_x)* on D_{train} using selected features (F) and hyperparameters *x*. Evaluate GB_x on D_{test} to obtain a score *y* and update *h.*

vi) The prediction is updated by combining the previous model $F_{t-1}(x)$ and the current model $G_t(x)$, weighted by λ (see Equation (8)).

$$F_t(x) = F_{t-1}(x) + \lambda G_t(x) \tag{8}$$

vii) Train optimized GB (GB_{opt}) using the x best hyperparameters and evaluate its performance on D_{tes} to obtain the final score, denoted as S_{final}.

viii) If the obtained S_{final} is better than the S_{init},GB_{opt} is returned; otherwise, the GB classifier is returned.

The pseudocode of the above procedure is shown in Algorithm 1 also.

V. EXPERIMENTAL RESULTS

In this section, experimental results and their comprehensive analyses are presented. We first outline the design setup used for conducting a series of experiments in Subsection A. Subsequently, in Subsections B and C, we provide the results of the GB classifier using the preprocessed data and the subset of features obtained through LASSO FS. In Subsection D, the performance of the HyperOpt method is evaluated using various evaluation metrics. In Subsection E, we compare the performance of the HypGB system with that of the previous research.

A. DESIGN SETUP

We conducted a series of experiments to classify heart diseases and evaluated the performance of the HypGB system using the Cleveland and the Kaggle datasets. First, the GB classifier was implemented on the raw datasets, including all the features. Second, the datasets were preprocessed using multiple techniques to help the model learn better and to make more accurate predictions. The LASSO FS algorithm was then used to extract the relevant features from the datasets. The GB classifier was then evaluated on the selected set of features to obtain better performance. Finally, the GB classifier was optimized using the HyperOpt optimization framework. Multiple evaluation metrics were computed to analyze the performance of the HypGB. All the experiments were conducted using multiple machine learning libraries in the Python language on an Intel (R) Core i7-2600 CPU @3.40 GHz system.

Algorithm 1 Pseudocode of Proposed HypGB System

Require: Dataset $D(X, Y)$, Selected features by F by LASSO,Search space S, Search algorithm A, Max. number of evaluations **N**
Ensure: Optimized GB Classifier GB_{opt}

```
1: Preprocess $D$ and split $D$ into $D_{train}$, $D_{test}$
2: Train $GB$ on $D_{train}$ using $F$
3: Evaluate the performance of $GB$ on $D_{test}$ and get the initial score $S_{init}$
4: Initialize history $h$ of hyperparameters with the score: $h = [\,]$
5: For $i$ in range $N$ do
6:     Sample a set of hyperparameters $x$ from $S$ using $A$
7:     If $x$ has been evaluated before, skip to step 6 then
8:         Train $GB$ model on $GB_x$ on $D_{train}$ using $F$ and hyperparameters $x$
9:         Evaluate the performance of $GB_x$ on $D_{test}$ and obtain a score $y$
10:        Update the history with the new hyperparameters and score: $h[x] = y$
11:    end if
12: end for
13: Update the ensemble: $F_t(x) = F_{t-1}(x) + \lambda\ G_t(x)$, where $G_t(x)$ is the prediction function of $GB_x$ with hyperparameters selected from $h$ using the best-performing $x$
14: Train $GB_{opt}$ on $D_{train}$ using $F$ and $x$
15: Evaluate the performance of $GB_{opt}$ on $D_{test}$ and obtain a final score $S_{final}$
16: If $S_{final}$ is better than $S_{init}$ then
17:     return $GB_{opt}$
18: Else
19:     return $GB$
20: end if
```

B. RESULTS OF DATA PREPROCESSING

The results obtained by the GB model on the raw and preprocessed datasets are presented in this subsection. For the raw datasets, we considered all the top 13 input features of the Cleveland and the 12 features of the Kaggle datasets to investigate the presence of the disease. The GB model achieved accuracies of 85.24% and 85% on the Cleveland and the Kaggle datasets, respectively, as shown in Table 4. We then applied multiple data preprocessing techniques, as explained earlier in Section IV.A. First, we separated the input features and the target labels. A target or a label feature highlights the presence or absence of a heart disease in a patient. Next, we handled the missing values and applied the Min-Max Scalar to the input data. Furthermore, one-hot encoding was applied to the categorical features so that the machine learning model could make better predictions. After applying the preprocessing techniques, the GB model improved the accuracy by 1.61% and 3.33% on the Cleveland and the Kaggle datasets respectively. In Table 4, preprocessed results highlight that the preprocessing approaches applied to clinical data enhanced the robustness of the model and improved the prediction performance. These preprocessed data are valuable for further experiments in the proposed system. Similarly, other evaluation metrics were improved with a valuable margin, which shows that the model improved the overall performance on the preprocessed data. The improved results of the GB model using preprocessed data are summarized in Table 4.

