GA-NN APPROACH FOR ECG FEATURE SELECTION IN RULE-BASED ARRHYTHMIA CLASSIFICATION

by

Gözde Aslantaş

B.S., Computer Engineering, Ege University, 2008

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Computer Engineering

Boğaziçi University

2012

GA-NN APPROACH FOR ECG FEATURE SELECTION IN RULE-BASED ARRHYTHMIA CLASSIFICATION

APPROVED BY:

Prof. Fikret Gürgen .....
(Thesis Supervisor)

Assoc. Prof. Cengizhan Öztürk .....

Assist. Prof. Albert Ali Salah .....

DATE OF APPROVAL: 19.09.2012

ACKNOWLEDGEMENTS

I would like to thank to my supervisor, Prof. Fikret Gürgen for his support and encouragement. His guidances and feedbacks were precious during my thesis.

I also would like to thank to Assist. Prof. Albert Ali Salah for his comments which were helpful in progress of my study.

I want to express my appreciation to Fikret Küçükdeveci from TEPA. He shared a part of their datasets with me and gave me chance to verify my study with real ECG data.

I would like to thank my thesis examining committee, Assist. Prof. Albert Ali Salah and Assoc. Prof. Cengizhan Öztürk...

Finally, I want to express my gratitude to my parents to whom I owe all my success in my entire life, for their endless support and encouragement. Their belief in me was very important to accomplish this thesis.

ABSTRACT

GA-NN APPROACH FOR ECG FEATURE SELECTION IN RULE-BASED ARRHYTHMIA CLASSIFICATION

This paper presents rule extraction and feature selection system to detect abnormality in ECG signals. Genetic Algorithm-Neural Network (GA-NN) Approach is used to distinguish between the presence and absence of cardiac arrhythmia and perform feature selection. Following this process, rule sets are extracted in order to guide the diagnosis of cardiac arrhythmia. The rule sets are extracted based on selected features because rule extraction without feature selection may result in rules to be more complex than human may realize. C4.5, RIPPER, PART and HotSpot methods are used to perform rule extraction. The ECG dataset used in this study is obtained from UCI Arrhythmia Database. In this dataset, all the anomalies are grouped into one abnormal class and the rest is grouped into one normal class. As a comparison, k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Naive Bayes and Bayesian Networks have been tested on the arrhythmia dataset. For dimensionality reduction purpose, recursive feature extractor (RFE-SVM), correlation based feature selection (CFS), principal component analysis (PCA) and factor analysis (FA) have been applied. According to test results, GA-NN outperforms other techniques.

ÖZET

KURAL TABANLI ARİTMİ SINIFLANDIRMASINDA EKG ÖZNİTELİK SEÇİMİNE GA-YSA YAKLAŞIMI

Bu çalışma, EKG sinyallerindeki anormalliklerin tespiti için kural çıkarım ve öznitelik seçim sistemi sunar. Kardiyak aritminin varlığı ve yokluğu arasındaki ayrımın yapılması ve öznitelik seçimi için Genetik Algoritması-Yapay Sinir Ağı (GA-YSA) Yaklaşımı kullanılmıştır. Bu süreci takiben, aritmi teşhisine rehberlik etmesi amacıyla kural seti çıkarılmıştır. Bu kural setleri, seçilmiş özniteliklere dayanılarak çıkarılmaktadır; çünkü öznitelik seçimi olmadan yapılan kural çıkarımı insanın anlayabileğinden daha karmaşık sonuçlar üretebilmektedir. Kural çıkarımında, C4.5, RIPPER, PART ve HotSpot metotları uygulanmıştır. Bu çalışmada kullanılan EKG veri kümesi, UCI Aritmi Veritabanı'ndan elde edilmektedir. Bu veri kümesindeki tüm anomaliler, tek bir anormal sınıf altında toplanırken geriye kalanlar normal sınıfı oluşturacak şekilde düzenlenmiştir. Karşılaştırma amacıyla k en yakın komşu (k-NN), Destek Vektör Makineleri (SVM), Naive Bayes ve Bayes Ağları, EKG veri kümesine uygulanmıştır. Boyut indirgemek amacıyla ise, özyineli nitelik çıkarımı (RFE-SVM), korelasyon tabanlı nitelik seçimi (CFS), temel bileşen analizi (PCA) ve faktör analizi (FA) yöntemleri kullanılmıştır. Deney sonuçlarına göre, GA-YSA yöntemi, diğerlerinden daha iyi sonuç vermiştir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS/ABBREVIATIONS

AHA	American Hospital Association
ANN	Artificial Neural Network
BP	Backpropagation Algorithm
C4.5	C4.5 Decision Tree
CFS	Correlation Based Feature Selection
DT	Decision Tree
FA	Factor Analysis
FN	False Negative
FP	False Positive
GA	Genetic Algorithm
GDA	Generalized Discriminant Analysis
k-NN	k-Nearest Neighbors
LDA	Linear Discriminant Analysis
MCC	Mathew's Correlation Coefficients
MIT-BIH	The MIT Arrhythmia Dataset
MLP	Multi Layer Perceptron
NN	Neural Network
PART	Partial Decision Tree
PCA	Principal Component Analysis
RFE-SVM	Recursive Feature Elimination with Support Vector Machine
RIPPER	Repeated Incremental Pruning to Produce Error Reduction
SVM	Support Vector Machine
TN	True Negative
TP	True positive
UCI	The UCI Arrhythmia Dataset
VALE	The VALidation-Ecg

1. INTRODUCTION

The heart has a unique place among other muscles in terms of being capable of automatic rhythmic contraction. The impulses that stimulate mascular contraction arise in the conduction system of the heart. The conduction system consists of the sino-atrial (SA) node, the internodal atrial pathways, the atrioventricular (AV) node, the bundle of His, the right and the left bundle branches, and the Purkinje system. Electrical currents which spread through the entire body is produced by the formulation and conduction of the impulse. SA node is known as the cardiac pacemaker because heart beat is controlled by the rhythmic impulses which originate from SA node. The impulse passes through the internodal atrial pathways to depolarize the atria to pump the blood to ventricles and reaches the AV node. Then the impulse passes on to the bundle of His, the right and left bundles, the branches of the Purkinje system into the ventricles. This part comprises the first phase of the cardiac cycle, namely *diastole*. The electrical signal, then traverses through the ventricles causing them to contract. Blood is pumped to the lungs from the right ventricle and to the rest of the body from the left ventricle. This part is the second phase of the cardiac cycle which is called *systole*. The conduction system is given in Figure 1.1.

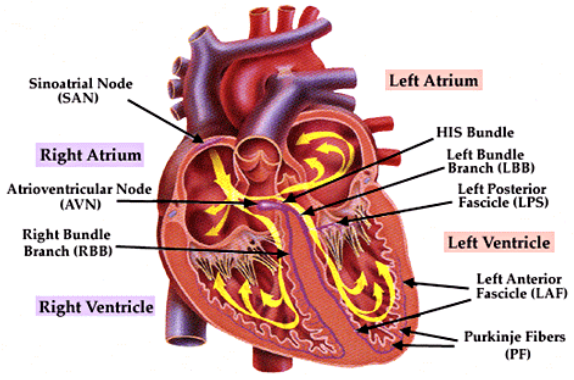

Figure 1.1. Cardiac conduction system [1].

Electrocardiogram (ECG) constitutes recording of the heart's electrical activity over time. Electrodes are applied to specific positions on the skin. Electrical changes

detected by the electrodes are transmitted to an electrocardiograph which processes the changes and outputs on a scaled paper. The position diagram of electrodes is given in Figure 1.2.

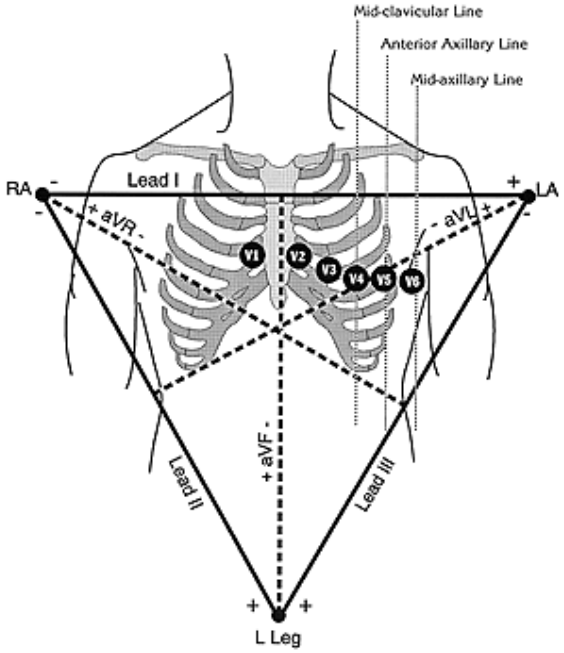

Figure 1.2. Position diagram of electrodes [2].

The standard 12-lead ECG records with 3, 6 or 12 leads simultaneously. Each lead represents a particular orientation in space as indicated in Table 1.1.

The plot produced by ECG consists of waves which are referred as letters: P, Q, R, S, T, U. Scaled paper has horizontal and vertical lines. Time is measured along the horizontal lines and voltage is measured along the vertical lines. Electrical activity during the cycle is plotted with waves. For example, P wave indicates atrial depolarization, QRS complex is a ventricular depolarization, and T wave is a ventricular repolarization. Waves on a normal plot is given in Figure 1.3.

Table 1.1. Orientation of 12 Lead ECG.

Lead Name	**Orientation**
Lead 1	RA(-) to LA(+)
Lead 2	RA(-) to LF(+)
Lead 3	LA(-) to LF(+)
Lead aVR	RA(+) to [LA and LF](-)
Lead aVL	LA(+) to [RA and LF](-)
Lead aVF	LF(+) to [RA and LA](-)
Lead V1	V1(+) to WCT
Lead V2	V2(+) to WCT
Lead V3	V3(+) to WCT
Lead V4	V4(+) to WCT
Lead V5	V5(+) to WCT
Lead V6	V6(+) to WCT
	WCT: Wilson's Central Terminal is the average of the three limp leads (aVR, AVL, AVF).

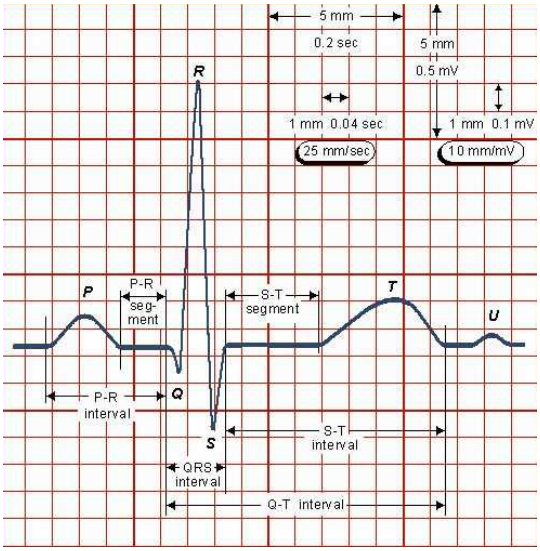

Figure 1.3. Waves on a normal ECG plot [3].

1.1. Motivation

Alterations that disrupt the regular functioning of the heart cycle may cause a cardiac arrhythmia. These anomalies are worth considering in terms of being a potential reason for a heart disease. Thus, early detection of heart disease can save lives.

ECG output is just a laboratory test result and must be integrated with a clinical assessment to become more meaningful. For this reason, ECG analysis is a part of medical evaluation. ECG analysis requires to know the meaning of the waves, complexes, intervals, amplitudes and regularity of the waves. Knowledge in normal ECG patterns can be a key to detect abnormality in ECG signals.

P wave indicates atrial depolarization and duration of the P wave should not exceed 0.12 s. for normal rhythm.

P-R interval includes the time required for atrial depolarization and the onset of ventricular depolarization. This interval is measured from the onset of the P wave to the beginning of the QRS complex. The normal value is in the range of 0.12-0.20 s. Longer P-R intervals are seen in the cases of AV block while shorter ones in different arrhythmias.

QRS complex indicates the total ventricular depolarization time which is measured from the onset of the Q wave to the offset of the S wave. The upper limit for normal is 0.10 s. Heart rate can be computed from two successive QRS complexes or R-R interval. Normally, heart rate ranges from 60 to 90 beats per minute.

S-T segment is the duration between the end of ventricular depolarization which is indicated by QRS complex and the beginning of the T wave.

T wave indicates the ventricular repolarization.

Q-T interval represents the duration of electrical systole and it is measured from the onset of the Q wave to the end of the T wave.

U wave is a deflection which follows the T wave preceding the next P wave and usually shows the same polarity as the T wave.

The order of the waves is as P-QRS-T-U. An extra or missing wave may indicate an anomaly.

As well as the knowledge and experience, ECG analysis requires full attention. Computer-aided ECG analysis is important because of the lack of such human needs and being a quick advisor. Researches in this area try to find more robust and reliable methods for automated ECG analysis that detect abnormal patterns.

This research aims to classify arrhythmic patterns in a rule-based manner, and thus intends to aid the cardiologists.

1.2. Previous Work

In the literature, there are many researches that address arrhythmia classification problem. The purpose of such studies is to find an accurate and reliable classification method that can contribute to automated ECG analysis.

UCI Arrhythmia Repository [4] and MIT-BIH [5] are two most commonly used databases for ECG Analysis. Some researches that use these databases are summarized in Table 1.2.

MIT-BIH database contains ECG signal data. Therefore, characteristic wave detection in ECG signal is an important preprocessing step before the classification. Pan and Tompkins [22] developed an algorithm to detect QRS complexes. Algorithm steps include digital bandpass filtering to reduce noise, differentiation, squaring and moving window integration. The optimized implementation of QRS detection in C language

Table 1.2. Previous work using various data.

Ref.	**Year**	**Protocol**	**Algorithm**	**Results**	**Dataset**
Yeap *et al.* [6]	1990	5% tra. 5% test. and 100% test.	ANN BP	98.36% sens. 67.80% sens.	AHA
Hu *et al.* [7]	1993	3-fold CV	51-25-2 MLP	≈90% acc.	MIT-BIH
Silipo and Marchesi [8]	1998	15% tra. 85% test.	ANN with BP	≈90% sens.	MIT-BIH and VALE
Chazal and Reilly [9]	2000	10-fold CV	LDA	69.3-74.7% acc.	Frank Lead ECG Data [10]
Gao *et al.* [11]	2005	t-test	Bayesian ANN	≈76% sens.	UCI
Niwas *et al.* [12]	2005	58% tra.42% test.	ANN	≈99% acc.	MIT-BIH
Zhang and Zhang [13]	2005	2/3 tra.1/3 test.	PCA-SVM	≈99% acc.	MIT-BIH
Song *et al.* [14]	2005	CV	LDA-SVM	99.35% avg. acc.	MIT-BIH
Uyar [15]	2006	10-fold CV	PCA-SVM	83.7% acc.	UCI
Kara and Okandan [16]	2007	24 of 72 normal - 28 of 52 AF signals test.	ANN BP	100% acc	MIT-BIH
Asl *et al.* [17]	2008	2/3tra. 1/3 test.	GDA-SVM	≈99% acc.	MIT-BIH
Oliveira *et al.* [18]	2010	75% tra. 25% test.	Bayesian Networks	≈99% sens.	MIT-BIH and QT database
Ozcan [19]	2010	10-fold CV	FSVM	85.71% acc.	UCI and Real ECG
Jadhav *et al.* [20]	2011	90% tra. 10% test.	ANN BP	86.67% sens.	UCI
Homaeinezhad *et al.* [21]	2012	4035 beats tra. 3150 beats test.	SVM-KNN-four MLP-BP (Neuro-SVM-KNN)	98.06% acc.	MIT-BIH

is presented in [23]. In the study by Niwas *et al.* [12], parameters like heartbeat intervals, RR intervals are extracted using ECG filtering method. Asl *et al.* [17] applied QRS detection algorithm [22, 23]. Wavelet transform analysis is also widely used to detect P wave, QRS complex and T wave in ECG signal. Song *et al.* [14] utilized the wavelet transform based method in detection of the QRS complex which is proposed by Park *et al.* [24]. In [24], a wavelet adaptive filter is undertaken to minimize the distortion of the S-T segment. Kadambe *et al.* [25] present a dyadic wavelet transform based QRS detector which computes local peaks across two successive dyadic scales and determines the presence of the QRS complex. In a subsequent research by Park *et al.* [26], a wavelet interpolation filter is designed for the removal of motion artifacts in the S-T segment of stress ECGs. Homaeinezhad *et al.* [21] applied an ECG detection and delineation method implemented by Ghaffari *et al.* [27] to find out P wave, QRS complex and T wave. Based on wavelet transform, Lin *et al.* [28] used Morlet wavelet to extract features. Benitez *et al.* [29] present Hilbert based transform with an average detection rate of 99%. Kara and Okandan [16] extracted features using wavelet coefficients and Welch method [30]. Hu *et al.* [7] investigate applications of artificial neural network in both QRS detection and beat classification. A two layer MLP ANN is used for QRS complex detection. They compared detection rates of the proposed method with the linear adaptive filtering [31] and bandpass filtering [22]. In the study by Oliviera *et al.* [18], features such as distance between two consecutive QRS complexes, QRS complex shape are provided by hidden Markov model based framework which is developed by Andreão *et al.* [32].

1.2.1. NN based approaches

Hu *et al.* [7] studied QRS complex beat classification feeding QRS beat patterns to MLP for classification of one normal and 12 abnormal classes. MIT-BIH database was used. QRS complex templates were extracted using adaptive nonlinear ANN. When QRS beats in data are examined, variations are seen in morphology of QRS waveforms belonging to the same class but different patients, and also similarities are observed in QRS beats of different class and different patients. In the study, two experiments were performed. First experiment was to classify 13 classes using 51-40-13 structured MLP.

The accuracy of 65% was obtained. Second experiment was to combine two MLPs. One MLP with 51-25-2 structure was employed to classify data for normal and abnormal classes. Second MLP with 51-30-12 structure categorized data which was classified as abnormal by the previous MLP, into one of 12 abnormal classes. This cascaded model of MLPs improved the accuracy rate to 84.4%.

Niwas *et al.* [12] studied arrhythmia classification using a multilayer feedforward neural network trained with backpropagation algorithm. In this work, QRS duration, RR-Interval features which were extracted with the heartbeat detection by Pan and Tompkins [22] and spectral entropy that describe heart rate variability comprised the feature set. Including normal beat, 10 different beat types were classified. An overall accuracy of 99.02% was obtained.

Another study for the use of the neural network with the backpropagation algorithm to classify ECG beats was presented by Yeap *et al.* [6]. In experiments, AHA database was used. Before classification, data was preprocessed to extract features. The width of the QRS, the amplitude of the QRS, the offset of the QRS, T wave slope and the prematurity were the extracted features. A neural network with two hidden layers of 20 hidden nodes was employed. Four samples out of 80 were used for training. 98.36% sensitivity was achieved on training set and 67.80% sensitivity was obtained on complete dataset.

Three different ANN models: multilayer perceptron (MLP) neural network model, generalized feedforward neural network and modular neural network model were compared for arrhythmia classification byJadhav *et al.* [20]. UCI arrhythmia dataset was used in the experiments. All network models were trained with backpropagation algorithm using gradient descent with momentum learning. The original dataset was grouped into five different data sets. Each data set was partitioned into training set and test set with different ratio (e.g. data set_1 was partitioned into 80% tra. and 20% test, data set_5 was 90%tra. and 10% test.). Among three models, MLP ANN was better with 86.67% accuracy and 93.75% sensitivity for data set_5.

1.2.2. SVM based approaches

The use of SVM in ECG classification is very popular recently. Uyar *et al.* [15] applied SVM with PCA and ICA on UCI Arrhythmia dataset to detect absence or presence of arrhythmia. According to experiments, SVM result was improved with PCA from 79% to 84%. The result of SVM was compared to k-NN and DT algorithms. Experiments showed that SVM outperformed other classifiers.

Song *et al.* [14] used linear discriminant analysis (LDA) for dimension reduction before applying SVM on data. 17 extracted features were reduced to four by LDA. Experiments showed that LDA over SVM gives higher performance than PCA over SVM. Six arrhythmia categories were classified and 99.35% average accuracy was achieved.

Based on heart rate variability, Asl *et al.* [17] classified six dfferent types of arrhythmias. These arrhythmias were normal sinus rhythm, premature ventricular contraction, atrial fibrillation, sick sinus syndrome, ventricular fibrillation and 2° heart block. 15 features were extracted from HRV signal. Using generalized discriminant analysis (GDA) which is a nonlinear extension to LDA, features were reduced to five. One-against-all SVM was used as a classifier and obtained high accuracy in the experiments.

