CHAPTER 6

SUPER LEARNER ENSEMBLE MODEL IN HEART DISEASE PREDICTION

In the analysis of statistical data, an excessive degree of human bias is introduced when inappropriate models are chosen. The choice of a statistical model is the foundation of all estimating and hypothesis testing techniques, therefore achieving reliable estimates and making sound statistical inferences depends heavily on the model's correct depiction of the process that produced the data. This chapter briefly discusses on utilizing different ML base classifiers and devising the proposed SLEM. Comparing the Super Learner Model to state-of-the-art ML techniques is also demonstrated.

6.1. INTRODUCTION

Cardio Vascular Diseases (CVDs) are receiving a growing degree of attention due to increasing incidences and fatalities worldwide. There are many risk factors which on exceeding their normal limits would lead to heart disease at a later stage. Diagnosing heart disease at an early phase is also a crucial task. In the modern medical system, the role of ML is crucial, particularly in the early prediction of cardiovascular illnesses. Even though single classifiers may perform well on few datasets, it may not work very well for all new samples. This disadvantage of individual classifier can be overcome in ensemble models. The ensemble of classifiers provides robust solutions than individual classifiers.

There are three ways by which ensemble models can differ:

i) data selection for ensemble base model training

ii) by deciding on which foundational models to use for the ensemble

iii) by the method used to synthesize the ensemble model's results from those of its constituent base models.

By carefully selecting appropriate base models, this study proposes a unique Super-Learner Ensemble Model (SLEM) for categorizing data on heart disease. Early heart disease detection with the SLEM is achieved through the merging of many individual base models and tree based models.

6.2. MACHINE LEARNING MODELS

6.2.1 Support Vector Machine (SVM)

Developed by Vladimir Vapnik, SVM , is widely used for both linear and non- linear classification in the machine learning community. The SVM technique uses n- dimensional vectors (features) to represent each data point (Uddin et al., 2019). The SVM kernel converts a non-separable issue into a separable one by mapping the input space from a lower dimension to a higher one. Once the hyper-plane separating the two groups has been located, classification can be finalized. However, this model takes long time to train the larger datasets.

6.2.2 Logistic Regression (LR)

Probability theory forms the basis of LR, a method for predictive analysis. It makes use of the function sigmoid, a logistic function, which is effective in the classification of binary or multiple classes because it accepts any real input and returns a value between zero and one. In predicting the likelihood of an event, LR takes into account a wide range of variables. Where x is the predictor variable and $f(x)$ is the predicted variable, we get the logistic function, denoted by Eq. (6.1).

$$f(x) = 1/(1 + e^{-x}) \tag{6.1}$$

LR should be avoided since it can lead to overfitting if there are fewer observations than features.

6.2.3 Gaussian Naive Bayes (GNB)

Using a probabilistic strategy and the Gaussian distribution, GNB is a classification method. When predicting a class/outcome/target variable, GNB considers that all parameters (also known as features or predictors) contribute equally and independently. GNB is based on the 'naive' assumption that features of the dataset are uncorrelated with one another. Given 'n' independent features ($X_1, X_2, X_3, \ldots X_n$) in a dataset, the probability of a data point belonging to class 'y' given its features ($X_1, X_2, X_3, \ldots X_n$) is:

$$P(Y = y|\{X_1, X_2, X_3, \ldots X_n\}) = \frac{P(X_1/Y).\ P(X_2/Y)\ldots\ P(X_n/Y)\ .P(Y)}{P(X_1).P(X_2)\ldots P(X_n)} \tag{6.2}$$

However, this approach requires more extensive data sets in order to yield useful results.

6.2.4 K-Nearest Neighbors (KNN)

The KNN technique is widely used in ML for both classification and regression purposes. A new data point is assigned a label based on its closeness to labeled training set points. This is accomplished by calculating the average distance of each newly contributed data point to the entire training dataset. Euclidean distance is commonly used for continuous features, while other distance metrics may be used for categorical data. Whena new data point is introduced, the K training examples closest in distance are chosen. Thenew data points nearest neighbors are these examples. The majority class of a point's K neighbors is used to assign a class label to the point. The classification performance of KNN can vary widely depending on the values used for K and the distance metric.

6.2.5 Decision Tree (DT)

Attribute-based data partitioning is the basis for the DT method, which uses decision rules at internal nodes to choose which attribute to evaluate next. After a particular depth is reached or when all instances in a partition belong to the same class, thesplitting process stops. A leaf node is then generated and given the class. To determine thequality of a split, entropy is used to quantify the uncertainty or randomness in the class labels of the instances within a particular node. Eq. (6.3) shows the formula for entropy computation.

$$Entropy\ H(S) = -\sum_{i=1}^{n} p_i\ (S) * log_2\ p_i(S)) \quad (6.3)$$

If S is the collection of cases, $pi(S)$ is the chance that one of those examples falls into class i, and a total across all possible groups is computed.

After the division, the IG is determined by contrasting the parent node's entropy with the weighted average of the entropies of the child nodes. The criterion for the divide is determined by the attribute that provides the most useful information. Eq. (6.4) shows the formula for IG computation.

$$Information\ Gain(S,A) = H(S) - \sum_{v \in values(A)} . \frac{|S_v|}{S} H(S_v) \quad (6.4)$$

DT has high interpretability but prone to over-fitting. A limitation of DT model is its instability. What this means is that the model's resilience can be impacted by even modest changes in the training data, which might lead to different DT's. Ensemble methods can help address this issue by combining multiple DT's to make more stable predictions.

6.2.6 Ensemble Classifiers

Ensemble classifiers are either a combination of homogenous classifiers or heterogeneous classifiers. Among the many benefits they offer is enhanced prediction accuracy, resistance to overfitting, and the ability to identify intricate relationships within the data. On the other hand, they could be more difficult to interpret and computationally intensive than separate models. The challenge at hand, the data, and the needs of the task all factor into which ensemble classifier is selected. Ensemble classifiers outperformindividual models by pooling their best features.

