IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 19 September 2023, accepted 13 October 2023, date of publication 18 October 2023, date of current version 25 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3325681

RESEARCH ARTICLE

Heart Disease Prediction Using Stacking Model With Balancing Techniques and Dimensionality Reduction

AYESHA NOOR[1], NADEEM JAVAID[1], (Senior Member, IEEE), NABIL ALRAJEH[2], BABAR MANSOOR[3], ALI KHAQAN[3], AND SAFDAR HUSSAIN BOUK[4]

[1]Department of Computer Science, COMSATS University Islamabad, Islamabad 44000, Pakistan
[2]Department of Biomedical Technology, College of Applied Medical Sciences, King Saud University, Riyadh 11633, Saudi Arabia
[3]Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad 44000, Pakistan
[4]School of Cybersecurity, Old Dominion University, Norfolk, VA 23529, USA

Corresponding author: Nadeem Javaid (nadeemjavaidqau@gmail.com)

This work was supported by King Saud University, Riyadh, Saudi Arabia, through the Researchers Supporting Project under Grant RSP2023R648.

ABSTRACT Heart disease is a serious worldwide health issue with wide-reaching effects. Since heart disease is one of the leading causes of mortality worldwide, early detection is crucial. Emerging technologies like Machine Learning (ML) are currently being actively used by the biomedical, healthcare, and health prediction industries. PaRSEL, a new stacking model is proposed in this research, that combines four classifiers, Passive Aggressive Classifier (PAC), Ridge Classifier (RC), Stochastic Gradient Descent Classifier (SGDC), and eXtreme Gradient Boosting (XGBoost), at the base layer, and LogitBoost is deployed for the final predictions at the meta layer. The imbalanced and irrelevant features in the data increase the complexity of the classification models. The dimensionality reduction and data balancing approaches are considered very important for lowering costs and increasing the accuracy of the model. In PaRSEL, three dimensionality reduction techniques, Recursive Feature Elimination (RFE), Linear Discriminant Analysis (LDA), and Factor Analysis (FA), are used to reduce the dimensionality and select the most relevant features for the diagnosis of heart disease. Furthermore, eight balancing techniques, Proximity Weighted Random Affine Shadowsampling (ProWRAS), Localized Randomized Affine Shadowsampling (LoRAS), Random Over Sampling (ROS), Adaptive Synthetic (ADASYN), Synthetic Minority Oversampling Technique (SMOTE), Borderline SMOTE (B-SMOTE), Majority Weighted Minority Oversampling Technique (MWMOTE) and Random Walk Oversampling (RWOS), are used to deal with the imbalanced nature of the dataset. The performance of PaRSEL is compared with the other standalone classifiers using different performance measures like accuracy, F1-score, precision, recall and AUC-ROC score. Our proposed model achieves 97% accuracy, 80% F1-score, precision is greater than 90%, 67% recall, and 98% AUC-ROC score. This shows that PaRSEL outperforms other standalone classifiers in terms of heart disease prediction. Additionally, we deploy SHapley Additive exPlanations (SHAP) on our proposed model. It helps to understand the internal working of the model. It illustrates how much influence a classifier has on the final prediction outcome.

INDEX TERMS Data balancing, dimensionality reduction, heart disease, machine learning, prediction, stacking model.

I. INTRODUCTION

Each organ that is a part of the human body has a distinct function. The heart is one such organ that pumps blood

The associate editor coordinating the review of this manuscript and approving it for publication was Md Kafiul Islam.

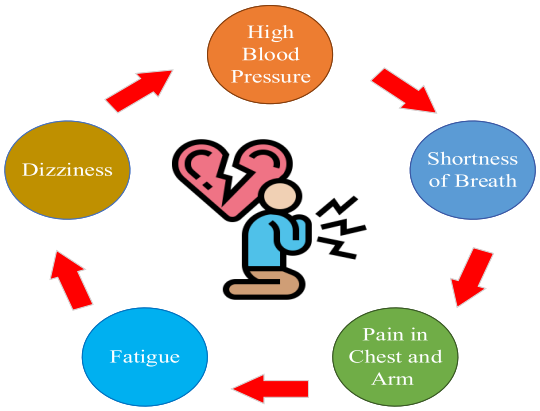


FIGURE 1. Major symptoms of heart attack.

throughout the body; if it malfunctions, this could lead to major health issues [1]. Heart disease is one of the leading causes of death on a massive scale worldwide. Major symptoms of heart attack are chest pain, fatigue, high blood pressure, pain in the arm and dizziness which are expressed in Figure 1. Professionals have recently enhanced their use of computer technology to improve decision-making support. The diagnosis of patients using Machine Learning (ML) is becoming increasingly important in the healthcare industry [2].

ML is an analytical technique that is employed when a task is difficult and complex to program, such as turning medical records into knowledge, forecasting pandemics, and analyzing genetic data. Understanding the large dataset in the healthcare industry is greatly helped by data mining. The extraction of patterns and significant information from the data enhances forecasts [3].

Treatment costs are reduced overall because ML algorithms are effective in identifying possibly at-risk patients early on. There are many ML classifiers that are used to diagnose heart illness. Regarding medical data, they must be dependable and function well in the healthcare industry. The early detection of cardiovascular disease before major issues arise is the main objective of these classifiers. The complications are significantly worse when a healthy patient is mistaken as having a problem than when a patient with heart illness is misdiagnosed.

ML techniques have been applied in research to diagnose heart issues. It has been shown that when the models are generated correctly, traditional classifiers perform effectively in terms of accuracy [4]. A variety of strategies can be used to improve the performance of these classifiers. In the work covered in this study, the adoption of the stacking model improves the performance of numerous algorithms because it has more than one stage for learning [5].

A. CONTRIBUTIONS

In this paper, the major contributions of our suggested model are as follows.

- A stacking model is designed using four classifiers at the base level and a classifier at the meta layer for the heart disease prediction.
- Useful information is extracted from the dataset using three different dimensionality reduction techniques.
- Data balancing is performed using eight balancing techniques to balance the dataset before classification.
- SHAP is deployed on the model to evaluate the importance of features influenced the model performance.
- Extensive simulations are performed using various performance metric to check the proposed model's efficacy.

II. RELATED WORK

To effectively treat and prevent heart failure, cardiac illness must be accurately and quickly diagnosed. ML algorithms are badly impacted by data drift, requiring constant monitoring and modification. The authors in [6] investigated the properties of data drift for predicting sepsis, assisting in creating efficient patient monitoring systems. This study uses Electronic Health Records (EHRs) to evaluate the impact of data drift on sepsis patients. The study shows that XGBoost outperforms baseline models.

In [7], the authors utilized traditional approaches which frequently suffer from loss of functionality and early convergence and historical medical data. The suggested technique reduces loss functions and prevents from getting stuck in local minima using a Levy Flight-Convolutional Neural Network (LV-CNN) and Sunflower Optimization Algorithm (SFO) for the diagnosis of heart issues. In a MATLAB simulation, the suggested technique achieves 95.74% accuracy. Millions of people are affected each year by the common and fatal condition known as Chronic Kidney Disease (CKD). Due to the lack of significant symptoms that serve as a benchmark, diagnosing CKD is difficult. Using information from 400 persons, a deep neural network-based Multi-Layer Perceptron (MLP) classifier was proposed to detect CKD in [8]. The approach outperformed traditional ML models like Support Vector Machines (SVM) and Naive Bayes (NB) in classification. Neural network models act as a superior alternative for categorizing chronic diseases.

The strategy for predicting diseases using ensemble deep learning is presented in [9], which blends supervised and unsupervised learning paradigms. It uses disease scoring techniques, builds a score selection mechanism, learns potential sample representations, and trains composite features with gradient boosting classifiers for the efficient diagnosis of heart failure. In [10], a classification strategy is suggested for people with heart disease and healthy subjects using ML algorithms. To improve classification accuracy and shorten calculation time, the system employs a sequential backward selection of feature techniques. 70% of the Cleveland heart disease dataset is used for training and the remaining 30% is used for validation in the assessment. The experimental findings demonstrate that the suggested model successfully distinguishes between patients having heart disease

and healthy patients. Using Principal Component Analysis (PCA), Agglomerative Hierarchical Clustering (AHC), and Random Forest (RF) in [11], the authors offer an Integrated Decision Making System (IDMS) for predicting cardiac disease. In comparison to previous methods, the system performs better than the six traditional categorization strategies. The technology can provide significant information on heart illness and assist clinicians in making accurate diagnoses of heart patients. Large-scale data generated by digitization in a variety of industries allows ML algorithms to identify patterns and make predictions. Not all qualities, though, are crucial for algorithmic training. The four well-known ML algorithms investigated in [12] are Decision Tree (DT), SVM, NB, and RF. The dimensionality reduction techniques investigated in this study are Linear Discriminant Analysis (LDA) and PCA. In all measurements, the results reveal that PCA beats LDA. For treating cardiovascular patients before heart failure, an accurate prognosis of heart disease is essential. Datasets related to cardiovascular disease can be processed using Artificial Intelligence (AI) methods like gradient boosting-based sequential feature selection to extract features. A comparison model is developed using ML methods in [13], outperforming existing frameworks and reaching a test accuracy of 98.78% with the stacking model. When it comes to forecasting cardiac sickness, this paradigm is more effective than other cutting-edge methods.

Coronary arteriography is an accurate invasive technique for diagnosing coronary heart disease. However, its invasive nature makes it unsuitable for annual physical examinations. In [14], authors use ML to integrate multiple algorithms and verify feature selection methods with personal clinical information. A two-level stacking-based model is designed with a meta-level input being the base-level classifiers' output. The model achieves accuracy, sensitivity, and specificity of 95.43%, 95.84%, and 94.44% respectively. This effectively aiding clinicians in detecting normal coronary arteries from those with Congenital Heart Disease (CHD). For making predictions based on historical data, data scientists employ ML, a potent approach. Weak algorithms' accuracy is increased through ensemble classification, a technique that mixes different classifiers. On a dataset of cardiac disease, experiments were carried out with the goal of enhancing prediction precision in [15]. The results show that ensemble techniques, such as bagging and boosting, efficiently increase the prediction accuracy of weak classifiers and determine the risk of heart disease. Ensemble classification yielded an accuracy boost of up to 7%, while feature selection implementation further improved the method, leading to considerable gains in prediction accuracy.

Atherosclerotic plaques on the coronary arteries are the primary cause of Coronary Artery Disease (CAD), an illness that affects the heart. Medical intervention and lifestyle modifications can postpone or stop CAD. In [16], authors apply ML algorithms to predict long-term risk. After Synthetic Minority Over-sampling Technique (SMOTE), the stacking ensemble model achieved 90.9% accuracy, 96.7% precision, 87.6% recall, and 96.1% AUC using 10-fold cross-validation. The diagnosis of Cardio Vascular Disease (CVD), a leading cause of death, requires excellent accuracy. In [17], classification techniques, sound signal processing, and image processing are used to compare ML methods. RF, which has a dataset accuracy of 90.16%, is the best classifier. Experts can enter patient health information into a web program to forecast CVD. The effectiveness of Artificial Neural Networks (ANNs) as predictors has increased their popularity. In [18], the authors suggest a novel method for heart disease prediction using PCA, LR, and DNN that have already been trained. With accuracy rates of 91.79% and 93.33% in training and testing data, respectively, the suggested method exceeds cutting-edge methods.

TABLE 1. Description of dataset before and after data balancing.

	Total Instances	Healthy	Unhealthy
Imbalanced Dataset	100000	91500	8500
Balanced Dataset	146416	73208	73208

III. PROPOSED SYSTEM MODEL

The system models that are proposed in this work are thoroughly discussed in this section. The two modules that make up the system models are data pre-processing and classification.

