ECG ARRHYTHMIA CLASSIFICATION USING CLASS-MODULAR MLP

by
Haydar Vural
B.S., Computer Engineering, Boğaziçi University, 2001

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
2010

ECG ARRHYTHMIA CLASSIFICATION USING CLASS-MODULAR MLP

APPROVED BY:

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

Prof. Ethem Alpaydın ………………..

Prof. Mehmed Özkan ………………..

DATE OF APPROVAL: 28. 04. 2010

ACKNOWLEDGEMENTS

I would like to thank my thesis advisor, Prof. Fikret Gürgen, for his contributions during the development and preparation of the ideas in this thesis, and for helpful comments on this research. I also thank Prof. Ethem Alpaydın and Prof. Mehmed Özkan for their valuable comments and critics during thesis defense and development of study.

I am very grateful to my wife Gülfer, my mother Serpil, my father Hasan, my sisters Özlem and Neslihan and Levent Özgür, who is both my friend and colleague. Throughout my thesis, they provided encouragement and supported me to complete this work. This thesis would not be completed without the motivation they provided.

This thesis is the final output of a study started in September, 2001. During this long, stressful and tiring time, many anonymous people contributed this thesis. I am also grateful to those, who somehow helped me while I was a student, a research assistant and an alumnus of Boğaziçi University.

ABSTRACT

ECG ARRHYTHMIA CLASSIFICATION USING CLASS-MODULAR MLP

ECG (Electrocardiography) is a graphical signal of electrical activity recorded from electrodes on the body surface. It is one of the most important biosignal used by cardiologists for diagnostic purposes. In this study, our main objective is automatically recognition of arrhythmic signal abnormalities, which may be a clue for diagnosis. The detection of an abnormality in ECG signals by human is both complex and error-prone. This motivated researchers to study automatic detection of cardiac arrhythmia disorders, using intelligent data analysis techniques. Computer software using machine learning techniques could easily analyze complex ECG signals, transform signals, make some predictions about the presence of arrhythmia, and provide decision-support information to humans. In this study Multilayer Perceptron (MLP), which is a neural network-based machine learning technique and Class-Modularity concept were applied to two ECG datasets for arrhythmia classification. Class-modularity was also used by class-dependent feature selection to obtain robust modules also providing dimensionality reduction. RELIEF was selected as a well-known technique for class-specific feature list creation. One of the datasets is from UCI repository and it was used on similar studies before. A local dataset is created using real-life ECG recordings collected from Turkish patients. These records are digitized and examined by a medical doctor. The performances of learning methods are improved by feature selection (Decision Trees, SVM-RFE) and feature extraction (PCA) dimensionality reduction techniques. As a comparison, Decision Tree and SVM algorithms have been tested on the arrhythmia dataset. Weka and Matlab were used as machine learning tools during the study. According to test results, MLP performs better than decision trees and similar to SVM on both ECG datasets. The class-modular MLP has slightly less performance, while providing several advantages over MLP.

ÖZET

SINIF MODÜLER ÇGY KULLANILARAK EKG ARİTMİ SINIFLANDIRMASI

EKG (elektrokardiyogram) insan vücuduna iliştirilen elektik algılayıcılarla kaydedilen elektriksel aktivitenin sinyal grafiğidir. Kardiyologlar tarafından teşhiste kullanılan en önemli biyosinyallerden birisidir. Bu çalışmada temel amacımız, teşhiste yardımcı olabilecek aritmik sinyal anormalilerini otomatik olarak tespit etmektir. EKG'deki bu anormalilerin insanlar tarafından tespiti hem zor hem de hataya açıktır. Bu nedenler, araştırmacıları kalple ilgili aritmi düzensizliklerini otomatik olarak tespit etmeye yönelik araştırma yapmaya yönlendirmiştir. Özdevimli öğrenme teknikleri kullanan bilgisayar yazılımları, karmaşık EKG sinyallerini kolayca analiz edebilir, bunları dönüşütrebilir, aritmi varlığı hakkında tahminlerde bulunabilir ve insanlara kararlarında destek olabilecek bilgiler sağlayabilirler. Bu çalışmada, sinir ağlarına dayanan öğrenme tekniklerinden birisi olan, Çok Katmanlı Geriye Yayılma Algoritması (ÇGY) ve Sınıf-Modülü kavramı iki EKG veri kümesine uygulanmıştır. Sınıf-Modülü kavramı, sınıfa dayalı özellik seçimiyle kullanılarak aynı zamanda boyut azaltma da sağlayan dayanıklı modüller elde edilmesi hedeflenmiş ve bunun için RELIEF tekniği kullanılmıştır. Veri kümelerinden birisi UCI veri havuzundan alınmış daha önce benzer çalışmalarda kullanılmıştır. Bulunulan ülkeye ait bir veri kümesi ise Türk hastalardan toplanan gerçek EKG kayıtlarından yaratılmıştır. Bu kayıtlar dijital ortama aktarılmış ve bir tıp doktoru tarafından incelenmiştir. Özellik seçme (Karar Ağaçları, DVM-Döngüsel Özellik Azaltılması) ve özellik genişletme (Asıl Bileşen Analizi) boyut azaltma teknikleri kullanılarak öğrenme tekniklerinin performansı arttırılmaktadır. Karşılaştırma amaçlı olarak Karar Ağaçları ve Destek Vektör Makineleri aritmi veri kümelerinde test edilmiştir. Weka ve Matlab çalışmalar sırasında özdevimli öğrenme araçları olarak kullanılmışlardır. Yapılan test sonuçlarına gore, ÇGY her iki EKG veri kümesi üzerinde de Karar Ağaçlarından daha iyi, DVM'yle yaklaşık sonuçlar vermektedir. Sınıf-modüler ÇGY'nin biraz daha az başarılı olsa da ÇGY'ye gore sunduğu ek avantajlar vardır.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Computational sciences and engineering has been focusing on medical data for a while and this effort created a new field of scientific research namely, bioinformatics. Bioinformatics is concerned with prevention, diagnosis and treatment of diseases and aims to combine disciplines like medicine, computer science and electronics together. Intelligent methods, new algorithms, software and hardware are included in bioinformatics applications.

Automatic data analysis in medicine has been researched by scientists. Previous studies and applications in medicine have been concentrated on various topics like Artificial Neural Networks [1, 2] Fuzzy Systems [3], Statistical Approaches [4] and Support Vector Machines [5], etc. Automatic analysis methods presented advantages over manual analysis in medical applications. Intelligent software could easily interpret complex medical data, predict the presence of a disease based on past data, provide automated real-time analysis and diagnosis and allow both identification and classification of input data quickly. “Machine learning” methods are expected to create advanced and more successful medical diagnostic techniques in the future [1].

ECG signals can be used to determine the cardiac diseases. The analysis of these signals requires a detailed examination of graphic representations and common patterns with their respective classification [1]. Annotating ECG signals has been a difficult task for human beings, since there are multiple properties of an ECG signal. This empowered the studies on automatic detection of cardiac arrhythmic disorders. Artificial Neural Networks have been widely used for arrhythmia classification [3, 6, 7] like other methods. In this study, Multilayer Perceptron (MLP), a neural network based classification method, and class-modularity concept were applied to detect arrhythmic abnormalities.

This study involves data acquisition, dimensionality reduction with feature selection and extraction, classification and interpretation of results. Additionally, a local dataset was created for further studies. ECG signals of patients were transformed into numeric features that were used in classification of patients. UCI Repository [8] and MIT-BIH [9] database are two most popular arrhythmia databases used in literature.

UCI Arrhythmia Database is used in this study to train and test MLP. Due to the high dimensional characteristics of data Decision Trees and Recursive Feature Elimination with Support Vector Machines used as feature selection techniques and Principal Component Analysis (PCA) has been applied as a feature extraction method for dimensionality reduction. After dimensionality reduction, MLP and Class-Modular MLP have been applied to final datasets. The modules of Class-Modular MLP are trained and tested by class-dependent feature subset. In addition to that, Decision Tree and SVM methods also applied to same datasets to compare the results. Different classification statistics were finally presented for all methods.

1.1. Motivation

Arrhythmia is a very common health problem both in developed countries and emerging countries. Currently, 2.2 million Americans are living with atrial fibrillation [10], one type of arrhythmia, and the leading cause of death in emerging countries is the cardiovascular related disease [6]. ECG recording analysis one of the most commonly used diagnosis tool in cardiac arrhythmias. It is suggested that most of deaths could be avoided with efficient detection, monitoring and diagnosis of these disease using ECG recordings.

High availability of ECG machines in hospitals and easy operation on patients, made ECG analysis a major diagnosis method for cardiac diseases. Automatic ECG analysis needs intelligent methods to detect and analyze abnormal patterns in signals and it is critical in quick diagnosis of cardiovascular diseases and related health problems. Additionally, computer-aided ECG analysis is more reliable and error-free. Various techniques have been used to classify arrhythmias. Finding a robust and reliable classification method is a challenging task in diagnosis. In the literature there exist samples of different methods and algorithms [1, 2, 3, 6, 7] and most of the effort was spent on testing different learning methods for accurate diagnosis of arrhythmias.

The objective of this study is to classify certain cardiac arrhythmias using Multilayer Perceptron (MLP) with Class-Modularity. Artificial Neural Networks applied for ECG arrhythmia classification and their performance was shown on different datasets [3, 4, 6, 7]. MLP has been tested on MIT-BIH dataset [6] with limited parameters and no comparison is provided for other learning methods like SVM and Decision Trees. In this study, a comparison for learning methods on ECG dataset was provided and class modularity is applied to MLP.

Modularity can be defined as subdivision of a complex object into simpler objects [11]. Class modularity in machine learning has been mostly used in handwriting detection [12, 13, 14] because of the modular characteristic of dataset. It enhances performance and has advantages to traditional approach. The class modularity combined with MLP, allows us to have high accuracy prediction with modular architecture. Additionally, using subset of features for each module provides dimensionality reduction.

1.2. Outline

Chapter 1 is the introductory part of this thesis. The motivation of the research and the outline is given.

Chapter 2 gives detailed information about the arrhythmia disorders, ECG signals and their interpretations. The UCI dataset with its characteristics are explained.

Chapter 3 presents the overview of methodology and creation of optimized datasets. Dimensionality reduction with feature selection, feature extraction and class-dependent feature selection are described here.

Chapter 4 includes all the learning methods used for ECG arrhythmia detection. The proposed arrhythmia classification system, classification by Decision Trees and SVM, implementation of MLP and proposed architecture for class modularity is also explained in that chapter.

Chapter 5 shows the local data acquisition process, which is an important issue in medical data analysis. One of the outputs of this study is creating a small dataset for further research. The detailed information about dataset is presented with methodology of acquisition.

Chapter 6 includes the experimental results of this application. Performance of learning methods, affect of dimensionality reduction and feature extraction, analyses of model and comparison of classification techniques are discussed.

Chapter 7 is the final summary of study and the conclusion part. Possible future research topics are also discussed in this chapter.

2. BACKGROUND

2.1. ECG Signals

Electrocardiography (ECG) is an interpretation of the electrical activity of the heart captured over time and recorded by electrodes connected to skin. Electrical impulses in the heart originate in the sinoatrial node and travel to the heart muscle. The electrical waves can be measured at electrodes placed at certain points on the skin. Electrodes on different sides of the heart measure the activity of different parts of the heart muscle. The ECG displays the voltage difference between pairs of these electrodes, and the muscle activity that they measure [15].

The overall rhythm in ECG shows how the heart's working and weaknesses in different parts of the heart muscle. It is the best way to measure and diagnose arrhythmias of the heart, conductive muscles that carries electrical signals and electrolytes the signal is carried. Detection of abnormal ECG signals is a critical step in treatment to patient. Early detection of heart diseases can prolong life and enhance the quality of living through appropriate treatment.

A typical ECG graphic consists of repeating P wave, PR interval, QRS complex, ST segment and T wave. A normal ECG rhythm is given in Figure 2.1.

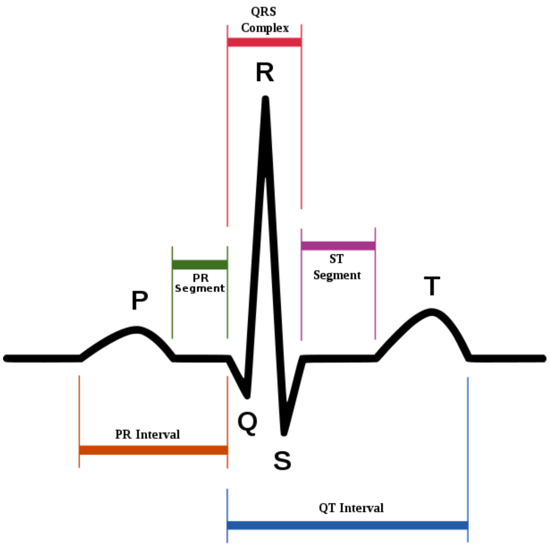


Figure 2.1. Normal ECG signal [16]

P wave: Atrial depolarization produces the P wave on the electrocardiogram. The duration of the P wave should not exceed 0.12 s.

PR interval: The PR interval is the time between the onset of atrial depolarization and the onset of ventricular depolarization and it is measured from the beginning of the P wave to the first deflection of the QRS complex. The normal duration of the PR interval is 0.12 s to 0.20 s.

QRS complex: The QRS complex represents the electrical forces generated by ventricular depolarization. The duration of the QRS complex should not exceed 0.10 s.

ST segment: The ST Segment lies between the QRS complex and the beginning of the T wave, and represents the period between the end of ventricular depolarization and the beginning of repolarization.

T wave: Ventricular repolarization produces the T wave. The normal T wave is asymmetrical, the first halve having a more gradual slope than the second half.

QT interval: The QT interval is measured from the beginning of the QRS complex to the end of the T wave and represents the total time taken for depolarization and repolarization of the ventricles. The QT interval increases slightly with age and tends to be longer in women than in men.

U wave: The U wave is a small deflection that follows the T wave. Many electrocardiograms have no discernible U waves. Prominent U waves may be found in athletes and are associated with hypokalaemia and hypercalcaemia.

The various parameters electrocardiographics (ECG) are basic in characterization of the forces generated during the cardiac activity. Actually, it is an essential tool for the diagnosis of cardiac abnormalities. The analysis consist of the measure of the amplitudes, various segment the durations and the morphologies of the P, QRS and T waves [17].

2.2. 12-lead Standard ECG

Timed interpretation of ECG is recorded on a scaled paper, which shows time on x axis and magnitude of voltage difference on y axis. ECG is mostly recorded with 12-lead standart configuration. In that configuration 10 electrodes are placed on body surface and 12 voltage differences are recorded. A normal 12-lead ECG is shown in Figure 2.2.

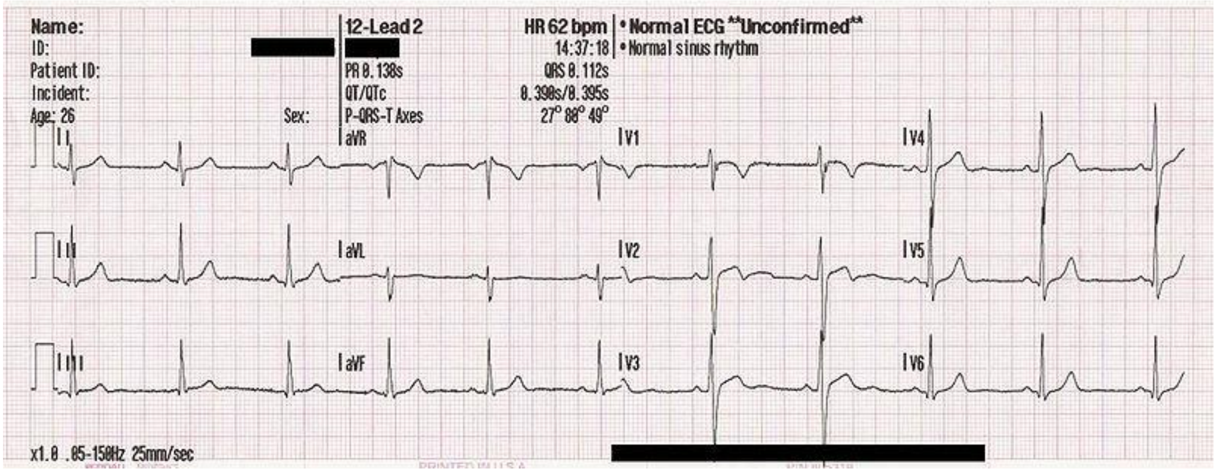


Figure 2.2. A normal 12-lead ECG [16]

2.3. Arrhythmia and Abnormal Rhythms

Arrhythmia is used to refer any abnormal cardiac rhythm, which is different than normal sinus rhythm. The nature of the problem may lead to a change or shift in the shape and size of P-QRS-T waves, the time intervals between its various peaks. These deviations provide useful information about the problem and how body is affected. But, it is not easy to observe and determine the arrhythmias due to nature of biosignals.

Biosignals are highly subjective and they are affected by factors like age and sex. The symptoms may appear at random in the time scale or may be discarded during diagnosis. Therefore, the signal parameters, extracted and analyzed using computers are highly useful in diagnostics.

Automated arrhythmia detection has been used since early 1960s. These systems mostly use QRS complex and PR interval to group arrhythmias into ventricular and supra ventricular categories. Then ventricular arrhythmias can be analyzed further. However supra ventricular arrhythmias need detection of P waves in addition to the QRS complex. Some arrhythmias are life-threatening medical emergencies that can result in cardiac arrest and sudden death. Others cause symptoms such as an abnormal awareness of heart beat, and may be only annoying.

Different kinds of arrhythmia disorders can be summarized as Ischemic Changes

(Coronery Artery Disease), Old Arterior Myocardial Infarction, Old Interior Myocardial Infarction, Sinus Tachcardy, Sinus Bradycardy, Ventricular Premature Contraction, Supraventricular Premature Contraction, Left Bundle Branch Block, Right Bundle Branch Block, Left ventricule hypertrophy and Atrial Fibrillation or Flutter.

2.4. UCI Arrhythmia Database

The Arrhythmia dataset used in this study is obtained from UCI Repository [8]. The UCI Machine Learning Repository is a collection of databases, domain theories, and data generators that are used by the machine learning community for the analysis of machine learning algorithms. The dataset includes 452 patient records which are described by 279 features. The records are classified into 16 different classes. Class 01 refers to normal ECG, classes 02-15 refers to the different kinds of arrhythmia and class 16 refers to the unclassified records. The total number of samples for each class is different and the dataset doesn't include record of class 11, 12 and 13. Table 2.1 shows the class distribution in UCI dataset.