TABLE 4. Results of GB model using raw and preprocessed data (Metrics (%), Prediction time (seconds)).

	Raw Data		Preprocessed Data	
Metrics	Clevelan d dataset	Kaggle dataset	Clevelan d dataset	Kaggle dataset
Accuracy	85.24	85.00	86.85	88.33
Precision	91.66	68.42	91.85	82.35
Recall	84.61	81.25	87.17	77.77
F1-score	87.99	74.28	89.47	79.99
AUC-ROC	85.48	83.80	86.77	85.31
Prediction Time (sec)	6.74	4.57	3.73	7.43

C. RESULTS OF LASSO FS APPROACH

We used the LASSO feature selection method to select relevant and important features from both the datasets. LASSO conducts a binary classification and categorizes the most important features as true and the remaining as false. Table 5 shows the important features selected by the LASSO. For the Cleveland dataset, we selected the top 10 features with the highest MSE scores. CP, MajorVC, OldPST, Sex, and TS are the most appropriate features that contribute to the prediction of heart disease. Fig. 6 illustrates these important features along with their importance scores and ranking. Similarly, out of 12 features in the Kaggle heart failure dataset, we selected the eight most important features as shown in Table 5. TIME, EF, SC, AGE, SS, CPK, SEX, and SMOK features have higher importance scores compared with the remaining features. The selected features are ranked according to their importance scores. The "– "in Table 5 indicates that the rows are empty as we only selected eight features. Fig. 7 illustrates the important features with their importance scores and rankings. The 'Y' axis represents the importance score of each feature achieved by LASSO FS, whereas the 'X' axis represents the selected features. The order of each feature was based on their contribution toward the final heart disease prediction.

Table 6 shows the GB classifier results obtained using the subset of features selected by LASSO. The model achieved its highest level of performance on the Cleveland dataset, obtaining 90.16% accuracy, 93.10% precision, 87.09% recall, 90% F10-score, and 90.21% AUC-ROC score. The 93.10% precision shows that the GB model identified healthy people

TABLE 5. Feature importance score of cleveland and kaggle heart disease dataset obtained using LASSO FS method and their order.

Cleveland Dataset			Kaggle Dataset		
Feature	Order	Importance Score	Feature	Order	Importance Score
CP_0	1	0.1030	TIME	1	0.2040
MajorVC	2	0.0906	EF	2	0.1061
OldPST	3	0.0601	SC	3	0.0816
Sex	4	0.0580	AGE	4	0.0577
TS_2	5	0.0578	SS	5	0.0282
TS_3	6	0.0525	CPK	6	0.0218
ExeIA	7	0.0501	SEX	7	0.0172
SlopePE_1	8	0.0452	SMOK	8	0.0162
MaxHR	9	0.0442	-	-	-
ResBP	10	0.0180	-	-	-

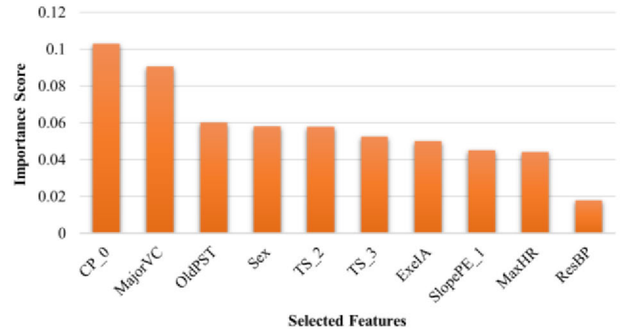

FIGURE 6. Feature importance score of selected features of cleveland dataset using LASSO FS method.

