AN EFFECTIVE APPROACH FOR EARLY LIVER DISEASE PREDICTION

FARIA AFRIN

This thesis is submitted in fulfilment of the requirements for the award of the degree of Master of Science in Information and Communication Engineering

Department of Information and Communication Engineering
NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

8th December 2020

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science in "Information and Communication Engineering"

Signature
Name of Supervisor: PROFESSOR DR. MD. ASHIKUR RAHMAN KHAN
Position: CHAIRMAN, DEPARTMENT OF ICE, NSTU
Date:

Signature
Name of Co-Supervisor: ZAYED-US-SALEHIN
Position: ASSISTANT PROFESSOR, DEPARTMENT OF ICE, NSTU
Date:

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my won except for quotations and summaries which have been duly acknowledged. The thesis has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature
Name: FARIA AFRIN
ID Number: BKH1711MSc117F
Date:

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my thesis supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guideline, continuous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science and his belief that a MSc program is only a start of a life-long learning experience. I appreciate his consistent support from the first day of I applied to graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my naïve mistakes. I also would like to express very special thanks to my co-supervisor Zayed-Us-Salehin for his suggestions and co-operation throughout the study. I also sincerely thank for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to my lab mates and members of the staff of the Information and Communication Engineering Department, NSTU, who helped me in many ways.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life.

ABSTRACT

Liver is one of the main organ of our body. It will be functioning normally even when it is partially damaged, therefore problems with liver patients cannot easily discovered in an early stage. Patient's survival rate can be increased by an early identification of liver problems. Liver disease can be diagnosed by observing the levels of enzymes in the blood. Many researchers working on this issue and try to find the best algorithm with using Liver patient dataset which is suitable for predicting disease in early age. In this research liver patient dataset is investigated for building different classification models to get better result by comparing with the accuracy of the classifiers for disease prediction. Real data in this purpose is collected from the hospitals. This paper uses different classification methods including Bagged Trees, Support vector machine (SVM), K-Nearest Neighbor (KNN), Fine Tree classification methods. It can give a good impact on the liver disease diagnosis and it can be beneficial for physicians and can help to reduce the cost of diagnosis in the medical sector.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

SVM	Support Vector Machine
KNN	K-Nearest Neighbor
MATLAB	Matrix Laboratory
MLP	Multilayer Perceptron
WHO	World Health Organization
CLD	Chronic Liver Diseases
TB	Total Bilirubin
DB	Direct Bilirubin
Alkphos	Alkaline Phosphatase
Sgpt	Alanin Aminotransferase
Sgot	Aspartate Aminotransferase
A/G	Albumin/Globulin ratio
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
ERR	Error Rate
ACC	Accuracy
PREC	Precision
ROC	Receiver Operating Characteristic
TPR	True Positive Rate
FNR	False Negative Rate
PPV	Positive Predictive Values
FDR	False Discovery Rates
NN	Neural Network
RS	Rough Set

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

In recent years, Healthcare organizations produces huge amounts of different kind of data from hospitals, resources, disease diagnosis, patient records, etc. The medical data are globally exaggerated and distributed in nature. This increase in data volume automatically requires the data to be retrieved when needed. The large amount of data is critical to be processed and analyzed for knowledge extraction that allows support for understanding the usual circumstances in healthcare sector. A major problem in health science is in managing the correct diagnosis of certain important information. In this respect, the healthcare industry look into how data can be better captured, stored, prepared and evaluated.

The main goal behind the Healthcare System is to deliver the best quality of services and to predict the diseases at an early stage. Liver is an essential body part of our body. It is the challenging duty for doctors to predict the disease accurately. There is a great essential for an early detection of liver disease so that prevent complete liver failure which can consequence in patient's death. For the proper diagnosis, it is important to evaluate some of the leading attributes of liver patient's dataset. Recently, classification techniques in the field of artificial intelligence began to be used in clinical treatment process so that clinical mining process had become an essential technique to be utilized in diagnosing and treating patients by physicians. Therefore, it is desirable to learn the process about how the data can be suitably treated on which evaluating criteria after understanding many useful algorithms. Having access to classification algorithms with large amount of data will help physicians to make better decisions and improve patient outcomes with an accurate prediction of liver disease. With the use classification techniques is possible to extract the knowledge. The knowledge gained in this way can be used in the proper in order to improve work efficiency and enhance the quality of decision making.

In this paper, demonstrated experiments using properly proposed techniques for prediction test and compared them on some evaluating criteria to the data set containing

liver disease patients. For the experiments, total four numbers of classification algorithms are used.

1.2 PROBLEM STATEMENT

In new era, the diagnosis of disease is a difficult and tedious task in medical field. The main challenge facing in healthcare organizations is providing quality services in reasonable costs. Quality service denotes that diagnosing of patients correctly and give treatment which is effective. Terrible incidents can be happened by poor clinical decisions which is unacceptable. In the human body, liver is the largest body organ and it supports almost every body part in the body and is also vital for our survival. The liver organ plays a very important role in many functions like decomposition of red blood cells, in metabolism and serving several vital functions. Liver disease may not cause any indications at earlier stage or the symptoms may be indefinite, like weakness and loss of energy. Symptoms of this disease partially be determined by on the type and in result leads liver disease. Liver diseases are diagnosed based on liver functional test. Diagnosis of liver disease from various factors or symptoms is a multi-layered issue which may lead to false assumptions. Only human cannot alone not enough for proper diagnosis. A number of difficulties will arrive during diagnosis, such as less accurate results, less experience, time dependent performance, knowledge up gradation difficulties. Hospitals and clinics accumulate a huge amount of patient data over the years. These data provide a basis for the analysis of risk factors for many diseases. This huge data can be used for prediction of the liver disease. The diagnosis process of detecting liver disease is comparatively time consuming and laboured job for a person to go to the hospital and do all the tests and wait for getting the test report to know whether he/she has disease or not. Testing and treatment is not always possible, but scaling up of interventions can lead to a rapid reduction in infection rates and lower costs for diagnosis and management. If we can detect the probability of the disease in the early stage, then it will help to minimize the percentage of death. The necessity of the prediction of the disease in advance become high. For this reason, more research is needed to find the most effective techniques for early prediction of liver disease. This research includes the method of analysing liver disease by using various classification techniques Bagged Trees, Support Vector Machine (SVM), K-Nearest Neighbour (KNN) and Fine Tree.

1.3 OBJECTIVES OF THE STUDY

This research paper is worked on classifier algorithms and provides better result that will directly help to the physicians for accurate diagnosis. The main purposes of research are as follows:

I. To collect real data from patients by patients from hospitals.
II. To compare classifiers by accuracy and performance analysis.
III. To find out the best one classification algorithm.
IV. To extract the predictive class for effective diagnosis.

1.4 SCOPE OF THE STUDY

In order to make effective medical diagnosis it is important to discovery knowledge from medical databases. The Liver disease become a life threatening disease not only for the Bangladesh but also for the overall people of the world. Predicting disease in early age become high demand for new era. Many researchers working on this fact and try to discover the best algorithm for predicting disease. By inspecting the survey of previous works, this paper is proposed to help physicians not only to diagnose and predict the liver disease at early stage but also helps to minimize the cost of the diagnosis process. This research work with the Liver patients' dataset which is based on different zones of Bangladesh. Data of liver patients is collected from various hospitals, diagnostic center and clinic center. This paper has focused mainly on classification techniques to process and analysis Liver patient dataset. This research uses different classification methods including Bagged Trees, Support vector machine (SVM), K-Nearest Neighbor (KNN), Fine Tree classification methods. Classification is a category of supervised machine learning in which an algorithm can learn to classify new observations from the examples with labeled data. To explore classification models, Classification Learner app are used. For greater flexibility, can be tendered predictor or feature data with corresponding labels or responses to an algorithm-fitting function.

1.5 THESIS OUTLINE

The paper is divided into five sections and arranged as follows, in chapter 1 it provides a detailed discussion on the importance of the work that has been done and why the current topic is selected for thesis. In the next that is chapter 2 is about some related works of this thesis is described in this section. It gives the information about the contribution of several researches in this field. Next is chapter 3 which introduces the dataset and the methods and explain deep understanding about the algorithms and evaluating criteria. Then the chapter 4 describes experimental plans and prediction test results on evaluation criteria. Finally, in Chapter 5, paper will end up as allowing to know why this study is a valuable for effective techniques in medical diagnosis in the conclusion.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

The liver is the largest organ of the body and it is essential for digesting food and releasing the toxic element of the body. Liver disease is one of the main health problems in the world. The main goal behind the Healthcare System is to provide the best quality of services and to predict the diseases at an early stage. There is a great need for an early detection of liver disease so as to prevent complete liver failure, which can result in patient's death. Classification algorithms are frequently used in different medical procedures. Classification algorithms are very much appropriate and used in different automated medical analysis tools. If look at the related work regarding in this issue, almost all of the studies mentioned below have been carried out based on the medical data set. Classification algorithms were implemented to the data as part of these studies and in the other part of the study, other machine approaches of learning were used.