Table 2.1. Class Distribution of Arrhythmia Database

Code	Class	# of Instances
1	Normal	245
2	Ischemic Changes (Coronary Artery Disease)	44
3	Old Anterior Myocardial Infarction	15
4	Old Inferior Myocardial Infarction	15
5	Sinus Tachycardy	13
6	Sinus Bradycardy	25
7	Ventricular Premature Contraction (PVC)	3
8	Supraventricular Premature Contraction	2
9	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

Each record contains basic personal information like age, sex, height, weight; attributes that are easy to measure by cardiologists like QRS duration, PR interval, QT interval and more complex information like QRSA and QRSTA.

Each record has annotation attached by cardiologists after examining all features and it is accepted as the true classification of records. All the attributes and their definition are given in Appendix A.

There are some missing values in the dataset and most of them are related with same attribute, f14. This attribute and other missing values are handled during creation of optimized dataset and the details are given in Chapter 3 and Chapter 6.

Despite the fact that a patient may have a few arrhythmias, one of the major assumptions that the dataset carries is no patient has more than one cardiac arrhythmia [18]. The distribution of classes is unfair and 54% of records are normal. The top 4 classes with highest frequency constitute 80% of records. The distribution of classes is shown in Figure 2.3.

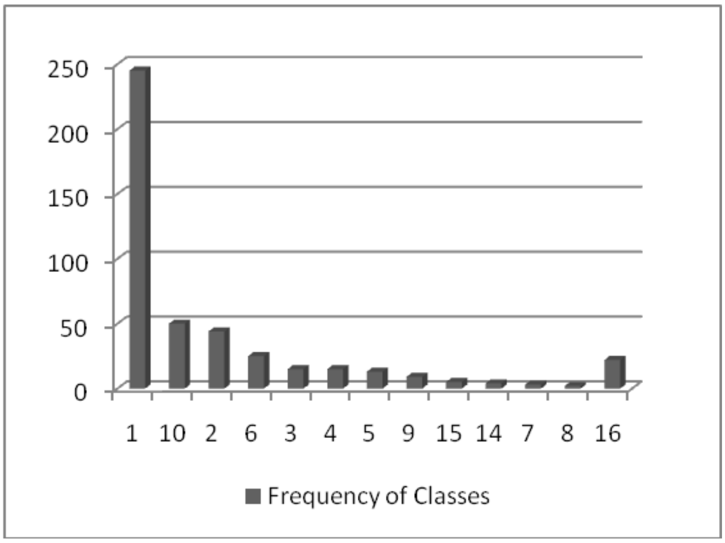


Figure 2.3. Arrhythmia class distribution

The dataset was initially used by its owners, in the “A Supervised Machine Learning Algorithm for Arrhythmia Analysis” [18]. This study uses VFI5 algorithm to diagnose cardiac arrhythmia using majority voting technique. The VFI5 algorithm achieved an accuracy of 62% on the data set. Another study “ANN Based Diagnostic System for Arrhythmia with ECG Signals” [3] used the same dataset and presented an artificial neural network classifier solution based on a Bayesian framework. “Intelligent Arrhythmia

Classification Based on Support Vector Machines” [19] focused prediction of arrhythmia using SVM. In that study k-Nearest Neighbor and Decision Tree methods were evaluated for comparison with SVM.

3. METHODOLOGY AND DIMENSIONALITY REDUCTION

3.1. Introduction

The ECG arrhythmia classification on UCI dataset is a supervised learning problem with data acquisition, data optimization, training and testing phases. During each phase several methods are applied to obtain higher accuracy and finally comparing performances of different learning techniques. Figure 3.1 shows the overall learning process and phases.

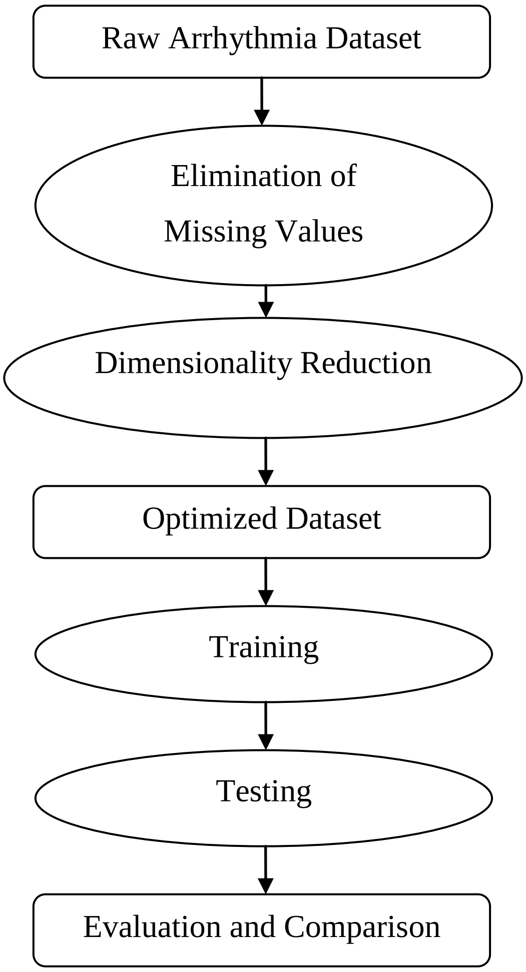


Figure 3.1. Arrhythmia learning process

3.2. Raw Dataset and Elimination of Missing Values

The UCI dataset consists of 452 records with 279 attributes and a class number associated with each record. All of the values are numeric and there is no information about the distribution of attributes.

Missing data occurs frequently in real-life medical data. They may be missing because of malfunctioning equipments or lack of observation by the operating staff. There is no best general handling algorithm for missing values [20]. In sequential methods original datasets with missing attribute values are converted into complete data sets and then classification process is applied. When the raw dataset is examined attribute 14 is missing in 83% of records. Modified listwise deletion [20] is applied to dataset and this attribute is removed from all records. After this process, 278 attributes are used during rest of the study.

There are total 32 missing values in the rest of the dataset, which sums up to 0.025 % of all values. These missing values are observed in multiple records and they do not belong to a specific attribute. Most Common Value of an Attribute technique [20] or similar recovery techniques are usually used in machine learning problems. But, this approach may create serious misclassification problems in medical data. To prevent a probable mistake, records with the missing values were removed from original dataset. Final dataset contains 446 records.

3.3. Dimensionality Reduction

As the available data becomes more high dimensional in machine learning, dimension reduction techniques are frequently used as a preprocessing step. Dimensionality reduction is the process of choosing a reduced set of original attributes or using new attributes derived from original attributes [21].

Main benefits of dimensionality reduction are:

- Reducing time complexity,
- Reduces space complexity
- Reducing cost of observing unnecessary features

- Creating simpler models on small datasets, which also allow creating more robust models
- Fewer features make it easier to understand the underlying process
- Visualization of 2 or 3 dimensions is easy to understand [21].

Successful methods of removing irrelevant features increase efficiency in medical applications and increase prediction rate, too [22]. There are two main techniques of dimensionality reduction. In feature selection the original dataset contains d dimensions and k dimensions are chosen from d (k<d). The remaining d-k is ignored. In feature extraction, a new set of k dimensions are derived from d dimensions. The learning problem is solved by using these new k dimensions.

The UCI arrhythmia dataset is a high dimensional dataset with 278 attributes available for learning. In this study, both feature selection and feature extraction methods are applied to this high dimensional dataset. A strong attribute subset or new derived attributes are aimed to achieve. For future extraction, Principal Component Analysis is chosen. Decision Trees and Recursive Feature Elimination with Support Vector Machines are used as feature selection techniques. A new optimized dataset is created following each dimensionality reduction method.

3.3.1. Principal Component Analysis

Principal Component Analysis (PCA) is widely used unsupervised feature extraction method [6, 21, 23, 24]. Instead of using the output information, PCA tries to maximize variance of attributes and use covariance matrix of input variables for eigen analysis. Eigenvector and their corresponding eigenvalues are calculated in eigen analysis.

In PCA, to determine the optimal number of dimensions, proportion of variance is used which is preferred to be higher than a predefined threshold value. If there is an input dataset with d dimensions, then proportion of variables is calculated according to the formula below where λ_i is the eigenvalue of eigenvector w_i and λ_i are in the decreasing order.

$$\frac{\lambda_1 + \lambda_2 + \ldots + \lambda_k}{\lambda_1 + \lambda_2 + \ldots + \lambda_k + \ldots + \lambda_d} \qquad (3.1)$$

The principal components are the eigenvectors with the highest *k* eigenvalues that meet proportion of variance shown as Equation 3.1. In order to obtain k dimensional reduced set, the linear projection is applied to principal components on original data [21].

Since there are multiple eigenvectors, deciding the number of eigenvectors is an important issue in dimensionality reduction. Scree graphs are used to decide number of eigenvectors to keep. The variance as a function of eigenvectors is displayed on scree graphs and it may show a certain point where adding one more eigenvector is not affecting variance.

3.3.2. Decision Trees

Decision tree is a well known hierarchical data structure for supervised learning and It is used both for classification and regression. Decision trees implement the divide-and-conquer strategy. The hierarchical placement of regions in a decision tree allows a quick localization of a region for a given input and the search time is logarithmic.

A decision tree has two main components: decision nodes and terminal leaves. Each decision node applies its test function to the given input and produces a discrete value that determines which branch is taken. A decision node creates a discriminant in the d-dimensional input space and dividing it into smaller regions as shown in Figure 3.2. Each leaf has an output label for all income which is a class label for classification problem and a numeric value for regression problem.

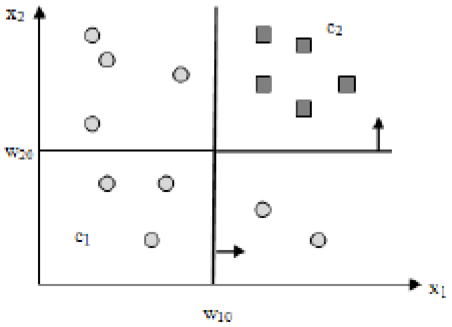


Figure 3.2. Decision Node Discriminant

Decision tree can be examined in two sub-groups: Univariate Trees where each internal node uses only one variable as is shown in Figure 3.3 and Multivariate Trees where all features can be used in each decision node.

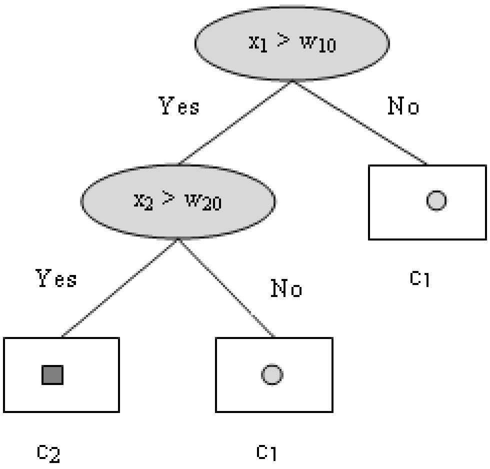


Figure 3.3. Univariate Decision Tree

In a univariate classification tree, learning starts at the root node with all features and the aim is obtaining the best split. This process continues recursively with the corresponding subset until a leaf node is obtained. The measure of the good split is impurity which is determined as if all instances of the branch are labeled as the same class.

$$\hat{P}(c_i \mid x, m) = p_m^i = \frac{N_m^i}{N_m} \tag{3.2}$$

For node m, N_m is the number of training instances reaching node m and N_m^i of them belong to class c_i. Node m is pure if p_m^i is zero or one.

The measure of impurity is entropy [21]. The best split is obtained when entropy is minimized. Entropy formula for node m is given in Equation 3.2.

$$\mathrm{I}_m = -\sum_{i=1}^{k} p_m^i \log_2 p_m^i \tag{3.3}$$

Decision tree is also known as a feature selection algorithm. The final univariate tree consists of the most relevant features and discards irrelevant ones. In this study, C4.5 tree is used as a feature selection method [25]. C4.5 tree is a univariate classification tree and recursively searches the input data until maximizes the classification performance and extracts the features that create the best splits.

3.3.3. Recursive Feature Elimination with Support Vector Machines

Recursive Feature Elimination (RFE) is a wrapper method that utilizes the generalization capability embedded in support vector machines (SVM). RFE keeps the independent features containing the original information stored in dataset and eliminates weak or redundant features [26]. However, the subset produced by SVM-RFE is not necessarily the ones that are individually most relevant. Only taken together the features of a produced subset are optimal informative [27].

The working methodology of SVM-RFE is based on backward selection where algorithm starts with whole features and iteratively eliminates the worst one until the predefined size of the final subset is reached. The remaining features must be ranked again [26] during each iteration.

SVM-RFE working principles could be examined in three steps in each iteration:

- Training the classifier (SVM)
- Computing the ranking criterion for all features
- Removing the feature with smallest ranking criterion

There are different ranking criterions proposed for SVM-RFE such as entropy [28] or square of the weight of separating hyperplane (w^2) [29]. In this work, square of weight as ranking criteria is used. In each iteration, the feature that causes minimum variation in the SVM cost function is removed from feature space. It is assumed that, in each step, trained SVM produces weight vector w^* according to the formula below where α_i are Lagrange multipliers which is greater than zero for support vectors:

$$w^* = \sum_{i \in SV} y_i \alpha_i^* x_i \tag{3.3}$$

For the trained SVM with the weight vector w^*, the cost function is $J(w)$:

$$J(w) = \frac{1}{2} \| w \|^2 \tag{3.4}$$

In order to find the variation in cost function of SVM *(δJ(i))*:

$$\delta J(i) = \frac{1}{2}\frac{\partial^2 J(w)}{\partial w_i^2}(\delta w_i)^2 = \frac{1}{2}(w_i)^2 \qquad (3.5)$$

Feature, which causes minimum variation is ranked and removed from feature space. SVM-RFE algorithm is given in Figure 3.4. In SVM-RFE, computational cost is higher while only one feature is removed in each step. When several features are removed at a time, feature subset ranking must replace with feature ranking.

```
Function RFE-SVM(TD, AF, RS)
Initialize
        TD : Training data
        AF : All Fetures in the dataset
        RS : Reduced feature subset
Begin
While( number of AF > RS)
                Train SVM on TD with the feature space AF
                Rank the features of F in the descending order
                RFS := AF – { feture with the smallest rank in AF}
                AF = RFS
        End
        Return AF
end
```

Figure 3.4. SVM-RFE Algorithm

3.4. Class-Dependent Feature Subset Selection

The dimensionality reduction techniques described in previous sections create an optimized dataset that contains useful formation for all classes that a learning problem has to be efficient for. But, one or more of the features may be correlated with specific classes. Redundant features degrade the performance of learning methods both in speed and predictive accuracy. Discarding the redundant features with feature subset selection may overcome these problems [30].

Modularity provides an alternative technique that may be incorporated with the initial feature selection process. In a modular learning algorithm, subset selection may be applied for each module separately. In our study, each module is optimized for a specific class and class-dependent feature subset selection was used to both increase accuracy and speed of learning algorithm. RELIEF is a simple and effective technique for class-dependent feature selection and it was previously used for machine learning applications [31, 32]. The finalized input datasets for class-modular learning algorithm was prepared with that additional step to achieve a compact input dataset.

3.4.1. RELIEF

RELIEF tries to estimate the quality of features according to how well their values distinguish between the instances that are near to each other. For this purpose, given a randomly selected instance *X* from a dataset *S* with k attributes, RELIEF searches the dataset for its two nearest neighbors. One of the neighbors should be of the same class and it’s called nearest hit *H* . The other neighbor should be of different class and it’s called nearest miss *M*. It updates the quality estimation $W[A_i]$ for all the features *Ai* depending on the difference *diff()* on their values for *X*, *M*, and *H*. The number of repeat process is defined by the user by parameter, *m*.

```
Given m-number of samped instances, and k- number of features
Set all weights $W[Ai]:=0$;
for $j := 1$ to $m$ do begin
    randomly select an instance $X$;
    find the nearest hit $H$ and nearest miss $M$;
    for $i := 1$ to $k$ do begin
        $W[Ai] := W[Ai] – diff( Ai, X, H) /m+ diff (Ai, X, M)/m$
    end
end
```

Figure 3.5. Original Relief Algorithm [30]

4. ECG ARRHTYMIA CLASSIFICATION

4.1. Introduction

ECG arrhythmia classification has two main stages after dimensionality reduction and creation of an optimized dataset. In Chapter 3, alternative methods to create an optimized dataset are presented. An optimized dataset has fewer attributes and carries remarkable information stored in original dataset. This dataset is used to both train the classification algorithm and test the accuracy of learning.

ECG Arrhythmia Classification is previously performed on UCI dataset using k-Nearest Neighbor and Decision Tree and SVM [19]. MLP as a popular artificial neural network is also implemented on MIT/BIH Arrhythmia database [6]. MLP has been also used to compare performance of different learning methods [24]. But the literature does not include application of MLP on UCI dataset, combined with feature extraction and feature selection methods. In this study, prediction of ECG arrhythmia with MLP is provided. In addition to that, the class-modularity is also applied to MLP to use advantages provided by modularity. One of these advantages is providing additional dimensionality reduction by fetaure subset selection for each module.