6.2.6.1 Random Forest (RF)

The RF classifier is trained using a group of DTs, each of which is fed a distinct subset of the full training set. Each node of each DT additionally undergoes a split-check using a randomly selected set of characteristics. These randomizations improve generalization by reducing overfitting since they increase variation among the trees. RFs are less interpretable than DT's due to their ensemble nature. For huge datasets or difficult tasks, the computational cost of building many DT's can be prohibitive. Training times, however, can be cut in half by performing the steps in parallel.

6.2.6.2 Extreme Gradient boosting (XGB)

By merging the results of several different DT's, XGBoost (eXtreme Gradient Boosting) generates an accurate predictive model. The technique works by training successive DT's to reduce the error made by the preceding trees. Overfitting is avoided by pruning newly-added DT's in the method, which is used to maximize an objective function. XGBoost's capacity to deal with intricate connections in the data is one of its greatest features. In XGBoost, the objective function is a mixture of the Training Loss andthe Regularization, where the Training Loss is the differentiable convex loss function that determines the difference between the target y_i and the prediction $\hat{y}$. Eq. (6.5) is the XGBoost objective function.

$$Obj\ \vartheta = \sum_{i}^{n} l(y_i - \hat{y}_i) + \sum_{i=1}^{j} \Omega\,(f_i) \tag{6.5}$$

The model's complexity is punished by the second term Ω. The regularization term is then applied to the learned weights as a whole to prevent over-fitting (Tianqi Chen & Carlos Guestrin, 2016). For large datasets in particular, XGBoost's memory requirements

can be prohibitive. Because of this, it may be difficult to run on systems with minimal memory, resulting in decreased performance.

6.2.6.3 CatBoost (CatB)

Categorical Boosting (CatBoost) is intended to work on categorical dataimpeccably. The distinct features of CatB compared to other boosting algorithms are:

- High performance even without hyper parameter tuning
- Categorical features support
- Improved accuracy by reducing overfitting
- Fast prediction
- CPU as well as fast and scalable GPU version

CatB uses a random permutation-driven ordered boosting technique. In ordered boosting, a new training dataset is obtained in each step of boosting. This assures that the previously obtained model has not seen the labels in the new training set. This prevents biasing of the trained model at each step of boosting. Its claim to fame is that it can deal with categorical variables without resorting to label encoding or OHE. CatB is distinct in its usage of symmetric trees, for example. In other words, the split condition is the same across all the decision nodes at every depth.

6.2.6.4 Majority Voting Ensemble (MVE):

The predictions of several different ML models are pooled together to form an MVE. It's a method for boosting model efficiency, with the goal of outperforming even the best individual model. Multi-model prediction is the key to an MVE's success. The majority label is anticipated by adding up the predictions for all of the labels in a classification.

6.2.6.5 Stacking:

Stacking is a method of ensemble learning that makes predictions by combining numerous individual models in a hierarchical fashion. To stack multiple models, one trains a meta-model that learns to make predictions based on a combination of the predictions from the individual models used as input.

6.2.7 Advantages of opting Ensemble classifier over Individual classifier

i) Performance:

The effectiveness of any one ML classifier may be capped by the problem's complexity and the model's inability to capture all the underlying patterns. When compared to individual classifiers, ensemble classifiers may get better results. By combining multiple models, ensemble methods can reduce bias, variance and overfitting, leading to better generalization and higher prediction accuracy.

ii) Robustness:

An individual ML classifier is vulnerable to noise, outliers, and dataset biases. Poor generalization may result from inaccurate predictions or sensitivity to outliers in the training data. Ensemble classifiers are generally more robust due to their ability to averageor vote on multiple predictions. They can mitigate the impact of individual model errors oroutliers, making them more resilient to noise and improving overall stability.

iii) Interpretability:

It is typically easier to understand and explain the behavior of a single ML classifier, leading to improved interpretability. Ensemble classifiers tend to be less interpretable. The combination of multiple models makes it harder to extract specific insights or understand the decision-making process.

iv) Training and Inference Time:

Training a single model is faster compared to training an ensemble. However, during inference, ensemble classifiers can often make predictions faster by leveraging parallelization and averaging/voting the results.

v) Handling Complex Relationships:

An individual ML classifier may struggle to capture complex relationships or interactions between features, especially if the problem is highly nonlinear or involves high-dimensional data. Ensemble classifiers, especially gradient boosting and neural network ensembles, have the capability to capture complex relationships and interactions between features. They excel in handling challenging and intricate patterns within the data.

Potential benefits of using ensemble classifiers include enhanced performance, increased robustness, and the ability to deal with complex interactions. However, ensemble classifiers may sacrifice interpretability and require more computational resources for training. The choice between a single ML classifier and an ensemble classifier depends on the specific problem, dataset characteristics, interpretability requirements, and the trade-offs between performance and complexity.

6.3 PROPOSED METHODOLOGY

The most crucial characteristics of each heart disease dataset are isolated using the 'ModifiedBoostARoota' (MBAR) technique. SVM, LR, DT Classifier, Gaussian Nave Bayes (GNB), RF, XGBoost (XGB), KNN, CatBoost (CatB), and MVE are just some of the models that make up the Super Learner Ensemble Model (SLEM) for classification (Van der Laan et al., 2007). LR is the chosen meta-learner.

The base models chosen in the proposed SLEM model is a combination of linear, probabilistic, bagging, boosting and stacking models. The ensemble classifier is most effective when its base models are significantly different. These classifiers are first trained using three iterations of a k=10 stratified k-fold cross-validation. Each classifier's performance is evaluated by calculating their average accuracy over all folds and repeats. Then Super Learner, devised with a combination of all these classifiers and LR as meta learner, is trained and tested on the Heart Datasets. The backward elimination method was applied to the ensemble model's base classifiers to determine the optimal mix of classifiers to optimize the SLEM's performance. When compared to existing classifier combinations, the SLEM's innovative mix of DT and CatB classifiers provides superior performance. The SLEM's performance is compared with individual classifiers' performances, on both low and high dimensional datasets.

