IMPLEMENTATION OF A MEDICAL DECISION MAKING TOOL

A Paper
Submitted to the Graduate Faculty
of the
North Dakota State University
of Agriculture and Applied Science

By

Sudheer Gadiparthi

In Partial Fulfillment of the Requirements
for the Degree of
MASTER OF SCIENCE

Major Department:
Computer Science

March 2014

Fargo, North Dakota

North Dakota State University

Graduate School

Title

Implementation of a Medical Decision Making tool

By

Sudheer Gadiparthi

The Supervisory Committee certifies that this ***disquisition*** compiles with North Dakota State University's regulations and meets the accepted standards for the degree of

MASTER OF SCIENCE

SUPERVISORY COMMITTEE:

Dr. Simone Ludwig

Advisor

Dr. Gursimran Walia

Dr. Azer Akhmadov

Approved by Department Chair:

03/28/2014

Date

Dr. Brian M. Slator

Signature

ABSTRACT

Nowadays timely medical assistance is necessary for physicians/medical practitioners for decision making. Sometimes it is necessary for physicians to make decisions to diagnose the disease of a patient promptly. 'Implementation of a Medical Decision Making Tool', is a software tool was developed by implementing and incorporating efficient data mining techniques. In this project, with the help of WEKA software, which is a collection of classification algorithms for data analysis and feature selection, the decision making tool was implemented for the data preprocessing and classification on collected data sets of patients. Dataset for three kinds of diseases, heart disease, dermatology, and hepatitis were used to evaluate the performance of eight well-known classification algorithms. The area under receiver operating characteristic (ROC) and the accuracy were calculated for the selected algorithms and the best suitable algorithm for each disease was discussed.

ACKNOWLEDGEMENTS

I would take this opportunity to thank my advisor, Dr. Simone Ludwig, who has given me valuable support, encouragement and advice without which this work would not have been completed. I am thankful to the members of the committee, Dr. Gursimran Walia and Dr. Azer Akhmadov for their time and support. I would also like to thank my parents, all my teachers in school, lecturers, professors in college, Dr. Nygard and my friends for their valuable support and constant encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1. INTRODUCTION

Timely decision-making is a very important aspect in the medical field. For the process of diagnosing a patient many factors have to be taken into account, which sometimes is a lengthy and tedious process. It is very important to administer timely medical assistance in hospitals.

Prediction of patient health condition is crucial to manage clinical resource utilization. It is very important to consider all the required cases about the health condition of a patient before making a diagnosis of the disease and selecting an appropriate treatment. One should be extremely cautious when a patient's health condition is very critical. Usually physicians judge diagnosis by assessing the current test results of a patient and also with reference of previous judgments made on other patients with similar kind of disease. Thus, it depends on the physician knowledge, which may be problematic sometimes because there are large numbers of factors that a physician has to evaluate before a diagnosis of the disease is possible. When evaluating the different factors, sometimes it would take more time than estimated and the costs for the tests would increase based on the number of the tests to be taken to diagnose the disease. This will sometimes be a burden for patients. In the cases where treatment has to be given in less time, the situation may become critical. In this paper 'Implementation of a Medical Decision Making tool' a tool was developed for physicians or doctors to support them in making a prognosis regarding current health condition of patient based upon their disease symptoms. This project proposes a medical decision making tool that helps the physician in determining the disease accurately and quickly. This also saves time and money for the patient.

Classification is a powerful data mining concept, which is used to train an algorithm with known input and output values in order to create a model to predict the class of data with unknown values. The trained classification algorithm results are analyzed to see the reliability and accuracy of the algorithm predictions.

By definition “Data mining is the process of discovering meaningful and actionable patterns hidden in large amounts of data” [2]. The main goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use [11]. In data mining, decision trees and neural networks are two important algorithms used in various domains in solving practical problems related to classification, prediction and diagnosis.

Data mining has been continuously used in different areas like game design, business strategies, medical sector, etc. The performance of decision tree and neural network algorithms has been evaluated and proved to be efficient in many cases [6].

Data mining has become one of the fundamental methodologies applied in the medical field for the prediction/prognosis of a disease. Specifically it is used in healthcare organizations, health informatics, patient care for information extraction and automatic identification of unknown classes. Many algorithms associated with data mining have helped in medical decision making by distinguishing irrelevant data from normal data. Different classification and clustering algorithms help in identification of hidden complex relationships between diagnostic features of different patient groups.

Since data mining and specifically classification algorithms are being successfully used in various medical domains to study complex diseases, the prediction of a medical condition is applied in this project to provide decision assistance for physicians [2]. The goal of this project is to build a software tool that allows making predictions based on data that is provided. Eight different classifiers were implemented. However, since decision-making is highly critical in medical domains, classifiers that result in higher decision confidence are preferred. To be able to evaluate such confidence in different classifiers, we propose a measurement procedure and compare the accuracy and area under receiver operating characteristic (ROC) curve (AUC) measure [9].

The remainder of this paper is structured as follows. In the second chapter, literature review, explains about data mining, classification and the different classification techniques used in this project. The third chapter explains about the different data sets for three specific diseases, feature selection, which helps to remove the irrelevant attributes from the dataset and also describes each classification technique in detail. In the fourth chapter, implementation details of the tool are explained. In the fifth chapter, experimental design and results of each combination with the selected dataset, filter, and classification algorithm are given listing the best classifier for each disease. In the sixth chapter, conclusion and future work are discussed.

2. LITERATURE REVIEW

In the process of developing this project many sources have been referred pertaining to data mining, classification, WEKA, data set, decision trees, data preprocessing, etc. In any industry or practical problem solving, while discovering new concept/methods it is common to consider the old patterns already developed in that area to further enhance the concept. The present and future of research in the medical field related to decision-making is becoming data-driven [1]. Numeric data is becoming freely available in large amount and there is a need for new data analysis tools and techniques. Data mining is one of the new and emerging areas of computational intelligence that offers new theories and uses various pattern recognition techniques, artificial intelligence and analysis of large datasets [2]. The basis of the methodologies of data mining is its ability to find patterns and relationships within large amounts of data [3]. These patterns and relationships helps in the construction of models having available training data and assigning the class label to unlabeled cases for the unknown or new data.

Data mining techniques have been successfully applied in a variety of forecasting procedures and were used to find unknown results or hidden patterns. By identifying hidden patterns, data mining can get information that allows a new perspective on certain diseases and to find knowledge that can foster more research in several areas of medicine thus enabling physician to accurately cure disease. In [5], the author mentioned that the high degree of accuracy of already developed models is a good example of data mining's contribution to medicine.

In many areas of medicine, data mining has been proven to be added value by contributing with new discoveries and improving the results obtained with other methodologies [5]. There are different data mining techniques, some of the popular and important techniques are association, classification, clustering, sequential processing, decision trees, neural networks, etc.

Classification is one of the important data mining concepts, which consists of predicting new data output value or class label. The goal of classification is to accurately predict the target class for each case in a data set [6]. Data mining researchers use classification to predict problems related to know an unknown value. In order to predict the new data output value with attributes and data, classification algorithms analyses and processes the given training dataset, which contains same set of attributes and associated data with output values. Some of the general classification algorithms are decision trees, nearest neighbors, rule induction, fuzzy rule induction, neural networks, etc. In general, if you already have a set of predefined classes and want to predict which class a new data belongs to using classification may yield better results.

A decision tree is a predictive model, which uses a binary tree like structure to predict the output values. It takes the given data set as input and forms a binary tree like structure using set of decision rules from root node to leaf node and based on the decisions at each node, predicts the output of unknown data [7]. Some of the well-known decision tree algorithms are J48, Naïve Bayes, Random tree, REP tree, AD tree, etc.

A neural network is a biologically inspired mathematical model, which is also called as a parallel distributed processing network. It is an adaptive system, which is interconnected with all the nodes that flows through network and produces the output. Some of the neural networks are multi-layer perceptron (MLP), radial basis function (RBF) network, a hybrid genetic algorithm neural network (GANN).

According to the project requirements and based on the papers reviewed, classification techniques will be expected to give the best predicted results to determine the patient disease.

Waikato Environment for Knowledge Analysis (WEKA) software was chosen mainly because of its characteristics like free availability and ease of use. WEKA is a machine learning software with different classification algorithms and it is written in java programing language. Since it is fully implemented in java programming language, it is portable in most of the modern computing platforms. In this project, WEKA developed methods were used to train and test the application [9].

Compared with the studies identified in the literature it is expected that data mining classification techniques could induce predictions with greater accuracy compared to known traditional methods. An analysis of prediction methods indicates that automatically generated diagnostic rules outperform the diagnostic accuracy of physicians [2].

3. DATASETS, FEATURE SELECTION AND CLASSIFICATION

In the process of prediction, the accuracy of the predicted results in data mining depends mainly on how well the classifier is being trained [8]. A data set is a collection of data. Feature selection is selecting the relevant features from the data set. Classification is finding the unknown target value based on the known target values in the data set. A detailed explanation of data, dataset, feature selection and classification is discussed in this chapter.

The training of the classifier is done mainly based on the selection of classification algorithm and the data sets, which are given as input to the classifier. Some of the data in the data sets may not be useful for the prediction, which if eliminated would reduce the burden on the classification algorithms. This can be achieved with the help of feature selection, which is also called as filtering. This process of filtering the data helps to obtain better features to be selected among many features. For example, before the feature selection process, if there are features like id, age, gender, blood group, blood pressure for a patient in the diabetic data set, after the filtering process the irrelevant features like ‘id’ are removed and the necessary features are selected. Even though when we use filters there may be some unnecessary features and data left, which makes classification techniques giving a non-satisfactory results. With the use of only decision trees it would be highly unreliable to depend on the predicted results. So by making decision trees as filtering technique and neural networks as the classifier, better predictions can be expected. Figure 1 below shows the flow of these components in the process of training the classifier.

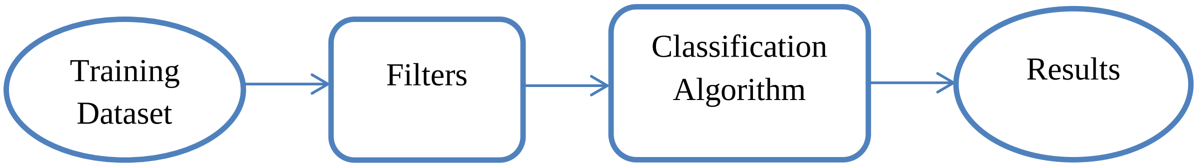


Figure 1: Flow of medical decision making tool

3.1. Datasets

A dataset is a collection of data that is related to one category. All the datasets for heart disease, dermatology and hepatitis were collected from the UCI machine learning repository [10]. Since WEKA takes an input data set in the Attribute Relationship File Format (ARFF) format, all the data sets were converted to the ARFF file format. This format has attributes and instances (data). Two portions of the data for each disease were used in this project, one is for training the classifier and the other is to test the trained classifier. Also the test dataset output is generated in the ARFF format. All the datasets used in this project are discussed in more detail.

ARFF is the text format file used by WEKA to store data. The ARFF file contains two sections, one is the header and the other is the data section. The first line of the header defines the relation name, which is usually the dataset name. Then, there is the list of the attributes. Each attribute is associated with a unique name and a type. The type describes the kind of data contained in the variable and what values it can have. The variables types are: numeric, nominal, etc. The class attribute is by default the last one of the list, but it can be changed and depends on the researcher/user and datasets. Then there is the data, each line stores the attribute of a single entry separated by a comma.

Relational: dataset name (usually).

Attribute Types:

Nominal: One of a predefined list of values; e.g., male, female.

Numeric: A real or integer number.

The following subsection explains in detail about the heart disease, dermatology and hepatitis attributes and the data type of each attribute.

3.1.1. Heart disease

Heart disease is not only related to heart attack but may also include functional problems such as heart-valve abnormalities. These kinds of problems can lead to heart failure. Heart disease is also known as cardiovascular disease (CVD) in medical terms.

This dataset is collected from UCI machine learning repository [10]. In this data set (HDD) 14 attributes and historical data (instances) were used as shown in Table 1 below. The "goal" field refers to the presence of heart disease in the patient. It is integer valued 1(True) and 0 (False).