4.2. Multilayer Perceptron

Artificial neural networks are inspired from operation of brain and human nerve system. The biological neuron is the basic structure of nerve system and it has a very simple operation. It receives inputs along the "dendrites" and sums them up to compare with a threshold value. If the sum is greater than threshold value, the neuron shall produce an output. And the output is connected to other neurons. In that model, the neuron performs a weighted sum on its inputs and compares this to its internal threshold level to turn on the neuron. This system is known as a feedforward model. The mathematical formulation this model for synaptic weights w_j and inputs x_j and threshold value Φ can be modeled as:

$$y = \sum_{j=1}^{d} w_j x_j + w_0 \qquad (4.1)$$

y is the total input and if we call our total output k,

$k = f_h[t-\Phi]$ where,

$f_h(j)$ is known as activation function, which is unit step function here:

$f_h(j)=1, j > 0$

$f_h(j)=0, j < 0$

A perceptron is the term used to describe the connections of simple neurons into networks [33] and it is the basic processing element. The output of a perceptron can be written as a dot product for vector operations.

$$y = w^T x \qquad (4.2.)$$

During training stage of a perceptron, *w* is tried to be estimated using *x* and *y*. After calculation of *w*, the output can be calculated for a given input set and testing. The output of perceptron and the activation function may be modified to sigmoid, Gaussian, etc depending on desired learning method. Equation 4.3 shows the sigmoid function as an activation function.

$$o = w^T x$$

$$y = sigmoid(o) = \frac{1}{1 + e^{-w^T x}} \qquad (4.3)$$

The training of a perceptron follows a supervised learning where perceptron learns from its mistakes. The pseudocode of percepton learning algorithm is shown in Figure 4.1.

```
i.   Set initial weight and thresholds of the perceptron to
     random values.
ii.  Present an input.
iii. Calculate the output of the perceptron.
iv.  If the perceptron is active for desired input, and
     inactive fort he rest of output terminate the
     algorithm
v.   Else update the weights to reduce the error. So for
     the network to learn,  increase the weights on the
     active inputs when the output is wanted to be active,
     and to decrease them when output is wanted to be
     inactive.
vi.  Now present the next input and repeat steps iii. - v.
```

Figure 4.1. Perceptron Training Algorithm

A perceptron with a single layer of weights is unable to solve problems like XOR, where the discriminant to be estimated is nonlinear [21]. A simple problem which requires non-linear discriminant is shown in Figure 4.2. The region which separate starts from circles can't be determined using a single line and by single a perceptron.

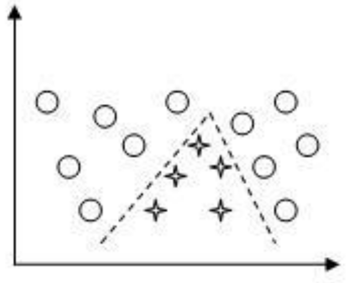


Figure 4.2. Non-Linear Discrimination Problem

The limitations of single-layer perceptrons are eliminated by using multilayer perceptrons. A multilayer perceptron (MLP) has intermediate or hidden layers between input and output layers. These hidden layers allow an MLP to implement nonlinear discriminants and MLPs are used to solve non-linear problems. Feedforward MLPs are the most widely used Artificial Neural Network (ANN) models. In MLP, using one hidden layer is generally preferred to reduce the complexity. Additionally, large number of hidden units may cause overfitting. The units in the hidden layer may be constant or decided during learning phase.

General structure of an MLP is shown in Figure 4.3. MLP is composed of three layers: an input layer, hidden layer and an output layer. The MLP shown in Figure 4.3 has only one layer of hidden nodes and each hidden node applies nonlinear sigmoid function to activate the output.

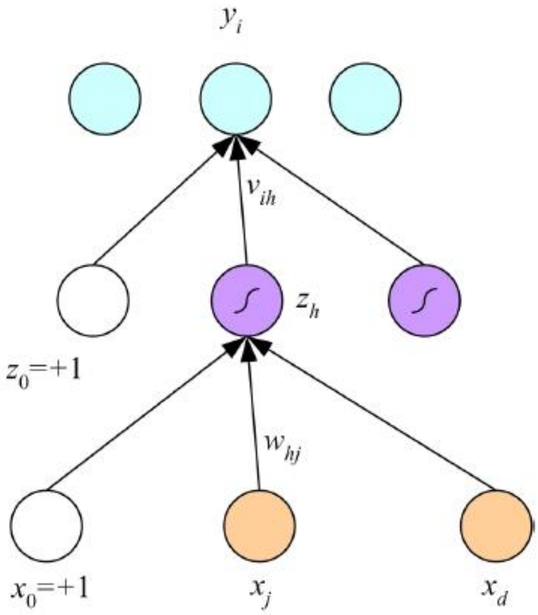


Figure 4.3. Structure of three layer MLP [21]

The inputs of the MLP are x_j, $j=0,\dots,\ d$. x_0 is the extra bias unit to input layer. z_h, $h=1,\dots,H$ are the hidden units. The dimensionality of the hidden space is H and it may not be equal to d. The hidden layer has an extra bias unit z_0 like the input layer. The output units y_i, $i\ =\ 1,\dots,\ K$ are activated by the weighted sum of hidden nodes. w_{hj} are weights in input layer and v_{ih} are weights in the hidden layer.

The nonlinear sigmoid function applied by each hidden unit is

$$z_h = sigmoid(w_h{}^T x) = \frac{1}{1+\exp[-(\sum_{j=1}^{d} w_{hj} + w_{h0})]} \tag{4.4}$$

The output vector y is produced by the propagation of activation from input layer to hidden layers and finally to output layer. For the three layers MLP of Figure 4.3 the output can be written as:

$$y_i = v_i{}^T z = \sum_{h=1}^{H} v_{ih} z_h + v_{i0} \tag{4.5}$$

MLP learning process starts at the input layer where no calculation is applied. At initialization step, weights are initialized to random values. Then, in each epoch, weighted sum of input variables are sent as input to hidden units where nonlinear activation function is applied. Hidden units produce h dimensional data as inputs for output unit which calculates weighted sum of inputs to produce output value. In back-propagation algorithm, output value of each layer is used for previous layer weight updates. This process continues until one of the stopping criterions is reached. Learning rate and momentum are two parameters used in update of weights in each epoch. The pseudocode of backpropagation algorithm is given in Figure 4.4.

```
Initialize all $v_{ih}$ and $w_{hj}$ in the range of (-0.01,0.01)
Repeat
    For all $(x^t, r^t) \in X$
        For h= 1,2,…,H
            $z_h \leftarrow sigmoid(w_h^T x_t)$
        For i = 1,2,…,d
            $y_i = v_i^T z$
        For i = 1,2,…,d
                $\Delta v_i = \eta(r_i^t - y_i^t)z$
        For h = 1,2,…,H
                $\Delta w_h = \eta(\sum_i (r_i^t - y_i^t)v_{ih})z_h(1-z_h)x^t$
        For i = 1,2,…,d
                $v_i \leftarrow v_i + \Delta v_i$
            For h = 1,2,…,H
                $w_h \leftarrow w_h + \Delta w_h$
    Until convergence
```

Figure 4.4. MLP Backpropagation Algorithm [21]

4.3. Class-Modular MLP

Modularity can be defined as subdivision of a complex object into simpler objects [11]. Modularity is observed in nature and human nervous system. Different part of the brain has different functions and interconnected with different parts. To give an example, the brain utilizes two separate parts to recognize a picture in a movie and the sound associated with it. Modular schemes in machine learning are inspired from that idea.

An n-class problem can be decomposed into n 2-class problems, which may be solved by different techniques or parameters. Figure 4.5 shows a sample class decomposition example for normal ECG signal recognition problem.

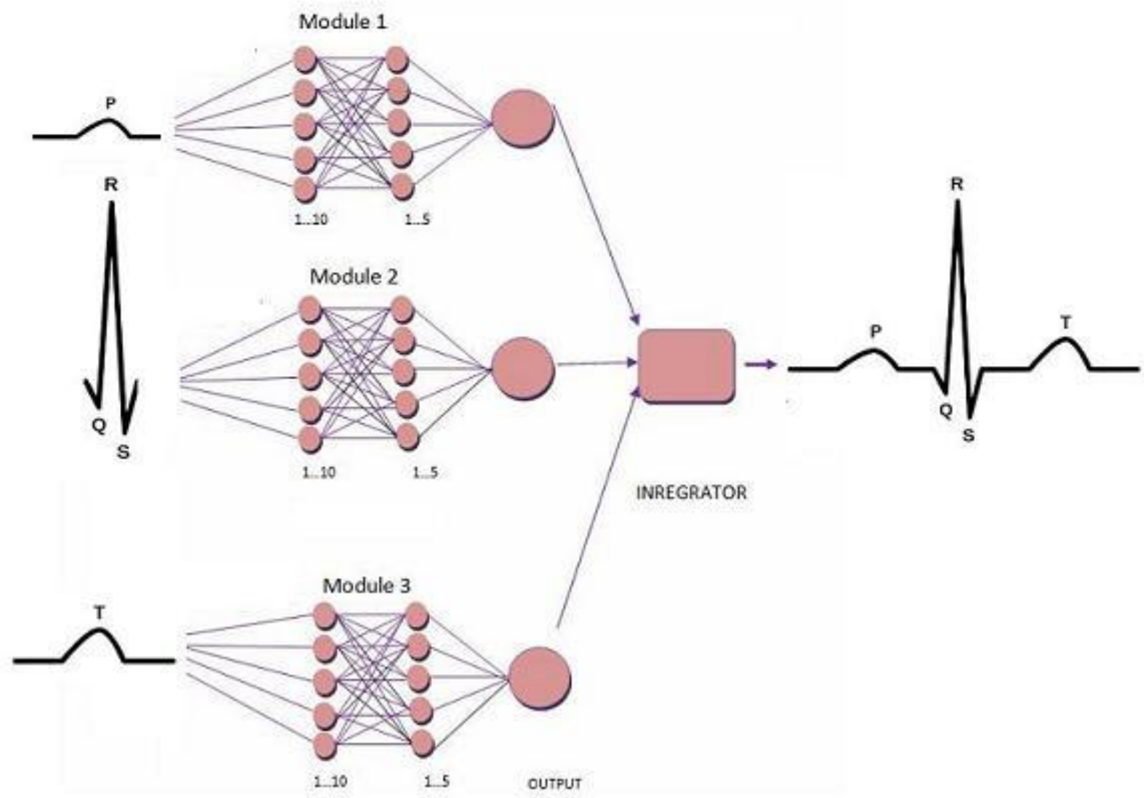


Figure 4.5. A Modular MLP Network for ECG Signal Recognition

Class modularity with MLP previously used for both learning problems like handwriting recognition [12, 13, 14, 34], face recognition [35] and electro-magnetic analysis [36]. Literature also contains studies where a class modular MLP schemes is applied to multiple datasets [37].

ECG Arrhythmia detection is an n-class problem where each class represents a specific type of arrhythmia. In this study, each module is an MLP, optimized for a binary classification problem. C_m , $m=0,\dots,n$ represents an instance of class m and each module is responsible for predicting whether the input dataset belongs to one type of arrhythmia or not. There are three stages of creating a class modular MLP using this approach. These are building modules, creating optimized datasets and combining the decisions of modules.

Building modules is very similar to building an MLP. Figure 4.6 shows a module for arrhythmia type *C_1*. It has the same structure shown in Figure 4.3. There are 2 outputs of each module, one hidden layer and *d* dimensional input. The *d* dimensional input is a subset of all features *k*. The subsets are different for each module, but the total number of features in subset is constant.

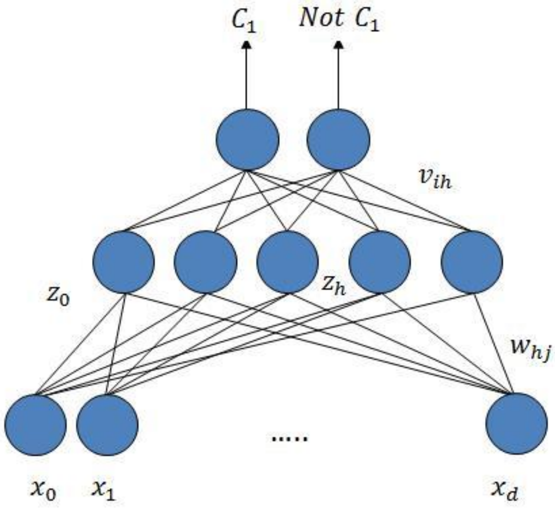
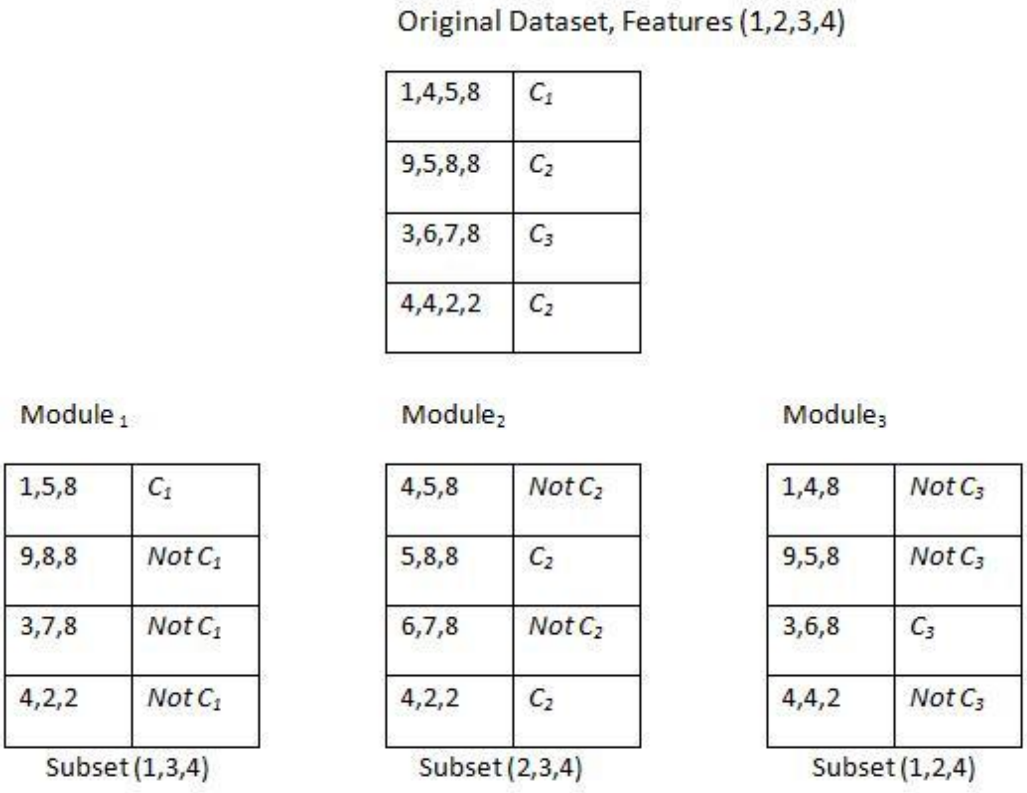


Figure 4.6. Class Modular MLP for *C_1*

An MLP network is trained with same dataset for all classes. But, class modular MLP needs modified dataset for each module. The modified dataset for module *Cm* should include all instances defined as either type *C_m* or *Not C_m* and the most important feature subset *d*. Figure 4.7 shows the transformation of a 3-class MLP input dataset for 3 class-modular MLP datasets. The first part of the dataset is the values of attributes. The second part is class the instance belongs to.The original dataset contains 4 features while the optimized dataset contains 3 class-dependent features for each module.



Original Dataset, Features (1,2,3,4)

1,4,5,8	C_1
9,5,8,8	C_2
3,6,7,8	C_3
4,4,2,2	C_2

Module $_1$

1,5,8	C_1
9,8,8	*Not* C_1
3,7,8	*Not* C_1
4,2,2	*Not* C_1

Subset (1,3,4)

Module$_2$

4,5,8	*Not* C_2
5,8,8	C_2
6,7,8	*Not* C_2
4,2,2	C_2

Subset (2,3,4)

Module$_3$

1,4,8	*Not* C_3
9,5,8	*Not* C_3
3,6,8	C_3
4,4,2	*Not* C_3

Subset (1,2,4)

Figure 4.7. Dataset Transformation for Class Modular MLP

After training each module and creating optimized datasets, the output of modules should be combined and inconsistencies between predictions of several modules should be resolved. A module combining algorithm (MCA) is developed for final decision. The MCA gives its final decision using the Probabilities Vector (PV) that represents the output of all modules. The MCA is shown in Figure 4.8.

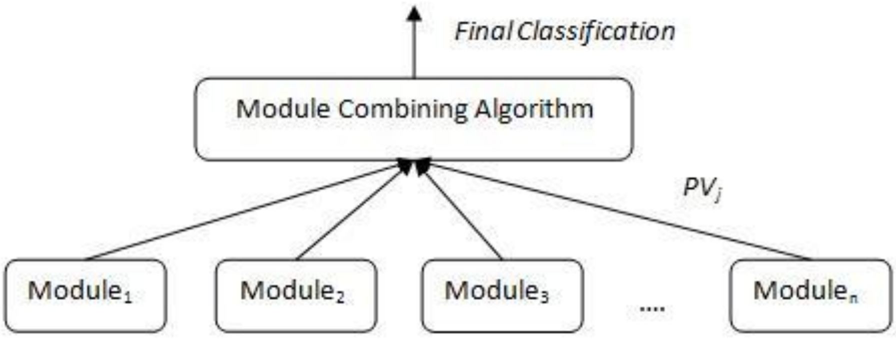


Figure 4.8. Module Combining Algorithm for Class Modular MLP

PV is a row vector of dimension *n* and it is built using the following information:

PV_j = Probability Distribution of Module *j*, where the output is Class C_j

It may be also expressed that

$1\text{-}PV_j$ = Probability Distribution of Module *j*, where the output is Not Class C_j

MCA aims to find the column with maximum probability. If there are multiple candidate columns, it chooses the one with maximum value. Using the maximum value both provides loss of useful information and creates a simpler final classification decision. The pseudocode of MCA is given in Figure 4.9.

```
$m = 0, ..., n$
For all $C_m$ calculate output of CM-MLP
$PV_m$ = Probability of module m for Class $Cm$
Find max $PV_m$
Classify as the one with maximum $PV_m$
```

Figure 4.9. Pseudocode of MCA

Class modular MLP has both advantages and disadvantages over MLP. Availability of defining different MLP parameters like initial weights, learning rate, momentum, number of hidden nodes and epoch amount allow customization of each MLP. The customization is not limited to MLP parameters, also different input subsets are chosen for each class. This provides additional dimensionality reduction and removes the redundant information. All these parameter and input selection may reduce training time and create more dynamic MLP networks.

Additionally, multiple learning methods may be applied to same dataset for different classes. It is possible to use SVM, Decision Trees MLP or any other learning method in combination with the class modular architecture. In some training datasets, same accuracy for C_j may need more complex MLP architecture while higher accuracy may be achieved by simpler MLP for C_i. Since the implementation of each module is separated, C_i and C_j may be handled by two different MLP.

Class modular architecture needs differentiation of input datasets, which is an extra work in dataset optimization. The extra level of complexity with module combining and selecting module dependent feature subsets should also be considered as a drawback of class modular MLP scheme. Any arbitrary function with continuous input and outputs can be approximated by with an MLP [21] and overtraining may be an issue for more complex networks.

