Model Ensembles for Heart Disease Classification

1st Shuyue Wang
Dept. of Electrical & Computer Engineering
University of Waterloo
Waterloo, Canada
s754wang@uwaterloo.ca

2nd Su Sun
Dept. of Electrical & Computer Engineering
University of Waterloo
Waterloo, Canada
s266sun@uwaterloo.ca

3rd Yue Teng
Dept. of Physics & Astronomy
University of Waterloo
Waterloo, Canada
y9teng@uwaterloo.ca

***Abstract*—Including strike, coronary heart disease, congenital heart disease, etc., heart-related disease is among the reasons of the highest mortality. Accordingly, an early finding and diagnosis of heart disease is inevitable in the way to achieving health. In this project, machine learning methods are proposed and trained to predict the existence of heart disease for a candidate. Five of them are relatively simple baseline modes, including logistic regression, KNN, SVM, decision tree, and random forest. The rest of them are ensemble models constructed on the baseline models. The number of the most powerful models are three. Two of them implemented voting scheme, including soft and hard voting. The last one is an hybrid combination of multi-layer perceptron (MLP) with the five baseline models. After training and test on the data set from UCI's repository, we found that the ensemble models, especially the hard voting model and hybrid model generally outperform any other models, in terms of different metrics.**

***Index Terms*—heart disease, classification, logistic regression, KNN, SVM, decision tree, random forest, voting, multi-layer perceptron**

I. Introduction

Heart disease is one of the most severe threats to human beings and contribute to mortality greatly. According to World Health Organization, by 2030, cardiovascular diseases will kill 23.6 million people globally [1]. The diagnosis of heart diseases plays an important role in prevention and medical treatment. The development of machine learning techniques helps a lot with early finding and diagnosis of many diseases.

The popular machine learning method implementation in heart disease diagnosis has been investigated by quite a number of researchers. Mohan [2] proposed hybrid random forest with linear model (HRFLM) which has 88.47% accuracy in predictions of heart disease diagnosis on Cleveland data set from UCI repository. Rather than accuracy, in terms of other metrics, HRFLM also exhibits excellent performances. Its error rate is the smallest, and its precision, f-measure, sensitivity, and specificity are at the top level, compared with naive bayes, generalized linear model, logistic regression, decision tree, random forest, gradient boosted trees, support vector machine, and VOTE [2]. Buettner and Schunter [3] used random forest algorithm to classify a patient to have heart disease or not. Their model has overall accuracy of 84.448% accuracy with cross validation and 82.895% accuracy without cross validation. Dwivedi [4] evaluated and compared six machine learning techniques, artificial neural network, SVM, logistic regression, KNN, classification tree and naive bayes, using ROC curves, accuracy, precision, sensitivity, specificity, and f1 measure. The researcher found that logistic regression is among the best classifiers since it has the highest accuracy 85% and excellent sensitivity 89%. Artificial neural network also performed well with accuracy 84% and sensitivity 87%. SVM has the highest specificity 89% and precision 90% [4] on StatLog data set from UCI [5]. Saxena et al. [6] conceived a pruned decision tree classifier and realized efficient classification of heart disease patients on Cleveland database. They used ten-fold method to train their model and obtained 86.3% accuracy on test set and 87.3% accuracy on training set.

Among all the machine learning algorithms, neural networks related researches grows fast and revealed many advantages of neural networks. Li et al. [7] developed an artificial neural network to classify congenital heart disease cases in pregnant women. Their model was trained and tested on 358 samples from hospital and has training and test accuracy 0.91 and 0.86, respectively [7]. This model has an AUC of 0.87 under the ROC curve for test set. Dangare and Apte [8] compared their results from neural network with those from decision tree and naive bayes. They found that the neural network outperformed any other models in their cases, in terms of not only accuracy, but also confusion matrix [8]. The confusion matrix of neural network has diagonal entries dominate the matrix and other entries be zeros. Abushariah et al. [9] associated artificial neural network with adaptive fuzzy inference system and obtained 87.04% accuracy on test set, which beat their fuzzy inference system with only 75.93% accuracy. Their models were trained on Cleveland database from UCI repository. The neural networks implemented in most researches related to heart disease are of relatively naive configuration, multi-layer perceptrons. As the researches go deep inside, more diverse configurations specific for heart disease diagnosis may appear.

A. Overview

Correctly classifying heart disease samples from the data set of UCI machine learning repository as much as possible is the goal of this project. At the very beginning, preprocessing the data set is of vital importance, since we have to properly deal with NAs if they exist and normalize some of the data if necessary. The data set would be checked to confirm that it is balanced. Apart from this, some primary coarse analysis was done based on the distribution of different features and how positive and negative cases distribute in these features.

Before constructing the models, data set would be divided into training and test sets stochastically.

As we know, individual models generally have many drawbacks in classification qualities. Even if one model stands out in terms of one metric, it sometimes behaves bad in other metrics. Newly proposed models would be built upon five baseline models, logistic regression, KNN, SVM, decision tree, and random forest. There are several complicated models on the basis of baseline models, but three of them are mainly used. The first two of the three models exploit soft and hard voting scheme to process the output from a single model layer based on five baseline models. The other hybrid model is a combination of five baseline models and multi-layer perceptrons. There is no voting done in this model. All the three complicated models have the baseline models run separately.

