IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 2 July 2022, accepted 11 July 2022, date of publication 18 July 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3191669

RESEARCH ARTICLE

An Effective Heart Disease Detection and Severity Level Classification Model Using Machine Learning and Hyperparameter Optimization Methods

ABDALLAH ABDELLATIF[1], (Member, IEEE), HAMDAN ABDELLATEF[2], (Member, IEEE), JEEVAN KANESAN[1], CHEE-ONN CHOW[1], (Senior Member, IEEE), JOON HUANG CHUAH[1], (Senior Member, IEEE), AND HASSAN MUWAFAQ GHENI[3]

[1]Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Kuala Lumpur 50603, Malaysia
[2]Electrical and Computer Engineering Department—School of Engineering, Lebanese American University, Byblos, Lebanon
[3]Computer Techniques Engineering Department, Al-Mustaqbal University College, Hillah 51001, Iraq

Corresponding author: Jeevan Kanesan (jievan@um.edu.my)

This work was supported by the Ministry of Higher Education through the Fundamental Research Grant Scheme under Grant FRGS/1/2020/ICT02/UM/02/2.

ABSTRACT Cardiovascular disease (CVD) is the leading cause of death worldwide. A Machine Learning (ML) system can predict CVD in the early stages to mitigate mortality rates based on clinical data. Recently, many research works utilized different machine learning approaches to detect CVD or identify the patient's severity level. Although these works obtained promising results, none focused on employing optimization methods to improve the ML model performance for CVD detection and severity-level classification. This study provides an effective method based on the Synthetic Minority Oversampling Technique (SMOTE) to handle imbalance distribution issue, six different ML classifiers to detect the patient status, and Hyperparameter Optimization (HPO) to find the best hyperparameter for ML classifier together with SMOTE. Two public datasets were used to build and test the model using all features. The results show that SMOTE and Extra Trees (ET) optimized using hyperband achieved higher results than other models and outperformed the state-of-the-art works by achieving 99.2% and 98.52% in CVD detection, respectively. Also, the developed model converged to 95.73% severity classification using the Cleveland dataset. The proposed model can help doctors determine a patient's current heart disease status. As a result, it is possible to prevent heart disease-related mortality by implementing early therapy.

INDEX TERMS CVD detection, severity classification, hyperparameter optimization, extra trees, imbalance, hyperband.

I. INTRODUCTION

Heart disease, also known as cardiovascular disease (CVD), is the leading cause of death globally. In a recent study, the World Heart Federation found that one in three deaths occurs due to CVD [1]. According to World Health Organization (WHO) statistics, by 2030, more than 23.6 million people may die from CVD, mainly from strokes and heart failure [2]. CVD may be caused by various factors, including stress, alcohol, smoking, lack of a healthy diet, inactive lifestyle, and other related health problems such as high blood pressure or diabetes. However, most CVD-related diseases are known to be completely curable once diagnosed in their early stages [3]. In this case, healthcare providers must place a greater emphasis on heart failure diagnosis and prediction. By evaluating a patient's health record, emerging novel strategies for data analysis may enable early diagnosis of CVD [2].

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

In general, health status can be measured by assessing it to the lowest-risk level, the highest-risk level, the presence of primary lesions, and finally, the defective diagnosis. The detection process can take longer than expected, associated with genetic aspects of people's social and medical status, natural factors, and personal lifestyle. However, current health care may be derived from predicting and evaluating risk factors to avoid illness as it progresses to a more severe level to achieve better health outcomes [4]. This leads to developing a high-accurate system that analyzes the heart disease (HD) clinical data to detect heart failure levels.

Various studies used Machine Learning (ML) algorithms to forecast CVD using clinical datasets. Still, clinical datasets present considerable difficulties owing to class imbalance and their high dimensionality. As a result, employing machine learning without addressing these challenges reduces the efficiency and thus accuracy of the methods. Prior researchers focused on feature selection (FS) and used several ML systems to predict CVD. Long *et al.* [5] developed a heart disease decision system based on Rough Sets (RS) and Choas firefly algorithm (CFARS-AR) to select the best features and then employed a type-2 fuzzy logic system for HD detection. The developed model achieved 88.3% accuracy. The authors in [6] combine the RS with a backpropagation neural network (BPNN) to predict CVD. In addition, Dwivedi performed a comparative study on HD prediction using different ML models, such as artificial neural network (ANN), Logistic Regression (LR), classification trees, and Naïve Bayes (NB). The author concludes that LR performed better than the other models for CVD detection [7].

Also, Haq *et al.* conducted a comparative study using different FS methods, such as relief, with different ML models such as ANN, RF, and LR. The authors reported that removing features affects the models' performance. The study concluded that LR with relief FS achieved 89% accuracy compared to other models used in the same study [8]. Amin *et al.* performed a comparative analysis by identifying the significant features and employed seven ML models, including ANN, LR, Decision Trees (DT), Support Vector Machine (SVM), k-Nearest Neighbor (kNN), NB, and a hybrid model (vote with LR & NB) [9]. The results showed that the hybrid model achieved the best accuracy (87.41%). Another study developed a hybrid model consisting of RF with a linear model (HRFLM) to improve CVD prediction accuracy. The conducted research was applied to different combinations of features; the proposed method achieved accuracy up to 88.4% [10]. Afterwards, Vijayashree *et al.* introduced a novel function for determining optimum weights dependent on the population diversity and optimization technique. They also introduced an improved PSO fitness function with the help of SVM, which reduced the number of attributes. As a result, their method achieved accuracy up to 84.36% [11].

In a successful attempt to develop better accuracy, Ali *et al.* improved an HD diagnosis process by introducing two stacked SVMs. The first SVM removed the non-significant attributes, while the second SVM was utilized to predict the presence and absence. As a result, the Stacked developed model achieved 92.22% accuracy [12]. Moreover, Gupta *et al.* created a machine intelligence framework for HD diagnosing based on FAMD and RF. FAMD selected the significant features, and RF predicted the absence and presence of HD. The outcomes showed that the proposed model achieved 93.44% accuracy [13]. The authors in [14] undertook comparative analysis research. On various datasets, the authors used multiple classifiers. The conditional inference tree forest (cforest) outperformed the other classifiers. Tama *et al.* [15] introduced a 2-tier ensemble model with particle swarm optimization (PSO) based feature selection for heart disease prediction. The developed model reached accuracy up to 93.55%. Nevertheless, due to the imbalanced clinical datasets, the abovementioned studies have certain limitations in detecting HD using the methodologies suggested.

Few studies developed support decision systems, including a balancing technique to address the problem mentioned. Fitriyani *et al.* developed an HD prediction method, including density-based spatial clustering of applications with noise (DBSCAN), hybrid synthetic minority over-sampling technique-edited nearest neighbor (SMOTE-ENN) to detect and eliminate outliers and balance data distribution. In addition, the XGBoost classifier forecasts the patient status. The developed method attained an accuracy of 95.9% [16]. Waqar *et al.* proposed SMOTE-based deep learning to predict heart attacks. The author used SMOTE technique to balance the dataset without feature selection. The balanced dataset was trained and tested by a deep neural network to predict the absence and presence of a heart attack and obtained 96% accuracy [17]. Recently, Ishaq *et al.* used SMOTE to balance data distribution and extremely randomized trees (ET) on selected parameters to predict patient survival using RF importance ranking [18].