4.4. Decision Trees

Decision Trees (DT) are one of the most popular approaches for both classification and regression type predictions. DT are in the form of a tree structure where each node is either a leaf node that indicates the value of the target class of examples or a decision node that specifies test on a single attribute value with one branch and sub-tree for each possible outcome of the test [38].

DT is typically constructed recursively in a top-down manner. If a set of labeled instances is sufficiently pure, then the tree is a leaf with the assigned label being that of the most frequently occurring class in that set. Otherwise, a test is constructed and placed into an internal node that constitutes the tree so far. A branch is created for each block of the partition, and a tree is constructed recursively for each block.

The main problem of the decision tree growing algorithms is selecting which attribute to test at each node in the tree. The concept of entropy is used for the selection of the attribute with the most inhomogeneous class distribution. Entropy characterizes the impurity of an arbitrary collection of examples. Information gain uses entropy to measure how well a given attribute separates the training examples according to their target classification [29].

Given a set of *S*, containing two classes of examples, the entropy of *S* is defined as

$$Entropy(S) = -p_1 \log_2 p_1 - p_2 \log_2 p_2 \tag{4.7}$$

where p_1 is the proportion of class 1 in S and p_2 is the proportion of class 2 (*0log0* is assumed to be 0). The entropy is 0 if all members of *S* belong to the same class and the entropy is 1 if the classes include equal number of examples. In other cases the entropy is between 0 and 1. When generalized to multiclass case:

$$Entropy(S) = \sum_{i=1}^{c} - p_i \log_2 p_i \tag{4.8}$$

where p_i is the proportion of *S* belonging to class *i*.

The information gain *Gain(S, A)* of an attribute A is defined as:

$$Gain(S,A) = Entropy(S) - \sum_{v \xi Value(A)} \frac{|S_v|}{|S|} Entropy(S_v) \tag{4.9}$$

where *Values(A*) is a set of all possible values for attribute *A*, and *Sv* is the subset of *S* for which attribute *A* has value *v*. If the attribute values are continuous, we should to define new discrete-valued attributes that partition the continuous attribute value into a set of discrete intervals.

DTs are able to generate understandable rules that it is possible to define each path from the root to a leaf node as a set of IF-THEN rules [21].

4.5. Support Vector Machines

Support Vector Machines (SVM) map data points to a high dimensional feature space where a separating hyperplane can be found. This mapping can be carried on by applying the kernel trick which implicitly transforms the input space into high dimensional feature space. The separating hyperplane is computed by maximizing the distance of the closest patterns.

SVM is an inductive machine learning technique based on the structural risk minimization which aims at minimizing the true error. SVMs generate black box models which lack the explanation capability on how to reach a decision [39]. Rather than minimizing the training error (empirical risk), SVMs minimize the structural risk which expresses an upper bound on the generalization error.

In most classification tasks, SVM generalization performance either matches or is significantly better than competing methods [40]. Also, as mentioned in the experimental results, SVM classifier can deal with high dimensional data.

In this study, non-linear SVMs and kernel functions are used for comparison. Using them, if the two classes are not linearly separable, instead of fitting a nonlinear function, the solution may be mapping the data to a higher dimensional space.

The key idea with the non-linear SVMs is that the original input space can always be mapped to some higher dimensional feature space where the training set is separable. In such a case we are interested in a method whose complexity does not depend on the input dimensionality but depends on the number of training instances [21].

Linear operation in feature space is equivalent to the non-linear operation in original input space. We use soft margin hyperplane because the problem may not be linearly separable in the new feature space. It is critical here to choose the penalty factor. If it is too large, high penalty will be given for non-separable points and there is a risk of overfitting due to storing many support vectors. If it is too small underfit may occur [21].

As the key idea of non-linear SVMs, kernel functions are used for mapping data to a higher dimensional space. The most popular kernel functions are:

Linear: $K(x_i, x_j) = x_i^T x_j$

Polynomial of degree p : $K(x_i, x_j) = (1 + x_i^T x_j)^p$

Radial Basis Function: $K(x_i, x_j) = \exp\left[-\frac{\left\| x_i - x_j \right\|^2}{\sigma^2} \right]$

Sigmoidal Function: $K(x_i, x_j) = \tanh(2x_i^T x_j + 1)$

5. LOCAL DATASET CREATION

UCI Dataset, which has been used on similar previous studies, is a high-dimensional dataset with large number of instances. But, medical data usually ocurs in small datasets with fewer attributes in real-life. Building a realistic learning process includes collection of realistic or if possible updated and unbiased real-life data. Creating a small dataset for further research and running learning methods on this dataset became one of the aims of this study.

ECG machines provide both graphical representation of signal and print some attributes available from the data. If an attribute varies in time, the machine generates a weighted average. Most of the attibutes like heart beat and QRS interval are that type of attributes. The local data is collected from these printed attributes that include the main features used during diagnosis. The dataset includes total 52 records with 11 attributes. The final attribute is the annotation an it is either Normal or Abnormal. All of the attributes are common with UCI Dataset. The dataset contains some critical attributes like Heart Rate, PR and QRS Intervals. Name of attributes are shown in Table 5.1.

Table 5.1. Local Dataset Features

ID	Description
1	Heart Rate
2	PR Intervals
3	QRS Intervals
4	QT Interval
5	OTc Interval
6	P Axis
7	QRS Axis
8	T Axis
9	RV5 Amplitude
10	SV1 Amp
11	Annotation

The local dataset is appropriate for binary classification and multi-class methods are not applicable it. 23 records (44.2%) are annotated as Abnormal and 29 are Normal. The dataset is more balanced when compared with UCI dataset.

6. EXPERIMENTAL RESULTS

6.1. Introduction

Each stage of ECG Arrhythmia Classification uses similar methods and algorithms to create an optimized output. The input parameters have to be optimized for achieving more accurate and realistic results. Grid-search [41] technique is mostly used during the search for the optimized parameters. In grid-search basically pairs of parameters are tried and the parameters providing best accuracy are selected. In this study, some common metrics used in previous machine learning applications are evaluated to show the performance of dimensionality reduction and classification accuracy. Details of metrics are given in section 6.2.

Creating training and test sets is another accuracy problem. Cross-validation is one of the common methods used when the datasets are not large enough to separate training and testing datasets. In k-fold cross validation, dataset is divided into *k* equal size subsets. One of the subsets is used for testing and the remaining for training. Each time a different subset is used for testing and the process is repeated *k* times. Typically *k* is between 10 and 30 [21] and during this study 10-fold cross-validation is applied to datasets.

Both of the arrhythmia datasets contain numerical attributes and they are in
[-177, 524]. All attributes are normalized to [0, 1] before MLP, DT and SVM classification algorithms are applied. For the class-dependent subset selection with RELIEF, features are normalized to [-1, +1]. The attributes in the top of the list are selected and then transformed for the classification algorithms.

All the learning algorithms are implemented in Weka [42]. Weka’s built-in MLP, DT, PCA, RELIEF functions and Weka interface of LibSVM [43] were used to implement the learning and dimensionality reduction algorithms.

6.2. Performance Metrics

Different performance metrics can be used to show the classification results. Confusion matrix and proportions derived from entries in this matrix are commonly used in previous works [5, 19, 24]. Confusion matrix shows the predictions correct values of a class. A confusion matrix is shown in Table 6.1.

Table 6.1. Confusion Matrix

		Predicted Class	
		Positive	Negative
True Class	Positive	**TP**	**FN**
	Negative	**FP**	**TN**

The entries in the confusion matrix are:

- True positive (TP): The number of correct predictions that class is positive
- False negative (FN): The number of incorrect predictions that class is negative.
- False positive (FP): The number of incorrect predictions that a class is positive.
- True negative (TN): The number of correct predictions that class is negative.

Sensitivity (also called Recall, TP Rate) measures the proportion of positives which are correctly classified. Sensitivity is calculated according to Formula 6.1.

$$Sensitivity = \frac{TP}{TP + FN} \tag{6.1}$$

Specificity is another metric measures the proportion of negatives which are correctly identified. Formula 6.2 shows the calculation of specificity.

$$Specificity = \frac{TN}{TN + FP} \tag{6.2}$$

Receiver operating characteristic (ROC), or ROC curve, is a graphical plot of the *sensitivity* versus *(1 − specificity),* also called as *TPR* versus *FPR.* A ROC space depicts trade-offs between true positives and false positives. Each prediction result represents one point in the ROC space. The classification performance of points A, B and C plotted in Figure 6.1 show the usage of ROC space. Point (0, 1) is perfect classification and the diagonal line shows the perfect random guess. Point A has better performance than B and C. But an inverse selection may be applied to reach point C'.

The Area Under Curve (AUC) is also another useful metric that shows the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one.

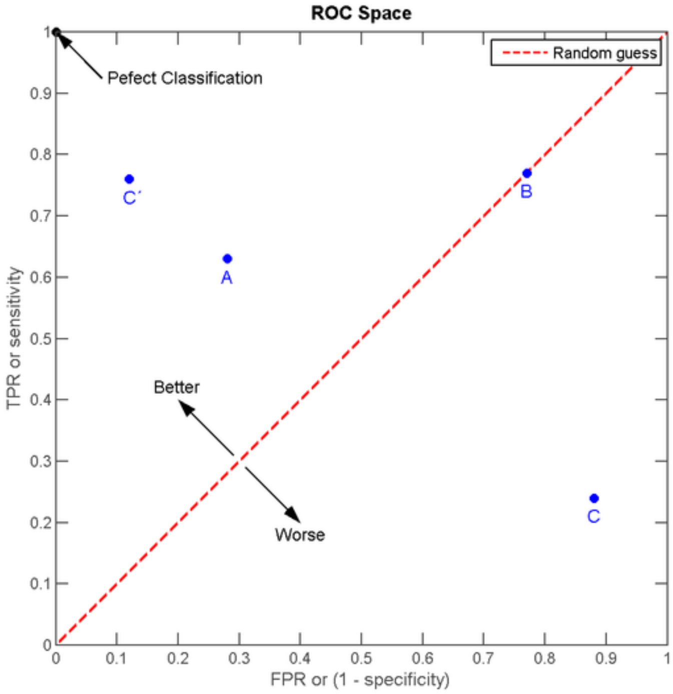


Figure 6.1. Sample ROC Space [44]

F-score is a statistical measure of a test's accuracy and it is commonly used in similar studies [45, 46, 47]. It considers both the precision and the recall of the test to compute the score. Precision shows number of items correctly classified as belonging to the positive class.

$$Precision = \frac{TP}{TP + FP} \tag{6.3}$$

The F-score can be interpreted as a weighted average of the precision and recall, where an F1 score reaches its best value at 1 and worst score at 0.

$$F - score = 2x \frac{Precision x Recall}{Precision + Recall} \tag{6.4}$$

6.3. Parameter Selection and Class-Module Optimization

Dataset and its structure also affect accuracy like input parameters. The imbalanced datasets affect classification accuracy and performance start to deteriorate even with small imbalances [48, 49]. UCI dataset has an imbalanced nature as shown in Chapter 2. The instances can be grouped as shown in Table 6.2 depending on the class they belong to.

Table 6.2. UCI Dataset Instance Groups

Type	Class Codes	Min-Max %	Total %
Normal	1	53.5%	53.5%
Group 1	02, 06, 10	5.6-9.4%	26.7%
Group 2	03, 04, 05	2.9-3.3%	9.7%
Group 3	Rest	up to 1.9%	10.1%

Using this grouping information, Group 2 and Group 3 classes and related instances are removed from UCI dataset to achieve a minimum representation rate of 7% on reduced dataset, which is acceptable for a 10-fold cross validation and still useful for a multi-class learning problem. The final dataset class distributions are shown in Table 6.3.

Table 6.3. Class Distribution of Reduced UCI Dataset

Code	**Class**	**# of Instances**
1	Normal	239
2	Ischemic Changes (Coronary Artery Disease)	44
6	Sinus Bradycardy	25
10	Right Bundle Branch Block	50
	Total	*358*

All learning algorithms were tested on Reduced UCI Dataset with 278 features. For SVM testing, $C = 4$ and $\gamma = 0.003$, for MLP tests *Learning Rate = 0.3* and *Momentum = 0.2* was used to see performance of all classifier algorithms with 278 features. In addition to that three MLP configurations with 2, 3 and 4 hidden nodes were used.

Class-Modular MLP (CM-MLP) is built by 4 modules, each one having a MLP with 2-4 hidden nodes. The initial tests were performed for different number of hidden nodes to see the effect of hidden nodes on performance and evaluate the trade-off between the additional complexity a hidden node brings and the accuracy provided by that node.

Each module of CM-MLP is optimized with class-dependent subset selection before applying the 10-fold cross-validation. Four separate tests for each module were performed by top 5, 10, 15 and 20 of 278 attributes selected by RELIEF as the input attribute subset. Table 6.4 shows the attributes used for each module.

Table 6.4. Feature Quality Estimation for all Modules

Module Class1		Module Class2		Module Class6		Module Class10	
Quality Est.	Feature Id	Quality Est.	Feature Id	Quality Est.	Feature Id	Quality Est.	Feature Id
0.0479396	91	0.1417867	197	0.13433370	15	0.21804945	91
0.0424214	93	0.1195856	267	0.12884615	2	0.16643615	93
0.0356239	15	0.1191083	277	0.11065456	40	0.11506651	90
0.0318681	2	0.1059538	167	0.08637123	57	0.10173196	57
0.0247863	103	0.0966557	11	0.07736832	41	0.10028998	103
0.0229814	40	0.0958791	2	0.07673992	42	0.09590528	30
0.0191502	197	0.0895882	177	0.07578492	136	0.08663461	53
0.0177993	90	0.0879032	257	0.06785714	47	0.07848901	18
0.0171978	53	0.0740444	40	0.06691235	184	0.07839972	78
0.0170807	52	0.0651099	47	0.06682692	43	0.06575091	42
0.0166338	167	0.0621476	57	0.06236263	35	0.05827407	69
0.0162088	47	0.0616903	260	0.06161986	150	0.05740659	65
0.0160251	191	0.0601966	279	0.06129120	53	0.05659340	2
0.0158112	260	0.0583455	269	0.06030873	30	0.05530719	64
0.0153727	277	0.0571928	247	0.05729548	148	0.05273707	150
0.0147390	54	0.0552198	123	0.05691591	66	0.05257242	88
0.0145739	7	0.0546703	66	0.05480769	78	0.04479595	224
0.0143250	30	0.0536892	42	0.05178571	88	0.04445771	40
0.0141484	18	0.0528894	41	0.05037593	90	0.04168657	66
0.0140270	267	0.0509512	199	0.05014285	65	0.04031830	41

It could be observed from Table 6.4 that, each class has its own unique best information carrying subset although there are some attributes that are common for one or more classes. Attributes 53, 90, 91, 93 and 103 are both included in the 10-attribute subset tests of Module Class 1 and Module Class 10. But their contribution to the learning process and relative importance within all attributes are not same. The results of 2-4 hidden node runs for 5, 10, 15 and 20 attribute subset sizes are given in Tables 6.5, 6.6, 6.7 and 6.8.

Table 6.5. CM-MLP Correctly Classified Instances with 5 features

	TP for Classes				
Hidden Nodes	1	2	6	10	Total
2	227	30	14	41	312
3	224	30	13	38	305
4	227	32	13	38	310

Table 6.6. CM-MLP Correctly Classified Instances with 10 features

	TP for Classes				
Hidden Nodes	1	2	6	10	Total
2	204	30	15	40	289
3	210	28	15	34	287
4	205	28	14	36	283

Table 6.7. CM-MLP Correctly Classified Instances with 15 features

	TP for Classes				
Hidden Nodes	1	2	6	10	Total
2	209	33	16	30	288
3	202	34	15	32	283
4	203	32	14	31	280

Table 6.8. CM-MLP Correctly Classified Instances with 20 features

	TP for Classes				
Hidden Nodes	1	2	6	10	Total
2	208	29	14	33	284
3	206	26	13	29	274
4	203	23	14	32	272

It is observed from Tables 6.5 to Table 6.8 that there is no direct relation between the number of hidden nodes and the number of correctly classified instances. Increasing the number of hidden nodes doesn’t affect the accuracy of learning algorithm for UCI dataset. One possible explanation of that may be overfitting due to increasing complexity. As the number of hidden units increase in a MLP network, the model memorizes the noise in training set and becomes overcomplex to generalize the validation set [21].

The effect of feature subset size on learning accuracy is one of the information that may be derived from these four tables. There is no constant relation observed between the input subset size and the TP rate for all classes. CM-MLP modules were able to detect arrhythmias almost with the same accuracy both with 5, 10, 15 and 20 feature subsets. In addition to that, the total number of correctly classified instances decreases as the subset size increases for all hidden number node parameters used in tests. The TP rate decreased for all abnormal rhythms when the subset size is increased from 5 to 15, 10 and 20 features, but the degradation in performance is relatively low. The same accuracy change was also observed when subset size is increased from 10 to 15 and 20. Usually most important part of the information is carried by most meritful attributes. If these attributes are capable of separating the classes, adding extra attributes will not contribute to prediction accuracy.The ROC curves of 10 and 20 subset size for all classes with 2 hidden nodes are shown in Figure 6.2, 6.3, 6.4 and 6.5 and they provide how TPR and FPR affected by subset size.