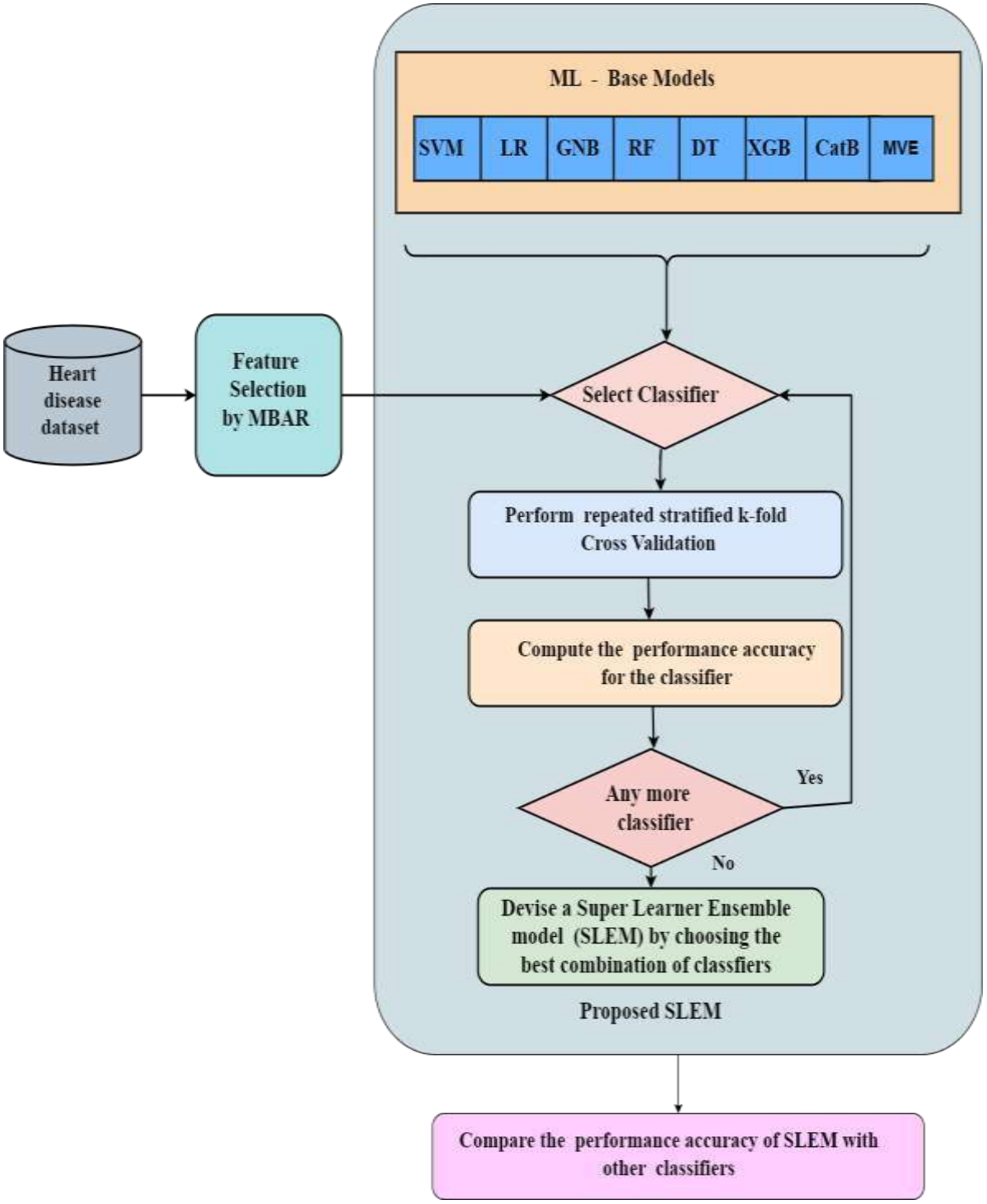

Figure 6.1 Proposed methodology

6.3.1 Proposed Super Learner for Classification of Heart Disease

The Figure 6.2 depicts the block diagram of the Super Learner Ensemble Model.