Evaluations on the baseline models and three ensemble models are of great importance. After training and test processes, four metrics on the basis of confusion matrix were calculated. They are accuracy, precision, recall, and f1-score. All the four metrics on both training and test sets for the eight were given, to provide an overall judgement on the models' performance regarding overfitting, generalization, and predictions for positive and negative cases. Subsequent analyses and proposals were deduced from these evaluations.

B. Models

There are 5 baseline models used as classifiers in this project, logistic regression, KNN, SVM, decision tree, and random forest. The most common form of logistic regression is binary for classification tasks. It uses a mathematical expression for the probability of belonging to a class,

$$P(y = 0|x) = \frac{1}{1 + \mathrm{e}^{-(w^{\mathrm{T}}x+b)}}. \tag{1}$$

Thus, the probability of $y = 1$ is $1 - P(y = 0|x)$. The overall probability of all the training data points in their own classes is the product of their probabilities. The log-likelihood which is the log function of the overall probability is going to be maximized. For multi-class classification, sigmoid function will be replaced with softmax function for the probabilities of different points in more than two classes. Or the one-versus-rest or one-versus-one strategy could be applied to binary classifiers instead.

K-nearest neighbours is a supervised learning algorithm. Similar data points of the same class are assumed to be grouped together in feature space. Thus, a test point obtains its predicted class by the algorithm traversing all the training points and find the nearest k neighbours. The most frequently appeared label in the neighbours is the predicted label for the test point. The test point's probabilities of being in different classes could also be approximated by frequency of corresponding labels. K-nearest neighbours algorithm virtually does not have a explicit process of training the model, but spend time generating predictions.

Support vector machine (SVM) could be leveraged as classifier in binary classification. It is an algorithm trying to find the hyperplane in feature space best separating two classes. Its margins maximize their distance, and fewer data points within the margins are preferable than many points, if the SVM is soft. There is a parameter C denoting SVM's penalty for points existing between margins. If C is quite large, SVM will hardly tolerate points between the margins, not so soft. The nonlinear classification problem could be addressed by mapping data points using kernel function (not mapping function itself) into a new feature space where they are more linearly separable. Multi-class classification problems need multiple SVMs, by using one-versus-one or one-versus-rest strategy.

Decision tree is an algorithm able to accomplish classification tasks utilizing a tree composed of branches, nodes, and leaves. A node accommodates a feature where data set would be split by the feature's value, continuous or discrete. Different branches from this node directs their data subset to different nodes. A leaf node is assigned a label according to its data subset's most frequent label. Decision tree is an algorithm doing feature selection while training itself. Training process in each node will traverse every available features to obtain the one bringing the most information gain. Decision tree is quite likely to overfit, so it is important to prune some nodes to acquire better generalization ability.

Random forest is an algorithm based on decision tree. It is an algorithm with the help of an ensemble of decision trees. Each decision tree was trained in data set extracted from the original data set by bootstrap sampling. Besides random data point selection, feature selection is also random. Instead of regarding all the available features of a node as candidates, a randomly selected subset would be formed. The size of the subset could be adjusted. Feature selection rules applied to this subset. Random forest is quite simple and usually works really well.

The ensemble model needs a method to combine the baseline models' output. The continuous outputs from different models could be averaged with weights to conclude a number, but the classification outputs here have to leverage voting strategy. In terms of voting strategy, there are majority voting, plurality voting, and weighted voting. Majority voting choose the class with more than half scores in the whole. Plurality voting favors the class possessing the most scores. Weighted voting based on plurality voting calculates the score for a class using weighted average. Voting strategy has two options to present an output score for a sample in a class, soft voting or hard voting. Soft voting refers to the scores as probabilities, but hard voting regards the scores as 0-1 encoded results. Namely, for hard voting, the score of a sample in a class is 1 and the other scores for it in other classes are 0.

Multi-layer perceptron (MLP) is a model prospering in recent years due to optimization techniques like backpropagation and the development of computation abilities. It is a neural network with at least three layers of fully connected neurons. It uses nonlinear activation function like sigmoid, tanh, ReLU, etc., so MLP is highly promising in solving nonlinear classification problems. MLP is trained based on supervised learning using backpropagation technique. It has

been known that the MLP has ability to mimic any continuous functions, and a brief proof was given in [10].

C. Data set

The heart disease data set from UCI machine learning repository [11] is going be to the training and test source for this project. To access the data set, please follow the link https://archive.ics.uci.edu/ml/datasets/Heart+Disease. This data set from UCI has four databases, Cleveland, Hungary, Switzerland, and the VA Long Beach. Among the 76 features, 14 features, age, sex, cp, trestbps, chol, fbs, restecg, thalach, exang, oldpeak, slope, ca, thal, num, should be used. The label called goal, has integers 0, 1, 2, 3, and 4 indicating whether the disease presents (0) or not (1, 2, 3, and 4), and what category the sample belongs to if disease presents. There are 920 samples in all the four databases and 303 samples in the most commonly used Cleveland database. Cleveland is going to be the database used in this project. A summary of the type of the features in the data set is summarized in table I. For the first column, nominal features with categorical values are listed, and their own categories are presented in the middle column. The last column shows ordinal features, where the features except age are continuous. The meaning of every integers standing for different categories are not provided, the same to the features' meanings, since they are not the prerequisites for this project. Who are interested in the meanings could look for the website provided above for help.

TABLE I
A BRIEF SUMMARY OF THE FEATURE TYPES AND THE INTEGERS FOR DIFFERENT CATEGORIES OF NOMINAL FEATURES.