Ozcan *et al.* [19] combined fuzzy approach with SVM for arrhythmia classification. Problem was defined as a two-class problem: normal or abnormal. Five types of fuzzy membership techniques were applied.These techniques were one class weighing method (OCW), distance to class mean (DTCM), distance to one class mean (DTOCM), cardinality (CAR) and fuzzy c-means (FCM) clustering. According to experiments, FSVM-DTCM provided highest accuracy. Results were compared to k-NN, MLP and SVM. Proposed method overperformed others with 83.33% accuracy. Based on the classified data, rule extraction algorithms were applied.

1.2.3. Other approaches

Homaeinezhad *et al.* [21] proposed a hybrid approach combining SVM, k-NN and four different MLP models for arrhythmia classification. A fusion network was created using median weighted voting. An accuracy of 98.06% was achieved.

Oliveira *et al.* [18] used Bayesian Networks to detect premature ventricular beats (PVC) in ECG signal. In experiments, both channels and the information of the last beat were taken into account to reach the best results. Bayesian Networks comprised of discrete and continuous variables. The nodes corresponding to the discrete variables were the PVC Beat, the Premature Beat and the Ventricular Beat while the continuous variables were R-R interval from two leads and the likelihood of the QRS complexes of the last normal beat from two leads. Different network topologies were applied and evaluated. Results showed that networks using both channels were better.

Chazal and Reilly [9] concentrated on different feature sets obtained from wavelet transform, standard cardiology features described by Chazal and Celler [33] and directly from time-domain samples. They applied linear, quadratic and logistic discriminants and compared the results over different feature sets. Frank Lead ECG records were used. In the study, both feature selection and classifier model selection were performed. According to accuracy, LDA performed best on time-domain samples.

Based on the previous work in ECG analysis, it is seen that many feature extraction methods and classifiers have been implemented. MIT-BIH database and UCI Arrhythmia database are two common databases. The original ECG signal and preprocessed ECG signal have been used. There are many preprocessing methods implemented. For example, bandpass filtering [22], linear adaptive filtering [31], adaptive nonlinear ANN [7], discrete wavelet transform (DWT), continuous wavelet transform (CWT) [24], Hilbert transform [29]. In order to improve performance of the classifiers, dimension reduction methods have been applied such as LDA [14], PCA [13, 15, 19], GDA [17]. In this thesis, UCI Arrhythmia dataset is used. We aim to model neural network classifier which eliminate irrelevant features using genetic algorithm (GA)

technique. The rule set is extracted based on the classification model. The details of the proposed method are described in the following sections.

1.3. The Use of GA-NN in Medical Diagnosis

A combined form of genetic algorithms and neural networks has increasingly been used in the literature. The common reason of this increase is the idea that two of them may provide an efficiency for solving more problems than either of them alone. The way to use this combination is varying according to the problem. Amma [34] used the optimization technique of genetic algorithm to select optimal weights for neural network classifier. In the study, each chromosome was designed as a set of weights and best chromosome was obtained using the global search ability of the genetic algorithm in the weight space and thereby avoiding local minima. Selection was performed according to fitness function which was based on the root mean square errors. The weights were set to neural network and the neural network was trained with backpropagation algorithm. The RMSE was calculated using the output of the network. At the end of this iterative process, the final results were stored in weigth base to be used as a final classifier. In experiments, 103 out of 303 instances were used for testing set and obtained 94.17% accuracy. Another use of genetic algorithms for optimization purpose is seen in the study of Jiang *et al.* [35]. Neural network structure and weights were encoded in chromosomes. The encoding of neural network structure was done as signal bit sequence whilst connection weights and biases were represented as real valued numbers. Optimization was performed by maximizing the fitness function. Results of the experiments showed that the neural network achieved 98% accuracy. Ölmez *et al.* [36] proposed a neural network which was trained by genetic algorithms and called GARCE for ECG classification. They used genetic algorithm to design both network topology and associated connection weights to improve accuracy. In the study, optimization performed by genetic algorithms was limited with the first layer of the network for time concerns. The use of genetic algorithm assisted neural network approach was also seen in the study of Dokur and Ölmez [37]. Genetic algorithms were used to determine weights and number of nodes in the first layer by optimizing the location and the radius of the class boundaries. 96% accuracy in beat classification

was achieved. Genetic algorithms might also be used for learning purpose. Zhou and Li [38] studied premature ventricular contraction (PVC) detection in ECG signals using genetic algorithm trained perceptrons. Every weight vector of any single neuron in the perceptrons was encoded as 16-bit string each representing a real valued number in range of [-1.0, 1.0]. Model structure was adaptively optimized using genetic algorithm steps. According to experimental results, 96.96% average accuracy was achieved.

1.4. Proposed Method

In this thesis, we propose genetic algorithm-neural network approach for ECG feature selection in rule based arrhythmia classification. This proposed system is comprised of four main stages: model selection, feature selection, classification and rule extraction. In this genetic algorithm-neural network combination, genetic algorithm has an optimizing role whilst neural network plays the role of classifier. Therefore, the type of this combination might be considered as *supportive combination* according to Schaffer *et al.* [39] because the way to use genetic algorithm is to assist neural network in feature selection.

Each neural network with the same topology in a population is considered as a candidate solution. ECG features are represented as inputs to neural network and output nodes indicate class labels. Each candidate solution is assigned a fitness value which is an assessment that indicates how good a solution is. Fitness values are calculated based on the performance of the solutions. The possibility of the solutions to be in next generation is inversely proportional to their fitness values. It is intended that the new generation to include solutions better than their ancestors. Parent solutions are selected using roulette wheel selection. Then crossover is applied to produce offsprings. These offsprings are mutated so as to provide variation. Additionally, mutate2 operator is applied which prunes the network from irrelevant features by zeroing their weights and fully described by Sexton *et al.* [40]. The best offsprings produced as a result of these operations comprise of the 90% of the next generation. The remaining part is filled by the best solutions from old generation in order to carry best characteristics from ancestors. The solution with the best fitness value of all its generation is

compared with the one from old generation. The iteration is stopped when the previus best solution is still better than the recent solution.

As a candidate solution, a neural network is evaluated according to its performance on validation data. The performance is generally related to the error which is the difference between the actual output and the predicted output. Backpropagation algorithm is used for training the neural network. Training parameters are predefined. All networks in a population are trained and tested using the same parameters.

We aim to detect presence or absence of arrhythmia accurately. In order to achieve this, selection of network parameters improving the accuracy of the classifier is important. So, using trial-and-error method, some promising parameter sets are specified and for each parameter set, neural networks are trained and tested so as to find optimum parameter set. Parameter selection is summarized in Figure 1.4. Both feature selection and classification are performed simultaneously using the neural network with selected parameter set. Thereby we obtain classification model with only relevant features that mostly contribute to the classification.

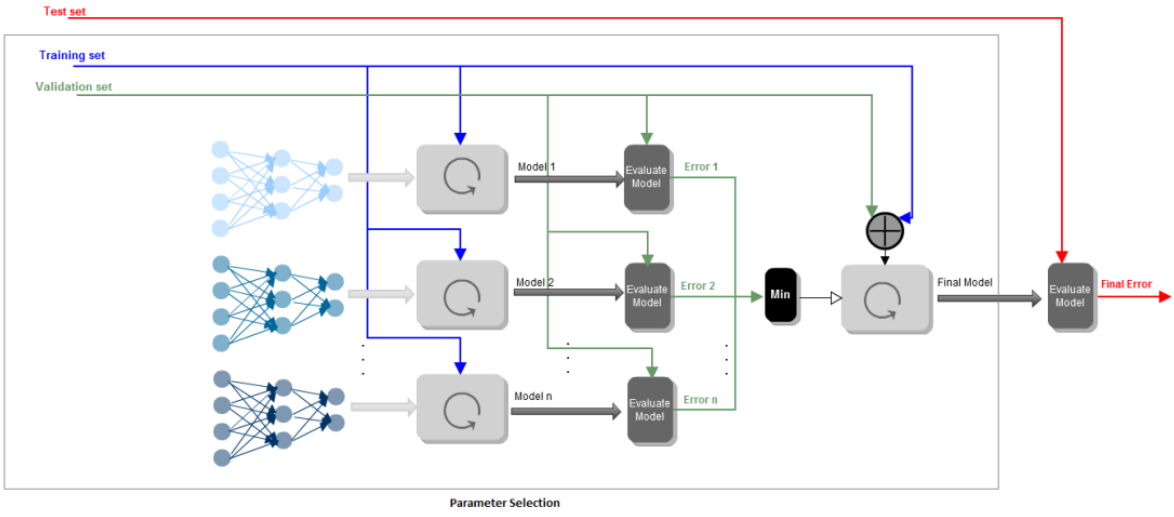

Figure 1.4. A diagram of parameter selection [41].

The last stage of the proposed system is the rule extraction. As mentioned in

previous sections, we are mainly interested in producing robust and reliable arrhythmia classification results to be applicable in diagnostic decision support systems and to aid cardiologists with medical assessments. Rule extraction from classification output is important in terms of producing the results in human readable format. Applying solely rule extraction methods will increase the complexity in rules as fully described by Halford *et al.* [42]. So, genetic algorithm-neural network approach is used as a preclassifier and feature selector to produce rules concisely. Main flow of the system is summarized in Figure 1.5. C4.5, RIPPER, PART and HotSpot methods are used to perform rule extraction.

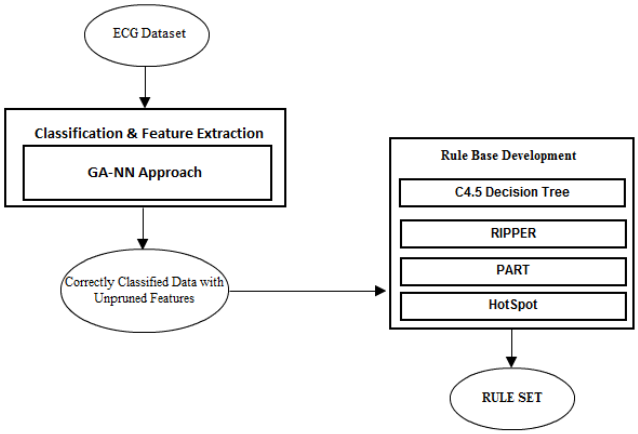

Figure 1.5. Main flow of the system.

As a comparison, k-Nearest Neighbor (k-NN), Support Vector Machines (SVM), Naive Bayes and Bayesian Networks are tested on the arrhythmia dataset. Dimensionality reduction techniques are used such as recursive feature extractor (RFE-SVM), correlation based feature selection (CFS), principal component analysis (PCA) and factor analysis (FA). These classifiers are applied both original dataset and the reduced data sets to observe the effect of the dimension reduction techniques on the improvement of the classifiers. The flow diagram of the comparison is given in Figure 1.6. The performance of these techniques on original dataset and reduced data sets are compared with the proposed approach.

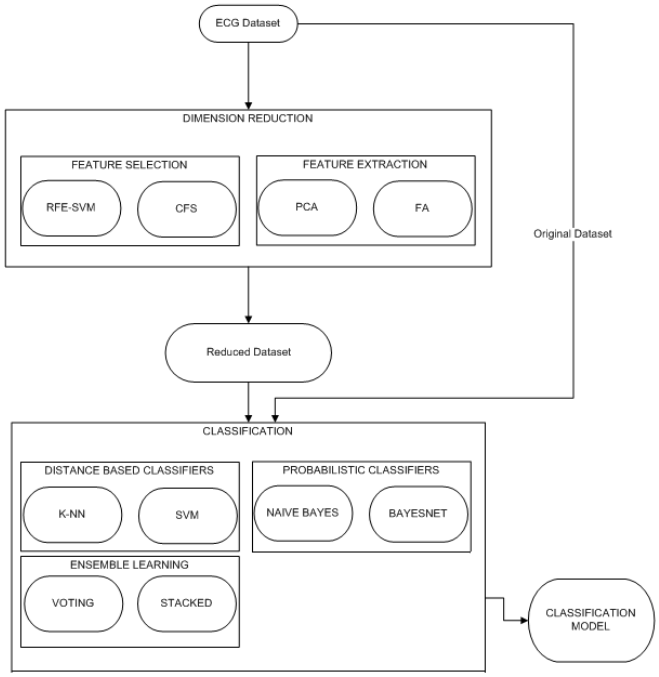

Figure 1.6. Flow diagram of the comparative classifiers.

1.5. Dataset

The dataset used in this study is obtained from UCI Repository [4]. UCI Arrhythmia dataset originally contains 452 instances with 279 attributes. There are 16 arrhythmia cases associated with each instance. 15 of them indicate anomalies and one of them is normal rhythm. 206 attributes are numeric and the rest is nominal.The first four features are about personal information such as age, sex, heigth and weight. The rest presents different measurements from the ECG paper.

About 0.33% of the feature values in the dataset are missing. Class distribution of this dataset is very unfair and instances of classes 11, 12 and 13 do not exist in the dataset while instances of class 01 which indicates normal rhythm is frequent.

Dataset is preprocessed to have better sensitivity rates. All anomalies are grouped into one class as abnormal rhythm and the resulting dataset contains inputs of two groups: Normal and Abnormal. Missing values among the features are removed. This

leads a reduction to 278 features and 420 instances. The official dataset description of the UCI Arrhythmia dataset can be found in APPENDIX A.

The second dataset is a set of real ECG data. This data is obtained from Kardiosis ECG Tool of the manufacturing firm TEPA [43]. This dataset is smaller than UCI Arrhythmia dataset. There are 20 records, 13 of them are normal and the rest is abnormal. Only relevant features which are obtained as a result of the GA-NN approach are computed from these records because this dataset is intended to verify the proposed method.

1.6. Outline

In previous sections, medical information about cardiac arrhythmia and ECG is briefly mentioned. A summary of the literature works on arrhythmia classification and ECG analysis is provided. The proposed method and the data used in this study are described in detail.

Chapter 2 explains dimension reduction techniques which are applied on original dataset for comparative classifiers.

Chapter 3 presents the technical details of the proposed method, other classifiers used for comparison and rule extraction methods.

In Chapter 4, experiments and experimental results are described. Comparative performance results of the proposed approach with other classifiers are given. The resulting rule sets are presented.

Chapter 5 states the experimental results for real life ECG dataset.

In Chapter 6, the thesis is concluded with a summary of the empirical results.

2. DIMENSION REDUCTION

Dimensionality reduction is the search for a small set of features to describe a large set of observed dimensions. Reducing the number of dimensions can separate the important features from the less important ones, thus providing insight into the nature of the data that may otherwise be left undiscovered. In addition, dimensionality reduction is useful in decreasing space complexity and computational processing time [44, 45].

There are two main methods for reducing dimensionality: feature selection and feature extraction. Feature selection aims to find a subset of the original dataset which give us the most information. Feature extraction transforms the original features into a new set of features which are the combinations of the original features.

A high dimensional dataset is used in this thesis. There are 278 features for each instance. The dimension reduction has impact on the results. Four techniques are applied to the dataset. Classification results of reduced datasets are then compared with each other. The following subchapters explain dimension reduction techniques used in this study.

2.1. Feature Selection

In feature selection, dimension of the original dataset is reduced by elimination of unnecessary features. To determine the necessity of features, ranking is performed among them and all features are sorted in decreasing importance. This ranking and elimination are done either at one step or iteratively. Feature selection can be formed as forward and backward. In forward selection, the most important features are selected at each iteration whereas in backward selection, the least important features are removed at each step. The resulting set consists of a subset of the original features.

In this study, two feature selection techniques are applied: Recursive Feature

Elimination with Support Vector Machine (RFE-SVM) and Correlation Based Feature Selection (CFS). These algorithms are explained in detail in the following sections.

2.1.1. Recursive Feature Elimination with Support Vector Machine

Recursive feature elimination is a recursive backward elimination technique. At each iteration, weights are assigned to all features in the dataset. Based on the magnitude of the weights, features are ranked. In this ranking, features with the lowest weight are eliminated and the algorithm is run with the rest of the features until desired dimension is met. As feature ranking criterion, weights can be produced by several ways. Usage of classifiers is one option. Recursive Feature Elimination combined with Support Vector Machine is developed in [46].

The criteria estimate the effect of removing one feature at a time on the objective function. The first step of this iterative procedure is to train SVM model with all features and then compute the weight vector of dimension length. The weight vector is calculated with the following function:

$$w = \sum_{k=1}^{l} a_k y_k x_k \tag{2.1}$$

The weight vector w is a linear combination of training patterns. l is the count of input. x_k is the kth input and y_k is the corresponding class label of the x_k. a_k is the dual coefficient. Based on the weight vector, the ranking criteria is computed for all features where w_k is the weight of the kth feature.

$$rankCriteria_k = (w_k)^2 \tag{2.2}$$

According to the ranking criteria, feature with smallest ranking score is eliminated from the feature set. This process is iterated until the desired number of resulting features is met. This algorithm can be applied to remove one feature per step or can be generalized to eliminate more than one feature at each iteration. The latter is preferred for speed reasons.

2.1.2. Correlation Based Feature Selection

Correlation Based Feature Selection is a feature selection algorithm which is proposed in [25]. This algorithm ranks feature subsets according to a correlation based heuristic evaluation. The evaluation is performed in feature subsets rather than individual features. The aim of the evaluation is to end up with a subset which contains features highly correlated with a class value but uncorrelated with each other.

Ranking is done in a search space for all possible feature subsets. The ranking scores are computed with the following heuristic function:

$$Merit_s = \frac{k\overline{r_{cf}}}{\sqrt{k + k(k-1)\overline{r_{ff}}}} \tag{2.3}$$

$Merit_s$ represents the heuristic "merit" of feature subset S which contains k features. $\overline{r_{cf}}$ is the mean feature-class correlation, and $\overline{r_{ff}}$ is the mean feature-feature correlation. The feature subset with the highest heuristic merit is the output of this algorithm.

In order to apply Equation 2.3 it is important to compute feature-class and featurefeature correlations. Hall, [47] proposes, two methods for computing the intercorrelation between two features: Relief and MDL.

Relief is an instance based attribute ranking scheme which is proposed in Kira *et al.* [48]. This algorithm iteratively samples an instance from the data and locates

its nearest neighbour from the same and the different class. The difference between the values of the attributes of the nearest neighbours and the sampled instance states the relevance scores of each attribute. The idea behind this algorithm is to find useful attribute which should have different scores between instances from different classes and the same value for instances from the same class.

MDL is the abbreviation for the minimum description length algorithm which is proposed in Rissanen *et al.* [49]. MDL is defined in Kononenko *et al.* [50] as below:

$$MDL = \frac{Prior_{MDL} - Post_{MDL}}{n} \tag{2.4}$$

$$Prior_{MDL} = \log_2 \left(\begin{array}{c} n \\ n_1, ...n_c \end{array} \right) + \log_2 \left(\begin{array}{c} n + C - 1 \\ C - 1 \end{array} \right) \tag{2.5}$$

$$Post_{MDL} = \sum_{j=1} \log_2 \left(\begin{array}{c} n_{.j} \\ n_{1j}, ...n_{cj} \end{array} \right) + \sum_{j=1} \log_2 \left(\begin{array}{c} n_{.j} + C - 1 \\ C - 1 \end{array} \right) \tag{2.6}$$

Above equations give definition of MDL. In these equations, n is the number of training instances, C is the number of class labels. n_i is the number of instances in class C_i. $n_{.j}$ states the number of instances with the jth value of the given feature and n_{ij} is the number of instances from class C_i which have the jth value of the feature.

2.2. Feature Extraction

The aim of feature extraction is to map the original dataset to a new fewer dimensional dataset with minimum loss of information. As a difference from feature

selection, the resulting feature set of feature extraction is a combination of the original features instead of a feature subset. The concern is to combine features in a group to decrease the processing cost without losing information.

In this study, two feature extraction techniques are applied: Principal Component Analysis (PCA) and Factor Analysis (FA). These algorithms are explained in the following subchapters.

2.2.1. Principal Component Analysis

PCA is well-known and widely used feature extraction technique. PCA tries to maximize variance of features by changing the rotation of the axes of the original coordinate system to a new set of orthogonal axes on which the difference between the sample points becomes most apparent.

In PCA, features are transformed into a smaller group of uncorrelated set called principal components. For the first step of this transformation, the mean of every feature is subtracted from each dimension. As a result, a feature set with zero mean is obtained. Then, covariance matrix of this feature set is calculated. The eigenvectors and the corresponding eigenvalues are computed from the covariance matrix. The eigenvalues are sorted in descending order. The eigenvector associated with the highest eigenvalue has the same direction as the first principal component. The higher eigenvalue is the more significant the corresponding eigenvector. The resulting set is the multiplication of the selected set of eigenvectors and the input data.

