ISSN 2277-3061

Performance Analysis of Rule Based Algorithms Applied to a Cardiovascular Dataset

Dev Mukherji, Nikita Padalia

B.Tech Computer Science & Engineering, VIT University, Vellore – 632014, TN, India

devmukherji2009@vit.ac.in

npadalia14@gmail.com

ABSTRACT

Cardiovascular disease is one of the dominant concerns of society, affecting millions of people each year. Early and accurate diagnosis of risk of heart disease is one of major areas of medical research, aimed to aid in its prevention and treatment. Most of the approaches used to predict the occurrence of heart disease use single data mining techniques. However, performances of predictive methods have recently increased upon research into hybrid and alternative methods. This paper analyses the performance of logistic regression, support vector machine, and decision trees along with rule-based hybrids of the three in an attempt to create a more accurate predictive model.

Indexing terms/Keywords

Heart disease; cardiovascular; rule based; support vector machines; logistic regression; decision trees; hybrid mining.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 12, No.2

editor@cirworld.com

www.cirworld.com, member.cirworld.com

INTRODUCTION

Heart disease has long been a research topic of discussion. Most significant contributions to accurate predictive models of the disease stem from the use of a database created by Dr. Robert Detrano [1] of the Cleveland Clinic in 1989. Almost all predictive methods have been tested on the database over the past 14 years, such as Naïve Bayes, regression, neural networks, and decision trees. The performance of these ranges between higher 70's to mid-80's percentage of accuracy. However in the past 6 years, with the development of hybrid data mining techniques, these thresholds have increased to higher 80's. By combining the homogenous techniques, there is a trend of increased accuracy.

Three of many predictive techniques are support vector machines, decision trees, and logistic regression. A support vector machine finds the best hyperplane with the largest margin that separates all the data points amongst two classes. Decision tree is a tool which creates a path by splitting attributes in order to create leaf nodes which classify the data. Logistic regression is a type of regression that is used when the attributes in the data set are categorical. In this paper, all three methods are tested individually and then combined using a rule based algorithm. This is then tested and compared in efficacy.

Mythili et al. [2] proposed a model for using a rule based algorithm wherein the results of all three methods are permuted and compared to each other. The same paper also presents a literature survey that broadly covers a comparison of most of the predictive methods carried out on the Cleveland Heart Clinic Database (CHDD). The model is briefly discussed in the framework section, followed by the results gathered from the implementation of the framework.

FRAMEWORK

The framework presented in Figure [1] was introduced in Mythili et al. [2] as a pretext to this paper. It has therefore been discussed in detail previously. This paper presents a summary of the database details for the benefit of the reader.