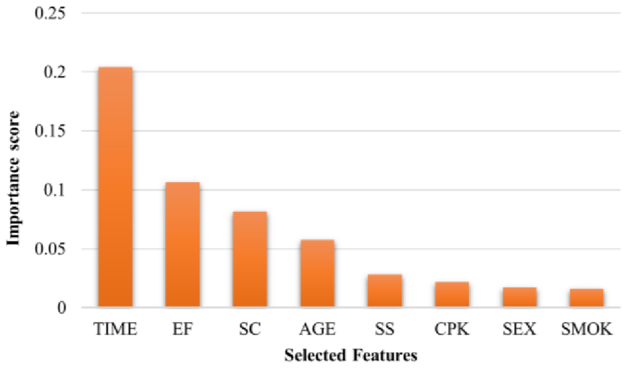

FIGURE 7. Feature importance score of selected features of Kaggle dataset using LASSO FS method.

well. The prediction time of the GB model on the selected features was 3.64 seconds which is quite fast. The overall evaluation metrics results show that the GB model was able to learn better using the selected subset of features and improved the classification accuracy by 3.31% compared with the preprocessed data. Similarly, the GB model obtained 91.66% accuracy and 100% precision for the Kaggle heart failure dataset with a prediction time of 4.62 seconds. Overall, the GB model learned better using the important features extracted by LASSO by improving the accuracy by 3.33% for the Kaggle dataset.

TABLE 6. Results of GB using features selected by LASSO FS method (Metrics (%), prediction time (seconds)).

Metrics	Cleveland dataset	Kaggle dataset
Accuracy	90.16	91.66
Precision	93.10	100
Recall	87.09	72.22
F1-score	90.00	83.87
AUC-ROC	90.21	86.11
Prediction Time (sec)	3.64	4.26

D. RESULTS USING HypGB METHOD

For the experiments using our HypGB method, we first tuned the GB model using the HyperOpt optimization framework to obtain the optimal combinations of hyperparameters to enhance the performance of the model. The obtained configured set of hyperparameters for the GB model is provided in Table 7, which includes their data types and tuned value.[1] We applied the HyperOpt optimization to enhance its performance during the training on the selected subset of features using the LASSO experiment. The HypGB achieved 97.32% and 97.732% accuracy for the Cleveland and the Kaggle heart failure datasets, respectively, as shown in Table 8.

Fig. 8 presents a detailed performance comparison of all the experiments for the Cleveland dataset. The HypGB system outperformed the other experiments and accurately predicted the positive samples. Evaluation metrics, such as accuracy, are critical for evaluating performance. In the HypGB system, the optimized GB classifier achieved the highest accuracy of 97.32%, which was 7.16% improvement over the LASSO FS experiment. Both experiments used a subset of 10 features obtained using the LASSO FS method. For all 13 features, the GB classifiers achieved accuracies of 86.85% and 85.24% for the preprocessed and raw data experiments, respectively. The HypGB significantly improved the accuracy by 10.47% over the preprocessed

[1] *n_estimators* specifies the number of boosting stages that correct the errors of the previous tree.*max_depth* controls the maximum depth of each tree in the GB which is important for avoiding the overfitting. *max_leaf_nodes* controls the nodes in trees and helps manage the complexity of the model.*min_samples_leaf*sets a threshold to select the samples for a leaf which is useful for preventing the overfitting. *learning_rate* controls the behavior of GB trees which results in the robustness of the model.

TABLE 7. Optimal combinations of GB hyperparameters with range and tuned values.

Hyperparameters	Data Type	Range	Cleveland	Kaggle
n_estimators	integer	50 -500	250	350
max_depth	real	1 - 10	2	3
max_leaf_nodes	integer	1 - 10	3	5
min_samples_leaf	integer	1 - 25	20	15
learning_rate	float	-5 – 0	0.01408	0.01954
min_samples_split	integer	1 - 10	7	7
subsample	numeric	0.5 – 1	0.6059	0.5346
max_features	integer	sqrt, log2	sqrt	sqrt
random_state	real	1 - 100	32	7
loss	exponential, log	deviance, exponential	exponential	Exponential

TABLE 8. Results of our proposed HypGB system on the heart datasets (Metrics (%), Prediction time (seconds)).

Metrics	Cleveland dataset	Kaggle dataset
Accuracy	97.32	97.72
Precision	96.58	94.14
Recall	98.72	98.26
F1-score	97.88	96.81
AUC-ROC	96.60	94.88
Prediction Time (sec)	6.82	6.42

data and by 12.08% over the raw data. For the other evaluation metrics, the HypGB approach also improved the performance and correctly identified the positive samples. The HypGB achieved higher recall and precision scores of 98.72% and 96.58%, respectively. The higher values showed that our approach correctly identified heart patients and improved recall by 14.11% over the raw data experiments. Our proposed HypGB achieved a higher F1-score and AUC-ROC of 97.88% and 96.60%, respectively, higher than those of all the other experiments. These results suggest that our HypGB system significantly improved the performance of the machine learning GB algorithm.