The optimum number of dimension of the resulting set is determined according to the proportion of variance. Proportion of variance is the proportion of sum of highest k eigenvalues to sum of all eigenvalues. The optimum value for k is the one where the proportion of variance is greater than a predefined threshold value. Once k is computed, the linear projection is applied to principal components of the dataset in order to obtain k dimensional reduced data.

2.2.2. Factor Analysis

Factor Analysis is a method to describe the variability among the correlated, observed variables in the dataset in terms of fewer unobserved variables which are called factors. In other words, in factor analysis, it is assumed that observed variables are linear combinations of underlying factors and error terms. FA assumes that each observed variable can be expressed as a weighted sum of factors plus the error term. If two variables are highly correlated, then they are related through a factor whereas if two variables are uncorrelated, then they are related through different factors.

Once the number of factors are determined, features are then rewritten as a linear combination of these factors plus the error term.

$$x_i = a_1 f_1 + a_2 f_2 + a_3 f_3 + ... + a_n f_n + e_i, \quad \forall X_i \in S \tag{2.7}$$

$$X = AF + \varepsilon \tag{2.8}$$

Equation 2.7 depicts a rewritten feature in terms of factors and the error term. In Equation 2.8, a matrix form of features is presented. Error terms, e_i, are assumed to be independent from each other. F and ε are also assumed to be independent from each other. Factors are standardized to have 0 mean and 1 variance.

As the new features are defined in terms of factors, the smaller the number of factors result in the smaller dimension the resulting feature set.

3. ECG ANALYSIS

Among the most important sources of diagnostic information, ECG signals and the improvements in their analysis have been essential by means of telling consequences. Hence, the progress in the area of signal processing, classification, and interpretation has been a great deal. In order to classify the ECG signal, a reliable extraction of the characteristic ECG parameters is needed. A set of algorithms for signal conditioning, QRS detection, measurement of wave amplitude, duration, area and regularity are applied to perform the parameter extraction.

In order to detect anomalies in ECG signal for better sensitivity, focusing on the key features is essential. Hence, as a first step in preprocessing, dimension reduction techniques are applied to the dataset. Given the selected attributes as input, classification techniques are applied and their results are compared.

For better understanding of the performance of a classifier, the dataset is separated into train and test segments. The train segment is used to extract model parameters of the classifier and given the test segment as input, the classifier is run with the extracted parameters to predict the classes of the input. According to the results calculated in terms of the performance metrics, classifiers are compared with each other.

On basis of correctly classified data, rules are generated in terms of the features that mostly contribute to the classification.

3.1. ECG Classification

In ECG analysis, recognition of the features in order to gain insight of the data is very important. As a part of this process, ECG classification helps analyze the features in terms of their effects in the existence of the target class.

In this study, all anomalies in the UCI Arrhythmia dataset [4] are considered as abnormal class. ECG Classification is responsible to assign the test data to either normal class or abnormal class. Since the objective of this study is to detect anomalies, the target class is considered as the abnormal class.

In addition to the proposed technique, four other classifiers are applied to the UCI Arrhythmia dataset to compare the classifiers. As distance based classifiers, k-NN and SVM algorithms are used in order to experiment local based and globally based classifier performance. For probabilistic approach to the problem: Naive Bayes and BayesNet classifiers are applied.

3.1.1. k-Nearest Neighbor

k-Nearest Neighbor (k-NN) is a nonparametric distance based classifier which assumes similar inputs have similar outputs. Based on this assumption, inputs having nearest distance to neighbors are considered to belong to the similar class. k-NN classifies a new instance according to class of k closest training inputs which has the majority of inputs. All neighbors have equal vote, and the class having majority of voters among the k neighbors is chosen. k is generally chosen as an odd number in order to minimize confusion between two neighboring classes [45].

The distance calculations are performed over the training instances. There are different distance metrics used in this technique: Euclidian, Mahalanobis, Minkowski, Manhattan, or some other distance measurements. Most generally used distance metric is Euclidean. In Euclidean, a linear distance between two points is calculated. Mahalanobis calculates the distance between two points by variation in each component of the points [51].

3.1.2. Support Vector Machine

Support Vector Machine (SVM) is an algorithm of machine learning, introduced by Vapnik, based on the Structural Risk Minimization principle from Statistical Learn-

ing Theory [52].

SVM tries to find a hyperplane that separates data points of each class while maximizing the minimum distance between the hyperplane and the data points. SVM maps the input space into a higher dimensional feature space by using appropriate kernel functions where each coordinate corresponds to one feature in the dataset. In SVM, there are four basic kernels:

- linear
- polynomial
- radial basis function (RBF)
- sigmoid

The mathematical formulation of SVM is a quadratic programming problem. Assume a set S of n labeled training set is given in the following form:

$$S = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n) \quad \forall x_i \in R^n, \quad y_i \in 1, 2, ...k \tag{3.1}$$

Each data point, x_i is a member of either of k classes which is represented as y_i. Training data is mapped to higher order feature space Z to make search for optimal hyperplane easier. This mapping is done by a kernel function. The separating hyperplane is computed by maximizing the distance of the closest patterns, namely margin maximization. The distance from the hyperplane on each side is called margin. The hyperplane is rewritten as:

$$w \times z + b = 0 \tag{3.2}$$

$$y_i(w \times x_i + b) \geq 1 \tag{3.3}$$

The parameters w is the weight vector, and x is the input. Maximum margin is calculated for the minimum value of w.

3.1.3. Naive Bayes

A Naive Bayes classifier is a simple probabilistic classifier based on applying Bayes' theorem. The term naive is because this classifier relies on two important simplifying assumptions. First assumption is that the predictive attributes are conditionally independent given the class, and as a second, it posits that no hidden or latent attributes affect the prediction process [53].

Naive Bayes is based on Bayes' rule. Suppose that C is a variable denoting the class of an instance while c indicates a particular class and X is a vector of variables holding observed attribute values while x represents a particular observed attribute value vector. Given an x to classify, the probability of each class given x can be expressed in the following form:

$$p(C = c|X = x) = \frac{p(C = c)p(X = x|C = c)}{p(X = x)} \tag{3.4}$$

Equation 3.4 predicts the most probable class. Because the observed attributes are assumed to be conditionally independent, the following equation helps classify test data x from the training data:

$$p(X = x|C = c) = \prod_i p(X_i = x_i|C = c) \tag{3.5}$$

In Naive Bayes, model parameters are approximated using maximum likelihood estimates assuming distribution of the features is a known distributions such as Gaussian distribution or kernel density estimation. Estimation of the density of each attribute can be written:

$$p(X = x|C = c) = g(x, \mu_c, \sigma_c) \tag{3.6}$$

When kernel estimation is considered using Gaussian kernels, the following equation is obtained:

$$p(X = x|C = c) = \frac{1}{n} \sum_{i=1} g(x, \mu_i, \sigma_i) \tag{3.7}$$

where n is the number of attribute X in class c, and $\mu_i = x_i$.

3.1.4. Bayesian Networks

A Bayesian Network, also known as belief network, is a graphical model for probability relationships among a set of features. The structure of the Bayesian Network is a directed acyclic graph where the nodes represent random variables and the arcs represent conditional independence assumptions. In directed model, a Conditional Probability Distribution (CPD) is defined at each node which hold the data about the conditional probability of child node given the current node as parent [54].

Bayesian Networks mainly provides two advantages: one is that bayesian networks provides visualizing the process which give better understanding and secondly, Bayes' rule is applicable making use of graph operations [45].

The most common task for Bayesian networks is to probabilistic inference. Bayes'

rule is used to compute the posterior probability of θ given the data D:

$$p(\theta|D) = \frac{p(\theta)p(D|\theta)}{p(D)} \tag{3.8}$$

3.1.5. Genetic Algorithm-Neural Network (GA-NN) Approach

Genetic Algorithm integrated Neural Network classifier is the part of the proposed system which performs both feature selection and classification simultaneously. As a result of this process, it is aimed to select relevant features which mostly contribute to the classification. Based on the resulting classification model with relevant features, rules are intended to be produced in human readable format.

As mentioned in the previous sections, a population is comprised of neural networks with the same topology. Therefore, each candidate solution represent a neural network. These neural networks are created with the same parameter set which is predefined.

Algorithm is run iteratively. At each iteration, a new generation is produced and its best solution is put aside to be evaluated. Assume that current iteration number is i and x_i indicates the best solution of i^{th} iteration. The iteration is stopped when the performance of x_{i+1} is not better than the performance of x_i.

Performance of the neural networks is based on the fitness value [40]. The fitness value is obtained from the fitness function is as below:

$$Min\{f = \sum (O_i - \hat{O}_i)^2 + C\sqrt{\frac{\sum_{i=1}(O_i - \hat{O}_i)^2}{N}}\} \tag{3.9}$$

where O is the target class, $\hat{O}$ is the estimated class of the instance i and N is the number of the instances. Here, C represents the number of nonzero weights in the network. Each network is assigned a probability based on its fitness value [40]. The probability is computed as in Equation 3.10

$$P(X = x) = \frac{f_{cur} - f_{bad}}{\sum_{i=1} f_i - f_{bad}} \tag{3.10}$$

where f_i indicates the fitness value of the solution i, f_{bad} is the worst fitness value which is greater than the rest of the fitness values in the population and f_{cur} is the fitness value of the current solution.

3.2. Rule Extraction

The output of the classification phase is the input to the rule base development. In rule base development, the pattern in correctly classified data is extracted and, thus rule set is generated as output. The extracted rule set gives the definition of the dataset in readable form.

In this study, C4.5 decision tree, Repeated Incremental Pruning to Produce Error Reduction (Ripper), Partial Decision Tree (Part) techniques are applied to the correctly classified data. Furthermore, HotSpot rule extraction algorithm is employed in order to observe effective rules on arrhythmia detection. The methods are described in detail in the following sections.

3.2.1. C4.5 Decision Tree

C4.5 decision tree algorithm is proposed by Quinlan in [55]. In this study, J48 method which is Java implementation of C4.5 in Weka [56] is used.

Decision tree is a hierarchical tree which is composed of two main elements:

decision nodes and leaves. At each decision nodes, a test function is implemented and regarding the evaluated value, the branch to be taken is decided. The aim is to obtain the best split and the goodness of a split is quantified by an impurity measure [45] which is entropy based measure in [55]. The root node with maximum entropy is selected as the root node and the path from the root node is defined according to features having maximum entropy and which is not used in the path from the root. This process continues recursively until a leaf node is reached. In classification tree, leaf nodes represent the class labels. In order to avoid constructing the tree too specific to the training data, tree is pruned. Pruning is done by replacing a sub-tree by a leaf node. This replacement is performed whether the expected error rate in the sub-tree is greater than in the leaf node.

As a result of the tree construction and pruning, the pruned decision tree is produced which represent a rule base. Each rule in the rule base can be rewritten by adding the conditions on each path from each leaf node to the root node.

3.2.2. Repeated Incremental Pruning to Produce Error Reduction

Repeated Incremental Pruning to Produce Error Reduction (Ripper) is proposed by William W. Cohen [57]. This algorithm greedily builds a rule selecting features with highest information gain. After growing, pruning is performed on each rule. This algorithm has a post-processing stage in which each rule is replaced or removed for optimization. The criteria for optimization is the minimum description length. Java implementation of this algorithm is used which is called JRip in Weka.

3.2.3. Partial Decision Tree

Part Decision Tree uses separate-and-conquer technique which builds a partial C4.5 decision tree in each iteration and turns the "best" leaf into a rule [58]. At each iteration, a pruned C4.5 decision tree is constructed and the leaf node having maximum coverage is taken as the rule of that iteration. When the rule is decided, instances which are covered by that rule is removed and this iteration is continued until all instances in

input dataset is processed. In this study, Java implementation of this method which is called Part method is used in Weka.

3.2.4. HotSpot

HotSpot is a rule extraction algorithm which is implemented in Weka. This algorithm inspects the training data and generates the association rules corresponding to class labels in a tree-like structure. In this study, a set of rules for the abnormal class is generated using Weka.

4. EXPERIMENTS AND RESULTS

4.1. Performance Metrics

Classification efficiency has been widely used as the main criterion for comparing the classification quality of classifiers [59]. This efficiency may be defined in terms of performance of the classifier. In order to measure performance of a classifier, test data is applied to the classifier model which has learnt parameters from training data. Examining the confusion matrix of the test results may be an informative indicator for performance. Confusion matrix is a square matrix where the rows represent the actual class and the columns indicate the predicted class of the input data. The basic performance metrics can be obtained from the confusion matrix. The cells of the confusion matrix have special names, as shown in Table 4.1.

Table 4.1. Confusion matrix for two class.

Actual/Predicted	Positive	Negative
Positive	True Positive (TP)	False Negative (FN)
Negative	False Positive (FP)	True Negative (TN)

The cells in the confusion matrix have the following meanings:

TP: The number of unhealthy incidents classified as unhealthy (Correct prediction that patient has arrhythmia)

FN: The number of unhealthy incidents classified as healthy (Incorrect prediction that patient has normal rhythm)

FP: The number of healthy incidents classified as unhealthy (Incorrect prediction that patient has arrhythmia)

TN: The number of healthy incidents classified as healthy (Correct prediction that patient has normal rhythm)

Healthy incidents indicate incidents having normal rhythm; unhealthy incidents indicate incidents having arrhythmia. In this study, positive (target) class is considered as arrhythmia.

Classification sensitivity assumes equal misclassification costs. This assumption is problematic as for most real world problems, one type of classification error is much more expensive than another [60]. For instance, classifying arrhythmia incidents to have normal rhythm (FN) will be cost more than classifying normal rhythm incidents to have arrhythmia (FP) since the first case could cost patient's life.

Other performance metrics are calculated based on these four criteria. In this study, the main performance measure is sensitivity. Sensitivity is also called recall, or the positive rate or hit rate. It is the proportion of exact detection of the target class to the total inputs in target.

$$Recall = \frac{TP}{TP + FN} \tag{4.1}$$

The second important metric is f-score which is the harmonic mean of Recall and Precision. Precision is the ability of producing the same results under changing conditions.

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

Since f-score is a summary of both Recall and Precision, the equation for f-score is as below:

$$F - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.3}$$

In addition to these, Matthews Correlation Coefficient (MCC) measure is chosen which indicates the quality of classifier for two class problem taking both positive and negative

measures into account.

$$Mcc = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{4.4}$$

Specificity is also considered in experiments. Specificity is related to the performance of the classifier to identify negative class which is normal class in this study.

$$Specificity = \frac{TN}{TN + FP} \tag{4.5}$$

4.2. Experimental Results For Dimensionality Reduction

Dimensionality reduction is the process of choosing a reduced set of original features to minimize the time and space complexity. Another aim of this process is improving the performance of the classifiers. In this study, Recursive Feature Elimination with Support Vector Machine (RFE-SVM) and Correlation based Feature Selection (CFS) are used for feature selection while Principal Component Analysis (PCA) and Factor Analysis (FA) are getting used for feature extraction.

Because the original dataset is high dimensional with 278 features, both feature selection and feature extraction algorithms are applied on normalized form of the dataset.

4.2.1. Experimental Results For Feature Selection

In feature selection, the aim is to find the best subset of features. The best subset contains the least number of dimensions that most contribute to sensitivity [45]. In order to achieve this, unnecessary features are removed from the dataset. Features to be removed are selected by ranking. The resulting dataset is expected to consist of the most informative features.

4.2.1.1. RFE-SVM. Weka platform is used for this method. SVMAttributeEval algorithm in WEKA corresponds to RFE-SVM. Ranker is selected as search method. 10 features are eliminated at each iteration. Seven sets with different feature counts are constructed and labeled as RFE-SVM$_i$ where i represents the number of features. *RFE-SVM*$_3$, *RFE-SVM*$_5$, *RFE-SVM*$_7$, *RFE-SVM*$_{10}$, *RFE-SVM*$_{13}$, *RFE-SVM*$_{20}$ and *RFE-SVM*$_{30}$ are the resulting datasets.

4.2.1.2. CFS. CfsSubsetEval method in WEKA platform is used for this algorithm. Backward selection is applied and 10 features are eliminated at each iteration. Seven sets with different feature counts are constructed. The datasets are labeled as CFS$_i$ where i is the number of features. *CFS*$_3$, *CFS*$_5$, *CFS*$_7$, *CFS*$_{10}$, *CFS*$_{13}$, *CFS*$_{20}$ and *CFS*$_{30}$ are the resulting datasets.

4.2.2. Experimental Results For Feature Extraction

In feature extraction, the original dataset is transformed using a linear transformation to a reduced dimension space [44]. In this study, PCA is used to reduce the input data dimension and the resulting dataset is compared to Factor Analysis results.

4.2.2.1. PCA. The purpose of PCA is to derive new variables that are linear combinations of the original variables and which are uncorrelated [44]. In PCA, covariance matrix of the input data is used to obtain eigen vector and corresponding eigen values. We choose the eigen vector with the largest eigen value for the variance to be maximum [45]. To determine the number of dimensions, proportion of variance is used which is preferred to be higher than a defined threshold. Proportion of variance is the proportion of sum of the highest k eigen values to sum of all eigen values. The principal component is the eigenvector of a covariance matrix with the highest k eigen values which meet the proportion of variance. PCA algorithm is applied to data which is implemented in Matlab. The resulting datasets are *PCA*$_3$, *PCA*$_5$, *PCA*$_7$, *PCA*$_{10}$, *PCA*$_{13}$, *PCA*$_{20}$ and *PCA*$_{30}$ where the numbers in the label represent the feature count.

4.2.2.2. FA. The object of factor analysis is to find a lower-dimensional representation that accounts for the correlations among the features [61]. In this study, factoran function in Matlab is used [62]. The original dataset is huge for being a parameter for factoran function. Hence, a smaller set is used as an input to FA. The largest RFE-SVM output is selected as the starting dataset for FA. factoran function is executed on $RFE\text{-}SVM_{30}$. Factor Loadings represent the correlation coefficients between the variables and factors. 'promax' is used to rotate the factor loadings and the scores. The reduced dataset is obtained by projection of factor loadings and estimated covariance on observed variables. The resulting datasets are FA_3, FA_5, FA_7, FA_8 where the numbers in the label represent the feature count. Considering the initial dataset is 30-featured RFE-SVM dataset then maximum number of factors revealed can be 8. Extracted factors are given in Table 4.2.

Table 4.2. Factors extracted from $RFE\text{-}SVM_{30}$.

Factor1	V5_Amplitude_JJ_wave, V2_Existence_Diphasic_Derived_P_wave, DI_Amplitude_Q_wave, AVF_Avg_width_S2_wave and V6_Amplitude_JJ_wave
Factor2	V1_Avg_width_R2_wave, V2_Avg_width_R2_wave, V1_N_intrinsic_deflections and V2_N_intrinsic_deflections
Factor3	V6_Amplitude_T_wave, DI_Avg_QRSTA, weights, DI_Amplitude_T_wave, V3_Avg_QRSTA and AVR_Existence_Ragged_R_wave
Factor4	QRSduration, DI_N_intrinsic_deflections and Tinterva
Factor5	DII_Amplitude_S_wave, AVR_Existence_Diphasic_Derived_P_wave and AVF_Avg_width_R_wave
Factor6	AVF_Amplitude_Q_wave, DIII_Amplitude_Q_wave, AVF_Avg_width_S_wave and DI_Existence_Diphasic_Derived_P_wave
Factor7	V3_Avg_width_Q_wave and V2_Avg_width_Q_wave
Factor8	V3_Avg_width_Q_wave and V2_Avg_width_Q_wave

4.3. Experimental Results For ECG Classification

All of the experiment results that belong to each of the techniques can be found in relevant Appendix section.

4.3.1. k-Nearest Neighbor

The k-nn classifier assigns an instance to the class having most examples among the k neighbors. In the experiments, 6 different k values are used which are 5, 7, 9, 13, and 15. For each k value, k-nn is applied to each dataset and performance metrics of every trial is computed.

When sensitivity results are considered, Figure 4.1, KNN_5 is found to be consistent among the trials. The highest sensitivity rates are reached by KNN_{13} and KNN_{15} which provide 75.64% for CFS_5. KNN_5 and KNN_9 provide 71.79% for RFE-SVM_{13} and CFS_5 respectively. On the other hand, when overall performance is taken into account, which can be calculated by averaging the sensitivity results for the datasets, KNN_5 becomes more accurate. KNN_5 provides an average of 61.60% sensitivity for the ECG datasets whereas the average sensitivity is 60.11% for KNN_7 and 58.42% for KNN_9.