Moreover, some researchers used the hybrid method to detect the patients' severity levels. Salari *et al.* introduced a hybrid method including a genetic algorithm (GA) for feature selection, a modified kNN, and a backpropagation neural network for severity classification. The results revealed that the method obtained 62.1% accuracy [19]. In the same context, Wiharto *et al.* proposed a hybrid method based on the binary tree (BT) and SVM for predicting the sickness level of HD. The study obtained a 61.86% accuracy [20]. Unlikely, Khateeb *et al.* implemented a re-sampling filter for the data and then utilized kNN (IBK) to predict the severity level. The model obtained accuracy up to 79.20% [21]. Magesh and Swarnalatha developed an optimal FS method based on cluster-based DT learning (CDTL) to find the significant features and then employed the RF to predict the severity level. The developed model achieved 89.3% accuracy for severity level predictions [22]. Recently, the authors in [23] developed a decision system for HD severity level prediction based on an ML-based fusion approach. The results showed that the developed model achieved 75% accuracy for severity

TABLE 1. A chronological review of existing heart disease diagnosis systems.

Study	Method	Target	Data Balance	Hyperparameter Optimization	Validation	Dataset
[5]	CFARS-AR	Binary	No	No	No	Statlog, SPECTF
[6]	RS-BPNN	Binary	No	No	10 CV	Statlog
[7]	LR	Binary	No	No	10 CV	Statlog
[8]	Relief + LR	Binary	No	No	10 CV	Cleveland
[9]	Vote with NB + LR	Binary	No	No	10 CV	Cleveland, Statlog
[10]	HRFLM	Binary	No	No	Not mentioned	Cleveland, Hungarian, Long-beach-va, and Switzerland
[11]	SVM-PSO based FS	Binary	No	No	Not mentioned	Cleveland
[12]	Stacked SVM	Binary	No	No	Holdout	Cleveland
[13]	FAMD + RF	Binary	No	No	Holdout	Cleveland
[15]	Two-tier ensemble PSO based FS	Binary	No	No	10 CV	Statlog, Z-Alizadeh Sani, Cleveland, Hungarian
[14]	CF	Binary	No	No	10 CV	Statlog
[16]	DBSCAN + SMOTE-ENN + XGBoost	Binary	Yes	No	10 CV	Statlog, Clevelan
[18]	SMOTE + RF +ET	Binary	Yes	No	10 CV	HD clinical record
[17]	SOMTE- ANN	Binary	Yes	No	Not mentioned	Cleveland
[19]	GA + Modified KNN	Binary, Multiclass	No	No	Not mentioned	Cleveland, Hungarian
[20]	BT-SVM	Multiclass	No	No	10 CV	Cleveland
[21]	Resampling + kNN	Multiclass	Yes	No	10 CV	Cleveland
[22]	CDTL-RF	Multiclass	No	No	Not mentioned	Cleveland
[23]	LR + RF	Binary, Multiclass	Yes	No	Different data split	Cleveland
Current Study	HB-SMOTE-ET	Binary, Multiclass	Yes	Yes	10 CV	Statlog, Cleveland

level prediction. The studies mentioned in the literature are summarized in Table 1 regarding the method used, the target, dataset used, data balancing and hyperparameters tuning.

Most studies focused on the importance of HD attributes in selecting the best factors and the outlier's removal to enhance the ML model performance. However, in the field of ML, imbalance data and hyperparameter tuning may arise and impact the prediction model's performance since many ML parameters have different optimums to reach the best performance on various tasks and datasets. Mainly for complex ML systems with many hyperparameters and large datasets. Still, none of the previous studies integrated optimization methods with data balancing and ML techniques for model Hyperparameter Optimization (HPO) to influence the model's prediction performance.

Previous studies have reported that by integrating hyperband (HB) for ML HPO [24], [25], the model classification performance was significantly improved compared to Bayesian optimization (BO) and Random search (RS). However, to our knowledge, no studies integrated HB for optimizing SMOTE and ML to predict the presence and severity level of CVD. This article is considered the first paper to employ HPO in CVD detection and severity classification. This study draws on all the attributes concerning heart failure and develops an ML model to forecast heart failure and predict heart disease severity. We apply SMOTE to assist with the imbalance problem in both forecast problems. We employ six ML algorithm variations, SVM, Stochastic Gradient Descent (SGD), kNN, Extra Trees (ET), XGBoost, and LR, to predict heart failure and severity levels. The selected classifiers, together with SMOTE, were optimized using HB to find the best performing hyperparameters since there are a variety of optimum values for ML hyperparameters for different datasets. The main contribution of this paper can be summarized as follows:

- This study proposed an effective decision support system integrating SMOTE, ML classifier and HB method. Combining these techniques is thought to improve the efficiency of existing methods for predicting CVD using clinical datasets.
- The HB method finds the best SMOTE hyperparameters for a suitable number of synthesized samples achieving the highest prediction with an optimized classifier.
- The impact of tree-based, statistical-based, and regression-based models with different optimization algorithms, including HB, improved Particle Swarm Optimization (PSO) [26], [27], GA, and RS [28], is discussed in predicting CVD.
- The proposed model was evaluated using Cleveland and Statlog datasets. Hence, various comparisons with earlier studies are performed to prove the effectiveness of the proposed model.

The rest of the paper is structured as follows: Section 2 describes the proposed method. Section 3 describes experimental design and performance evaluation metrics. Section 4 describes the results and discussion of the

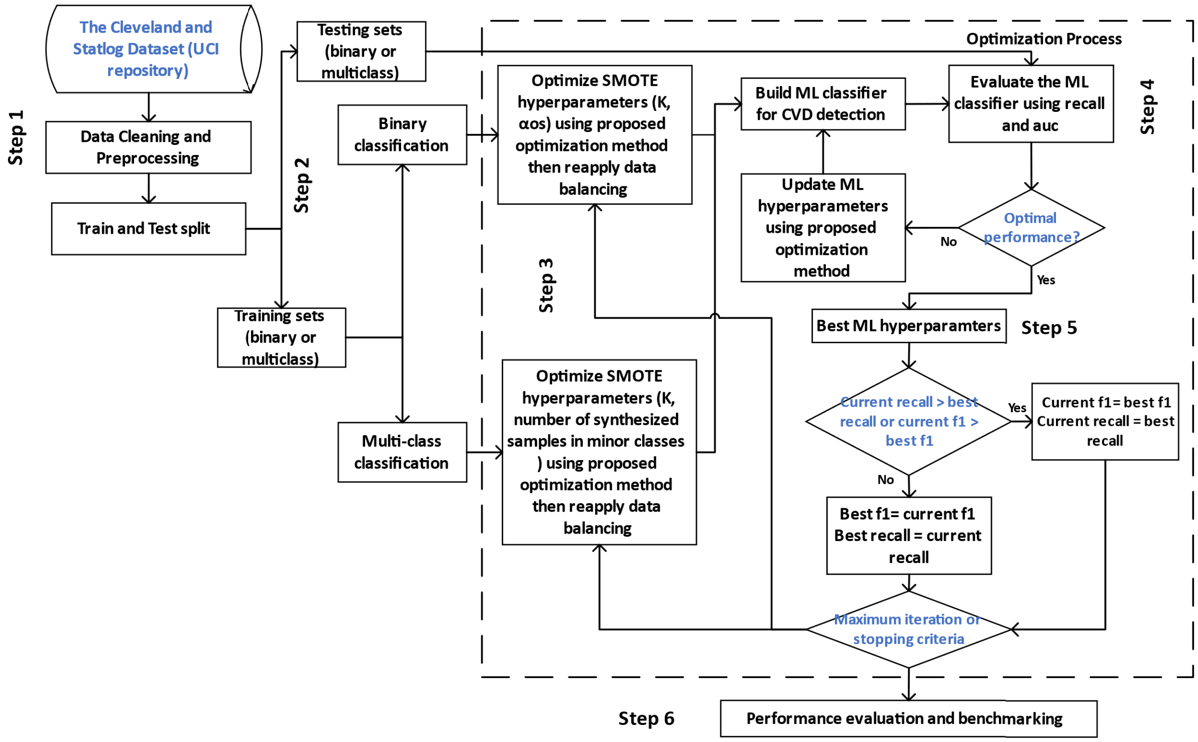

FIGURE 1. The proposed flowchart for the heart disease detection and patients' survival.

conducted experiments and the comparison with the state-of-art. Finally, the conclusion and future work are presented in section 5.