Nominal features	Categories	Ordinal features
sex	1, 0	age
cp	1, 2, 3, 4	trestbps
fbs	1, 0	chol
restecg	0, 1, 2	thalach
exang	1, 0	oldpeak
slope	1, 2, 3	
ca	0, 1, 2, 3	
thal	3, 6, 7	
num	0, 1, 2, 3	

II. Implementation

A. Preparation

1) Data exploration: Data exploration was conducted to get familiar with this dataset.

By checking the summary information of each column, there is no *NaN* value need to be fix. And there are 303 samples in total and each of them contains 13 features and one label indicating if this patient has heart disease or not. As shown in Figure 1, there are 165 positive samples (patients with heart disease) counting 54.46% of the whole dataset and 138 negative samples, normally sampling method is enough for this dataset as it is balanced.

Figure 2 shows the relationship between each categorical feature with the target, from which it can be observed that

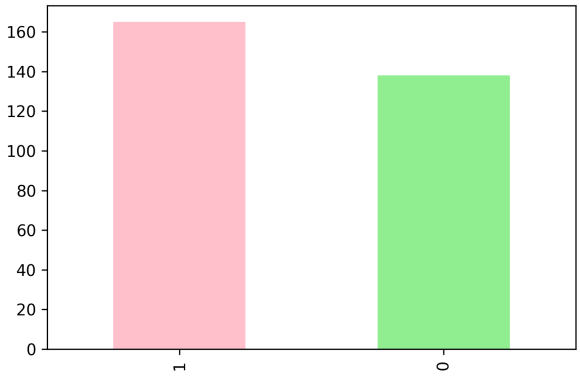

Fig. 1. Distribution of the Two Categories

people already have chest pain (cp = 1, 2, 3) are more likely to have heart disease than people did not have chest pain. Also, people having abnormal ST-T waves (restecg = 1) or who have had the experience of angina induced by exercising (exang = 0) have a higher possibility of getting heart disease. Besides, people whose ST segment is downsloping (slope = 2) at peak exercise and who have had a heart attack and remain reversible defect (thal = 2) are more likely to get heart disease. Additionally, people with more number of major vessels (ca $\geq$ 0) tend to be healthier.

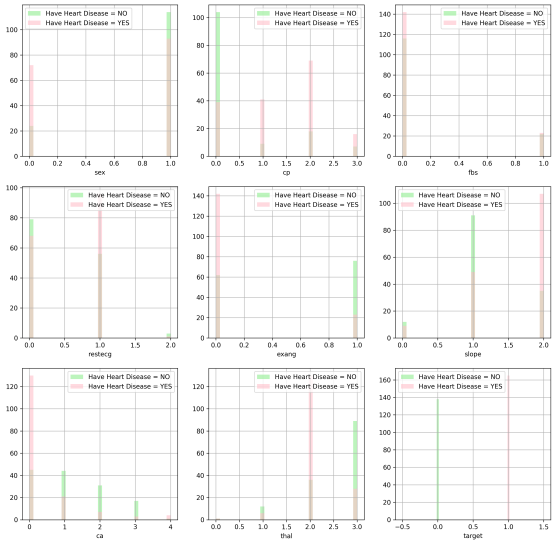

Fig. 2. Exploration of Categorical Features

Figure 3 shows the relationship between each continuous feature with the target. People whose resting blood pressure is higher than 130, or serum cholesterol is higher than 200 have

higher possibilities to get heart disease. And people who can achieve a higher than 200 maximum heart rate will put their hearts under more pressure.

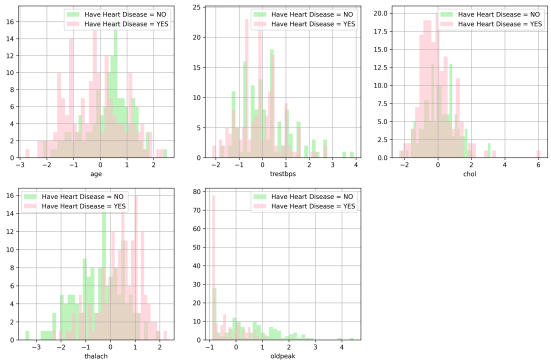

Fig. 3. Exploration of Continuous Features

2) Data Preprocessing: To dive into this dataset, more descriptive statistics information was generated including the shape of the dataset's distribution, the mean, min, max, and standard deviation value of each column. The values of the features 'age', 'trestbps', 'chol', 'thalach', and 'oldpeak' are not categorical values like the other features, especially the average values of 'age', 'trestbps', 'chol', and 'thalach' are 54.5, 131.6, 246.3, and 149.6 while others are fluctuating around 1. Thus, standard normalization was applied to scale these five features mentioned above individually by computing the relevant statistics on each sample in the data set. The standard score is calculated using Equation 2:

$$Z = \frac{x - \mu}{\sigma} \tag{2}$$

where μ is the mean of one column of the training samples and σ is the standard deviation of one column of the training samples. After scaling the necessary values and divide the dataset into features and labels, this dataset was further split into a training set and a testing set using by randomly choosing 20 percent of the data as testing set and the samples remain are training set.