A. PRE-PROCESSING OF DATA

In our suggested system models, we employed the dataset of heart patients gathered by the Centers for Disease Control and Prevention (CDC). The kaggle website makes this dataset accessible to the general audience. The dataset has 10 features and a total of 100000 instances. The description of the dataset is mentioned in Table 1.

B. DATA BALANCING

The process of data balancing helps to mitigate class imbalance issue. The dataset is an imbalance when one class has more instances than the other, and this could affect the model's performance. There are several ways to balance the classes, but it's important to remember that balancing should be done with great caution because it could result in information loss and overfitting.

1) PROXIMITY WEIGHTED RANDOM AFFINE SHADOWSAMPLING

Proximity Weighted Random Affine Shadowsampling (ProWRAS) uses synthetic sampling as part of its oversampling process. Minority class is divided, and then clusters made up of those members of minority class are constructed. Each cluster is given a particular weight. The cluster closest to the dominant class is given the most weight. Each cluster's weights is expressed in a normalized form. The ProWRAS

then determines the number of samples to be generated from each cluster. The fresh samples are used as input for the model's classification [19]. The ProWRAS balancing method is extremely effective since it generates new samples from the largest possible minority class.

Algorithm 1 shows the basic steps of ProWRAS.

Algorithm 1 Proximity Weighted Random Affine Shadowsampling

Input: Training data
ProWRAS-Oversampling (Dataset)
Start
Generate clusters of dataset
Initialize synthetic samples with an empty set
For (Cluster, Weight) ∈ Clusters **do**
Num_samples ← [num_samples_generate * Weight]
Synth_samples ← synth_samples ∪ synth
End For
Resulting set of generated data points
End

2) LOCALIZED RANDOMIZED AFFINE SHADOWSAMPLING

Localized Randomized Affine Shadowsampling (LoRAS) is an oversampling technique that is used to generate new synthetic samples of the minority class samples. The minority class samples are surrounded by small regions where LoRAS generates Gaussian noise, and convex combinations of several noisy data points are used to build our final synthetic data points [20].

Algorithm 2 shows the basic steps of LoRAS.

Algorithm 2 Localized Randomized Affine Shadowsampling

Inputs:
Majority class = C_{maj}
Minority class = C_{min}
Start
Initialize loras_set as an empty list
For each minority class data point p in C_{min} **do**
Calculate k nearest neighbor of p
Initialize neighborhood_shadow_sample as an empty list
For each minority class data point p in C_{min} **do**
S_p = Draw shadow samples of the minority class
Repeat until desired resulting points are created
Return resulting set of LoRAS data points as loras_set
End For
End

3) ADAPTIVE SYNTHETIC

The Adaptive Synthetic (ADASYN) sampling technique creates minority data samples from the total samples [21]. Using the minority class as a source, new data points are produced. To create a synthetic sample between a minority class sample and one of its k closest neighbors, ADASYN computes a synthetic sample for each minority class sample. The basis for this balancing strategy is the difference in feature values between the minority sample and its k nearest neighbors. The weight given to a minority sample increases as the feature values of that minority sample diverge from those of its neighbors. To create fresh data points that serve to balance the dataset, new synthetic samples and minority samples are mixed [22].

Algorithm 3 shows the basic steps of ADASYN.

Algorithm 3 Adaptive Synthetic

Input: Training data
No. of majority samples N^-
No. of minority samples N^+
Start
Set the threshold value
Maximum degree of class imbalance = d^{th}
Synthetic samples $G = (N^- - N^+) * \beta$
Normalized each minority sample $r_x = r_i / \sum r_i$
Total no. of synthetic samples $g_i = r_x * G$
End

The total number of minority data points to generate is G. The ratio of minority classes is represented as β. For each minority class, the k nearest neighbor is calculated as r_i [23].

4) SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

To overcome the issue of data imbalance, Synthetic Minority Oversampling Technique (SMOTE) is used. It generates synthetic samples using the minority class attributes. SMOTE increases the percentage of only the minority cases after taking the complete dataset as input. SMOTE works the best and helps to balance the dataset [24].

Algorithm 4 presents the working of SMOTE.

Algorithm 4 Synthetic Minority Oversampling Technique

Input: Training data
The training set = T_r as an input
Nearest neighbor = p
Nearest neighbor during data cleaning = k
Output: The training set after SMOTE = New_T_r
Start
For i = 1 to N **do**
Generate the new samples from the minority class and add it to *New_*T_r
End For
End

5) RANDOM OVER SAMPLING

Random Over Sampling (ROS) is the most common oversampling method. ROS chooses samples at random and produces fresh samples of the minority class samples. Even when the number of samples is increased, new samples are frequently

very identical to the original samples, which could lead to overfitting because the generated samples are perfect replicas of the original samples. Algorithm 5 presents the working of ROS [25].

Algorithm 5 Random Over Sampling

1: **Input:** Training data
2: Samples of minority class *C_min* as input
3: Samples of majority class *C_maj* as input
4: Start
5: **For** i = 1 to N **do**
6: Randomly generate the new samples from the minority class
7: **End For**
8: End

6) MAJORITY WEIGHTED MINORITY OVERSAMPLING TECHNIQUE

The hard-to-learn informative minority class samples are first identified by Majority Weighted Minority Oversampling Technique (MWMOTE), which then weights them based on how far they are from the nearby majority class samples in the Euclidean space. Then, using a clustering method, it creates the synthetic samples from the weighted informative minority class samples. All of the created samples are done so that they all fall into a single minority class cluster [26]. Algorithm 6 presents the working of MWMOTE.

Algorithm 6 Majority Weighted Minority Oversampling Technique

1: **Input:** Data x
2: Minority class samples *S_min*
3: Majority class samples *S_maj*
4: N = Number of synthetic samples
5: Minority class clusters= $L_1, L_2, \ldots, L_M$
6: Start
7: **For** j = 1, 2, …, N **do**
8: Select a random sample from the set of clusters.
9: Generate new sample *s*
10: Add new sample in S_min
11: **End For**
12: End

7) BORDERLINE- SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE

The Borderline - Synthetic Minority Oversampling Technique (B-SMOTE) technique focuses specifically on the samples that are located on the periphery of minority classes, from which new samples are created. As the boundaries between classes grow more distinct, a successful classifier prediction is achieved. However, the generated synthetic samples will emphasize the overlapping regions, particularly when the distinction between classes is unclear [27]. Algorithm 7 presents the working of B-SMOTE.

Algorithm 7 Borderline-Synthetic Minority Oversampling Technique

1: **Input:**
2: M = Minority class samples
3: S = New synthetic samples
4: k = Nearest neighbor
5: **Output:** (S/100)* M
6: Start
7: **For** i = 1 to M **do**
8: Compute k nearest neighbor for each minority instance
9: Check the number of majority instances M'
10: **If** $\frac{k}{2}$ < M' < k
11: Add instance M to to borderline subset
12: **End If**
13: **End For**
14: End

8) RANDOM WALK OVERSAMPLING TECHNIQUE

By randomly walking from the real data, the Random Walk Over Sampling (RWOS) technique balances the samples of each class. The minority class instance set and the training data set T are both specified. A point in the m-dimensional space is represented by each of the m qualities corresponding to m dimensional vectors. We give the attribute set a name and use it to indicate an instance's attribute value. The number s indicates how many synthetic samples are produced from each instance. If, each real instance in P is utilized once, n synthetic instances are made, and the minority class is oversampled at a rate of 100% [28]. The working of RWOS is explained in Algorithm 8.

Algorithm 8 Random Walk Oversampling Technique

1: **Input:** Training data
2: Start
3: **For** i = 1 to m **do**
4: Calculate the mean $\mu_i = \frac{\sum_{j=1}^{n} a_i(j)}{n}$
5: Calculate the variance $\sigma_i^2 = \frac{1}{n}\sum_{j-1}^{n}(a_i(j) - \mu_i)^2$
6: Generate random synthetic samples
7: **End For**
8: End

C. DIMENSIONALITY REDUCTION

Dimensionality reduction is a method of reducing the number of variables taken into account. It can be used to extract latent features from raw datasets or to reduce data while maintaining its original structure. In this work, three different dimensionality reduction methods are used. These methods are discussed below.

1) RECURSIVE FEATURE ELIMINATION

Recursive Feature Elimination (RFE) is a method used to select the most important features. Features are ranked according to their importance in the dataset. The least

important features are ranked the lowest and discarded [29]. When utilizing RFE, there are two crucial configuration choices: the number of features to choose and the algorithm that will be used to aid in feature selection. Although the effectiveness of the approach is not largely dependent on the configuration of these hyperparameters, it is possible to investigate both of them.

The working of RFE [30] is explained in Algorithm 9.

Algorithm 9 Recursive Feature Elimination

Input
Start
Training set
Set of features F= $f_1, \ldots, f_M$
Number of features to select = N
Output:
Most highest ranked set of features $\rightarrow F^t$
For each n in N, **do**
Perform RFE and select n features $\rightarrow F^t$
Train with RF using F^t
End For
F = F^t
End

2) LINEAR DISCRIMINANT ANALYSIS

LDA is a supervised method used to reduce the dimension of data. LDA works in a linear combination, that uses several data items and applies a function to that location to individually analyze distinct classes. Finding the linear combination of features that distinguishes two or more classes of objects is performed by LDA approach. In other words, LDA is a method that enables us to determine which features are most advantageous for classifying various types of data [31]. The working of LDA is explained in Algorithm 10.

Algorithm 10 Linear Discriminant Analysis

Input:
Training data: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$, *where* $x_i \in R^d$ *and* $y_i \in 1, 2, \ldots, k$ *for* $i = 1, 2, \ldots, n$
Start
Learn projection matrix: W $\in R^{d*(k-1)}$
Compute class means $m_j = \frac{1}{n_j} \sum$ i:$y_i = jx_i$ for j = 1,2, …, k
Compute between class matrix: SB = $\sum$ j=$1^k n_j(m_j - m)(m_j - m)^T$, *wherem* = $\frac{1}{n} \sum i = 1^n x_i$
Compute within class matrix: SW = $\sum$ j=$1^k \sum i : y_i = j(x_i - m_j)(x_i - m_j)^T$
Compute eigen decomposition of $S_W^{-1} S_B$: $S_W^{-1} S_B = U \Lambda U^T$
Select the first k-1 eigen vectors W = $[u_i, u_2, \ldots, u_{k-1}]$, where u_i is the ith eigen vector of $S_W^{-1} S_B$
Project data onto the learned subspace: $x' = W^T x$
End

The input data D has a wide range of features and k number of classes. Firstly, the mean class m_j is calculated. Secondly, the between class matrix and within class matrix are calculated as *SB* and *SW*. Projection matrix *W* is composed using the eigen vector k-1 using the eigen decomposition of a matrix. At last, the data is projected as *x*.

3) FACTOR ANALYSIS

An unsupervised learning technique used to reduce the dimension of data is Factor Analysis (FA). In order to express the common variance, or variation resulting from correlation among the observed variables, this approach constructs factors from the observed variables. The mathematical formulation of FA is given in the equation 1 [32].

$$x_i - \mu_i = l_1 F1 + \ldots + l_n F_n + \epsilon_i \tag{1}$$

Variable *x*, factor *F* and the loading factor which acts as the factor weight is *l*, for the corresponding variable.

4) CRITICAL ANALYSIS

The optimal dimensionality reduction approach can have a big impact on the efficiency of the proposed model. A crucial phase of model creation is dimensionality reduction. In the proposed model, RFE and LDA have shown great results, however, FA has not shown good results. For feature selection, RFE is a helpful method because it shows the features that are important for the proposed model. In order to improve the class separation, LDA is very helpful in restructuring the feature space, which enhances the performance of the base classifiers. The low performance of FA indicates that the underlying latent factors generating the observed features are not supporting the dataset used for the prediction of heart disease.