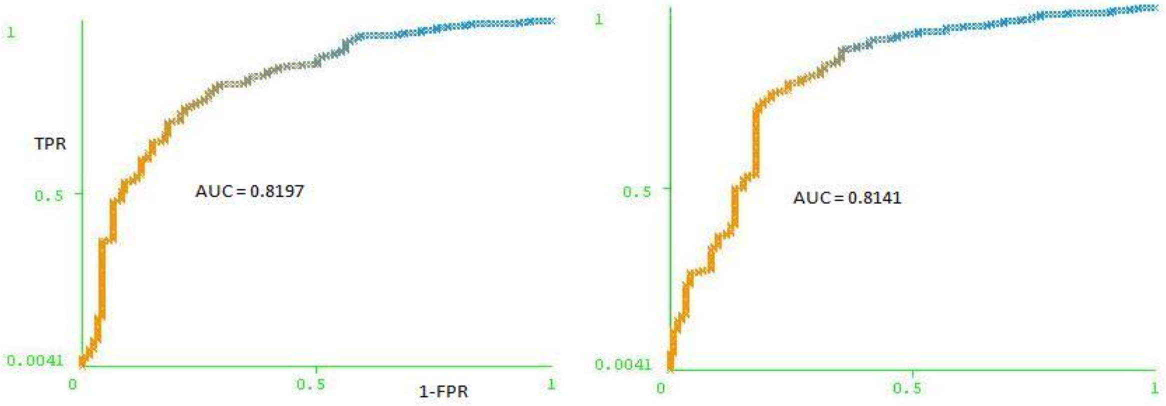


Figure 6.2. ROC Curves for Class 1

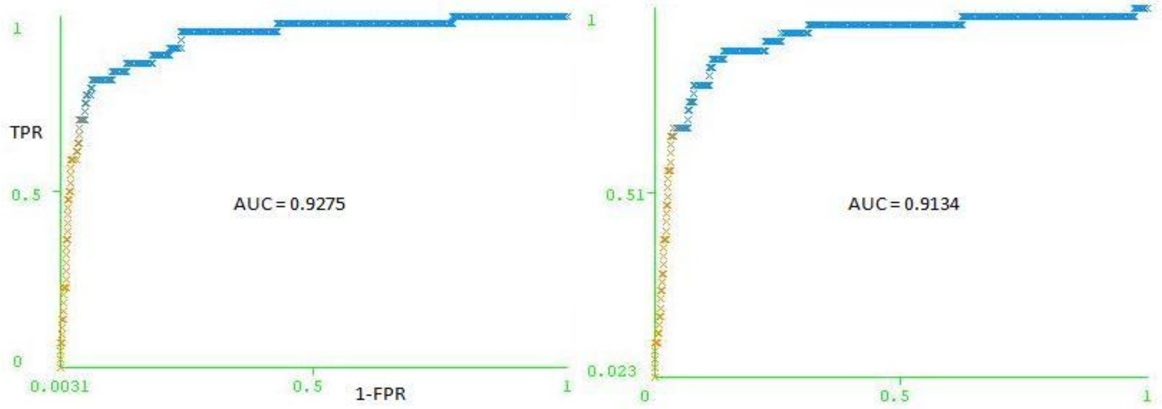


Figure 6.3. ROC Curves for Class 2

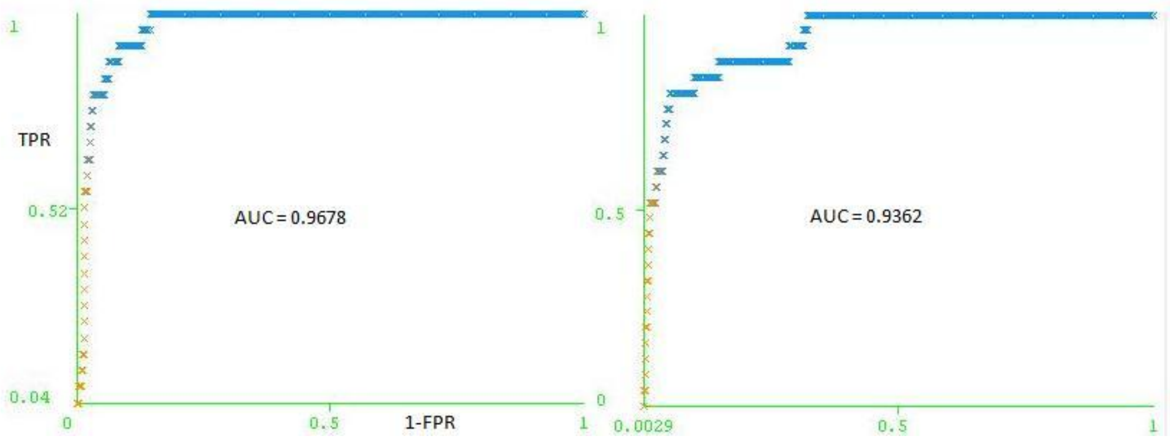


Figure 6.4. ROC Curves for Class 6

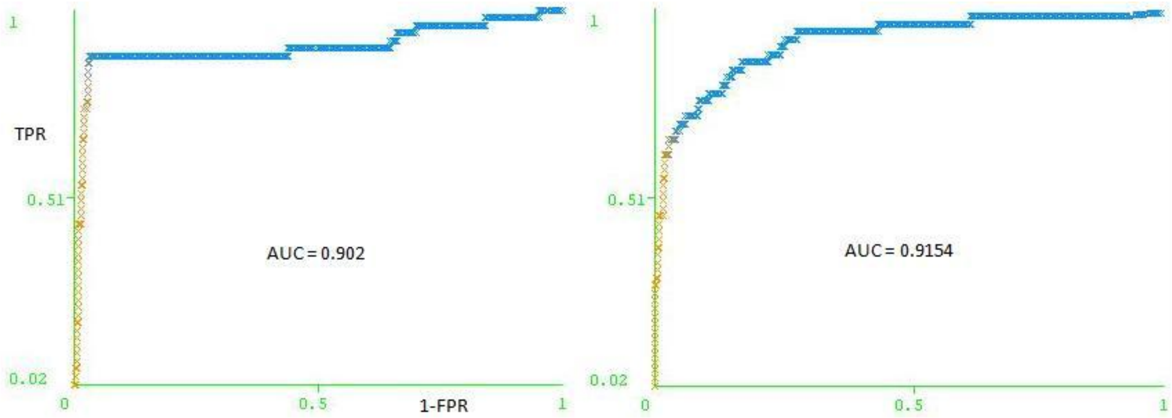


Figure 6.5. ROC Curves for Class 10

All ROC curves show that there is no significant sensitivity vs. specificity change in all modules depending on the input subset size. The AUC of all arrhythmia classes 2, 6 and 10 shows that these modules will identify a randomly chosen positive instance with a higher probability than a randomly chosen negative one. All of the curves are far higher than random guess line. The CM-MLP has very high AUC values between [0.902, 0.9678]. The f-scores of CM-MLP modules may also show the success to identify the arrhythmias. Table 6.9, 6.10, 6.11 and 6.12 show the confusion matrix, f-score, sensitivity (recall) and precision for all CM-MLP runs with 2 hidden nodes with standard deviations of 10-fold runs.

Table 6.9. Detailed Accuracy Parameters for CM-MLP with 5 features

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	227	65	54	12	0.94±0.005	0.54±0.014	0.80±0.009	0.87±0.006
Class 2	30	303	11	14	0.68±0.024	0.96±0.003	0.73±0.014	0.70±0.011
Class 6	14	328	5	11	0.56±0.011	0.98±0.008	0.73±0.006	0.63±0.014
Class 10	41	297	11	9	0.82±0.020	0.96±0.008	0.78±0.006	0.80±0.009

Table 6.10. Detailed Accuracy Parameters for CM-MLP with 10 features

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	204	69	50	35	0.85±0.004	0.57±0.018	0.80±0.008	0.82±0.005
Class 2	30	301	13	14	0.68±0.027	0.95±0.007	0.69±0.011	0.68±0.015
Class 6	15	326	7	10	0.60±0.019	0.97±0.007	0.68±0.008	0.63±0.011
Class 10	40	299	9	10	0.80±0.025	0.97±0.004	0.81±0.011	0.80±0.015

Table 6.11. Detailed Accuracy Parameters for CM-MLP with 15 features

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	209	70	49	30	0.87±0.002	0.58±0.015	0.81±0.008	0.84±0.003
Class 2	33	297	17	11	0.75±0.021	0.94±0.006	0.66±0.008	0.70±0.014
Class 6	16	323	10	9	0.64±0.040	0.97±0.006	0.61±0.018	0.62±0.022
Class 10	30	298	20	10	0.75±0.024	0.93±0.006	0.60±0.008	0.66±0.014

Table 6.12. Detailed Accuracy Parameters for CM-MLP with 20 features

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	208	77	42	31	0.87±0.002	0.64±0.018	0.83±0.007	0.85±0.003
Class 2	29	302	12	15	0.65±0.026	0.96±0.005	0.70±0.007	0.68±0.011
Class 6	14	323	10	11	0.56±0.062	0.97±0.004	0.58±0.0015	0.57±0.024
Class 10	33	298	10	17	0.66±0.026	0.96±0.004	0.76±0.009	0.70±0.013

Sensitivity shows the learning method's accuracy in detecting positive instances. For all arrhythmia types, the CM-MLP's prediction power decreases as the feature subset size increases. In contrast to that, there is a small increase in sensitivity of normal ECG instances. Specificity has similar decrease for all arrhythmia types. Specificity shows the algorithm's accuracy to detect negative instances. In our problem negative instances are normal ECG records and that makes sensitivity prior to specificity. Right Bundle Branch Block (Class 10) has the highest prediction rate within all arrhythmias with highest sensitivity value as 0.82. The precision rate, which shows the correctly predicted positive classes, also has the highest value for that arrhythmia type. If we examine the f-score values, the general accuracy of algorithm in Right Bundle Branch Block may be also observed. F-score value of 0.80, which is significantly higher than 0.5 shows that how algorithm is both predict the arrhythmia and separate the normal ECG rhythm. The general accuracy measured by f-score shows that for all arrhythmia detection modules, the increasing subset size also decreases the performance.

The findings from both accuracy measurements and ROC curves show that

- Increasing the hidden nodes does not increase prediction performance and
- Using more attributes does not increase the overall accuracy of CM-MLP modules for UCI dataset with 278 attributes.

Using 2 hidden nodes with 10 feature subset for each module may show the performance of CM-MLP algorithm relative to DT, SVM and MLP. 10 attributes may also provide efficiency if module optimization costs are considered as an important factor.

6.4. Performance Evaluation of CM-MLP with 278 Attributes

The module based accuracy measurements shown in previous section may be misleading due the imbalanced nature of input dataset, different number of instances in the training data and the MCA that combines the output of all modules. The accuracy of well-known learning algorithms may be used to compare the overall performance of CM-MLP on the UCI dataset. For the overall accuracy tests, SVM, DT and MLP are trained on the optimized dataset with 278 attributes. The performance of CM-MLP with 10 features was shown in section 6.3. In that architecture 10 most meritful features selected by RELIEF were used as input to the modules. Using 10 of 278 features provides a (10*4)/278 = 1/7 efficiency for the following calculations

- Updating weights in Modular MLPs
- Backward propagation
- Weighted sum of inputs and the output of sigmoid

In order to have a reliable learning algorithm, these gains from processing time should not be offset by a decrease in overall prediction accuracy. Table 6.13 shows the total successful predictions for CM-MLP and other algorithms.

Table 6.13. CM-MLP overall performance for 278 features

Method	Correctly Classified Instances	%
DT (J48)	280 ±5.09	78.21
SVM(RBF)	287 ±0.60	80.17
MLP2 (2 Hidden)	275 ±4.10	76.82
MLP3 (3 Hidden)	270 ±4.21	75.42
MLP4 (4 Hidden)	270 ±4.25	75.42
CM-MLP(1/7 efficiency)	289 ±2.17	80.73

From Table 6.13 it can be decided that the performance of comparison algorithms are similar with 278 features and 4 classes. CM-MLP has almost equal prediction accuracy with best performing SVM. Due to the similar performances of all learning algorithms, most successful two methods on each class were selected for comparative evaluation. The strong prediction rate on one of the classes may lead to higher total accuracy, but weaker performances on other classes should also be avoided.

Table 6.14. Highest Two Sensitivities for overall performance for 278 features

Class	Highest Sensitivity	Methods
1	0.947-0.927	SVM - MLP2
2	0.6818-0.545	CM-MLP - DT
6	0.720-0.6000	DT - MLP3/CM-MLP
10	0.8-0.740	CM-MLP-SVM

Both Table 6.13 and 6.14 shows that, Class-Modular MLP has similar prediction rate with other learning methods (the class-modular run results are given in Appendix C). When the sensitivity of MLP and its class-modular implementation are compared, class-modularity provides a minimum TP rate for all classes. The modularity provides an advantage to classical approach, with this minimum classification rate. The sensitivities for both approaches are shown in Table 6.15. For a Sinus Bradycardy (Class 6), no arrhythmias can be detected with MLP even different number of hidden nodes is used. CM-MLP can detect 15 of 25 patients.

Table 6.15. CM- MLP and MLP Sensitivities for 4 classes

Sensitivity (TP Rate)		
Class	***CM-MLP***	***MLP2***
1	0.854	0.927
2	0.682	0.523
6	0.600	0
10	0.800	0.62

The ROC Curves of CM-MLP and MLP2 for all classes are shown in Figure 6.6, 6.7, 6.8 and 6.9. It can be observed from these figures how prediction capability of CM-MLP is similar to and sometimes better than MLP2. When the AUC of both curves are examined, the CM-MLP's AUC is 0.789 while MLP2 has 0.746. The CM-MLP observed to be more efficient to detect the Normal ECG rhythms (Class 1) with these statistics.

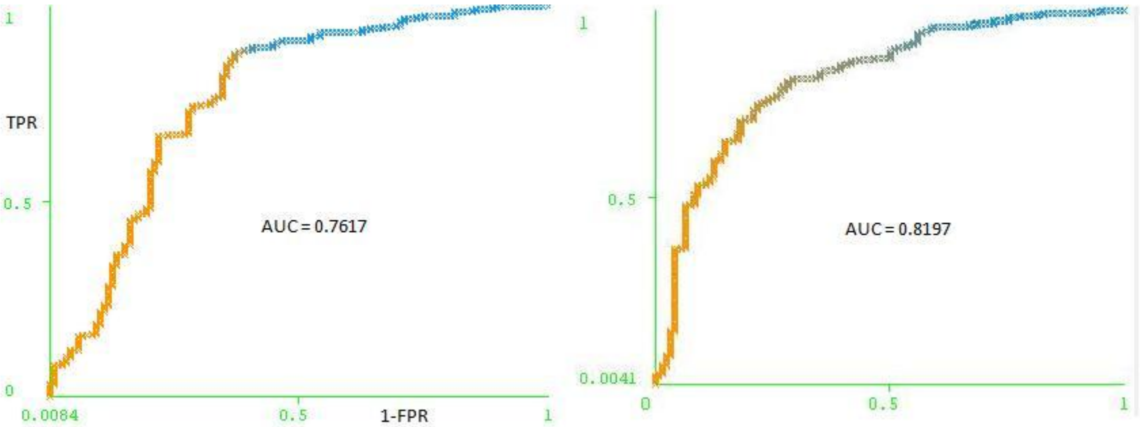


Figure 6.6. MLP2 and CM-MLP ROC Curves for Class 1

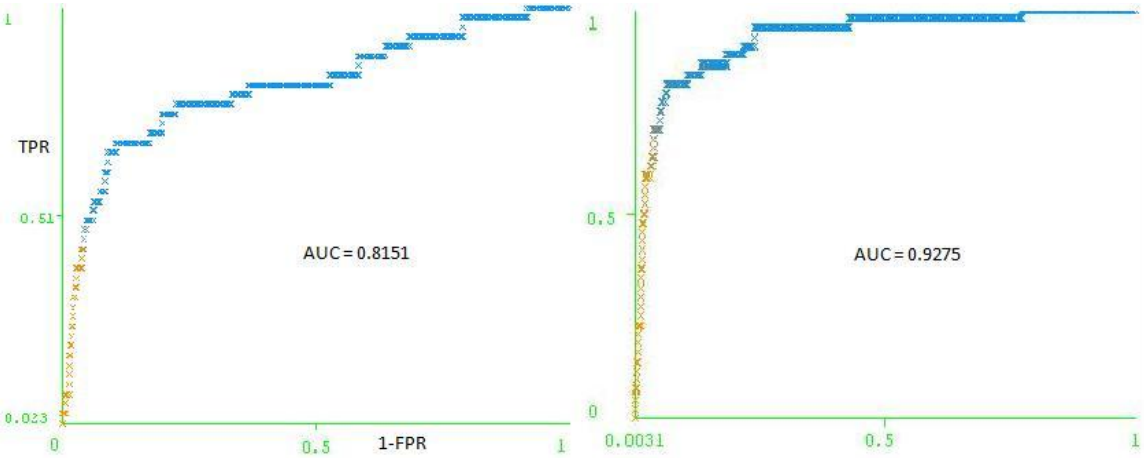


Figure 6.7. MLP2 and CM-MLP ROC Curves for Class 2

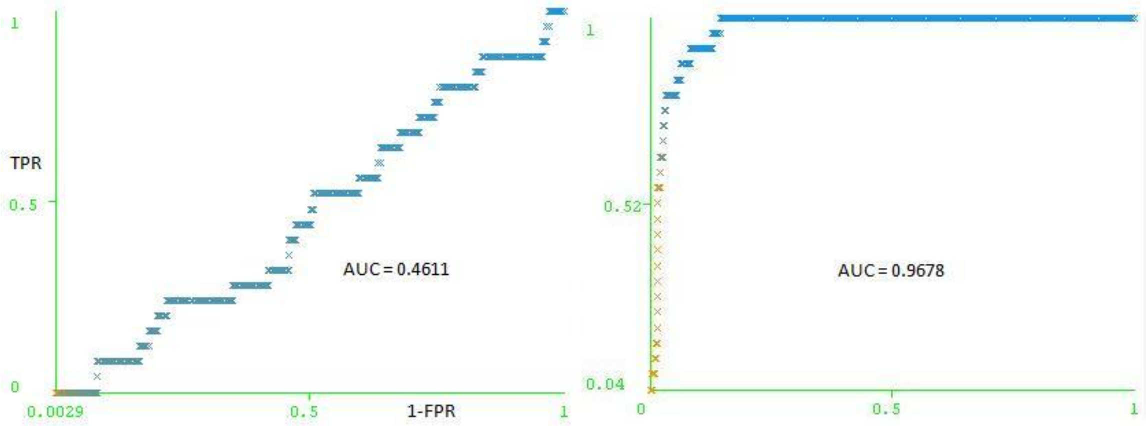


Figure 6.8. MLP2 and CM-MLP ROC Curves for Class 6

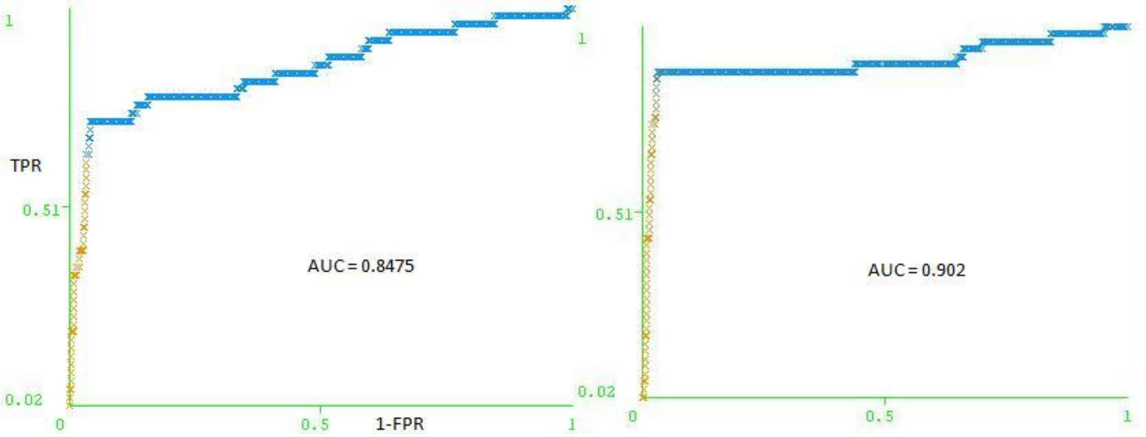


Figure 6.9. MLP2 and CM-MLP ROC Curves for Class 10

In all of the four figures above, the AUC of ROC Curves for CM-MLP is higher than the MLP curve. Especially for the Sinus Bradycardy (Class 6), the MLP algorithm performs worse than the random guess. The customized parameters of Class 6 module prevent the occurrence of same error in CM-MLP algorithm. This is also consistent with the minimum prediction rate provided by modular architecture that was also shown in Table 6.15. The sensitivity of MLP2 was higher than CM-MLP for normal ECG rhythms. But the negative effect of classification errors in arrhythmias took the weighted performance below than that of CM-MLP.