2.2 RELATED WORKS

P.Rajeswari et al. has proposed the data classification is based on liver disease [1]. Algorithms are developed by collecting data from UCI repository containing of 345 instances with 7 different attributes. This paper deals with results in the field of data classification obtained with Naïve Bayes algorithms. FT tree algorithms, and K-Star algorithms and on the whole performance made know FT Tree algorithm when tested on liver disease datasets. Time taken to run the data is fast when compare to other algorithm with accuracy of 97.10%. Based on the experimental results, is found better classification accuracy of FT Tree algorithm compare to other algorithms.

For successful prediction accuracy of liver patients in three parts, Anju Gulia et al. has proposed to implement hybrid model construction and comparative analysis [2]. In first part of research, classification algorithms are applied on the original liver patient datasets

which is collected from UCI repository. In second part, by the use of feature selection, a subset of liver patient dataset from whole liver patient datasets is obtained which includes only significant attributes and then applying this on the selected classification algorithms. SVM technique is measured as the better performance algorithm, as it gives higher accuracy in respective to other algorithms before applying feature selection. But after applying feature selection Random Forest algorithm is given the better performance algorithm. In third part, the results of classification algorithms with and without feature selection are compared with each other. The results obtained from experiments point out that Random Forest algorithm overtook all other techniques with the help of feature selection with an accuracy of 71.8696%.

A.S. Aneesh Kumar et al. has defined the categorization of liver disorder over feature selection and fuzzy K-means classification [3]. Here for data analysis MATLAB was used. Different types of liver disorders use same attribute values but it need more effort to classify liver disorder type appropriately with basic attributes. For each type of liver disorder Fuzzy based classification technique gives better performance and achieved above 94% accuracy.

Onwodi Gregory has recommended two real liver patient datasets were considered for building classification models in order to predict liver disease [4]. Here about eleven data mining classification algorithms were applied in the datasets. Based on the experimental outcomes the classification accuracy is found to be better by using FT algorithm compare to other algorithms. It also shows the greater performance and it gives 78.0% of Accuracy, 77.5% of Precision, 86.4% of Sensitivity and 38.2% of Specificity results respectively.

Ebenezer Obaloluwa Olaniyi et al. has proposed back propagation neural network and radial basis function neural network are considered to diagnose these diseases and also avoid misdiagnosis of the liver disorder patients [5]. The algorithms were compared with the c4.5, CART, Naïve Bayes, Support Vector Machine (SVM) and concluded that the radial basis function neural network is the optimal model because it has a recognition rate of 70% which has proved more accurate and efficient than the other algorithms.

Tapas Ranjan Baitharua et al. has focused on the part of Medical diagnosis by learning process through the collected data of Liver disease to develop intelligent medical decision support systems which will help the physicians [6]. In this paper the use of several classifications (J.48, SVM, Random Forest) algorithms to classify these diseases and compare the effectiveness, correction rate among them. In this paper, a comparative

analysis of data classification accuracy using Liver disorder data in different circumstances is obtainable. The predictive performances of common classifiers are compared quantitatively. By analyzing the results Multilayer perceptron gives the overall best classification result with the accuracy 71.59% than other classifiers.

Bendi Venkata Ramana et al. has used the classification algorithms considered here are Naïve Bayes classifier, C4.5, Back propagation Neural Network algorithm, and Support Vector Machines [7]. These algorithms are evaluated based on four criteria: Accuracy, Precision, Sensitivity and Specificity.

S. Karthik et al. used in first part ANN classification which is applied for classifying the liver disease. In second part rough set rule induction is used and LEM (Learn by Example) algorithm is applied to produce classification rules. In third part, fuzzy rules are applied for identifying the types of the liver disease [8].

Otoom et al. proposed a system to detect and monitor coronary artery disease. Two tests are applied and three algorithms which are Bayes Net, Support vector machine, and Functional Trees FT were used [9]. In this paper WEKA tool used for prediction process. In this paper 7 selected features were used and Bayes Net gave 84.5% of accuracy, SVM gave 85.1% accuracy and FT classified 84.5% correctly.

Dr.S.Vijayarani et al., has proposed description of this research work is to predict liver diseases using classification algorithms [10]. The algorithms used in this paper are Naïve Bayes and support vector machine (SVM). Comparisons of these algorithms are done by based on the performance factors like classification accuracy and execution time. From the results, this paper determines the SVM classifier is considered as a best classification algorithm because it give highest classification accuracy. On the other hand, while comparing with the execution time, the Naïve Bayes classifier gives minimum execution time from the implementation where the SVM is a better Classifier for predict the liver diseases and comparing the execution time, the Naïve Bayes classifier needs lowest execution time.

Iyer et al. conducted an experiment to predict diabetes disease with the help of decision tree and Naive Bayes [11]. Different tests were carried out using WEKA data mining tool. Naive Bayes showed 79.5652% accuracy by using percentage split test. Maximum accuracy of algorithm was achieved by using percentage split test.

Weng et al. used an Artificial Neural Network (ANN) model to construct three type of classifiers the Individual Classifier (IC), Ensemble Classifier (EC), and Solo Classifier

(SC) [12]. Also, they applied the Liver Patient Dataset (ILPD) for their proposed method. The result showed that when k ¼3, accuracy is 68.49%, and the classifiers were EC–IC-2, EC–IC-Best.

Jin et al. used the Indian liver patient dataset from the UCI machine learning repository with three models - Decision Tree, K-Nearest neighbor, and Logistic [13]. The accuracy for Decision tree was 69.40, with the other models being lower.

Ramana et al. in another study, for liver disease classification, a Modified Rotation Forest model is presented so that it has a multi-layer perception (MLP) model and Random Subset feature selection technique for the UCI dataset. By applying MLP with the random subset method, accuracy was given 74.78% on the UCI liver dataset. Also, the accuracy of the KStar model showed 73.07 with Correlation based feature selection (CFS) on the India liver disease dataset [14].

Dhamodharn evaluated the data mining techniques for liver disease disorder [15]. He particularly compared two models, that is FT growth and Naïve Bayes. He observed that Naïve Bayes (75.54%) outperforms in accuracy as compared with the FT growth model (72.66%). This comparison was carried out on 29 datasets with 12 different features.

Alfisahrin et al. have projected the NBTree algorithm, generated by combining Decision Tree and Naïve Bayes algorithms [16]. The result of accuracy of the NB Tree algorithm was 67.01%, so that the accuracy of Decision Tree and Naïve Bayes were 66.14% and 56.14%, respectively. However, The Naïve Bayes algorithm has the fastest runtime then the other algorithms. Here a ranking method was applied to determine the features for UCI dataset.

Bendi Venkata Ramana et al. modified rotation forest algorithm was proposed with multilayer perception classification algorithm and random subset feature selection method for UCI liver data set [17].

A. S. Aneeshkumar et al. showed an approach for liver disorder by using data mining techniques [18]. Achieved better accuracy for diagnosis of liver disorder by using Fuzzy based classification.

D. Sindhuja et al. described classification techniques for considering liver disorder [19]. He also showed advantages and disadvantages of algorithms such as C 4.5, Naïve Bayes, Decision Tree, Support Vector Machine, Back propagation and Classification and Regression Tree and they are compared. C 4.5 gives better performance than other algorithms.

According to V. Shankar Sowmein et al. decision tree algorithm is used on the study diagnosis of Hepatitis. C 4.5 decision tree algorithm provides an efficient result in liver disease prediction [20].

Ba-Alwi et al. reported a comparative analysis on hepatitis disease diagnosis [21]. Naive Bayes, Naive Bayes updatable, FT Tree, KStar, J48, LMT (Logistic Model Tree) and NN (Neural Network) algorithms were used. Classification results were measured in terms of accuracy and time. Analysis was done by neural connections and WEKA. Compared to algorithms, results obtained from neural connection were lower used in WEKA. For medical data analysis it performed better than NN. Naive Bayes gave the accuracy of 96.52%, Naive Bayes updateable algorithm showed 84% exactness, FT Tree presented 87.10% correctness, K star exhibited 83.47% accuracy, J48 achieved 83% accurateness, LMT provided 83.6% accuracy and NN showed 70.41% of accuracy. In this paper Naive Bayes was the best classification algorithm. It offered highest accuracy in lowest time.