Table 1: Heart disease attributes

No	Field name	Class Label	Description
1	Age	Real	Age in years
2	Sex	{0,1}	Sex type
			Value 1: Male
			Value 0: Female
3	CP	{1,2,3,4}	Chest pain type
			Value 1: typical angina
			Value 2: atypical angina
			Value 3: non-angina pain
			Value 4: asymptomatic
4	TRESTBPS	Real	Resting blood pressure (in mm HG)
5	Cholesterol	Real	Serum cholesterol (in mg/dl)

Continued

Table 1: Heart disease attributes (Continued)

No	Field name	Class Label	Description
6	FBS	Number	Fasting blood sugar > 120 mg/dl
			Value 1: True
			Value 2: False
7	RESTECG	Number	Resting electrocardiographic results
			Value 0: Normal
			Value 1: ST-T wave abnormality
8	THALACH	Number	Maximum heart rate achieved (beats/minute)
9	EXANG	Number	Exercise induced angina
			Value 1: Yes
			Value 2: No
10	Old peak	Number	ST depression induced by exercise relative to rest
11	Slope	Number	The slope of the peak exercise ST segment
			Value 1: Up sloping
			Value 2: Flat
			Value 3: Down sloping
12	CA	Number	Number of major vessels (0-3) colored by fluoroscopy
13	THAL	Number	Type of defect
			Value 3: Normal
			Value 5: Fixed defect
			Value 7: Reversible defect
14	Pv	Number	Goal field

3.1.2. Dermatology

The differential diagnosis of erythemato-squamous disease is a real problem in dermatology. This dataset is collected from UCI machine learning repository [10] and it contains 34 attributes, 33 of which are linear valued and one of them is nominal. They all share the clinical features of erythema and scaling, with very little differences. In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3.

Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values. Table 2 below explains the attribute information for dermatology diagnosis.

Attribute Information:

Clinical Attributes: (take values 0, 1, 2, 3, unless otherwise indicated)

Table 2: Dermatology attributes

No	Field name	Class Label
1	Scaling	{0,1,2,3}
2	Definite borders	{0,1,2,3}
3	Itching	{0,1,2,3}
4	Koebner phenomenon	{0,1,2,3}
5	Polygonal papules	{0,1,2,3}
6	Follicular papules	{0,1,2,3}
7	Oral mucosal involvement	{0,1,2,3}
8	Knee and elbow involvement	{0,1,2,3}

Continued

Table 2: Dermatology attributes (continued)

No	Field name	Class Label
9	Scalp involvement	{0,1,2,3}
10	Family history, (0 or 1)	{0,1}
11	Melanin incontinence	{0,1,2,3}
12	Eosinophils in the infiltrate	{0,1,2,3}
13	PNL infiltrate	{0,1,2,3}
14	Fibrosis of the papillary dermis	{0,1,2,3}
15	Exocytosis	{0,1,2,3}
16	Aanthosis	{0,1,2,3}
17	Hyperkeratosis	{0,1,2,3}
18	Parakeratosis	{0,1,2,3}
29	Clubbing of the rete ridges	{0,1,2,3}
20	Elongation of the rete ridges	{0,1,2,3}
21	Thinning of the suprapapillary epidermis	{0,1,2,3}
22	Songiform pustule	{0,1,2,3}
23	Munro microabcess	{0,1,2,3}
24	Focal hypergranulosis	{0,1,2,3}
25	Disappearance of the granular layer	{0,1,2,3}
26	Vacuolisation and damage of basal layer	{0,1,2,3}
27	Spongiosis	{0,1,2,3}
28	Saw-tooth appearance of retes	{0,1,2,3}
29	Follicular horn plug	{0,1,2,3}
30	Perifollicular parakeratosis	{0,1,2,3}

Continued

Table 2: Dermatology attributes (continued)

No	Field name	Class Label
31	Inflammatory monoluclear inflitrate	{0,1,2,3}
32	Band-like infiltrate	{0,1,2,3}
33	Age (linear)	Real
34	Erythema	{0,1}

3.1.3. Hepatitis

Inflammation of the liver, burning or swelling of the liver cells refers to hepatitis. When a patient is affected with the hepatitis virus, it affects the liver and causes swelling and redness. Risk factors are blood transfusions, tattoos, etc. [6].

This dataset is collected from UCI machine learning repository. It has 20 attributes and instances are shown in table 3 below.

Table 3: Hepatitis attributes

No	Attribute/ Field name	Type/Example
1	Class	Die, Live
2	Age	10, 20, 30, 40, 50, 60, 70, 80
3	Sex	Male or Female
4	Steroid	Yes or No
5	Antivirals	Yes or No
6	Fatigue	Yes or No
7	Malaise	Yes or No
8	Anorexia	Yes or No

Continued

Table 3: Hepatitis attributes (continued)

No	Attribute/ Field name	Type/Example
9	Liver big	Yes or No
10	Liver firm	Yes or No
11	Spleen Palpable	Yes or No
12	Spiders	Yes or No
13	Ascites	Yes or No
14	Varices	Yes or No
15	Bilirubin	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
16	Alk Phosphate	33, 80, 120, 160, 200, 250
17	SGOT	13, 100, 200, 300, 400, 500
18	Albumin	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
19	Protime	10, 20, 30, 40, 50, 60, 70, 80, and 90
20	Histology	Yes or No

3.2. Feature Selection

Feature selection has become very important in all areas of data mining such as pattern recognition, data mining, statistics, etc. Feature selection is the process of selecting a subset of relevant input variables for use in model construction from large datasets. Most of the times the data in the datasets contain many redundant or irrelevant attributes or features, which are not useful for the model construction. Redundant attributes are those, which provide no more information than the already selected features, and irrelevant features provide no useful information and sometimes make the training data less feasible [11]. By using feature selection, we can reduce the irrelevant and redundant features and hence it takes less time to train the model, which can help to improve the performance of the resulting classifiers. It is known that the machine learning methods themselves will automatically select the most appropriate attributes and delete the irrelevant ones. But in practical cases, the performances of those algorithms are still affected and can be improved by pre - processing. So by using some of the WEKA provided filtering methods to pre - process the data set, and possibly improve the final prediction results. Feature selection techniques are often used in domains where there are many features and comparatively few samples. The coming subsections deals with a few filtering techniques used in this project.

3.2.1. CfsSubsetEval + greedyStepwise search

CfsSubsetEval evaluates the worth of a subset of attributes by considering the individual predictive ability of each feature along with the degree of redundancy between them [9]. Subsets of features, which have high correlation with the class and low inter-correlation are preferred.

Attributes having the highest correlation with the class are iteratively added as long as there is no attribute in the subset that has a higher correlation with the attribute. A missing value is treated as a separate value.

Greedy Stepwise performs a greedy forward or backward search through the space of attribute subsets. The search starts with no or all attributes or from an arbitrary point in the space and stops when the addition/deletion of any remaining attributes results in a decrease in evaluation. By traversing the space from one side to the other it produces a ranked list of attributes and records the order in which the attributes are selected.

3.2.2. Gain ratio + ranker search

Gain Ratio will evaluate the worth of an attribute by measuring the gain ratio with respect to the class [9]. Based on the gain ratio the ranker will rank all the attributes. Irrelevant attributes will be deleted by setting the threshold of the ranker. If a subset evaluator is specified, then a forward selection search is used to generate a ranked list. Subsets of increasing size are evaluated from the ranked list of attributes, i.e. the best attribute, the best attribute plus the next best attribute, etc. The best attribute set is reported. Rank search is linear in the number of attributes if a simple attribute evaluator is used such as GainRatioAttributeEval.

3.3. Classification

Classification is a data mining technique, which is used to predict the unknown values by training one of the classifiers using known values. The concept of using a "training set" is to produce the model. The classifier takes a data set with known output values and uses this data set to build the classification model. Then, whenever there is a new data point with test data, with an unknown output value, the already trained classification model produces the output.

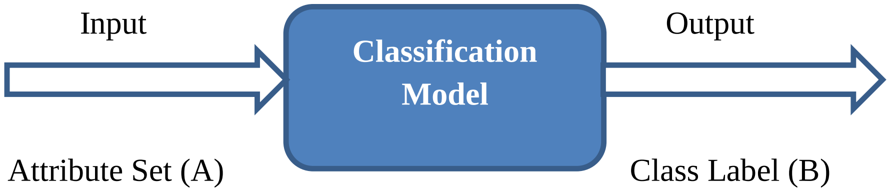


Figure 2: Classification as the task of mapping attribute set (A) into its class label (B)

The following subsections introduce a few classification techniques such as decision trees and neural networks, which are used to build the model for this project. In addition, some decision trees and neural network techniques are discussed.

3.3.1. Decision trees and neural networks

Decision trees and neural networks are two important algorithms used in various domains in solving practical problems related to classification, prediction, diagnosis and many more areas. The performance of these two algorithms has been evaluated and proved to be efficient in lot of circumstances [6]. However, it is also true that the performance will vary from each other based on different datasets. In this project, a combination of a few common decision tree algorithms and neural networks are used to train and predict the desired output.

3.3.2. Decision tree algorithms

A decision tree is a predictive machine-learning model that decides the target value or output value of a new sample based on various attribute values of the available data. The internal nodes of a decision tree denote the different attributes; the branches between the nodes conveys the possible values that these attributes can have in the observed samples, while the terminal nodes or leaf nodes conveys the final value of the dependent variable.

Here is a list of different decision trees that are used to conduct the experiments in this project;

3.3.2.1. J48 Decision Tree (C4.5)

J48 is a java implementation of the C4.5 algorithm in WEKA. This algorithm creates a decision tree based on the attribute values of the available training data in order to classify a new item. During training it identifies the attribute that discriminates the various instances most clearly. This algorithm selects the data instances based on the highest information gain. At each node of the decision tree, C4.5 algorithm chooses the attribute based on the data that most effectively splits the sample into subsets of target classes. Here the splitting category is the information gain.

The algorithm works as, if there is any value for which there is no ambiguity, that is, for which the data instances falling within its category have the same value for the target variable, then it terminate that branch and assign to it the target value that it has obtained. For the other cases, C4.5 algorithm looks for another attribute that gives the highest information gain. It continues in this manner until it gets a clear decision of what combination of attributes gives a particular target value, or it runs out of attributes. In the event that it runs out of attributes, or if it cannot get an unambiguous result from the available information, it assigns this branch a target value that the majority of the items under this branch possess [12].

3.3.2.2. Random Tree

When constructing decision tree, random tree algorithm picks an attribute randomly at each node expansion without any purity function check like information gain [13]. This algorithm does not prune the randomly built decision tree in a conventional way, however, it removes unnecessary nodes. In a decision path, if none of the descendants have different class distribution from this node, then the algorithm treats it as it is an unnecessary node expansion. At that node, the algorithm makes it as the leaf node and removes the expansion. In random tree, classification is always done at leaf node level and each tree outputs a class. The class distribution outputs from multiple trees are averaged as the final class distribution from this node. In the situation like, if the leaf node is empty, it goes one level up and makes parent node as leaf node.

A tree stops growing any deeper if it meets any one of the following conditions:

- When a node becomes empty or there are no more examples to split in the current node.
- When the depth of tree exceeds some limits.

Random tree will generate a tree that considers K randomly chosen attributes at each node.

3.3.2.3. Naïve Bayes Tree

The Naïve Bayes classifier is a probabilistic classifier based on the Bayes rule of conditional probability. It means, the Naive Bayes classifier uses probability to classify the new instance. It makes use of all the attributes contained in the dataset, and analyzes them individually as though they are equally important and independent of each other. The Naïve Bayes classifier considers each of these attributes separately when classifying a new instance. It works under the assumption that the presence or absence of a particular feature of a class is unrelated to the presence or absence of another feature [15]. An advantage of the Naïve Bayes classifier is that it does not require large amounts of data to train the model, because independent variables are assumed; only the variances of the attributes for each class need to be determined, not the entire attributes.