Similarly, Fig. 9 presents a performance comparison of all the experiments, including HypGB using the Kaggle heart failure dataset. The HypGB system achieved the highest accuracy of 97.72%, which is an improvement of 6.06% over the LASSO experiment. The GB model was trained on a subset of six features obtained using the LASSO FS method. The dataset is small, with limited features compared with the Cleveland. However, our approach still improved the performance and overcame the overfitting problem. In comparison with the raw and the preprocessed data experiments, our model used only eight features but still achieved higher results, as shown in Fig. 9. It shows that the HypGB improved the accuracy by 12.72% over the raw data and by 9.39% over the preprocessed data. Similar to the Cleveland data, the HypGB improved the scores of other evaluation metrics on the Kaggle dataset, too. It achieved 94.14% precision, 98.26% recall, 96.81% F1-score, and 94.88% AUC-ROC curve. Overall, the results show that the GB model optimized with the HyperOpt optimization improved the performance and achieved higher scores.

Table 9 shows the prediction time of the model while performing experiments using the raw data, the preprocessed data, the LASSO FS, and the HyperOpt optimization. The HypGB system took a prediction time of 6.82 seconds on the Cleveland heart disease dataset and 6.42 seconds on the Kaggle heart failure dataset, as shown in Table 9.

TABLE 9. Prediction time (seconds) of the GB model during the experiments.

Experiments	Prediction Time (sec)	
	Cleveland dataset	Kaggle dataset
Raw data	6.74	4.57
Preprocessed data	3.73	7.43
LASSO FS	3.64	4.26
HypGB	6.82	6.42

E. PERFORMANCE COMPARISON WITH EXISTING MODELS

In this subsection, we compare the performance of the HypGB system with the previous research and explain the advantages of our approach.

Table 10 and Fig. 10 show the performance comparison of the HypGB with the existing models from the previous research using the Cleveland dataset. Compared with all the previous research results, HypGB achieved 6.21% higher accuracy on average. Fig. 11 shows the curve of 96.6% AUC-ROC of our proposed model on the Cleveland dataset.

Table 11 and Fig. 12 show the performance comparison of the HypGB with the existing models from the previous research using the Kaggle heart failure dataset. Overall, the accuracy of the HypGB is superior to those of all the previous approaches. Compared with all the previous research results in Table 11, HypGB achieved 8% higher accuracy on average.

Overall, our proposed HypGB system has the following advantages over previous approaches:

- Our approach selected the top 10 most important features by incorporating the LASSO FS. This reduced

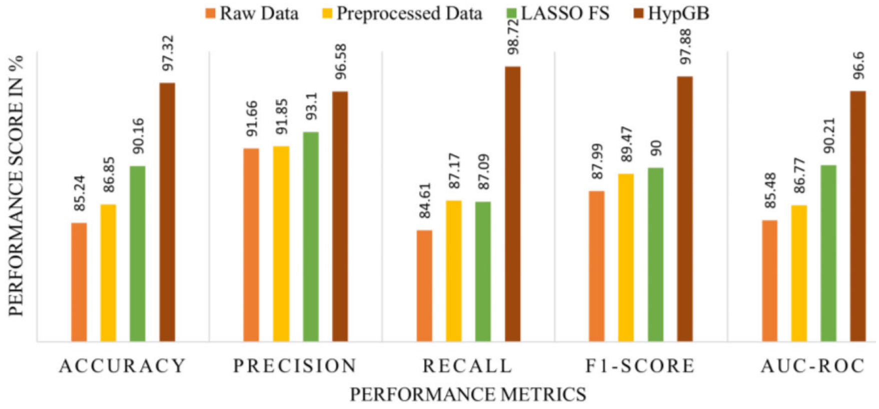

FIGURE 8. Detailed comparison of the GB models' performance for the Cleveland dataset.