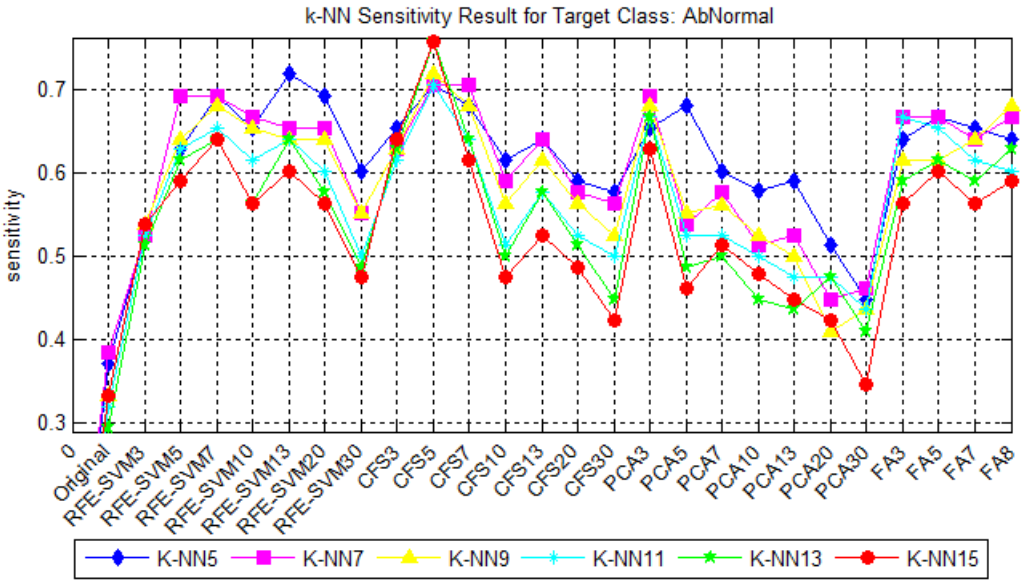

Figure 4.1. Sensitivity test results of k-NN.

F-score results are demonstrated in Figure 4.2. KNN_5 is shown to outperform other k-NN trials. When CFS_5 dataset is considered, 76.13% is scored by KNN_{13} whereas 75.16% is scored by KNN_{15}. However, KNN_5 outperforms KNN_{13} and KNN_{15}

with average score. The average f-score value for KNN_5 is 0.6239 which is higher than 0.603 for KNN_{13} and 0.59 for KNN_{15}.

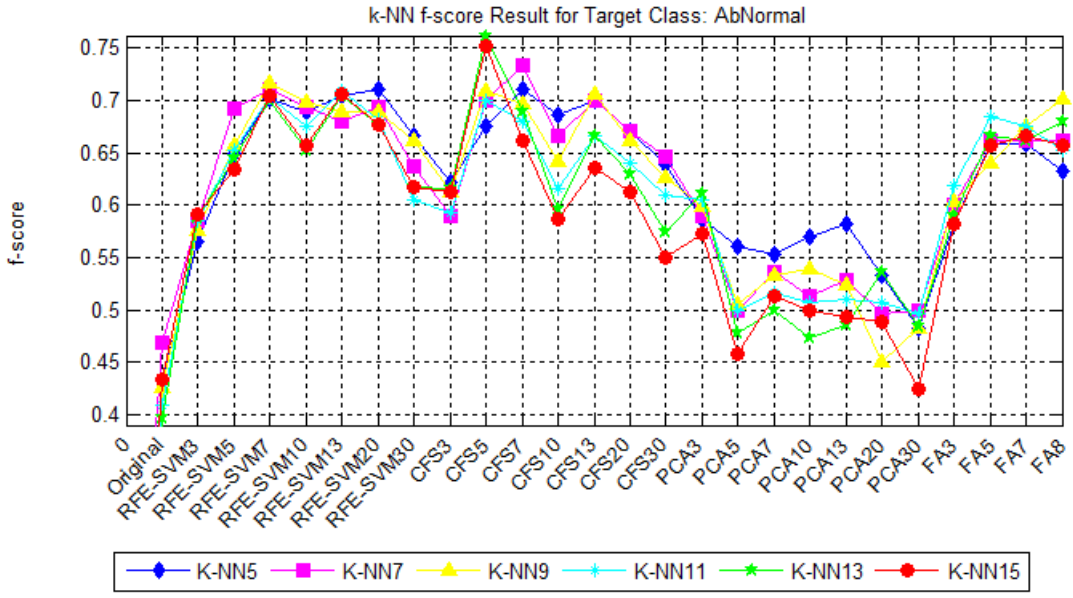

Figure 4.2. F-score test results of k-NN.

When Mathew's Correlation Coefficient is considered, KNN_{13} has better performance than others. MCC result of KNN_{13} is 0.622 for CFS_5. When average MCC result is considered, KNN_{13} performs better with result 0.417. As a result, average MCC result of KNN_5 is 0.409.

When all three performance metrics are taken into account, KNN_5 is found to be the leading k-NN trial. In the inter-comparison of the classification methods, KNN_5 is used to represent k-NN method.

A consensus on the behavior of experiments with k-NN on datasets can be drawn when the three graphs are considered together. The highest values are obtained for RFESVM and CFS datasets. FA results come the second whilst k-NN values for PCA are not as good as the other datasets. Furthermore, k-NN results for original dataset is also not satisfying. These results show that RFE-SVM and CFS datasets have a more concentrated distribution in their feature space by means of their contribution to

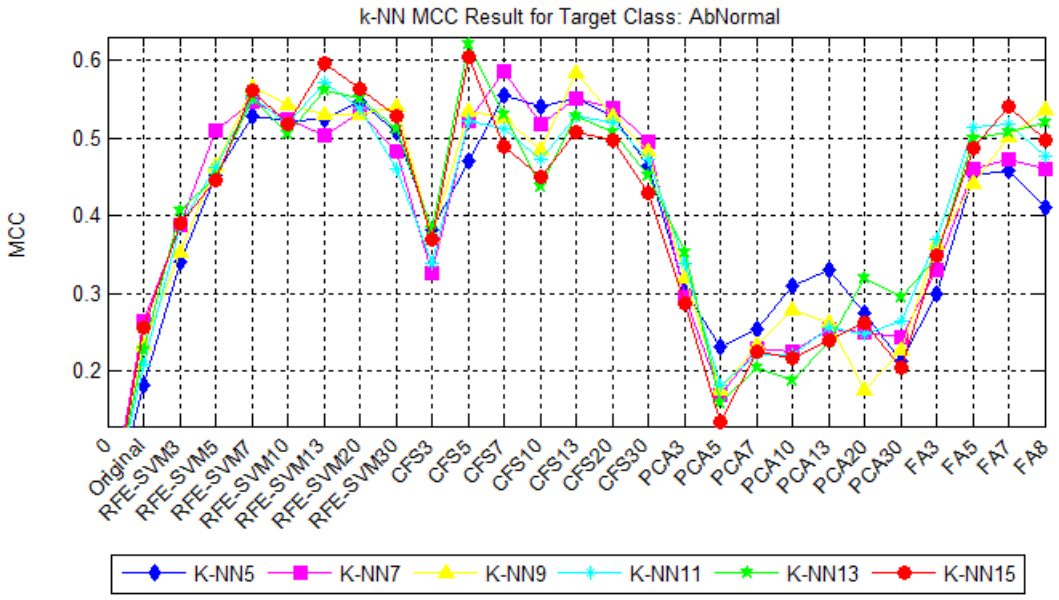

Figure 4.3. MCC test results of k-NN.

classification.

4.3.2. Support Vector Machine

Support Vector Machines map data points to a high dimensional feature space where a separating hyperplane can be found. This mapping can be carried on by transforming the input space into high dimensional feature space using appropriate kernel functions. The optimal hyperplane is found by maximizing the distance between data points between different classes. The smallest distance between the decision boundary and the closest data points is called as margin and the location of the boundary is determined by support vectors.

In this study, Radial Basis Function is chosen as kernel function. The optimum cost and gamma parameters are experimented by performing grid search and cross-validation for each dataset. These parameters are given in Table 4.3. Parameters which give the best sensitivity are chosen to train SVM models for each dataset.

In MATLAB libSVM tool is used for SVM classification [63, 64]. The whole

Table 4.3. Optimum cost and gamma values for SVM.

Dataset	**Cost**	**Gamma**
Original	8	0.03125
$SVMRFE_3$	32768	0.002
$SVMRFE_5$	2	8
$SVMRFE_7$	2	8
$SVMRFE_{10}$	2	2
$SVMRFE_{13}$	0.5	2
$SVMRFE_{20}$	2	2
$SVMRFE_{30}$	512	0.0078
CFS_3	8192	2
CFS_5	512	0.5
CFS_7	32	2
CFS_{10}	512	0.5
CFS_{13}	128	2
CFS_{20}	2	8
CFS_{30}	8	2
PCA_3	0.5	2
PCA_5	0.5	0.1250
PCA_7	32	0.1250
PCA_{10}	32	0.1250
PCA_{13}	512	0.0313
PCA_{20}	2	0.5
PCA_{30}	512	0.002
FA_3	0.5	0.5
FA_5	0.5	0.1250
FA_7	32768	1.2207e-04
FA_8	128	0.0078

dataset is split as 50% for training the SVM model and 50% for testing.

The sensitivity result show that SVM performance is better with CFS dataset. The highest sensitivity rate which is 35.71% is provided for CFS_{20} dataset. An average sensitivity of 30.90% is achieved with CFS datasets and RFE-SVM comes the second with 29.44%.

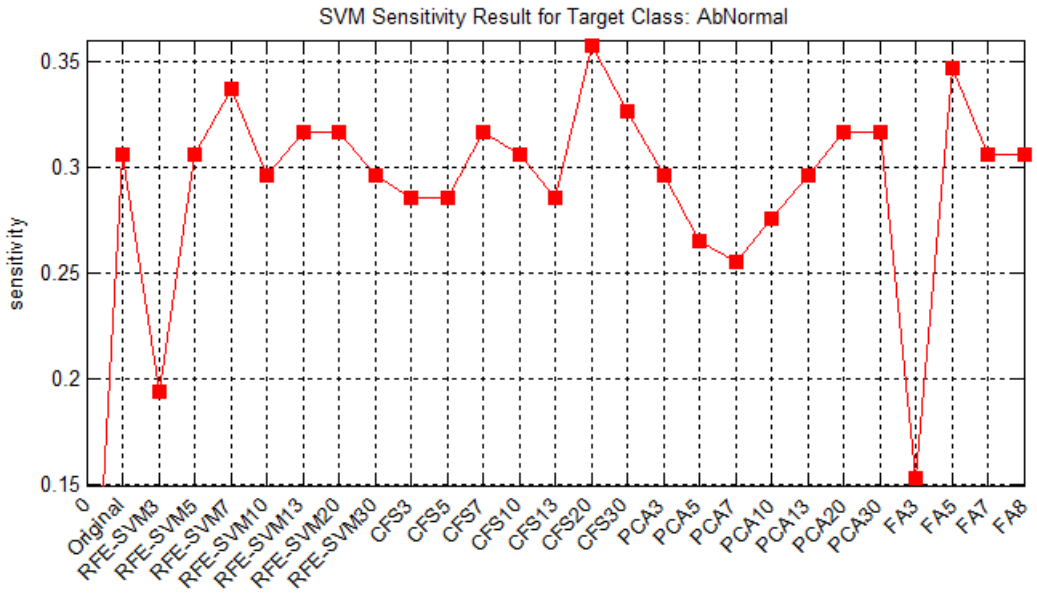

Figure 4.4. Sensitivity test results of SVM.

According to average f-score, CFS datasets come first with 0.3667 and RFE-SVM come the seconds with 0.365. Individually, FA5 dataset has the highest f-score which achieved 0.4121.

In this performance metric, the highest MCC result, 0.0634 is achieved for $RFE\text{-}SVM_{20}$ dataset.

4.3.3. Voted Combination of SVM and k-NN

Voted combination of SVM and k-NN is experimented using Vote classifier in Weka. Combination rules that are applied are: average of probabilities, product of

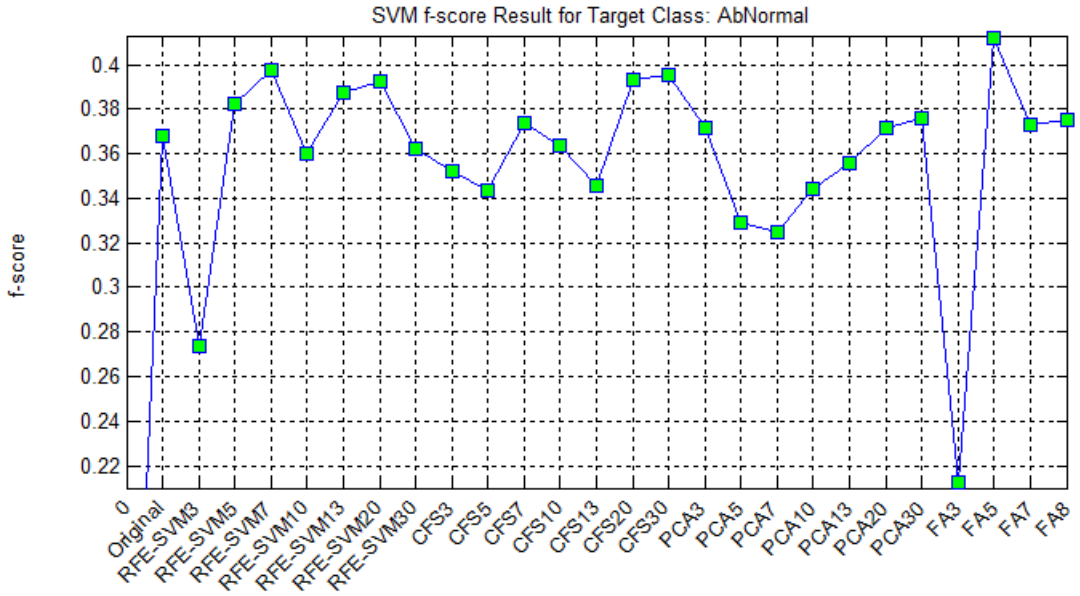

Figure 4.5. F-score test results of SVM.

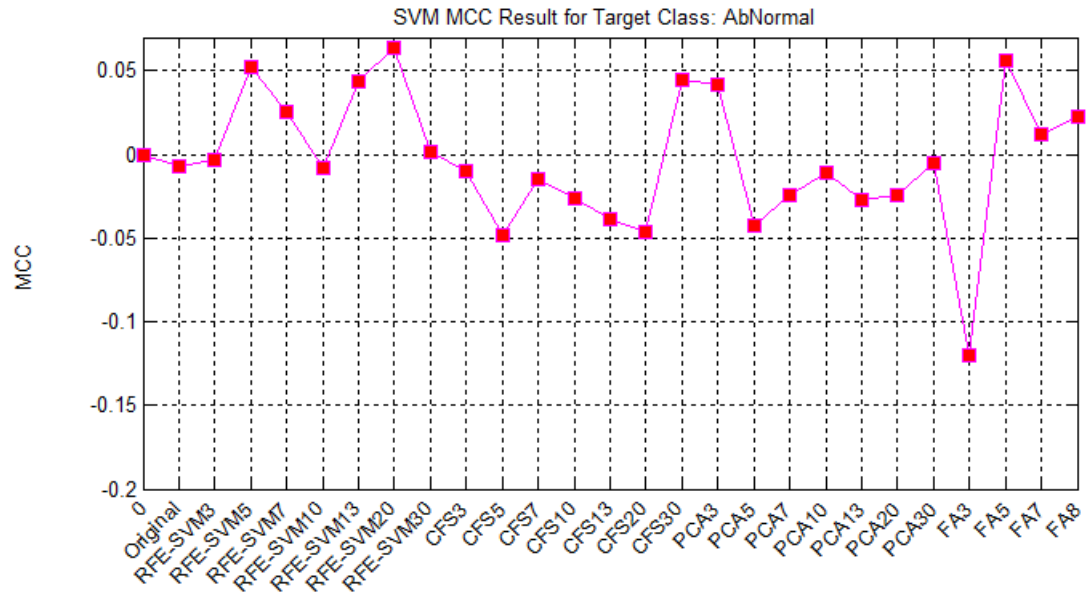

Figure 4.6. MCC test results of SVM.

probabilities, majority voting, minimum probability, and maximum probability. Majority Voting has worse performance among others. Besides the original dataset, this combination is applied to datasets RFE-SVM$_{13}$ and CFS$_{20}$ that k-NN and SVM has better performance in terms of sensitivity for target class: abnormal. For "weak" base-learner, k-NN where k=5, is used as it is more consistent among other trials for k value. Cost and gamma parameters of SVM are set to optimum values for original, RFE-SVM$_{13}$ and CFS$_{20}$ datasets respectively. 10-fold cross-validation is used to test the combination model.

In experiments, for voted combination of SVM and k-NN, the highest sensitivity result which is 73.77% is provided with CFS$_{20}$ dataset. RFE-SVM$_{13}$ dataset is the second with 65% and the original dataset is the last with 60.70%.

As a consensus on the comparative result of voted combination of SVM and k-NN experiment with SVM and k-NN, it can be said that, combination of classifiers draw more consistent performance than individual performance. k-NN as a local classifier with smaller featured dataset may give general idea about the case and SVM as a global classifier with a few greater featured dataset may give more details. The voted combination of them can draw a more consistent picture which may help identify the case.

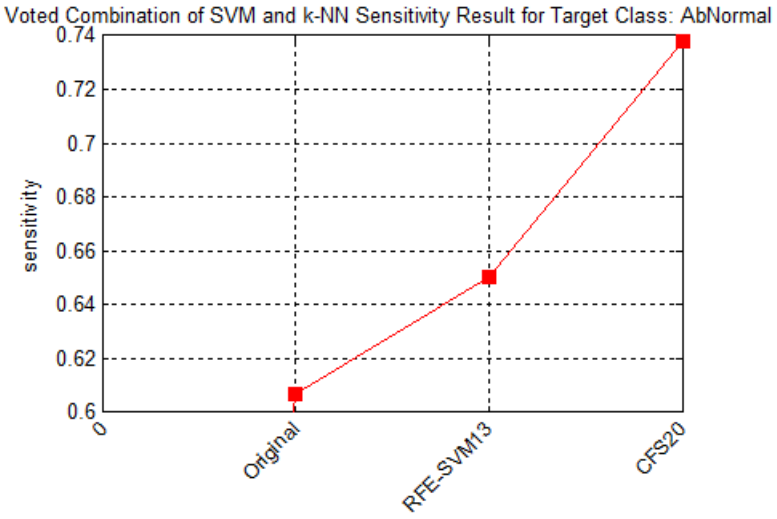

Figure 4.7. Sensitivity test results of voted combination of SVM and k-NN.

4.3.4. Stacked Generalization of SVM and k-NN

Stack meta-classifier in Weka is used to perform stacked generalization of SVM and k-NN. As a combiner, C4.5 decision tree algorithm is chosen which is called J48 in Weka. Optimum parameters for both of the base learners are chosen based on their individual performance. Datasets are also chosen which provide best performance on behalf of both classifiers. Thus, RFE-SVM$_{13}$ and CFS$_5$ datasets are used in addition to the original dataset in experiments.

According to results, the highest sensitivity is achieved with CFS$_{20}$ which is 73.20%. RFE-SVM$_{13}$ has provided 63.90% sensitivity and the original dataset has given sensitivity of 61.20%.

A consensus on the performance of stacked generalization of SVM and k-NN versus performance of SVM and k-NN individually may be said that combination of classifiers with stacked generalization is better than the individual performance. For instance, SVM achieves 58.97% for RFE-SVM$_{13}$ but is improved in stacked combination with k-NN and increases to 63.9%.

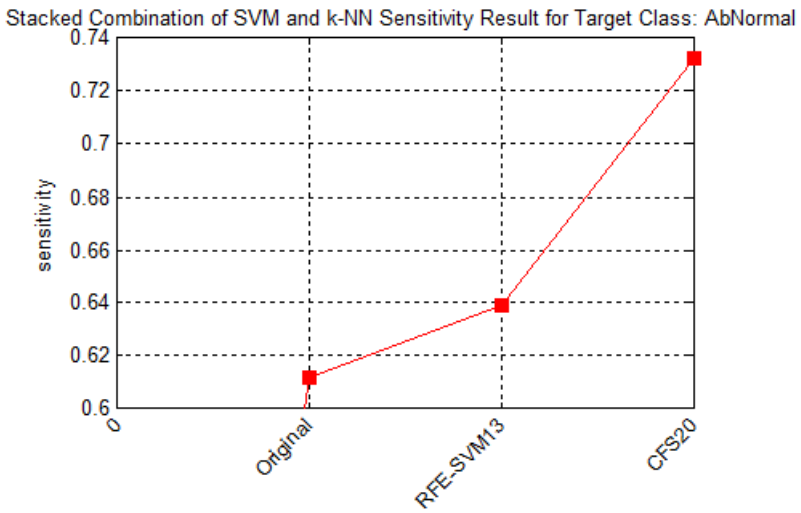

Figure 4.8. Sensitivity test results of stacked combination of SVM and k-NN.