3.3.2.4. REP Tree

The REP (Reduced Error Pruning) tree is a rapid decision tree learning algorithm that builds the tree using information gain and prunes the tree with reduced error pruning. Pruning methods reduce the complexity of tree structure without decreasing the accuracy of the decision tree [14]. Reduced error pruning removes sub tree rooted at that node, making it as a leaf node and assigning it the most common classification of the training data affiliated with that node. Nodes are pruned iteratively with choosing the node whose removal most increases the accuracy of the decision tree and pruning continues until further pruning is harmful, means when the accuracy of the tree is being reduced. In this algorithm, a node will be removed only if the resulting sub tree performs worse than original. REP tree is also called as fast decision tree learner. The drawback of REP tree is when data is limited, if the pruning done on that limited data, the calculated accuracy is not correct.

3.3.2.5. AD Tree (Alternating Decision Tree)

An alternating decision tree combines the simplicity of a single decision tree with the effectiveness of boosting. The knowledge representation combines tree stumps, a common model deployed in boosting, into a decision tree type structure. The different branches are no longer mutually exclusive. The root node is a prediction node, and has just a numeric score. The next layers of nodes are decision nodes, and are essentially a collection of decision tree stumps. The next layer then consists of prediction nodes, and so on, alternating between prediction nodes and decision nodes.

A model is deployed by identifying the possibly multiple paths from the root node to the leaves through the alternating decision tree that correspond to the values for the variables of an observation to be classified. The AD Tree can only deal with the binary class.

3.3.2.6. LAD (Least Absolute Deviation) Tree

A LAD tree is one of the oldest and mostly widely used algorithm which tries to ensure that the resulting model has the smallest possible deviation from the true goal variable values. It is a mathematical optimization technique that attempts to find a function which is closely approximates a set of data in a single dataset. It minimizes the sum of absolute values of errors.

3.3.3. Neural network algorithms

3.3.3.1. Multilayer Perceptron

Multi-Layer perceptron (MLP) is a feed forward neural network with one or more layers between input and output layer. Feed forward means that data flows in one direction from input to output layer (forward). This type of network is trained with the back propagation learning algorithm. MLPs are widely used for pattern classification, recognition, prediction and approximation. Multi-Layer perceptron can solve problems, which are not linearly separable.

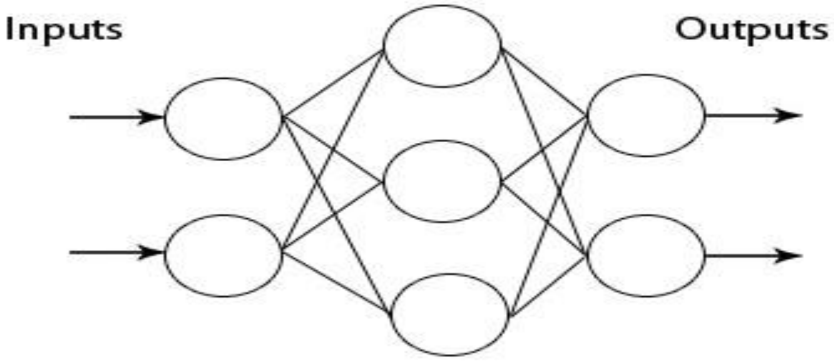


Figure 3: Multilayer perceptron

3.3.3.2. RBF (Radial basis Function) Network

Radial basis function (RBF) networks are known to have very good performance in data mining. K-means clustering algorithm is used to determine the centers and radii of the radial basis functions of the networks. Mostly, the performance of generated RBF networks depends upon given training data sets.

3.3.3.3. WEKA

WEKA is a powerful data mining tool which provides various classifiers, data processing techniques, and feature selection methods to explore and find the suitable and reasonable combined model for data sets.

4. IMPLEMENTATION DETAILS

4.1. Introduction

This section explains about the design and logical flow of the code. This project is developed using java and WEKA software. Java Server Pages (JSP's) were used to design the graphical user interface (GUI). The development of the java classes is discussed later in this chapter. Using the GUI, the user is able to select the provided filters, data sets, decision trees and neural networks. Upon the selection of the disease from the front end, the project loads the respective dataset for filtering and classification. The user selection from the front end is taken as input. Some of the inputs required for this project are defined at the java class level and some user selected inputs are directly been used in the required methods. As mentioned in the third chapter, the data sets were collected from the UCI machine learning repository.

To run the project, one should install java on their local machine, integrated development environment (IDE) like eclipse, server like tomcat to load the project.

4.2. Graphical User Interface (GUI)

For the GUI, two java server pages (JSPs) are defined named as diagnosisEngine.jsp and output.jsp respectively. In diagnosisEngine.jsp, all the input field variables for the datasets, filters, decision trees and neural networks are defined as a dropdown box for each variable to make the selection from the front end, and output.jsp is used to show the output.

4.3. Prediction System Class

This is the class where all the methods are defined and implemented for this project. This class extends httpServlet, which helps the user to select the input variables available in this class. All the packages needed from WEKA are imported into this class to make use of its methods and all the required fields, which use these methods, are defined at the class level. Decision trees and neural networks used in this project have been implemented in this class. The primary function of this class is to remove the irrelevant attributes from the dataset using filters and train the selected classification algorithm using the user selected dataset with already known output values and predict the test dataset with unknown values to known values.

Table 4 below shows the methods that are defined and implemented in this class, and Table 5 shows the different classification methods used.

Table 4: Methods defined in prediction system class

Access Specifier	Return Type	Method	Purpose
Public	Void	modelBuilder()	This is the main method in this class and this method makes available all the user selected inputs to the whole project.
Public	Void	initializeDataSet()	This method is called to initialize the data set.
Public	Void	loadTrainingDataset()	This method is called to read the training dataset from the project.
Public	Void	openTestDataset()	This method is called to read the test dataset from the project.
Public	Void	selectedFilterAndClassifier()	This method is used to filter the training dataset and send it to classifier.
Private	Void	selectedClassifier()	This method is used to call the user selected classifier.

4.4. Prediction System Form Class

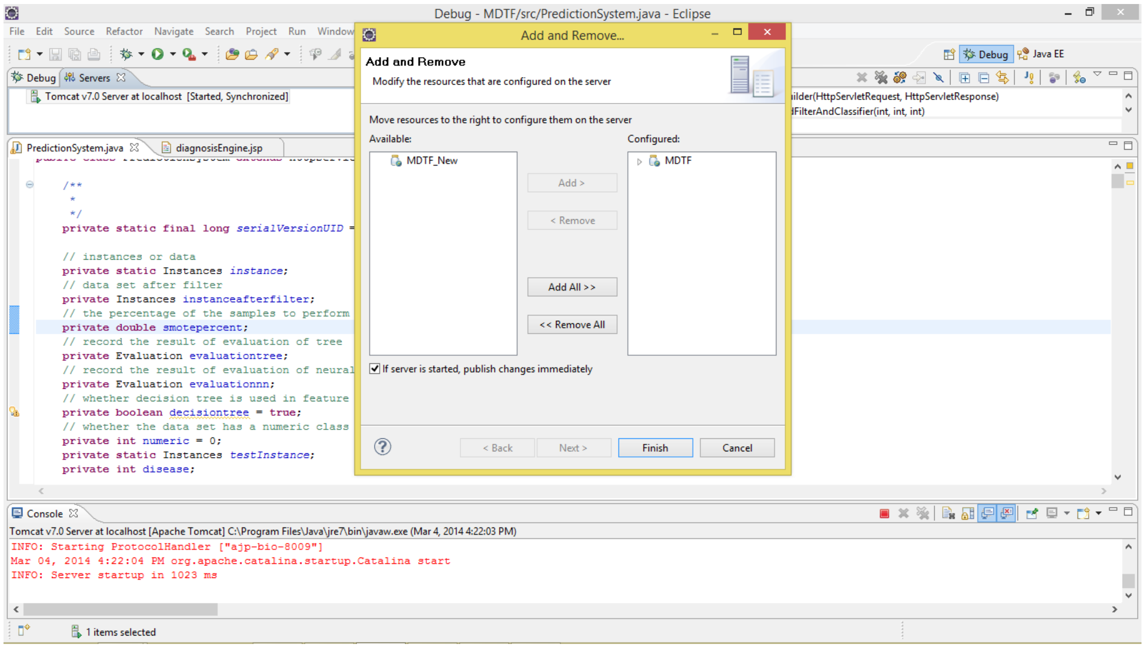
In this class, the required fields and properties (setter and getters) are defined, to make those fields available to the main class. These fields will be called from the prediction system class, to project the output on to output.jsp.

Table 5: Different classification methods

Access Specifier	Return Type	Classifiers	Purpose
Public	Void	J48DecisionTree ()	This method calls WEKA J48 class and will use its methods internally.
Public	Void	randomTree()	This method calls WEKA RandomTree class and will use its methods internally.
Public	Void	callNaiveBayes()	This method calls WEKA NaiveBayes class and will use its methods internally.
Public	Void	REPTree()	This method calls WEKA REPTree class and will use its methods internally.
Public	Void	ADTree()	This method calls WEKA ADTree class and will use its methods internally.
Public	Void	LADTree()	This method calls WEKA LADTree class and will use its methods internally.

4.5. Steps to Load and Run the Project

1. Start eclipse and import the project (MDTF.war) into the integrated development environment (IDE).
2. Add one of the application servers like tomcat to eclipse and load the project into server.



Debug - MDTF/src/PredictionSystem.java - Eclipse
File Edit Source Refactor Navigate Search Project Run Window
Debug
Servers
Tomcat v7.0 Server at localhost [Started, Synchronized]
PredictionSystem.java
diagnosisEngine.jsp
Add and Remove...
Add and Remove
Modify the resources that are configured on the server
Move resources to the right to configure them on the server
Available:
MDTF_New
Add >
< Remove
Add All >>
<< Remove All
Configured:
MDTF
☑ If server is started, publish changes immediately
< Back
Next >
Finish
Cancel
Java EE
ilder(HttpServletRequest, HttpServletResponse)
dFilterAndClassifier(int, int, int)
Console
1 items selected

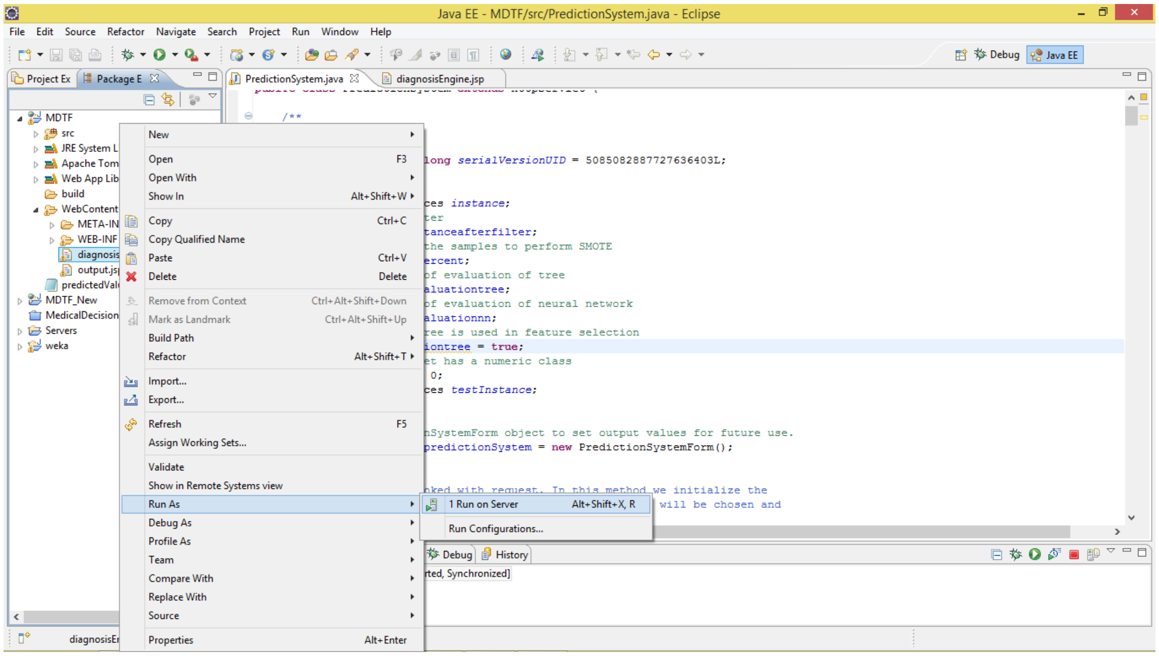
```
    /**
     *
     */
    private static final long serialVersionUID =

    // instances or data
    private static Instances instance;
    // data set after filter
    private Instances instanceafterfilter;
    // the percentage of the samples to perform
    private double smotepercent;
    // record the result of evaluation of tree
    private Evaluation evaluationtree;
    // record the result of evaluation of neural
    private Evaluation evaluationnn;
    // whether decision tree is used in feature
    private boolean decisiontree = true;
    // whether the data set has a numeric class
    private int numeric = 0;
    private static Instances testInstance;
    private int disease;
```

```
Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 4, 2014 4:22:03 PM)
INFO: Starting ProtocolHandler ["ajp-bio-8009"]
Mar 04, 2014 4:22:04 PM org.apache.catalina.startup.Catalina start
INFO: Server startup in 1023 ms
```