Weighted f-scores of learning algorithms can be calculated to provide a measurement for overall accuracy. MLP2 has 0.738 weighted f-score while CM-MLP has 0.794. Despite the fact that MLP2 has higher f-score for Normal ECG, the significant prediction accuracy gap between CM-MLP and MLP2 in arrhythmias created a higher probability to correctly estimate the patient instances from healthy ones.

The tests performed with MLP networks of 2-4 hidden nodes again showed that, the extra complexity added by new hidden nodes does not increase the performance or prediction accuracy. For the following tests, both CM-MLP and MLP networks with 2 hidden nodes were used.

6.5. Performance Evaluation of CM-MLP on Reduced UCI Dataset

278 features of UCI Dataset provide powerful information for classification. But it may be useful to use reduced dataset if achieved accuracy does not deteriorate. It was shown before that successful method of removing irrelevant features increase efficiency in medical applications and increase prediction rate [22]. During this study three methods applied for dimensionality reduction.

- Feature Extraction with PCA
- Feature Selection with DT
- Feature Selection with SVM-RFE

6.5.1. Feature Extraction with PCA

PCA tries to maximize variance of attributes and use covariance matrix of input variables for eigen analysis. After applying PCA to cover 0.95 variance of original dataset, the resulting dataset consists of 94 attributes. The first 10 Eigenvalues for new dimensions are shown in Table 6.16. All Eigenvalues are given in Appendix B.

Table 6.16. Eigenvalues of Dimensions

Dimension	Eigenvalue
1	23.04833
2	16.42062
3	13.30601
4	11.78891
5	9.73232
6	7.34801
7	6.99246
8	6.61948
9	5.81401
10	5.31559

All of the learning tecniques were tested on new dataset to see the effect of PCA on classification accuracy. All 94 attributes were used for SVM, MLP and DT tests. CM-MLP modules were fed with 5, 10, 15 and attributes. These extra dimensionality reduction provides (5*4)/94 = 1/ 4.70 efficiency for 5 attributes, and 1/2.35, 1/1.56 and 1/1.17 respectively on calculations. Table 6.17 shows the total successful predictions for all attribute subset sizes.

Table 6.17. Performance of CM-MLP with Different Attribute Sizes

Attributes	Class 1	Class 2	Class 6	Class 10	Total
5	212	15	3	22	252
10	205	27	3	25	260
15	207	25	4	25	261
20	206	24	3	26	256

It can be observed from Table 6.17 that, as the attribute subset size increases, there is so significant increase in TP for all classes. In addition to that, increasing attribute subset negatively affect the performance of modules due to increasing cost related with module optimization. The efficiency drops to half value when subset size is increased from 10 to 20. As a result of that 10 is selected as subset size for bencmarking with other learning algorithms. Table 6.18 shows the total successful predictions for CM-MLP and other algorithms for 10-fold cross-validation.

Table 6.18. Performance of all Algorithms for PCA Feature Extraction

Method	Correctly Classified Instances	%	Change relative to 278 features
DT	219 ±4.72	61.17	-61
SVM	289 ±0.62	80.73	2
MLP	251 ±0.60	70.11	-24
CM-MLP	260 ±1.93	72.63	-29

PCA negatively affected the total prediction accuracy of DT, MLP and CM-MLP. But, it has no significant effect on SVM's classification accuracy. Both neural network based algorithms reacted similarly to the use of extracted information. Additionally, CM-MLP has higher total TP rates for all classes. The detailed accuracy parameters of CM-MLP are shown in Table 6.19. Table 6.20 shows change in Table 6.19 relative to the classification with 278 features and it is calculated by subtracting Table 6.10 from Table 6.19.

Table 6.19. Detailed Accuracy Parameters for CM-MLP with PCA

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	201	52	67	38	0.84±0.006	0.43±0.035	0.75±0.012	0.79±0.009
Class 2	25	297	17	19	0.56±0.045	0.94±0.010	0.59±0.018	0.58±0.025
Class 6	3	324	9	22	0.12±0.046	0.97±0.012	0.25±0.014	0.16±0.021
Class 10	25	296	12	25	0.50±0.029	0.96±0.011	0.67±0.017	0.57±0.022

Table 6.20. Δ of Detailed Accuracy Parameters after PCA

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	-3	-17	17	3	-0.0126	-0.1429	-0.0531	-0.0347
Class 2	-5	-4	4	5	-0.1136	-0.0127	-0.1024	-0.1083
Class 6	-12	-2	2	12	-0.4800	-0.0060	-0.4318	-0.4761
Class 10	-15	-3	3	15	-0.3000	-0.0097	-0.1407	-0.2334

There is a significant performance decline in all of the accuracy parameters for CM-MLP after using PCA feature extraction. The most significant decrease was driven by the 0.48 decrease in sensitivity of Class 6 arrhytymia (highlighted in tables) followed by a 0.30 decrease in the sensitivity of Class 10 arrhytmia. Specificity was not affected since it was mostly dependent on prediction capability of normal ECG rhythms. It can be also observed that the performance decline is not related with the amount of instances in the training and testing data. Class 10 has the highest instance rate after normal ECG and it was severely affected like the Class 6, which has the minimum number of instances in the sample space.

The shift in ROC curves for Class 6 and Class 10 may be another indicator of prediction capability. Figure 6.10 shows the shift in the ROC Curve for Class 6 for full dataset and reduced dataset with PCA. Figure 6.11 shows the same shift for Class 10. The shift in both curves is driven by the decrease in sensitivity.

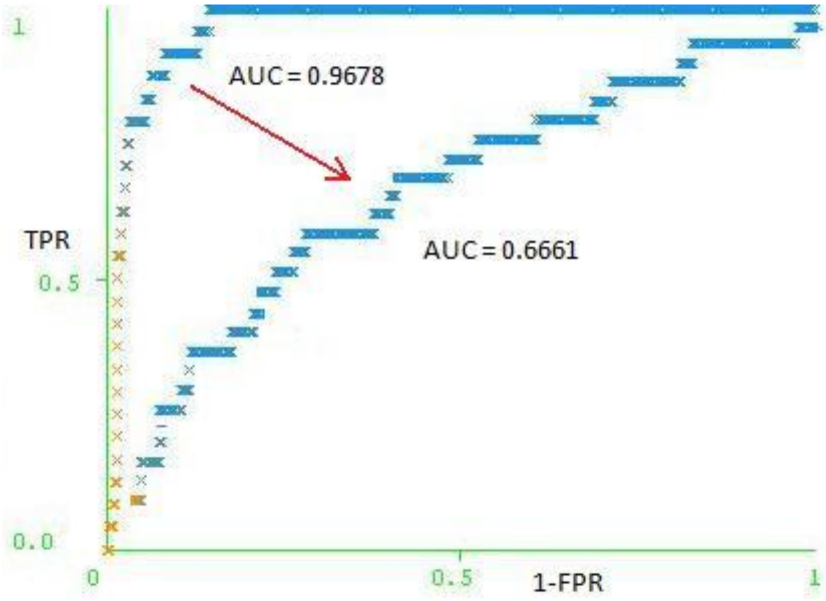


Figure 6.10. Shift in ROC Curve for Class 6

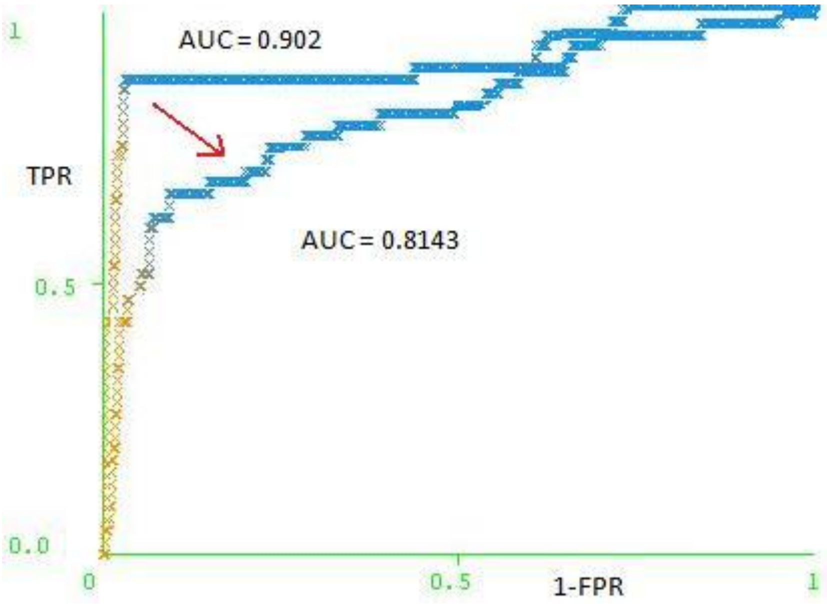


Figure 6.11. Shift in ROC Curve for Class 10

The decrease in prediction accuracy was not a specific problem of neural network based algorithms. DT based learning was affected negatively more than MLP and CM-

MLP. The number of total correctly classified instances dropped by 21.7% of it was in 278 features. Table 6.21 shows the detailed accuracy parameters of DT on feature extracted dataset. The f-score values of arrhythmia classes are very low and the DT was not able to detect abnormal ECG rhythms, especially Class 6 type. CM-MLP was better than DT both in sensitivity and specificity for all classes either normal or abnormal.

Table 6.21. Detailed Accuracy Parameters for DT with PCA

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	183	48	71	56	0.76±0.014	0.40±0.032	0.72±0.014	0.74±0.013
Class 2	13	300	14	31	0.29±0.041	0.95±0.015	0.48±0.011	0.36±0.018
Class 6	4	302	31	21	0.16±0.041	0.90±0.015	0.11±0.007	0.13±0.011
Class 10	19	285	23	31	0.38±0.033	0.92±0.016	0.45±0.009	0.41±0.014

If highest 2 sensitivities for all classes are calculated as shown in Table 6.22, CM-MLP generally performs slightly better than MLP.

Table 6.22. Highest Two Sensitivity for overall performance for PCA

Class	**Highest Sensitivity**	**Methods**
1	0.95-0.84	SVM - CM-MLP/MLP
2	0.57-0.52	CM-MLP - SVM
6	0.28-0.16	MLP/DT
10	0.76-0.50	SVM - CM-MLP

6.5.2. Feature Selection with DT

Decision trees both used for classification and feature selection. Weka has a built-in implementation of C 4.5 and it is called J48 trees. DT feature selection method was applied to UCI dataset to receive the most important 9 attributes based on merit. The selected attributes are shown in Table 6.23.

Table 6.23. Attributes Selected with DT Feature Selection

Merit Index	Attribute
1	Heart Rate
2	Existence of diphasic derivation of P wave (DI)
3	Average width of Q wave(AVF)
4	Average width of S wave (V3)
5	Amplitude of T wave (AVL)
6	Amplitude of R' wave (V1)
7	Amplitude of R' wave (V2)
8	Amplitude of T wave (V5)
9	Amplitude of JJ wave (V6)

Figure 6.12 shows the correctly classified instances for all techniques. Table 6.24 shows the total correctly classified instances for all algorithms with the weighted f-score values.

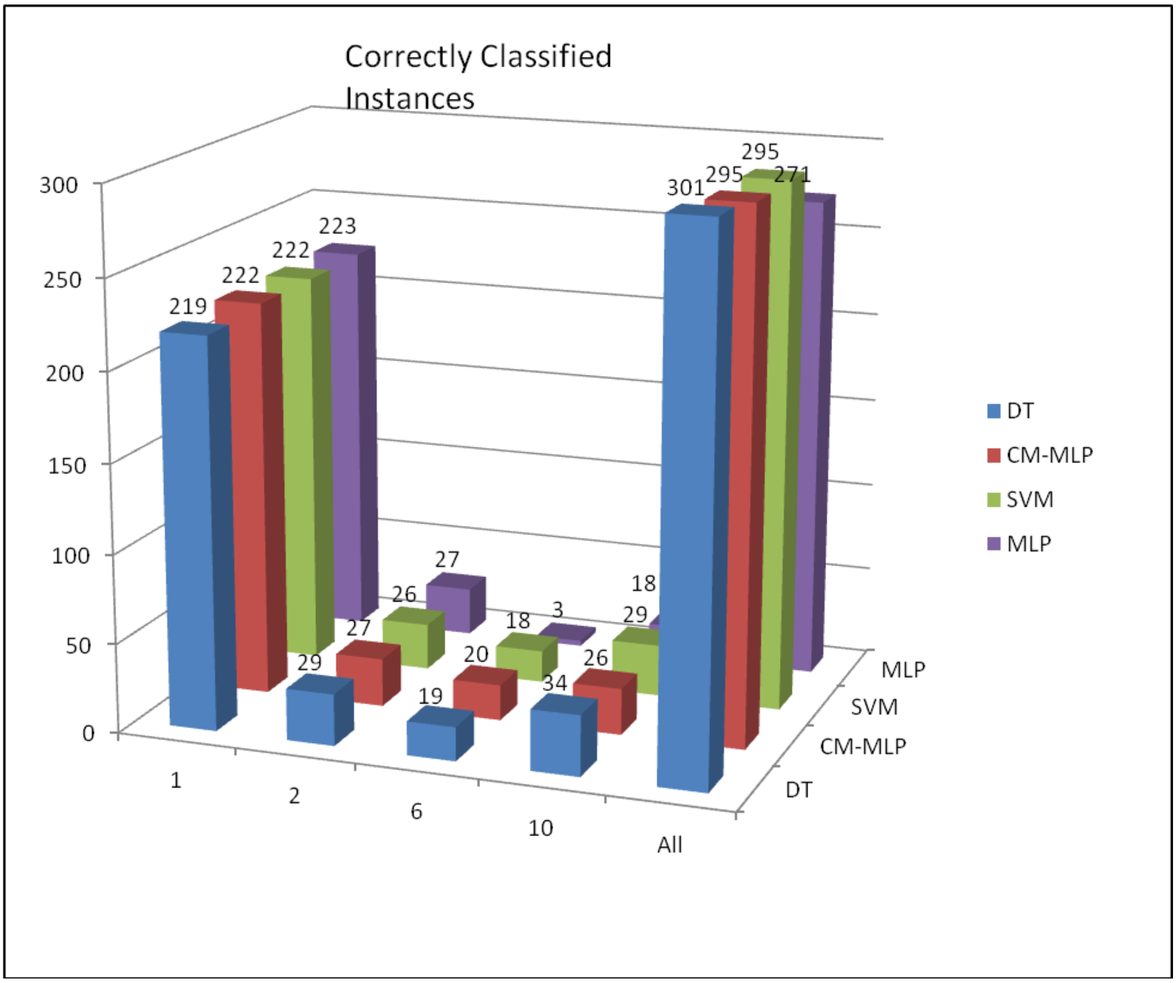


Figure 6.12. Correctly Classified Instances for Reduced DT Dataset

Table 6.24. Performance of all Algorithms for DT Feature Selection

Method	Correctly Classified Instances	%	Weighted f-score
DT	301 ±2.19	84.08	0.8386
SVM	295 ±0.41	82.40	0.8175
MLP	271 ±2.74	75.70	0.7295
CM-MLP	295 ±3.16	82.40	0.8135

CM-MLP has very close overall accuracy to DT based learning, which has the highest prediction rate for the reduced dataset. The general prediction capabilities of DT, SVM and CM-MLP are similar to each other. MLP has some problems associated with the prediction of Sinus Bradycardy (Classes 6) arrhythmias. Table 6.25 shows the detailed accuracy parameters for MLP on reduced dataset.

Table 6.25. Detailed Accuracy Parameters for MLP with DT Feature Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	223	64	55	16	0.93±0.011	0.53±0.026	0.80±0.010	0.86±0.010
Class 2	27	296	18	17	0.61±0.029	0.94±0.013	0.60±0.025	0.60±0.026
Class 6	3	326	7	22	0.12±0.045	0.97±0.011	0.30±0.022	0.17±0.029
Class 10	18	301	7	32	0.36±0.021	0.97±0.013	0.72±0.031	0.48±0.027

The sensitivity of MLP for Sinus Bradycardy was the main reason of low f-score values. Like the problem occurred in CM-MLP for dataset created with PCA feature extraction, MLP is not able to successfully detect Class 6 arrhythmias. Table 6.26 shows detailed accuracy parameters for CM-MLP. If two tables are compared, it can be observed that the sensitivity and precision values of CM-MLP for Class 6 were main drivers of performance. ROC Curves of Sinus Bradycardy prediction in Figure 6.13 also shows how CM-MLP was successful in prediction. The ROC Curve of CM-MLP is very close to perfect classification point (1, 0), where no FNs are occurred.