D. MACHINE LEARNING TECHNIQUES

In the field of medicine, data science and ML have been extensively applied to the risk assessment of multiple disorders. The primary goal of the most widespread use of these models is to identify the most appropriate variables for long-term risk prediction in order to prevent significant health issues and support specialists. This study presents the forecasting abilities of the stacking model. To estimate the long-term risk of an individual developing CVD, in particular, the proposed system model is compared with benchmark techniques.

1) PASSIVE AGGRESSIVE CLASSIFIER

PAC is an ML algorithm commonly known as online learning algorithm. It works efficiently on big data. It almost looks like the perceptron algorithm where the data is entered in sequence and updated. PAC works in a way that it has no learning rate. The data comes in an order, the classifier learns from it and then discards that data automatically. This algorithm is quick enough to detect and modify the changes in the model as new data comes. It has a regularization

parameter for the efficiency of the model. The regularization parameter helps to validate the model in a way that it does not misclassify the results. It only changes the parameter when it is unable to accurately classify the data. If the model is unable to classify the new incoming data, then it will change the weight and move the hyperplane for the accurate classification. This is known as the aggressive part of the algorithm. Whereas, the passive part of PAC is that when it classifies the new data accurately and does not make any change in the model. It is a memory efficient model as it is not able to store previous data. It reads the data, updates it and then discards the data [33]. This process is explained in Algorithm 11. X represents the training instances and target labels are represented as Y in the dataset N. The cost function of PAC is ρ. Whereas, weights are assigned as W and l denotes the loss function.

Algorithm 11 Passive-Aggressive Classifier

Require: Dataset N
Weight =W
x_i = Training Set
y_i = Testing Set
1: Cost function ρ (y, y')
2: Start
3: Weight Vector W = 1/N where i = 1, 2, …, N
4: **If** PA == PB
5: Return $y_i^{'} = argmax_{y \in Y}(Wi.\phi(x_i, y))$
6: **End If**
7: **If** PA method==Max-loss(ML)
8: Return y'= arg $max_{r \in Y}(Wi.\phi(x_i, r)) - (Wi.\phi(x_i, y_i)) + \sqrt{\rho}(y_i, r)$
9: **End If**
10: Loss function: l_i= (Wi.ϕ (xi, $y_{i'}$))-(Wi.ϕ (xi,$y_{i'}$))+$\sqrt{\rho}$(yi, $y_{i'}$)
11: Update weight W
12: End

PAC is modified in two updated versions, PA-I and PA-II, which are used to increase the dimensions of x and to minimize the misclassification. The mathematical formulation of PA-I and PA-II is given in the equations below.

$$\tau_t = min\{C, \parallel l_t/x_t \parallel^2\} \quad (2)$$

$$\tau_t = min\{\parallel l_t/x_t \parallel^2 + 1/2C\} \quad (3)$$

In equations 2 and 3, τ is the lagrange multiplier and the positive parameter is represented as C.

2) RIDGE CLASSIFIER

RC is an ML algorithm that deals with the linear discriminant model. It is used for regularization and to prevent the model from overfitting. In the regularization method, a penalty is given to the model coefficient (cost function), which helps to reduce the complexity of the model. When we are giving the penalties, we have to check the coefficient value. When we are dealing with small data, the penalty will be more and the coefficient value will be smaller. This helps to prevent the model from overfitting. On the other hand, the model will underfit as the coefficient value is large with the penalty while dealing with big data [33]. RC uses L2 (ridge) regularization, which is basically a ridge function that helps to improve model performance and increase the training speed. This process is explained in Algorithm 12. Regression parameter α is calculated for each entry in the dataset.

Algorithm 12 Ridge Classifier

1: **Input** Data matrix X holds the training set
2: Data matrix Y holds the testing set
3: Start
4: **For** each test data y ∈ Y **do**
5: Calculate the regression parameter vector:
$\alpha = arg_{\alpha i} min \parallel x - X_{i\alpha i} \parallel_2^2 + \lambda \parallel \alpha_i \parallel_2^2$
6: Calculate distance between the test sample y and y_i
7: Assign y to that class whose distance is minimum
8: **End For**
9: End

RC seeks to reduce the sum of error terms as well as the sum of the squares of the coefficients [34]. Regularization refers to the sum of the squares of the coefficients, and it also has the regularization coefficient, represented by λ and β is the regression coefficient used in equation 4.

$$argmin_{\beta o \beta}\{1/\mathrm{N} \sum_{i=1}^{N} (y_i - \beta_o \sum_{j=1}^{p} x_i j \beta_j)^2\} \quad (4)$$

3) STOCHASTIC GRADIENT DESCENT CLASSIFIER

Traditionally, gradient descent was designed to find a slope. It starts from a random point and then shifts its slope step by step downwards to find the minimum value of a function. It performs well on small data. However, if we have big data, it slows down the processing. This will cause overhead and become computationally expensive [35]. To encounter this issue, SGDC is introduced where Stochastic generally means random. So, in this case, we just work on a single point from the entire dataset at every iteration. This reduces the computation complexity and speeds up the model. This process is explained in Algorithm 13.

The mathematical representation of SGDC is written in equation 5.

$$\omega = \omega - \eta \nabla Q_i(\omega) = \omega - \eta/n \sum_{i-1}^{n} \nabla Q_i(\omega) \quad (5)$$

η is the learning rate, ω is the initial vector from where we start and Q_i is the loss function.

4) EXTREME GRADIENT BOOSTING CLASSIFIER

XGBoost is an optimization algorithm. It is a modified version of gradient boosting classifier that helps to enhance the accuracy of a model. It is an ensemble learning method, which works in a sequential order [36]. It works on building trees in a sequential order. The wrong predictions are moved to the next tree, which will train it again with the original

Algorithm 13 Stochastic Gradient Descent Classifier

Require: **Input** Data N
Training Set X
W = Initialize as small random numbers

while not converged **do**
For n ← 1, N **do**
$y_k^n \leftarrow \sum_d^{i=1} \omega_{ki} Xi^n + b_k$
For k ← 1, K **do**
For i ← 1, d **do**
$\omega = \omega - \eta \nabla Q_i(\omega) = \omega - \eta/n \sum_{i-1}^{n} \nabla Q_i(\omega)$
End For
$b_k \leftarrow b_k - \eta.\omega$
End For
End For
End while
End

dataset and this process continues. The implementation is done in a parallelization loop between leaf nodes and the features [37]. Equation 6 expresses the quadratic functions of one variable in XGBoost.

$$L^{(t)} = \sum_{i=1}^{N} [g_i f_t(x_i) = 1/2 h_i f_t^2(x_i)] + \Omega(f_t) \quad (6)$$

L is the loss function, g is the gradient and Ω is the optimization.

In Algorithm 14, f(x) is the number of weak learners for k number of iterations. Base classifiers are represented by $b_(k)$ and weight of each tree is given by ω.

Algorithm 14 Extreme Gradient Boosting Classifier

Require: Input Data N
Training samples in X
Gain = G

Start
Dataset N = (xi, yi),…, (xn, yn), where $xi \in X$ and $yi \in Y$ {(0,1)}
Initialize $f(x) = \sum b_k(x)$
For k=1, 2, … M where M is the number of base learners **do**
Calculate $g_k = \frac{\partial L(y,f)}{\partial f}$
Determine the structure by choosing split with maximum gain
A = $\frac{1}{2}[\frac{G_L^{(2)}}{H_L} + \frac{G_R^{(2)}}{H_R} + \frac{G^{(2)}}{H}]$
Determine the leaf weights $w* = -\frac{G^{(2)}}{H}$
Determine the base learner b(x) = $\sum^{(} T)(j-1)$ wi
Add tree $f_k(x) = f_{k-1}(x) + b(x)$
End For
Return output
End

5) LOGITBOOST CLASSIFIER

It is a boosting algorithm and is used to improve the performance of DT classifiers on binary classification. It is commonly known as additive logistic regression and is used to minimize the logit loss by using L2 regularization. Value of L2 regularization is not mentioned during the classification. The model tends to drive the slope towards zero and leads to overfitting. To resolve this issue, L2 regularization is initialized. Logitboost is less sensitive to outliers and noise. In this model, on every round, new weights are assigned to the previous wrongly predicted samples. This process is explained in Algorithm 15 [38].

Algorithm 15 LogitBoost Classifier

Require: **Input** Data N
Training samples in X
Weight =W

Start
Dataset N = (x1, y1),…,(xi, yi),…, (xn, yn), where $xi \in X$ and $yi \in Y$ {(0, 1)}
K = Number of iterations (1, 2, …, k)
Weight Vector W = 1/N where i = 1, 2,…, N
Function F(x) = 0
Probability p(xi)=1/2
Calculate the weights on each iteration
Wi=$p(x_i)/(1-p(x_i))$
$z_i = y_i - p(x_i)/p(x_i)(1-p(x_i))$
Fit the function F (x) by a weighted least squares regression of zi to xi using weights Wi
Update F(x) = F(x) + 1/2fk(x)
$p(x) \leftarrow \frac{e^{f(x)}}{e^{F(x)}+e^{-F(x)}}$
Output sign [F(x)] =$\sum$[Fk(x)]
Return Output
End

The dataset is separated into two classes and is represented by N in Algorithm 15. Let the training set consists of the following values: {(x1, y1), …, (xi, yi), …, (xn, yn)}, where X represents the feature vector and Y represents the target class. The mathematical formulation to minimize the logistic loss [37] is mentioned in equation 7.

$$\sum_i log(1 + e^{-y_i f(x)}) \quad (7)$$

E. HYPERPARAMETER TUNING

Hyperparameters are a set of parameters that are used to regulate how a model or algorithm behaves and can be changed to produce an improvised model with the best performance. The process of choosing the best collection of hyperparameters for an ML model is known as hyperparameter tuning. Since the selection of hyperparameters has a significant influence on the model's performance, it is an important step in the model-building process [39]. We have used the Random Search approach for tuning the hyperparameters. The model is trained using a set of combinations of hyperparameters that are randomly selected from a preset list in the random search approach [40]. The Table 2 shows the used ranges

TABLE 2. Hyperparameters' range of the machine learning techniques used in this work.

Classifiers	Hyperparameters	Range	Selected Values
PAC	Regularization 'C'	loguniform(1e-5, 1e5)	1.6063676259174505e-05
	max_iter	[1000, 5000]	2000
	random_state	[0 , 42]	30
RC	Regularization 'α'	uniform(0.1, 10.0)	7.3
	solver	[auto, svd, cholesky, lsqr, sparse_cg, saga]	saga
	random_state	[0 , 42]	42
SGDC	alpha	loguniform (1e-5, 1e0)	0.000745
	loss	[hinge, log, modified_huber, squared_hinge, perceptron]	perceptron
	penalty	[l2, l1, elasticnet]	l2
	max_iter	[1000, 2000, 3000, 4000, 5000]	3000
XGBoost	learning_rate	uniform(0.01, 0.2)	0.05820509320520235
	n_estimators	randint(50, 200)	57
	max_depth	randint(1, 20)	7
	subsample	uniform(0.5, 1.0)	0.5343885211152184
	colsample_bytree	uniform(0.5, 1.0)	0.730893825622149
LogitBoost	learning_rate	uniform(0.01, 0.2)	0.1323705789444759
	max_iter	[100, 200, 300, 400, 500]	400
	max_leaf_nodes	[15, 31, 63, 127]	127
	min_samples_leaf	[1, 2, 4]	4
	max_depth	randint(1, 20)	5
Stacking Model	n_estimator	randint(50, 200)	51
	learning_rate	[0.1, 0.01, 0.001]	0.01
	max_depth	randint(3, 10)	10

of the hyperparameters used by the classifiers (ML). The Random Search optimization technique is used to adjust hyperparameters of all of the machine learning classifiers used in this work. The range of hyperparameters of the many thresholds in the base (benchmark) classifiers and also in the proposed modesl are provided in the Table 2. The Random Search method is used to modify the hyperparameters of all classifiers. By using these specific hyperparameters, the performance of the proposed model is enhanced and it becomes more efficient than the base classifiers.