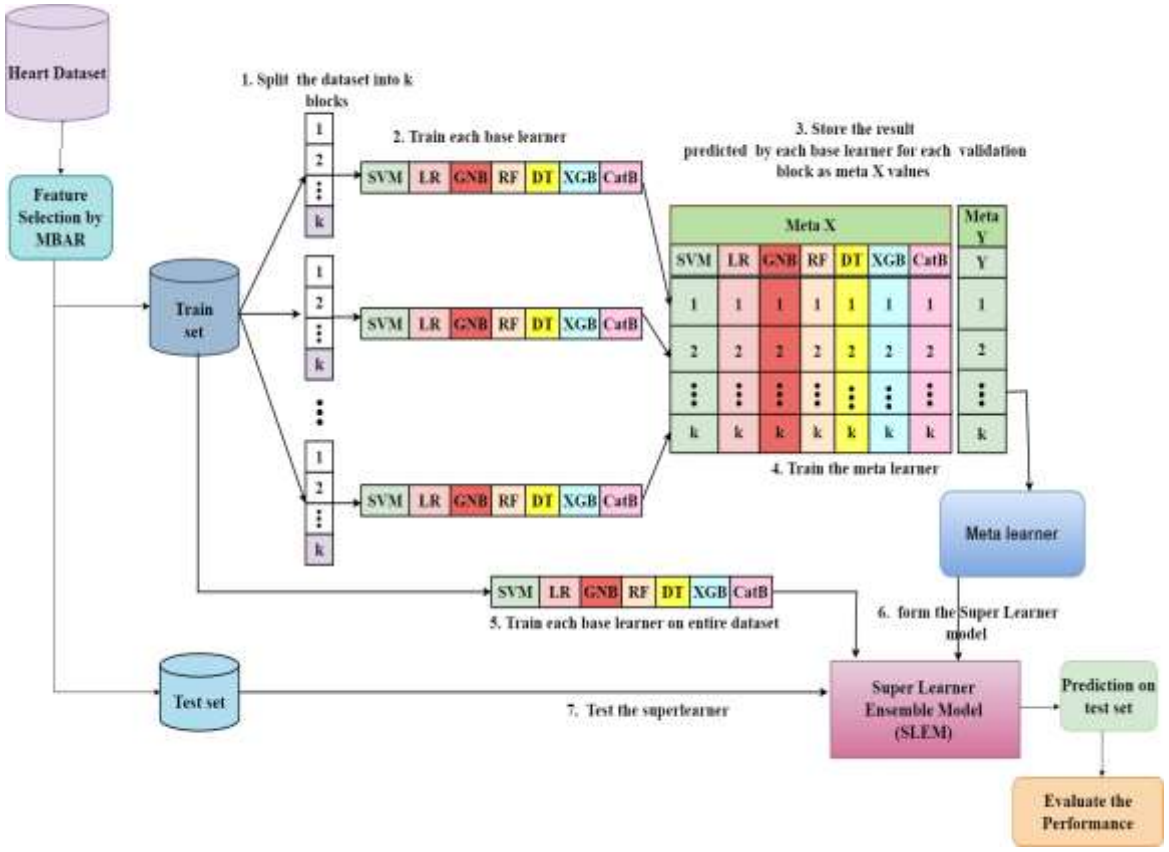

Figure 6.2 Block Diagram for the proposed SLEM

As depicted in Figure 6.2, the SLEM's working is briefly illustrated in following steps:

Step 1: Relevant characteristics are chosen from the dataset (depicted in section 3.4) and preprocessed using MBAR. Next, they partition this dataset with the chosen features into *k* blocks, each of which has *k* subparts. While the remaining blocks are used to teach the fundamental learner, one of the *k* pieces is set aside for checking. The algorithms used to build the super learner are the ones that were chosen as the base learners, also known as the candidate learners *S*. The ML base learners utilized in this framework are briefly described in section 6.2 and include SVM, LR, GNB, RF, DT, XGB, CatB, and *MVE*.

Step 2: Until all k of the blocks' sections have been swapped as the validation set, the candidate learners are trained and tested.

Step 3: As the reference to Figure 6.2, a matrix is constructed with all the outputs of the candidate learners in the validation block. The meta-learner takes the matrix as its input. Predictions are saved alongside determined sample outcomes y.

Step 4: The resulting super-learning model is built by integrating the predictions of each candidate with those of the meta-learning model, which is trained using the matrix.

Step 5: In order to be employed on previously unseen samples, the candidate learners are fully trained with the entire dataset.

Step 6: The SLEM is evaluated by first applying the base models to the novel test data and then feeding the resulting classifier outputs into the SLEM in the form of a vector. The ensemble learner predicts the output for the given vector.

Step 7: SLEM's effectiveness can be enhanced by using the right classifiers.

Algorithm: Super Learner Ensemble Model

1: Input: Dataset (d) on coronary heart disease, containing a sample set (x) and a response column (y)

2: Output: Models of Prediction in an Ensemble

3: Set up the ensemble

- Indicate a set of model parameters (S in this case) and provide a list of S potential learners.
- Indicate a result you expect the S algorithm to produce.

4: Train the ensemble.

- Train each of the S on the training set represented by $\{x_1, x_2, x_3, \ldots, x_z\}$, where z is the total number of samples.
- Apply k-fold cross-validation (CV), where v is the total number of data points, to each of the S to make predictions for $\{y_1, y_2, y_3, \ldots, y_V\}$.

- The k -fold CV predicted values from each of the S base learners serve as X, the meta-learner's training input.
- Integrate the CV predictions from S to form a new meta matrix, R $(K\ X\ S)$. Train the meta learner model on the meta matrix
  $\{(x_1, x_2, x_3, \ldots., x_s, y_1), (x_1, x_2, x_3, \ldots., x_s, y_2), \ldots, ((x_1, x_2, x_3, \ldots., x_s, y_k))\}$
- To improve test-set prediction accuracy, merge the basic learners with the meta model.

5. Predict new data.

- Ensemble prediction requires first capturing the decision made by the base learners' output.
- To create the ensemble predictions, include these forecasts into the super learner model.

6.4 RESULTS AND DISCUSSION

Using Ubuntu OS and Python in Jupyter Notebook, the experiment was done on a system with an i5 processor and 4 GB of RAM. Cleveland HD, Statlog HD, and South Africa (SA) HD, Cardiovascular Disease Dataset (Mendeley Data) and a novel real world dataset comprising of cardiac biomarkers named as 'Cardio Biomarkers dataset' (described in section 3.2) are the low_dimensional datasets used. High_dimensional datasets are employed in this study, specifically the Arrhythmia heart dataset and the Z- Alizadeh Sani dataset on heart illness. Brief descriptions of each dataset are provided in subsection 3.4. In addition, section 3.5 details the assessment measures employed for scoring the efficiency of the suggested algorithms.

6.4.1 Performance Evaluation on Low and High Dimensional Heart Datasets

SMOTE is first applied to all datasets in order to balance them by over-sampling the minority class. After filling in missing data in each dataset, continuous attributes are normalized with Min-Max scaling. Eq.5.1 provides the formula for Min-Max normalization.