B. Single Model

All the 5 baseline classifiers were trained on the training set which contains 242 samples to find patterns and then use these patterns to predict the results on the test set. Confusion matrix, overall accuracy, precision, recall, and F1-score were calculated using the equations given below to evaluate each classifier's performance on both the training set and testing set.

$$Accuracy = \frac{TN + TP}{TN + TP + FN + FP} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F1Score = \frac{2 * (Recall * Precision)}{Recall + Precision} \tag{6}$$

1) Logistic Regression: Logistic regression is widely used in multi-class classification problems and so was chosen as the first baseline method in this project. L2 regularization was applied to increase this classifier's generalization ability by avoiding the algorithm from overfitting the training samples.

2) K-Nearest Neighbors: K-Nearest Neighbors method can be used for both regression and classification as well as nonlinear classification. As KNN is a kind of lazy learning, it only needs to store the input samples' feature vectors and class labels in the training period and doesn't pre-build any actual model but does all the calculations in the prediction process. The test sample's class label was assigned based on its nearest five neighbors' labels.

3) Support Vector Machine: SVM is a supervised machine learning algorithm that is popularly used in classification problems. It tries to find a hyperplane to separate the samples into two classes. The hyperplane would try to classify the sample without error and maximize the distance to the closest vectors from itself. Using kernel trick allows SVM to achieve nonlinear mapping. Proper C (1) and relatively smaller gamma (0.1) was used in this classifier, where a proper C value will let the model not only try to classify the samples correctly but also keep certain generalization ability and smaller gamma value can lower the model complexity.

4) Decision Tree: The decision tree classifier used Gini impurity to decide the optimal split (use which features). Gini Impurity shows the probability of misclassifying a sample when its label is selected randomly at the current node, which will peak at uniform probability more strongly than entropy impurity.

5) Random Forest: Random forest is a model which consists of multi decision trees. The random forest algorithm has a crucial feature to prevent over-fitting and does not require feature scaling as well as categorical feature encoding. This classifier used 100 individual decision trees as an ensemble to get final predictions.

C. Model Ensembles

As the performance of each single model is not ideal for this dataset, three complex ensembles of models are developed to further digest this dataset. This introduces the voting scheme as the strategy of how to deal with the predictions from different single models, as well as the complex architectural design of the whole ensemble networks. The performance of these three ensembles are compared with the baseline models stated in the previous section, in terms of accuracy, precision, recall and f1-score. According to the comparison results which demonstrated in the Section III, these three ensembles outperforms each of the single baseline model. The details of the model ensembles are illustrated in the following section.

1) Soft Voting: The first ensemble of models is called Soft Voting model since it uses the soft voting scheme in the layer after the layer of multiple single-model. The input, which is either training set or testing set, is fed into several single models as a whole set. By that it means the input set is learned and trained by several single models simultaneously in the multiple single-model layer. In this ensemble, all of the five baseline models are regarded as single models separately in the multiple single-model layer. In other words, there are five models in the single-model layer, which are logistic regression, KNN, SVM, decision tree and random forest. Each of the single model can generate a prediction for the input dataset. Then, the probabilistic prediction results from different single models are transited into the soft voting layer where the voting scheme is performed. To be more specific, the voting scheme generates the unweighted mean of the predictive results from the previous layer, and output the overall prediction. The reason of why this model is named soft voting is that the input of the voting layer is the probabilistic result rather than the classification results from previous single models. The architecture of the soft voting model is shown in Figure 4.

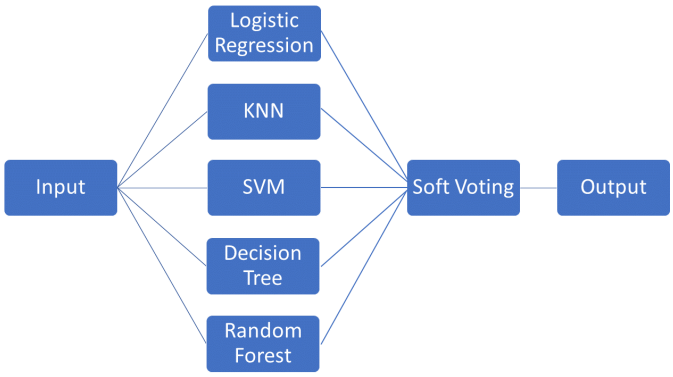

Fig. 4. The architecture of soft-voting model

2) Hard Voting: The second ensemble of models is called Hard Voting model. Compared to the first model which uses the soft voting layer after multiple single-model layer, this model chooses to use the hard voting layer instead. As what is designed in the first ensemble, the samples and features are input into different single models respectively. Moreover, the single models are also logistic regression, KNN, SVM, decision tree and random forest models, exactly the same as the multi-single model layers in the first ensemble. Then, the five outputs from the five single models are collected and fed into the voting engine. However, what is different from the first ensemble is that the results generated from the five single models are not probabilistic predictions, and they are, by contrast, classification results which contains only zeros and ones. This is also the reason why this model is called hard voting model. Finally, the voting scheme is applied to the predictions from the different classifiers, and the class with the majority of votes should be assigned to the final predicted class. The number of single models utilized is odd, so that the functionality of the voting scheme can be guaranteed, and there should not be a case that two classes win the same number of votes. The architecture of the hard voting model is shown in Figure 5.