F. SHAPLEY ADDITIVE exPLANATIONS

SHAPley Additive exPlanations (SHAP) is a visualization tool that is used to make a ML model's output more understandable. It can be used to explain the prediction of any model by estimating the contribution of each feature to the forecast. In order for SHAP to work, a model's output must be divided into the sums of the impacts of all of its features. The value obtained by SHAP represents the contribution of each feature to the model outcome. These values can be used to explain the model's result to someone and help them understand the importance of each component. This is especially beneficial for businesses and teams that report to clients or management [41].

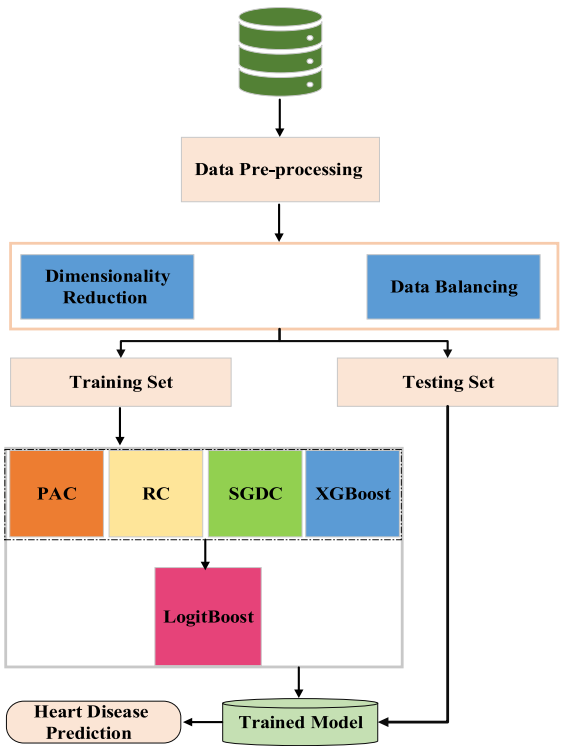


FIGURE 2. Proposed system model's flow diagram.

G. PROPOSED STACKING MODEL

ML plays an important role in the diagnosis of cardiovascular disease prediction. Many ML techniques have been introduced for the prediction of heart patients. The proposed stacking model is a great contribution to heart disease prediction.

The proposed system model flow diagram is expressed in Figure 2.

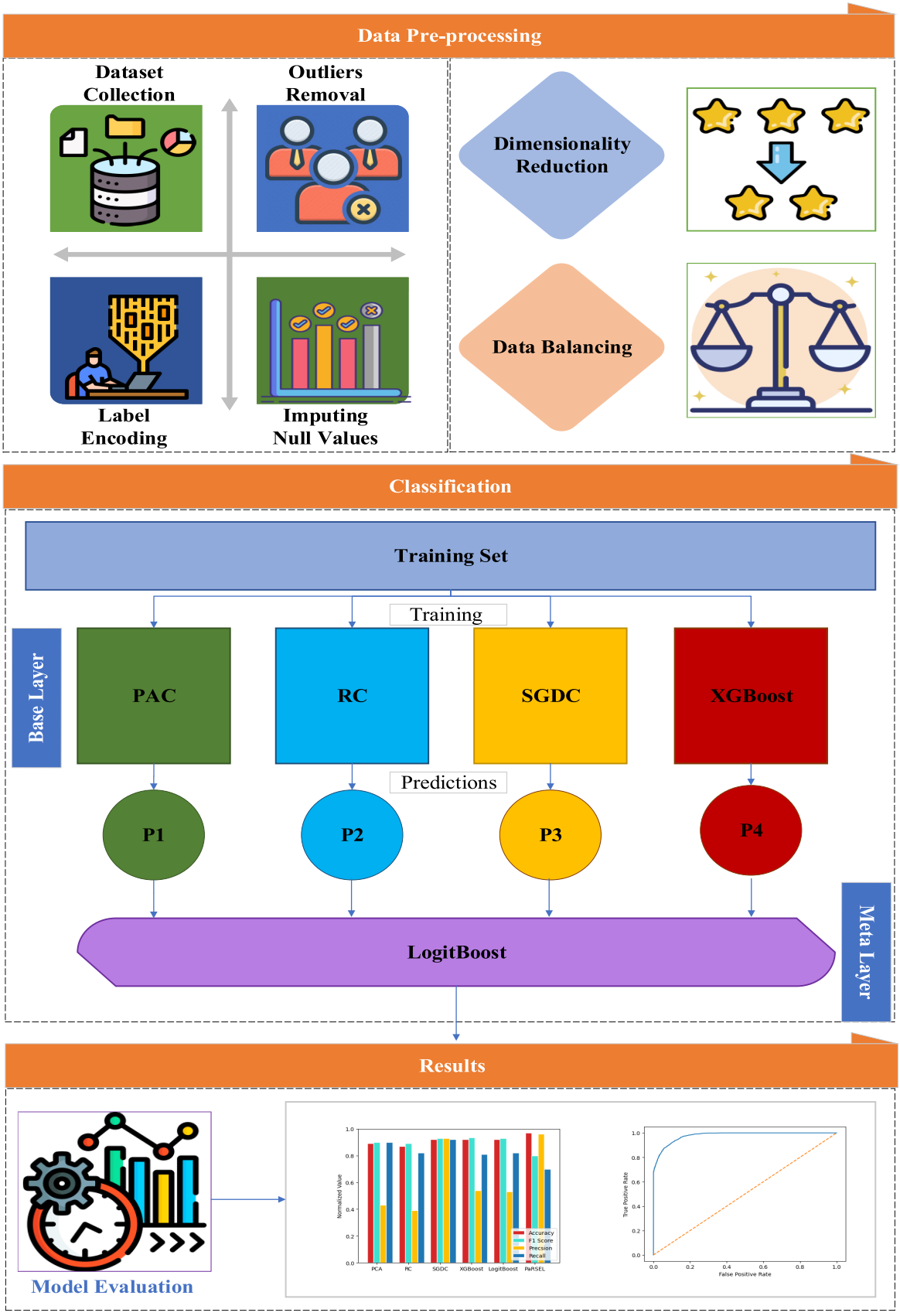


FIGURE 3. Proposed system model PaRSEL.

Firstly, the data is transformed into low dimension from high dimension without compromising on the quality of data using three different approaches. Secondly, eight different balancing techniques are implemented on the dataset to solve the issue of class imbalance. Classification is being performed using a stacking model named PaRSEL. The base layer of the PaRSEL consists of four passive classifiers, namely PAC, RC, SGDC and XGBoost. Each classifier has its own prediction based on the training data. The training data serves as the primary building block for classifier algorithms. The data is fed to the algorithms, which then discover patterns and correlations between features and related labels present in the data. Meta layer uses the LogitBoost classifier as an active learner. The PaRSEL is used to predict the heart disease. The model is trained on a labeled dataset. The most relevant features are used for the training of model such as hypertension, physical health, stress level, smoking and alcohol consumption.

Algorithm 16 explains how our proposed model PaRSEL works. First, the dataset D is loaded. Data is labeled in the form of 0 and 1. Null values and outliers present in the dataset are removed. Dimensionality reduction is used to reduces the number of features in the dataset. The dataset is split into the two sub sets, the training set and the testing set. Afterward, data balancing techniques are used to solve the imbalance nature of the dataset, as in [42]. Four base classifiers, namely PAC, RC, SGDC and XGBoost, are initialized at level 0 and trained on the training data. Each classifier learns different patterns and makes predictions. The classifications done by the base machine learning machine classifiers are fed as input to the meta layer classifiers for final predictions.

Algorithm 16 Proposed Model: PaRSEL's Algorithm

Input: Training data D
Start
Remove null values
Apply dimensionality reduction techniques one by one
Split D into train and test
Apply balancing techniques to balance the dataset
Create base layer classifiers
PAC = (level 0)
RC = (level 0)
SGDC = (level 0)
XGBoost = (level 0)
For t = 1 to T **do**
learn h_t based on D
End For
Construct a new dataset of predictions
For i = 1 to m **Do**
$D_h = \{x_i', y_i\}$, where $x_i' = \{h_1(x_i), \ldots, h_t(x_i)\}$
End For
Create meta layer classifier
LogitBoost = (level 1)
learn H based on D_h
return H
Train the model
Fit the model (x-train, y-train)
Final Prediction
End

IV. RESULTS AND DISCUSSION

The proposed model PaRSEL's simulation results are evaluated and discussed in this section. Four different ML classifiers, PAC, RC, SGDC and XGBoost, are implemented at the base layer and their results are compared with the proposed model. LogitBoost is deployed at the meta layer in the proposed model. The simulation tool for Python's code implementation is Google Colab. We use the heart patients' dataset, which comprises 100000 instances and is imbalanced. To balance the dataset, eight different balancing techniques have been used, and for accurate and efficient results, three dimensionality reduction techniques are implemented.

A. PERFORMANCE RESULTS

The best outcomes of our suggested model, PaRSEL, are discussed in this section using dimensionality reduction and data balancing approaches. Figure 4 and Table 3 show that PaRSEL achieves 97% accuracy, 80% F1-score, 96% precision and 70% recall value after applying dimensionality reduction and ProWRAS balancing approach. As the number of features are reduced using RFE, LDA and FA, the model's performance is enhanced. For RFE and LDA, the proposed model outperforms the standalone techniques. However, with FA, it underperforms and does the achieve the desired outcomes.

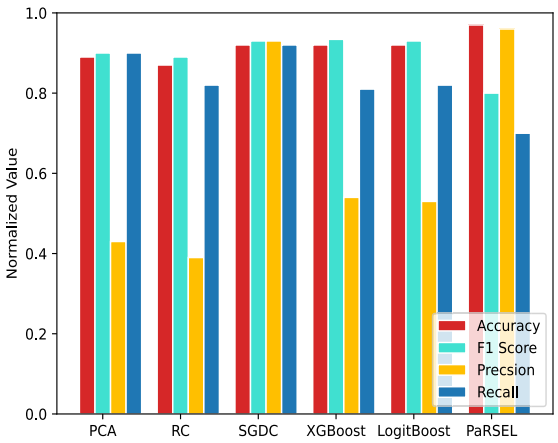


FIGURE 4. PaRSEL's comparison with the base classifiers using ProWRAS, RFE, LDA and FA.

Table 4 shows that PaRSEL achieves 98% accuracy, 95% F1-score, 94% precision and 97% recall value after applying dimensionality reduction, ProWRAS balancing approach and hyperparameter tuning. This clear change caused by hyperparameter tuning is given in Table 3.