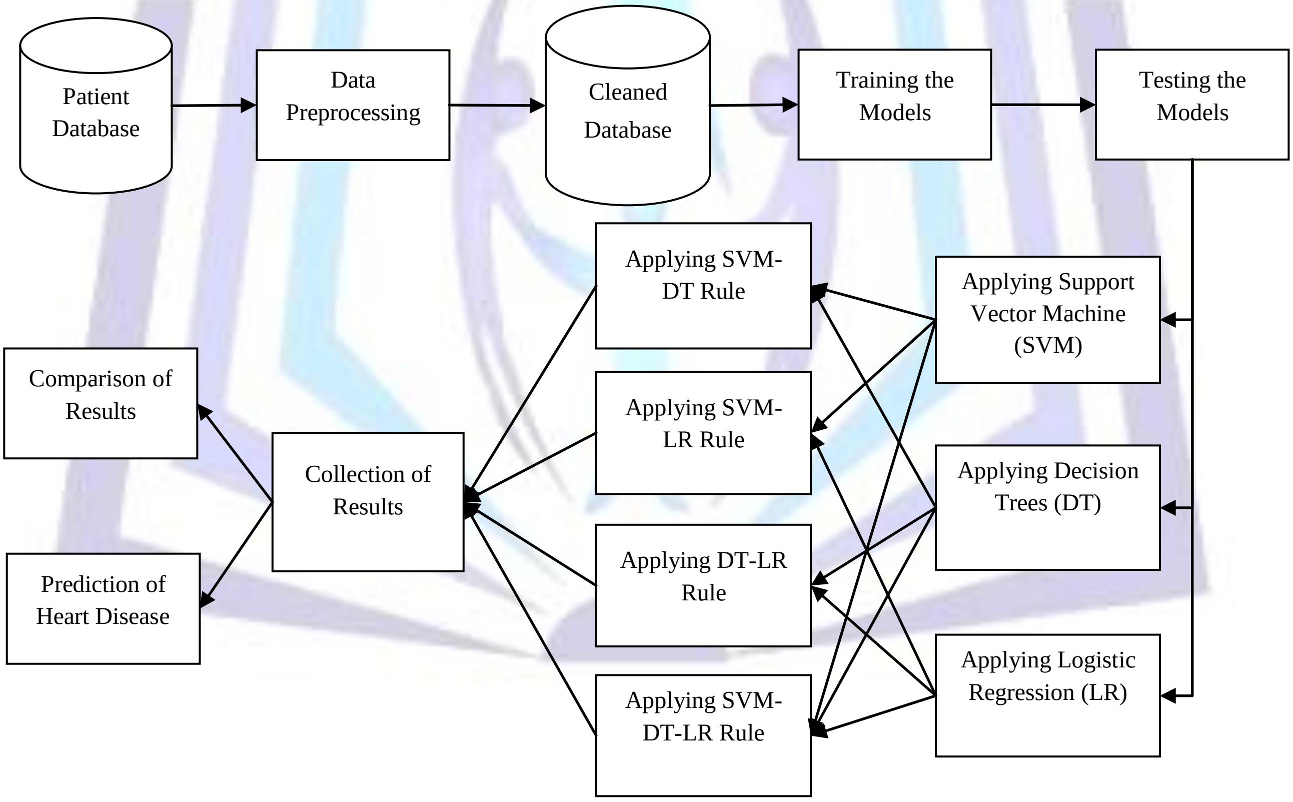

Figure1: Proposed rule-based framework

The dataset used for testing is the Cleveland Heart Disease Database [1] found on the University of California at Irvine (UCI) Machine Learning Repository. There are a total of 14 attributes including one concept class that are considered, as seen in Table [1].This data can be used by one of more data mining techniques to help us develop profiles for differentiating individuals with heart disease from those without known heart conditions.

Table 1. List of Attributes in the Cleveland Heart Disease Database

No.	Attribute	Description/Type	Values
1.	Age	Numeric (in years)	Any
2.	Sex	Male, Female	1, 0
3.	Chest Pain Type	Typical Angina, Atypical Angina, Non Angina Pain, Asymptomatic	1, 2, 3, 4
4.	Blood Pressure	Numeric (in mm Hg)	Any
5.	Cholesterol	Numeric (in mg/dl)	Any
6.	Fasting Blood Sugar<120	True, False	0, 1
7.	Resting ECG	Normal, ST-T Wave Abnormality, Showing signs of Left Ventricular Hypertrophy by Estes' Criteria	0, 1, 2
8.	Maximum Heart Rate	Numeric (in BPM)	Any
9.	Exercise Induced Angina	True, False	1, 0
10.	Old Peak	ST depression induced by exercise relative to rest	Any
11.	Slope	Up, Flat, Down	1, 2, 3
12.	Number of Colored Vessels by Fluoroscopy	Numerical	0, 1, 2, 3
13.	Thal	Normal, Fixed Defect, Reversible Defect	3, 6, 7
14.	Concept Class	Healthy, Sick	0, 1

The preprocessing of data involves the extraction of it from the original database such that the output format is uniform. This is done through transforming the data, removal of missing data, normalization of data, and removal of outliers.