Figure 4: Adding project to the application server

3. Run the project by expanding and right clicking on diagnosisEngine.jsp and select run on server option.



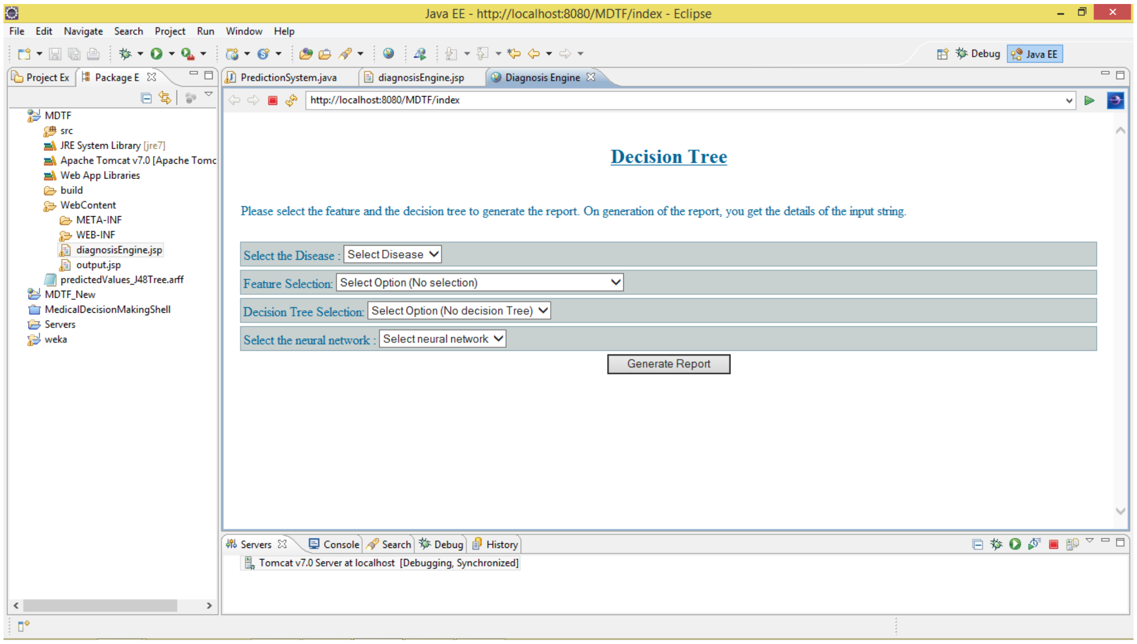
```
    /**
long serialVersionUID = 5085082887727636403L;
ces instance;
ter
tanceafterfilter;
the samples to perform SMOTE
ercent;
of evaluation of tree
aluationtree;
of evaluation of neural network
aluationnn;
ree is used in feature selection
iontree = true;
et has a numeric class
 0;
ces testInstance;

nSystemForm object to set output values for future use.
predictionSystem = new PredictionSystemForm();

oked with request. In this method we initialize the
                       will be chosen and
```

Figure 5: Running the project

4. Choose the selection criteria from graphical user interface (GUI).


Java EE - http://localhost:8080/MDTF/index - Eclipse
File Edit Navigate Search Project Run Window Help
Debug
Java EE
Project Ex
Package E
MDTF
src
JRE System Library [jre7]
Apache Tomcat v7.0 [Apache Tomc
Web App Libraries
build
WebContent
META-INF
WEB-INF
diagnosisEngine.jsp
output.jsp
predictedValues_J48Tree.arff
MDTF_New
MedicalDecisionMakingShell
Servers
weka
PredictionSystem.java
diagnosisEngine.jsp
Diagnosis Engine
http://localhost:8080/MDTF/index
Servers
Console
Search
Debug
History

Decision Tree

Please select the feature and the decision tree to generate the report. On generation of the report, you get the details of the input string.

Select the Disease : Select Disease

Feature Selection: Select Option (No selection)

Decision Tree Selection: Select Option (No decision Tree)

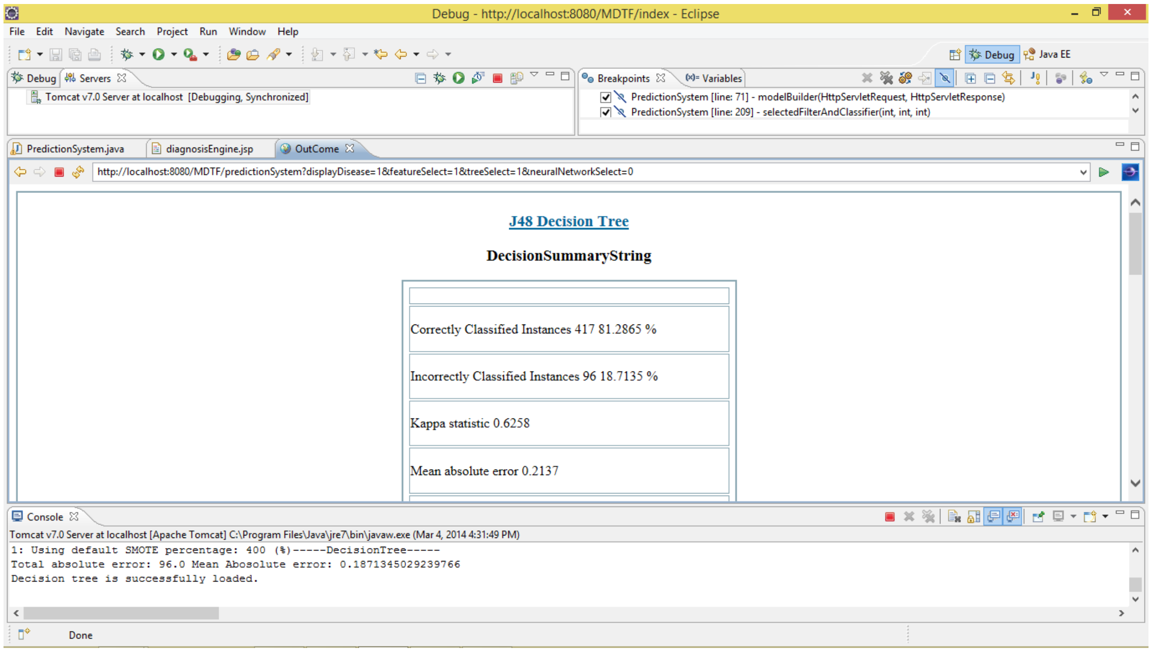
Select the neural network : Select neural network

Generate Report

Tomcat v7.0 Server at localhost [Debugging, Synchronized]

Figure 6: Graphical user interface for user selection

5. Upon the selection, respective output is shown on output page.



Tomcat v7.0 Server at localhost [Debugging, Synchronized]

☑ PredictionSystem [line: 71] - modelBuilder(HttpServletRequest, HttpServletResponse)
☑ PredictionSystem [line: 209] - selectedFilterAndClassifier(int, int, int)

http://localhost:8080/MDTF/predictionSystem?displayDisease=1&featureSelect=1&treeSelect=1&neuralNetworkSelect=0

J48 Decision Tree

DecisionSummaryString

Correctly Classified Instances 417 81.2865 %
Incorrectly Classified Instances 96 18.7135 %
Kappa statistic 0.6258
Mean absolute error 0.2137

Tomcat v7.0 Server at localhost [Apache Tomcat] C:\Program Files\Java\jre7\bin\javaw.exe (Mar 4, 2014 4:31:49 PM)

```
1: Using default SMOTE percentage: 400 (%)-----DecisionTree-----
Total absolute error: 96.0 Mean Abosolute error: 0.1871345029239766
Decision tree is successfully loaded.
```

Figure 7: Output

From the above output screen, it will display the correctly classified instances, incorrectly classified instances, kappa statistics, etc. All the output fields of the experiment are explained below.

Precision is the fraction of retrieved instances that are relevant to the selected disease, i.e., positive predictive value. Precision is a measure of exactness or quality of the dataset. Thus, high precision means that the algorithm returned more relevant instances than irrelevant instances [12].

Precision = Number of attributes retrieved that are relevant / Total Number of attributes

Recall is the fraction of relevant instances that are retrieved. It is also called as sensitivity. Recall is a measure of completeness or quantity. Thus, high recall means that the algorithm returned most of the relevant instances [12].

Recall = Number of attributes retrieved that are relevant **/** Total Number of attributes that are relevant

F-Measure is a combined measure for precision and recall.

F-Measure = 2 * Precision * Recall **/** (Precision + Recall)

The sensitivity (TP Rate) and specificity (FP Rate) is calculated from the weighted average of the instances [13]. The numbers shown in the confusion matrix with ‘a’ and ‘b’ represents the class labels. The true positive (TP) rate is the proportion of instances, which classified as “True” class among all instances, which implies how many of the instances were captured. It is equivalent to recall.

In the confusion matrix, to give an example:

```
a      b   <------- Classified as
7      2  | a = yes
3      2 | b = no
```

TP = diagonal elements / sum over the relevant row.

i.e. TP Rate = 7/ (7+2) = 0.778 for class “yes”, and FP Rate = 2/ (3+2) = 0.4 for class “no”.

4.5.1. Accuracy

The percentage of correctly classified instances in the dataset is measured as accuracy.

For example, if there are 100 instances:

aa + bb = 69+16 = 85;

ab + ba = 11+ 4= 15.

So, from the above, for 100 instances, 85 instances are correctly classified, and 15 instances are not.

4.5.2. ROC area

Area under ROC curve is a preferred measure. It is a single number summary of the performance [21]. Algorithms with a large area under ROC are said to be robust.

4.5.3. Kappa

Kappa is a chance measured of agreement between the classifications and the true classes. If the kappa value is greater than zero, this means that the classifier is better than chance [13].

5. EXPERIMENTAL DESIGN AND RESULTS

5.1. Experimental Design

In this section two sets of experiments were discussed and the results of each set are considered. Each set of experiment is done in a two-step process. Step one is implemented to remove or reduce the irrelevant attributes from the dataset and the step two is to send those attributes to classifier and analyze the data. In addition, classification with no feature selection is also calculated. To present the results, the accuracy and area under ROC are calculated to show how reliable the model predictability is. High accuracy and area of ROC is used to show how reliable the prediction is for the outcome.

The first set of experiments removes or reduces irrelevant attributes from the dataset using a few well known filtering techniques such as cfsSubsetEval, greedy stepwise, gain ratio and ranker search. The dataset after filtering that result in a reduced number of attributes is called the preprocessed data set. The preprocessed dataset is sent to the selected decision tree for classification.

In the second set of experiments, irrelevant attributes from the dataset are removed by the decision tree that was selected from the graphical user interface (GUI). That selected decision tree is treated as the preprocessing technique and the attributes used in the decision tree are forwarded to the neural network.

The performance of the above two sets of experiments are compared under different measures using accuracy and the ROC area and the best classifier is chosen for each disease. The analysis and discussion are also included in this section based on the results obtained in terms of accuracy and ROC area of each experiment; the best suitable classification algorithm for each disease is chosen.

In the subsection below, experiments are calculated by observing each decision tree and neural network with no feature selection for the heart disease.