Figure 5 and Table 5 show that PaRSEL achieves 97% accuracy, 81% F1-score, 99% precision and 68% recall

TABLE 3. Proposed Model: PaRSEL's comparison with the base classifiers.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ProWRAS without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	83%	86%	30%	83%	44 sec
RC	87%	89%	38%	82%	33 sec
SGDC	92%	93%	93%	92%	55 sec
XGBoost	92%	92%	52%	81%	32 sec
LogitBoost	92%	92%	52%	82%	51 sec
Proposed Model	97%	80%	93%	70%	1 min 3 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ProWRAS with RFE Dimension Reduction Technique					
PAC	89%	90%	43%	90%	57 sec
RC	87%	89%	39%	82%	14 sec
SGDC	88%	89%	93%	88%	7 sec
XGBoost	92%	93%	54%	81%	12 sec
LogitBoost	92%	93%	53%	82%	20 sec
Proposed Model	97%	80%	93%	70%	1 min 9 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ProWRAS with LDA Dimension Reduction Technique					
PAC	81%	84%	30%	81%	36 sec
RC	87%	89%	38%	82%	16 sec
SGDC	88%	89%	93%	88%	22 sec
XGBoost	88%	89%	40%	80%	8 sec
LogitBoost	87%	89%	39%	81%	14 sec
Proposed Model	97%	80%	96%	68%	1 min 7 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ProWRAS with FA Dimension Reduction Technique					
PAC	75%	74%	69%	75%	13 sec
RC	76%	76%	71%	87%	9 sec
SGDC	76%	76%	77%	76%	16 sec
XGBoost	80%	80%	77%	87%	17 sec
LogitBoost	82%	82%	79%	87%	11 sec
Proposed Model	82%	83%	79%	87%	1 min 3 sec

TABLE 4. Proposed Model: PaRSEL's comparison with the base classifiers using balancing technique ProWRAS with LDA dimension reduction technique using hyperparameter tuning.

Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	91%	90%	91%	90%	15 sec
RC	91%	90%	87%	95%	12 sec
SGDC	91%	91%	91%	91%	10 sec
XGBoost	91%	90%	92%	59%	15 sec
LogitBoost	91%	91%	90%	92%	14 sec
Proposed Model	98%	95%	94%	97%	10 sec

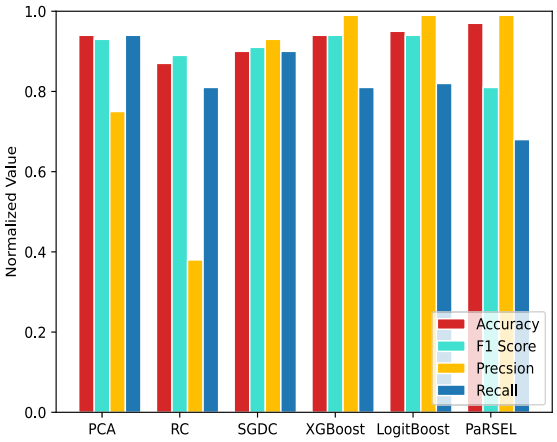


FIGURE 5. PaRSEL's comparison with the base classifiers using LoRAS, RFE, LDA and FA.

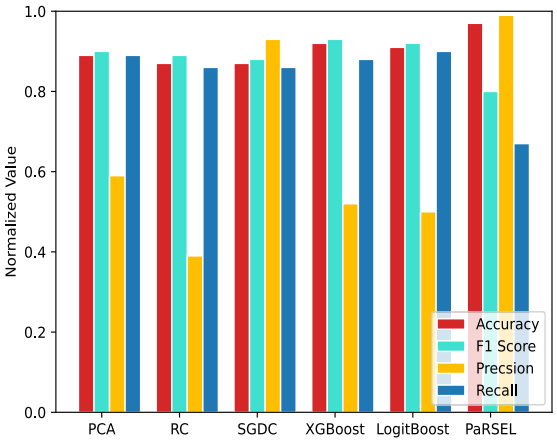


FIGURE 6. PaRSEL's comparison with the base classifiers using ROS, RFE, LDA and FA.

value after applying dimensionality reduction and LoRAS balancing approach. As the number of features reduced using RFE, LDA and FA, the model's performance is enhanced. The model performs efficiently when using RFE and LDA. However, when using FA, the model's performance is reduced and less than 50% accuracy is achieved.

Figure 6 and Table 6 show that PaRSEL achieves 97% accuracy, 80% F1-score, 99% precision and 67% recall value

TABLE 5. PaRSEL's comparison with the base classifiers using LoRAS.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique LoRAS without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	70%	76%	10%	70%	41 sec
RC	76%	80%	30%	78%	39 sec
SGDC	68%	70%	45%	62%	45sec
XGBoost	83%	83%	52%	81%	31 sec
LogitBoost	83%	82%	52%	82%	55 sec
Proposed Model	97%	80%	97%	68%	1 min 40 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique LoRAS with RFE Dimension Reduction Technique					
PAC	69%	75%	18%	69%	19 sec
RC	85%	87%	34%	75%	10 sec
SGDC	90%	91%	93%	90%	17sec
XGBoost	94%	94%	67%	73%	13 sec
LogitBoost	95%	94%	69%	73%	24 sec
Proposed Model	97%	80%	99%	67%	1 min 7 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique LoRAS with LDA Dimension Reduction Technique					
PAC	94%	93%	75%	94%	19 sec
RC	87%	89%	38%	81%	24 sec
SGDC	90%	91%	93%	90%	14sec
XGBoost	92%	90%	99%	18%	13 sec
LogitBoost	92%	90%	99%	17%	16 sec
Proposed Model	97%	81%	99%	65%	1 min 8 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique LoRAS with FA Dimension Reduction Technique					
PAC	49%	33%	10%	50%	15 sec
RC	49%	33%	9%	45%	45 sec
SGDC	49%	33%	25%	50%	50 sec
XGBoost	50%	33%	15%	41%	15 sec
LogitBoost	50%	35%	25%	31%	30 sec
Proposed Model	50%	31%	20%	35%	10 sec

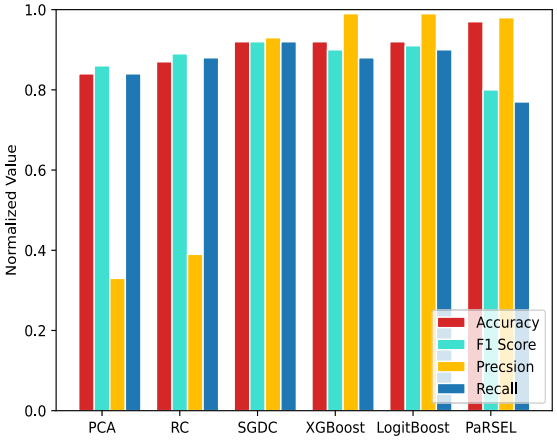


FIGURE 7. PaRSEL's comparison with the base classifiers using ADASYN, RFE, LDA and FA.

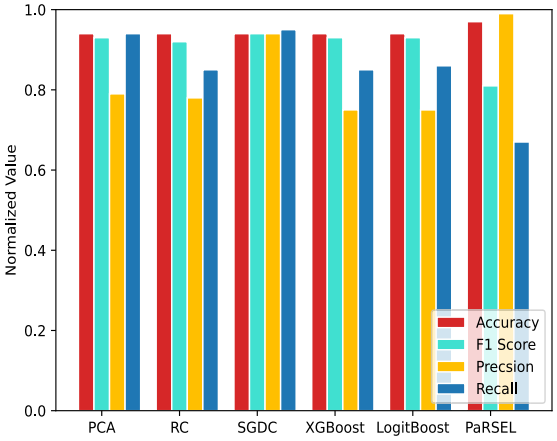


FIGURE 8. PaRSEL's comparison with the base classifiers using SMOTE, RFE, LDA and FA.

after applying dimensionality reduction and ROS balancing approach. The model's performance is enhanced via reducing the number of features using RFE, LDA and FA. The model performs poorly when using FA while it performs efficiently when RFE and LDA are used.

Figure 7 and Table 7 show that PaRSEL achieves 97% accuracy, 81% F1-score, 99% precision and 67% recall value after applying dimensionality reduction and ADASYN balancing approach. Due to the feature reduction application of RFE, LDA and FA, the model's performance is enhanced. However, unlike when RFE and LDA are used, the model does not work efficiently when FA is used.

Figure 8 and Table 8 show that PaRSEL achieves 97% accuracy, 80% F1-score, 99% precision and 77% recall value after applying dimensionality reduction and SMOTE balancing approach. As the no of features are reduced using RFE, LDA and FA, the model's performance is enhanced. However, due to random selection of features by FA, the model is not performing as efficiently as with RFE and LDA.

Figure 9 and Table 9 show that PaRSEL achieves 97% accuracy, 80% F1-score, 99% precision and 68% recall value after applying dimensionality reduction and B-SMOTE balancing approach. Due to the feature reduction application of RFE, LDA and FA, the model's performance is enhanced. However, the model does not work efficiently when FA is used unlike when RFE and LDA are used.

Figure 10 and Table 10 show that PaRSEL achieves 97% accuracy, 80% F1-score, 93% precision and 68% recall value

TABLE 6. PaRSEL's comparison with the base classifiers using ROS.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ROS without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	89%	90%	41%	89%	49 sec
RC	87%	89%	39%	86%	53 sec
SGDC	86%	88%	93%	86%	32 sec
XGBoost	92%	93%	52%	88%	1 min
LogitBoost	91%	92%	50%	90%	37 sec
Proposed Model	97%	80%	99%	67%	1 min 19 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ROS with RFE Dimension Reduction Technique					
PAC	49%	90%	59%	13%	49 sec
RC	87%	89%	39%	87%	10 sec
SGDC	76%	81%	93%	76%	25 sec
XGBoost	92%	93%	52%	88%	28 sec
LogitBoost	91%	92%	50%	90%	21 sec
Proposed Model	97%	80%	99%	67%	1 min 3 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ROS with LDA Dimension Reduction Technique					
PAC	79%	84%	28%	79%	12 sec
RC	87%	89%	38%	81%	26 sec
SGDC	87%	89%	93%	87%	23 sec
XGBoost	87%	90%	39%	80%	6 sec
LogitBoost	88%	89%	40%	80%	18 sec
Proposed Model	97%	81%	99%	64%	1 min 4 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ROS with FA Dimension Reduction Technique					
PAC	49%	31%	0%	50%	8 sec
RC	49%	31%	0%	0%	9 sec
SGDC	50%	33%	25%	50%	14 sec
XGBoost	49%	33%	0%	0%	14 sec
LogitBoost	49%	33%	0%	0%	9 sec
Proposed Model	50%	0%	0%	0%	13 sec

TABLE 7. PaRSEL's comparison with the base classifiers using ADASYN.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ADASYN without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	83%	86%	31%	83%	59 sec
RC	82%	85%	30%	88%	43 sec
SGDC	65%	73%	92%	65%	41 sec
XGBoost	88%	90%	43%	88%	1 min 4 sec
LogitBoost	88%	90%	41%	89%	56 sec
Proposed Model	97%	80%	98%	67%	1 min 18 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ADASYN with RFE Dimension Reduction Technique					
PAC	83%	86%	31%	83%	39 sec
RC	81%	85%	30%	88%	10 sec
SGDC	92%	92%	93%	92%	56 sec
XGBoost	88%	90%	43%	88%	56 sec
LogitBoost	88%	90%	41%	90%	52 sec
Proposed Model	96%	78%	78%	77%	1 min 8 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ADASYN with LDA Dimension Reduction Technique					
PAC	84%	86%	33%	84%	7 sec
RC	87%	89%	39%	81%	5 sec
SGDC	88%	90%	93%	88%	14 sec
XGBoost	92%	90%	99%	18%	13 sec
LogitBoost	92%	91%	99%	17%	9 sec
Proposed Model	97%	80%	98%	66%	1 min 5 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique ADASYN with FA Dimension Reduction Technique					
PAC	50%	34%	0%	50%	14 sec
RC	50%	33%	0%	0%	24 sec
SGDC	50%	33%	25%	50%	14 sec
XGBoost	49%	33%	0%	0%	15 sec
LogitBoost	49%	33%	0%	0%	60 sec
Proposed Model	50%	10%	0%	0%	19 sec

after applying dimensionality reduction and MWMOTE balancing approach. Due to the feature reduction capability of RFE, LDA and FA, the proposed model's efficiency is enhanced. However, unlike when RFE and LDA are used, the model performs poorly using FA.