II. PROPOSED METHODOLOGY

The proposed method is developed to obtain a high-performance heart disease prediction in two cases, the CVD's absence or presence and CVD's severity levels. Figure 1 presents the flowchart of the proposed system. We evaluate the proposed method with the Cleveland and Statlog datasets collected from the UCI (University of California Irvine) machine learning repository [29], [30]. The following steps summarize the flowchart:

Step 1: The data is collected from the UCI database, cleaned and preprocessed, including removing missing data, label grouping, and data normalization using min-max.

Step 2: After data preprocessing, the data is split into training and testing.

Step 3: The training data is then entered into the SMOTE technique for data distribution balancing

Step 4: The ML model is built and evaluated using recall and AUC; when the ML model reaches the optimal performance, it will be selected. Otherwise, it will return to update the hyperparameters.

Step 5: After the best ML hyperparameters is selected, the ML model enters the criteria for updating SMOTE hyperparameters. If the ML model satisfies the stopping criteria, it will proceed to the next step. Otherwise, it returns to step 3 for the updating SMOTE hyperparameters.

Step 6: In the last step, the model is evaluated and benchmarked with the existing models.

A. MACHINE LEARNING CLASSIFIERS

Different ML-based algorithms are employed to predict CVD status and severity level in this study. To improve the accuracy of the classifiers, HPO was applied.

1) SUPPORT VECTOR CLASSIFIER (SVC)

SVM is a well-known supervised learning algorithm used to solve complex problems, including regression and classification. SVM maps the data points from low-dimensional to high-dimensional space to distinguish them linearly; a hyperplane is then created as a classification border for the data points [31]. SVM objective function for n number of points is

$$arg \min_{\omega} \left\{ \frac{1}{n} \sum_{i=1}^{n} \max\left\{0, 1 - y_i f\ (x_i)\right\} + C\omega^T \omega \right\} \tag{1}$$

The SVM objective comprises two parts loss function and regularization. The regularization coefficient (C) and normalization vector (ω) are vital hyperparameters in SVM models. C is an internal variable initialized before SVM training; it establishes a trade-off between penalty and separation margin incorrect forecasts.

In addition, the kernel function f(x) measures the similarity between data points (x_i, x_j) selected from different kernel types in the SVM model. The most SVM kernel types used include linear, polynomial, radial basis function (RBF), and sigmoid kernels [28], [31]. The coefficient γ (gamma) is a conditional kernel hyper-parameter for RBF, Polynomial, or Sigmoid kernels. Gamma determines the curvature in a decision boundary.

2) K-NEAREST NEIGHBOR (kNN)

kNN is applied to classify data points based on the distances between them. This method uses the distance technique to classify unseen data and then correlates it with trained data to generate the classification result [32]. Classification is accomplished by the highest number of neighbors belonging to the same class, determined by the k value. Also, the weight function in the prediction can be set depending on the problem, where the points can be weighted equally or by the inverse of their distance.

3) LOGISTIC REGRESSION (LR)

LR is a linear model deals with classification tasks. It is a regression-based algorithm for predictive analysis that is based on the concept of probability [33]. LR is often used for binary data to discover the output of one or more features. According to the regularisation approach used, the cost function of LR may be different. Elastic-net, L1-norm, L2-norm regularization are the main three types of regularization. Also, two main elements define LR, the regularization coefficient (C) and solver, where C explains the regularization strength and solver represents the optimization algorithm.

4) STOCASTIC GRADIENT DESCENT (SGD)

SGD employs the one-versus-all strategy to combine several binary classifiers. SGD has been frequently utilized for datasets since it uses all samples in each iteration [34]. The working concept is similar to the regression approach; thus, it is straightforward to understand and implement. In a loss function, a regularizer is a penalty that reduces model parameters toward zero using either the absolute norm (L1) or the squared euclidean norm (L2) or a combination of both (Elastic Net). SGD hyperparameters must be appropriately valued to get reliable results, including the regularization coefficient alpha that explains the regularization and is utilized to compute the learning rate. In terms of feature scaling sensitivity of SGD is high.

5) TREE-BASED ENSEMBLE

DT is a widely used classification technique that employs a tree structure to describe actions and their probable repercussions by summarising a set of classification rules extracted from the data. A DT comprises three components: a root node reflecting the complete data set, multiple decision nodes representing sub-node splits and decision tests for each attribute, and several leaf nodes representing the outcome classes. DT algorithms iteratively divide the training set into subsets with superior feature values so that each subset may be used to make good decisions. To minimize over-fitting, DT uses pruning, which implies eliminating part of the subnodes of decision nodes. Because a deeper tree contains more sub-trees that enable more precise decision-making, the maximum tree depth is a critical hyper-parameter that regulates the complexity of the model.

Numerous decision-tree-based ensemble techniques, including ET, RF, and XGBoost models, have been developed to increase model performance by mixing multiple decision trees. The bagging approach integrates many decision trees in RF. ET [35] is a different tree-based ensemble learning approach similar to RF in that it builds DTs from all samples and picks feature sets at random. Additionally, RF optimizes DT splits, whereas ET produces splits at random.

The XGBoost model is a widely used tree-based ensemble that employs boosting and gradient descent to integrate basic DTs for speed and performance enhancement [36]. In XGBoost, the current DT's input sample depends on the prior DTs' results. Where it intends to minimize the objective function presented below:

$$\mathrm{Obj} = -\frac{1}{2}\sum\nolimits_{j=1}^{t}\frac{G_j^2}{H_j+\lambda}+\gamma t \tag{2}$$

where G and H are the sums of the cost function's first and second-order gradient statistics, t presents the number of leaves in DT, and two penalty coefficients λ and γ. Since the tree-based ensemble models are constructed with DT base learners, the number of DT needs to be combined needs to be set for the mentioned models to achieve accurate results.

B. HYPERBAND OPTIMIZATION METHOD

The Hyperband (HB) method [37] is developed to solve successive halving (SH) algorithms' dilemmas by selecting reasonable configurations dynamically. The objective is to strike a balance between the hyper-parameter configurations (n) and their budget allocations (b) through splitting a whole number of budgets (B) into n bits and allocating one to each structure (b = B/n). SH executes as a subprocess to each series of randomly generated structures to remove the hyper-parameter group that performs poorly and enhances accuracy. Algorithm 1 shows the main steps of the Hyperband algorithm [33].

The budget limits b_{max} and b_{min} are decided by the whole data points, the available budgets, and the minimum number of cases needed to train a sensible model. Following that, in steps two and three of Algorithm 1, the budget size (s) and the number of configurations (n) assigned to each configuration are calculated depending on budget constraints. Then in steps four and five, the configurations are sampled based on budget allocations (b) and the number of configurations (n), then sent to the SH model. The SH algorithm passes the structures that perform well to the next iteration while removing the detected poorly-performing configurations. This algorithm stops upon finding the best hyper-parameter configurations or reaching the maximum iteration.