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}} \tag{5.1}$$

The base classifiers used in this SLEM model are listed in section 5.2. Repeated Stratified K-Fold cross validation was used to train and test the classifiers. The Stratified K-Fold cross validation is performed three times, each time with a different randomization. Maintaining a constant fraction of samples from each target class throughout iterations. The Stratified K-Fold used here creates 10 separate groups from the full dataset. A cross-validation cycle uses 9 folds as the training set and 1 fold as the validation set. Iterations are set at 10, the same as the number of folds. Each classifier's performance is evaluated by calculating their average accuracy over all folds and repeats. Training and evaluation are repeatedly performed using the Low-dimensional Heart Datasets. Table 5.1 displays the outcomes of feature-less, repeated stratified K-fold cross validation on the selected datasets.

On the heart datasets mentioned in Section 3.2, we model classifiers such as the SVM, LR Classifier, DT Classifier, GNB, RF, XGBoost (XGB), KNN, CatB, and MVE.

6.4.1.1 Evaluation of Classification Accuracy on Low-Dimensional Heart Datasets with All Features

All the features in the low dimensional datasets are taken into consideration and the accuracies of all the classifiers mentioned in section 5.4.1 are obtained. A comparison of the accuracies of the proposed SLEM model and all the other classifiers is provided in Table 6.1. Proposed SLEM gives high prediction accuracy compared to other classifiers on all the five low-dimensional heart datasets. From Table 6.1, it is clear that on Cardiac Biomarkers dataset and Statlog heart dataset the performance of CatBoost is greater than the SLEM model with all classifiers. Hence, on experimenting the SLEM model with different combination of classifiers by backward elimination method, SLEM with CatB and Decision Tree combination of base classifiers gives higher performance on all heart datasets.

Table 6.1 Comparison of performance accuracy of individual classifiers with SLEM on low-dimensional datasets without feature selection

Classifiers	Accuracy (in %) on datasets (without Feature selection)				
	Cardiac Biomarkers Dataset	**Cleveland HD Dataset**	**SA HD Dataset**	**Statlog HD Dataset**	**Cardio-vascular Dataset**
CatBoost (CatB)	96.5	82.5	81.0	80.0	98.37
Gaussian Naive Bayes	78.1	83.5	72.7	80.0	94.6
XGBoost	96.3	80.9	74.4	80.0	98.0
Random Forest	95.7	82.6	78.5	81.1	97.8
Decision Tree (DT)	95.7	74.5	72.7	70.0	96.3
Majority Voting	92.7	84.3	70.2	83.3	96.1
Support Vector Machine	91.5	82.9	71.9	80.0	96.5
K Nearest Neighbors	79.2	80.3	74.4	81.1	84.5
Logistic Regression	86.1	83.7	68.6	80.0	95
Super Learner Ensemble Model (SLEM)	91.7	85.2	80.2	82.2	97.9
SLEM with CatB & DT	**97.9**	**88.5**	**81.0**	**82.2**	**98.8**

6.4.1.2 Evaluation of Classification Accuracy on Low-Dimensional Heart Datasets with Chosen Features

The relevant features which are closely associated with the target variable are selected by MBAR and the SLEM's performance is compared with other classifiers. The accuracies of the SLEM model and all classifiers on the reduced features of the datasets is showcased in Table 6.2. The SLEM model with all base learners gave an accuracy of 95%, 85.5%, 98.6% , 81.0% and 83.3% whereas, SLEM comprising of CatB and DT as base classifiers, provides an improved accuracy of 98.33%, 93.4%, 99.0%, 81.8% and 83.3% on Cardiac Biomarkers, Cleveland HD, South African (SA) HD, Statlog HD and Cardiovascular Disease datasets respectively. In comparison with other models, the SLEM (CatB, DT) demonstrates significantly higher performance.

Table 6.2 Comparison of Accuracies of Classifiers on Low-Dimensional Datasets with Features Chosen by MBAR

Classifiers	Accuracy (in %) on datasets (with Feature selection)				
	Cardiac Biomarkers Dataset	**Cleveland Heart Dataset**	**SA Heart Dataset**	**Statlog Heart Dataset**	**Cardio-vascular Dataset**
CatBoost (CatB)	98.33	85.5	81.8	83.3	99.0
Gaussian Naive Bayes	90.00	86.8	70.2	77.8	94.5
XGBoost	98.33	84.2	76.9	84.4	98.2
Random Forest	98.33	85.5	76.8	83.3	98.6
Decision Tree (DT)	98.33	80.3	76.0	83.3	96.1
Majority Voting	95.00	88.2	71.9	83.3	98.3
Support Vector Machine	78.33	85.5	72.7	77.8	79.7
K Nearest Neighbors	90.00	82.9	72.7	83.3	83.1
Logistic Regression	93.00	86.8	69.4	77.8	96.9
Super learner Ensemble Model (SLEM)	95.00	85.5	81.0	83.3	98.6
SLEM with CatB & DT	**98.33**	**93.4**	**81.8**	**83.3**	**99.0**

6.4.1.3 Evaluation of Performance Metrics of SLEM with MBAR on Low - Dimensional Heart Datasets

The performance measures recall, f1-score, precision, and accuracy of the proposed SLEM on low-dimensional heart datasets with all characteristics andon the features chosen by MBAR are displayed in Table 6.3. Results show that the MBAR with SLEM model has consistently performed better across all the heart diseases datasets.