Figure 11 and Table 11 show that PaRSEL achieves 97% accuracy, 81% F1-score, 99% precision and 67% recall value after applying dimensionality reduction and RWOS balancing approach. The proposed model's efficiency is enhanced due to reducing the number of features using RFE, LDA and FA. On the contrary to the model's performance with RFE and LDA, the model performs poorly with FA.

Moving ahead, the proposed model is tested on a new independent dataset using the same process flow. The dataset of cardiovascular patients is acquired from Kaggle, which makes this dataset accessible to the general audience. The

TABLE 8. PaRSEL's comparison with the base classifiers using SMOTE.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique SMOTE without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	69%	76%	20%	69%	39 sec
RC	85%	87%	35%	85%	41 sec
SGDC	65%	73%	92%	65%	1 min
XGBoost	91%	92%	49%	84%	55 sec
LogitBoost	90%	91%	47%	86%	34 sec
Proposed Model	97%	80%	99%	67%	2 min 12 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique SMOTE with RFE Dimension Reduction Technique					
PAC	81%	85%	30%	81%	41 sec
RC	85%	87%	35%	85%	15 sec
SGDC	91%	92%	93%	91%	7 sec
XGBoost	91%	92%	50%	85%	12 sec
LogitBoost	90%	91%	47%	86%	21 sec
Proposed Model	97%	81%	99%	64%	1 min 2 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique SMOTE with LDA Dimension Reduction Technique					
PAC	94%	93%	79%	94%	14 sec
RC	94%	92%	78%	49%	29 sec
SGDC	94%	94%	94%	95%	18 sec
XGBoost	94%	93%	75%	47%	6 sec
LogitBoost	94%	93%	75%	51%	21 sec
Proposed Model	97%	81%	99%	65%	1 min 5 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique SMOTE with FA Dimension Reduction Technique					
PAC	48%	32%	0%	50%	6 sec
RC	49%	33%	0%	0%	14 sec
SGDC	50%	33%	25%	50%	17 sec
XGBoost	48%	31%	0%	0%	43 sec
LogitBoost	49%	33%	0%	0%	6 sec
Proposed Model	50%	0%	0%	0%	12 sec

TABLE 9. PaRSEL's comparison with the base classifiers using B-SMOTE.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique B-SMOTE without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	87%	89%	37%	87%	1 min
RC	83%	86%	33%	87%	56 sec
SGDC	74%	80%	92%	74%	49 sec
XGBoost	91%	92%	50%	85%	1 min 5 sec
LogitBoost	90%	91%	47%	83%	1 min
Proposed Model	97%	80%	99%	68%	1 min 45 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique Borderline-SMOTE with RFE Dimension Reduction Technique					
PAC	87%	88%	38%	87%	27 min
RC	83%	86%	33%	87%	5 sec
SGDC	81%	85%	93%	81%	17 sec
XGBoost	91%	92%	49%	85%	12 sec
LogitBoost	90%	91%	47%	87%	22 sec
Proposed Model	97%	80%	98%	67%	1 min 15 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique Borderline-SMOTE with LDA Dimension Reduction Technique					
PAC	94%	94%	72%	94%	22 sec
RC	94%	93%	78%	49%	30 sec
SGDC	93%	94%	93%	94%	15sec
XGBoost	91%	87%	94%	40%	6 sec
LogitBoost	94%	93%	80%	46%	15 sec
Proposed Model	97%	80%	98%	65%	1 min 7 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique Borderline-SMOTE with FA Dimension Reduction Technique					
PAC	47%	31%	0%	50%	5 sec
RC	48%	33%	0%	0%	6 sec
SGDC	49%	33%	25%	50%	14 sec
XGBoost	49%	33%	0%	0%	15 sec
LogitBoost	50%	34%	0%	0%	3 sec
Proposed Model	50%	0%	0%	0%	13 sec

dataset has 11 features and a total of 50000 instances. Table 12 shows that PaRSEL achieves 98% accuracy, 95% F1-score, 94% precision and 97% recall after applying dimensionality reduction and ProWRAS balancing approach on the new dataset. The proposed model efficiently outperforms the individual classifiers.

In Figures 12 the execution time of PaRSEL is mentioned using eight different balancing approaches and labeled them as PaRSEL 1, PaRSEL 2, PaRSEL 3, PaRSEL 4, PaRSEL 5, PaRSEL 6, PaRSEL 7 and PaRSEL 8. By applying data balancing and dimensionality reduction approaches, the time complexity is reduced. The models take less time to predict the disease as compared to the time taken by the model using all features.

Figure 13 displays the proposed model's AUC-ROC curves. The proposed model obtains an AUC-ROC score

TABLE 10. PaRSEL's comparison with the base classifiers using MWMOTE.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique MWMOTE without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	90%	91%	45%	90%	53 sec
RC	87%	89%	39%	86%	39 sec
SGDC	82%	85%	93%	82%	59 sec
XGBoost	97%	96%	96%	69%	1 min 2 sec
LogitBoost	97%	97%	97%	69%	49 sec
Proposed Model	97%	80%	95%	65%	1 min 43 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique MWMOTE with RFE Dimension Reduction Technique					
PAC	85%	87%	35%	85%	20 sec
RC	87%	89%	39%	86%	6 sec
SGDC	94%	93%	93%	94%	7 sec
XGBoost	97%	96%	95%	69%	17 sec
LogitBoost	97%	97%	97%	70%	8 sec
Proposed Model	97%	81%	98%	64%	1 min 18 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique MWMOTE with LDA Dimension Reduction Technique					
PAC	93%	92%	95%	93%	9 sec
RC	92%	91%	99%	18%	18 sec
SGDC	92%	90%	93%	92%	13 sec
XGBoost	92%	91%	99%	18%	8 sec
LogitBoost	93%	90%	99%	18%	4 sec
Proposed Model	97%	80%	93%	67%	1 min 6 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique MWMOTE with FA Dimension Reduction Technique					
PAC	48%	33%	0%	50%	12 sec
RC	45%	31%	0%	0%	5 sec
SGDC	50%	35%	25%	50%	24 sec
XGBoost	50%	34%	0%	0%	15 sec
LogitBoost	49%	33%	0%	0%	35 sec
Proposed Model	50%	0%	0%	0%	10 sec

TABLE 11. PaRSEL's comparison with the base classifiers using RWOS.

Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique RWOS without any Dimension Reduction Technique					
Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	82%	86%	31%	82%	43 sec
RC	87%	89%	39%	86%	54 sec
SGDC	78%	83%	93%	78%	1 min 2 sec
XGBoost	92%	93%	52%	88%	55 sec
LogitBoost	91%	92%	50%	90%	49 se3c
Proposed Model	97%	80%	98%	67%	1 min 19 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique RWOS with RFE Dimension Reduction Technique					
PAC	85%	87%	35%	85%	20 sec
RC	87%	89%	39%	86%	6 sec
SGDC	94%	93%	93%	94%	7 sec
XGBoost	97%	96%	95%	69%	17 sec
LogitBoost	97%	97%	97%	70%	8 sec
Proposed Model	97%	81%	98%	64%	1 min 18 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique RWOS with LDA Dimension Reduction Technique					
PAC	93%	91%	97%	93%	9 sec
RC	92%	91%	99%	18%	37 sec
SGDC	92%	90%	97%	92%	23 sec
XGBoost	92%	90%	99%	18%	5 sec
LogitBoost	92%	90%	99%	17%	18 sec
Proposed Model	97%	81%	99%	67%	1 min 7 sec
Proposed Model: PaRSEL's Comparison with the Base Classifiers using Balancing Technique RWOS with FA Dimension Reduction Technique					
PAC	48%	33%	0%	50%	12 sec
RC	45%	31%	0%	0%	5 sec
SGDC	50%	35%	25%	50%	24 sec
XGBoost	50%	34%	0%	0%	15 sec
LogitBoost	49%	33%	0%	0%	35 sec
Proposed Model	50%	0%	0%	0%	10 sec

TABLE 12. Proposed Model: PaRSEL's comparison with the base classifiers using ProWRAS with LDA on a new dataset.

Classifier	**Accuracy**	**F1-Score**	**Precision**	**Recall**	**Execution Time**
PAC	92%	94%	92%	94%	15 sec
RC	79%	86%	53%	60%	12 sec
SGDC	81%	88%	97%	81%	10 sec
XGBoost	79%	86%	52%	59%	15 sec
LogitBoost	79%	86%	54%	60%	14 sec
Proposed Model	98%	95%	94%	97%	10 sec

of 98%. It simply indicates that our proposed model distinguished between the two classes quite efficiently and correctly, which validates its performance, and makes it effective and useful for heart prediction.

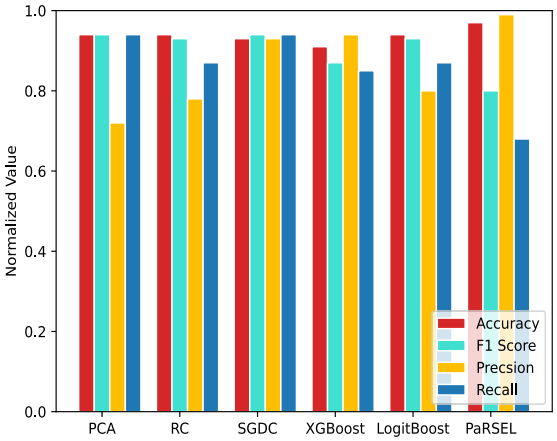


FIGURE 9. PaRSEL's comparison with the base classifiers using B-SMOTE, RFE, LDA and FA.

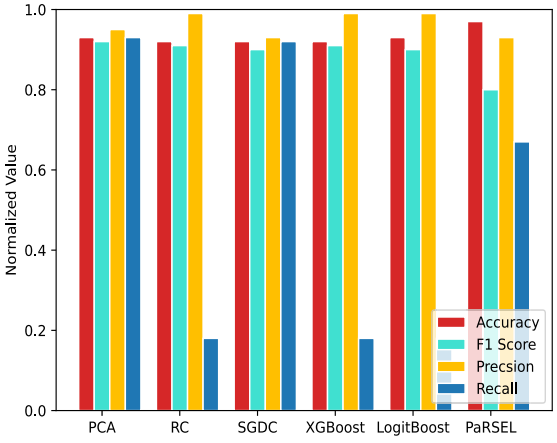


FIGURE 10. PaRSEL's comparison with the base classifiers using MWMOTE, RFE, LDA and FA.

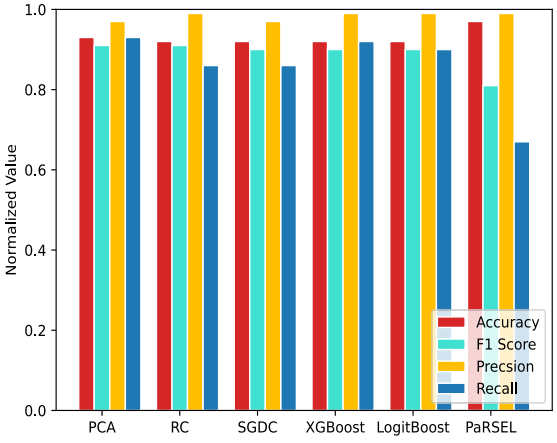


FIGURE 11. PaRSEL's comparison with the base classifiers using RWOS, RFE, LDA and FA.