Algorithm 1 Hyperband Algorithm Using SH.

input: budgets b_{min}, b_{max}, and η
Output: best configuration
$s_{max} = \left[log_\eta \frac{b_{max}}{b_{min}} \right]$
for $s \in (s_{max}, s_{max} - 1, \ldots, 0)$ **do**
    determine budgets $n = \left[\frac{s_{max}+1}{s+1} * \eta^s \right]$
    γ = sample configurations(n)
    run SH on γ with $\eta^s.b_{max}$ as initial budget
End
return best configuration

Finally, the Hyperband time complexity is calculated through big-O-notation linearithmic form $O(n\log n)$, including the SH searching technique [37].

1) HYPERPARAMETER OPTIMIZATION FORMULATION

ML tasks may be summarised as training a model M that minimizes a preset loss function $L(X_{Ts}; M)$ on a given test set X_{Ts} where the loss function (L) is the error rate. Learning algorithm A uses a training set X_{Tr} to create the model M, often solving the nonconvex optimization problem. The learning algorithm A consists of some hyperparameters lambda λ, where the model M is expressed as $M = A(X_{Tr}; \lambda)$. Hyperparameter optimization aims to find the best configurations $\lambda*$ that gives an optimum model $M*$ that minimizes $L(X_{Ts}; M)$. The hyperparameter optimization formulation is as follows:

$$\begin{aligned} \lambda^* &= \underset{\lambda}{\operatorname{argmin}}\, L\left(X_{Ts}; A\left(X_{Tr}; \lambda\right)\right) \\ &= \underset{\lambda}{\operatorname{argmin}}\, F\left(X_{Ts}, X_{Tr}, A, \lambda, L\right) \end{aligned} \tag{3}$$

where F is the model objective function takes λ, a tuple of hyperparameters and associated loss returns. The loss function L and the learning algorithm are chosen, and the datasets X_{Ts}, X_{Tr} are given [38]. Those factors depend on the model selected and the hyperparameter search space; the selected ML classifiers and SMOTE are presented in Table 2 and Table 3, respectively.

C. SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE (SMOTE)

The SMOTE approach is an oversampling technique that has been frequently utilized in medicine to cope with imbalanced class datasets [39]. SMOTE generates random synthetic data of the minority class from its nearest neighbo***u***rs using Euclidean distance to increase data instances. Because new samples are created based on original characteristics, they ***r***esemble the original data [40]. SMOTE is widely regarded as one of the most reliable and effective preprocessing methods in machine learning and data analyses. Algorithm 2 shows the main steps of SMOTE algorithm [40].

TABLE 2. Selected hyperparameter with search space for ML classifiers.

Model	Selected Hyperparameter	Type	Search Space
SVM	Kernel	Categorical	['linear',' sigmoid',' poly',' rbf']
	C (penalty par.)	Continuous	[0.1,20]
	gamma (kernel paramter)	Categorical	[0.05,0.2]
kNN	n_neighbors	Discrete	[1,20]
	weight	Categorical	['uniform', 'distance']
LR	penalty	Categorical	['L1', 'L2']
	C	Continuous	[0.1,20]
	solver	Categorical	['liblinear', 'lbfgs', 'sag'] (binary) ['lbfgs', 'sag', 'newton-cg'] (multiclass)
SGD	Alpha	Continuous	[0.0001,0.1]
	penalty	Categorical	['L1', 'L2']
ET	n_estimators	Discrete	[10, 100]
	min_samples_splits	Discrete	[2, 7]
	min_samples_leaf	Discrete	[1, 7]
	max-depth	Discrete	[5, 35]
	max-features	Discrete	[1, 12]
	criterion	Categorical	['gini', 'entropy']
XGBoost	n_estimators	Discrete	[10, 100]
	Learning rate	Continuous	[0.01, 0.5]
	subsample	Continuous	[0.5, 1]
	colsample_bytree	Continuous	[0.5, 1]
	max_depth	Discrete	[5, 30]

TABLE 3. The selected hyperparameters for SMOTE with the search space.

Balancing Method	Selected Hyperparameter	Type	Search Space
SMOTE	k_neighbors	Discrete	[2,10] (for binary classification) [2,9] (for multiclass classification)
	α_{os} (binary classification)	Continous	[0.9, 1]
	Sampling strategy (multiclass classification)	Discrete	1: [40, 70] 2: [35, 60] 3: [35, 60] 4: [10, 45]

III. EXPERIMENTAL SETUP AND EVALUATION METRICS

This subsection presents the two experiments that have been performed on the Cleveland and Statlog datasets to detect HD and Cleveland dataset to classify the severity level of CVD and the performance evaluation.

A. CVD DETECTION

In the first experiment, we converted the multiclass label to binary labels found in the Cleveland dataset. The value 0 represents the absence of heart disease with 160 observations, and the values 1 to 4 represent the subjects diagnosed with heart disease with 137 instances. The Statlog dataset was taken with no changes, including 150 and 120 regarding absence and presence, respectively. The data train-test split ratio is 0.25. The training sets were balanced using SMOTE. Then ML models were applied to the train datasets and optimized using the HB algorithm to achieve ultimate performance. The HB optimization method is employed to

TABLE 4. Classifiers' comparison before and after applying SMOTE on Cleveland and Statlog datasets.

Model	Cleveland				Statlog			
	Before SMOTE		After SMOTE		Before SMOTE		After SMOTE	
	Acc.	F_1	Acc.	F_1	Acc.	F_1	Acc.	F_1
LR	85.33	80.7	88	84.65	88.89	85	88.89	84.48
SGD	86.22	82.27	88.44	85.34	85.806	80.26	89.506	85.2
ET	90.22	87.24	93.78	92.18	90.123	86.003	**91.97**	**89.18**
XGBoost	**94.22**	**92.82**	**94.67**	**93.403**	**91.356**	**87.943**	**91.97**	89.126
kNN	86.67	82.76	92.89	91.07	85.19	78.95	87.65	82.336
SVM	88	84.21	92.44	90.48	87.04	82.05	88.27	83.71

TABLE 5. Classifiers' comparison before and after applying SMOTE on Cleveland and Statlog datasets.

Model	HB			PSO			GA			RS		
	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC
LR	96.002	95.03	0.916	95.46	94.36	0.905	92.89	91.21	0.852	90.22	87.58	0.79
SGD	96.53	95.6	0.927	96	95.036	0.916	94.22	92.89	0.880	93.77	92.25	0.871
ET	**99.2**	**99**	**0.983**	98.40	98.19	0.97	98.67	98.34	0.972	97.77	97.25	0.954
XGBoost	98.4	98.19	0.97	98.93	98.66	0.978	98.22	97.81	0.963	98.22	97.78	0.963
kNN	96.8	96.01	0.933	97.33	96.68	0.944	94.22	92.82	0.879	96	94.97	0.916
SVM	96.8	96.18	0.936	96.53	95.68	0.927	95.55	94.48	0.907	94.66	93.22	0.888

Algorithm 2 SMOTE Algorithm.

input: T = Number of Minority Instances;
N =Number of Synthetic Samples;
K= Number of Nearest Neighbors;
Output: $\mathbf{X_{new}}$: Set of the synthetic Samples
begin SMOTE (T, N, K)
  for $i = 1 to T$ **do**
    Compute k nearest neighbours for i
    Save the indices in (nnarray)
    Populate (N, I, nnarrary)
  end
  Populate (N, I, nnarrary)
  while $N \neq 0$ **do**
    for feature to number of features **do**
      diff = X_{si} - X_i
      α = random number ϵ [0,1]
      $X_{new} = X_i + \alpha * \text{diff}$
    end
    N = N-1
  end
  return X_{new}
end

tune ML classifiers and the balancing technique; the selected hyperparameters of each model are discussed in detail in Table 2 and Table 3, respectively.