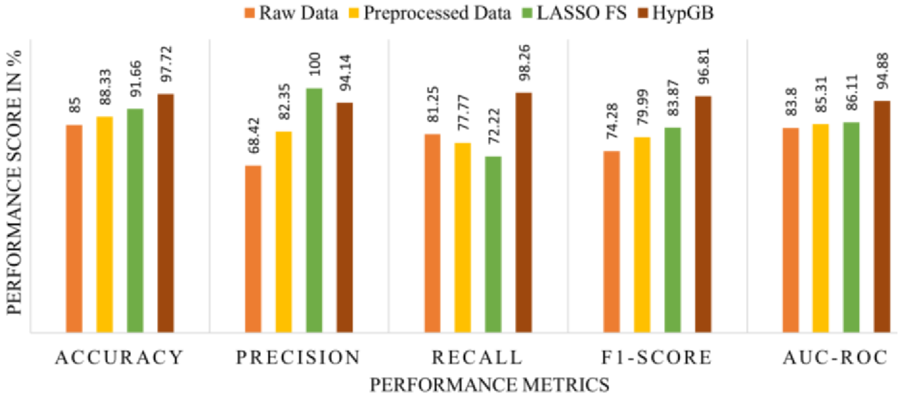

FIGURE 9. Detailed comparison of the GB models' performance for the Kaggle heart failure dataset.

TABLE 10. Performance comparison of the HypGB with the existing models on the cleveland dataset.

Study	Year	Methods	Accuracy (%)
Miao et al. [31]	2016	Adaptive Boosting	80.14
Shah et al. [11]	2020	BPNN+SVM	85.12
Mohan et al. [12]	2019	HRFLM	88.7
Sharma et al. [32]	2020	Talos HPO	90.78
Li et.al [14]	2020	FCMIM	92.37
Nahiduzzaman et al. [33]	2019	SVM	92.45
Farzana et al. [34]	2021	RF with PCA	92.85
Javeed et al. [15]	2019	RSA+RF	93.33
Proposed	**2023**	**HypGB**	**97.32**

TABLE 11. Performance comparison of the HypGB with the existing models on the Kaggle dataset.

Study	Year	Methods	Accuracy (%)
Mamun et al. [35]	2022	LightGBM	85
Priyadarshinee et al. [36]	2022	DTNB	87.08
Srinivas et al. [37]	2022	hyOPTXg	89.3
Nishat et al. [38]	2022	SMOTE-ENN	90
Abdellatif et al. [39]	2022	Inf-FSs-IWRF	97.2
Proposed	**2023**	**HypGB**	**97.72**

the dimensionality and allowed the model to focus on the informative aspects of the data. This also improved the performance of the model.

- Our approach can handle complex problems better by employing the GB classifier, because it performed well in heart disease prediction tasks.
- Our approach combined the LASSO FS with the HyperOpt optimization of the GB model to predict heart disease, whereas the previous studies typically focused on individual feature selection or the use of simple optimization methods such as grid search, random search, etc. This combination enhanced the performance, surpassing the limitations of the previous approaches.

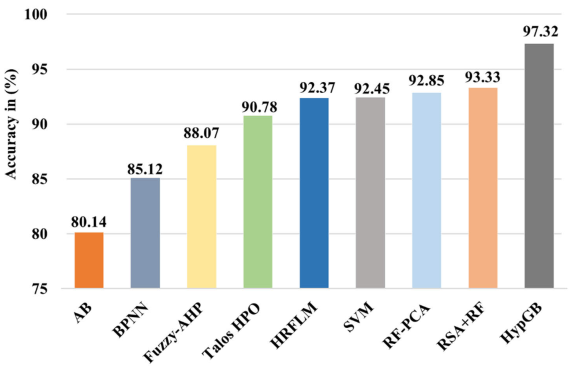

FIGURE 10. Performance comparison of HypGB with existing models using the cleveland dataset.

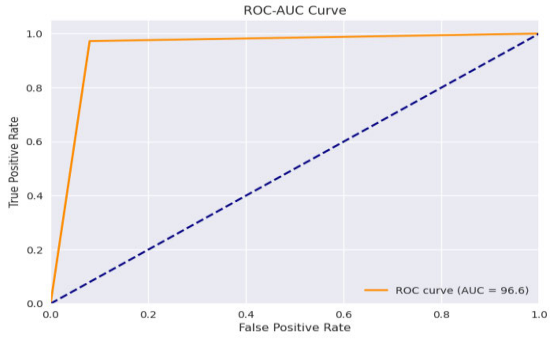

FIGURE 11. AUC-ROC curve of the proposed HypGB on the cleveland dataset.