Young et al. used three types classification algorithms with UCI repository dataset with 11 attributes. By applying Logistic Regression, Decision tree, Neural Network they got highest accuracy in Neural Network 72.09% [22].

Hiroshi Nakano et al. applied 31 instances and 5 attributes based Laboratory dataset with Neural Network and found accuracy Neural Networks 78% [23].

Comak et al. showed an approach for liver disorder in their paper. SVM hybridizing, least square support vector machine (LSSVM), Fuzzy Weighting Pre-processing hybridizing method got accuracy 94.29% [24].

Mohaimenul et al. evaluated the classification techniques for liver disease disorder with 994samples and 10 attributes dataset from Taipei Medical University Hospital under a liver protection project. Here used Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Logistic Regression and finally got accuracy Logistic Regression 76.30% [25].

Mehdi et al. in their paper Prediction and diagnosis of [26]non-alcoholic fatty liver disease (NAFLD) and identification of its associated factors using the classification tree method applied Classification Tree (CT) in dataset of 1,600 individuals and 30 attributes in Kavar, a town in the south of Fars province, Iran and accuracy gave Classification tree (CT) was 80% [26].

Chieh-Chenet al.in their paper Prediction of fatty liver disease using machine learning algorithms used dataset of New Taipei City Hospital, Taiwan. Here applied Random Forest

(RF), Naïve Bayes (NB), artificial neural networks (ANN), and logistic regression (LR). Finally got accuracy Random Forest (RF) 87.48% [27].

Bahramirad et al. applied eleven data mining classification models for classification the data in both AP and BUPA datasets, whose calculation criteria were accuracy, precision, and recall [28]. The results are shown in some models; the performance on the AP dataset was slightly better than the BUPA dataset.

In paper, A Survey and Compare the Performance of IBM SPSS Modeler and Rapid Miner Software for Predicting Liver Disease by Using Various Data Mining Algorithms author Moloud Abdar applied 583 instances with 11 attributes of UCI dataset in C5.0 and C4.5, Decision tree Neural Network by IBM SPSS Modeler and Rapid Miner tools had 72.37% C4.5 and 87.91% C5.0 accuracy [29].

Jankisharan et al. in their paper Liver Patient Classification using Intelligence Techniques, used 583 instances and 11 attributes based UCI dataset in Multilayer, Support Vector Machine, Linear Regression, Multiple, Feed Forward, Neural Network, J-48, Random Forest and Genetic Programming. Finally got highest accuracy in Random Forest 84.75% [30].

2.3 TABLE OF PREVIOUS WORK

In this section the previous works related to this research is summarized by a table.

Table 2.1: Table of Previous Work

Sl No.	Title	Author	Year	Considered Parameters	Source of Data	Implemented Methods/ Techniques	Result
1	Analysis of Liver Disorder Using Data mining Algorithm	P.Rajeswari G.Sophia Reena	2010	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	UCI	Naïve Bayes, K Star, FT Tree	FT Tree 97.1%
2	Liver Patient Classification Using Intelligent Techniques	Anju Gulia, Dr. Rajan Vohra, Praveen Rani	2014	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot,	ILPD data set	J-48, Multilayer Perceptron, SVM, Random Forest, Bayesian	Random Forest 71.8696%

				TP, ALB , A/G Ratio, Patients			
3	A novel approach for Liver disorder Classification using Data Mining Technique	A.S. Aneeshkumar, Dr. C.Jothi	2015	6078 samples 48 attributes	Southern region of Tamil Nadu	Snake peel enabled Hybrid Ant colony optimization and Genetic Algorithm Fuzzy K-means (Unsupervised)	Fuzzy K-means 94 %
4	Prediction of Liver Disease (Biliary Cirrhosis) Using Data Mining Technique	Onwodi Gregory	2015	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	11classification	FT 78%
5	Liver Disease Diagnosis Based on Neural Networks	Ebenezer Obaloluwa Olaniyi, Khashman Adnan	2015	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	ILPD data set	Back propagation NN and Radial basis function NN (feed forward)	Radial basis function neural network 70%
6	Analysis of Data Mining Techniques For Healthcare Decision Support System Using Liver Disorder Dataset	Tapas Ranjan Baitharu, Subhendu Kumar Pani	2016	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	UCI	J48, Naive Bayes, MLP, ANN, ZeroR, 1BK and VFI	MLP 71.59%
7	A Critical Study of Selected Classification Algorithms for Liver Disease Diagnosis	Bendi Venkata Ramana, Prof. M.Surendra Prasad Babu, Prof. N. B. Venkateswarlu	2012	751 samples 12 attributes & 345 samples 7 attributes	UCI	Naïve Bayes, C4.5, Back propagation Neural Network, SVM	Back Propagation 71.59%
8	Classification and Rule Extraction using	S. Karthik, A.	2011	Mcv, Alkphos,	UCI	Naïve Bayes,	MLP 67.48%

	Rough Set for Diagnosis of Liver Disease and its Types	Priyadarishini and J. Anuradha and B. K. Tripathy		SGPT, SGOT, Gammagt, patient		MLP, RBF	
9	Datamining techniques for optimization of liver disease classification	T. M. Sa'sdiyah Noor Novita Alfisahrin	2013	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	Decision Tree, Naive Bayes, and NB Tree	NB Tree
10	Liver Disease Prediction using SVM and Naïve Bayes Algorithms	Dr.S.Vijaya rani, Mr.S.Dhaya nand	2015	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	ILPD data set	Naïve Bayes SVM	SVM 79.66%
11	Diagnosis of diabetes using classification mining techniques	A.Iyer S.Jeyalatha R. Sumbaly	2015	768 instances 9 attributes	UCI	Decision tree Naive Bayes	Naive Bayes 79.5652%
12	Disease prediction with different types of neural network classifiers	CH. Weng, TCK. Huang, RP. Han	2016	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	Liver Patient Dataset (ILPD)	(ANN) Classifier(IC), Ensemble Classifier(EC), and Solo Classifier (SC)	Ensemble Classifier 68.49%
13	Decision factors on effective liver patient data prediction	H. Jin, S. Kim, J. Kim	2014	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	Andhra Pradesh area located in north-east of India	Decision Tree, K-Nearest neighborLogistic	DT 69.40%

14	Liver classification using modified rotation forest	B.V.Raman a, M.S.P. Babu, N.B.Venkat eswarlu	2012	Age,Gende r,TB, DB,Alkpho s Alkaline Phosphotas e, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	India liver disease dataset	Modified Rotation Forest model KStar model MLP	MLP 74.78%
15	Liver disease prediction using bayesian classification	S. Dhamodhar n	2014	12 different features. 54instances	UCI	FT growth and Naïve Bayes	Naïve Bayes 75.54
16	Datamining techniques for optimization of liver disease classification	S.N.N. Alfisahrin, T. Mantoro	2013	Age,Gende r,TB, DB,Alkpho s Alkaline Phosphotas e, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI dataset.	NBTree Decision Tree Naïve Bayes	NB Tree 67.01%
17	A critical study of selected classification algorithms for liver disease diagnosis	B. Venkata Ramana Prof.M.Sure ndra Prasad Babu Prof.N.B. Venkateswa rlu	2011	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	AP liver dataset and UCLA dataset	Back Propagation, K-Nearest Neighbor	K-NN 97.47%
18	A novel approach for liver disorder classification using datamining techniques	A.S.Aneesh kumar Dr.C.Jothi Venkateswa ran	2015	48 attributes 6078 instances	Souther n region of Tamil Nadu	Genetic Algorithm Fuzzy K-means	Fuzzy K-means 94%
19	A survey on classification techniques in data mining for analyzing liver disease disorder	D. Sindhuia, R. Jenina Priyadarshin i	2016	Age,Gende r,TB, DB,Alkpho s Alkaline Phosphotas e, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	India liver disease dataset	C4.5 Naïve Bayes, Decision Tree, SVM, NN, R Tree	C 4.5 83%

20	Diagnosis of hepatitis using decision tree algorithm	VS Sowmien, V Sugumaran, CP Karthikeyan	2016	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	Decision tree	C 4.5 decision tree
21	Comparative study for analysis the prognostic in hepatitis data: datamining approach	F. M. Ba-Alwi, H. M. Hintaya	2013	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	India liver disease dataset	Naive Bayes, Naive Bayes updatable, FT Tree, KStar, J48, Logistic Model Tree, NN	Naive Bayes 96.52%,
22	Screening test data analysis for liver disease prediction model using growth curve.	Young Sun Kim, So Young Sohn, Chang No Yoon	2003	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	LR, Decision tree Neural Network	Neural Network 72.09%.
23	Application of neural network to the interpretation of laboratory data for the diagnosis of two forms of chronic active hepatitis	Hiroshi Nakano, Yasuyuki Okamoto, Hitomi Nakabayashi, S.Takamatsu, H.Tsujii	1996	31instances 5 attributes	Laboratory data	Neural Network	Neural Networks 78%
24	A new medical decision making system: least square support vector machine (LSSVM) with fuzzy weighting pre-processing	Comak E, Polat K, Gunes¸ S, Arslan A	2007	Mcv, Alkphos, SGPT, SGOT, Gammagt, patient	Liver Disorder dataset	SVM hybridizing least square support vector machine (LSSVM) Fuzzy Weighting	hybridizing method 94.29%