The HB was applied to optimize ML classifiers together with SMOTE for the proposed method. The optimization of SMOTE affords the best data balancing for the selected classifier. The hyperparameters of SMOTE are presented in Table 3. The hyperparameter $\alpha_{os} = N_{min}/N_{max}$. If α_{os} equals one yield to both classes, have equal samples. The HPO method's maximum number of iterations is set to thirty, and the experiments are executed ten times using different random seeds to avoid randomness impacts. Our experiment used a 10-fold cross-validation (CV) procedure to avoid overfitting [41].

B. SEVERITY LEVEL CLASSIFICATION

The second experiment is conducted to predict the severity of heart disease by maintaining the output class from 0 to 4, where 0 indicates an absence and 1 to 4 symbolizes the severity, where four is the highest. The observations are numbered 160, 54, 35, 35, and 13, labelled as absent, low, medium, high, and severe [21]. SMOTE was applied to increase the number of samples in the minority classes to deal with the imbalance problem. We used the sampling strategy in SMOTE to set the number of synthesized instances in each class. Since the data is highly imbalanced, equally balancing for all classes will cause overgeneralization, create noise between samples, and increase overlapping classes. Therefore, the selected HPO methods optimize the ML classifier and SMOTE, where SMOTE is optimized including the number of synthesized samples and k-neighbours, based on ML classifier performance until the model reaches the stopping criteria to increase the severity level of classification. The selected hyperparameters for SMOTE are different from those used in CVD detection. For example, the k_neigbours range is different since the k_neigbours cannot reach more the 9; else, the number of neighbours will be more than the samples in class 4. Also, the other hyperparameter selected is the sampling strategy formed in a dictionary to choose the number of synthesized instances in each class of the minority classes. The hyperparameters selected for optimization are mentioned in Table 2 and Table 3 for the classifiers and SMOTE, respectively.

TABLE 6. Classifiers' comparison before and after applying SMOTE on Cleveland and Statlog datasets.

Model	HB			PSO			GA			RS		
	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC
LR	92.22	89.27	0.841	92.59	89.82	0.845	90.74	86.72	0.81	90.12	86.22	0.794
SGD	92.96	90.56	0.854	91.85	89.15	0.828	91.35	88.19	0.822	87.03	82.69	0.731
ET	**98.52**	**98.09**	**0.969**	98.15	97.65	0.961	97.53	96.82	0.948	96.91	96.08	0.936
XGBoost	97.78	97.18	0.954	98.15	97.65	0.961	96.91	96.12	0.936	96.91	96.08	0.935
kNN	94.44	92.8	0.884	95.18	93.5	0.902	91.35	88.42	0.819	91.97	89.18	0.834
SVM	96.66	95.78	0.931	95.55	94.12	0.907	93.2	91.13	0.86	93.82	91.87	0.873

TABLE 7. Performance evaluation of proposed method compared with previous studies for CVD detection on Cleveland dataset.

Author	Method	Performance Evaluation					
		Accuracy	Precision	Recall	Specificity	F_1	MCC
[8]	Relief + LR	89		77	98		0.89
[9]	Vote + NB + LR	87.41	90.2			84.4	
[10]	HRFLM	88.4	90.1	92.8	82.9	90	
[12]	Stacked SVMs	92.22		82.92	**100**		0.851
[13]	FAMD + RF	93.44		89.28	96.96	92.59	0.87
[15]	Two-tier ensemble PSO based FS	85.71				86.49	
[16]	DBSCAN + SMOTE-ENN + XGBoost	98.4	98.57	98.33	98.33	98.32	0.97
[17]	SMOTE + Deep Learning	96	96.1	95.7		95.7	
[23]	ANN + RF	95.08	95	95		95	
Proposed Method	**HB + SMOTE + ET**	**99.2**	**98.7**	**99.33**	99.12	**99**	**0.983**

TABLE 8. Performance evaluation of proposed method compared with previous studies for CVD detection on Statlog dataset.

Author	Method	Performance Evaluation					
		Accuracy	Precision	Recall	Specificity	F_1	MCC
[5]	CFARS-AR	88.3		84.9			
[6]	RS-BPNN	90.4		94.67			
[7]	LR	85	85	89		87	
[9]	Vote + NB + LR	87.41					0.851
[15]	Two-tier ensemble PSO based FS	93.55				91.67	
[16]	DBSCAN + SMOTE-ENN + XGBoost	95.9	97.14	94.67	95.48	95.35	0.92
[14]	CF	89.8					
Proposed Method 2	**HB (SMOTE + ET)**	**98.52**	**98.13**	**98.09**	**98.72**	**98.09**	**0.969**

C. EVALUATION METRICS

We evaluated the proposed model using six performance metrics: accuracy, precision, recall, specificity, f-measure, and Mathew's Correlation Coefficient (MCC). In addition, we used weighted averaging for the second experiment, which computed the score to the average weighted by its size to provide equal weight to all levels. The six-performance metrics are calculated as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + FN + FP + TN} \tag{4}$$

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

$$\text{Specificity} = \frac{TN}{TN + FP} \tag{7}$$

$$f = \frac{2 * percision * recall}{percision + recall} \tag{8}$$

$$MCC = \frac{(TP * TN)-(FP * FN)}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}} \tag{9}$$

TABLE 9. ML classifiers comparison before and after applying SMOTE on multiclass.

Model	Before SMOTE		After SMOTE	
	Acc.	F_1	Acc.	F_1
LR	60	57.58	62.67	55.85
SGD	61.33	56.08	63.2	57.14
ET	70.13	67.37	**84.53**	**83.72**
XGBoost	75.73	73.44	84.26	83.11
kNN	65.33	65.31	68	64.36
SVM	69.33	65.29	84	81.66

IV. EXPERIMENTAL RESULTS

In this section, two experiments are carried out to verify and evaluate the performance of the proposed methods:

A. CVD ABSENCE AND PRESENCE PREDICTION

Classifiers trained were compared to see which performed better on both datasets. On evaluation metrics, some classifiers performed well, while others performed poorly. Table 4 shows the different classifiers' results before and after applying SMOTE using the default hyperparameters of each classifier as a baseline model. It is clear from Table 4

TABLE 10. Comparison using different HPO methods on severity level classification.

Model	HB			PSO			GA			RS		
	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC	Acc.	F_1	MCC
LR	70.39	66.56	0.497	71.73	68.88	0.526	66.4	62.01	0.424	65.6	61.1	0.408
SGD	71.73	68.09	0.524	72.26	68.99	0.535	64.8	61.54	0.409	64.53	60.27	0.395
ET	**95.73**	**95.78**	**0.934**	95.46	95.09	0.928	92.8	92.43	0.886	90.93	89.77	0.855
XgBoost	94.44	93.14	0.91	94.22	93.25	0.907	92.22	91.856	0.877	92.66	92	0.883
kNN	89.86	88.28	0.837	89.6	87.97	0.832	84	81.54	0.732	84.26	82	0.734
SVM	91.11	89.79	0.857	91.99	90.52	0.871	90.88	89.301	0.854	89.77	88.03	0.835

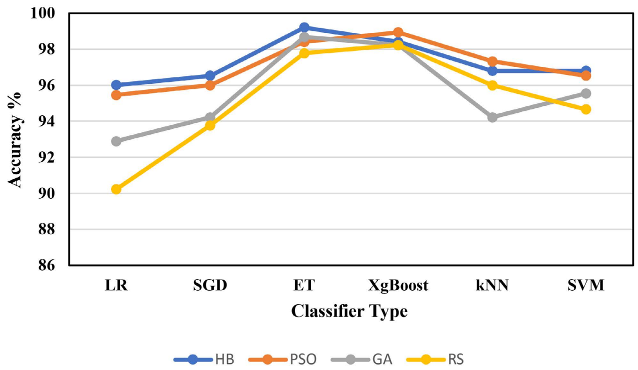

FIGURE 2. HPO comparison on SMOTE and ML algorithms using Cleveland dataset.