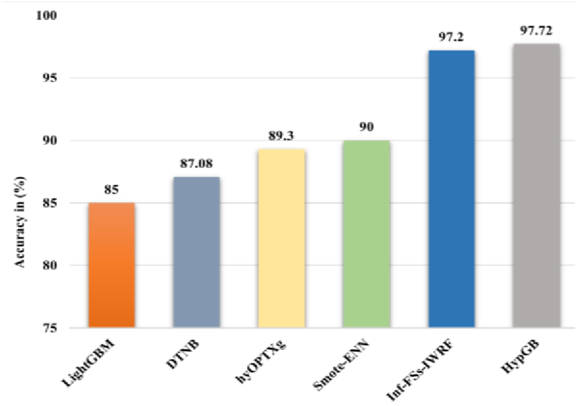

FIGURE 12. Performance comparison of HypGB with existing models using the Kaggle dataset.

- With the highest accuracy during training and testing, HypGB minimized the overfitting and improved the interpretability of the model.
- The selected features using the FS approach improve the diagnostic accuracy, diagnostic efficiency, and clinical utility of heart disease risk assessment. It offers better patient outcomes and more efficient use of available resources in healthcare institutions, while also simplifying the decision-making process for medical practitioners. This study can improve the early detection and risk assessment of heart disease by integrating the results of machine learning into real clinical applications.

VI. CONCLUSION

In this paper, an automatic HypGB heart disease prediction system was proposed. This system was tested on two heart disease datasets, namely Cleveland heart disease and Kaggle heart failure. A Gradient-Boosting (GB) classifier was used to detect patients with heart disease. We first cleaned the dataset using the data preprocessing techniques, and then extracted the relevant features using the LASSO FS approach. The dataset with important features was divided into training and testing sets respectively. The HyperOpt hyperparameter optimization was conducted to enhance the performance of the GB model by updating the parameters. Our proposed HypGB system achieved 97.32% accuracy on the Cleveland heart disease dataset and 97.72% accuracy on the Kaggle heart failure dataset, outperforming all the other existing machine learning models. Based on the experimental results, we believe that our HypGB system increases the chances of correctly predicting heart patients and can be possibly installed in healthcare systems. In the future, we plan on extending the proposed approach to ensure it is compatible with a wider range of FS algorithms and enables robust performance across high-dimensional datasets with a larger number of missing values. Furthermore, we will investigate the most recent deep-learning applications to enhance heart disease prediction.

REFERENCES

[1] *Cardiovascular Diseases*. Accessed: Feb. 1, 2023. [Online]. Available: https://www.who.int/health-topics/cardiovascular-diseases

[2] *Classes of Heart Failure | American Heart Association*. Accessed: Feb. 1, 2023. [Online]. Available: https://www.heart.org/en/health-topics/heart-failure/what-is-heart-failure/classes-of-heart-failure

[3] S. Ghwanmeh, A. Mohammad, and A. Al-Ibrahim, ''Innovative artificial neural networks-based decision support system for heart diseases diagnosis,'' *J. Intell. Learn. Syst. Appl.*, vol. 5, no. 3, pp. 176–183, 2013.

[4] M. M. Ahsan and Z. Siddique, ''Machine learning-based heart disease diagnosis: A systematic literature review,'' *Artif. Intell. Med.*, vol. 128, Jun. 2022, Art. no. 102289.

[5] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. New York, NY, USA: Cambridge Univ. Press, 2014, pp. 101–113.

[6] P. Melillo, N. De Luca, M. Bracale, and L. Pecchia, ''Classification tree for risk assessment in patients suffering from congestive heart failure via long-term heart rate variability,'' *IEEE J. Biomed. Health Informat.*, vol. 17, no. 3, pp. 727–733, May 2013.

[7] M. M. A. Rahhal, Y. Bazi, H. AlHichri, N. Alajlan, F. Melgani, and R. R. Yager, ''Deep learning approach for active classification of electrocardiogram signals,'' *Inf. Sci.*, vol. 345, pp. 340–354, Jun. 2016.

[8] C. R. Olsen, R. J. Mentz, K. J. Anstrom, D. Page, and P. A. Patel, ''Clinical applications of machine learning in the diagnosis, classification, and prediction of heart failure,'' *Amer. Heart J.*, vol. 229, pp. 1–17, Nov. 2020.

[9] Q. Hang, J. Yang, and L. Xing, ''Diagnosis of rolling bearing based on classification for high dimensional unbalanced data,'' *IEEE Access*, vol. 7, pp. 79159–79172, 2019.