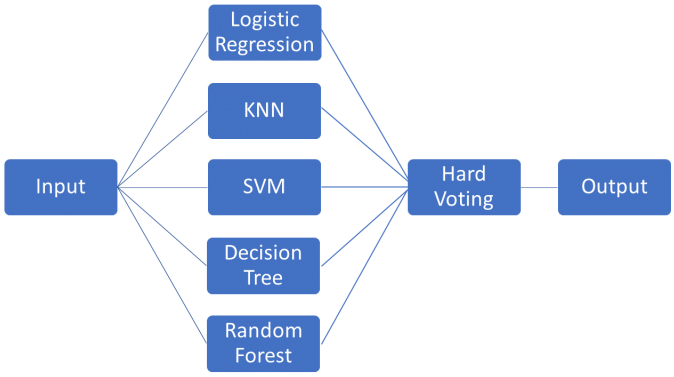

Fig. 5. The architecture of hard-voting model

3) Hybrid Model: The third ensemble of models is a hybrid model including the applications of logistic regression, KNN, SVM, decision tree, random forest and neural network. There is no voting scheme in this ensemble, but the ensemble relies on the collaboration among different models to enhance the overall performance. The architecture of this ensemble contains seven layers, five of which are hidden layers, and the architecture is shown in Figure 6.

The same as the first ensemble and the second ensemble, the whole data set is fed into five single models separately. The five single models are five baselines stated above, namely logistic regression, KNN, SVM, decision tree as well as random forest. The classification results, which contains either zero or one, from the five models are re-shaped into a ten dimensional dataset, and this new dataset is regarded as the input of the neural network. The first two dimensions of the newly built dataset is the predicted result from the first baseline model, specifically logistic regression model in this case. The third and the forth dimensions are the classification results from the second model, which is KNN in this ensemble, and so on so forth. The functionality of the neural network is to analyze the features that are captured by the five different models, and increase the classification accuracy by increasing the complexity of the network. In details, the neural network contains three hidden layers, first of which uses 16 neurons as well as ReLu activation function. The second layer and the third layer contains 8 and 4 neurons respectively, followed by a ReLu activator. Furthermore, all of the three layers are followed by a dropout layer with the dropout rate of 0.3, for the purpose of avoiding overfitting. The function of softmax is applied in the output layer in order to generate the final classification result.

4) Other Models: There are some other designs of the ensembles have been tried out in this paper, but the performance is not as good as the three ensembles illustrated above. The

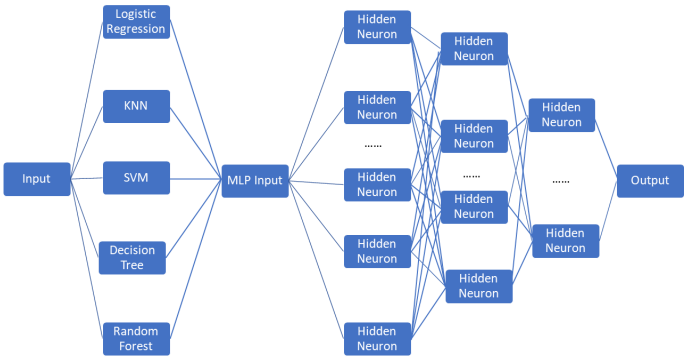

Fig. 6. The architecture of hybrid model

overviews of the other ensembles and the potential reasons of the failure are discussed in this section.

The first failed ensemble uses the logistic regression in the layer after the multiple single-model layer. The add-on logistic regression is treated as the function to generate the final output based on the predictions of the five single models. This ensemble extremely overfits the dataset because it results in 100% accuracy on the training set while only 75% accuracy on the testing set.

The second failed model is the random forest using random search for parameter tuning. The random search includes the searching scheme for the parameters of the number of trees ranging from 100 to 2000, the maximal number of features to split a tree node, the maximal depth ranging from 3 to 30 in each decision tree, the minimal number of data points in the range of 1 to 5 allowed in a leaf node, the minimal number of data points in the range of 1 to 10 for stopping a split, as well as the different sampling methods. However, though a long-time tuning is developed, the result only achieves 85% testing accuracy with 97% training accuracy. The potential reason of the failure may be the unsuitable model being used. The overall performances of random forest and decision tree are not as good as other algorithms, and the tree-based model is very easy to overfit this dataset.

III. Results

A. Baseline Method Results

Table II is a summary of the training performance of the five baseline models. And table III shows the baseline models' testing performance.

TABLE II
Training Performance

Classifier	Accuracy	Precision	Recall	F1-score
Logistic Regression	86.36%	85.71%	90.32%	87.91%
K-Nearest Neighbors	88.02%	86.11%	93.23%	89.53%
Support Vector Machine	90.08%	87.59%	95.49%	91.37%
Decision Tree	100.00%	100.00%	100.00%	100.00%
Random Forest	100.00%	100.00%	100.00%	100.00%

It can be observed from the tables that the highest testing accuracy using machine learning methods with almost default

TABLE III
Test Performance

Classifier	Accuracy	Precision	Recall	F1-score
Logistic Regression	85.25%	87.10%	84.38%	85.71%
K-Nearest Neighbors	86.89%	85.29%	90.62%	87.88%
Support Vector Machine	86.89%	85.29%	90.62%	87.88%
Decision Tree	75.41%	84.00%	65.62%	73.68%
Random Forest	83.61%	84.38%	84.38%	84.38%

parameters can achieve is 86.89% (provided by KNN and SVM classifier). It is worth mentioning that the decision tree classifier did not perform very well, although its training accuracy can reach 100%, overfitting did occur, the test accuracy is only 75.4%. In the random forest classifier, the testing accuracy can be increased by using more decision trees in the forest, but the time cost will also be increased at the same time.