25	Applications of Machine Learning in Fatty Live Disease Prediction	Md.Mohaimenul Islam, Chieh-ChenWu, Tahmina Nasrin Poly, HsuanChia Yang, Yu-Chuan (Jack) Li	2018	994samples 10 attributes	Taipei Medical University Hospital under a liver protection project	Pre-processing Random Forest, Support Vector Machine (SVM), Artificial Neural Network (ANN), and Logistic Regression	Logistic Regression 76.30%,
26	Prediction and diagnosis of non-alcoholic fatty liver disease (NAFLD) and identification of its associated factors using the classification tree method	Mehdi Birjandi Seyyed, Mohammad Taghi Ayatollahi, Saeedeh Pourahmad Ali Reza Safarpour	2016	1,600 individuals 30 attributes	Kavar, a town in the south of Fars province, Iran	Classification Tree (CT)	Classification tree (CT)80 %
27	Prediction of fatty liver disease using machine learning algorithms	Chieh-Chen Wua, Wen-Chun Yehb , Wen-Ding Hsuc , Md. Mohaimenul Islama , Phung Anh (Alex) Nguyene , Tahmina Nasrin Polya, Yao-Chin Wanga, Hsuan-Chia Yange , Yu-Chuan (Jack) Li	2019	Abdomina_ gridle GPT_ALT, triglyceride , HDL_C, Glucose_A C, Systolic, Gender, Diastolic, GOT_AST, Age	New Taipei City Hospital, Taiwan	Random Forest (RF), Naïve Bayes (NB), artificial neural networks (ANN), and logistic regression (LR)	Random Forest (RF) 87.48%
28	Classification of Liver Disease Diagnosis: A Comparative Study	Sina Bahramirad, Aida Mustapha, Maryam Eshraghi	2013	583 instances with 13 attributes 354 instances	(UCI) AP dataset BUPA dataset	Eleven Classification Algorithms	Logistic 73.39% Neural Net 73.91%

				with 7 attributes			
29	A Survey and Compare the Performance of IBM SPSS Modeler and Rapid Miner Software for Predicting Liver Disease by Using Various Data Mining Algorithms	Moloud Abdar	2015	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	C5.0 and C4.5, Decision tree Neural Network	IBM SPSS Modeler and Rapid Miner tools had 72.37% C4.5 and 87.91% C5.0
30	Liver Patient Classification using Intelligence Techniques	Jankisharan Pahareeya, Rajan Vohra, Jagdish Makhijani, Sanjay Patsariya	2014	Age,Gender,TB, DB,Alkphos Alkaline Phosphotase, Sgpt, Sgot, TP, ALB , A/G Ratio, Patients	UCI	Multilayer, Support Vector Machine, Linear Regression, Multiple, Feed Forward, Neural Network, J-48, Random Forest, Genetic Programming	Random Forest 84.75%

2.4 SUMMARY

In the privies section is about the reviews of the previous related works. All the related paper work on the Liver disease dataset. Some researcher is used online based dataset and others used survey dataset. Overall used attributes are Age, Gender, TB, DB, Alkphos, Alkaline, Phosphotase, Sgpt, Sgot, TP, ALB, A/G Ratio, Mcv, Gammagt. Some researchers got satisfactory result but some are not. Still no one can get hundred percent accuracy. So that research on this topic is still continued and become a hot topic for new era. For this reason, this paper used the commonly used attributes Age, Gender, TP, DB, Sgpt, Sgot, A/G Ratio for the Liver patients. In this research used real time dataset which is unique and collected from the different region of the country. Selected are techniques Bagged Tree, SVM, Fine tree, KNN.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

An early prediction is one of the most important steps in liver disease treatment as it is capable of maintaining normal function even when partially damaged. Therefore, the project of an effective diagnosis model is a significant issue in liver disease treatment. In this chapter, different machine learning classification algorithms and their implementations that have been used for predicting liver disease are discussed in detail and the whole research process is shown step by step. First, in the section 3.2 shows the work flow diagram of the process. In this section, more details about our proposed model have been presented. In section 3.3 data collection and processing that includes which are 3.3.1 dataset description, 3.3.2 data collection, 3.3.3 data processing. the whole process of the data collection and processing is discussed in this sections. The next section is 3.4 methods for disease prediction. this gives more information about all the methods which is used in this research in details. The whole process of the implementation has been expressed in the 3.5 implementation section which includes more sub sections they are 3.5.1 training and testing, 3.5.2 validation, 3.5.3 comparison, 3.5.4 tool. The finally the section 3.6 model evaluation or performance analysis which includes sub sections 3.5.1 confusion matrix, 3.5.2 auc – roc curve, 3.5.3 error and 3.5.4 accuracy.

3.2 WORK FLOW DIAGRAM

The liver diseases patient records are collected that preprocess and extract some features for analyzing further manipulation of it. Then, have used four different classification algorithms to measure the performance of liver disease. Those steps are described briefly. This research of analyzing liver disease is summarized by the following figure 3.1 given below. It starts with the collection of raw data like collection of the patient detail. The data of the patients is then being analyzed. Data was normalized so that can get

better result. After preparing the training dataset the different classification techniques is applied. This study used four techniques. The obtained results from them were then be compared. The selected technique gives the highest accuracy. A schematic view of flowchart related to proposed methodology in present work can be observed in Fig. 3.1