[10] J. Petch, S. Di, and W. Nelson, ''Opening the black box: The promise and limitations of explainable machine learning in cardiology,'' *Can. J. Cardiol.*, vol. 38, no. 2, pp. 204–213, Feb. 2022.

[11] S. M. S. Shah, F. A. Shah, S. A. Hussain, and S. Batool, "Support vector machines-based heart disease diagnosis using feature subset, wrapping selection and extraction methods," *Comput. Electr. Eng.*, vol. 84, Jun. 2020, Art. no. 106628.

[12] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81542–81554, 2019.

[13] X. Liu, X. Wang, Q. Su, M. Zhang, Y. Zhu, Q. Wang, and Q. Wang, "A hybrid classification system for heart disease diagnosis based on the RFRS method," *Comput. Math. Methods Med.*, vol. 2017, 2017, Art. no. e8272091.

[14] J. P. Li, A. U. Haq, S. U. Din, J. Khan, A. Khan, and A. Saboor, "Heart disease identification method using machine learning classification in e-healthcare," *IEEE Access*, vol. 8, pp. 107562–107582, 2020.

[15] A. Javeed, S. Zhou, L. Yongjian, I. Qasim, A. Noor, and R. Nour, "An intelligent learning system based on random search algorithm and optimized random forest model for improved heart disease detection," *IEEE Access*, vol. 7, pp. 180235–180243, 2019.

[16] Y. Li, T. Li, and H. Liu, "Recent advances in feature selection and its applications," *Knowl. Inf. Syst.*, vol. 53, no. 3, pp. 551–577, Dec. 2017.

[17] P. Li, "Robust LogitBoost and adaptive base class (ABC) LogitBoost," 2012, *arXiv:1203.3491*.

[18] J. Bergstra, B. Komer, C. Eliasmith, D. Yamins, and D. D. Cox, "Hyperopt: A Python library for model selection and hyperparameter optimization," *Comput. Sci. Discovery*, vol. 8, no. 1, Jul. 2015, Art. no. 014008.

[19] J. Zhang, Q. Wang, and W. Shen, "Hyper-parameter optimization of multiple machine learning algorithms for molecular property prediction using hyperopt library," *Chin. J. Chem. Eng.*, vol. 52, pp. 115–125, Dec. 2022.

[20] M. Gudadhe, K. Wankhade, and S. Dongre, "Decision support system for heart disease based on support vector machine and artificial neural network," in *Proc. Int. Conf. Comput. Commun. Technol. (ICCCT)*, Sep. 2010, pp. 741–745.

[21] H. Kahramanli and N. Allahverdi, "Design of a hybrid system for diabetes and heart diseases," *Exp. Syst. Appl.*, vol. 35, nos. 1–2, pp. 82–89, 2008.

[22] S. Palaniappan and R. Awang, "Intelligent heart disease prediction system using data mining techniques," in *Proc. IEEE/ACS Int. Conf. Comput. Syst. Appl.*, Mar. 2008, pp. 108–115.

[23] M. Jabbar, B. Deekshatulu, and P. Chandra, "Classification of heart disease using artificial neural network and feature subset selection," *Glob. J. Comput. Sci. Technol.*, vol. 13, no. 3, pp. 4–8, 2013.

[24] S. Nandy, M. Adhikari, V. Balasubramanian, V. G. Menon, X. Li, and M. Zakarya, "An intelligent heart disease prediction system based on swarm-artificial neural network," *Neural Comput. Appl.*, vol. 35, no. 20, pp. 14723–14737, Jul. 2023.

[25] O. Terrada, S. Hamida, B. Cherradi, A. Raihani, and O. Bouattane, "Supervised machine learning based medical diagnosis support system for prediction of patients with heart disease," *Adv. Sci., Technol. Eng. Syst. J.*, vol. 5, no. 5, pp. 269–277, 2020.

[26] P. Ghosh, S. Azam, M. Jonkman, A. Karim, F. M. J. M. Shamrat, E. Ignatious, S. Shultana, A. R. Beeravolu, and F. De Boer, "Efficient prediction of cardiovascular disease using machine learning algorithms with relief and LASSO feature selection techniques," *IEEE Access*, vol. 9, pp. 19304–19326, 2021.

[27] C. Zhou and A. Wieser, "Jaccard analysis and LASSO-based feature selection for location fingerprinting with limited computational complexity," in *Progress in Location Based Services*, vol. 14. Springer, 2018, pp. 71–87.