Table 6.3 Performance Metrics of SLEM with MBAR on Low-dimensional Heart Datasets

Dataset	Classifier	Performance Metrics in %			
		Precision	**Recall**	**f1-score**	**Accuracy**
Cleveland Heart Disease Dataset	SLEM	95.2	76.9	85.1	88.5
	MBAR+SLEM	95.8	88.5	92.0	**93.4**
SA Heart Dataset	SLEM	83.3	84.5	83.9	81.0
	MBAR+SLEM	85.5	83.1	84.3	**82.0**
Statlog Heart Disease Dataset	SLEM	79.5	79.5	79.5	80.0
	MBAR+SLEM	83.7	81.8	82.8	**83.3**
Cardio Vascular Disease Dataset	SLEM	97.9	97.9	97.9	97.9
	MBAR+SLEM	97.9	100.0	98.9	**99.0**
Cardiac Biomarkers Dataset	SLEM	92	100	96	**97.9**
	MBAR+SLEM	95	100	97.24	**98.3**

6.4.1.4 Visualization of the Performances of the Classifiers on Low Dimensional Datasets

Accuracy comparisons between the proposed SLEM model and other classifiers on datasets with all features are displayed in Figure 6.3. After using MBAR to narrow down the datasets' most pertinent features, the suggested SLEM model is compared to the classifiers in Figure 6.4. It is evident that all models that are tree based, show good performance but, the proposed SLEM model together with MBAR, produces the best results across all heart datasets.

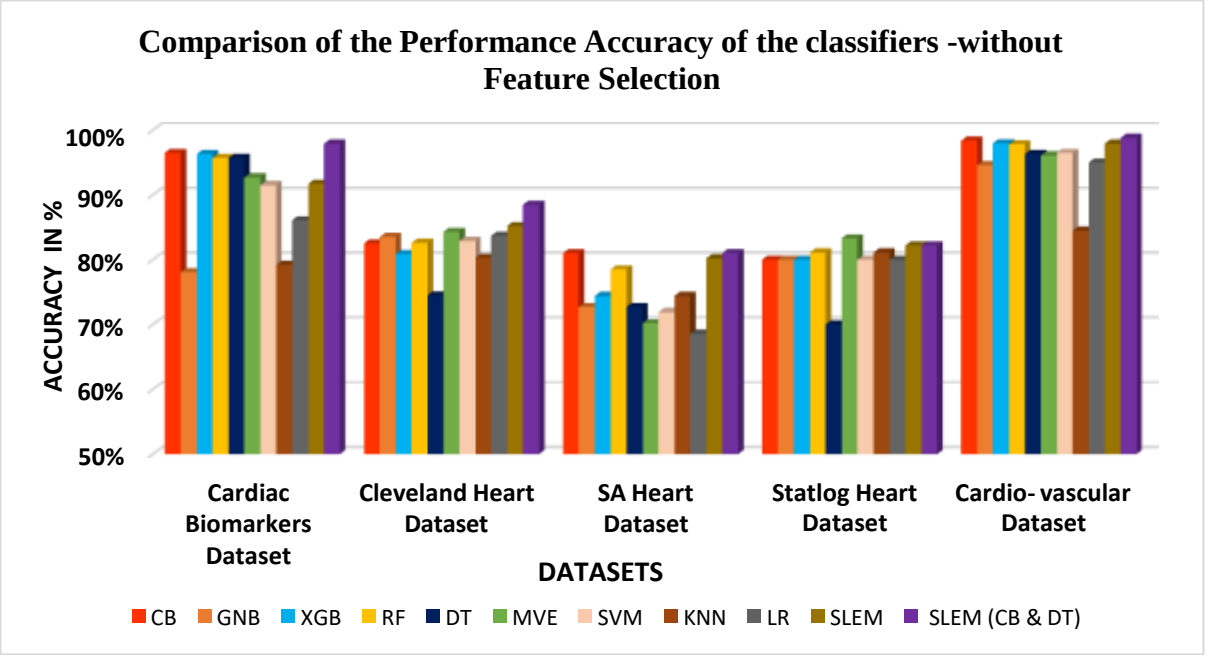

Figure 6.3 Performance Accuracy of the classifiers on Low Dimensional Heart Datasets without Feature Selection

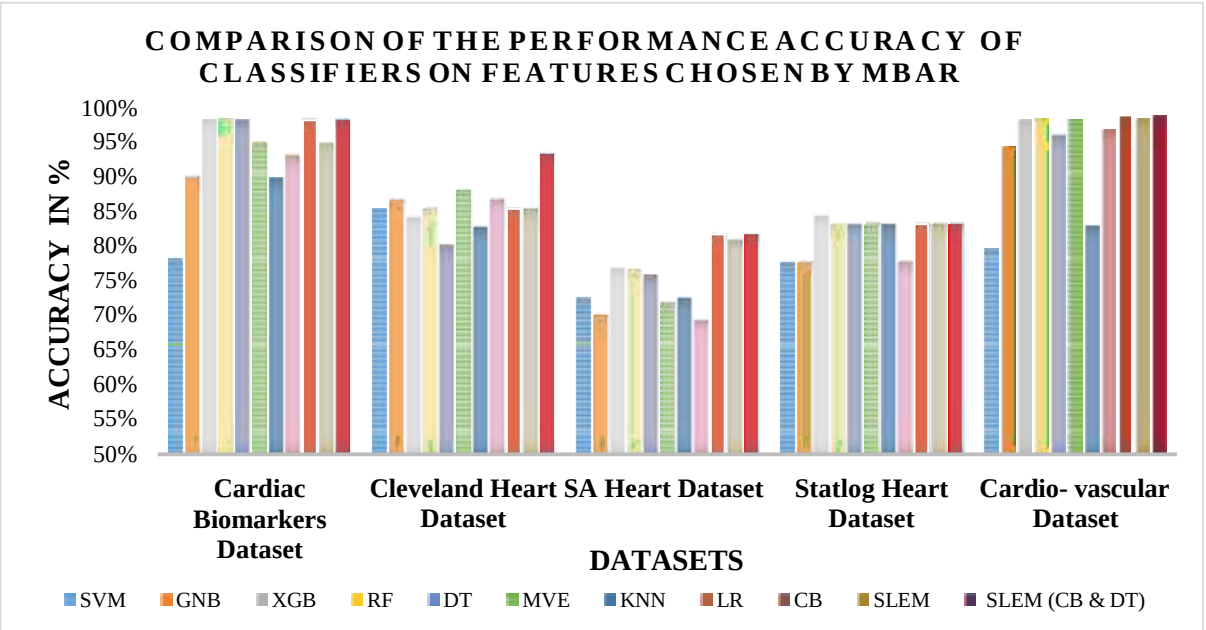

Figure 6.4 Comparison of the Performance Accuracy of the Classifiers on Features Chosen by MBAR