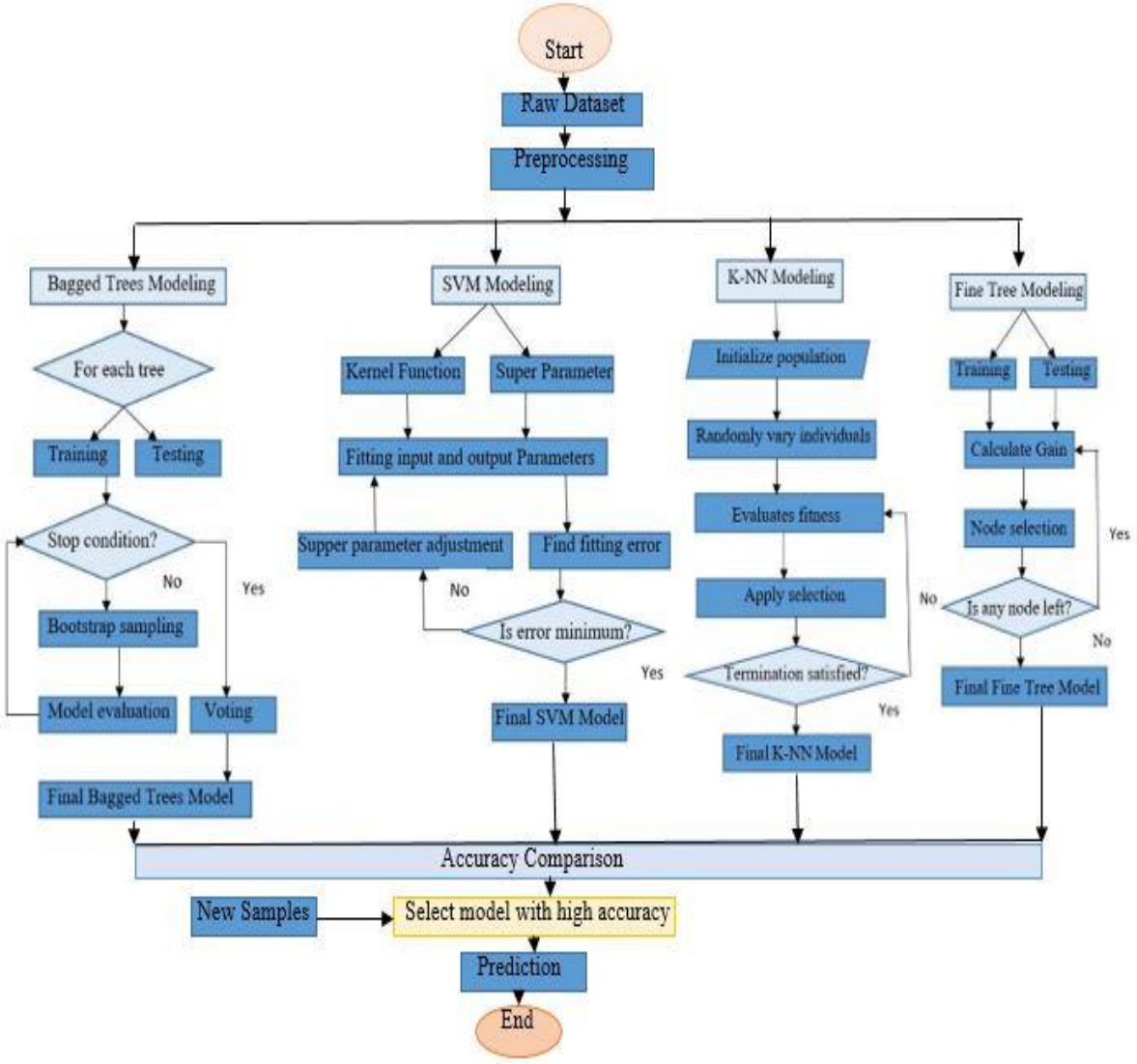


Figure 3.1: Work flow diagram

3.3 DATA COLLECTION AND DATA PROCESSING

Machine learning involves extracting and analyzing the medical data in building certain prediction model to increase the accuracy of diagnosis in any specific disease. However, only few works in this sector investigate liver disorders, although this disease is

increasing day by day and becoming one of the deadliest diseases in some countries. Various machine learning algorithms have been discovered for liver disease diagnostic classification. In classification model development, first step is data collection and processing.

3.3.1 Dataset Description

For the proper diagnosis, it is necessary to evaluate some of the main attributes of liver patient's dataset. Accurate predict of the liver disease is a challenging task for doctors. Different classification techniques are used to classify the data and predict the liver disease by using the datasets of liver patients. Some of the core attributes of liver disease include- Total bilirubin (TB), direct bilirubin (DB), alkaline phosphotas (Alphs), total protein (TP), albumin and globulin ratio (A/G). Age**,** gender, total bilirubin, direct bilirubin, Alanin Aminotransferase, Aspartate Aminotransferase, Albumin/Globulin ratio these attributes use for primary Liver disease diagnoses which is suggested by the doctors. For this reason, these seven main attributes are used in this research. The used dataset contains Liver function test details. Total bilirubin (TB) is a combination of direct and indirect bilirubin. It results for direct and total bilirubin. Normal results for a total bilirubin test are 1.3 mg per deciliter (mg/dL) for adults and usually 1 mg/dL for those under 18. Normal results for direct bilirubin are generally 0.3 mg/dL. Alanine transaminase or (SGPT) is an enzyme. It is found in plasma and in various body tissues but is most common in the liver. It catalyzes the two parts of the alanine cycle. Aspartate Aminotransferase or (SGOT) is an important enzyme in amino acid metabolism. This blood test is used to detect liver damage. The normal range is 10 to 34 U/L. After a liver function test, the amount of albumin is divided by the amount of globulin to get the Albumin/Globulin ratio. A ratio ranging 1.70 to 2.2 is the ideal. Table 3.1 shows a brief description of the dataset that is being considered.

Table 3.1: Dataset description

Attribute Name	Description
Age	young (for age 0-20), middle (for age 21-40) and old (for age 41-above).
Gender	male female

TB (total bilirubin)	tb_normal (less than 0.3), tb_high (from 0.3 to 1.9 mg/dl)
DB (direct bilirubin)	low (less than zero), normal (0 to 0.3 mg/dl) and high (above 0.3 mg/dl)
Sgpt (Alanin Aminotransferase)	for male, sgpt_low (less than 10 U/L), sgpt_normal (between 10 to 40 U/L) and sgpt_high (above 40 U/L), for female, sgpt_low (less than 7 U/L), sgpt_normal (between 7 to 35 U/L) and sgpt_high (above 35 U/L).
Sgot (Aspartate Aminotransferase)	for male, sgot_low (below 14 U/L), sgot_normal (between 14-20 U/L) and sgot_high (above 20 U/L), for female, sgot_low (below 10 U/L), sgot_normal (between 10-36 U/L) and sgot_high (above 36 U/L).
A/G (Albumin/Globulin ratio)	ag_low (below 1.5 g/dl), ag_normal (between 1.5 to 3.5 g/dl) and ag_high (above 3.5 g/dl).
Disease	Class 1 (Patient have liver disease) and Class 0 (Patient have no liver disease)

3.3.2 Data Collection

Data of liver patients is collected from hospitals, diagnostic center and clinic center. It is a tough task as there are some limitations of collecting data. The data for this proposed research is be collected from the following zones: Dhaka, Cumilla, Noakhali, Jamalpur etc. All together data are gathered around 304 samples both male and female. In the dataset, 203 data samples are liver disease patients and 101 data samples are healthy patients.

3.3.3 Data Processing

Only a small data pre-processing is required. First of all, after taking the all the collected raw data samples are included in the excel file. To handle the missing values and set the average value for replacement of those missing values. Next, used the Remove Duplicate records operator in order to avoid any duplicate record in our datasets. Then in all the column apply the normalized rules. The rule is given bellow.

$$x_{\text{normalized}} = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

All supervised learning methods start with an input data matrix, called X here. Each tuple represents one observation. Each column represents one variable or. Represent missing entries with average value of X. Statistics and Machine Learning Toolbox supervised learning algorithms can handle missing values, either by ignoring them or by ignoring any row. For classification, Y can be any data types, here used numeric type- 0 or 1 for the response.

3.4 METHODS FOR DISEASE PREDICTION

This paper based on the supervised classification algorithms. Classification constructed on categorical that is discrete and unordered. This technique based on the supervised learning that is targeted output for a given input is known from the previous. Classification is one of the most popularly used methods in Healthcare sector. It splits data samples into target classes. The classification technique predicts the target class for each data sample. Classification approach used to find out a risk factor can be related to patients by analyzing their patterns of diseases. It is a supervised learning approach which have known class labels. Binary and multilevel are the two methods of classification. In binary classification, there are two possible classes such as, “high risk” or “low risk” patient may be considered and for the multiclass approach has more than two targets for example, “high risk”, “medium risk” and “low risk” patient. Data set is separated as training and testing dataset. It provides predicting a certain outcome based on a given input. Training the dataset is the process which consists of a set of attributes and predict the outcome. In order to predict the outcome, it attempts to discover the relationship between attributes. Its goal is to prediction the outcome. It consists of same set of attributes as that of training set. In order to process the prediction, it mostly analyses the input. Training dataset involves all the information about the patients which were recorded previously, whether a patient had a Liver disease or not is the prediction attribute there. Classification goals to categorize unseen input data samples into known classes. The model or classifier learns from the training data set where the relationship between records and classes is given.

This paper work is implemented by building classifier models in software. The objective of selecting this software is to understand the basic concepts and also application of the algorithms in real time. It is helpful in learning the basic concepts of machine learning with different options and analyzes the output that is being produced. It is a well-known and respected data analysis software environment and programming language. Four different classification algorithms are used to calculate the performance for the desired class. For this, performed cross validation with 10-Folds on full categorized dataset. Here selected algorithms are: Bagged trees, Support Vector Machine (SVM), K-Nearest Neighbor, Fine Tree.

3.4.1 Bagged Trees

Bagging is an abbreviation of bootstrap aggregating, with bootstrap or sampling technique on the original data n times with replacement to create training sets [31]. It is a machine learning ensemble algorithm designed to improve the stability and accuracy of machine learning algorithms used in statistical classification and regression. It also reduces variance and helps to avoid overfitting. Although it is usually applied to decision tree methods, it can be used with any type of method. Bagging is a special case of the model averaging approach. Bagging uses a simple approach that shows up in statistical analyses again and again improve the estimate of one by combining the estimates of many. Bagging constructs n classification trees using bootstrap sampling of the training data then each training data is made a classification tree and the aggregate process or the majority vote for the classification case and the average for regression case and then combines their predictions to produce a final meta-prediction. Bagging is an ensemble method commonly applied to classification cases, with the aim of improving classification accuracy by combining single classifications, and the results are better than random sampling [32]. Bagging was able to reduce the classification error rate in the classification case with 50 repetitions for classification cases and 25 times for regression cases [33].