5.2. Classification with no Feature Selection

In this subsection, both decision trees with no feature selection, and neural networks with no feature selection are calculated for the heart disease data set.

5.2.1. Decision trees with no feature selection

The results of the decision trees with no feature selection are shown in Table 6.

Table 6: Decision trees with no feature selection

Measure / Algorithm	J48 decision tree	Random tree	Naive Bayes tree	REP tree	AD tree	LAD tree
Sensitivity (TP rate)	0.813	0.772	0.712	0.786	0.791	0.805
Specificity (FP rate)	0.187	0.23	0.291	0.212	0.209	0.193
Accuracy	81.28%	77.19%	71.73%	78.55%	79.14%	80.5%
ROC area	0.84	0.771	0.781	0.85	0.869	0.878

5.2.2. Neural networks with no feature selection

The results of the neural networks with no feature selection are shown in Table 7.

Table 7: Neural networks with no feature selection

Measure/Algorithm	Multilayer Perceptron	RBF N/W
Sensitivity (TP rate)	0.723	0.683
Specificity (FP rate)	0.287	0.327
Accuracy	70.23%	66.87%
ROC area	0.765	0.721

5.3. Classification with Feature Selection

In this subsection, both decision trees with feature selection and neural networks with feature selection are applied on heart disease data set.

5.3.1. Decision trees with feature selection

The results of the decision trees with cfsSubsetEval+greedyStepwise feature selection are shown in Table 8.

Table 8: Decision trees with cfsSubsetEval+ greedyStepwiseSearch feature selection

Measure / Algorithm	J48 with Cfs	Random Tree with Cfs	Naive Bayes with Cfs	REP Tree with Cfs	AD Tree with Cfs	LAD Tree with Cfs
Sensitivity (TP rate)	0.832	0.805	0.873	0.795	0.795	0.83
Specificity (FP rate)	0.161	0.194	0.118	0.197	0.2	0.163
Accuracy	83.29%	80.50%	87.33%	79.5%	79.5%	83.04%
ROC area	0.879	0.805	0.912	0.871	0.877	0.89

The results of the decision trees with gain ratio+ ranker search feature selection are shown in Table 9.

5.3.2. Neural networks with feature selection

The results of the neural networks with cfsSubsetEval+greedyStepwise feature selection are shown in Table 10.

Table 9: Decision trees with gain ratio+ ranker search feature selection

Measure / Algorithm	J48 with GainRatio	Random Tree with GainRatio	Naive Bayes with GainRatio	REP with GainRatio	AD Tree with GainRatio	LAD Tree with GainRatio
Sensitivity -TP	0.811	0.799	0.712	0.788	0.791	0.805
Specificity -FP	0.19	0.202	0.291	0.21	0.209	0.193
Accuracy	81.09%	339.92%	71.15%	78.75%	79.14%	80.5%
ROC area	0.838	0.799	0.771	0.858	0.869	0.878

Table 10: Neural networks with cfsSubsetEval+ greedyStepwiseSearch feature selection

Measure / Algorithm	Multilayer Perceptron with Cfs	RBF N/W with Cfs
Sensitivity (TP rate)	0.712	0.69
Specificity (FP rate)	0.294	0.318
Accuracy	71.15%	69.00%
ROC area	0.755	0.732

The results of the neural networks with cfsSubsetEval+greedyStepwise feature selection are shown in Table 11.

Table 11: Neural networks with gain ratio+ ranker search feature selection

Measure/Algorithm	Multilayer Perceptron with Cfs	RBF N/W with Cfs
Sensitivity(TP rate)	0.735	0.708
Specificity(FP rate)	0.272	0.298
Accuracy	73.49%	70.76%
ROC area	0.797	0.762

5.3.3. Results and Analysis of classification with feature selection

Figures 3 and 4 illustrate the accuracy and ROC area results based on the conducted experiments of decision trees and neural networks with feature selection.

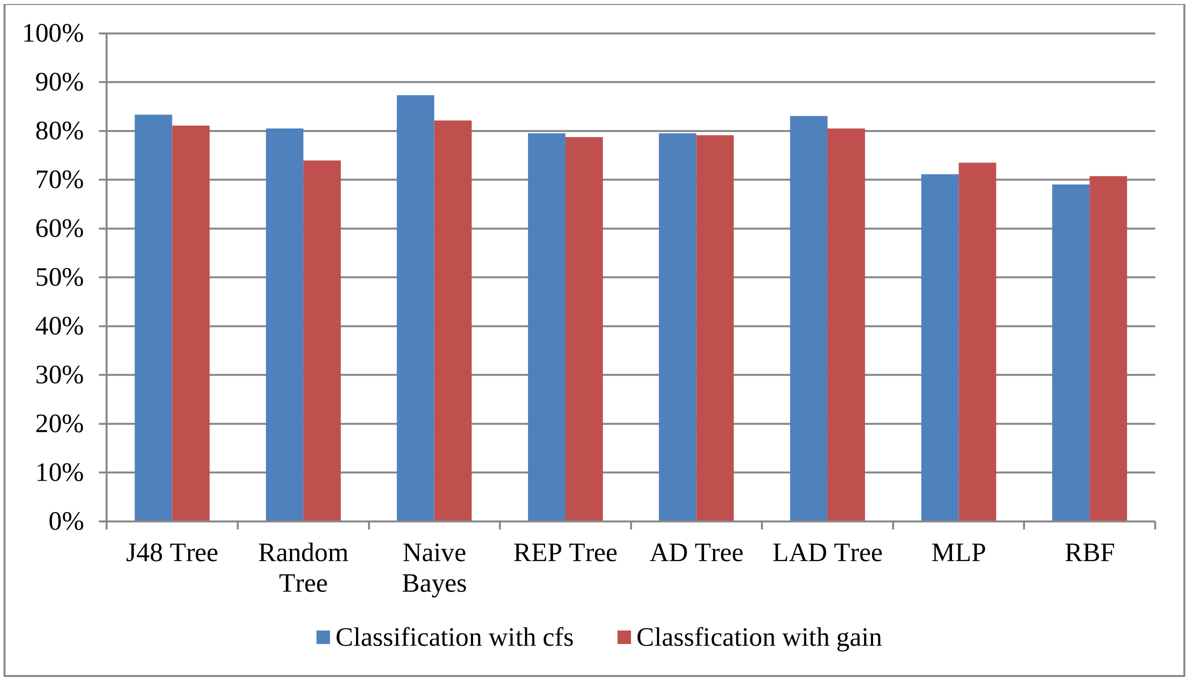


Figure 8: Accuracy of classification algorithms with feature selection

Figure 8 shows the accuracy result with cfsSubsetEval+greedyStepwise and gainRatio+rankerSearch feature selection for all decision trees and neural networks. From the figure one can see that the Naïve Bayes classifier gives the highest accuracy with 87.33% with cfsSubsetEval+ greedyStepwise, and 82.15% with gainRatio+rankerSearch. Based on observation the accuracy of the decision tree is higher than that of neural networks.

From Figure 9, one can observe that the ROC area with cfsSubsetEval+greedyStepwise and gainRatio+rankerSearch feature selection is higher for all decision trees and neural networks. Among all the results, Naïve Bayes tree with cfsSubsetEval+greedyStepwise feature selection gives the highest ROC Area with 0.912 and the second best ROC area is achieved by the LAD decision tree with 0.89. J48 tree and AD tree also results in almost equal performance as LAD with 0.879 and 0.877 respectively. From Figures 8 and 9, Naïve Bayes decision tree with cfsSubsetEval+greedyStepwise gives the highest accuracy and ROC area.

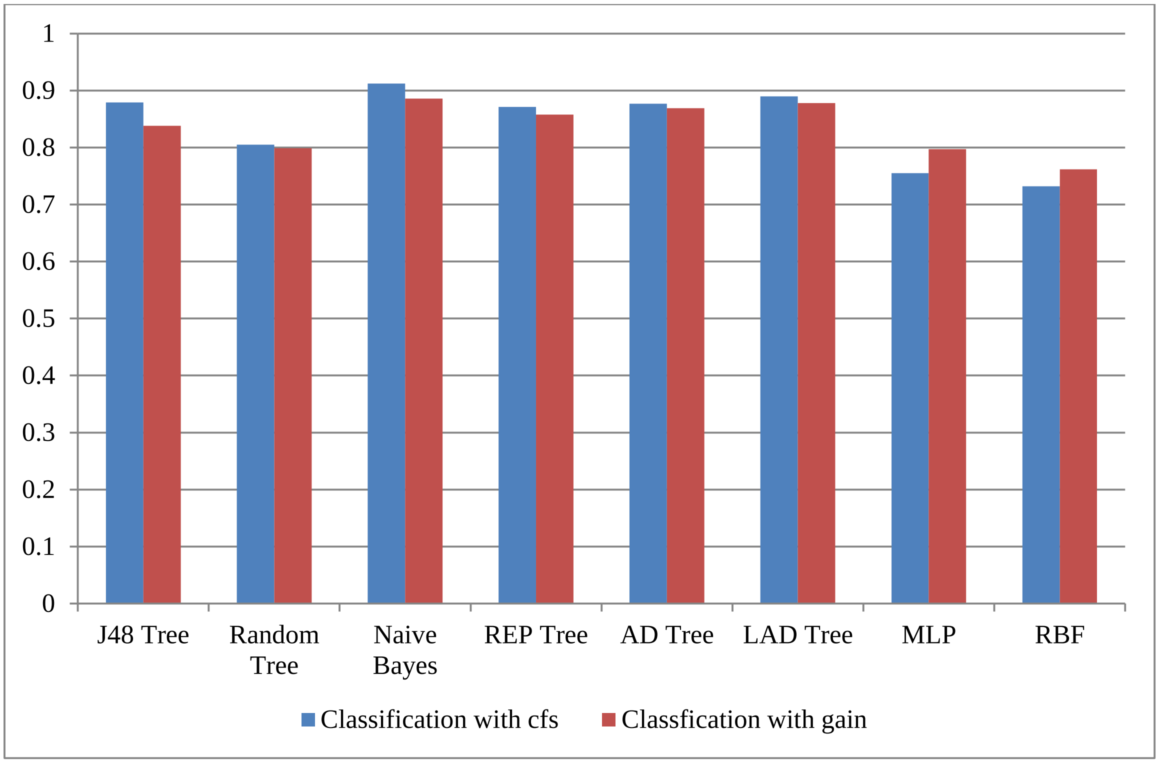


Figure 9: ROC area of classification algorithms with feature selection

5.4. Combined Model of Decision Trees and Neural Networks

In this subsection, the decision tree algorithms are used as preprocessing techniques and the neural network algorithms are used as classifiers.

5.4.1. Neural network with decision tree as feature selection

The accuracy and ROC area of the multi-layer perceptron (MLP) with decision trees is shown in Table 12.

Table 12: Multi-layer perceptron (MLP) with decision trees as feature selection

Measure / Algorithm	MLP with J48	MLP with Random Tree	MLP with Naive Bayes	MLP with REP Tree	MLP with AD Tree	MLP with LAD Tree
Sensitivity (TP rate)	0.745	0.729	0.745	0.731	0.68	0.729
Specificity (FP rate)	0.257	0.276	0.257	0.273	0.324	0.276
Accuracy	74.46%	72.904%	72.90%	73.10%	68.03%	72.9%
ROC area	0.789	0.779	0.78	0.776	0.748	0.779

The accuracy and ROC area of the RBF networks with decision trees as feature selection is shown in the Table 13.

Table 13: Radial basis function (RBF) network with decision trees as feature selection

Measure / Algorithm	RBF with J48	RBF with Random tree	RBF with Naive Bayes	RBF with REP tree	RBF with AD tree	RBF with LAD tree
Sensitivity (TP rate)	0.708	0.706	0.708	0.669	0.645	0.706
Specificity (FP rate)	0.298	0.3	0.298	0.331	0.365	0.3
Accuracy	70.76%	70.56%	70.56%	66.86%	64.52%	70.56%
ROC area	0.762	0.763	0.763	0.702	0.687	0.763

5.4.2. Results and analysis of neural network with decision tree as feature selection technique

Figure 6 and 7 illustrates the accuracy and ROC area based on the experimental results from neural networks with decision trees as feature selection.