B. Ensemble Models Results

Figure 7 shows the comparison of accuracy score between baseline models and the three ensemble models, where all the three ensemble models achieve higher overall accuracies on the testing set. Hard Voting and the Hybrid model can even reach 90.16 percent accuracy.

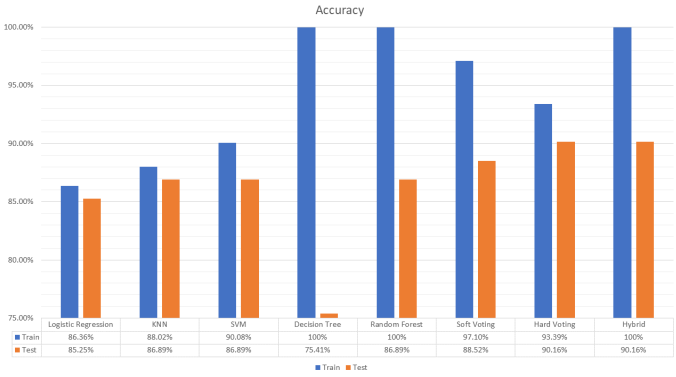

Fig. 7. Accuracy Comparison

The precision score, a quality measurement to evaluate how useful the predicted results are, are plotted in Figure 8. Ensemble models' performance is also better than single models, Hard Voting and Hybrid model can achieve 90.62% precision score, indicating more than 90 percent of the results are relevant.

The recall score, a quantity measurement to evaluate how complete the predicted results are, are plotted in Figure 9. Ensemble models' performance is also better than single models, Hard Voting and Hybrid model can achieve 90.62% recall score, meaning that most of the relevant results (more than 90 percent) were returned by these two models.

Figure 10 shows the comparison of f1 score among single and ensemble models. Since F1 score is the weighted average of the precision and recall, and the best precision and recall score provided by the Hard Voting model and Hybrid model

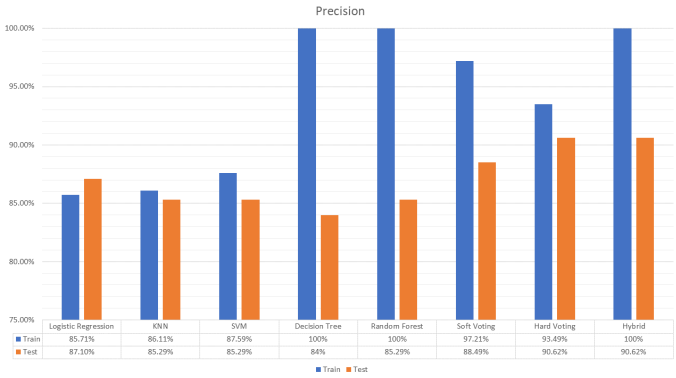

Fig. 8. Precision Comparison

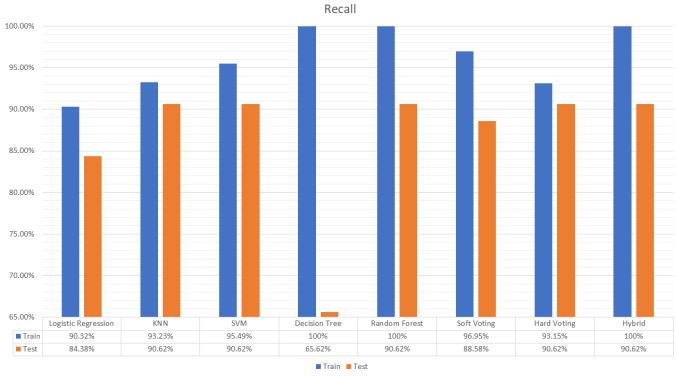

Fig. 9. Recall Comparison

are the same, the best F1 score is 90.62 percent among all the models.

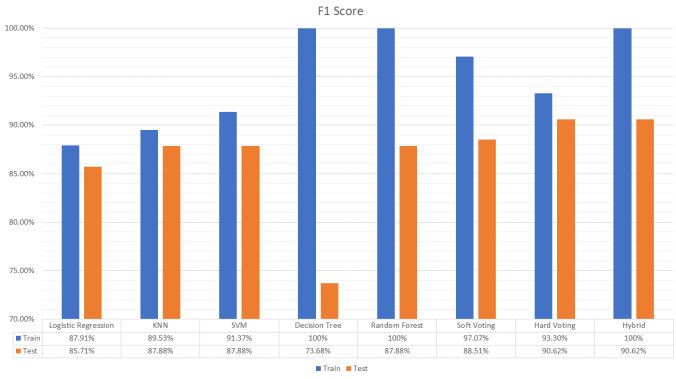

Fig. 10. F1 Score Comparison

IV. Analysis

According to the no free lunch (NFL) theorem [12], there is no single model which is optimized for every set. Different models use different algorithms and they focus on different pattern recognition. Therefore, the ensemble of models is necessary to be explored and developed for better generalization and performance. The three ensembles illustrated above doubtlessly outperforms each of the single model in terms of all of the four metrics: accuracy, precision, recall and f1-score.

The four metrics are all taken into account for the purpose of performance comparison because each metric represents different performance measurements.