6.4.1.5 Evaluation of classification accuracy on High-Dimensional Heart Datasets with All Features

The proposed SLEM model and all the classifiers mentioned in 6.4.1 are applied on the high-dimensional Heart Datasets, mentioned in section 6.4, with all features. Table 6.4 provides the accuracy of all classifier on performing 3 rounds of stratified–ten -fold cross-validation on the datasets with all features. The SVM model has low performance on Z-alizadeh Sani dataset. On exploring SVM with different kernels and hyper parameter tuning, the SVM's performance could be increased. All bagging and boosting models have shown good performance. The proposed SLEM model with all base learners gives anaccuracy of 87.8% and 90.8% on Arrhythmia dataset and Z-Alizadeh Sani dataset. On choosing, (by backward elimination), CatB and DT as base learners in the ensemble SLEM, an increased performance accuracy of 88.8% in Arrhythmia dataset and 93.1% in Z-Alizadeh Sani dataset is achieved which is also higher than that of other classifiers.

Table 6.4 Classification Accuracy without Feature Selection on High-Dimensional Heart Datasets

High dimensional heart datasets	**Classification Accuracy (in %) without Feature Selection on High-Dimensional Heart Datasets**										
	LR	**RF**	**DT**	**SVM**	**GNB**	**XGB**	**KNN**	**CatB**	**MVE**	**SLEM**	**SLEM** (CB,DT)
Arrhythmia heart dataset	74.5	86.7	75.5	73.5	62.2	86.7	65.3	88.8	78.6	87.8	**88.8**
Z-Alizadeh Sani heart dataset	92.0	92.0	86.2	46.0	87.4	93.1	61.0	93.1	93.1	90.8	**93.1**

6.4.1.6 Evaluation of classification accuracy on high-dimensional Heart Datasets with chosen features

MBAR when applied on the high dimensional datasets considerably reduced the set of features. The performance accuracy of all classifiers and the proposed SLEM model on features selected by MBAR on high-dimensional datasets are tabulated in the Table 6.5. The proposed SLEM model with all base learners gives an accuracy of 88.1% and 93.1% on Arrhythmia dataset and Z-Alizadeh Sani dataset, which is higher than the accuracy of other classifers. On modeling with SLEM consisting of CatB and DT base

learners, an increased accuracy of 90.0% on Arrhythmia heart dataset and 96.1% on Z – Alizadeh Sani HD dataset is achieved. In Table 6.5, it is apparent that the performance of the SLEM (CatB and DT) has significantly increased compared to all other classifiers as well as SLEM with all base learners.

Table 6.5 Performance Accuracy on High-Dimensional Heart Datasets with Feature Chosen by MBAR

Heart datasets	**LR**	**RF**	**DT**	**SVM**	**GNB**	**XGB**	**KNN**	**CatB**	**MVE**	**SLEM**	**SLEM** (CB,DT)
Arrhythmia heart dataset	74.5	89.8	71.4	73.5	75.5	86.7	68.4	89.8	82.7	88.8	**90.0**
Z-Alizadeh Sani heart dataset	94.2	92.0	90.8	74.7	87.4	93.1	61.0	96.1	94.3	93.1	**96.1**

6.4.1.7 Evaluation of Performance Metrics of SLEM with MBAR onHigh -Dimensional Heart Datasets

Considering all features of the two high dimensional Heart Datasets and relevant features selected using MBAR, a comparison of the model evaluation metrics of the proposed SLEM model with CatB and DT as base learners is shown in Table 6.6. Results indicate that the proposed SLEM model when used on the datasets with chosen features demonstrates superior performance.

Table 6.6 Performance Metrics of SLEM with MBAR on High-Dimensional Heart Datasets

Dataset	**Classifier**	**Performance Metrics in %**			
		Precision	**Recall**	**f1-score**	**Accuracy**
Arrhythmia HeartDataset	SLEM	89.4	87.5	88.4	89
	MBAR+SLEM	89.5	89.5	89.5	**90**
Z-Alizadeh Sani Dataset	SLEM	91.8	95.70	93.70	93
	MBAR+SLEM	95.9	97.9	97.29	**96.1**

6.4.1.8 Visualization of the performances of the classifiers on high dimensional datasets

A comparison of the proposed SLEM model with other classifiers on high-dimensional datasets without feature selection is displayed in Figure 6.5. The proposed SLEM provides efficient performance accuracy of 88.8% and 93.1% respectively on the datasets.

Figure 6.6 exhibits the performance accuracies of the classifiers on high- dimensional datasets with features selected by MBAR. The proposed MBAR with SLEM model displays higher accuracy of 90% and 96.1% on the Arrhythmia dataset and Z - Alizadeh Sani dataset, compared to other models.

From both Figures 6.5 and 6.6, it is clear that KNN performs lesser than its counterparts. By experimenting with hyperparameter tuning, KNN's performance can be improved. On observing both Figures 6.5 and 6.6, it is apparent that the bagging and boosting models are showcasing higher performance than the other models considered in this study.