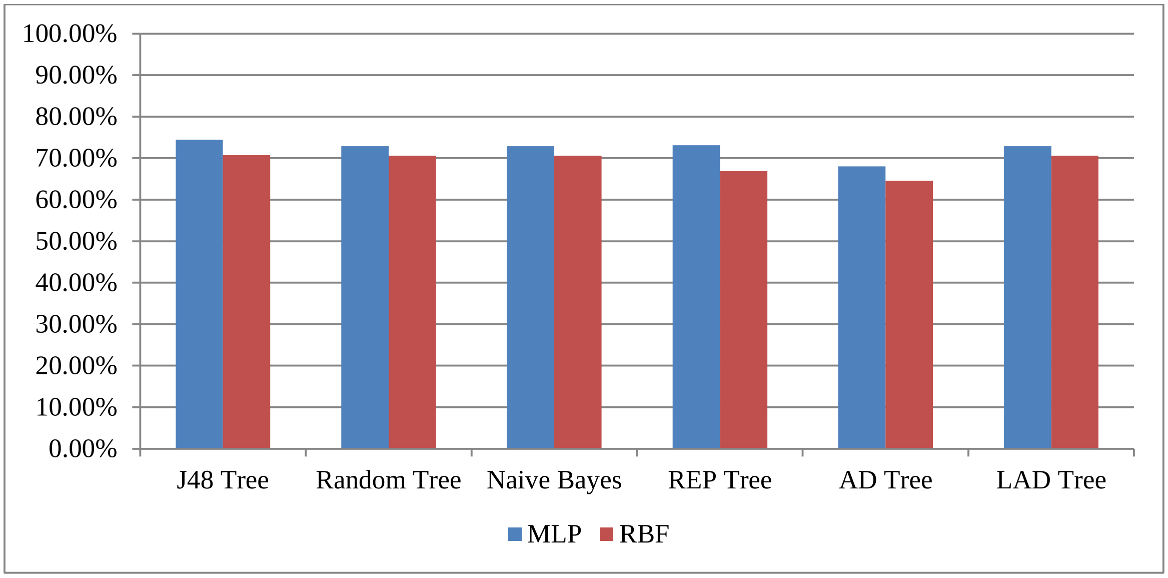


Figure 10: Accuracy of neural network with decision tree as feature selection

From Figure 10, one can observe that the multilayer perceptron with J48 decision tree gives the highest accuracy with 74.46%. For the other four decision tree algorithms, they also give higher accuracy when they are compared with RBF neural network. From the figure we can also find that the RBF networks give the lowest accuracy. In conclusion, with the default settings in WEKA, multilayer perceptron with J48 decision tree give best accuracy among all the algorithms and the second best accuracy is achieved by the multilayer perceptron with REP tree with 73.1%. Comparison between ROC areas of neural networks with different decision trees is shown in figure 11.

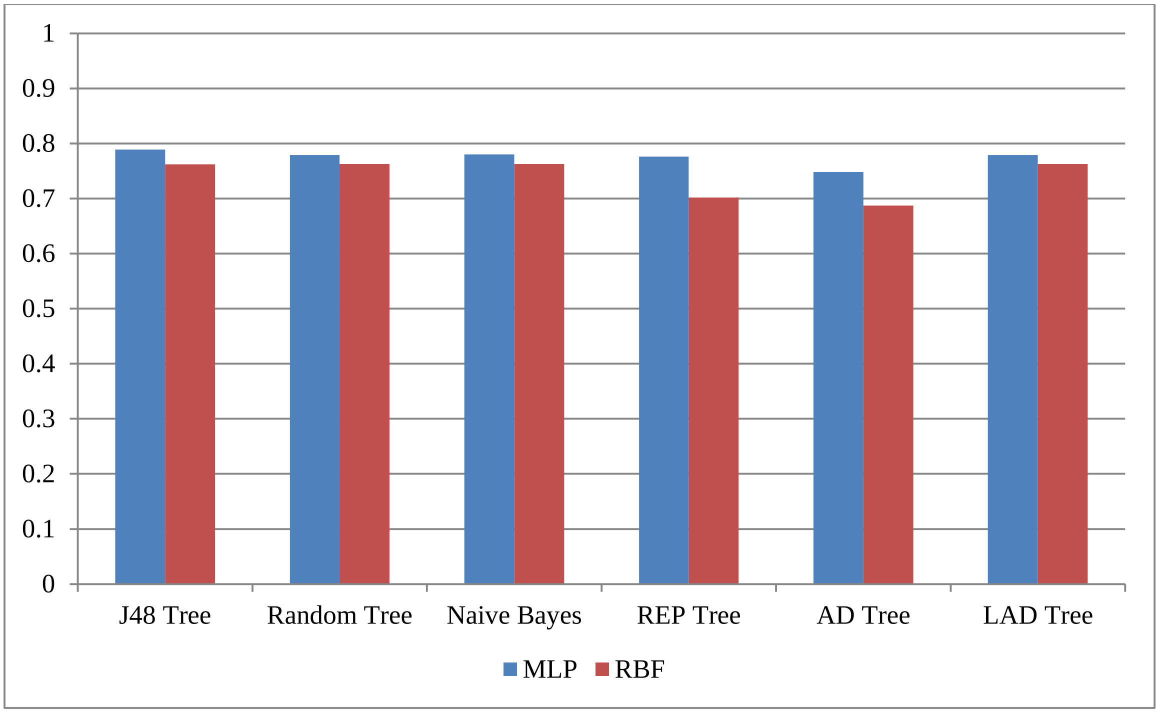


Figure 11: ROC area of neural network with decision tree as feature selection

Among all the algorithms, the multilayer perceptron with J48 decision tree gives the highest ROC area, which is 0.789, and the second best ROC area is obtained by the multilayer perceptron with Naïve Bayes decision tree with a ROC area 0.78. The RBF networks give the lowest ROC area among all algorithms.

5.5. Evaluation of Classification Algorithm

In this subsection, the two sets of experiments, (i) classification algorithms with filters; (ii) combination of neural networks with decision trees are compared. On the basis of accuracy and ROC area, the best combination from Sections 5.3 and 5.4 are selected. Figure 12 displays the results of all the combined models.

5.5.1. Evaluation of classification for heart disease

Some of them used feature selection, some of them did not use feature selection. The highest accuracy for the heart disease dataset as seen from the figure above is given by Naïve

Bayes with cfsSubsetEval feature selection. It indirectly proves that the feature selection can give a better performance.

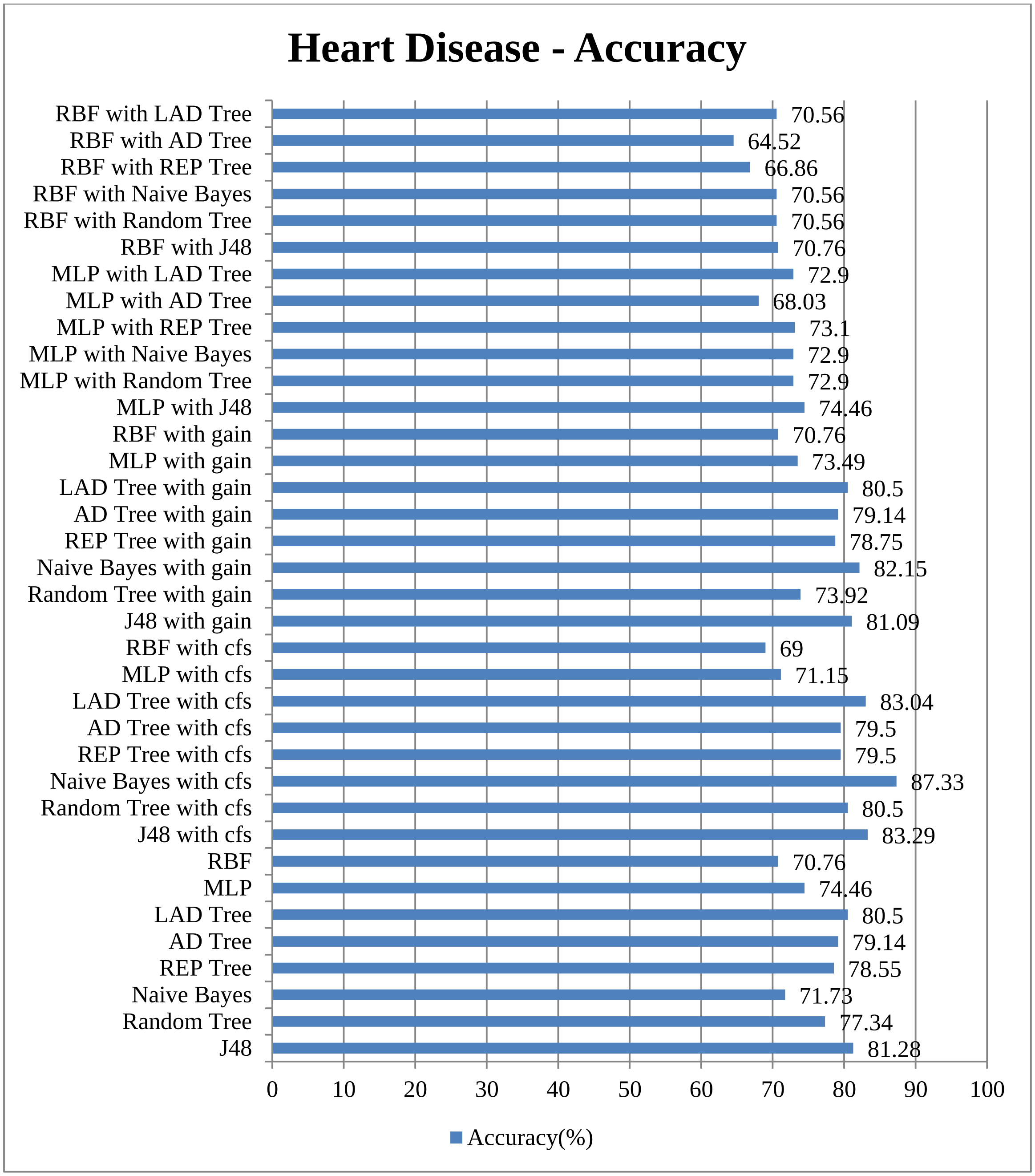


Figure 12: Evaluation of classification for heart disease through accuracy

From the graph above, we can find the cfsSubsetEval feature selection improves the performance of Naïve Bayes tree. For Random tree, the accuracy with feature selection is higher than that without feature selection. Multilayer Perceptron (with Random Tree)'s accuracy is lower than that with feature selection. Also, the performance of Multilayer Perceptron (with AD tree) is lower than that with feature selection. In conclusion, we find that the accuracy of the many combined models is improved when feature selection is applied. Figure 13 shows the ROC area in different combined models.

Among all the algorithms, the Naïve Bayes Tree (with cfsSubsetEval feature selection) gives the highest ROC area with 0.912. From the above figure, for J48 tree, the ROC area increases with feature selection. While for the multilayer perceptron (with J48), the feature selection does not give a higher ROC area. Through detailed observation, it seems that feature selection does not have a distinct impact on the ROC area.

In conclusion, for the heart disease data set, the Naïve Bayes tree gives the highest accuracy and ROC area. Decision trees with feature selection give better performance than neural networks with decision trees.

5.5.2. Evaluation of classification for hepatitis

Based on the conducted experiments on the classification algorithms, the accuracy and ROC area of all the experiments were calculated and evaluated. Among those, the algorithm having the highest accuracy and ROC area was chosen as the best classification algorithm for the hepatitis disease. Figure 14 displays the results of the combined models.