Accuracy is the most widely-used measurement for model comparison, and it works well for the symmetric datasets where the number of the true labels in every class is the same. In this dataset, the distribution of the different classes is in the same level with different exact counts according to Figure 1. Accuracy represents the ratio of correct prediction and classification, so it should be used for measuring the correction of the performance.

By contrast, precision only measures the positive labels since it calculates the ratio between the true positives and all of the positives. Precision is more useful in the situation of the prediction of the positives are important. In our dataset, since the classification problem is the disease identification, the precision is very important for the medical diagnosis and in-time health care.

Recall measures the sensitivity of the model to detect the true positives against all correct classifications. It represents the ability of a model to classify the positive labels correctly compared to the overall correct predictions. In the bio-informatics researches, the recall measurement is very importance because the model with high recall value can be able to distinguish the real patients and apply the corresponding treatments.

Precision and recall are both important in bio-informatics research area [13], and this results in that there are many researchers prefer to use f1 score as the measurement. F1 score combines the metrics of precision and recall, and calculates the weighted average between them. Though f1 score is more popular for the imbalanced dataset, it can also be used for this dataset to provide an insight on the performance comparison.

From all of the four metrics applied on all the models, according to Figure 7, Figure 8, Figure 9 and Figure 10, it is noticeable that the three ensembles show better performance across all of the four metrics in terms of test results. To be more specific, the three ensembles achieve higher accuracy than any single models, with hard voting model as well as hybrid model showing the highest accuracy. The same pattern can be discovered for the precision comparison among all models developed, where the hard voting ensemble as well as hybrid model represents the best precision rate, followed by the soft voting model. While the three ensembles do not display an extreme advantage in recall comparison, the hard voting ensemble as well as hybrid model still take up the best recall rate in test. Again, for the f1 scores comparison, the three ensembles outperforms all of the single models, with the hard voting ensemble as well as hybrid model showing the best performance.

From the comparison between ensembles and single models, it is obvious that all of the three ensembles show the better performance than each of the single model. One of the underlying reasons may be that each single model can only extract a part

of the features in a linear or non-linear way, while ensembles of models take advantages of different single models and balance the pattern recognition abilities from different models. On the other hand, the ensembles of models enhance the confidence of each prediction for test samples by combining and summarizing the classification results across several single models. Thus, in terms of the comparison between single models and ensembles, a conclusion can be drawn that the ensembles have the better abilities to learn, fit and further predict the dataset.

By comparing the performance between soft voting ensemble and hard voting ensemble, it is undeniable that the hard voting ensemble has the better performance across all of the four metrics. As illustrated in the Section II-C, the only difference between the soft voting ensemble and the hard voting ensemble is the input for the voting layer. To be more specific, the input for the soft voting scheme is the probabilistic predictions from the single models, whereas that for the hard voting scheme is the classification result from the previous layer. The classification result is the sigmoid output from the probabilistic predictions, so that the classification results contains either zeros or ones while the probabilistic results are filled with decimals in the range from zero to one. The performance comparison between soft voting ensemble and hard voting ensemble represents the functionality of sigmoid application. By using the rounded results, the hard voting achieves better test accuracy, precision, recall and f1 score while requiring lower training accuracy, precision, recall and f1 score. There are some arguments about the necessity of using sigmoid function in the complex machine learning networks [14], and in this paper, we show that it is necessary and useful to use sigmoid function in the our ensemble network for better performance.

It is certainly that the hybrid model is the overall best model across all models developed, since the hybrid model achieves the best training and testing result across all of the four metrics. As for the training process, decision tree and random forest, as well as the hybrid model all achieve 100% correct classification. However, decision tree and random forest overfit the training set and show bad performance during the testing process. By contrast, the hybrid method not only avoids the overfitting, but also achieves the best result for the testing set, with only 6 out of 61 testing samples are mis-classified. Therefore, when taking both training and testing processes into account, the hybrid model is the best across all eight models proposed.

Through the comparison between the hard voting ensemble and the hybrid ensemble, it is noticeable that the testing performance of these two models are in the same level across four metrics, while the hybrid model trains the dataset more thoroughly. The hard voting ensemble should be more generalized than the hybrid model, because in the aspect of the algorithms, the hard voting scheme has better generalization ability than a specific neural network. In addition, through the observation of the results, the training and testing results from the hard voting ensemble are in the closer level than those from hybrid model, and this is another evidence showing that the hard voting ensemble can be better generalized.

In comparison with the other existing models in Kaggle competition [15], where the majority of the best models achieves 80% to 88% accuracy, it is surprising that our hard voting ensemble as well as hybrid model outperforms all of them, with 90.16% testing accuracy. Also, according to Table IV where listed the overall accuracy comparison among our models and the other models, our hard voting ensemble as well as the hybrid model not only outperform those best models in Kaggle competition, but also exceed the testing accuracy from the recent proposed complex models in [2], [3] as well as previous classic models in [6], [9]. Nevertheless, our soft voting ensemble is also better than most of the other models, with only KNN from [6] achieves better testing accuracy. Thus, our designs of the ensembles not only work very well for the heart disease dataset and provide an impressive performance, but also bring the insights of how to make full use of different single models and offset the blind spots in different models.