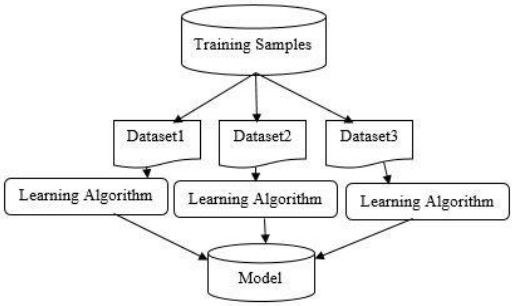


Figure 3.2: Bagged Trees classification algorithm structure

Assume a training data set S that contain T examples. Bootstrap sampling makes a sample of training examples Si, by choosing m examples consistently at arbitrary with replacement from S. The replacement implies that examples may be repeated in Si. Then T bootstrap samples are created and T classifier is trained by bagging on each bootstrap sample. With the help of weighted majority of the T learned classifiers a new instance is classified. The result is an ensemble of classifiers.

Algorithm

Input: training set S, Inducer Ғ, integer T (number of bootstrap samples)

i. For i = 1 to T {
ii. Si= bootstrap sample from S (sample with replacement).
iii. $C_i = Ғ(Si)$
iv. }
v. $C^*(x) = argmax_{y \epsilon Y} \sum_{i:Ci(x)=y} 1$ (the most often predicted label y)
vi. Output: classifier C^*

Code

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
```

```
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitensemble(predictors, response, 'Bag', 200, 'Tree', 'Type',
'Classification', 'PredictorNames', {'ageNormalized' 'gender0female1male'
'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized' 'sgotNormalized'},
'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1]);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

3.4.2 Support Vector Machin (SVM)

Support vector machine is a machine learning technique on the basis of statistical learning theory. It creates a discrete hyperplane in the descriptor space of the training data and compounds are classified based on the side of hyperplane located. The distance between a data point and the hyperplane can be calculated in a transformed feature space, lacking of the explicit transformation of the original descriptors. The radial basis function kernel (Gaussian kernel) which is the most commonly used. The kernel function is expressed as follows

$$K(x,xi) = \exp\left(-\frac{||x-xi||2}{2a2}\right) \quad 3.2$$

In the above equation, the kernel width parameters control the amplitude of the Gaussian function reflecting the generalization ability of SVM. The regularization parameter C is censurable for inhibiting transaction between maximizing the margin and minimizing the training error. SVMs have frequently been found to provide maximum classification accuracies than other widely used pattern recognition techniques, such as the maximum likelihood and the multilayer perceptron neural network classifiers. Mathematical formulation of support vector machine is given below [34].

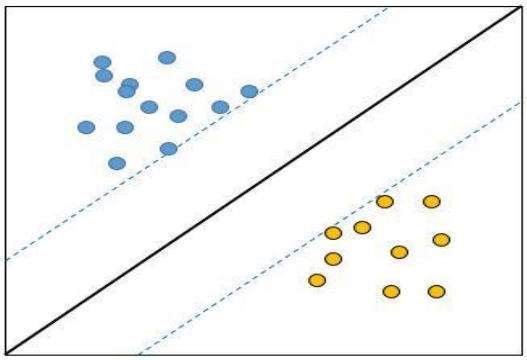


Figure 3.3: SVM classification algorithm structure

A support vector machine constructs a hyper plane or set of hyper planes in a high- or infinite-dimensional space. A good separation is achieved by the hyper plane that has the largest distance to the nearest training data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier.

Algorithm

Step 1: Let's assume a supervised binary classification problem. Let us consider that the training set consists of N vectors from the dimensional feature space.

$$x_i \in \Re^d \ (i = 1,2,...., N) \qquad 3.3$$

Step 2: A target $y_i \in \{-1,+1\}$ is associated to each vector x_i .

Step 3: Let us consider that the two classes are linearly separable. This points that it is possible to discovery at least one hyperplane (linear surface) defined by a vector $w \in \Re^d$ (normal to the hyperplane) and a bias $b \in \Re$ that could separate two classes without errors.

Step 4: The membership decision rule can be based on the function sgn [f(x)], where f(x) is the discriminant function associated with the hyperplane and defined as

$$f(x) = w.x + b \qquad 3.4$$

In case to find such a hyperplane, one should estimate w and so that

$$yi\ (w.xi + b) > 0 \text{ , with } i = 1,2,..., N \qquad 3.5$$

Step 5: The SVM approach involves in discovering the optimal hyperplane that increases the distance between the neighboring training sample and the splitting hyperplane. It is

possible to express this distance as equal to 1/ || w|| with a simple rescaling of the hyperplane parameters w and b such that

$$\min_{l=1,2,....N} yi(w.xi + b) \geq 1 \quad 3.6$$

Step 6: Consequently, it changes the optimal hyperplane which can be controlled by the following solution of convex quadratic programming problem

$$\text{Min} : \frac{1}{2} ||w||^2 \quad 3.7$$

Step 7: This traditionally linear constrained optimization problem can be interpreted (using a Lagrangian formulation) into the following dual problem

$$\text{Max: } \sum_{i=1}^{N} \alpha i - \frac{1}{2} \sum_{i=1}^{N} \quad \sum_{j=1}^{N} \alpha i \alpha j y i y j (xi.xj) \quad 3.8$$

Step 8: The Lagrange formulizers αi 's (i =1,2,...,N) represented in above can be assessed using quadratic programming (QP) methods. The discriminant function associated with the optimal hyperplane becomes an equation depending both on the Lagrange multipliers and on the training samples, i.e.

$$f(x) = \sum_{ies} \alpha i y i (xi.x) + b \quad 3.9$$

Where s is the subset of training samples corresponding to the nonzero Lagrange multiplier's. It is worth noting that the Lagrange multipliers effectively weight each training sample according to its importance in determining the discriminant function. The training samples associated to nonzero weights are called support vectors. These lie at a distance exactly equal to 1/ || w|| from the optimal separating hyperplane.

Code

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitcsvm(predictors, response, 'KernelFunction', 'linear',
'PolynomialOrder', [], 'KernelScale', 'auto', 'BoxConstraint', 1, 'Standardize', 1,
```

```
'PredictorNames', {'ageNormalized' 'gender0female1male' 'TBNormalized'
'DBNormalized' 'AGnormalized' 'sgptNormalized' 'sgotNormalized'}, 'ResponseName',
'is_patientyes1no0', 'ClassNames', [0 1]);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

3.4.3 K-Nearest Neighbor (K-NN)

K-NN is one of the nearest neighboring search method. The nearest neighboring search measures the degree of similarity between the case representing the current problem and all cases in the database which is the collection of past cases in order to search for similar cases of the current problem. After computing similarity, it finds instances higher than a certain threshold. Therefore, it is a method which does not intentionally find determined probability arguments (parameters) for each sample but classify it into the most similar in reference set or as belonging to the class closer in the distance by displaying the coordinate value for it directly. The model for KNN is the entire training dataset. When a prediction is required for an unseen data instance, the KNN algorithm will search through the training dataset for the k-most similar instances. The prediction attribute of the most similar instances is summarized and returned as the prediction for the unseen instance. The similarity measure is dependent on the type of data. For real-valued data, the Euclidean distance can be used. Other types of data such as categorical or binary data, hamming distance can be used. The KNN algorithm is belongs to the family of instance-based, competitive learning and lazy learning algorithms. Instance-based algorithms are those algorithms that model the problem using data in-stances (or rows) in order to make predictive decisions. The KNN algorithm is an extreme form of instance-based methods because all training observations are retained as part of the model. It is a competitive learning algorithm, because it internally uses competition between model elements (data instances) in order to make a predictive decision. The objective similarity measure between data instances causes each data instance to compete to win or be most similar to a given unseen data instance and contribute to a prediction.

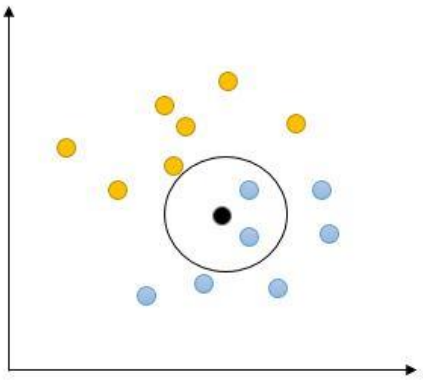


Figure 3.4: K-Nearest Neighbor algorithm structure

Algorithm

i. Determine the parameter K i.e., number of nearest neighbors beforehand.

ii. Calculate the distance between test data and each row of training data with the help of any of the method namely: euclidean distance measure algorithm.

   a. d(p,q) = $\sqrt{(p+q)^2}$

      3.10

iii. Distances for all the training samples are sorted and nearest neighbor based on the K-th minimum distance is determined.

iv. Since the K-NN is supervised learning, get all the Categories of your training data for the sorted value which fall under K.

v. The prediction value is measured by using the majority of nearest neighbors.