The proposed model performed the best with all eight data balancing techniques and two dimensionality reduction techniques, RFE and LDA. Whereas, the proposed model does not perform well and gives less accuracy when FA is employed for dimensionality reduction.

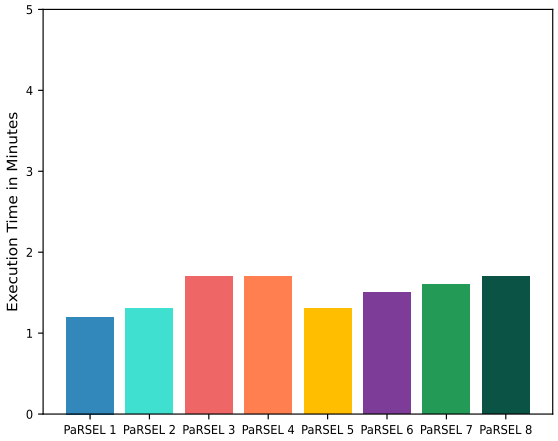


FIGURE 12. Comparing the execution time of PaRSEL with different techniques.

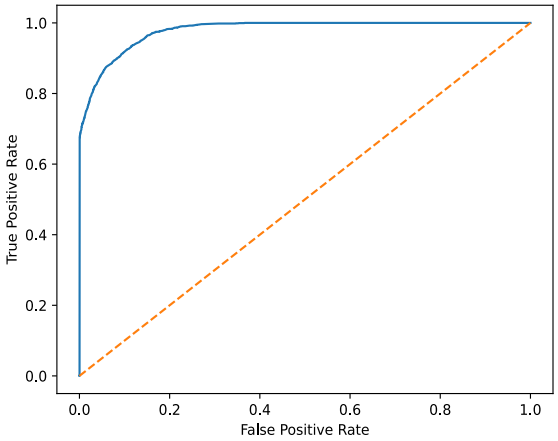


FIGURE 13. AUC-ROC curve of PaRSEL.

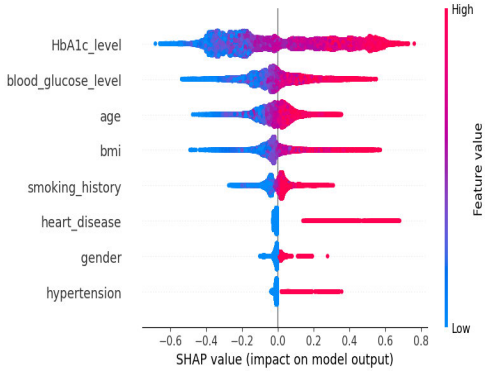


FIGURE 14. SHAP value impact on the PaRSEL using summary plot.

Figure 14 the feature names are displayed on the Y-axis from top to bottom. The X-axis shows the SHAP value, which represents the amount of change in log odds. To show the value of the relevant characteristic, each point on the graph is colored, with red suggesting high values and blue denoting low values. Each point represents a row of data from the original dataset. BMI, hypertension, blood sugar, and haemoglobin ALC levels are typically high and have a good

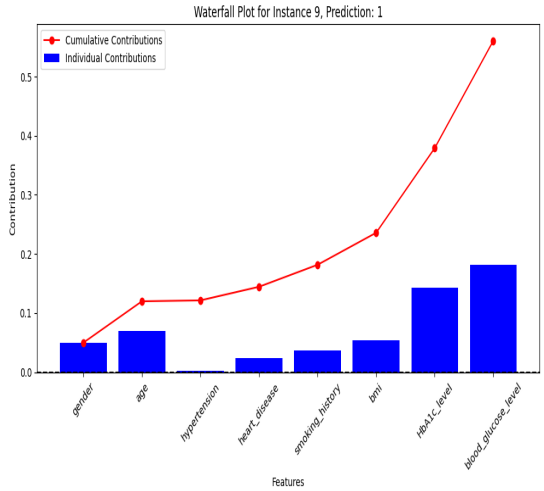


FIGURE 15. SHAP value impact on the PaRSEL using waterfall plot.

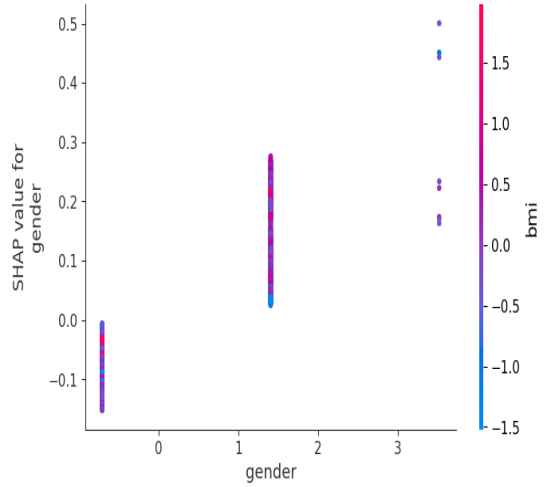


FIGURE 16. SHAP Value Impact on the PaRSEL using Dependence Plot.

SHAP value. This suggests that it has a beneficial impact on the result. Figure 15 show the individual contributions made by features used in the PaRSEL as well as the sum of all contributions that resulted in the final forecast. In Figure 16 we can notice an overall pattern that favors the ''bmi'' characteristic, where the contribution increases as the ''bmi'' value increases. This suggests that higher ''bmi'' values are beneficial for the model's forecast.

V. LIMITATIONS

In the disciplines of data science and medicine, the demand for automated diagnostic tools is on the rise. The field of medical care has benefited from a number of models developed by data scientists in an effort to save lives. The large dimensionality of the dataset is a significant challenge for ML. A considerable amount of memory is needed for the analysis of many features. The proposed stacking model PaRSEL is complex in nature which also results in overfitting. As the number of features rises, the amount of redundant data and processing time also rise. However, due to the nonlinear and high frequency data, ML-based heart disease prediction algorithms struggle with serious overfitting. Also, more computational resources are required to run a stacking model.

VI. CONCLUSION

This study suggests a new stacking model named PaRSEL that incorporates four classifiers at the base layer, PAC, RC, SGDC, and XGBoost, and one classifier, LogitBoost, at the meta layer. The redundant and inconsistent data present in the dataset, along with the unbalanced and irrelevant features, expand the search space and complicate the classification models. The data needs to be balanced, redundant data needs to be deleted, and unnecessary components need to be removed in order to improve classification accuracy. Therefore, efforts for dimensionality reduction and data balancing are crucial for reducing costs and raising accuracy. Three dimensionality reduction methods, RFE, LDA, and FA, are employed in PaRSEL to choose the most pertinent features for the diagnosis of heart disease. Eight balancing approaches are also utilised to address the dataset's imbalance nature, including ProWRAS, LoRAS, ROS, ADASYN, SMOTE, B-SMOTE, MWMOTE, and RWOS. On our suggested model, we apply SHAP. SHAP values are frequently used to get an objective and consistent explanation of how each feature affects the prediction of the model. It is beneficial to translate ML model predictions. The information of each input feature is given a value, which shows how much it affects the prediction's outcome. Using several performance metrics, including accuracy, F1-score, precision, recall, execution time, and AUC-ROC score, PaRSEL, is contrasted with other standalone classifiers. Our proposed model achieves 97% accuracy, 80% F1-score, precision is greater than 90%, 67% recall and 98% AUC-ROC score. This shows that PaRSEL outperforms other standalone classifiers in terms of heart disease prediction. Translational research aims to translate (convert) basic research results into clinical results that benefit humans directly. So, in future we will conduct clinical research to further verify the results obtained via this work.

ACKNOWLEDGMENT

The authors extend their appreciation to the Researchers Supporting Program for Project number (RSPD2023R648), at King Saud University for supporting this research project.

REFERENCES

[1] R. C. Das, M. C. Das, M. A. Hossain, M. A. Rahman, M. H. Hossen, and R. Hasan, ''Heart disease detection using ML,'' in *Proc. IEEE 13th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Mar. 2023, Art. no. 983987, doi: 10.1109/CCWC57344.2023.10099294.

[2] M. Jaffar, S. Shafiq, N. Shahzadi, N. Alrajeh, M. Jamil, and N. Javaid, ''Efficient deep learning models for predicting super-utilizers in smart hospitals,'' *IEEE Access*, vol. 11, pp. 87676–87693, 2023, doi: 10.1109/ACCESS.2023.3305379.

[3] S. K. Basha, D. Roja, S. S. Priya, L. Dalavi, S. S. Vellela, and B. V. Reddy, ''Coronary heart disease prediction and classification using hybrid machine learning algorithms,'' in *Proc. Int. Conf. Innov. Data Commun. Technol. Appl. (ICIDCA)*, Mar. 2023, p. 713, doi: 10.1109/ICIDCA56705.2023.10099579.

[4] S. Mohapatra, S. Maneesha, S. Mohanty, P. K. Patra, S. K. Bhoi, K. S. Sahoo, and A. H. Gandomi, "A stacking classifiers model for detecting heart irregularities and predicting cardiovascular disease," *Healthcare Anal.*, vol. 3, Nov. 2023, Art. no. 100133, doi: 10.1016/j.health.2022.100133.

[5] N. Javaid, M. Akbar, A. Aldegheishem, N. Alrajeh, and E. A. Mohammed, "Employing a machine learning boosting classifiers based stacking ensemble model for detecting non technical losses in smart grids," *IEEE Access*, vol. 10, pp. 121886–121899, 2022, doi: 10.1109/ACCESS.2022.3222883.

[6] K. Rahmani, R. Thapa, P. Tsou, S. C. Chetty, G. Barnes, C. Lam, and C. F. Tso, "Assessing the effects of data drift on the performance of machine learning models used in clinical sepsis prediction," *Int. J. Med. Informat.*, vol. 173, May 2023, Art. no. 104930, doi: 10.1016/j.ijmedinf.2022.104930.

[7] A. Jain, A. C. S. Rao, P. K. Jain, and Y.-C. Hu, "Optimized levy flight model for heart disease prediction using CNN framework in big data application," *Exp. Syst. Appl.*, vol. 223, Aug. 2023, Art. no. 119859, doi: 10.1016/j.eswa.2023.119859.

[8] R. Sawhney, A. Malik, S. Sharma, and V. Narayan, "A comparative assessment of artificial intelligence models used for early prediction and evaluation of chronic kidney disease," *Decis. Anal. J.*, vol. 6, Mar. 2023, Art. no. 100169, doi: 10.1016/j.dajour.2023.100169.

[9] Y. Shen, J. Zhu, Z. Deng, W. Lu, and H. Wang, "EnsDeepDP: An ensemble deep learning approach for disease prediction through metagenomics," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 20, no. 2, pp. 986–998, Mar. 2023, doi: 10.1109/TCBB.2022.3201295.

[10] A. U. Haq, J. Li, M. H. Memon, M. H. Memon, J. Khan, and S. M. Marium, "Heart disease prediction system using model of machine learning and sequential backward selection algorithm for features selection," in *Proc. IEEE 5th Int. Conf. Converg. Technol. (ICT)*, Mar. 2019, pp. 1–4, doi: 10.1109/I2CT45611.2019.9033683.

[11] A. Pati, M. Parhi, and B. K. Pattanayak, "IDMS: An integrated decision making system for heart disease prediction," in *Proc. 1st Odisha Int. Conf. Electr. Power Eng., Commun. Comput. Technology*, Jan. 2021, pp. 1–6, doi: 10.1109/ODICON50556.2021.9428958.