The original database has 303 tuples in total, out of which 297 are complete and the remaining 6 have missing values. The 6 tuples with missing values are removed as this data counts for less than 2% of the entire dataset. Normalization of the data is done on a need-basis based on the method employed during testing. Additionally, training is done using K-fold cross validation when it is required, so that all possible tuples are used are the result is of increased consistency.

IMPLEMENTATION

Support Vector Machine

A support vector machine (Cortes et al., 1995 [3]) is a binary classifier learning model which is used when your data has exactly two concept classes (Eg: Has heart disease vs. does not have heart disease). It can be used on both linear and nonlinear data through either classification or regression analysis.

SVM is a complex tool that works well when there are many attributes in the data set, even if there are few tuples to work with. It is a kernel based algorithm, which means that it maps all the data points on a high dimensional space (also known as a hyperplane) thus creating 'support vectors'. The distances between these support vectors create a margin separating both the concept classes, and points during testing are then mapped to either side of the margin, thereby classifying it. The complexity of SVM also keeps the chances of errors to a minimum while creating and mapping on the margin.

SVM has high prediction accuracy and fits data points fairly quickly, but this is largely dependent on how many support vectors exist. The more support vectors there are, the more memory is consumed and the slower the rate of the operation.

Algorithms like the sequential minimal optimization, quadratic programming, or least square method must first be used to train the SVM. There are multiple kernel spaces that can be used when creating the hyperplane. Depending on the kernel function, it can be difficult to interpret how SVM classifies data, though the default linear scheme is easy to interpret.

There are five commonly used kernel spaces – linear (uses dot product), quadratic, polynomial, Gaussian Radial Basis Function kernel with a default scaling factor of sigma = 1, and Multilayer Perceptron kernel with a default scale of -1 to 1.

This paper implemented a 10 fold cross validation for training. The 297 samples were divided into 9 groups of 30 and one group of 27 tuples. The model was trained using each of these groups exactly once. The results from each of these were then averaged to produce one performance estimation.

One difficult part about using SVM is the selection of a kernel. While it was possible to have used a combination of all the common kernels with all the training algorithms, it would be hard to determine which results are over-fitted and which ones are sub optimal. A good rule of thumb is to use the Gaussian kernel when the problem is nonlinear since it has been proven that the linear kernel is a degenerate version of RBF (Keerthi& Lin [4]). Sometimes, as it is a special case of the RBF Kernel, the linear kernel is a better fit when there are an extremely large number of attributes (Hsu et al. [5]). However, since the CHDD has only 297 tuples, the sequential minimization optimization algorithm was implemented with the Gaussian RBF kernel for this paper. Quadratic programming as a training method is a better choice over SMO only when the dataset is significantly larger since it has a more efficient time complexity than the SMO and LS. For a dataset such as the CHDD however, SMO is a good option.

Decision Trees

A decision tree, also known as Classification and Regression Trees (Breiman et al., [6]), is a tool that uses decisions at nodes to follow a path that leads to a decision. A standard decision tree is made up of one root node, at least two branches and two leaf nodes. Certain rules are obtained by traversing the tree-like structure and these are used to predict responses to data. While classification trees give binary results, regression trees give numeric ones.

A classification tree is created by splitting attributes into groups to create two or more leaf nodes. How we split the node is considered based on a certain criteria called the Splitting Criterion. The higher the ‘purity’ of the subsets of a split, the better the node will be at making a correct decision. This measure of purity is done using splitting criterion such as the Gini Coefficient, the Twoing Rule, and the Deviance rule.

In this study, the measure of Gini index (Gini, C. et al., [7]) is used as the splitting criterion. Similar to SVM, the data was trained using batches of 267 out of 297 tuples and tested with the remaining 30 tuples. This was done once for each set of testing tuples and the average of the individual performances was taken as the single measure of accuracy.

The most influential factors according to the Gini Index are: Thal, Testbps/CA, Chol/Age/Slope/Oldpeak.