Code

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
```

```
trainedClassifier = fitcknn(predictors, response, 'PredictorNames', {'ageNormalized'
'gender0female1male' 'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized'
'sgotNormalized'}, 'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1], 'Distance',
'Euclidean', 'Exponent', '', 'NumNeighbors', 1, 'DistanceWeight', 'Equal',
'StandardizeData', 1);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

3.4.4 Fine Tree

Decision tree classifier is mainly used for data classification. Normally, it divided into two stages. One is the tree structure and another is tree pruning. The training data is used to generate a test function, according to different Classification based on decision tree classification method in comparison with the other, with faster and easy to understand classification rules and gives advantages especially in problem areas of high dimension, it can be very good classification results.

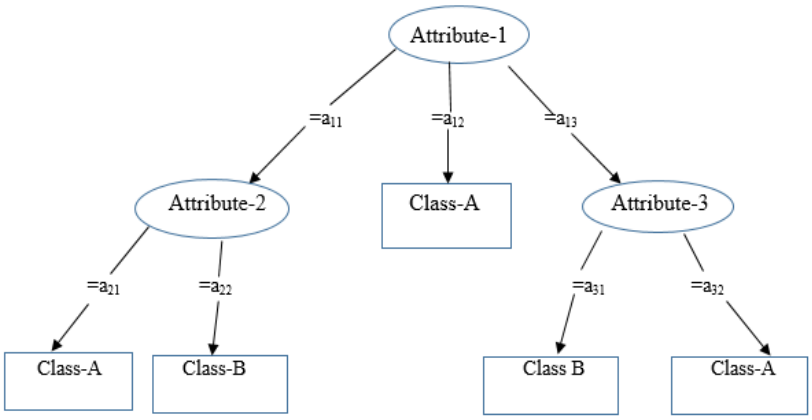


Figure 3.5: Fine Tree algorithm structure

The objective of the classifier is to maximize the Gain, dividing by overall entropy due to split argument by value j. Because of the outliers this is a significant step to the result. Some instances are present in all data sets which are not well defined and differ from the

other instances on its neighborhood. The classification is performed on the instances of the training set and tree is formed. The pruning is performed for decreasing classification errors which are being produced by specialization in the training set. Pruning is performed for the generalization of the tree.

Decision trees are easy to interpret and understand. It is fast for fitting and prediction and low on memory usage. But it can have low predictive accuracy. The depth can be controlled with the help of maximum number of splits setting. A fine tree with many leaves is usually highly accurate on the training dataset. Sometimes, the tree might not show comparable accuracy on an independent test set. If a leafy tree tends to over train, its validation accuracy is often far lower than its training accuracy. The Fine Tree model flexibility is high. Many leaves to make many fine distinctions between classes. In Fine tree maximum number of splits is 100.

Algorithm

Step 1: create a node N

Step 2: if samples are all of the same class, C then

Step 3: return N as a leaf node labeled with the class C;

Step 4: if attribute-list is empty then

Step 5: return N as a leaf node labeled with the most common class in samples;

Step 6: select test-attribute, the attribute among attribute-list with the highest information gain;

Step 7: label node N with test-attribute;

Step 8: for each known value ai of test-attribute

Step 9: grow a branch from node N for the condition test-attribute= ai;

Step 10: let s_i be the set of samples for which test-attribute= ai;

Step 11: if s_i is empty then

Step 12: attach a leaf labeled with the most common class in samples;

Step 13: else attach the node returned by generate decision tree (si, attribute-list, and test-attribute)

Counting Gain This process uses the “Entropy” which is a measure of the data disorder. The Entropy of is calculated by

$$Entropy(y) = \sum_{i=1}^{n} \frac{|yi|}{|y|} \log \left(\frac{|yi|}{|y|}\right) \qquad 3.11$$

$$Entropy(i|y) = \frac{|yi|}{|y|} \log \left(\frac{|yi|}{|y|}\right) \quad 3.12$$

And Gain is

$$Gain(y, i) = Entropy(y - Entropy(i|y)) \quad 3.13$$

Code

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitctree(predictors, response, 'PredictorNames', {'ageNormalized'
'gender0female1male' 'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized'
'sgotNormalized'}, 'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1],
'SplitCriterion', 'gdi', 'MaxNumSplits', 4, 'Surrogate', 'off');
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

3.5 TRAINING, TESTING AND VALIDATION

Classification is used to identify how data is being classified the exactly. Classification technique is a two phase process in which first step is the training phase where the classifier algorithm forms classifier model with the training set of rows and the second step is classification phase where the model is used for classification and its performance is examined with the testing set of rows. At first simply splits a dataset into training and test data and the secondly performs cross-validation using folds. The full dataset is divided in 10 portion, from the parts 1part is used for testing and rest of it used

for the training part. This process done for the full dataset in 10 times. Evaluation is usually described by the accuracy. In this research for checking the validation process used cross-validation. Cross-validation is a frequently used method for a model assessment technique. This is done by dividing a dataset and using a subset to train the algorithm and the remaining data for testing. Here used *k* (10) folds cross-validation. In each round of cross-validation involves randomly partitioning the main dataset into a training set and a testing set. The training set is then used to train a classification algorithm and the testing set is used to evaluate its performance. This process is repeated 10 times and the average cross-validation error is used for a performance indicator. classifier algorithm's. Cross-Validation steps are**:** Select a number of divisions to partition the dataset. Commonly choose *k* (10) folds, then:

i. Partitions the data into *k* disjoint sets or folds
ii. For each fold:
    a. Trains a model using the out-of-fold observations
    b. Assesses model performance using in-fold data
iii. Then calculates the average test error of all folds

This method provides a good estimate of the predictive accuracy of the model trained with all the dataset.

3.6 PERFORMANCE ANALYSIS

This paper carried out model development in order to assess the performance and usefulness of different classification algorithms for predicting Liver disease. After performing different classification algorithm and value of different term for each classification algorithm are listed in a table to measure and explore the performance on the classification methods. Comparison is based on different performance metrics like TP rate, FP rate, and Precision, Error, Recall and ROC area are for different classifiers. Model evaluation or Performance Analysis is based on confusion matrix and the all related measures like- ROC curve, True positives, True negatives, False positives, False negatives, Error rate (ERR), Accuracy, Sensitivity, Specificity, Precision etc.

3.6.1 Confusion matrix

Confusion matrix is one of the most common and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes. The confusion matrix is a table with two dimensions "Actual class" and "Predicted class" in both dimensions. Actual classifications are Rows and Predicted ones are columns. In the dataset there are two classes one is Class 1 and another is Class 0. A confusion matrix has created as follows:

Table 3.2: Confusion Matrix

		Predicted	
Actual		Negative(Class 0)	Positive(Class 1)
	Negative(Class 0)	TN	FP
	Positive(Class 1)	FN	TP

3.6.2 AUC – ROC curve

AUC - ROC is a graphical representation for performance measurement for classification problem. ROC is a probability curve and AUC represents degree or measure of area. It expresses how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

True Positives (TP): True positives are the cases when the actual class of the data point was True and the predicted is also True.

True Negatives (TN): True negatives are the cases when the actual class of the data point was False and the predicted is also false.

False Positives (FP): False positives are the cases when the actual class of the data point was False and the predicted is True.

False Negatives (FN): False negatives are the cases when the actual class of the data point was True and the predicted is False.

Sensitivity (Recall or True Positive Rate): Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR).

$$SN = \frac{TP}{TP + FN} = \frac{TP}{P} \quad 3.14$$

Specificity (True Negative Rate): Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR).

$$SP = \frac{TN}{TN + FP} = \frac{TN}{N} \quad 3.15$$

Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of the dataset. Accuracy comparison is based on the performance among the four classification algorithm's.

$$ACC = \frac{TP + TN}{TP+TN+FP+FN} = \frac{TP+TN}{P+N} \quad 3.16$$

3.6.3 Error

Error rate is one of the performance measurement parameter. Error rate (ERR) is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.0.

$$ERR = \frac{FP + FN}{TP + TN + FP + FN} = \frac{FP + FN}{P + N} \quad 3.19$$

3.6.4 Accuracy testing for new samples

After whole model development and performance analysis with different classification algorithm, the values of different models are investigated with the performance on the classification methods. Accuracy testing for new samples is done by using 10 numbers of new samples. These samples are completely new for the model. These will run on the model and then accuracy is testing by the values.