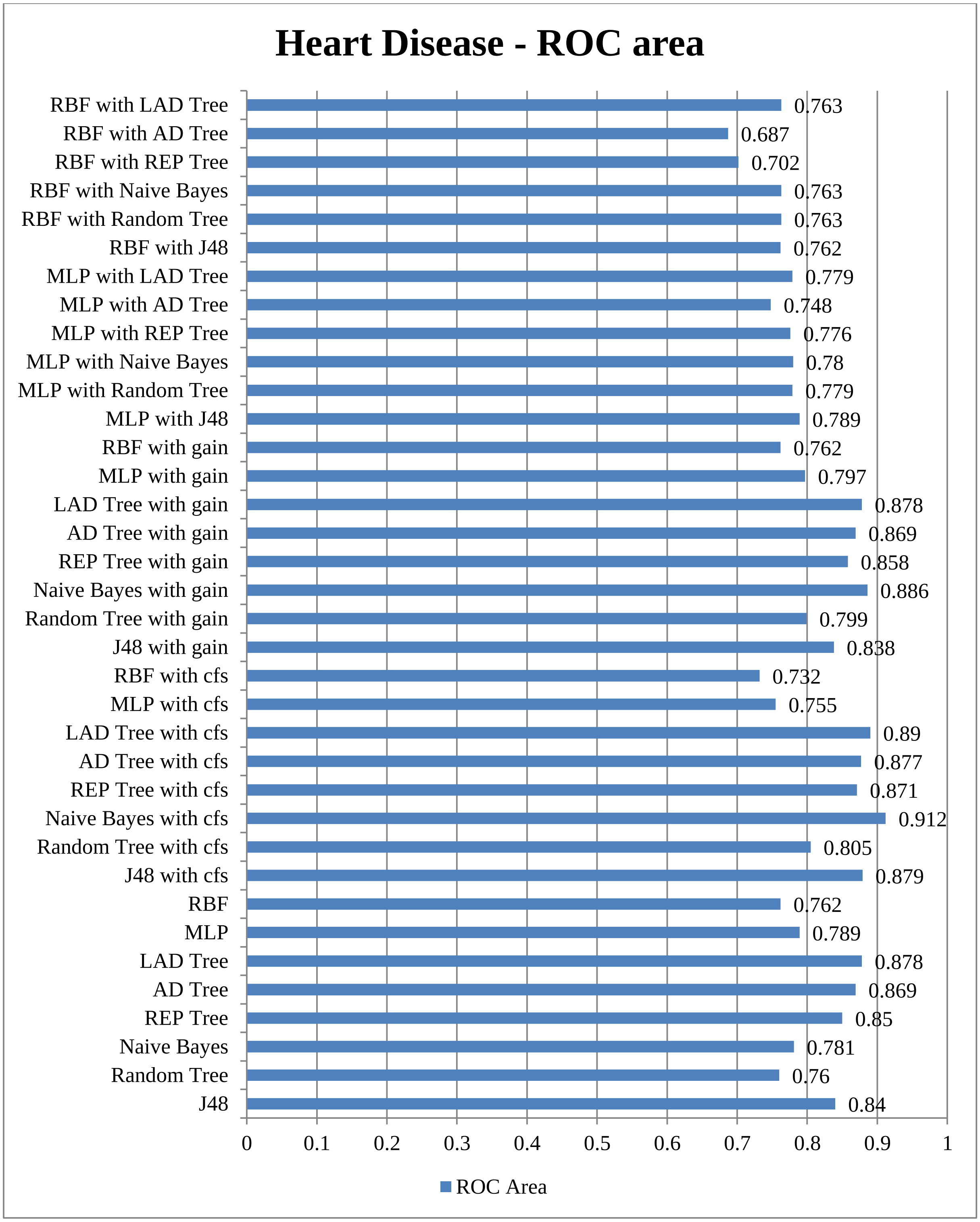


Figure 13: Evaluation of classification for heart disease through ROC area

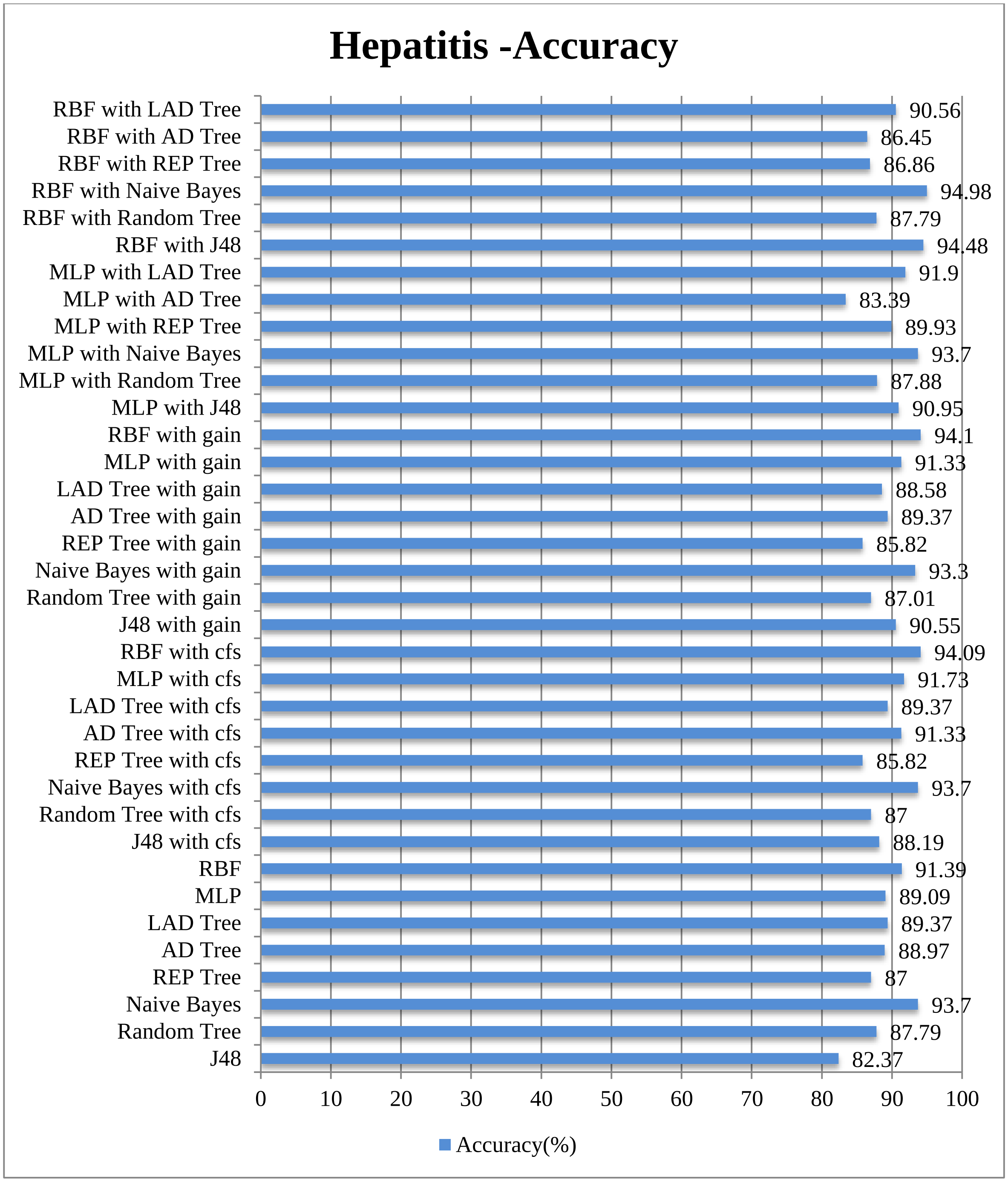


Figure 14: Evaluation of classification for hepatitis disease through accuracy

Some of them used feature selection, some of them did not use feature selection. The highest accuracy for the hepatitis disease from the figure above is given by the RBF neural network with Naïve Bayes decision tree as feature selection. In conclusion, we find that the accuracy of the many combined models is improved through feature selection. Figure 15 shows the ROC area for the different combined models.

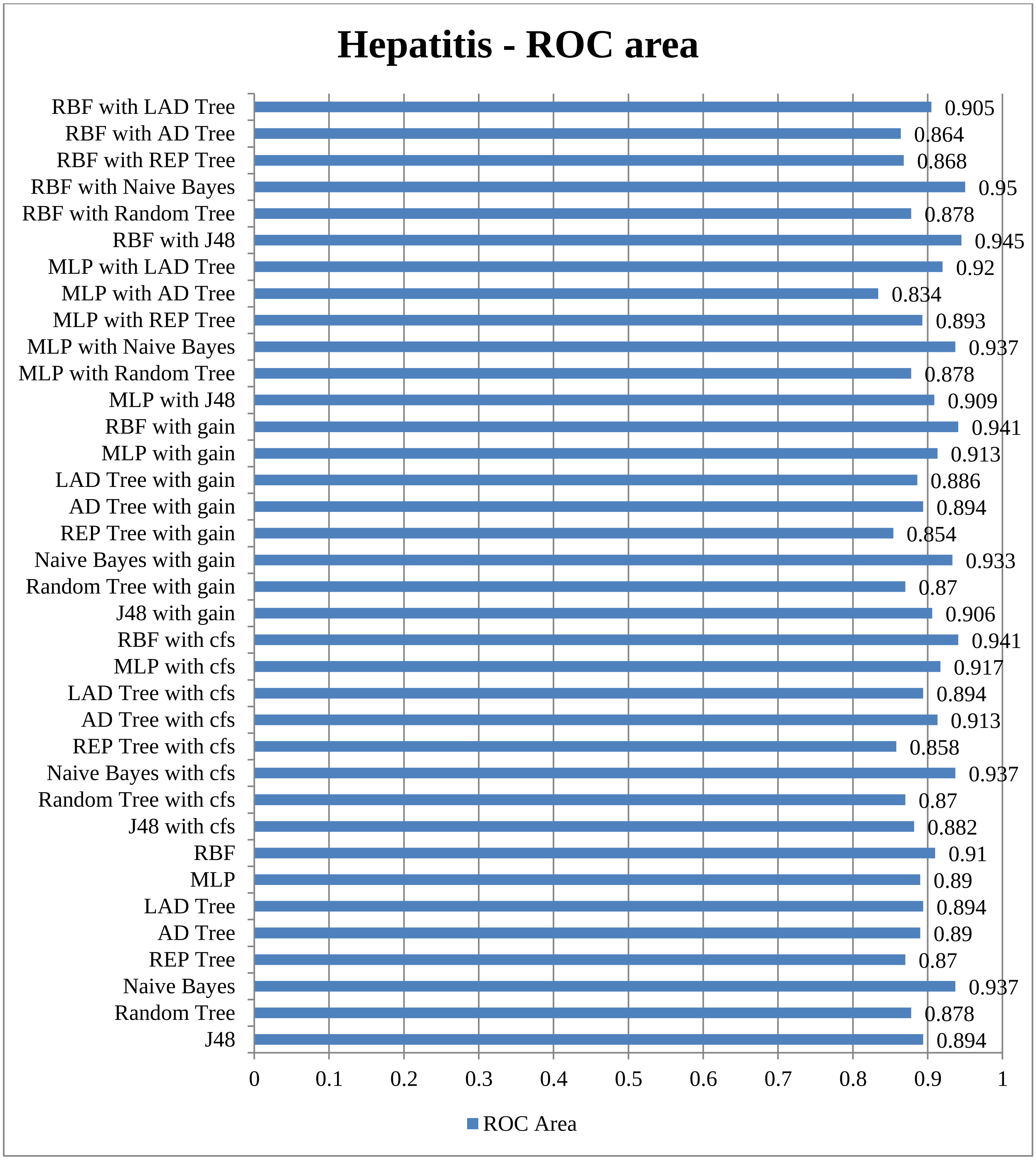


Figure 15: Evaluation of classification for hepatitis disease through ROC area

Among all the algorithms, the RBF neural network with Naïve Bayes Tree as feature selection gives the highest ROC area with 0.95.

In conclusion, for the hepatitis disease, the Naïve Bayes tree gives the highest accuracy and the RBF neural network with Naïve Bayes tree gives the highest ROC area. Decision trees with feature selection give better performance than neural networks with decision trees.

5.5.3. Evaluation of classification for dermatology

Based on the conducted experiments on the classification algorithms, the accuracy and ROC area of all the experiments were calculated and evaluated. Among those, the algorithm having the highest accuracy and ROC area was chosen as the best classification algorithm for the dermatology disease. Figure 16 displays the results of the combined models.

The highest accuracy for the dermatology disease in the figure 16 is given by the RBF neural network with cfsSubsetEval+greedyStepwise feature selection. In conclusion, we find that the accuracy of the many combined models is improved through feature selection. Figure 17 shows the ROC area for the different combined models.