[28] A. Jafar and M. Lee, "Comparative performance evaluation of state-of-the-art hyperparameter optimization frameworks," *Trans. Korean Inst. Electr. Eng.*, vol. 72, no. 5, pp. 607–620, May 2023.

[29] *UCI Machine Learning Repository: Heart Disease Data Set*. Accessed: Feb. 15, 2023. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease

[30] D. Chicco and G. Jurman, "Machine learning can predict survival of patients with heart failure from serum creatinine and ejection fraction alone," *BMC Med. Informat. Decis. Making*, vol. 20, no. 1, p. 16, Dec. 2020.

[31] K. H. Miao, J. H. Miao, and G. J. Miao, "Diagnosing coronary heart disease using ensemble machine learning," *Int. J. Adv. Comput. Sci. Appl.*, vol. 7, no. 10, pp. 30–39, 2016.

[32] S. Sharma and M. Parmar, "Heart diseases prediction using deep learning neural network model," *Int. J. Innov. Technol. Exploring Eng.*, vol. 9, no. 3, pp. 2244–2248, Jan. 2020.

[33] M. Nahiduzzaman, M. J. Nayeem, M. T. Ahmed, and M. S. U. Zaman, "Prediction of heart disease using multi-layer perceptron neural network and support vector machine," in *Proc. 4th Int. Conf. Electr. Inf. Commun. Technol. (EICT)*, Dec. 2019, pp. 1–6.

[34] F. Tasnim and S. U. Habiba, "A comparative study on heart disease prediction using data mining techniques and feature selection," in *Proc. 2nd Int. Conf. Robot., Electr. Signal Process. Techn. (ICREST)*, Jan. 2021, pp. 338–341.

[35] M. Mamun, A. Farjana, M. A. Mamun, M. S. Ahammed, and M. M. Rahman, "Heart failure survival prediction using machine learning algorithm: Am i safe from heart failure?" in *Proc. IEEE World AI IoT Congr. (AIIoT)*, Jun. 2022, pp. 194–200.

[36] S. Priyadarshinee and M. Panda, "Improving prediction of chronic heart failure using SMOTE and machine learning," in *Proc. 2nd Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA)*, Sep. 2022, pp. 1–6.

[37] P. Srinivas and R. Katarya, "HyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost," *Biomed. Signal Process. Control*, vol. 73, Mar. 2022, Art. no. 103456.

[38] M. M. Nishat, F. Faisal, I. J. Ratul, A. Al-Monsur, A. M. Ar-Rafi, S. M. Nasrullah, M. T. Reza, and M. R. H. Khan, "A comprehensive investigation of the performances of different machine learning classifiers with SMOTE-ENN oversampling technique and hyperparameter optimization for imbalanced heart failure dataset," *Sci. Program.*, vol. 2022, Mar. 2022, Art. no. e3649406.

[39] A. Abdellatif, H. Abdellatef, J. Kanesan, C.-O. Chow, J. H. Chuah, and H. M. Gheni, "Improving the heart disease detection and patients' survival using supervised infinite feature selection and improved weighted random forest," *IEEE Access*, vol. 10, pp. 67363–67372, 2022.

[40] A. Saboor, M. Usman, S. Ali, A. Samad, M. F. Abrar, and N. Ullah, "A method for improving prediction of human heart disease using machine learning algorithms," *Mobile Inf. Syst.*, vol. 2022, pp. 1–9, Mar. 2022.

ABBAS JAFAR received the B.S. degree in software engineering from the Government College University Faisalabad, Pakistan, and the master's degree from Myongji University, South Korea, in 2020, where he is currently pursuing the Ph.D. degree. He is a Research Assistant with the HPC Laboratory, Myongji University. His research interests include AI in healthcare systems, machine learning, deep learning, high-performance computing, and performance optimization with a special interest in GPU computing.

MYUNGHO LEE (Member, IEEE) received the B.S. degree in computer science and statistics from Seoul National University, South Korea, and the M.S. degree in computer science and the Ph.D. degree in computer engineering from the University of Southern California, Los Angeles, CA, USA. He was a Staff Engineer with the Scalable Systems Group, Sun Microsystems, Sunnyvale, CA, USA. He is currently a Full Professor with the Department of Computer Science and Engineering, Myongji University. His research interest includes high-performance computing, such as architecture, compilers, and applications.