The algorithm generated after calculating the Gini Impurities by using a modified version of the MATLAB function is as below.

```
1  ifthal<4.5 then node 2 elseifthal>=4.5 then node 3 else No
2  ifca<0 then node 4 elseifca>=0 then node 5 else No
3  ifcp<3.5 then node 6 elseifcp>=3.5 then node 7 else Yes
4  iftrestbps<157 then node 8 elseiftrestbps>=157 then node 9 else No
5  ifcp<3.5 then node 10 elseifcp>=3.5 then node 11 else No
6  ifca<0 then node 12 elseifca>=0 then node 13 else No
7  ifoldpeak<0.45 then node 14 elseifoldpeak>=0.45 then node 15 else Yes
8  if age<59.5 then node 16 else if age>=59.5 then node 17 else No
9  class = Yes
10  ifchol<237.5 then node 18 elseifchol>=237.5 then node 19 else No
11  class = Yes
12  ifchol<207.5 then node 20 elseifchol>=207.5 then node 21 else No
13  if slope<1.5 then node 22 elseif slope>=1.5 then node 23 else Yes
14  ifthal<149 then node 24 elseifthal>=149 then node 25 else Yes
15  ifoldpeak<0.55 then node 26 elseifoldpeak>=0.55 then node 27 else Yes
16  ifoldpeak<3.55 then node 28 elseifoldpeak>=3.55 then node 29 else No
17  ifcp<3.5 then node 30 elseifcp>=3.5 then node 31 else No
18  if age<55.5 then node 32 elseif age>=55.5 then node 33 else No
19  class = No
20  class = No
21  ifthal<132.5 then node 34 elseifthal>=132.5 then node 35 else No
22  class = No
```

23 class = Yes

24 class = No

25 class = Yes

26 class = Yes

27 ifthal<6.5 then node 36 elseifthal>=6.5 then node 37 else Yes

28 class = No

29 class = Yes

30 ifchol<302 then node 38 elseifchol>=302 then node 39 else No

31 class = Yes

32 class = No

33 class = Yes

34 class = Yes

35 ifchol<231.5 then node 40 elseifchol>=231.5 then node 41 else No

36 if age<65.5 then node 42 elseif age>=65.5 then node 43 else Yes

37 class = Yes

38 ifoldpeak<2.8 then node 44 elseifoldpeak>=2.8 then node 45 else No

39 class = Yes

40 class = Yes

41 iftrestbps<161 then node 46 elseiftrestbps>=161 then node 47 else No

42 class = Yes

43 class = No

44 class = No

45 class = Yes

46 class = No

The corresponding tree for the above algorithm is presented in Figure 2.