Table 6.26. Detailed Accuracy Parameters for CM-MLP with DT Feature Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	222	81	38	17	0.92±0.013	0.68±0.023	0.85±0.006	0.88±0.007
Class 2	27	299	15	17	0.61±0.033	0.95±0.008	0.64±0.014	0.62±0.019
Class 6	20	327	6	5	0.80±0.013	0.98±0.009	0.76±0.005	0.78±0.007
Class 10	26	301	7	24	0.52±0.030	0.97±0.012	0.78±0.015	0.62±0.019

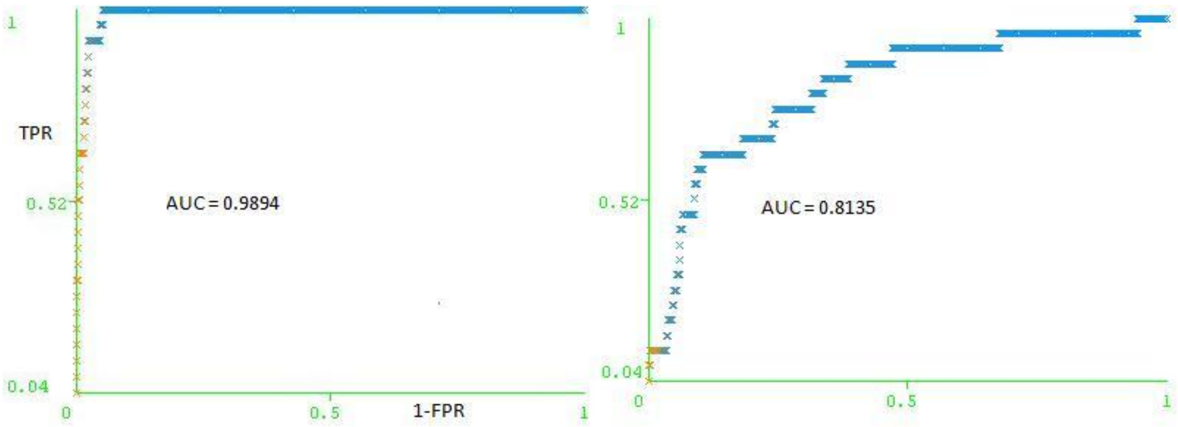


Figure 6.13. CM-MLP and MLP ROC Curves for Class 6

The highest 2 sensitivities of algorithms may be used to evaluate the general prediction accuracy of all algorithms. From Table 6.27, CM-MLP is generally performing better than MLP and in some classes it is also better than SVM. If results from 278 attributes, PCA and DT reduced sets, SVM has the best overall performance.

Table 6.27. Highest Two Sensitivity for overall performance for DT

Class	**Highest Sensitivity**	**Methods**
1	0.93-0.92	MLP - CM-MLP/SVM
2	0.66-0.61	DT - CM-MLP/MLP
6	0.80-0.76	CM-MLP/SVM
10	0.68-0.58	DT - SVM

6.5.3. Feature Selection with SVM-RFE

Like DT, SVM may be used for both classification and feature selection. SVM implementation of LibSVM includes SVM-RFE feature selection method. 9 most important attributes are selected from 278 attributes. The selected attributes are shown in Table 6.28.

Table 6.28. Attributes Selected with SVM-RFE Feature Selection

Merit Index	Attribute
1	Average Width of R' wave (V1)
2	Amplitude of T wave (AVR)
3	Heart Rate
4	Average Width of R' wave (V2)
5	Amplitude of T wave (V5)
6	Amplitude of R' wave (AVR)
7	Existence of diphasic derivation of R wave (V1)
8	Number of intrinsic deflections(V1)
9	Amplitude of JJ wave (V5)

The reduced dataset that was created with SVM-RFE has just two attributes in common with the dataset created with DT. These are Heart Rate and Amplitude of T wave (V5). Figure 6.14 shows the correctly classified instances and Table 6.29 shows the total correctly classified instances for all algorithms with the weighted f-score values.

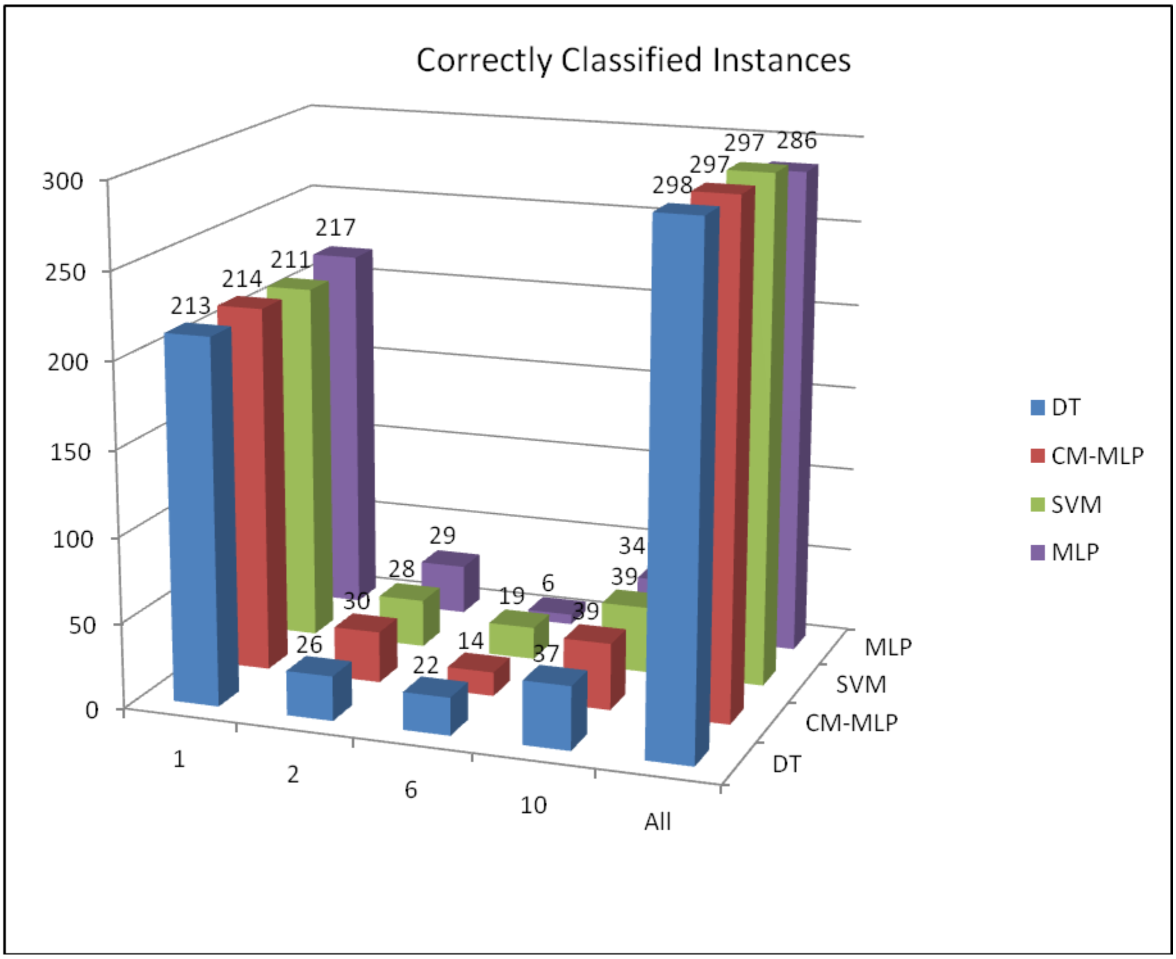


Figure 6.14. Correctly Classified Instances for Reduced SVM-RFE Dataset

Table 6.29. Performance of all Algorithms for SVM-RFE Feature Selection

Method	Correctly Classified Instances	%	Weighted f-score
DT	298 ±2.72	83.24	0.8313
SVM	297 ±1.10	82.96	0.8401
MLP	286 ±2.25	79.89	0.7876
CM-MLP	297 ±1.19	82.96	0.8167

CM-MLP has very close overall accuracy to DT and SVM and all of the algorithms have almost equal number of correctly classified instances. MLP has some problems associated with the prediction of Sinus Bradycardy (Classes 6) arrhythmias like the DT reduced dataset. CM-MLP was also affected on Class 6 arrhythmia, but due to its modular nature it may recover easily. In all other arrhythmia types, CM-MLP has better prediction accuracy than DT. Table 6.30, 6.31, 6.32 and 6.33 show the detailed accuracy parameters for all classes.

Table 6.30. Detailed Accuracy Parameters for DT with SVM-RFE F. Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	213	91	28	26	0.89±0.006	0.76±0.019	0.88±0.008	0.88±0.006
Class 2	26	298	16	18	0.59±0.021	0.94±0.009	0.61±0.015	0.60±0.016
Class 6	22	327	6	3	0.88±0.037	0.98±0.007	0.78±0.013	0.83±0.019
Class 10	37	298	10	13	0.74±0.021	0.96±0.007	0.78±0.015	0.76±0.017

Table 6.31. Detailed Accuracy Parameters for SVM with SVM-RFE F. Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	211	92	27	28	0.88±0.003	0.77±0.007	0.88±0.003	0.88±0.002
Class 2	28	303	11	16	0.63±0.012	0.96±0.003	0.71±0.007	0.67±0.007
Class 6	19	327	6	6	0.76±0.019	0.98±0.003	0.76±0.006	0.76±0.009
Class 10	39	301	7	11	0.78±0.008	0.97±0.003	0.84±0.006	0.81±0.006

Table 6.32. Detailed Accuracy Parameters for MLP with SVM-RFE F. Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	217	83	36	22	0.90±0.005	0.69±0.017	0.85±0.006	0.88±0.005
Class 2	29	301	13	15	0.65±0.016	0.95±0.005	0.69±0.011	0.67±0.013
Class 6	6	331	2	19	0.24±0.026	0.99±0.006	0.75±0.012	0.36±0.017
Class 10	34	287	21	16	0.68±0.013	0.93±0.007	0.61±0.010	0.64±0.010

Table 6.33. Detailed Accuracy Parameters for CM-MLP with SVM-RFE F. Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	214	77	42	25	0.89±0.002	0.64±0.011	0.83±0.005	0.86±0.002
Class 2	30	304	10	14	0.68±0.013	0.96±0.003	0.75±0.007	0.71±0.008
Class 6	14	325	8	11	0.56±0.019	0.97±0.004	0.63±0.006	0.59±0.009
Class 10	39	298	10	11	0.78±0.020	0.96±0.003	0.79±0.007	0.78±0.010

All of the accuracy metrics of SVM, DT and CM-MLP are similar to each other. It can be observed from Tables 6.30 to Table 6.33 that, MLP has lowest prediction accuracy by 0.24 sensitivity and 0.3636 f-score value for Sinus Bradycardy. The rest of accuracy values are close to that of other algorithms. CM-MLP is also in the two highest sensitivity lists for Class 1, 2 and 10. Table 6.34 shows the highest two sensitivities for all classes.

Table 6.34. Highest Two Sensitivity for overall performance for SVM-RFE

Class	Highest Sensitivity	Methods
1	0.90-0.89	MLP - CM-MLP/DT
2	0.68-0.66	CM-MLP - MLP
6	0.88-0.76	DT - SVM
10	0.78-0.78	SVM - CM-MLP

6.6. Local Dataset Results

Local dataset is smaller and more balanced when compared with UCI dataset. Additionally, only binary classification methods may be applied to it. CM-MLP was not applied to local dataset since class-modularity is applicable only when more than 2 classes are available. DT, SVM and MLP were applied to local dataset and accuracy parameters are given in Table 6.35.

Table 6.35. Detailed Accuracy Parameters for Local Dataset

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
MLP	16	25	4	7	0.69±0.024	0.86±0.023	0.80±0.031	0.74±0.025
SVM	10	14	15	13	0.43±0.032	0.48±0.023	0.40±0.024	0.41±0.027
DT	11	24	5	12	0.47±0.043	0.82±0.052	0.68±0.064	0.56±0.034

MLP has the highest prediction rate for both normal and abnormal ECG. The main performance difference between SVM and DT exists in the prediction accuracy of negative values. Specificity, which shows the accuracy on normal records, shows that DT was able to detect almost 2 times more instances than SVM. Figure 6.15, 6.16 and 6.17 shows the ROC curves for all algorithms.

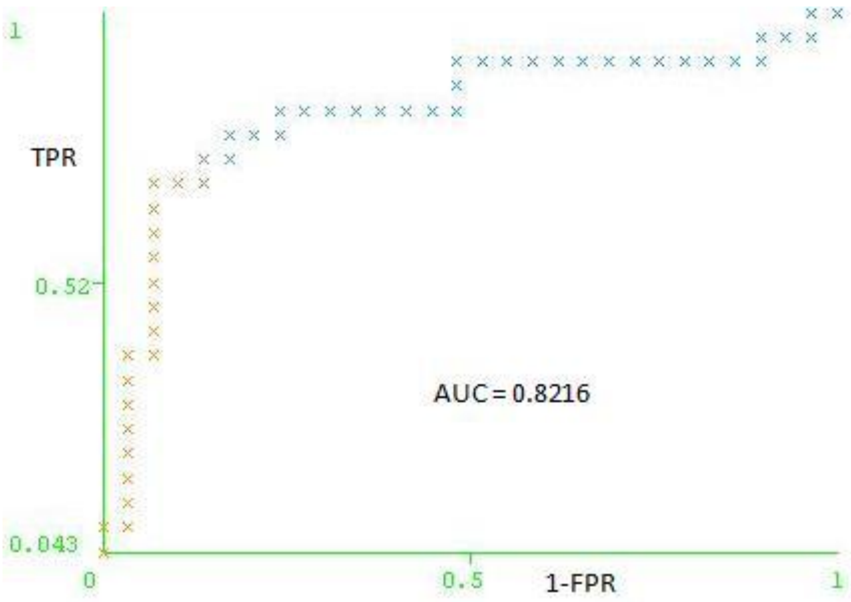


Figure 6.15. MLP ROC Curve for Local Dataset

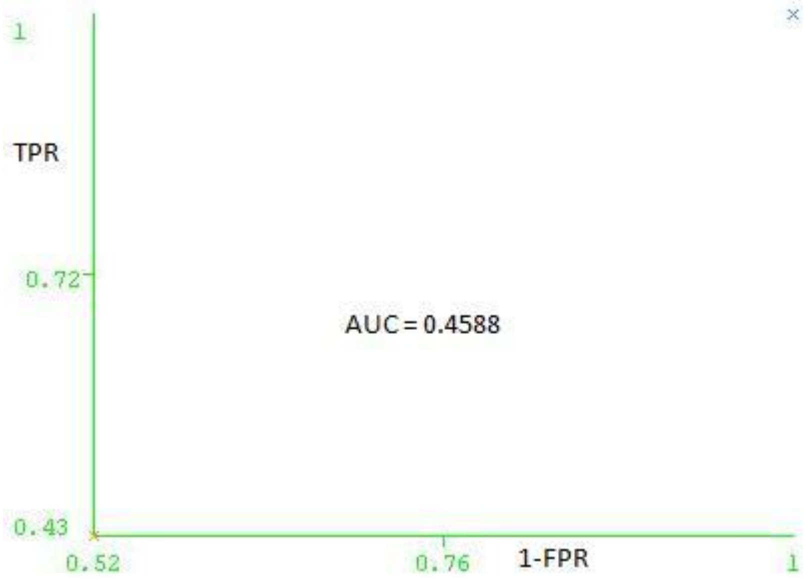


Figure 6.16. SVM ROC Curve for Local Dataset

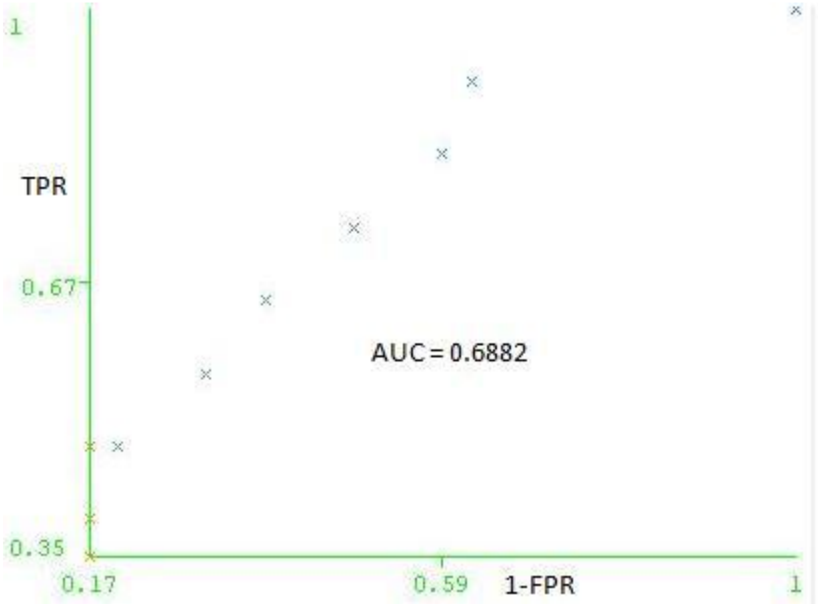


Figure 6.17. DT ROC Curve for Local Dataset

6.7. 10-fold Paired t-test

K-fold paired t-tests are commonly used for statistical significance of learning algorithms [21]. Let p^1 denote the error of the first classifier, and p^2 denote the error of the second classifier on the k-folds. Then, μ_1 is the average error of classifier 1, and μ_2 is the average error of the second classifier. This test is used to test the following hypotheses one versus another:

$$H_0 : \mu_1 - \mu_2 = 0 \tag{6.5}$$

$$H_1 : \mu_1 - \mu_2 \neq 0 \tag{6.6}$$

Let $p_i = p^1 - p^2$ be the difference of errors on folds. Under the null hypotheses, paired differences are t distributed with $k - 1$ degrees of freedom. We can calculate the estimates of the mean and the variance:

$$m = \frac{\sum_{i=1}^{k} p_i}{k} \tag{6.7}$$

$$S^2 = \frac{\sum_{i=1}^{k}(p_i - m)^2}{k-1} \tag{6.8}$$

We then calculate the t-statitics as

$$t' = \frac{m\sqrt{k}}{S} \tag{6.9}$$

If $t' \in (-t_{\alpha/2,k-1}, t_{\alpha/2,k-1})$, then the test accepts the hypothesis, else the test rejects. This is the two-sided test. When we check for statistical improvement, we use the one-sided version. In this case, the test accepts if $t' \in (-\infty, t_{\alpha,k-1})$.