TABLE IV
PERFORMANCE COMPARISON WITH OTHER MODELS TRAINED ON THE SAME DATA SET

Classifier	Training Accuracy	Testing Accuracy
Soft Voting	97.1%	88.52%
Hard Voting	93.39%	90.16%
Hybrid	100%	90.16%
[2]	N/A	88.47%
[3]	N/A	84.45%
[6]	87.30%	86.30%
[9]	90.74%	87.04%
KNN [16]	84.43%	89.01%
Binary Neural Network [17]	87.19%	81.97%

V. CONCLUSIONS

As the heart disease kills millions of people worldwide every year, the prediction and diagnosis of it is the first step and one of the most vital steps towards good curative effect. Machine learning is a fast and efficient tool in developing the auxiliary classification system. We trained and tested five baseline models, including logistic regression, KNN, SVM, decision tree, and random forest. Like other models in [2], [3], [6], [9], [16], [17], the four baseline models excluding decision tree have quite comparable accuracy with their results. Including our baseline results, most test and training accuracy from different classifiers are ranged from 80 to 90%. As what we expected, the ensemble models enhanced the classification quality on test set. Especially, soft voting and hybrid models have accuracy over 90% which are also high-quality classification compared with other heart disease classifiers in [2], [3], [6], [9], [16], [17]. This demonstrates that the hard voting and MLP combination strategies are efficient in leveraging different classifiers' different insights towards the problem. Although the absolute increments of accuracy and other metrics are not dramatic, given the overall level of prediction quality on Cleveland database, we still think the improvement from the ensemble models makes great sense.

The mentioned comparisons could be found in table IV, which intuitively exhibits the quality of our models.

VI. Future Work

A larger dataset is required to further improve the model's performance (classification accuracy). It has been shown that there are 13 features investigated in all the training processes, even if the sample number has been more than ten times of feature space dimension, the data set is still too small to conceive good enough prediction metrics. Considering only 13 of 76 features are commonly used even in other researches, the data set is quite insufficient. Cleveland database is the only one addressed in this project, as what other researchers did, and databases from other countries or regions are not favorable. This implies that the data set lacks enough diversity, which plays the essential role in model training quality. Hence, as this data set is too small which only contains 303 samples in total, and sufficient and diverse data are strongly required.

Besides the data set, improvement in baseline models, like pruning the decision tree, implementing regularization, may be modified to achieve better classification without overfitting to reinforce the basement of the ensemble models. Futhur improvement could also be done regarding the voting strategies, such as smartly assigning weights to different baseline results. Design of neural network configuration and selection of hyperparameters could be tuned and polished to better adopt the heart disease problem.

References

[1] "About cardiovascular diseases," Sep 2011. [Online]. Available: https://www.who.int/cardiovascular_diseases/about_cvd/en/

[2] S. Mohan, C. Thirumalai, and G. Srivastava, "Effective heart disease prediction using hybrid machine learning techniques," *IEEE Access*, vol. 7, pp. 81 542–81 554, 2019.

[3] R. Buettner and M. Schunter, "Efficient machine learning based detection of heart disease," in *2019 IEEE International Conference on E-health Networking, Application & Services (HealthCom)*. IEEE, 2019, pp. 1–6.

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

[5] R. D. King, "Statlog databases," *Department of Statistics and Modelling Science, University of Strathclyde, Glasgow, UK*, vol. 535, 1992.

[6] K. Saxena, R. Sharma *et al.*, "Efficient heart disease prediction system," *Procedia Computer Science*, vol. 85, pp. 962–969, 2016.

[7] H. Li, M. Luo, J. Zheng, J. Luo, R. Zeng, N. Feng, Q. Du, and J. Fang, "An artificial neural network prediction model of congenital heart disease based on risk factors: a hospital-based case-control study," *Medicine*, vol. 96, no. 6, 2017.

[8] C. S. Dangare and S. S. Apte, "Improved study of heart disease prediction system using data mining classification techniques," *International Journal of Computer Applications*, vol. 47, no. 10, pp. 44–48, 2012.

[9] M. A. Abushariah, A. A. Alqudah, O. Y. Adwan, R. M. Yousef *et al.*, "Automatic heart disease diagnosis system based on artificial neural network (ann) and adaptive neuro-fuzzy inference systems (anfis) approaches," *Journal of software engineering and applications*, vol. 7, no. 12, p. 1055, 2014.

[10] M. A. Nielsen, *Neural networks and deep learning*. Determination press San Francisco, CA, 2015, vol. 2018.

[11] A. Janosi, W. Steinbrunn, M. Pfisterer, and R. Detrano, "Uci machine learning repository-heart disease data set," *School Inf. Comput. Sci., Univ. California, Irvine, CA, USA*, 1988.

[12] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE transactions on evolutionary computation*, vol. 1, no. 1, pp. 67–82, 1997.

[13] T. Saito and M. Rehmsmeier, "The precision-recall plot is more informative than the roc plot when evaluating binary classifiers on imbalanced datasets," *PloS one*, vol. 10, no. 3, p. e0118432, 2015.

[14] J. Pennington, S. Schoenholz, and S. Ganguli, "Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice," in *Advances in neural information processing systems*, 2017, pp. 4785–4795.

[15] Kaggle, "Heart Disease UCI," https://www.kaggle.com/ronitf/heart-disease-uci/notebooks, 2018, online.

[16] F. Sayah, "Predicting heart disease using machine learning," https://www.kaggle.com/faressayah/predicting-heart-disease-using-machine-learning, 2020, online.

[17] B. Siyah, "Heart Disease Prediction using Neural Networks," https://www.kaggle.com/bulentsiyah/heart-disease-prediction-using-neural-networks, 2020, online.