As it can be seen from Table 4.4, combination usage of SVM and k-NN$_5$ has

Table 4.4. Classifier sensitivity comparison.

Method	Original	RFE-SVM$_{13}$	CFS$_{20}$
k-NN$_5$	0.3718	0.7179	0.5897
SVM	0.3061	0.3163	0.3571
Voted Comb. of SVM and k-NN$_5$	0.6066	**0.6503**	**0.7377**
Stacked Comb. of SVM and k-NN$_5$	0.6120	0.6393	**0.7322**

increased the sensitivity results. On original dataset, Stacked Combination of SVM and k-NN$_5$ performs better while Voted Combination of SVM and k-NN$_5$ achieves better results for RFE-SVM$_{13}$ and CFS$_{20}$ datasets.

4.3.5. Naive Bayes

The naive bayes is a simple probabilistic classifier based on Bayes' theorem. Naive Bayes classifier assumes inputs are independent given class. It ignores possible dependencies, namely, correlations, among the inputs and reduces a multivariate problem to a group of univariate problems [45].

In this study, Naive Bayes classifier is applied to each dataset using Matlab toolbox. Kernel density estimation is used to model the input attributes.

In experiments, FA$_5$ dataset has the highest sensitivity achieving 72.45%. An average sensitivity of 65.82% is achieved with FA datasets, and CFS comes the second with 54.22% sensitivity and PCA follows with an average sensitivity of 50.29%.

FA datasets outperform others. The average f-score value for FA is 0.735 which is higher than 0.684 for CFS and 0.592 for RFE-SVM. Individually, FA$_5$ dataset has the highest f-score result which is 0.789.

MCC results are similar to f-score results. The highest MCC result, 0.6404, is achieved by FA5 dataset. In the ranking for average MCC results, CFS dataset comes the first with 0.5732 MCC result, and FA has the second highest result with 0.5707,

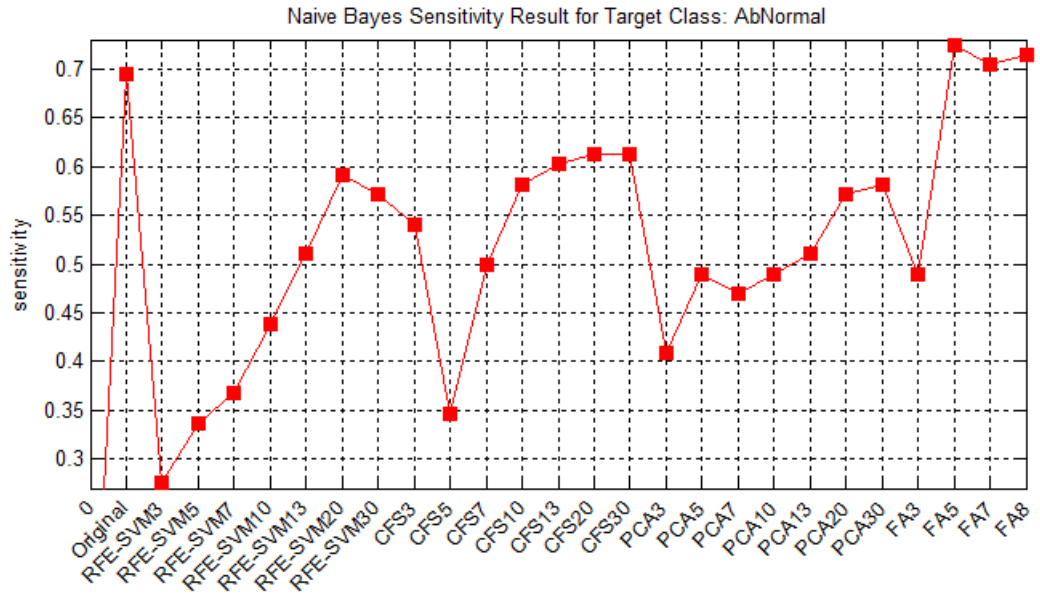

Figure 4.9. Sensitivity test results of Naive Bayes.

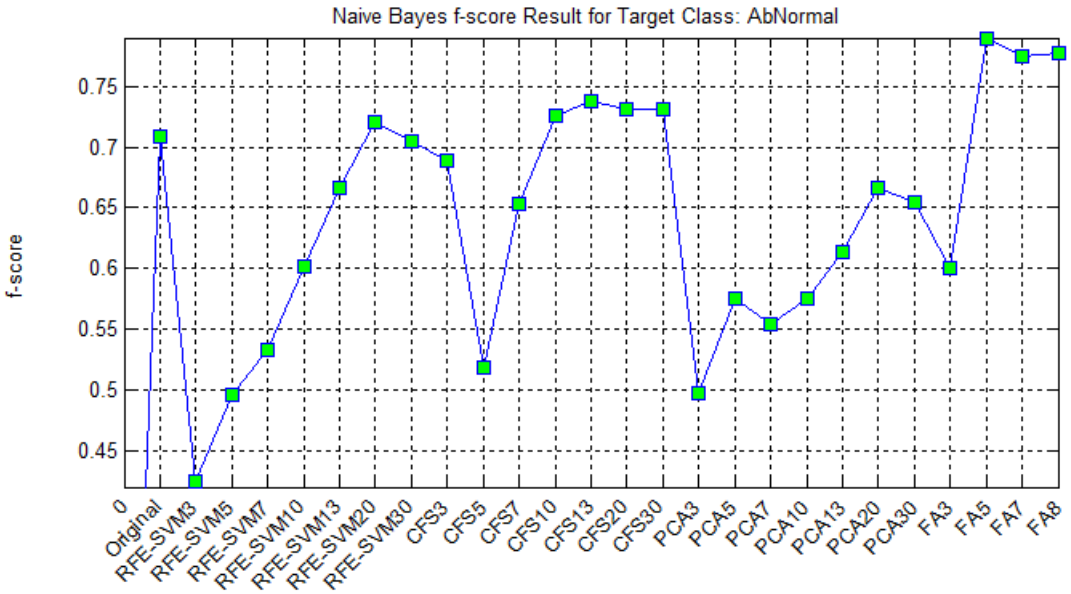

Figure 4.10. F-score test results of Naive Bayes.

whereas PCA is the last with 0.3515 MCC result.

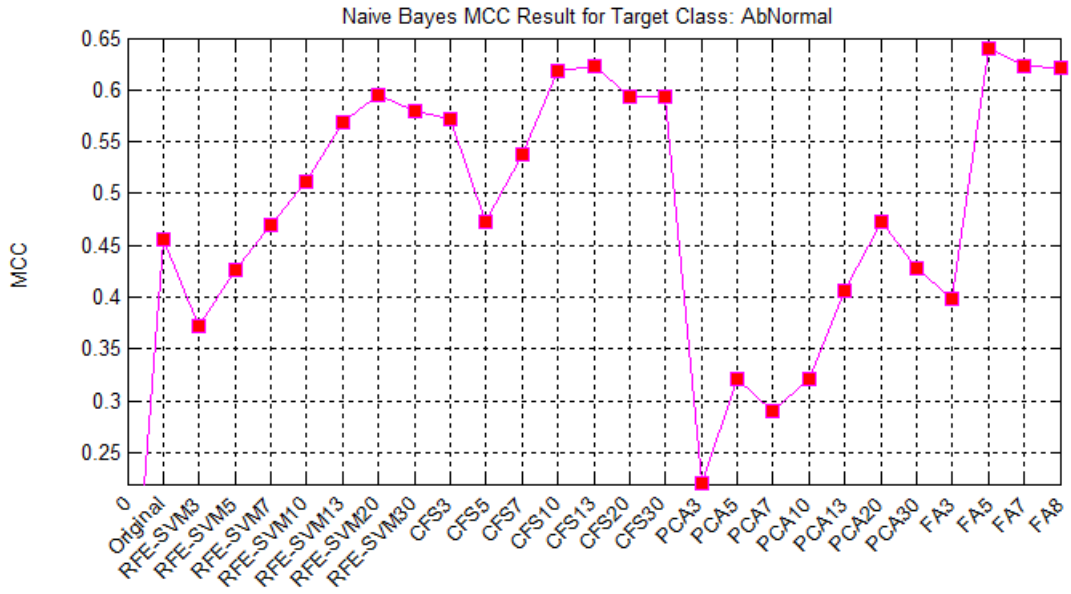

Figure 4.11. MCC test results of Naive Bayes.

4.3.6. BayesNet

The sensitivity results show that BayesNet has better performance with CFS datasets whilst has least performance with PCA datasets. The highest average which is 65.05% is provided for CFS. In this ranking, RFE-SVM is in the second order with average sensitivity of 64.44% and FA is the third with average sensitivity of 62.50%. However, FA_8 is the one which has achieved the highest sensitivity, 71.28% individually.

Maximum f-score which is 0.790 is provided for CFS_{20} dataset. CFS datasets also have the highest average f-score results among the others. The average f-score result of 0.749 is provided for CFS datasets. According to this order, RFE-SVM comes the second with average f-score result of 0.714 and FA follows with average f-score result of 0.702. PCA datasets have the least f-score result.

Similar to the f-score results, PCA is the last with average MCC result of 0.464

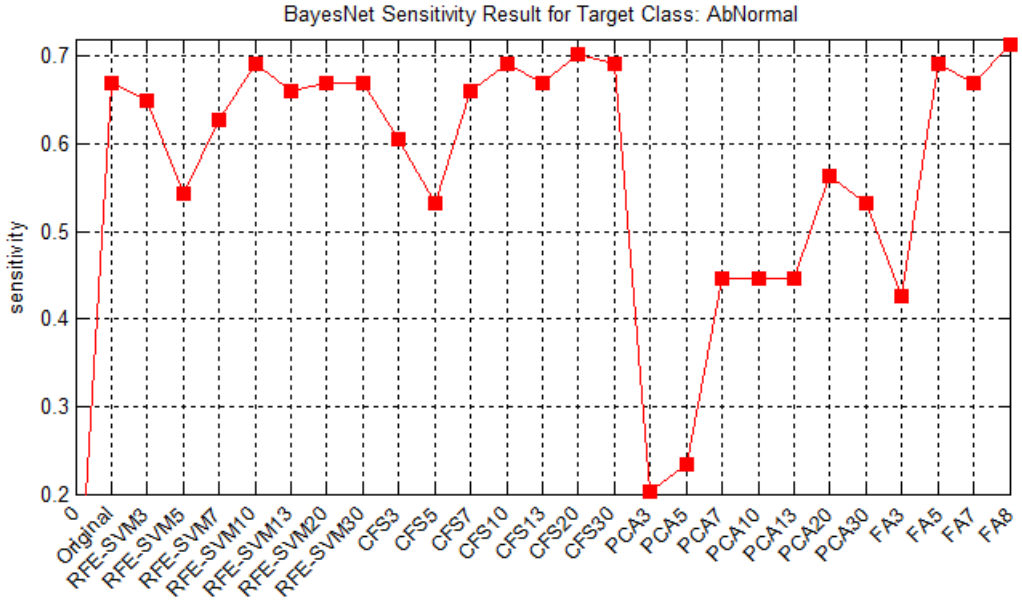

Figure 4.12. Sensitivity test results of BayesNet.

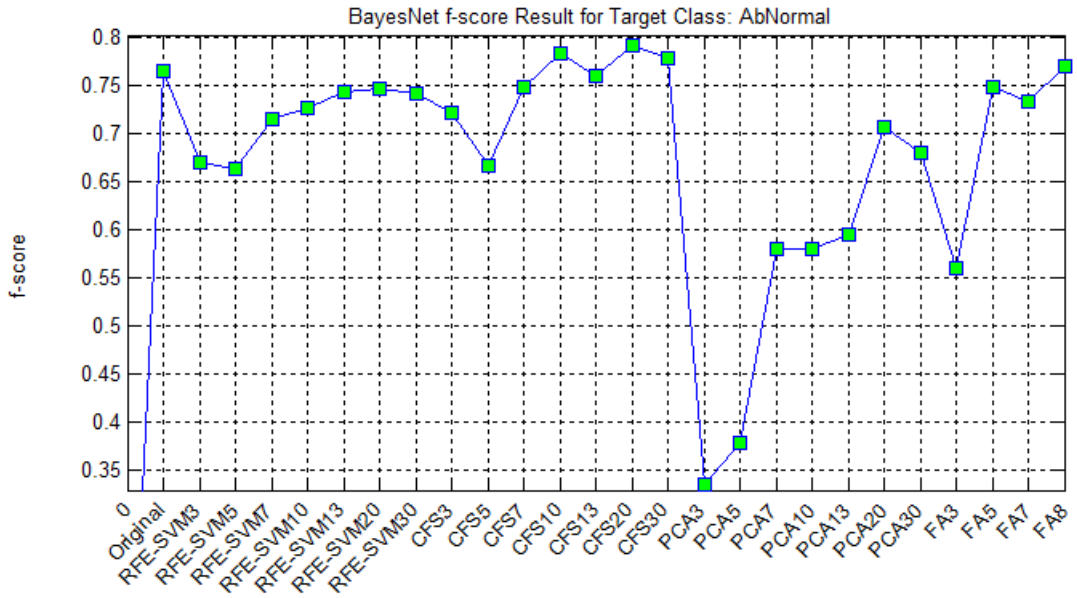

Figure 4.13. F-score test results of BayesNet.

whereas CFS is the first with average MCC result of 0.619. Individually, CFS_{20} dataset is the one which has the highest MCC result, 0.670.

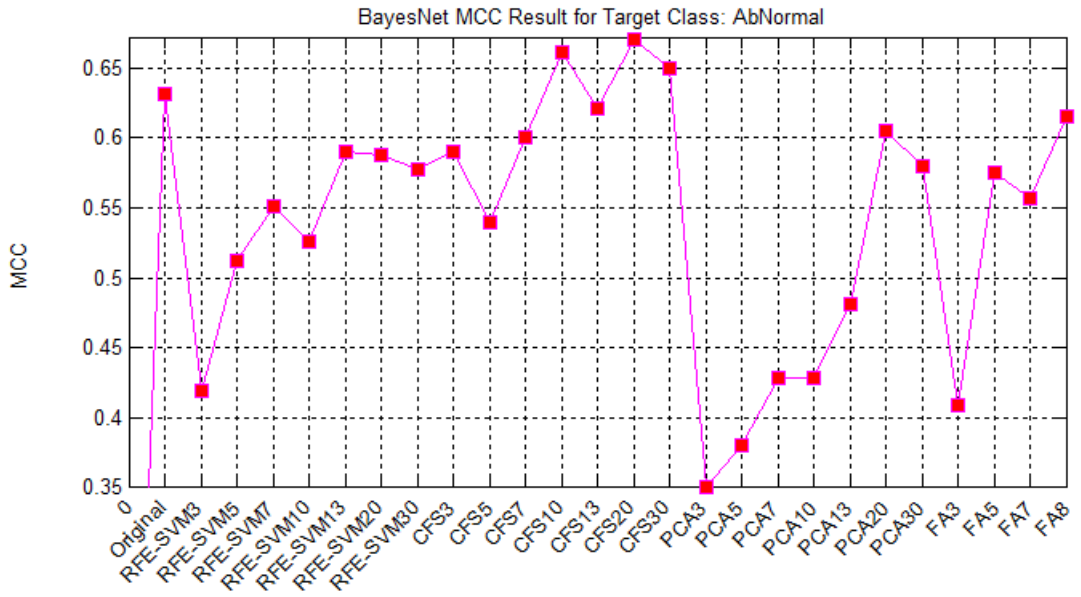

Figure 4.14. MCC test results of BayesNet.

4.3.7. Genetic Algorithm-Neural Network

In this proposed approach, genetic algorithm is used to assist neural network for performing feature selection and classification simultaneously. A population of neural networks with the same parameter set is initially created. The structure of the neural networks is comprised of one input layer, one hidden layer and one output layer. Each of them is trained with scaled conjugate gradient backpropagation algorithm, namely 'trainscg'. Mean squared error with regularization is used for performance function.Then genetic algorithm steps are applied to the population iteratively. A new generation is produced per iteration. During each cycle, neural network weights are applied crossover, mutation and mutation2 operators. Parent selection is performed based on roulette wheel selection. In experiments, crossover and mutation rates are chosen as 90% and 10% respectively. Fitness value of the offsprings is obtained from objective function and based on fitness value evaluation, probability of each solution to be in the next generation is computed. 90% of the new population is populated by

the offsprings with the highest probability. The rest of the population is completed by the best solutions from the old generation in order to carry the best characteristics of the ancestors.

In the experiments, 5% of the whole data is separated to be used as validation set for the final model. The algorithm is executed on various parameter sets to find out the optimal parameter set. 10-fold cross-validation is used per execution. As a result of each experiment, 10 candidate models are produced and an average and standart deviation of the accuracy are obtained. Thus, the parameter set and number of neurons in hidden layer are determined by evaluating the overall accuracy. The effect of the parameter changes on the overall accuracy is shown in Figure 4.15 and 4.16.

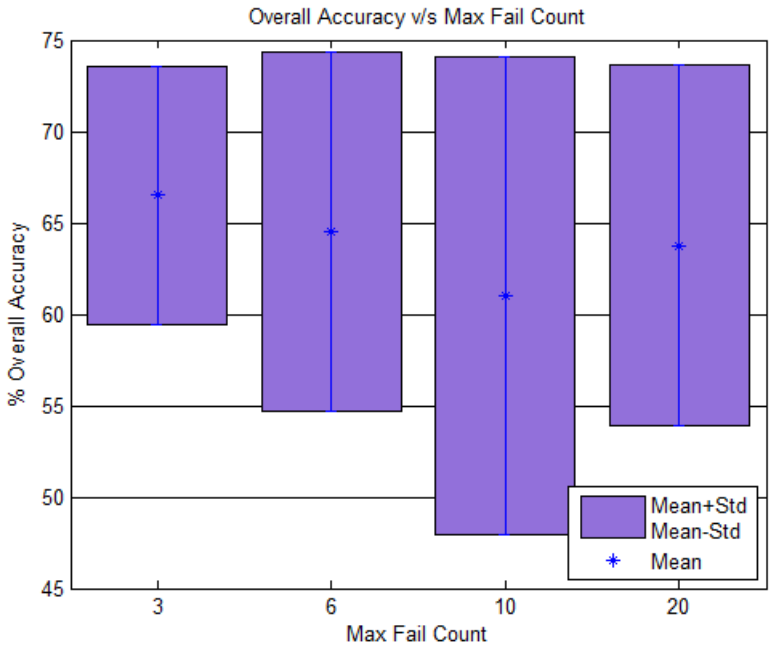

Figure 4.15. Max. fail selection versus overall accuracy.

The models with the selected parameter set are trained and tested using whole data. The model with the best performance is selected. This model selection is performed with the validation set which is separated before and without validation set.

According to the experiments, 86.75% of accuracy is obtained with the resulting

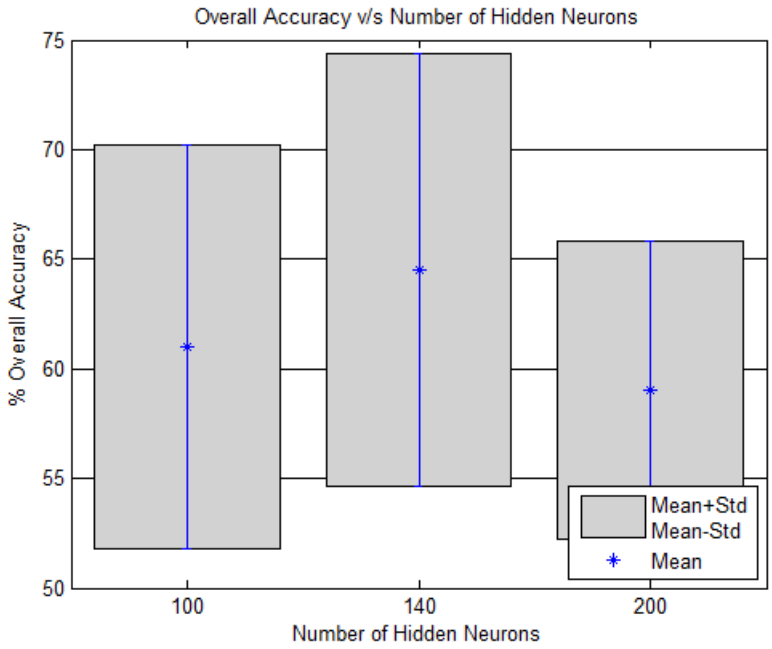

Figure 4.16. Number of neurons in hidden layer versus overall accuracy.

model. There is also approximately 95% decrease in the number of features. The original input feature number is 278 while the resulting input feature is 12. The list of the relevant features which are left unpruned in the final neural network is as follows:

- heartrate
- QRSduration
- V3_Avg_width_S_wave
- V1_Avg_QRSA
- V3_Amplitude_S_wave
- AVL_Amplitude_T_wave
- V2_Amplitude_R_wave
- DI_Avg_QRSTA
- V2_Avg_width_S_wave
- V1_N_intrinsic_deflections
- V2_N_intrinsic_deflections
- Tinterval

As representative datasets for each dimension reduction method: RFE-SVM$_{13}$, CFS$_{20}$, PCA$_{13}$ and FA$_8$ are selected. Sensitivity results for the abnormal class of classifiers are listed in Table 4.5.