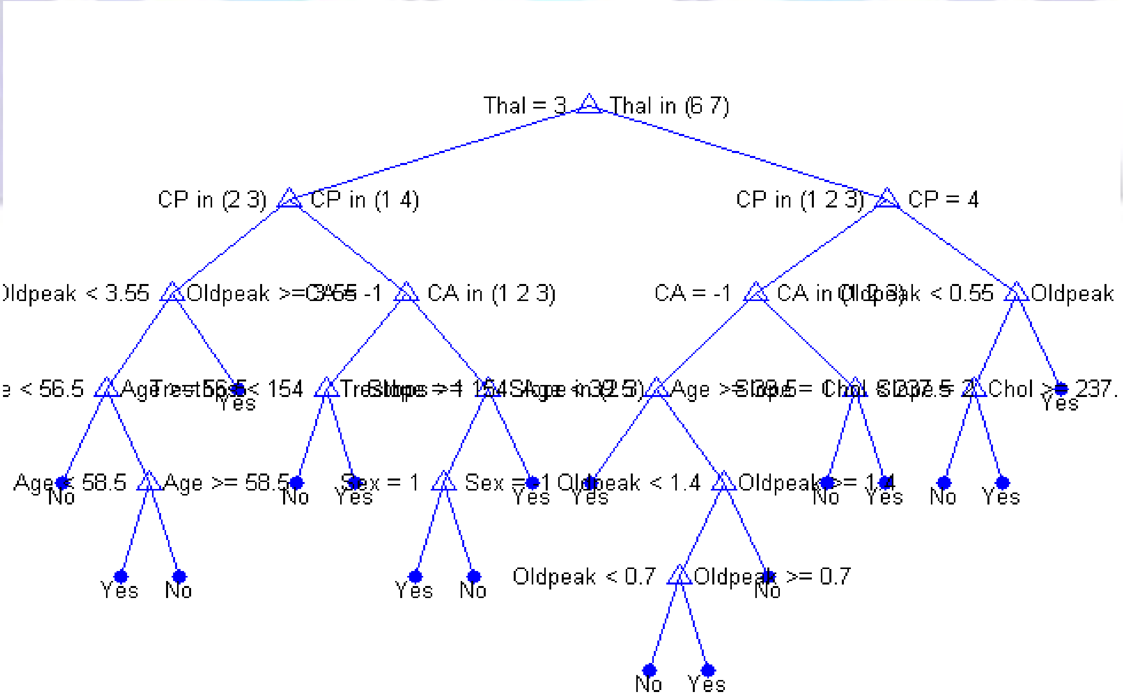

Figure 2: Tree generated by the calculation of Gini Impurities

Logistic Regression

Regression is a type of mathematical model that predicts the outcome of a variable based on information from other variables. A logistic regression model helps to predict a discrete or categorical outcome. The input set of variables may be continuous, discrete, or a mix of the two.

Logistic regression helps in the prediction of a group membership by calculating the chance of success over failure, thus giving results in the form of a ratio. It also gives an insight into the correlation and the strengths of the variables.

At first, the Hosmer-Lemeshow test for goodness fit was performed to check the model for usefulness. This resulted in a Chi-Square value of 3.929 with a corresponding p-value of 0.863 (p>0.05). Since the chi square value is not significant, it showed that the model is a good fit.

The variables and their corresponding regression coefficients and level of significances are shown in Table 2.

Table 2: Data Calculated for Logistic Regression

Variables	β	P-value	Significance	Exp(β)
Age(x1)	-.014	.555	NS	.986
Sex(x2)	1.312	.007	S	3.714
Cp(x3)	.576	.003	S	1.779
Trestbps(x4)	.024	.025	S	1.024
Chol(x5)	.005	.186	NS	1.005
Fbs(x6)	-1.022	.066	NS	.360
Restecg(x7)	.245	.185	NS	1.278
Thalach(x8)	-.021	.043	S	.980
Exang(x9)	.926	.025	S	2.525
Oldpeak(x10)	.247	.243	NS	1.281
Slope(x11)	.570	.116	NS	1.768
Ca(x12)	1.268	.000	S	3.553
Thal(x13)	.344	.001	S	1.410
Concept Class	-7.372	.010	S	.001

After finding the regression co-efficient and the intercept terms, one can find the Binary Logistic Regression Equation

$$Y = \text{ß}_0 + \text{ß}_1 X_1 + \text{ß}_2 X_2 + \cdots \text{ß}_n X_n$$

where P(Y = 1) is probability of presence of heart disease and β_0, β_1, β_2,..., βn are regression coefficients.

For any given individual if x_1, x_2, …x_n are given, one can find the value of Y and e^Y. From that, the odds ratio has been obtained as follows,

$$\frac{P}{1-P} = e^Y$$

From the above equation, the value of P can be found which gives us the probability that of the tuple having heart disease. For this paper, logistic regression was executed in IBM SPSS.