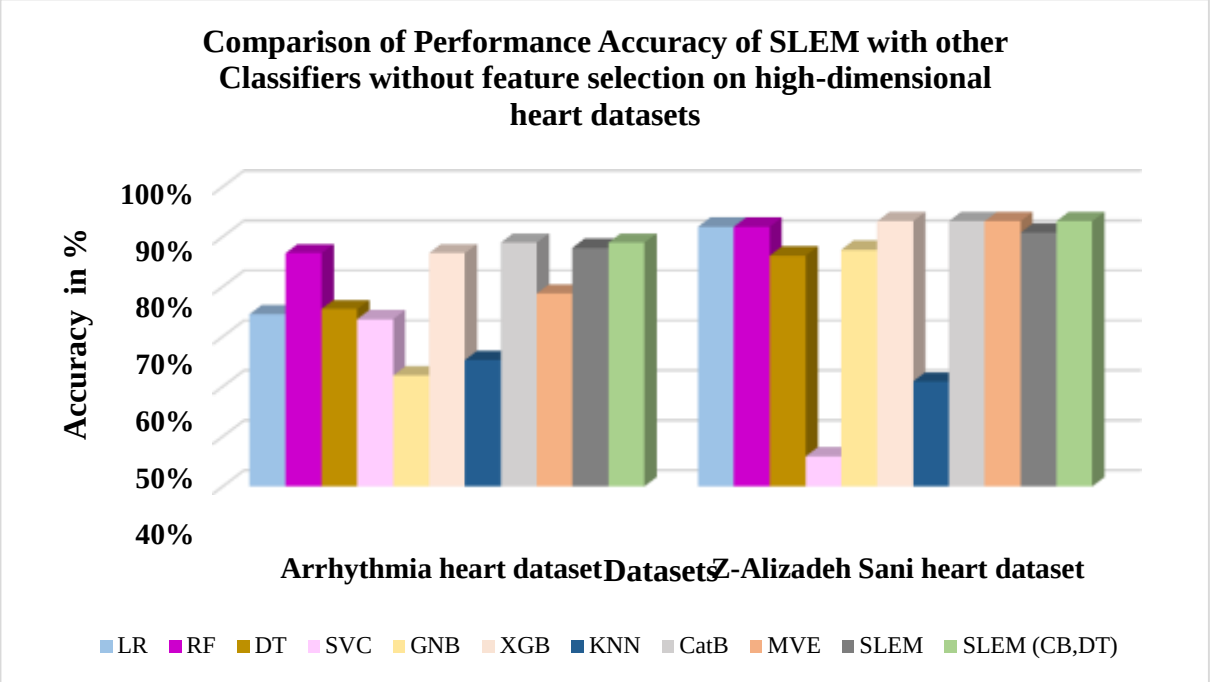

Figure 6.5 Performance Evaluation of the classifiers on High dimensional Heart Datasets

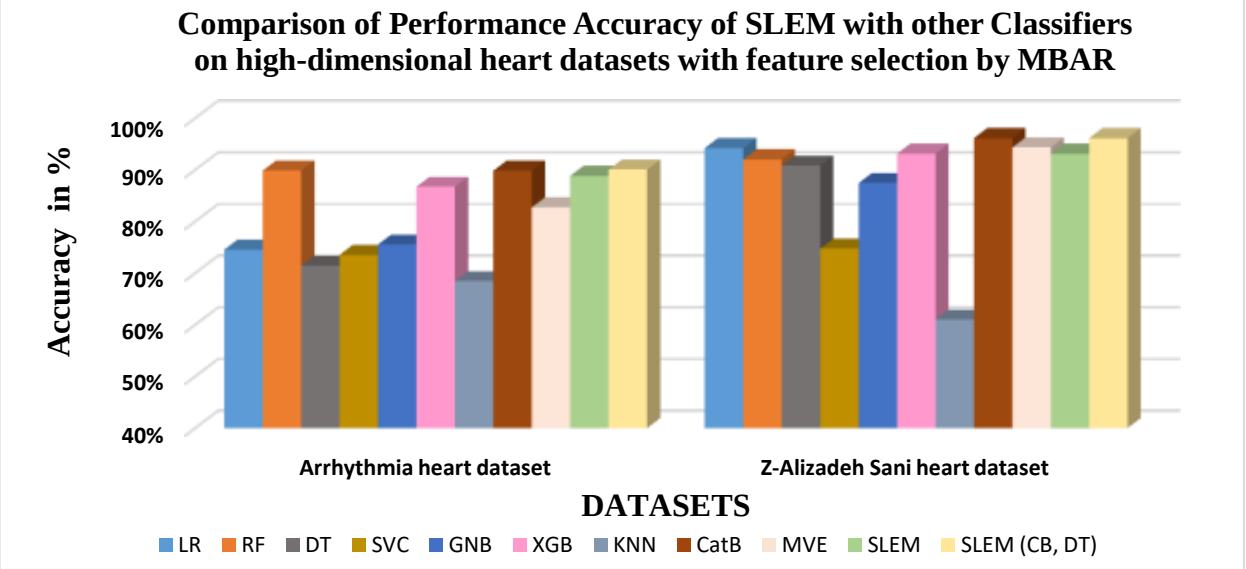

Figure 6.6 Performance Evaluation of the classifiers on High dimensional Heart Datasets with feature selection by MBAR

6.5 CHAPTER SUMMARY

In this chapter, a Super Learner Ensemble Model (SLEM) is devised with a novel combination of base models, for prediction of heart diseases. Using repeated stratified cross validation on the cardiac diseases datasets, a set of high-performing models (SVM, RF, LR, GNB, DT, XGB, CatB, and MVE) were chosen as the base learners of the proposed ensemble model. The suggested SLEM model is built on a foundation of several different types of models, including linear, probabilistic, bagging, boosting, and stacking models.

The ensemble classifier, SLEM, is developed by combining the predictions of all those base classifiers as input to the meta learner, LR, which learns by better integrating predictions from these base models. The SLEM model attains high accuracy when applied on Heart Datasets with features selected by MBAR.

The proposed SLEM's new combination of CatB and DT as foundation models was achieved via the backward elimination method. All things considered, the planned SLEM with CatB and DT as the basis models produced excellent precision, recall, f1 score, and accuracy on high and low dimensional Heart Datasets. In the next phase, the optimization of the selection of base models in the Super Learner Ensemble Classifier is experimented along with the feature selection algorithm MBAR to achieve greater performance.