Table 4.5. Classifier sensitivity comparison.

Method	**RFE-SVM$_{13}$**	**CFS$_{20}$**	**PCA$_{13}$**	**FA$_8$**	**Original**
k-NN$_5$	**0.7179**	0.5897	0.5897	0.6410	0.3718
SVM	0.3163	**0.3571**	0.2959	0.3061	0.3061
Naive Bayes	0.5102	0.6186	0.5102	0.7143	**0.7158**
BayesNet	0.6596	0.7021	0.4468	**0.7128**	0.6702
GA-NN	-	-	-	-	**0.9646**

According to the results, classifiers obtain their higher sensitivity result mostly with FA and CFS datasets. For 13-featured RFE-SVM dataset, k-NN$_5$ performs highest sensitivity. On original dataset, sensitivity results of SVM and k-NN$_5$ have increased when they are combined. Voted combination of SVM and k-NN$_5$ obtain 0.6066 and Stacked generalization of SVM and k-NN$_5$ achieve 0.6120 in terms of sensitivity. GA-NN obtains highest result on original dataset by reducing irrelevant features while performing classification. The classifiers with their best performance are summarized in Table 4.6.

Table 4.6. Classifier+Dataset sensitivity comparison.

k-NN$_5$ + RFE-SVM$_{13}$	**SVM+ CFS$_{20}$**	**Naive Bayes$_5$ + Original**	**BayesNet+ FA$_8$**	**Proposed+Original**
0.7179	0.3571	0.7158	0.7128	0.9646

Table 4.7 shows that GA-NN approach scores the highest value among other classifiers. BayesNet comes the second with 0.7904 f-score value. The highest score that the classifiers obtain is given in Table 4.8.

In terms of MCC results given in Table 4.9, GA-NN approach provides the highest performance with 0.7375. As overall; GA- NN, Naive Bayes and BayesNet perform

Table 4.7. Classifier f-score comparison.

Method	**RFE-SVM$_{13}$**	**CFS$_{20}$**	**PCA$_{13}$**	**FA$_8$**	**Original**
k-NN$_5$	**0.7044**	0.6715	0.5823	0.6329	0.4328
SVM	0.3875	**0.3933**	0.3558	0.3750	0.3681
Naive Bayes	0.6667	0.7317	0.6135	**0.7778**	0.7083
BayesNet	0.7425	**0.7904**	0.5957	0.7701	0.7636
GA-NN	-	-	-	-	**0.8916**

Table 4.8. Classifier+Dataset f-score comparison.

k-NN$_5$ + RFE-SVM$_{13}$	**SVM+ CFS$_{20}$**	**Naive Bayes$_5$ + FA$_8$**	**BayesNet+ CFS$_{20}$**	**Proposed+Original**
0.7044	0.3933	0.7778	0.7904	0.8916

Table 4.9. Classifier MCC comparison.

Method	**RFE-SVM$_{13}$**	**CFS$_{20}$**	**PCA$_{13}$**	**FA$_8$**	**Original**
k-NN$_5$	0.5247	**0.5281**	0.3305	0.4117	0.1827
SVM	**0.0432**	-0.0459	-0.0275	0.0223	-0.0069
Naive Bayes	0.5691	0.5942	0.4061	**0.6209**	0.4562
BayesNet	0.5897	**0.6701**	0.4816	0.6151	0.6320
GA-NN	-	-	-	-	**0.7375**

better for MCC than others. The highest MCC results of the classifiers are shown in Table 4.10.

Table 4.10. Classifier+Dataset MCC comparison.

k-NN$_5$ + CFS$_{20}$	**SVM+ RFE-SVM$_{13}$**	**Naive Bayes$_5$ + FA$_8$**	**BayesNet+ CFS$_{20}$**	**Proposed+Original**
0.5247	0.0432	0.6209	0.6701	0.7375

Table 4.11. Classifier accuracy comparison.

Method	**RFE-SVM$_{13}$**	**CFS$_{20}$**	**PCA$_{13}$**	**FA$_8$**	**Original**
k-NN$_5$	0.7762	**0.7857**	0.6857	0.7238	0.6381
SVM	**0.5333**	0.4857	0.5000	0.5238	0.5095
Naive Bayes	0.7619	0.7895	0.7000	**0.8095**	0.7295
BayesNet	0.7952	**0.8333**	0.7286	0.8095	0.8143
GA-NN	-	-	-	-	**0.8675**

When we consider the ratio of correctly classified instances, GA-NN comes the first achieving 86.75% accuracy. BayesNet follows with an accuracy of 83.33%. The highest accuracy rates captured by the classifiers are given in Table 4.12.

Table 4.12. Classifier+Dataset accuracy comparison.

k-NN$_5$ + CFS$_{20}$	**SVM+ RFE-SVM$_{13}$**	**Naive Bayes$_5$ + FA$_8$**	**BayesNet+ CFS$_{20}$**	**Proposed+Original**
0.7857	0.5333	0.8095	0.8333	0.8675

4.4. Experimental Results For Rule Extraction

We obtain 12-featured classified dataset as a result of the GA-NN approach. Based on the output, rules are extracted using C4.5, RIPPER, PART and HotSpot algorithms.

4.4.1. C4.5 Decision Tree

Decision tree results are obtained from J48 method in Weka which is a C4.5 decision tree implementation in java. This algorithm is applied to the dataset which has been correctly classified by the proposed method. The resulting rules are given in Table 4.13.

Table 4.13. C4.5 rule set.

1	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave > -0.8 and heartrate $\leq$ 58 then CLASS = ABNORMAL
2	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave > -0.8 and heartrate > 94 then CLASS = ABNORMAL
3	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave > -0.8 and 58 < heartrate $\leq$ 94 and V1_N_intrinsic_deflections > 28 then CLASS = ABNORMAL
4	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave > -0.8 and 58 < heartrate $\leq$ 94 and V1_N_intrinsic_deflections $\leq$ 28 and Tinterval > 218 and DI_Avg_QRSTA $\leq$ 25.5 then CLASS = ABNORMAL
5	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave > -0.8 and 58 < heartrate $\leq$ 94 and V1_N_intrinsic_deflections $\leq$ 28 and Tinterval $\leq$ 145 and QRSduration > 91 then CLASS = ABNORMAL
6	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave $\leq$ -0.8 and V3_Avg_width_S_wave > 56 then CLASS = ABNORMAL
7	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave $\leq$ -0.8 and V3_Avg_width_S_wave $\leq$ 56 and V2_Avg_width_S_wave $\leq$ 48 then CLASS = ABNORMAL
8	If V1_Avg_QRSA $\leq$ 1 and AVL_Amplitude_T_wave $\leq$ -0.8 and V3_Avg_width_S_wave $\leq$ 56 and V2_Avg_width_S_wave > 48 and V3_Amplitude_S_wave $\leq$ -14 then CLASS = ABNORMAL
9	If V1_Avg_QRSA > 1 then CLASS = ABNORMAL

The root of the resulting C4.5 tree is QRSA of the V1 Lead which is splitted into two branches according to being greater or less than 1. QRSA, the amplitude of T and S waves, the width of S wave, duration of T wave, heartrate and number of intrinsic deflections are determinant features in the resulting rule set.

4.4.2. Repeated Incremental Pruning to Produce Error Reduction

Repeated Incremental Pruning to Produce Error Reduction (RIPPER) is applied which is called JRip method in WEKA. Pruning is performed for rule base tree generation and stopping condition includes control for the error rate to be greater than 0.5. The generated rules are given in Table 4.14.

Table 4.14. RIPPER rule set.

1	If DI_Avg_QRSTA $\leq$ 15.7 and Tinterval $\geq$ 209 then CLASS = ABNORMAL
2	If QRSduration $\geq$ 91 and DI_Avg_QRSTA $\leq$ 19.7 and V1_N_intrinsic_deflections $\geq$ 24 then CLASS = ABNORMAL
3	If V1_Avg_QRSA $\geq$ -6.4 then CLASS = ABNORMAL
4	If heartrate $\leq$ 58 or heartrate $\geq$ 103 then CLASS = ABNORMAL
5	If QRSduration $\geq$ 92 and V3_Avg_width_S_wave $\leq$ 32 then CLASS = ABNORMAL
6	If Tinterval $\geq$ 195 and QRSduration $\geq$ 107 then CLASS = ABNORMAL
7	If V2_Amplitude_R_wave $\leq$ 1.2 and AVL_Amplitude_T_wave $\leq$ -0.3 then CLASS = ABNORMAL
8	If QRSduration $\geq$ 91 and Tinterval $\leq$ 143 and heartrate $\geq$ 66 then CLASS = ABNORMAL

According to resulting rule set, QRSTA of DI Lead, duration of T wave and QRS complex, amplitude of R and T waves, width of S wave, QRSA of V1 wave, number of intrinsic deflections and heartrate are selected as distinguishing features.

4.4.3. Partial Decision Tree

Partial Decision Tree is applied to dataset in WEKA. Unpruned tree is constructed. The resulting rules are given in Table 4.15.

The resulting rule set consists of 9 rules. QRSA of V1 Lead, amplitude of S and T waves, heartrate, duration of T wave and QRS complex, width of S wave, QRSTA of D1 Lead and number of intrinsic deflections are selected features. The resulting rules seem consistent with C4.5 rules. The rules are not in a successive order because Part

Table 4.15. PART rule set.

1	If V1_Avg_QRSA > 1 then CLASS = ABNORMAL
2	If AVL_Amplitude_T_wave ≤ -0.8 and V1_N_intrinsic_deflections ≤ 8 then CLASS = ABNORMAL
3	If heartrate ≤ 57 and Tinterval ≤ 165 then CLASS = ABNORMAL
4	If Tinterval > 221 and DI_Avg_QRSTA ≤ 25.5 then CLASS = ABNORMAL
5	If heartrate > 94 and Tinterval > 148 then CLASS = ABNORMAL
6	If V3_Avg_width_S_wave ≤ 28 and V3_Amplitude_S_wave > -6.8 and DI_Avg_QRSTA > 17.7 then CLASS = ABNORMAL
7	If V1_N_intrinsic_deflections > 24 and V3_Avg_width_S_wave > 40 then CLASS = ABNORMAL
8	If QRSduration > 107 and V1_Avg_QRSA ≤ -25 then CLASS = ABNORMAL
9	If V2_Avg_width_S_wave > 44 and V1_N_intrinsic_deflections > 4 and heartrate > 64 and V3_Avg_width_S_wave ≤ 56 then CLASS = ABNORMAL

produces rules from the repeated generated partial decision trees.

4.4.4. HotSpot

HotSpot algorithm is applied in order to find rules with the item of interest. The target class is abnormal and the produced rules cover the samples of abnormal class. In this study, HotSpot algorithm in Weka is used which examines the input data and learns the association rules corresponding to the target class. The rules are also displayed in a tree like structure. The produced rule set is given in Table 4.16.

Table 4.16. HotSpot rule set.

1	If Tinterval > 121 and QRSduration > 71 and DI_Avg_QRSTA ≤ 26.8 then CLASS = ABNORMAL
2	If Tinterval > 126 and DI_Avg_QRSTA ≤ 26.8 then CLASS = ABNORMAL
3	If DI_Avg_QRSTA ≤ 40.8 and QRSduration > 82 then CLASS = ABNORMAL
4	If Tinterval > 121 and QRSduration > 82 and DI_Avg_QRSTA ≤ 42.1 then CLASS = ABNORMAL

4.4.5. Rule Set Evaluation

The performance measures including accuracy, sensitivity, f-score, precision and MCC are computed for the rule extraction methods. The comparison of the rule extraction methods based on the performance measures is given in Table 4.17.

Table 4.17. Performance results of rule sets.

Method	**Accuracy**	**Sensitivity**	**F-score**	**Precision**	**MCC**
C4.5	87.8%	81%	85.2%	89.8%	75%
RIPPER	88.5%	85.6 %	86.6%	87.6%	76.5%
PART	**94.2**	87.9%	93%	98.7%	88.6%

Among these measures, accuracy indicates the ratio of correctly classified data to whole data while others focus on target class. The target class in this study refers to abnormal class. According to the measures, PART povides highest measures for all metrics. C4.5 and RIPPER have similarities but RIPPER outperforms C4.5 especially in terms of sensitivity which is related to its ability to identify target class.

5. EXPERIMENTS WITH REAL ECG DATA

The proposed method is compared with other classifiers using UCI Arrhythmia dataset in previous sections. Results showed that GA-NN approach outperforms others in terms of different performance metrics. The proposed method is also validated with a real ECG dataset. The real ECG data is obtained from Kardiosis ECG Tool of the manufacturing firm TEPA [43]. This dataset contains 20 instances and provides all the features of UCI Arrhythmia dataset. In this experiment, 12 selected features which are obtained as a result of the GA-NN approach are computed for validation. The results are listed in Table 5.1.

Table 5.1. Performance results on real ECG.

Accuracy	**Sensitivity**	**F-score**	**Precision**	**MCC**
85%	0.57	0.73	1	0.68

According to the results, the proposed method can be applied to real life ECG data with high accuracy rate.

6. CONCLUSION

In this study we propose GA-NN approach for feature selection and rule based arrhythmia classification. Genetic algorithm is used to assist neural network in feature selection while neural network performs classification. In experiments, UCI Arrhythmia dataset is used. Diverse classifiers are chosen for comparison. Experiments show that the proposed approach outperforms others. The number of features is also decreased from 278 to 12 as a result of feature selection performed by the proposed method.

A real ECG data set is also used for verification. The proposed method is run on the real data set and we obtain high accuracy rate.

Based on the reduced feature set, rule extraction methods are applied. C4.5, RIPPER, PART and HotSpot are used to perform rule extraction. PART outperforms C4.5 and RIPPER methods in terms of performance measures. HotSpot is employed to observe effective rules on target class which is abnormal class.

The experiments for the classifiers to be used in comparison are performed on original dataset and reduced data sets. The results show that dimension reduction has an optimized effect on classification. An increasing trend in sensitivity is seen among classifiers when reduced data sets are used. In addition to this, it is observed that performance of k-NN and SVM has increased when these techniques are combined as voting and stacked. Furthermore, it can be said that probabilistic techniques draw consistent performance in classification results.

The resulting rule set is considered to include determinant features which mostly contribute to the classification. For this reason, the generated rules may be useful in determination about the case to be cardiac arrhythmia.

The significance of data obtained from ECG cannot be undervalued. However, this is also important that interpretation of ECG in conjunction with the clinical as-

sessments will be more effective in arrhythmia diagnosis than deciding only on the basis of ECG data.

APPENDIX A: THE OFFICIAL DATASET DESCRIPTION OF THE UCI ARRHYTHMIA DATASET

1. Title: Cardiac Arrhythmia Database

2. Sources:

(a) Original owners od Database:

-- 1. H. Altay Guvenir, PhD.,
Bilkent University,
Department of Computer Engineering and Information Science,
06533 Ankara, Turkey
Phone: +90 (312) 266 4133
Email: guvenir@cs.bilkent.edu.tr

-- 2. Burak Acar, M.S.,
Bilkent University,
EE Eng. Dept.
06533 Ankara, Turkey
Email: buraka@ee.bilkent.edu.tr

-- 3. Haldun Muderrisoglu, M.D., Ph.D.,
Baskent University,
School of Medicine
Ankara, Turkey

(b) Donor: H. Altay Guvenir
Bilkent University,
Department of Computer Engineering and Information Science,
06533 Ankara, Turkey
Phone: +90 (312) 266 4133

Email: guvenir@cs.bilkent.edu.tr

(c) Date: January, 1998

3. Past Usage:

1. H. Altay Guvenir, Burak Acar, Gulsen Demiroz, Ayhan Cekin
"A Supervised Machine Learning Algorithm for Arrhythmia Analysis"
Proceedings of the Computers in Cardiology Conference,
Lund, Sweden, 1997.

The aim is to determine the type of arrhythmia from
the ECG recordings.

4. Relevant Information:

This database contains 279 attributes, 206 of which are linear
valued and the rest are nominal.

Concerning the study of H. Altay Guvenir: "The aim is to distinguish
between the presence and absence of cardiac arrhythmia and to
classify it in one of the 16 groups. Class 01 refers to 'normal'
ECG classes 02 to 15 refers to different classes of arrhythmia
and class 16 refers to the rest of unclassified ones. For the
time being, there exists a computer program that makes such a
classification. However there are differences between the
cardiolog's and the programs classification. Taking the
cardiolog's as a gold standard we aim to minimise this difference
by means of machine learning tools."

The names and id numbers of the patients were recently
removed from the database.

5. Number of Instances: 452

6. Number of Attributes: 279

7. Attribute Information:

-- Complete attribute documentation:

1 Age: Age in years , linear

2 Sex: Sex (0 = male; 1 = female) , nominal

3 Height: Height in centimeters , linear

4 Weight: Weight in kilograms , linear

5 QRS duration: Average of QRS duration in msec., linear

6 P-R interval: Average duration between onset of P and Q waves in msec., linear

7 Q-T interval: Average duration between onset of Q and offset of T waves in msec., linear

8 T interval: Average duration of T wave in msec., linear

9 P interval: Average duration of P wave in msec., linear

Vector angles in degrees on front plane of:, linear

10 QRS

11 T

12 P

13 QRST

14 J

15 Heart rate: Number of heart beats per minute ,linear

Of channel DI:

Average width, in msec., of: linear

16 Q wave

17 R wave

18 S wave

19 R' wave, small peak just after R

20 S' wave

```
 21 Number of intrinsic deflections, linear

 22 Existence of ragged R wave, nominal
 23 Existence of diphasic derivation of R wave, nominal
 24 Existence of ragged P wave, nominal
 25 Existence of diphasic derivation of P wave, nominal
 26 Existence of ragged T wave, nominal
 27 Existence of diphasic derivation of T wave, nominal

Of channel DII:
 28 .. 39 (similar to 16 .. 27 of channel DI)
Of channels DIII:
 40 .. 51
Of channel AVR:
 52 .. 63
Of channel AVL:
 64 .. 75
Of channel AVF:
 76 .. 87
Of channel V1:
 88 .. 99
Of channel V2:
 100 .. 111
Of channel V3:
 112 .. 123
Of channel V4:
 124 .. 135
Of channel V5:
 136 .. 147
Of channel V6:
 148 .. 159
```

```
Of channel DI:
 Amplitude , * 0.1 milivolt, of
 160 JJ wave, linear
 161 Q wave, linear
 162 R wave, linear
 163 S wave, linear
 164 R' wave, linear
 165 S' wave, linear
 166 P wave, linear
 167 T wave, linear

 168 QRSA , Sum of areas of all segments divided by 10,
     ( Area= width * height / 2 ), linear
 169 QRSTA = QRSA + 0.5 * width of T wave * 0.1 * height of T
     wave. (If T is diphasic then the bigger segment is
     considered), linear

Of channel DII:
 170 .. 179
Of channel DIII:
 180 .. 189
Of channel AVR:
 190 .. 199
Of channel AVL:
 200 .. 209
Of channel AVF:
 210 .. 219
Of channel V1:
 220 .. 229
Of channel V2:
 230 .. 239
```

Of channel V3:

240 .. 249

Of channel V4:

250 .. 259

Of channel V5:

260 .. 269

Of channel V6:

270 .. 279

8. Missing Attribute Values: Several. Distinguished with '?'.

9. Class Distribution:

Database Arrhythmia

Class code :	Class :	Number:
01	Normal	245
02	Ischemic changes (Coronary Artery Disease)	44
03	Old Anterior Myocardial Infarction	15
04	Old Inferior Myocardial Infarction	15
05	Sinus tachycardy	13
06	Sinus bradycardy	25
07	Ventricular Premature Contraction (PVC)	3
08	Supraventricular Premature Contraction	2
09	Left bundle branch block	9
10	Right bundle branch block	50
11	1. degree AtrioVentricular block	0
12	2. degree AV block	0
13	3. degree AV block	0
14	Left ventricule hypertrophy	4
15	Atrial Fibrillation or Flutter	5
16	Others	22

APPENDIX B: ALL PERFORMANCE RESULTS OF K-NN

- Target (Positive) class is abnormal
- Accuracy indicates the ratio of correctly classified instances to total instances

Table B.1.