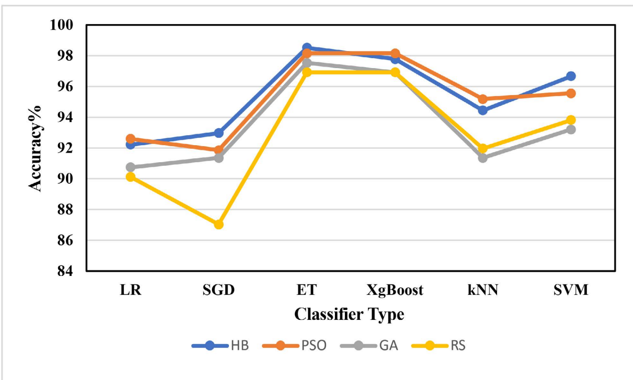

FIGURE 3. HPO comparison on SMOTE and ML algorithms using Statlog dataset.

that the model performance was significantly enhanced by applying SMOTE. However, boosting algorithm constructs trees by decreasing errors from previously created weak DT. Up-sampling comparable data does not affect enhancing findings. So, XGBoost didn't show any enhancement using SMOTE. XGBoost classifier achieved the highest accuracy by 94.67% for Cleveland, while XGBoost and ET classifiers achieved 91.97% for the Statlog dataset. In both datasets, the classifiers of the tree-based ensemble achieved better accuracies than the regression-based and statistical-based classifiers.

For HD prediction, we consider two datasets, so the optimal values of several ML hyperparameters vary, whereas HPO methods enhance the accuracy of ML models. The selected HB optimization method improved HD prediction results significantly, as shown in Table 5 and Table 6 for Cleveland and Statlog. The tree-based ensemble models most improved using HB since those models contain more hyperparameters to be optimized with a more extensive search space area, which significantly affects the classifier's performance, especially for ET. In contrast, the regression-based models didn't improve much since those models depend more on data distribution. ET classifier improved 5.42%, and 6.55% reached the highest accuracy after applying HB on both datasets, achieving 99.2% and 99% accuracy and f1 on the Cleveland dataset, 98.52%, 98.09% accuracy and f1 on the Statlog dataset.

Moreover, we compared the selected HB optimization method with commonly used HPO, including improved PSO, GA, and RS. To make a fair comparison, we used the same hyperparameter search space. The results are presented in Table 5 and Table 6 for both datasets. It is shown that RS can't guarantee to reach optimal near hyperparameters for all classifiers, where it didn't improve the results of the classifiers much, especially for LR and SGD. PSO and GA got better accuracies than RS for both datasets. PSO achieved good results, especially in the Statlog dataset. PSO achieved higher accuracies for kNN and XGBoost on the Cleveland dataset and kNN, LR, and XGBoost for Statlog; however, PSO didn't reach as high as HB as compared with the other classifiers. The accuracy comparison between HPO methods is presented in Figure 2 and Figure 3.

Finally, we compared the findings of our suggested methods to those of earlier research. We used the same datasets as earlier research, so we took the results from those studies without applying their methods. The comparison with the prior studies is presented in Table 7 and Table 8 for Cleveland and Statlog datasets. Our proposed model achieved better for the Cleveland dataset than the previous studies in terms of accuracy, f1, and MCC of 99.2%, 99%, and 0.983. In comparison, our method reached 98.52%, 98.09%, and 0.969 accuracy, f1, and MCC for the Statlog dataset improving the accuracy by 2.62%. Based on this result assessment and evaluation, the present study on the CVD detection prediction model delivers superior outcomes than the prior work in terms of accuracy, f1-score, and MCC.

B. CVD SEVERITY CLASSIFICATION

In this experiment, we predict the severity level of CVD. The results are presented in Table 9, showing the different

TABLE 11. Performance evaluation of proposed method compared with previous studies for HD severity level classification.

Author	Method	Performance Metrics				
		Accuracy	Precision	Recall	F_1	MCC
[19]	GA + Modified KNN	62.1	50.7	41.8	43.6	0.497
[20]	BT + SVM	61.86				
[21]	Re-sampling + KNN	79.2				
[42]	ATOVIC	81.927				
[22]	CDTL-RF	89.3				
[23]	LR + RF	75.41	72	75	71	
Proposed Method 2	**HB + SMOTE + ET**	**95.73**	**96.35**	**95.73**	**95.78**	**0.934**

classifiers' results before and after applying SMOTE using the default hyperparameters of each model. The model performance was significantly enhanced by applying SMOTE. The tree-based ensemble classifiers achieved better results before applying SMOTE. Also, SVM improved considerably after using SMOTE to reach an accuracy of 84%.

Furthermore, the classifiers performed way better after applying HB, reaching an accuracy of 95.73% and 95.46%, with an improvement of around 10% for ET and XGBoost. Also, kNN and SVM show good improvement, reaching 89.86% and 91.11%, respectively. On the other hand, the HB method didn't enhance the performance of SGD and LR much; the outcomes are presented in Table 10. Afterwards, the selected HPO method HB was compared with different techniques. It was noticed that all HPO techniques improved the accuracy of the prediction model; the results are presented in Table 10. Still, RS and GA performed well in enhancing the performance but didn't achieve more than 93% in severity classification. Besides, PSO obtained good results for classifiers reaching 91.99% accuracy for SVM, even better than HB for SVM, which got 91.11%. Also, PSO achieved better results in optimizing LR and SGD. However, HB performed better than other optimizers and achieved better outcomes for the ET classifier, showing its robustness. HB achieves better severity-level classification results, especially for the models with a more extensive search space. The tree-based ensemble models achieved higher results than the other types of models.

Finally, we compared the results of our proposed method with prior studies; the comparison is presented in Table 11. The results show that the proposed methods outperformed all previous work on the severity level prediction using Cleveland dataset by achieving accuracy, precision, recall, f-measure, MCC, 95.73%, 96.35%, 95.73%, 95.78%, 0.934, respectively. The proposed method attained the highest accuracy, averaging a 13.803% improvement over previous studies' results. The authors in [19] and [20] didn't achieve a high accuracy since they developed their system without balancing the data. Unlike the other studies [21]–[23], did data re-sampling and balancing to reach better results using an untuned method to predict the severity level. Machine learning hyperparameters have various optimums to reach the ultimate performance on different tasks and datasets. Hence, HPO improves the ML performance by detecting the optimal

TABLE 12. The table of abbreviation.