Rule Based Algorithm

Rule based algorithms were implemented based on a simple majority for four cases, namely for a combination of LR-DT-SVM, LR-DT, DT-SVM, and LR-SVM. PRED is the keyword used for the attribute in which the prediction of the rule is stored. If the chance of PRED is 50%, we cannot use simple majority, and therefore select either of the classes arbitrarily as choosing either will not change the accuracy. As always, the algorithms were generated using a subset of the dataset and tested on the remainder of the dataset. Below are the algorithms for the combinations of each of the combinations:

SVM-DT

1. If (SVM=1) && (DT=1)

Then PRED=1 (91.6%)

2. If (SVM=0) && (DT=0)

Then PRED= 1 (10.5%)

3. If (SVM=1) && (DT=0)

Then PRED=1 (50%)

4. If (SVM=0) && (DT=1)

Then PRED=1 (78.5%)

DT-LR

1. If (DT=1) && (LR=1)

Then PRED=1

2. If (DT=1) && (LR=0)

Then PRED=1 (66.6%)

3. If (DT=0) && (LR=0)

Then PRED=1 (15%)

4. If (DT=0) && (LR=1)

Then PRED=1 (0%)

SVM-LR

1. If (SVM=1) && (LR=1)

Then PRED=1

2. If (SVM=1) && (LR=0)

Then PRED=1 (33%)

3. If (SVM=0) && (LR=0)

Then PRED=1 (34.4%)

4. If (SVM=0) && (LR=1)

Then PRED=1 (75%)

LR-DT-SVM

1. If (SVM=1) && (DT=1) && (LR=1)

Then PRED=1

2. If (SVM=1) && (DT=1) && (LR=0)

Then PRED= 1 (0%)

3. If (SVM=1) && (DT=0) && (LR=1)

Then PRED does not exist.

4. If (SVM=1) && (DT=0) && (LR=0)

Then PRED=1= 50%

5. If (SVM=0) && (DT=1) && (LR=1)

Then PRED=1

6. If (SVM=0) && (DT=0) && (LR=1)

Then PRED = 1 (0%)

7. If (SVM=0) && (DT=1) && (LR=0)

Then PRED=1 (72%)

8. If (SVM=0) && (DT=0) && (LR=0)

Then PRED= 1 (5.5%)

RESULTS

Support Vector Machine

The Confusion Matrix for the data is (136, 26), (24,111), (0,0) where the columns are the actual values, the rows are the predicted values, and the last row has the unclassified instances.

Table 3: Results of Support Vector Machine

Accuracy	80.81%
Sensitivity	77.50%
Specificity	84.67%

The total CPU time taken to execute the function was 0.732s on a computer with two cores with four logical processors. The time has been mentioned only for the benefit of a relative comparison of the methods for the reader. This may vary depending on the processing environment.

Decision Trees

The Table 4 below summarizes the results of the Decision Tree generated using the Gini Coefficient as a Split Criterion.

Table 4: Results of Implementation of Decision Trees

	ACTUAL	GINI
YES	27	22
NO	20	25
ACCURACY		80.85%
TRUE POSITIVE		20
TRUE NEGATIVE		18
FALSE NEGATIVE		7
FALSE POSITIVE		2
SPECIFICITY		74.07%
SENSITIVITY		90%

Logistic Regression

The Table 5 below summarizes the results of Logistic Regression.

Table 5: Results of Implementation of Logistic Regression

Observed		Predicted		
		Concept Class		Percentage Correct
		0	1	
Concept Class	0	140	20	87.5
	1	25	112	81.8
Overall Percentage				84.8

Rule Based Algorithm

The Table 6 below summarizes the results of the rule based algorithmic approach.

Table 6: Results of Rule Based Implementation

	ACTUAL	SVM-LR-DT	SVM-LR	SVM-DT	LR-DT
YES	25	30	15	31	26
NO	22	17	27	16	21
TRUE POSITIVE		25	14	25	22
TRUE NEGATIVE		17	21	16	18
FALSE NEGATIVE		0	11	0	3
FALSE POSITIVE		5	1	6	4
SPECIFICITY		77.27%	95.40%	72.72%	81.81%
SENSITIVITY		100%	56%	100%	88%
ACCURACY		89.30%	74.40%	87.20%	85.10%

Performance Comparison

Table 7: Accuracies, Specificities, and Sensitivities of Different Methods Employed