[12] G. T. Reddy, M. P. K. Reddy, K. Lakshmanna, R. Kaluri, D. S. Rajput, G. Srivastava, and T. Baker, "Analysis of dimensionality reduction techniques on big data," *IEEE Access*, vol. 8, pp. 54776–54788, 2020, doi: 10.1109/ACCESS.2020.2980942.

[13] V. Chaurasia and A. Chaurasia, "Novel method of characterization of heart disease prediction using sequential feature selection-based ensemble technique," *Biomed. Mater. Devices*, Jan. 2023, doi: 10.1007/s44174-022-00060-x.

[14] J. Wang, C. Liu, L. Li, W. Li, L. Yao, H. Li, and H. Zhang, "A stacking-based model for non-invasive detection of coronary heart disease," *IEEE Access*, vol. 8, pp. 37124–37133, 2020, doi: 10.1109/ACCESS.2020.2975377.

[15] C. B. C. Latha and S. C. Jeeva, "Improving the accuracy of prediction of heart disease risk based on ensemble classification techniques," *Informat. Med. Unlocked*, vol. 16, Jan. 2019, Art. no. 100203, doi: 10.1016/j.imu.2019.100203.

[16] M. Trigka and E. Dritsas, "Long-term coronary artery disease risk prediction with machine learning models," *Sensors*, vol. 23, no. 3, p. 1193, Jan. 2023, doi: 10.3390/s23031193.

[17] M. A. Kadam, S. Patil, P. Pethkar, R. Shikare, and S. Sarnayak, "A cardiovascular disease prediction system using machine learning," *J. Pharmaceutical Negative Results*, pp. 7216–7225, Jan. 2023.

[18] D. Hassan, H. I. Hussein, and M. M. Hassan, "Heart disease prediction based on pre-trained deep neural networks combined with principal component analysis," *Biomed. Signal Process. Control*, vol. 79, Jan. 2023, Art. no. 104019, doi: 10.1016/j.bspc.2022.104019.

[19] S. Bej, K. Schulz, P. Srivastava, M. Wolfien, and O. Wolkenhauer, "A multi-schematic classifier-independent oversampling approach for imbalanced datasets," *IEEE Access*, vol. 9, pp. 123358–123374, 2021, doi: 10.1109/ACCESS.2021.3108450.

[20] S. Bej, N. Davtyan, M. Wolfien, M. Nassar, and O. Wolkenhauer, "LoRAS: An oversampling approach for imbalanced datasets," *Mach. Learn.*, vol. 110, no. 2, pp. 279–301, Feb. 2021, doi: 10.1007/s10994-020-05913-4.

[21] A. Ullah, N. Javaid, M. U. Javed, B.-S. Kim, and S. A. Bahaj, "Adaptive data balancing method using stacking ensemble model and its application to non-technical loss detection in smart grids," *IEEE Access*, vol. 10, pp. 133244–133255, 2022, doi: 10.1109/ACCESS.2022.3230952.

[22] Y. Li, Y. Wang, T. Li, B. Li, and X. Lan, "SP-SMOTE: A novel space partitioning based synthetic minority oversampling technique," *Knowl.-Based Syst.*, vol. 228, Sep. 2021, Art. no. 107269, doi: 10.1016/j.knosys.2021.107269.

[23] Y. E. Kurniawati, A. E. Permanasari, and S. Fauziati, "Adaptive synthetic-nominal (ADASYN-N) and adaptive synthetic-KNN (ADASYN-KNN) for multiclass imbalance learning on laboratory test data," in *Proc. 4th Int. Conf. Sci. Technol. (ICST)*, Aug. 2018, pp. 1–6, doi: 10.1109/ICSTC.2018.8528679.

[24] A. Arafa, N. El-Fishawy, M. Badawy, and M. Radad, "RN-SMOTE: Reduced noise SMOTE based on DBSCAN for enhancing imbalanced data classification," *J. King Saud Univ.-Comput. Inf. Sci.*, vol. 34, no. 8, pp. 5059–5074, Sep. 2022, doi: 10.1016/j.jksuci.2022.06.005.

[25] A. Moreo, A. Esuli, and F. Sebastiani, "Distributional random oversampling for imbalanced text classification," in *Proc. 39th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, Jul. 2016, Art. no. 805808, doi: 10.1145/2911451.2914722.

[26] S. Barua, Md. M. Islam, X. Yao, and K. Murase, "MWMOTE-majority weighted minority oversampling technique for imbalanced data set learning," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 2, pp. 405–425, Feb. 2014, doi: 10.1109/TKDE.2012.232.

[27] S. Smiti and M. Soui, "Bankruptcy prediction using deep learning approach based on borderline SMOTE," *Inf. Syst. Frontiers*, vol. 22, no. 5, pp. 1067–1083, Oct. 2020, doi: 10.1007/s10796-020-10031-6.

[28] H. Zhang and M. Li, "RWO-Sampling: A random walk over-sampling approach to imbalanced data classification," *Inf. Fusion*, vol. 20, no. 1, pp. 99–116, 2014, doi: 10.1016/j.inffus.2013.12.003.

[29] X. Ding, F. Yang, and F. Ma, "An efficient model selection for linear discriminant function-based recursive feature elimination," *J. Biomed. Informat.*, vol. 129, May 2022, Art. no. 104070, doi: 10.1016/j.jbi.2022.104070.

[30] M. M. Rahman, A. K. Sarkar, M. A. Hossain, and M. A. Moni, "EEG-based emotion analysis using non-linear features and ensemble learning approaches," *Exp. Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 118025, doi: 10.1016/j.eswa.2022.118025.

[31] Y.-J. Chen and Y.-M. Chen, "Forecasting corporate credit ratings using big data from social media," *Exp. Syst. Appl.*, vol. 207, Nov. 2022, Art. no. 118042, doi: 10.1016/j.eswa.2022.118042.

[32] N. Shrestha, "Factor analysis as a tool for survey analysis," *Amer. J. Appl. Math. Statist.*, vol. 9, no. 1, pp. 4–11, Jan. 2021, doi: 10.12691/ajams-9-1-2.

[33] B. Behera and G. Kumaravelan, "Performance evaluation of machine learning algorithms in biomedical document classification," *Int. J. Adv. Sci. Technol.*, vol. 29, no. 5, pp. 5704–5716, 2020.

[34] *The Mathematical Background of Lasso and Ridge Regression*. Accessed: May 8, 2023. [Online]. Available: https://medium.com/codex/mathematical-background-of-lasso-and-ridge-regression-23b74737c817

[35] *Stochastic Gradient Descent*. Accessed: May 7, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Stochastic_gradient_descent

[36] D. K. Choi, "Data-driven materials modeling with XGBoost algorithm and statistical inference analysis for prediction of fatigue strength of steels," *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 1, pp. 129–138, Jan. 2019, doi: 10.1007/s12541-019-00048-6.

[37] *XGBoost Mathematics Explained*. Accessed: May 8, 2023. [Online]. Available: https://dimleve.medium.com/xgboost-mathematics-explained-58262530904a

[38] *LogitBoost*. Accessed: May 8, 2023. [Online]. Available: https://en.wikipedia.org/wiki/LogitBoost

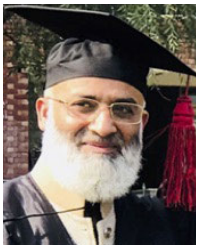
[39] K. E. Hoque and H. Aljamaan, "Impact of hyperparameter tuning on machine learning models in stock price forecasting," *IEEE Access*, vol. 9, pp. 163815–163830, 2021, doi: 10.1109/ACCESS.2021.3134138.

[40] 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, doi: 10.1109/ACCESS.2019.2952107.

[41] D. Fryer, I. Strümke, and H. Nguyen, "Shapley values for feature selection: The good, the bad, and the axioms," *IEEE Access*, vol. 9, pp. 144352–144360, 2021, doi: 10.1109/ACCESS.2021.3119110.

[42] N. Nayyer, N. Javaid, M. Akbar, A. Aldegheishem, N. Alrajeh, and M. Jamil, "A new framework for fraud detection in Bitcoin transactions through ensemble stacking model in smart cities," *IEEE Access*, vol. 11, pp. 90916–90938, 2023, doi: 10.1109/ACCESS.2023.3308298.

AYESHA NOOR received the bachelor's degree in computer science from Fatima Jinnah Women University, Rawalpindi, in 2019. She is currently pursuing the master's degree in information security with the Communication over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad, under the supervision of Prof. Nadeem Javaid. Her research interests include data science, the IoT, and machine learning.



NADEEM JAVAID (Senior Member, IEEE) received the bachelor's degree in computer science and physics from Gomal University, Dera Ismail Khan, Pakistan, in 1995, the master's degree in electronics from Quaid-i-Azam University, Islamabad, Pakistan, in 1999, and the Ph.D. degree from the University of Paris-Est, France, in 2010. He was a Visiting Professor with the University of Technology Sydney, Australia. He is currently a Tenured Professor and the Founding Director of the Communication over Sensors (ComSens) Research Laboratory, Department of Computer Science, COMSATS University Islamabad, Islamabad Campus. He has 25 years of teaching and research experience. He has supervised 187 master's and 30 Ph.D. theses. He has authored more than 950 papers in technical journals and international conferences. His research interests include energy optimization in smart/microgrids and wireless sensor networks using data analytics and blockchain. He was a recipient of the Best University Teacher Award (BUTA16) from the Higher Education Commission (HEC), Pakistan, in 2016, and the Research Productivity Award (RPA17) from the Pakistan Council for Science and Technology (PCST), in 2017. He is an Editor of *Sustainable Cities and Society*.

NABIL ALRAJEH received the Ph.D. degree in biomedical informatics engineering from Vanderbilt University, Nashville, TN, USA. He was a Senior Advisor with the Ministry of Higher Education, where the role was implementing development programs, including educational affairs, strategic planning, and research and innovation. He was a member of the boards of trustees for five private universities in Saudi Arabia. He is currently a Professor of health informatics with King Saud University. His research interests include e-health applications, hospital information systems, telemedicine, healthcare applications of smart cities, and wireless sensor networks.

BABAR MANSOOR received the B.Sc. degree in computer engineering from the COMSATS Institute of Information Technology (CIIT), Abbottabad, Pakistan, in 2005, the M.Sc. degree in microelectronic systems from the University of Liverpool, U.K., in 2007, and the Ph.D. degree in electrical engineering from COMSATS University Islamabad (CUI), Islamabad, Pakistan, in 2018. He is currently an Assistant Professor with the Department of Electrical and Computer Engineering, CUI. His research interests include wireless communications, signal processing, and MIMO communication systems.

ALI KHAQAN is currently an Assistant Professor with the Department of Electrical and Computer Engineering, COMSATS University Islamabad, Islamabad. His research interests include control systems, structure systems, and drug delivery systems.

SAFDAR HUSSAIN BOUK received the bachelor's degree in computer systems engineering from the Mehran University of Engineering and Technology (MUET), Jamshoro, in 2001, he was awarded with the Japanese Government's Monbukagakusho (MEXT) Scholarship for graduate studies, in 2005, and the master's and Ph.D. degrees in engineering from the Graduate School of Science and Technology, Keio University, Yokohama, Japan, in 2007 and 2010, respectively. He was a Visiting Scholar with UIUC, USA, in summer 2001. He was an Assistant Professor with the COMSATS Institute of Information Technology, Islamabad, Pakistan, from 2010 to 2016. In addition to that, he was a Research Professor with DGIST, Daegu, South Korea. He is currently a Research Assistant Professor with the Virginia Modeling Analysis and Simulation Center (VMASC), Old Dominion University, VA, USA. His research interests include future inter-net architectures, 5G, and NextG networks, and security perspectives in future networks.