kNN_5 Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	29	49	105	27	0.6381	0.5179	0.3718	0.4328	0.7955	0.1827
$RFESVM_3$	41	37	106	26	0.7000	0.6119	0.5256	0.5655	0.8030	0.3407
$RFESVM_5$	49	29	108	24	0.7476	0.6712	0.6282	0.6490	0.8182	0.4529
$RFESVM_7$	54	24	110	22	0.7810	0.7105	0.6923	0.7013	0.8333	0.5285
$RFESVM_{10}$	51	27	113	19	0.7810	0.7286	0.6538	0.6892	0.8561	0.5227
$RFESVM_{13}$	56	22	107	25	0.7762	0.6914	0.7179	0.7044	0.8106	0.5247
$RFESVM_{20}$	54	24	112	20	0.7905	0.7297	0.6923	0.7105	0.8485	0.5470
$RFESVM_{30}$	47	31	116	16	0.7762	0.7460	0.6026	0.6667	0.8788	0.5075
CFS_3	51	27	97	35	0.7048	0.5930	0.6538	0.6220	0.7348	0.3819
CFS_5	55	23	102	30	0.7476	0.6471	0.7051	0.6748	0.7727	0.4704
CFS_7	53	25	114	18	0.7952	0.7465	0.6795	0.7114	0.8636	0.5547
CFS_{10}	48	30	118	14	0.7905	0.7742	0.6154	0.6857	0.8939	0.5395
CFS_{13}	50	28	117	15	0.7952	0.7692	0.6410	0.6993	0.8864	0.5512
CFS_{20}	46	32	119	13	0.7857	0.7797	0.5897	0.6715	0.9015	0.5281
CFS_{30}	45	33	114	18	0.7571	0.7143	0.5769	0.6383	0.8636	0.4645
PCA_3	51	27	87	45	0.6571	0.5313	0.6538	0.5862	0.6591	0.3035
PCA_5	53	25	74	58	0.6048	0.4775	0.6795	0.5608	0.5606	0.2324
PCA_7	47	31	87	45	0.6381	0.5109	0.6026	0.5529	0.6591	0.2548
PCA_{10}	45	33	97	35	0.6762	0.5625	0.5769	0.5696	0.7348	0.3102
PCA_{13}	46	32	98	34	0.6857	0.5750	0.5897	0.5823	0.7424	0.3305
PCA_{20}	40	38	100	32	0.6667	0.5556	0.5128	0.5333	0.7576	0.2753
PCA_{30}	35	43	100	32	0.6429	0.5224	0.4487	0.4828	0.7576	0.2139
FA_3	50	28	88	44	0.6571	0.5319	0.6410	0.5814	0.6667	0.2990
FA_5	52	26	104	28	0.7429	0.6500	0.6667	0.6582	0.7879	0.4523
FA_7	51	27	106	26	0.7476	0.6623	0.6538	0.6581	0.8030	0.4581
FA_8	50	28	102	30	0.7238	0.6250	0.6410	0.6329	0.7727	0.4117

Table B.2.

kNN_7 Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	30	48	112	20	0.6762	0.6000	0.3846	0.4688	0.8485	0.2644
$RFESVM_3$	41	37	111	21	0.7238	0.6613	0.5256	0.5857	0.8409	0.3883
$RFESVM_5$	54	24	108	24	0.7714	0.6923	0.6923	0.6923	0.8182	0.5105
$RFESVM_7$	54	24	112	20	0.7905	0.7297	0.6923	0.7105	0.8485	0.5470
$RFESVM_{10}$	52	26	112	20	0.7810	0.7222	0.6667	0.6933	0.8485	0.5244
$RFESVM_{13}$	51	27	111	21	0.7714	0.7083	0.6538	0.6800	0.8409	0.5036
$RFESVM_{20}$	51	27	114	18	0.7857	0.7391	0.6538	0.6939	0.8636	0.5323
$RFESVM_{30}$	43	35	118	14	0.7667	0.7544	0.5513	0.6370	0.8939	0.4838
CFS_3	49	29	93	39	0.6762	0.5568	0.6282	0.5904	0.7045	0.3259
CFS_5	55	23	108	24	0.7762	0.6962	0.7051	0.7006	0.8182	0.5220
CFS_7	55	23	115	17	0.8095	0.7639	0.7051	0.7333	0.8712	0.5867
CFS_{10}	46	32	118	14	0.7810	0.7667	0.5897	0.6667	0.8939	0.5173
CFS_{13}	50	28	117	15	0.7952	0.7692	0.6410	0.6993	0.8864	0.5512
CFS_{20}	45	33	121	11	0.7905	0.8036	0.5769	0.6716	0.9167	0.5393
CFS_{30}	44	34	118	14	0.7714	0.7586	0.5641	0.6471	0.8939	0.4950
PCA_3	54	24	81	51	0.6429	0.5143	0.6923	0.5902	0.6136	0.2957
PCA_5	42	36	84	48	0.6000	0.4667	0.5385	0.5000	0.6364	0.1707
PCA_7	45	33	87	45	0.6286	0.5000	0.5769	0.5357	0.6591	0.2304
PCA_{10}	40	38	94	38	0.6381	0.5128	0.5128	0.5128	0.7121	0.2249
PCA_{13}	41	37	96	36	0.6524	0.5325	0.5256	0.5290	0.7273	0.2536
PCA_{20}	35	43	104	28	0.6619	0.5556	0.4487	0.4965	0.7879	0.2495
PCA_{30}	36	42	102	30	0.6571	0.5455	0.4615	0.5000	0.7727	0.2438
FA_3	52	26	89	43	0.6714	0.5474	0.6667	0.6012	0.6742	0.3309
FA_5	52	26	105	27	0.7476	0.6582	0.6667	0.6624	0.7955	0.4609
FA_7	50	28	109	23	0.7571	0.6849	0.6410	0.6623	0.8258	0.4736
FA_8	52	26	105	27	0.7476	0.6582	0.6667	0.6624	0.7955	0.4609

Table B.3.

kNN_9 Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	26	52	114	18	0.6667	0.5909	0.3333	0.4262	0.8636	0.2339
$RFESVM_3$	42	36	106	26	0.7048	0.6176	0.5385	0.5753	0.8030	0.3526
$RFESVM_5$	50	28	108	24	0.7524	0.6757	0.6410	0.6579	0.8182	0.4645
$RFESVM_7$	53	25	115	17	0.8000	0.7571	0.6795	0.7162	0.8712	0.5645
$RFESVM_{10}$	51	27	115	17	0.7905	0.7500	0.6538	0.6986	0.8712	0.5422
$RFESVM_{13}$	50	28	115	17	0.7857	0.7463	0.6410	0.6897	0.8712	0.5310
$RFESVM_{20}$	50	28	115	17	0.7857	0.7463	0.6410	0.6897	0.8712	0.5310
$RFESVM_{30}$	43	35	123	9	0.7905	0.8269	0.5513	0.6615	0.9318	0.5408
CFS_3	49	29	99	33	0.7048	0.5976	0.6282	0.6125	0.7500	0.3746
CFS_5	56	22	108	24	0.7810	0.7000	0.7179	0.7089	0.8182	0.5334
CFS_7	53	25	111	21	0.7810	0.7162	0.6795	0.6974	0.8409	0.5264
CFS_{10}	44	34	117	15	0.7667	0.7458	0.5641	0.6423	0.8864	0.4843
CFS_{13}	48	30	122	10	0.8095	0.8276	0.6154	0.7059	0.9242	0.5832
CFS_{20}	44	34	121	11	0.7857	0.8000	0.5641	0.6617	0.9167	0.5284
CFS_{30}	41	37	120	12	0.7667	0.7736	0.5256	0.6260	0.9091	0.4836
PCA_3	53	25	86	46	0.6619	0.5354	0.6795	0.5989	0.6515	0.3204
PCA_5	43	35	83	49	0.6000	0.4674	0.5513	0.5059	0.6288	0.1754
PCA_7	44	34	89	43	0.6333	0.5057	0.5641	0.5333	0.6742	0.2338
PCA_{10}	41	37	99	33	0.6667	0.5541	0.5256	0.5395	0.7500	0.2788
PCA_{13}	39	39	100	32	0.6619	0.5493	0.5000	0.5235	0.7576	0.2631
PCA_{20}	32	46	100	32	0.6286	0.5000	0.4103	0.4507	0.7576	0.1762
PCA_{30}	34	44	103	29	0.6524	0.5397	0.4359	0.4823	0.7803	0.2280
FA_3	48	30	99	33	0.7000	0.5926	0.6154	0.6038	0.7500	0.3627
FA_5	48	30	108	24	0.7429	0.6667	0.6154	0.6400	0.8182	0.4414
FA_7	50	28	112	20	0.7714	0.7143	0.6410	0.6757	0.8485	0.5017
FA_8	53	25	112	20	0.7857	0.7260	0.6795	0.7020	0.8485	0.5357

Table B.4.

kNN_{11} Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	25	53	113	19	0.6571	0.5682	0.3205	0.4098	0.8561	0.2096
$RFESVM_3$	41	37	111	21	0.7238	0.6613	0.5256	0.5857	0.8409	0.3883
$RFESVM_5$	49	29	109	23	0.7524	0.6806	0.6282	0.6533	0.8258	0.4621
$RFESVM_7$	51	27	116	16	0.7952	0.7612	0.6538	0.7034	0.8788	0.5522
$RFESVM_{10}$	48	30	116	16	0.7810	0.7500	0.6154	0.6761	0.8788	0.5187
$RFESVM_{13}$	50	28	119	13	0.8048	0.7937	0.6410	0.7092	0.9015	0.5721
$RFESVM_{20}$	47	31	119	13	0.7905	0.7833	0.6026	0.6812	0.9015	0.5392
$RFESVM_{30}$	39	39	120	12	0.7571	0.7647	0.5000	0.6047	0.9091	0.4610
CFS_3	48	30	96	36	0.6857	0.5714	0.6154	0.5926	0.7273	0.3380
CFS_5	55	23	108	24	0.7762	0.6962	0.7051	0.7006	0.8182	0.5220
CFS_7	50	28	113	19	0.7762	0.7246	0.6410	0.6803	0.8561	0.5114
CFS_{10}	40	38	120	12	0.7619	0.7692	0.5128	0.6154	0.9091	0.4723
CFS_{13}	45	33	120	12	0.7857	0.7895	0.5769	0.6667	0.9091	0.5281
CFS_{20}	41	37	123	9	0.7810	0.8200	0.5256	0.6406	0.9318	0.5190
CFS_{30}	39	39	121	11	0.7619	0.7800	0.5000	0.6094	0.9167	0.4727
PCA_3	52	26	90	42	0.6762	0.5532	0.6667	0.6047	0.6818	0.3386
PCA_5	41	37	87	45	0.6095	0.4767	0.5256	0.5000	0.6591	0.1815
PCA_7	41	37	92	40	0.6333	0.5062	0.5256	0.5157	0.6970	0.2210
PCA_{10}	39	39	95	37	0.6381	0.5132	0.5000	0.5065	0.7197	0.2209
PCA_{13}	37	41	102	30	0.6619	0.5522	0.4744	0.5103	0.7727	0.2561
PCA_{20}	37	41	101	31	0.6571	0.5441	0.4744	0.5068	0.7652	0.2473
PCA_{30}	34	44	107	25	0.6714	0.5763	0.4359	0.4964	0.8106	0.2650
FA_3	52	26	94	38	0.6952	0.5778	0.6667	0.6190	0.7121	0.3698
FA_5	51	27	112	20	0.7762	0.7183	0.6538	0.6846	0.8485	0.5131
FA_7	48	30	116	16	0.7810	0.7500	0.6154	0.6761	0.8788	0.5187
FA_8	47	31	113	19	0.7619	0.7121	0.6026	0.6528	0.8561	0.4774

Table B.5.

kNN_{13} Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	23	55	117	15	0.6667	0.6053	0.2949	0.3966	0.8864	0.2275
$RFESVM_3$	40	38	114	18	0.7333	0.6897	0.5128	0.5882	0.8636	0.4068
$RFESVM_5$	48	30	109	23	0.7476	0.6761	0.6154	0.6443	0.8258	0.4506
$RFESVM_7$	50	28	117	15	0.7952	0.7692	0.6410	0.6993	0.8864	0.5512
$RFESVM_{10}$	44	34	119	13	0.7762	0.7719	0.5641	0.6519	0.9015	0.5059
$RFESVM_{13}$	50	28	118	14	0.8000	0.7813	0.6410	0.7042	0.8939	0.5616
$RFESVM_{20}$	45	33	122	10	0.7952	0.8182	0.5769	0.6767	0.9242	0.5508
$RFESVM_{30}$	38	40	125	7	0.7762	0.8444	0.4872	0.6179	0.9470	0.5112
CFS_3	49	29	100	32	0.7095	0.6049	0.6282	0.6164	0.7576	0.3829
CFS_5	59	19	114	18	0.8238	0.7662	0.7564	0.7613	0.8636	0.6217
CFS_7	50	28	115	17	0.7857	0.7463	0.6410	0.6897	0.8712	0.5310
CFS_{10}	39	39	118	14	0.7476	0.7358	0.5000	0.5954	0.8939	0.4382
CFS_{13}	45	33	120	12	0.7857	0.7895	0.5769	0.6667	0.9091	0.5281
CFS_{20}	40	38	123	9	0.7762	0.8163	0.5128	0.6299	0.9318	0.5080
CFS_{30}	35	43	123	9	0.7524	0.7955	0.4487	0.5738	0.9318	0.4518
PCA_3	52	26	92	40	0.6857	0.5652	0.6667	0.6118	0.6970	0.3541
PCA_5	38	40	89	43	0.6048	0.4691	0.4872	0.4780	0.6742	0.1602
PCA_7	39	39	93	39	0.6286	0.5000	0.5000	0.5000	0.7045	0.2045
PCA_{10}	35	43	97	35	0.6286	0.5000	0.4487	0.4730	0.7348	0.1882
PCA_{13}	34	44	104	28	0.6571	0.5484	0.4359	0.4857	0.7879	0.2370
PCA_{20}	37	41	109	23	0.6952	0.6167	0.4744	0.5362	0.8258	0.3210
PCA_{30}	32	46	110	22	0.6762	0.5926	0.4103	0.4848	0.8333	0.2693
FA_3	46	32	100	32	0.6952	0.5897	0.5897	0.5897	0.7576	0.3473
FA_5	48	30	114	18	0.7714	0.7273	0.6154	0.6667	0.8636	0.4986
FA_7	46	32	117	15	0.7762	0.7541	0.5897	0.6619	0.8864	0.5067
FA_8	49	29	115	17	0.7810	0.7424	0.6282	0.6806	0.8712	0.5198

Table B.6.

kNN_{15} Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	26	52	116	16	0.6762	0.6190	0.3333	0.4333	0.8788	0.2562
$RFESVM_3$	42	36	110	22	0.7238	0.6563	0.5385	0.5915	0.8333	0.3903
$RFESVM_5$	46	32	111	21	0.7476	0.6866	0.5897	0.6345	0.8409	0.4464
$RFESVM_7$	50	28	118	14	0.8000	0.7813	0.6410	0.7042	0.8939	0.5616
$RFESVM_{10}$	44	34	120	12	0.7810	0.7857	0.5641	0.6567	0.9091	0.5170
$RFESVM_{13}$	47	31	124	8	0.8143	0.8545	0.6026	0.7068	0.9394	0.5956
$RFESVM_{20}$	44	34	124	8	0.8000	0.8462	0.5641	0.6769	0.9394	0.5636
$RFESVM_{30}$	37	41	127	5	0.7810	0.8810	0.4744	0.6167	0.9621	0.5273
CFS_3	50	28	97	35	0.7000	0.5882	0.6410	0.6135	0.7348	0.3700
CFS_5	59	19	112	20	0.8143	0.7468	0.7564	0.7516	0.8485	0.6033
CFS_7	48	30	113	19	0.7667	0.7164	0.6154	0.6621	0.8561	0.4887
CFS_{10}	37	41	121	11	0.7524	0.7708	0.4744	0.5873	0.9167	0.4499
CFS_{13}	41	37	122	10	0.7762	0.8039	0.5256	0.6357	0.9242	0.5069
CFS_{20}	38	40	124	8	0.7714	0.8261	0.4872	0.6129	0.9394	0.4983
CFS_{30}	33	45	123	9	0.7429	0.7857	0.4231	0.5500	0.9318	0.4287
PCA_3	49	29	88	44	0.6524	0.5269	0.6282	0.5731	0.6667	0.2868
PCA_5	36	42	89	43	0.5952	0.4557	0.4615	0.4586	0.6742	0.1354
PCA_7	40	38	94	38	0.6381	0.5128	0.5128	0.5128	0.7121	0.2249
PCA_{10}	38	40	96	36	0.6381	0.5135	0.4872	0.5000	0.7273	0.2169
PCA_{13}	35	43	103	29	0.6571	0.5469	0.4487	0.4930	0.7830	0.2404
PCA_{20}	33	45	108	24	0.6714	0.5789	0.4231	0.4889	0.8182	0.2621
PCA_{30}	27	51	110	22	0.6524	0.5510	0.3462	0.4252	0.8333	0.2050
FA_3	44	34	103	29	0.7000	0.6027	0.5641	0.5828	0.7803	0.3494
FA_5	47	31	114	18	0.7667	0.7231	0.6026	0.6573	0.8636	0.4873
FA_7	44	34	122	10	0.7905	0.8148	0.5641	0.6667	0.9242	0.5399
FA_8	46	32	116	16	0.7714	0.7419	0.5897	0.6571	0.8788	0.4963

APPENDIX C: ALL PERFORMANCE RESULTS OF SVM

Table C.1.

SVM Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	30	68	77	35	0.5095	0.4615	0.3061	0.3681	0.6875	-0.0069
$RFESVM_3$	19	79	90	22	0.5190	0.4634	0.1939	0.2734	0.8036	-0.0032
$RFESVM_5$	30	68	83	29	0.5381	0.5085	0.3061	0.3822	0.7411	0.0524
$RFESVM_7$	33	65	77	35	0.5238	0.4853	0.3367	0.3976	0.6875	0.0258
$RFESVM_{10}$	29	69	78	34	0.5095	0.4603	0.2959	0.3602	0.6964	-0.0083
$RFESVM_{13}$	31	67	81	31	0.5333	0.5000	0.3163	0.3875	0.7232	0.0432
$RFESVM_{20}$	31	67	83	29	0.5429	0.5167	0.3163	0.3924	0.7411	0.0634
$RFESVM_{30}$	29	69	79	33	0.5143	0.4677	0.2959	0.3625	0.7054	0.0014
CFS_3	28	70	79	33	0.5095	0.4590	0.2857	0.3522	0.7054	-0.0089
CFS_5	28	70	75	37	0.4905	0.4308	0.2857	0.3436	0.6696	-0.0482
CFS_7	31	67	75	37	0.5048	0.4559	0.3163	0.3735	0.6696	-0.0150
CFS_{10}	30	68	75	37	0.5000	0.4478	0.3061	0.3636	0.6696	-0.0259
CFS_{13}	28	70	76	36	0.4952	0.4375	0.2857	0.3457	0.6786	-0.0387
CFS_{20}	35	63	67	45	0.4857	0.4375	0.3571	0.3933	0.5982	-0.0459
CFS_{30}	32	66	80	32	0.5333	0.5000	0.3265	0.3951	0.7143	0.0442
PCA_3	29	69	83	29	0.5333	0.5000	0.2959	0.3718	0.7411	0.0413
PCA_5	26	72	78	34	0.4952	0.4333	0.2653	0.3291	0.6964	-0.0423
PCA_7	25	73	81	31	0.5048	0.4464	0.2551	0.3247	0.7232	-0.0245
PCA_{10}	27	71	80	32	0.5095	0.4576	0.2755	0.3439	0.7143	-0.0113
PCA_{13}	29	69	76	36	0.5000	0.4462	0.2959	0.3558	0.6786	-0.0275
PCA_{20}	31	67	74	38	0.5000	0.4493	0.3163	0.3713	0.6607	-0.0244
PCA_{30}	31	67	76	36	0.5095	0.4627	0.3163	0.3758	0.6786	-0.0055
FA_3	15	83	84	28	0.4714	0.3488	0.1531	0.2128	0.7500	-0.1198
FA_5	34	64	79	33	0.5381	0.5075	0.3469	0.4121	0.7054	0.0560
FA_7	30	68	79	33	0.5190	0.4762	0.3061	0.3727	0.7054	0.0125
FA_8	30	68	80	32	0.5238	0.4839	0.3061	0.3750	0.7143	0.0223

APPENDIX D: ALL PERFORMANCE RESULTS OF VOTED COMB. OF SVM AND K-NN

Table D.1.