Among all the algorithms, the RBF neural network with cfsSubsetEval+greedyStepwise filter gives the highest ROC area with 0.984. In conclusion, for the dermatology disease, the RBF neural network classification technique gives the highest accuracy and ROC area. Classification algorithms with feature selection give better performance than neural networks with decision trees.

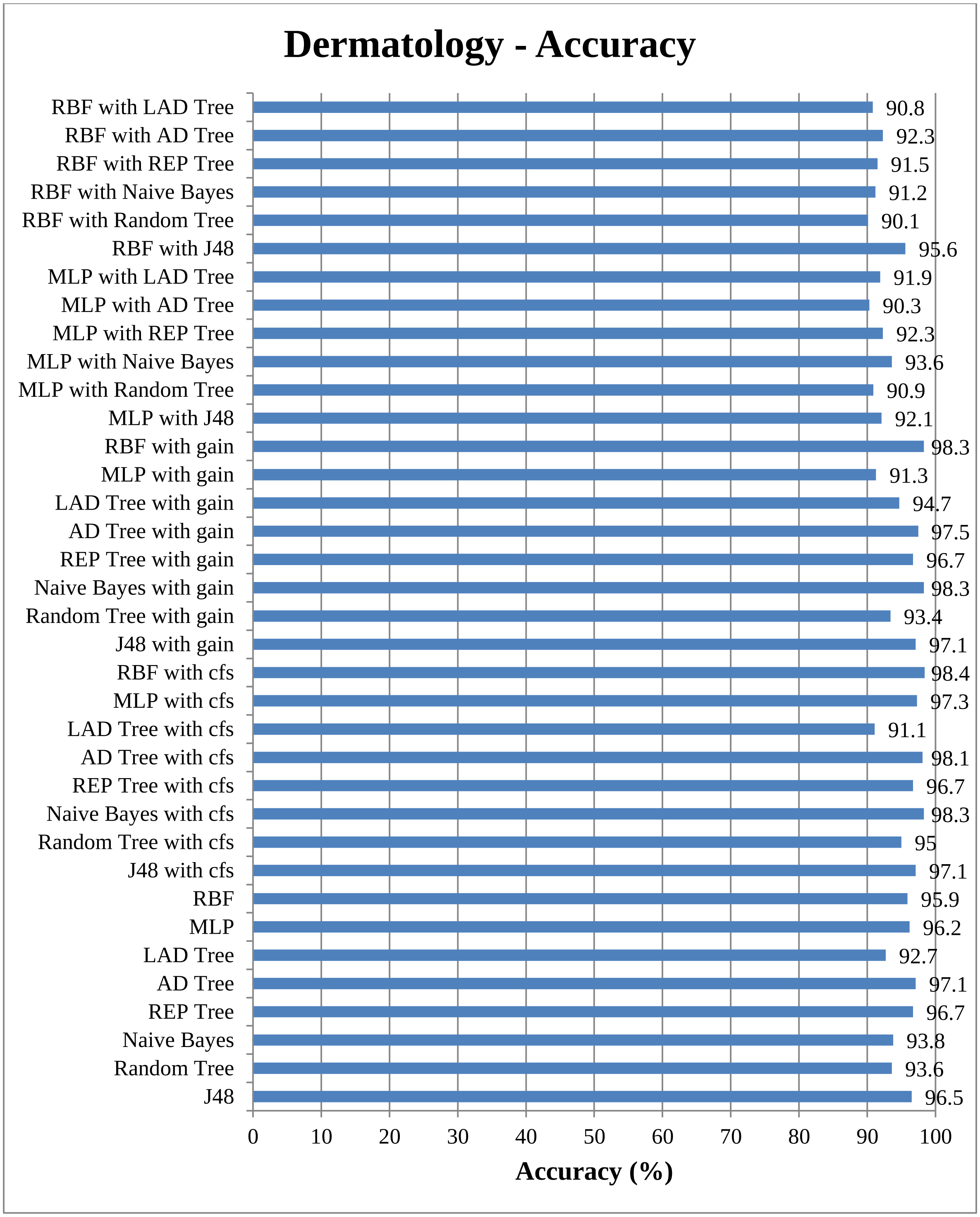


Figure 16: Evaluation of classification for dermatology disease through accuracy

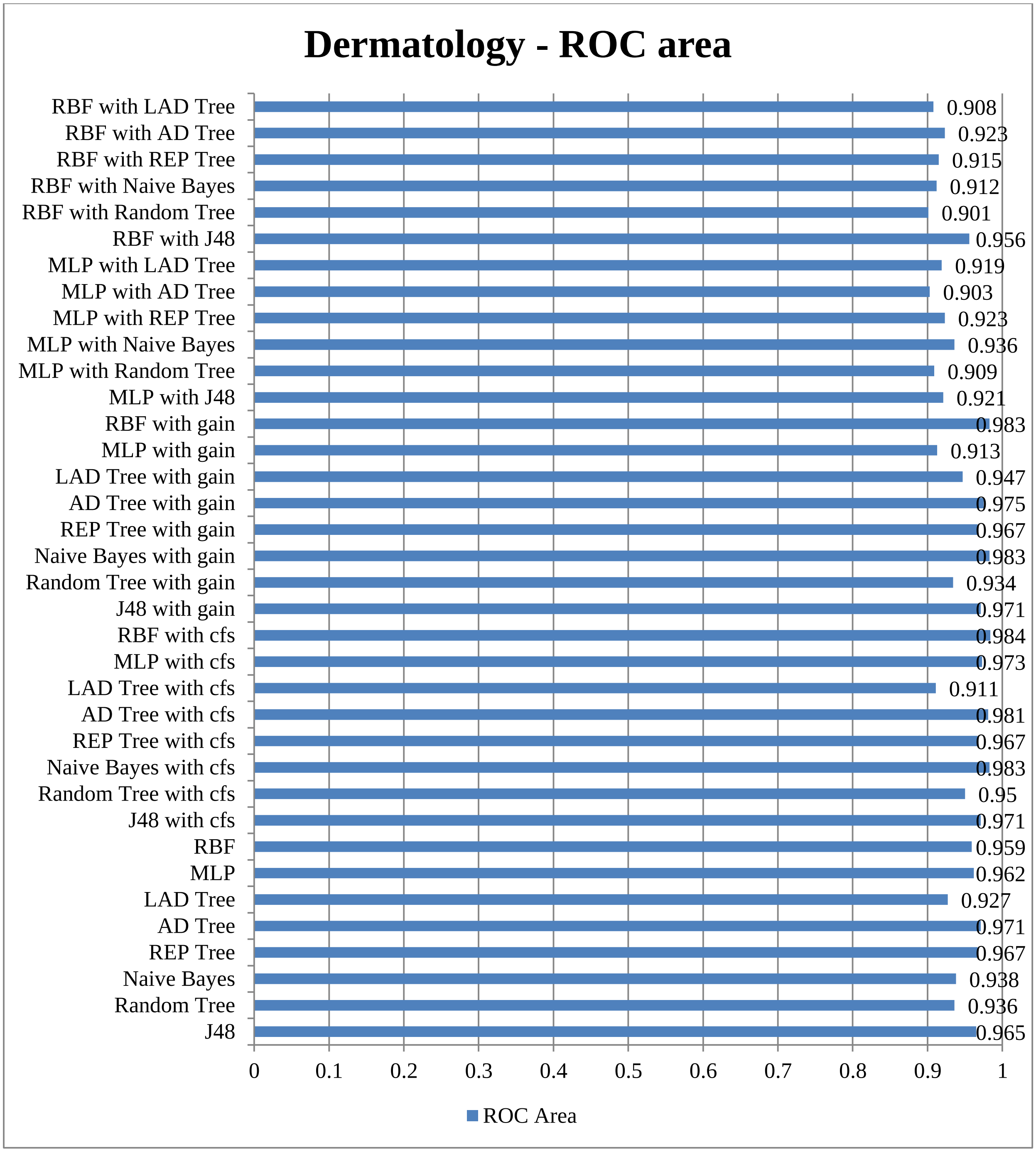


Figure 17: Evaluation of classification for dermatology disease through ROC area

6. CONCLUSIONS AND FUTURE WORK

The objective of this project was to provide a software tool for physicians or medical practitioners to help them in predicting/diagnosing a patient's health condition. For this, a literature review on data mining and different classification techniques was performed and with the help of WEKA methods, a software tool was developed. In the process of the tool development, an analysis is conducted on classification algorithms and six different decision trees and two neural networking algorithms were selected for the use in this project. By calculating the accuracy and ROC area of each classification algorithm for three data sets, the best classification algorithm is identified for each disease. The test data sets were passed to the classification algorithms of the chosen disease and the class was predicted for all classifiers. Using the filters we have successfully reduced the number of irrelevant attributes in the dataset and obtained better predicted results. Based upon the results observed using the accuracy and ROC area measures, we can conclude that the Naïve Bayes decision tree cfssubseteval+greedystepwise gives 87.3% accuracy and 91.2% ROC area for heart disease; and for hepatitis, the RBF neural network with Naïve Bayes tree gives the highest accuracy and ROC area with 94.98% and 0.95. For dermatology disease, the RBF neural network with cfssubseteval+greedystepwise filtering technique gives the highest accuracy and ROC area with 98.4% and 0.984.

As to future work, a more comprehensive study could be conducted including other data mining algorithms as well such as support vector machines, evolutionary algorithms, etc. Furthermore, since data balancing techniques also have a major influence on the prediction ability of classifiers, different techniques such as the smote algorithm could be applied.

REFERENCES

[1] Masci, Paolo, et al. "Using PVS to support the analysis of distributed cognition systems." *Innovations in Systems and Software Engineering* (2013): 1-18.

[2] Kusiak, Andrew, et al. "Autonomous decision-making: a data mining approach. "*Information Technology in Biomedicine, IEEE Transactions on* 4.4 (2000): 274-284.

[3] Pal, Jiban K. "Usefulness and applications of data mining in extracting information from different perspectives." (2011).

[4] Ferreira, Duarte, Abílio Oliveira, and Alberto Freitas. "Applying data mining techniques to improve diagnosis in neonatal jaundice." *BMC medical informatics and decision making* 12.1 (2012): 143.

[5] Acharya, U. Rajendra, and Wenwei Yu. "Data Mining Techniques in Medical Informatics." *The open medical informatics journal* 4 (2010): 21.

[6] Bellazzi, Riccardo, and Blaz Zupan. "Predictive data mining in clinical medicine: current issues and guidelines." *International journal of medical informatics* 77.2 (2008): 81-97.

[7] Eldin, Ahmed Mohamed Samir Ali Gamal. "A Data Mining Approach for the Prediction of Hepatitis C Virus protease Cleavage Sites." *International Journal of Advanced Computer Science & Applications* 2.12 (2011).

[8] Hall, Mark, et al. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11.1 (2009): 10-18.

[9] Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.

[10] Guillet, Fabrice, and Howard J. Hamilton. *Quality measures in data mining*. Vol. 43. Heidelberg: Springer, 2007.

[11] Powers, David MW. "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness & correlation." *Journal of Machine Learning Technologies* 2.1 (2011): 37-63.

[12] Kononenko, Igor, Ivan Bratko, and Matjaž Kukar. "Application of machine learning to medical diagnosis." *Machine Learning and Data Mining: Methods and Applications* 389 (1997): 408.

[13] Yasin, Huda, Tahseen A. Jilani, and Madiha Danish. "Hepatitis-C Classification using Data Mining Techniques." *International Journal of Computer Applications*24 (2011).

[14] Bhatia, Nidhi, and Kiran Jyoti. "An analysis of heart disease prediction using different data mining Techniques." *International Journal of Engineering Research and Technology (IJERT) Vol* 1 (2012).

[15] Fakhraei, Shobeir, et al. "Confidence in medical decision making: application in temporal lobe epilepsy data mining." *Proceedings of the 2011 workshop on Data mining for medicine and healthcare*. ACM, 2011.

[18] Chen, Kani, et al. "Analysis of least absolute deviation." *Biometrika* 95.1 (2008): 107-122.