Abbreviations	
Acronyms	
CVD	Cardiovascular disease
ML	Machine Learning
SMOTE	Synthetic Minority Oversampling Technique
HPO	Hyperparameter Optimization
ET	Extra Trees
WHO	World Health Organization
HD	heart disease
FS	feature selection
RS	Rough Sets
CFA	Choas firefly algorithm
BPNN	backpropagation neural network
ANN	Artificial neural network
LR	Logistic Regression
NB	Naïve Bayes
DT	Decision Trees
SVM	Support Vector Machine
kNN	k-Nearest Neighbor
cforest	conditional inference tree forest
PSO	particle swarm optimization
DBSCAN	density-based spatial clustering of applications with noise
ET	extremely randomized trees
GA	genetic algorithm
BT	binary tree
CDTL	cluster-based DT learning
SGD	Stochastic Gradient Descent
UCI	University of California Irvine
CV	cross-validation
SVC	Support Vector Classifier
RBF	radial basis function
HB	Hyperband
SH	successive halving
MCC	Mathew's Correlation Coefficient
Nomenclature	
n	The number of the points
C	The regularization coefficient
ω	normalization vector
L1	absolute norm
L2	squared euclidean norm
G & H	the sums of the cost function's first and second-order gradient statistics
t	the number of leaves in DT
λ and γ.	Two penalty coefficients
B	number of budgets
s	budget size
n	number of configurations
b	budget allocations
L	loss function
Acc.	Accuracy

hyperparameter configurations for balancing techniques and classifiers.

V. CONCLUSION AND FUTURE WORK

This paper proposes an accurate and efficient decision support system based on the ML model to predict the absence or presence of CVD and classify the CVD's severity level. The proposed model integrates SMOTE, ET, and HB. Those three methods are used for data balancing, classification, and hyperparameter optimization. The proposed model is evaluated and benchmarked against previous studies. We have presented a statistical evaluation of our model's performance using six performance metrics. The experimental results show that tree-based models were more effective in achieving higher quality results, and the HB optimization method has more impact on improving the models' accuracy. Thus, SMOTE and ET optimized by HB achieved the highest accuracy for binary and multiclass problems on both datasets. The experimental findings revealed that our model outperforms the state-of-the-art models concerning the accuracy, recall, f1-score, and MCC of up to 99.2%, 99.33%, 99%, and 0.983 for the Cleveland dataset 98.52%, 98.08%, 98.08, and 0.969 for Statlog dataset respectively. Our model achieved 95.73%, 96.35%, 95.73%, 95.78%, 0.939 in terms of accuracy, precision, recall, f1-score, and MCC, respectively, for multiclass (severity level) prediction. Healthcare specialists may utilize this research to predict heart failure and enhance patient treatment. We aim to create a general framework based on machine learning, including outlier detection and removal, and feature selection, to improve the detection and severity level classification of different diseases using real-time clinical data for our future work.

APPENDIX

See Table 12.

REFERENCES

[1] R. T. Selvi and I. Muthulakshmi, "An optimal artificial neural network based big data application for heart disease diagnosis and classification model," *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 6, pp. 6129–6139, 2021.

[2] G. Bazoukis, S. Stavrakis, J. Zhou, S. C. Bollepalli, G. Tse, Q. Zhang, J. P. Singh, and A. A. Armoundas, "Machine learning versus conventional clinical methods in guiding management of heart failure patients—A systematic review," *Heart Failure Rev.*, vol. 26, no. 1, pp. 23–34, Jan. 2021.

[3] A. Makhlouf, I. Boudouane, N. Saadia, and A. R. Cherif, "Ambient assistance service for fall and heart problem detection," *J. Ambient Intell. Humanized Comput.*, vol. 10, no. 4, pp. 1527–1546, 2018.

[4] M. Chen, S. Gonzalez, V. Leung, Q. Zhang, and M. Li, "A 2G-RFID-based e-healthcare system," *IEEE Wireless Commun. Mag.*, vol. 17, no. 1, pp. 37–43, Feb. 2010.

[5] N. C. Long, P. Meesad, and H. Unger, "A highly accurate firefly based algorithm for heart disease prediction," *Expert Syst. Appl.*, vol. 42, no. 21, pp. 8221–8231, 2015.

[6] K. B. Nahato, K. N. Harichandran, and K. Arputharaj, "Knowledge mining from clinical datasets using rough sets and backpropagation neural network," *Comput. Math. Methods Med.*, vol. 2015, pp. 1–13, Mar. 2015.

[7] A. K. Dwivedi, "Performance evaluation of different machine learning techniques for prediction of heart disease," *Neural Comput. Appl.*, vol. 29, no. 10, pp. 685–693, 2018.

[8] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, "A hybrid intelligent system framework for the prediction of heart disease using machine learning algorithms," *Mobile Inf. Syst.*, vol. 2018, pp. 1–21, Dec. 2018.

[9] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics Inform.*, vol. 36, pp. 82–93, Mar. 2019.

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

[11] J. Vijayashree and H. P. Sultana, "A machine learning framework for feature selection in heart disease classification using improved particle swarm optimization with support vector machine classifier," *Program. Comput. Softw.*, vol. 44, no. 6, pp. 388–397, Nov. 2018.

[12] L. Ali, A. Niamat, J. A. Khan, N. A. Golilarz, X. Xingzhong, A. Noor, R. Nour, and S. A. C. Bukhari, "An optimized stacked support vector machines based expert system for the effective prediction of heart failure," *IEEE Access*, vol. 7, pp. 54007–54014, 2019.

[13] A. Gupta, R. Kumar, H. S. Arora, and B. Raman, "MIFH: A machine intelligence framework for heart disease diagnosis," *IEEE Access*, vol. 8, pp. 14659–14674, 2020.

[14] B. A. Tama and S. Lim, "A comparative performance evaluation of classification algorithms for clinical decision support systems," *Mathematics*, vol. 8, no. 10, p. 1814, Oct. 2020.

[15] B. A. Tama, S. Im, and S. Lee, "Improving an intelligent detection system for coronary heart disease using a two-tier classifier ensemble," *BioMed Res. Int.*, vol. 2020, pp. 1–10, Apr. 2020.

[16] N. L. Fitriyani, M. Syafrudin, G. Alfian, and J. Rhee, "HDPM: An effective heart disease prediction model for a clinical decision support system," *IEEE Access*, vol. 8, pp. 133034–133050, 2020.

[17] M. Waqar, H. Dawood, H. Dawood, N. Majeed, A. Banjar, and R. Alharbey, "An efficient SMOTE-based deep learning model for heart attack prediction," *Sci. Program.*, vol. 2021, pp. 1–12, Mar. 2021.

[18] A. Ishaq, S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara, and M. Nappi, "Improving the prediction of heart failure patients' survival using SMOTE and effective data mining techniques," *IEEE Access*, vol. 9, pp. 39707–39716, 2021.

[19] N. Salari, S. Shohaimi, F. Najafi, M. Nallappan, and I. Karishnarajah, "A novel hybrid classification model of genetic algorithms, modified k-nearest neighbor and developed backpropagation neural network," *PLoS ONE*, vol. 9, no. 11, Nov. 2014, Art. no. e112987.

[20] W. Wiharto, H. Kusnanto, and H. Herianto, "Performance analysis of multiclass support vector machine classification for diagnosis of coronary heart diseases," 2015, *arXiv:1511.02352*.

[21] N. Khateeb and M. Usman, "Efficient heart disease prediction system using K-nearest neighbor classification technique," in *Proc. Int. Conf. Big Data Internet Thing (BDIOT)*, 2017, pp. 21–26.

[22] G. Magesh and P. Swarnalatha, "Optimal feature selection through a cluster-based DT learning (CDTL) in heart disease prediction," *Evol. Intell.*, vol. 14, no. 2, pp. 583–593, Jun. 2021.