Voted Combination Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	111	72	218	19	0.7833	0.8538	0.6066	0.7093	0.9198	0.5646
$RFESVM_{13}$	119	64	220	17	0.8071	0.8750	0.6503	0.7461	0.9283	0.6131
CFS_{20}	135	48	208	29	0.8167	0.8232	0.7377	0.7781	0.8776	0.6254

APPENDIX E: ALL PERFORMANCE RESULTS OF STACKED COMB. OF SVM AND K-NN

Table E.1.

Stacked Combination Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	112	71	217	20	0.7833	0.8485	0.6120	0.7111	0.9156	0.5636
$RFESVM_{13}$	117	66	220	17	0.8024	0.8731	0.6393	0.7382	0.9283	0.6038
CFS_{20}	134	49	208	29	0.8143	0.8221	0.7322	0.7746	0.8776	0.6206

APPENDIX F: ALL PERFORMANCE RESULTS OF NAIVE BAYES

Table F.1.

Naive Bayes Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	68	27	83	29	0.7295	0.7010	0.7158	0.7083	0.7411	0.4562
$RFESVM_3$	27	71	110	2	0.6524	0.9310	0.2755	0.4252	0.9821	0.3726
$RFESVM_5$	33	65	110	2	0.6810	0.9429	0.3367	0.4962	0.9821	0.4269
$RFESVM_7$	36	62	111	1	0.7000	0.9730	0.3673	0.5333	0.9911	0.4693
$RFESVM_{10}$	45	55	110	2	0.7286	0.9556	0.4388	0.6014	0.9821	0.5118
$RFESVM_{13}$	50	48	110	2	0.7619	0.9615	0.5102	0.6667	0.9821	0.5691
$RFESVM_{20}$	58	40	107	5	0.7857	0.9206	0.5918	0.7205	0.9554	0.5957
$RFESVM_{30}$	56	42	107	5	0.7762	0.9180	0.5714	0.7044	0.9554	0.5789
CFS_3	53	44	108	4	0.7703	0.9298	0.5464	0.6883	0.9643	0.5719
CFS_5	34	63	112	0	0.6986	1	0.3505	0.5191	1	0.4736
CFS_7	49	48	108	4	0.7512	0.9245	0.5052	0.6533	0.9643	0.5381
CFS_{10}	57	40	109	3	0.7943	0.9500	0.5876	0.7261	0.9732	0.6183
CFS_{13}	59	38	108	4	0.7990	0.9365	0.6082	0.7375	0.9643	0.6222
CFS_{20}	60	37	105	7	0.7895	0.8955	0.6186	0.7317	0.9375	0.5942
CFS_{30}	60	37	105	7	0.7895	0.8955	0.6186	0.7317	0.9375	0.5942
PCA_3	40	58	89	23	0.6143	0.6349	0.4082	0.4969	0.7946	0.2208
PCA_5	48	50	91	21	0.6619	0.6957	0.4898	0.5749	0.8125	0.3211
PCA_7	46	52	90	22	0..6476	0.6765	0.4694	0.5542	0.8036	0.2910
PCA_{10}	48	50	91	21	0.6619	0.6957	0.4898	0.5749	0.8125	0.3211
PCA_{13}	50	48	97	15	0.7000	0.7692	0.5102	0.6135	0.8661	0.4061
PCA_{20}	56	42	98	14	0.7333	0.8000	0.5714	0.6667	0.8750	0.4725
PCA_{30}	57	41	93	19	0.7143	0.7500	0.5816	0.6552	0.8304	0.4277
FA_3	48	50	98	14	0.6952	0.7742	0.4898	0.6000	0.8750	0.3990
FA_5	71	27	101	11	0.8190	0.8659	0.7245	0.7889	0.9018	0.6404
FA_7	69	29	101	11	0.8095	0.8625	0.7041	0.7753	0.9018	0.6224
FA_8	70	28	100	12	0.8095	0.8537	0.7143	0.7778	0.8929	0.6209

APPENDIX G: ALL PERFORMANCE RESULTS OF BAYESNET

Table G.1.

BayesNet Results										
Data set	**TP**	**FN**	**TN**	**FP**	**Accuracy**	**Precision**	**Recall**	**F-score**	**Spec.**	**MCC**
Original	63	31	108	8	0.8143	0.8873	0.6702	0.7636	0.9310	0.6320
$RFESVM_3$	61	33	89	27	0.7143	0.6932	0.6489	0.6703	0.7672	0.4194
$RFESVM_5$	51	43	107	9	0.7524	0.8500	0.5426	0.6623	0.9224	0.5118
$RFESVM_7$	59	35	104	12	0.7762	0.8310	0.6277	0.7152	0.8966	0.5510
$RFESVM_{10}$	65	29	96	20	0.7667	0.7647	0.6915	0.7263	0.8276	0.5258
$RFESVM_{13}$	62	32	105	11	0.7952	0.8493	0.6596	0.7425	0.9052	0.5897
$RFESVM_{20}$	63	31	104	12	0.7952	0.8400	0.6702	0.7456	0.8966	0.5882
$RFESVM_{30}$	63	31	103	13	0.7905	0.8289	0.6702	0.7412	0.8879	0.5775
CFS_3	57	37	109	7	0.7905	0.8906	0.6064	0.7215	0.9397	0.5899
CFS_5	50	44	110	6	0.7619	0.8929	0.5319	0.6667	0.9483	0.5399
CFS_7	62	32	106	10	0.8000	0.8611	0.6596	0.7470	0.9138	0.6006
CFS_{10}	65	29	109	7	0.8286	0.9028	0.6915	0.7831	0.9397	0.6612
CFS_{13}	63	31	107	9	0.8095	0.8750	0.6702	0.7590	0.9224	0.6208
CFS_{20}	66	28	109	7	0.8333	0.9041	0.7021	0.7904	0.9397	0.6701
CFS_{30}	65	29	108	8	0.8238	0.8904	0.6915	0.7784	0.9310	0.6500
PCA_3	19	75	116	0	0.6429	1	0.2021	0.3363	1	0.3504
PCA_5	22	72	116	0	0.6571	1	0.2340	0.3793	1	0.3800
PCA_7	42	52	107	9	0.7095	0.8235	0.4468	0.5793	0.9224	0.4282
PCA_{10}	42	52	107	9	0.7095	0.8235	0.4468	0.5793	0.9224	0.4282
PCA_{13}	42	52	111	5	0.7286	0.8936	0.4468	0.5957	0.9569	0.4816
PCA_{20}	53	41	113	3	0.7905	0.9464	0.5638	0.7067	0.9741	0.6049
PCA_{30}	50	44	113	3	0.7762	0.9434	0.5319	0.6803	0.9741	0.5793
FA_3	40	54	107	9	0.7000	0.8163	0.4255	0.5594	0.9224	0.4091
FA_5	65	29	101	15	0.7905	0.8125	0.6915	0.7471	0.8707	0.5756
FA_7	63	31	101	15	0.7810	0.8077	0.6702	0.7326	0.8707	0.5566
FA_8	67	27	103	13	0.8095	0.8375	0.7128	0.7701	0.8879	0.6151

REFERENCES

1. Yanowitz, F. G., *The Cardiac Conduction System Diagram - Marquette-KH*, `http://library.med.utah.edu/kw/ecg/mml/ecg_ccs.html`, accessed at June 2012.

2. Yanowitz, F. G., *Frontal and Horizontal Plane Lead Diagram-KH*, `http://library.med.utah.edu/kw/ecg/mml/ecg_torso.html`, accessed at June 2012.

3. *Normal ECG Image*, 2008, `http://www.swharden.com/blog/images/ecg2.jpg`, accessed at June 2012.

4. Murphy, P. and A. Asuncion, *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, 2010, `http://archive.ics.uci.edu/ml`, accessed at June 2012.

5. *MIT-BIH Arrhythmia Database*, 1997, `http://www.physionet.org/physiobank/database/html/mitdbdir/mitdbdir.htm`, accessed at June 2012.

6. Yeap, T. H., F. Johnson and M. Rachniowski, "ECG Beat Classification by a Neural Network", *Engineering in Medicine and Biology Society, 1990., Proceedings of the Twelfth Annual International Conference of the IEEE*, pp. 1457 –1458, 1990.

7. Hu, Y. H., W. J. Tompkins, J. L. Urrusti and V. X. Afonso, "Applications of Artificial Neural Networks for ECG Signal Detection and Classification", *Journal of Electrocardiology*, Vol. 26, pp. 66–73, 1993.

8. Silipo, R. and C. Marchesi, "Artificial Neural Networks for Automatic ECG Analysis", *IEEE Transactions on Signal Processing*, Vol. 46, No. 5, pp. 1417 –1425, 1998.

9. de Chazal, P. and R. Reilly, "A Comparison of the ECG Classification Performance of Different Feature Sets", *Computers in Cardiology 2000*, pp. 327 –330, 2000.

10. Frank, E., "An Accurate, Clinically Practical System for Spatial Vectorcardiography", *Circulation*, Vol. 13, No. 5, pp. 737–749, 1956.

11. Gao, D., M. Madden, D. Chambers and G. Lyons, "Bayesian ANN Classifier for ECG Arrhythmia Diagnostic System: a Comparison Study", *Proceedings of International Joint Conference on Neural Networks*, 2005.

12. Niwas, S. I., R. S. S. Kumari and V. Sadasivam, "Artificial Neural Network Based Automatic Cardiac Abnormalities Classification", *Proceedings of the Sixth International Conference on Computational Intelligence and Multimedia Applications*, ICCIMA '05, pp. 41–46, IEEE Computer Society, Washington, DC, USA, 2005.

13. Zhang, H. and L. Zhang, "ECG Analysis Based on PCA and Support Vector Machines", *International Conference on Neural Networks and Brain, 2005. ICNN&B '05.*, Vol. 2, pp. 743 –747, 2005.

14. Song, M. H., J. Lee, S. P. Cho, K. J. Lee and S. K. Yoo, "Support Vector Machine Based Arrhythmia Classification Using Reduced Features", *International Journal of Control, Automation and Systems*, Vol. 3, No. 4, 2005.

15. Uyar, A., *Intelligent Arrhythmia Classification Based on Support Vector Machines*, M.S., Boğaziçi University, 2006.

16. Kara, S. and M. Okandan, "Atrial Fibrillation Classification with Artificial Neural Networks", *Pattern Recognition*, Vol. 40, No. 11, pp. 2967 – 2973, 2007.

17. Asl, B. M., S. K. Setarehdan and M. Mohebbi, "Support Vector Machine-Based Arrhythmia Classification Using Reduced Features of Heart Rate Variability Signal", *Artificial Intelligence in Medicine*, Vol. 44, No. 1, pp. 51–64, 2008.

18. Oliveira, R. V. A., L. S. C. and M. Sarcinelli-Filho, "The Use of Bayesian Networks for Heart Beat Classification", A. Hussain, I. Aleksander, L. S. Smith, A. K. Barros, R. Chrisley and V. Cutsuridis (Editors), *Brain Inspired Cognitive Systems 2008*,

Vol. 657 of *Advances in Experimental Medicine and Biology*, pp. 217–231, Springer New York, 2010.

19. Özcan, N. Ö. and F. Gürgen, "Fuzzy Support Vector Machines for ECG Arrhythmia Detection", *Proceedings of the 2010 20th International Conference on Pattern Recognition*, ICPR '10, pp. 2973–2976, IEEE Computer Society, Washington, DC, USA, 2010.

20. Jadhav, S., S. L. Nalbalwar and A. A. Ghatol, "Artificial Neural Network Based Cardiac Arrhythmia Disease Diagnosis", *Process Automation, Control and Computing (PACC), 2011 International Conference on*, pp. 1 –6, 2011.

21. Homaeinezhad, M. R., S. A. Atyabi, E. Tavakkoli, H. N. Toosi, A. Ghaffari and R. Ebrahimpour, "ECG Arrhythmia Recognition via a Neuro-SVM-KNN Hybrid Classifier with Virtual QRS Image-Based Geometrical Features", *Expert Systems with Applications*, Vol. 39, No. 2, pp. 2047–2058, 2012.

22. Pan, J. and W. J. Tompkins, "A Real-Time QRS Detection Algorithm", *IEEE Transactions on Biomedical Engineering*, Vol. BME-32, No. 3, pp. 230 –236, 1985.

23. Hamilton, P. S. and W. J. Tompkins, "Quantitative Investigation of QRS Detection Rules Using the MIT/BIH Arrhythmia Database", *IEEE Transactions on Biomedical Engineering*, Vol. BME-33, No. 12, pp. 1157 –1165, 1986.

24. Park, K., K. Lee and H. Yoon, "Application of a Wavelet Adaptive Filter to Minimise Distortion of the ST-Segment", *Medical and Biological Engineering and Computing*, Vol. 36, pp. 581–586, 1998, 10.1007/BF02524427.

25. Kadambe, S., R. Murray and G. F. Boudreaux-Bartels, "Wavelet Transform-Based QRS Complex Detector", *IEEE Transactions on Biomedical Engineering*, Vol. 46, No. 7, pp. 838 –848, 1999.

26. Park, K., M. Khil, B. Lee, K. Jeong, K. Lee and H. Yoon, "Design of a Wavelet

Interpolation Filter for Enhancement of the ST-Segment", *Medical and Biological Engineering and Computing*, Vol. 39, pp. 355–361, 2001, 10.1007/BF02345291.

27. Ghaffari, A., M. Homaeinezhad, M. Khazraee and M. Daevaeiha, "Segmentation of Holter ECG Waves via Analysis of a Discrete Wavelet-Derived Multiple Skewness–Kurtosis Based Metric", *Annals of Biomedical Engineering*, Vol. 38, pp. 1497–1510, 2010, 10.1007/s10439-010-9919-3.

28. Lin, C. H., Y. C. Du and T. Chen, "Adaptive Wavelet Network for Multiple Cardiac Arrhythmias Recognition", *Expert Systems with Applications*, Vol. 34, No. 4, pp. 2601 – 2611, 2008.

29. Benitez, D., P. A. Gaydecki, A. Zaidi and A. P. Fitzpatrick, "The Use of the Hilbert Transform in ECG Signal Analysis", *Computers in Biology and Medicine*, Vol. 31, No. 5, pp. 399 – 406, 2001.

30. Welch, P., "The Use of Fast Fourier Transform for the Estimation of Power Spectra: a Method Based on Time Averaging over Short, Modified Periodograms", *IEEE Transactions on Audio and Electroacoustics*, Vol. 15, No. 2, pp. 70 – 73, 1967.

31. Xue, Q., Y. H. Hu and W. J. Tompkins, "Neural-Network-Based Adaptive Matched Filtering for QRS Detection", *IEEE Transactions on Biomedical Engineering*, Vol. 39, No. 4, pp. 317 –329, 1992.

32. Andreao, R., B. Dorizzi and J. Boudy, "ECG Signal Analysis through Hidden Markov Models", *IEEE Transactions on Biomedical Engineering*, Vol. 53, No. 8, pp. 1541 –1549, 2006.

33. de Chazal, P. and B. Celler, "Selection of Optimal Parameters for ECG Diagnostic Classification", *Computers in Cardiology 1997*, pp. 13 –16, 1997.

34. Amma, N. G. B., "Cardiovascular Disease Prediction System Using Genetic Algorithm and Neural Network", *2012 International Conference on Computing, Com-*

munication and Applications (ICCCA), pp. 1 –5, 2012.

35. Jiang, W., S. G. Kong and G. D. Peterson, "ECG Signal Classification Using Block-Based Neural Networks", *2005 IEEE International Joint Conference on Neural Networks (IJCNN '05)*, Vol. 1, pp. 326 – 331, 2005.

36. Ölmez, T., "Classification of ECG Waveforms by Using RCE Neural Network and Genetic Algorithms", *Electronics Letters*, Vol. 33, No. 18, pp. 1561 –1562, 1997.

37. Dokur, Z. and T. Ölmez, "ECG Beat Classification by a Novel Hybrid Neural Network", *Computer Methods and Programs in Biomedicine*, Vol. 66, No. 2–3, pp. 167 – 181, 2001.

38. Zhou, J. and L. Li, "Using Genetic Algorithm Trained Perceptrons with Adaptive Structure for the Detection of Premature Ventricular Contraction", *Computers in Cardiology 2004*, pp. 353 – 356, 2004.

39. Schaffer, J., D. Whitley and L. J. Eshelman, "Combinations of Genetic Algorithms and Neural Networks: a Survey of the State of the Art", *1992 International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, pp. 1 –37, 1992.

40. Sexton, R. S., R. E. Dorsey and N. A. Sikander, "Simultaneous Optimization of Neural Network Function and Architecture Algorithm", *Decision Support Systems*, Vol. 36, No. 3, pp. 283–296, 2004.

41. Gutierrez-Osuna, R., *Intelligent Sensor Systems, Lecture Topic: Validation*, Texas A&M University, Computer Science & Engineering, 2002, `http://courses.cs.tamu.edu/rgutier/ceg499_s02/l13.pdf`, accessed at June 2012.

42. Halford, G. S., W. H. Wilson and S. Phillips, "Processing Capacity Defined by Relational Complexity: Implications for Comparative, Developmental, and Cognitive Psychology", *Behavioral and Brain Sciences*, Vol. 21, No. 06, pp. 803–831, 1998.

43. *TEPA Inc*, 2007, `http://www.tepa.com.tr/`, accessed at Aug. 2012.

44. Tsai, F. S. and K. L. Chan, "Dimensionality Reduction Techniques for Data Exploration", *2007 6th International Conference on Information, Communications Signal Processing*, pp. 1 –5, 2007.

45. Alpaydın, E., *Introduction to Machine Learning*, MIT Press, Massachusetts Institute of Technology, 2010.

46. Furlanello, C., M. Serafini, S. Merler and G. Jurman, "Gene Selection and Classification by Entropy-Based Recursive Feature Elimination", *2003 Proceedings of the International Joint Conference on Neural Networks*, Vol. 4, pp. 3077 – 3082 Vol.4, 2003.

47. Hall, M. A., *Correlation-Based Feature Selection for Machine Learning*, Tech. rep., 1999.

48. Kira, K. and L. A. Rendell, "A Practical Approach to Feature Selection", *Proceedings of the ninth international workshop on Machine learning*, ML92, pp. 249–256, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1992.

49. Rissanen, J., "Modeling by Shortest Data Description", *Automatica*, Vol. 14, No. 5, pp. 465 – 471, 1978.

50. Kononenko, I., "On Biases in Estimating Multi-Valued Attributes", pp. 1034–1040, Morgan Kaufmann, 1995.

51. Weinberger, K. Q. and L. K. Saul, "Fast Solvers and Efficient Implementations for Distance Metric Learning", *Proceedings of the 25th international conference on Machine learning*, ICML '08, pp. 1160–1167, ACM, New York, NY, USA, 2008.

52. Vapnik, V. N., *The Nature of Statistical Learning Theory*, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

53. John, G. and P. Langley, "Estimating Continuous Distributions in Bayesian Classifiers", *In Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, pp. 338–345, Morgan Kaufmann, 1995.

54. Murphy, K. P., *Dynamic Bayesian Networks: Representation, Inference and Learning*, 2002.

55. Quinlan, J. R., *C4.5: Programs for Machine Learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

56. Hall, E. F., M., G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, "The WEKA Data Mining Software: an Update", *ACM SIGKDD Explorations Newsletter*, Vol. 11, No. 1, pp. 10–18, 2009.

57. Cohen, W. W., "Fast Effective Rule Induction", *In Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115–123, Morgan Kaufmann, 1995.

58. Frank, E. and I. H. Witten, "Generating Accurate Rule Sets Without Global Optimization", *Proceedings of the Fifteenth International Conference on Machine Learning*, ICML '98, pp. 144–151, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1998.

59. Downey, T. J., Jr, D. J. Meyer, R. K. Price and E. L. Spitznagel, "Using the Receiver Operating Characteristic to Asses the Performance of Neural Classifiers", *1999 International Joint Conference on Neural Networks (IJCNN '99)*, Vol. 5, pp. 3642 –3646, 1999.

60. Jadhav, S. M., S. L. Nalbalwar and A. A. Ghatol, "Modular Neural Network Based Arrhythmia Classification System Using ECG Signal Data", *International Journal of Information Technology & Knowledge Management*, Vol. 4, pp. 205–209, 2011.

61. Duda, R., P. Hart and D. Stork, "Pattern Classification, New York: John Wiley

& Sons, 2001, ISBN: 0-471-05669-3", *Journal of Classification*, Vol. 24, No. 2, pp. 305–307, 2007.

62. *Mathworks*, 2011, `http://www.mathworks.com/`, accessed at Aug. 2012.

63. Chang, C. C. and C. J. Lin, "LIBSVM: a Library for Support Vector Machines", *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, pp. 27:1–27:27, 2011, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

64. Hsu, C. W., C. C. Chang and C. J. Lin, *A Practical Guide to Support Vector Classification*, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan, 2003.