	LR	DT	SVM	LR-DT-SVM	LR-DT	SVM-DT	LR-SVM
Accuracy	84.80%	80.85%	80.81%	89.30%	85.10%	87.20%	74.40%
Specificity	87.50%	74.07%	84.67%	77.27%	81.81%	72.72%	95.40%
Sensitivity	81.75%	90%	77.50%	100%	88.00%	100%	56%

CONCLUSION

In summary, multiple algorithms in Support Vector Machines, multiple in Decision Trees, and the Logistic Regression algorithm itself have been carried out. In SVM, the model has been implemented using Sequential Minimization through the Gaussian kernel. This gave an accuracy of 80.81%. Decision trees has been implemented using the Gini index and gave an accuracy of 80.85%. Finally, the logistic regression model gave an accuracy of 84.80%. After using the rule based algorithm we get the accuracies for SVM-DT-LR as 89.30%, SVM-LR as 74.40%, SVM-DT as 87.20%, and LR-DT as 85.10%.

There are a few rudimentary conclusions that can be drawn from these results. Firstly, we may presume that the decision tree created on this database using the Gini Index is overfitted due to its high sensitivity. It would be prudent to follow a different approach, perhaps by using the Deviance rule. Another viable option would be to disregard the attribute set and develop a decision tree based on recommendations from domain experts. However, that would largely be dependent on opinion and would have no basis for fact.

Secondly, if the first is true and the Decision Tree model is indeed overfitted, then the only rule based algorithm that holds is the LR-SVM one, which seems to perform regressively from its individual counterparts in terms of accuracy and sensitivity.

Although Decision Trees are known to often be overfit, if it is not overfitted by a considerable amount, then the results of the rule-based algorithm improve the accuracy on almost all accounts. In fact, for the LR-DT-SVM and the SVM-DT models, the sensitivity is unusually high leading to the belief that people who actually have or do not have the disease may be correctly diagnosed almost every time. In the end, we leave the reader to draw certain conclusions of their own on the results.

For further work, one may take up a more efficient model than the ones we have chosen above. The SVM model with SMO and Gaussian Kernel seems to be of good fit but there may be alternatives to the other two models. One may also consider using a hybrid data mining technique such as the Bagging Algorithm or K-Nearest-Neighbor as the predictive models as hybrid techniques have been indicative of better results (Shouman et al.[8]).

ACKNOWLEDGMENTS

Our thanks to Prof. T. Mythili of VIT University for being our mentor along the way without whose advice, guidance, and constant support, we would not have direction on how to proceed with and accomplish this task; to Dr. Praveen Kumar, withouth whom we would not have the depth of understanding about the cardiovascular domain as we do now; and a special thanks to Abhiram Naidu, graduate student at George Washington University, our friend and classmate whose support and helping hand along the way made the project possible.

REFERENCES

[1] Robert Detrano “Cleveland Heart Disease Database” V.A. Medical Center, Long Beach and Cleveland Clinic Foundation, 1989.

[2] Mythili, Mukherji, Padalia, Naidu.“A Heart Disease Prediction Model using SVM-Decision Trees-Logistic Regression (SDL)”, International Journal of Computer Applications, April 2013.

[3] Cortes, Corinna; and Vapnik, Vladimir N.; "Support-Vector Networks", Machine Learning, 20, 1995.

[4] Keerthi& Lin, “Asymptotic Behaviors of Support Vector Machines with Gaussian Kernel”, 2003.

[5] Hsu et al., “A Practical Guide to Support Vector Classification”, 2010.

[6] Breiman, Leo, Jerome Friedman, R. Olshen and C. Stone (1984). Classification and Regression Trees. Belmont, California: Wadsworth.

[7] Gini, C. 1912 “Italian: Variabilità e mutabilità" (Variability and Mutability', C. Cuppini, Bologna,Reprinted in Memorie di metodologicastatistica (Ed. Pizetti E, Salvemini, T). Rome: LibreriaErediVirgilioVeschi 1955.

[8] Shouman et al., “Using Data Mining Techniques in Heart Disease Diagnosis and Treatment”, Japan-Egypt Conference on Electronics, Communications, and Computers, 2012.