T-tests with 95% confidence interval were used to measure the statistical significance of the 10-fold runs. The 10-fold paired t-test results for CM-MLP, MLP, SVM and DT are shown in Table 6.36. The tests are performed for 278 features dataset and datasets created with PCA, DT and SVM-RFE. Total TP numbers are used to give the accuracy evaluated on t-test. The statistically significant cases in which CM-MLP outperforms other algorithms are highlighted.

Table 6.36. Pairwise Comparison of Accuracies with t-test

	CM-MLP	*MLP*	*SVM*	*DT*
CM-MLP	0	4	1	2
MLP	0	0	0	1
SVM	1	4	0	2
DT	2	3	2	0

7. CONCLUSION

In this study arrhythmia prediction using ECG data was focused. Early detection of arrhythmias became more important as the effort for early diagnosis of heart related problems increase. A class-modular scheme was built and applied to multiple datasets with different size and attributes. In addition to that real-life ECG data was collected and used for comparison with previous datasets.

Initial stage in that study was examining the UCI Dataset and transforming it to a simpler and robust database for machine learning. Dimensionality reduction was the main technique used to achieve this aim. A feature extraction method (PCA) and two feature selection methods (DT and SVM-RFE) applied to UCI dataset. The resulting datasets show similar characteristics but has two attributes in common. Attributes meritful for both methods are not available. Another method of dimensionality reduction was achieved by class-dependent feature subset selection based on RELIEF algorithm. These reduced subsets were used as inputs to class-modular MLP modules.

Defining the prediction accuracy of new developed Class-Modular MLP was main target of this study. All runs on UCI and reduced dataset showed performances of MLP and Class-Modular MLP. In general CM-MLP performed better than MLP for all features and reduced dataset. The strength of CM-MLP originates from increasing sensitivity and f-score, which were also main design goals for modular algorithms. Especially in datasets created with feature selection, CM-MLP provided a minimum TP Rate on each class and has better sensitivity in average. This feature makes CM-MLP an ideal learning method when a minimum TP Rate is desired on each class. Imbalanced datasets may create bias on prediction rate. CM-MLP may be appropriate for such datasets.

The DT and SVM method were presented to compare them with MLP and CM-MLP. SVM’s overall performance was above MLP. To achieve higher accuracy with MLP, increasing hidden nodes technique is applied. The results show that adding hidden nodes does not increase sensitivity and may lead to overtraining on some datasets. The increase in complexity also increases the time to build training model. CM-MLP, like MLP, does not perform better with increasing complexity on UCI dataset. This is tested with increasing input feature size and increasing number of nodes.

When CM- MLP is compared with MLP, it provides both advantages of modularity and shows similar prediction rate with MLP. The average sensitivity observed for each class, makes CM-MLP an ideal arrhythmia classification method that may be used as an alternative to MLP.

APPENDIX A: CARDIAC ARRHYTHMIA DATABASE

Title: Cardiac Arrhythmia Database

Original owners of 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

Donor: H. Altay Guvenir

Date: January, 1998

Past Usage:

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.

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.

Number of Instances: 452

Number of Attributes: 279

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

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

Class Distribution:

Database: Arrhythmia

Table A.1. Class Distribution of Arrhythmia

Code	Class	# of Instances
1	Normal	245
2	Ischemic Changes (Coronary Artery Disease)	44
3	Old Anterior Myocardial Infarction	15
4	Old Inferior Myocardial Infarction	15
5	Sinus Tachycardy	13
6	Sinus Bradycardy	25
7	Ventricular Premature Contraction (PVC)	3
8	Supraventricular Premature Contraction	2
9	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: PCA EIGENVALUES

D	E	D	E	D	E	D	E	D	E
1	23.048	21	3.013	41	1.675	61	0.996	81	0.591
2	16.421	22	2.918	42	1.624	62	0.974	82	0.571
3	13.306	23	2.733	43	1.576	63	0.945	83	0.566
4	11.789	24	2.615	44	1.550	64	0.925	84	0.529
5	9.732	25	2.596	45	1.537	65	0.908	85	0.524
6	7.348	26	2.562	46	1.470	66	0.871	86	0.513
7	6.992	27	2.480	47	1.441	67	0.850	87	0.504
8	6.619	28	2.410	48	1.373	68	0.830	88	0.488
9	5.814	29	2.344	49	1.347	69	0.816	89	0.473
10	5.316	30	2.310	50	1.338	70	0.807	90	0.452
11	4.974	31	2.136	51	1.279	71	0.789	91	0.446
12	4.833	32	2.075	52	1.236	72	0.740	92	0.433
13	4.613	33	2.032	53	1.208	73	0.732	93	0.415
14	4.326	34	1.991	54	1.177	74	0.727	94	0.407
15	3.995	35	1.960	55	1.140	75	0.690		
16	3.627	36	1.903	56	1.110	76	0.685		
17	3.427	37	1.893	57	1.094	77	0.667		
18	3.380	38	1.846	58	1.058	78	0.650		
19	3.324	39	1.775	59	1.052	79	0.608		
20	3.123	40	1.753	60	1.030	80	0.597		

APPENDIX C: CLASS-MODULAR RUN RESULTS

Table C.1. 278 Features with Subset Size 5

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	227	65	54	12	0.94±0.005	0.54±0.014	0.80±0.009	0.87±0.006
Class 2	30	303	11	14	0.68±0.024	0.96±0.003	0.73±0.014	0.70±0.011
Class 6	14	328	5	11	0.56±0.011	0.98±0.008	0.73±0.006	0.63±0.014
Class 10	41	297	11	9	0.82±0.020	0.96±0.008	0.78±0.006	0.80±0.009

Table C.2. 278 Features with Subset Size 10

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	204	69	50	35	0.85±0.004	0.57±0.018	0.80±0.008	0.82±0.005
Class 2	30	301	13	14	0.68±0.027	0.95±0.007	0.69±0.011	0.68±0.015
Class 6	15	326	7	10	0.60±0.019	0.97±0.007	0.68±0.008	0.63±0.011
Class 10	40	299	9	10	0.80±0.025	0.97±0.004	0.81±0.011	0.80±0.015

Table C.3. 278 Features with Subset Size 15

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	209	70	49	30	0.87±0.002	0.58±0.015	0.81±0.008	0.84±0.003
Class 2	33	297	17	11	0.75±0.021	0.94±0.006	0.66±0.008	0.70±0.014
Class 6	16	323	10	9	0.64±0.040	0.97±0.006	0.61±0.018	0.62±0.022
Class 10	30	298	20	10	0.75±0.024	0.93±0.006	0.60±0.008	0.66±0.014

Table C.4. 278 Features with Subset Size 20

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	208	77	42	31	0.87±0.002	0.64±0.018	0.83±0.007	0.85±0.003
Class 2	29	302	12	15	0.65±0.026	0.96±0.005	0.70±0.007	0.68±0.011
Class 6	14	323	10	11	0.56±0.062	0.97±0.004	0.58±0.0015	0.57±0.024
Class 10	33	298	10	17	0.66±0.026	0.96±0.004	0.76±0.009	0.70±0.013

Table C.5. PCA Feature Extraction with Subset Size 10 for each module

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	201	52	67	38	0.84±0.006	0.43±0.035	0.75±0.012	0.79±0.009
Class 2	25	297	17	19	0.56±0.045	0.94±0.010	0.59±0.018	0.58±0.025
Class 6	3	324	9	22	0.12±0.046	0.97±0.012	0.25±0.014	0.16±0.021
Class 10	25	296	12	25	0.50±0.029	0.96±0.011	0.67±0.017	0.57±0.022

Table C.6. DT Feature Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	222	81	38	17	*0.92±0.013*	*0.68±0.023*	*0.85±0.006*	*0.88±0.007*
Class 2	27	299	15	17	*0.61±0.033*	*0.95±0.008*	*0.64±0.014*	*0.62±0.019*
Class 6	20	327	6	5	*0.80±0.013*	*0.98±0.009*	*0.76±0.005*	*0.78±0.007*
Class 10	26	301	7	24	*0.52±0.030*	*0.97±0.012*	*0.78±0.015*	*0.62±0.019*

Table C.7. SVM-RFE Feature Selection

	TP	TN	FP	FN	Sensitivity	Specificity	Precision	F-score
Class 1	214	77	42	25	*0.89±0.002*	*0.64±0.011*	*0.83±0.005*	*0.86±0.002*
Class 2	30	304	10	14	*0.68±0.013*	*0.96±0.003*	*0.75±0.007*	*0.71±0.008*
Class 6	14	325	8	11	*0.56±0.019*	*0.97±0.004*	*0.63±0.006*	*0.59±0.009*
Class 10	39	298	10	11	*0.78±0.020*	*0.96±0.003*	*0.79±0.007*	*0.78±0.010*

REFERENCES

1. Hussain, W. and W. Ishak, *The Potential of Neural Networks in Medical Applications*, http://www.generation5.org/content/2004/NNAppMed.asp, 2004.

2. Veropoulos, K., N. Cristianini and C. Campbell, “The Applications of Support Vector Machines to Medical Decision Support: A Case Study”, *ACAI99*, Chania, 1999.

3. Gao, D., M. Madden, M. Schukat, D. Chambers and G. Lyons, “ANN Based Diagnostic System for Arrhythmia with ECG Signals”, *Proc. 23rd IASTED International Multi-Conference Artificial Intelligence and Applications*, Innsbruck, 2005.

4. Acharya, U. R., P. S. Bhat, S. S. Iyengar, A. Rao and S. Dua, “Classification of Heart Rate Data Using Artificial Neural Network and Fuzzy Equivalence Relation”, *Pattern Recognition*, Vol. 36, pp. 61-68, 2003.

5. Zengin, Z., *Risk Estimation for Intrauterine Growth Restriction Using Ultrasound Indices and Classifiers in Emergency Cases*, M.S. Thesis, Boğaziçi University, 2008.

6. Vargas, F., M. Castro, M. Macarthy and D. Lettnin, "Electrocardiogram Pattern Recognition by Means of MLP Network and PCA: A Case Study on Equal Amount of Input Signal Types", VII *Brazilian Symposium on Neural Networks (SBRN'02)*, 2002.

7. 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.

8. Asuncion, A., and D.J. Newman, *UCI Machine Learning Repository*, http://www.ics.uci.edu/~mlearn/MLRepository.html, 2007.

9. Harvard-MIT Division of Health Sciences and Technology, *MIT-BIH Arrhythmia Database Hypertext Edition*, http://www.physionet.org/physiobank/database/mitdb/, 24 May 1997.

10. American Heart Asscociation Webpage, *Cardiovascular Disease Statistics*, http://www.americanheart.org/presenter.jhtml?identifier=4478, 2010.

11. Schmidt, A. and Z. Bandar, "Modularity - a Concept for new Neural Network Architectures", *IASTED International Conference on Computer Systems and Application*, CSA, 1998.

12. Avila S., L. Matos, C. Freitas, J.M. Carvalho, "Evaluating a Zoning Mechanism and Class-Modular Architecture for Handwritten Characters Recognition", *Progress in Pattern Recognition, Image Analysis and Applications*, Vol. 4756, pp. 515-524, 2008.

13. Kapp, M.N., C.O.D.A. Freitas, J.C. Nievola, R. Sabourin, "Evaluating the Conventional and Class-Modular Architectures Feedforward Neural Network for Handwritten Word Recognition", *16th Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2003)*, pp. 315-319, 2003.

14. Oh, I.S. and Suen, C.Y., "A Class-Modular Feedforward Neural Network for Handwriting Recognition", *Pattern Recognition*, Vol. 35, pp. 229-244, 2002.

15. ECG Learning Center, http://library.med.utah.edu/kw/ecg/index.html, 2006.

16. Wikipedia, Free Encyclopedia,
http://en.wikipedia.org/wiki/Electrocardiography#cite_note-1, 2010.

17. Hampton, John R., *Pratik ECG*, İstanbul Medikal Yayıncılık, 2007.

18. Guvenir, H.A., B. Acar, G. Demiroz and A. Cekin, "A Supervised Machine Learning Algorithm for Arrhythmia Analysis", *Proceedings of the Computers in Cardiology Conference*, Lund, Vol. 24, pp. 433-436, 1997.

19. Özkaya, A.U., *Intelligent Arrhythmia Classification Based on Support Vector Machines*, M.S. Thesis, Boğaziçi University, 2006.

20. Maimon, O. and L. Rokach, *The Data Mining and Knowledge Discovery Handbook*, Springer, 2005.

21. Alpaydın, E., *Introduction to Machine Learning*, MIT Press, 2005.

22. Yu, L. and H. Liu, "Feature Selection for High Dimensional Data: A Fast Correlation-Based Filter Solution", *20th International Conference on Machine Learning*, Washington, 2003.

23. Vidal, R., Y. Ma and S. Sastry, "Generalized Principal Component Analysis (GPCA)", *IEEE Computer Society Conference*, 2003.

24. Derelioğlu, G., *A Modular Approach for SMEs Credit Risk Analysis*, M.S. Thesis, Boğaziçi University, 2009.

25. Kotsiantis, S.B. "Supervised Machine Learning: A Review of Classification Techniques", *Informatica*, Vol. 31, pp. 249-268, 2007.

26. Chen, X. and J. C. Jeong, "Enhanced Recursive Feature Elimination", *IEEE Sixth International Conference on Machine Learning and Applications*, 2007.

27. Guyon, I., J. Weston, S. Barnhill and V. Vapnik, "Gene selection for cancer classification using support vector machines", *Machine Learning*, Vol.46, pp. 389–422, 2002.

28. Furlanello, C., M. Serafini, S. Merler and G. Jurman, "Gene Selection and Classification by Entropy-based Recursive Feature Elimination", *Proceedings of the International Joint Conference on Neural Networks*, 2003.

29. Thang, Y., Y. Zhang and Z. Huang, “Development of Two-Stage SVM-RFE Gene Selection Strategy for Microarray Expression Data Analysis*”, IEEE/ACM Transactions On Computational Biology and Bioinformatics*, Vol. 4 Issue 3, pp. 365-381, 2007.

30. Kira, K.and L.A. Rendell, “The feature selection problem: traditional methods and a new algorithm”, *Proceedings Tenth National Conference on Artificial Intelligence (AAAI-92)*, pp.129-134, 1992.

31. Tian, J., Y. He, X. Yang, L. Li and X. Chen, “Improving Fingerprint Recognition Performance Based on Feature Fusion and Adaptive Registration Pattern”, *Advances in Biometric Person Authentication*, Vol. 3338, pp.131-164, 2005.

32. Nina, Z. and L. Wang, “Class-Dependent Feature Selection for Face Recognition”, *Advances in Neuro-Information Processing*, Vol. 5507, pp. 551-558, 2009.

33. Travis, W., *Neural Networks Tutorial*,
http://www.cs.usyd.edu.au/~irena/ai01/nn/travtute.htm, 2010.

34. Daqi, G., L. Chunxia and Y. Yunfan, “Task decomposition and modular single-hidden-layer perceptron classifiers for multi-class learning problems”, *Pattern Recognition*, Vol. 40, No 8, pp. 2226-2236, 2007.

35. Ayala, E., M. Lopez and P. Melin, “Modular Neural Network with Fuzzy Integration of Responses for Face Recognition”, *Evolutionary Design of Intelligent Systems in Modeling, Simulation and Control*, Vol. 257, pp. 131–158, 2009.

36. Silva, P.H.F. and A.L.P.S. Campos, “Fast and accurate modelling of frequencyselective surfaces using a new modular neural network configuration of multilayer perceptrons”, *Microwaves, Antennas & Propagation*, Vol. 2, Issue 5, pp. 503 – 511, 2008.

37. Schmidt, A., *A Modular Neural Network Architecture with Additional Generalization Abilities for High Dimensional Input Vectors*, M.S.Thesis, Manchester Metropolitan University, 1996.

38. Hamilton, H., E. Gurak, L. Findlater and W. Olve, *Overview of Decision Trees*, http://www.cs.uregina.ca/~dbd/cs831/notes/ml/dtrees/4_dtrees1.html, 2001.

39. Barakat, N., “Rule-Extraction from Support Vector Machines for Medical Diagnosis – Prediction and Explanation”, *ITEE Seminar*, The University of Queensland, Australia, 2005.

40. Burges, C.J.C., “A Tutorial on Support Vector Machines for Pattern Recognition”, *Data Mining and Knowledge Discovery*, Vol. 2, pp. 121-167, 1998.

41. Chih-Wei, H., C. Chih-Chung and L. Chih-Jen, *A Practical Guide to Support Vector Classification*, http://www.csie.ntu.edu.tw/~cjlin/, 2009.

42. Hall M., E. Frank, G. Holmes, B. Pfahringer, P. Reutemann and I. H. Witten, “The WEKA Data Mining Software: An Update”, SIGKDD Explorations, Vol. 11, Issue 1, 2009.

43. Chang, C.-C and C-J. Lin, *LIBSVM: A Library for Support Vector Machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm, 2001.

44. Wikipedia, Free Encyclopedia, http://en.wikipedia.org/wiki/Receiver_operating_characteristic, 2010.

45. Özcan, N. Ö., *A Fuzzy Support Vector Machine Approach for ECG Analysis*, M.S. Thesis, Boğaziçi University, 2010.

46. Beitzel, S.M., *On Understanding and Classfying Web Queries*, Ph.D. Thesis, Illinois Institute of Technology, 2006.

47. Sokolova, M., N. Japkowicz and S. Szpakowicz,,“Beyond Accuracy, F-Score and ROC: A Family of Discriminant Measures for Performance Evaluation”, *AI 2006: Advances in Artificial Intelligence*, Vol. 4304, pp.1015-1021, 2006.

48. Ghanem, A.S., S. Venkatesh and G.West, “Learning in Imbalanced Relational Data”, *19th International Conference on Pattern Recognition (ICPR 2008),* 2008.

49. Mazurowski, Maciej A. , P. A. Habas, J. M. Zurada, J. Y. Lo, J. A. Baker and G. D. Tourassi, “Training Neural Network Classifiers for Medical Decision Making: The Effects of Imbalanced Datasets on Classification Performance”, *Neural Networks,* Vol. 21, Issues 2-3, pp. 427-436, 2008.