[23] H. B. Kibria and A. Matin, "The severity prediction of the binary and multi-class cardiovascular disease—A machine learning-based fusion approach," *Comput. Biol. Chem.*, vol. 98, Jun. 2022, Art. no. 107672.

[24] S. Shin, B. Ko, and H. So, "Noncontact thermal mapping method based on local temperature data using deep neural network regression," *Int. J. Heat Mass Transf.*, vol. 183, Feb. 2022, Art. no. 122236.

[25] P. Lacerda, B. Barros, C. Albuquerque, and A. Conci, "Hyperparameter optimization for COVID-19 pneumonia diagnosis based on chest CT," *Sensors*, vol. 21, no. 6, p. 2174, Mar. 2021.

[26] Q. Yang, Y.-W. Bian, X.-D. Gao, D.-D. Xu, Z.-Y. Lu, S.-W. Jeon, and J. Zhang, "Stochastic triad topology based particle swarm optimization for global numerical optimization," *Mathematics*, vol. 10, no. 7, p. 1032, Mar. 2022.

[27] M.-F. Leung, C. A. C. Coello, C.-C. Cheung, S.-C. Ng, and A. K.-F. Lui, "A hybrid leader selection strategy for many-objective particle swarm optimization," *IEEE Access*, vol. 8, pp. 189527–189545, 2020.

[28] L. Yang and A. Shami, "On hyperparameter optimization of machine learning algorithms: Theory and practice," *Neurocomputing*, vol. 415, pp. 295–316, Nov. 2020.

[29] R. Detrano. *Heart Disease*. UCI Machine Learning Repository. Accessed: Oct. 21, 2021. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/heart+disease

[30] C. G. D. Dua. *(UCI) Machine Learning Repository*. Accessed: Oct. 21, 2021. [Online]. Available: http://archive.ics.uci.edu/ml

[31] M.-W. Huang, C.-W. Chen, W.-C. Lin, S.-W. Ke, and C.-F. Tsai, ''SVM and SVM ensembles in breast cancer prediction,'' *PLoS ONE*, vol. 12, no. 1, Jan. 2017, Art. no. e0161501.

[32] C. Sowmiya and P. Sumitra, ''A hybrid approach for mortality prediction for heart patients using ACO-HKNN,'' *J. Ambient Intell. Humanized Comput.*, vol. 12, no. 5, pp. 5405–5412, 2021.

[33] M.-A. Zöller and M. F. Huber, ''Benchmark and survey of automated machine learning frameworks,'' *J. Artif. Intell. Res.*, vol. 70, pp. 409–472, Jan. 2021.

[34] W. A. Gardner, ''Learning characteristics of stochastic-gradient-descent algorithms: A general study, analysis, and critique,'' *Signal Process.*, vol. 6, no. 2, pp. 113–133, Apr. 1984.

[35] P. Geurts, D. Ernst, and L. Wehenkel, ''Extremely randomized trees,'' *Mach. Learn.*, vol. 63, no. 1, pp. 3–42, 2006.

[36] T. Chen and C. Guestrin, ''XGBoost: A scalable tree boosting system,'' in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2016, pp. 785–794.

[37] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, ''Hyperband: A novel bandit-based approach to hyperparameter optimization,'' *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 6765–6816, 2017.

[38] M. Claesen and B. D. Moor, ''Hyperparameter search in machine learning,'' 2015, *arXiv:1502.02127*.

[39] A. Fernández, S. Garcia, F. Herrera, and N. V. Chawla, ''SMOTE for learning from imbalanced data: Progress and challenges, marking the 15-year anniversary,'' *J. Artif. Intell. Res.*, vol. 61, pp. 863–905, Apr. 2018.

[40] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, ''Smote: Synthetic minority over-sampling technique,'' *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, Dec. 2002.

[41] R. Kohavi, ''A study of cross-validation and bootstrap for accuracy estimation and model selection,'' in *Proc. IJCAI*, vol. 14, no. 2. Montreal, QC, Canada, 1995, pp. 1137–1145.

[42] L. Baccour, ''Amended fused TOPSIS-VIKOR for classification (ATOVIC) applied to some UCI data sets,'' *Expert Syst. Appl.*, vol. 99, pp. 115–125, Jun. 2018.

ABDALLAH ABDELLATIF (Member, IEEE) received the M.Sc. degree in control systems from Universiti Tun Hussein Onn Malaysia, in 2019. He is currently pursuing the Ph.D. degree in control systems with the Department of Electrical Engineering, Universiti Malaya. His research interests include data mining, mainly working machine learning and deep learning.

HAMDAN ABDELLATEF (Member, IEEE) received the B.E. and M.Sc. degrees in communication and electronics engineering from Beirut Arab University, Lebanon, in 2012 and 2016, respectively, and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia, Malaysia, in 2020. He is currently a Research Faculty and a Postdoctoral Research Fellow at Lebanese American University, Byblos, Lebanon. His current research interests include deep learning, hardware/software co-design, signal processing, and stochastic computing.

JEEVAN KANESAN received the degree in electrical engineering from UTM, in 1999, and the M.Sc. and Ph.D. degrees from USM, Penang, in 2002 and 2006, respectively. He joined the Department of Electrical Engineering, Universiti Malaya, in 2008. He worked as an Engineer at Carsem Semiconductor, Ipoh, before resuming his master's degree. After obtaining his Ph.D. degree, he worked as a Research and Development Engineer at Intel, Penang, for two and half years. Currently, he has published more than 70 peer-reviewed journals. His research interests include optimization and machine learning.

CHEE-ONN CHOW (Senior Member, IEEE) received the Bachelor of Engineering (Hons.) and Master of Engineering Science degrees from the University of Malaya, Malaysia, in 1999 and 2001, respectively, and the Doctor of Engineering degree from Tokai University, Japan, in 2008. He joined the Department of Electrical Engineering, University of Malaya, in 1999, and currently an Associate Professor with the Department of Electrical Engineering. His research interests include communication networks, multimedia applications, data analytics, and artificial intelligence. He is a Registered Professional Engineer with the Board of Engineers Malaysia.

JOON HUANG CHUAH (Senior Member, IEEE) received the B.Eng. degree (Hons.) from Universiti Teknologi Malaysia, the M.Eng. degree from the National University of Singapore, and the M.Phil. and Ph.D. degrees from the University of Cambridge. He is currently the Head of the VIP Research Group and an Associate Professor at the Department of Electrical Engineering, Faculty of Engineering, University of Malaya. His research interests include image processing, computational intelligence, IC design, and scanning electron microscopy. He is also a Registered Chartered Engineer under the Engineering Council, U.K., and also a Registered Professional Engineer under the Board of Engineers, Malaysia. He was the Honorary Treasurer of IEEE Computational Intelligence Society (CIS) Malaysia Chapter and the Honorary Secretary of IEEE Council on RFID Malaysia Chapter. He is the Vice Chairperson of the Institution of Engineering and Technology (IET) Malaysia Network. He is also a fellow and the Honorary Secretary of the Institution of Engineers, Malaysia (IEM).

HASSAN MUWAFAQ GHENI received the bachelor's (B.Sc.) degree in electrical and electronic engineering from the Department of Electrical Engineering, University of Babylon, Hilla, Iraq, in June 2016, and the master's degree from the Faculty of Electrical and Electronics Engineering, Universiti Tun Hussein Malaysia, in February 2018. He is currently a Lecturer at the Department of Computer Techniques Engineering/ Al-Mustaqbal University College. His research interests include optical communication, the IoT, wireless sensor networks, communications, V2V systems, and artificial intelligent.