CHAPTER 4

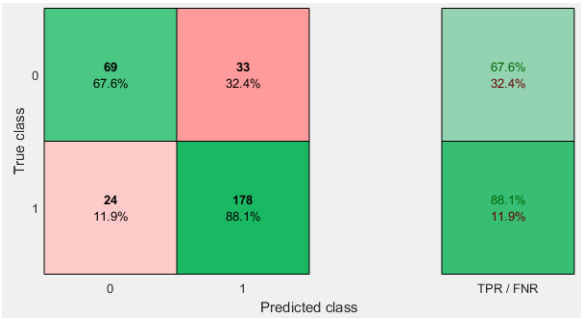
RESULTS AND DISCUSSION

4.1 INTRODUCTION

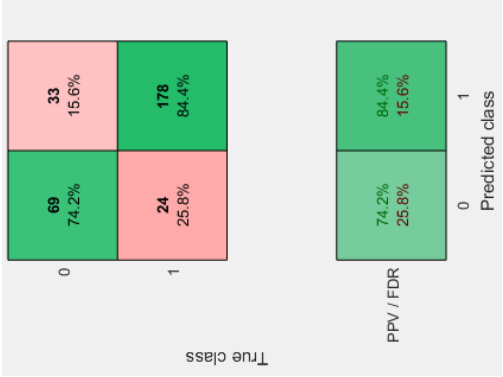
In this section, used different classification algorithm for the calculation of accuracy. Result produces from the all the selected algorithms are compared with result from the evaluation process. In this paper different classification algorithm work with SVM, K-NN, Bagging Trees and Fine Tree classifier algorithm. For this, first of all, have to load dataset file from directory. Then, import it to the software which will saved in the work place. The following figure shows importing Liver Patients Dataset.

4.1.1 Bagged Trees Classifier

For applying the Bagged Trees classifier on dataset, have to go to the App part of the software. After A 10-fold cross validation the resultant accuracy was obtained was 81.3%. The time taken to build cross-validation was 45.56 seconds. The following figure stands for confusion matrix by Bagged Trees Classifier where the green cell is shown where the predicted result is matched with the true class. And the red cell is shown where the predicted result is not matched with the true class.



(a) Confusion matrix for TPR/FNR



(b) Confusion matrix for PPV/FDR

Figure 4.1: Confusion matrix obtained after training SVM Classifier

This figure for the confusion matrix of the true positive rate (TPR) and false negative rate(FNR). In this figure the model uses dataset which is liver dataset, all rows classes stand for the true class. The columns show the predicted classes. In the top row, 67.6% of the class0 are correctly classified so 67.6% is the true positive rate for correctly classified points in this class and 32.4% are misclassified which is false negative rate. In the nest row, 88.1% of the class1 are correctly classified so 88.1% is the true positive rate for correctly classified points in this class and 11.9% are misclassified which is false negative rate. To see results per predicted class, under Plot, select the positive predictive values (PPV)**,** false discovery rates (FDR). The PPV is the proportion of correctly classified observations per predicted class. The FDR is the proportion of incorrectly classified observations per predicted class.

The confusion matrix generated from the 10-fold cross validation by using Bagged Trees classifier is shown in Table:

Table 4.1: Accuracy result of Bagged Trees classifier

Total no. of Instances	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Percentage of error	Percentage of Accuracy
Class '0'	102	69	33	32.4%	67.6%
Class '1'	201	178	24	11.9%	88.1%
Total	304	247	57	18.7%	81.3%

Form the above table get the values of confusion matrix for Bagged Trees where class 0 means negative value and class 1 means positive value. In the above table correctly classified instances are 247 and incorrectly classified instances are 57 which leads 81.3% and 18.7% respectively. Following curve shown in figure is for ROC curve for measuring performance.

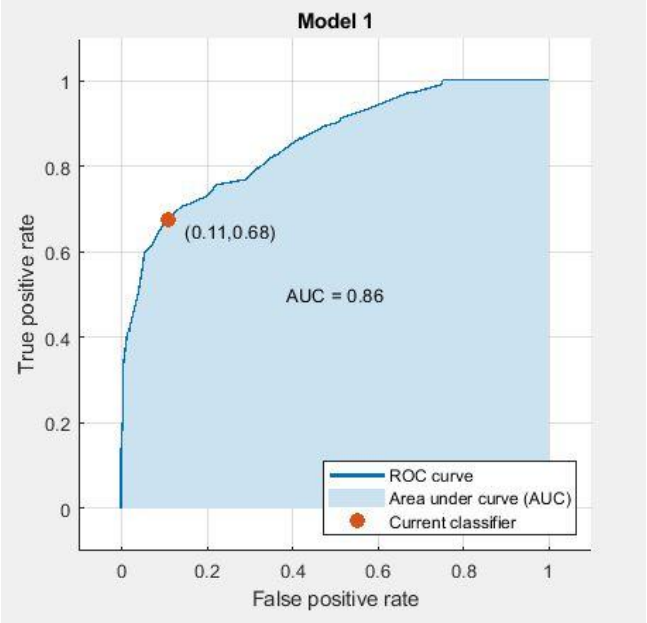


Figure 4.2: ROC curve for measuring performance of Bagged Trees

Fig shows Receiver Operating Characteristic (ROC) Curve of Bagged Trees classifier. The Y axis is stand for true positive value and X axis is for false positive value. Area under ROC(AUC) for both Class 0 and Class 1 is 0.86. The closer point of the true positive rate is (0.11,0.68).

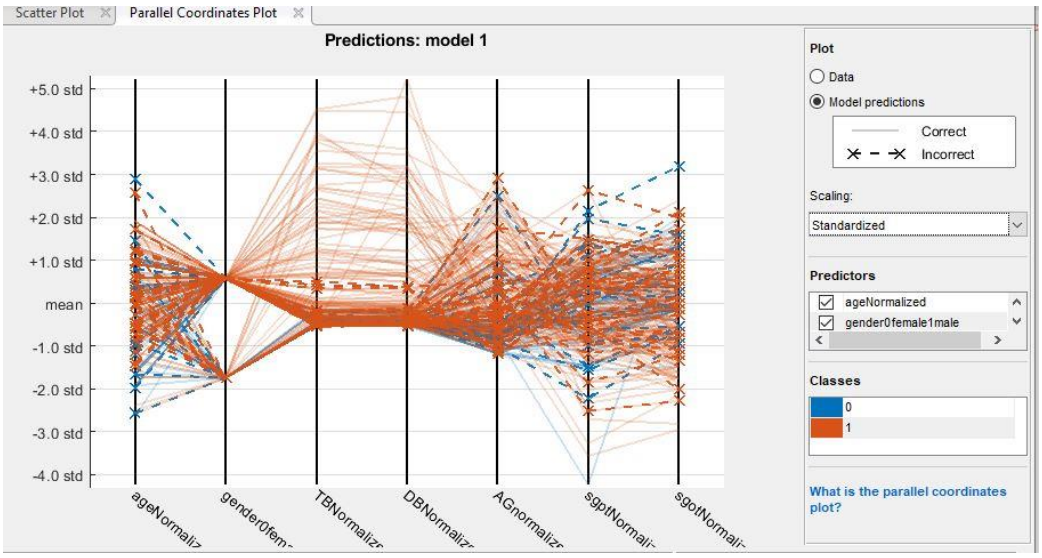
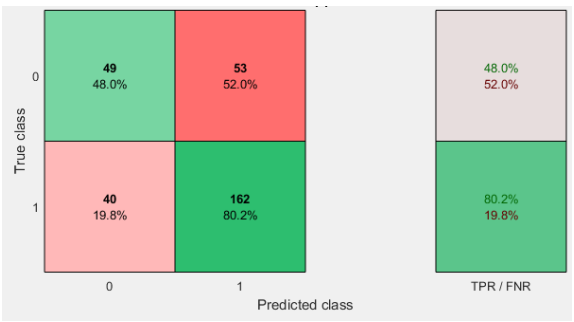


Figure 4.3: Parallel coordinates plot for Bagged Trees

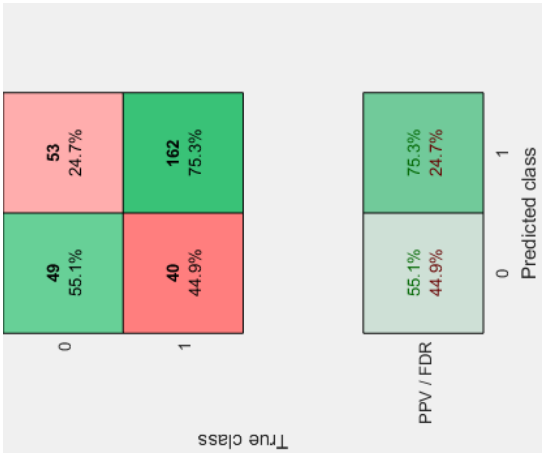
Parallel coordinates plot used to visualize high dimensional data, where each observation is represented by the sequence of its coordinate values plotted against their coordinate indices. The resulting plot contains one line for each observation. Here figure shows the parallel coordinates plot for Bagged Trees model. In the figure, red lines for the class 1 (positive value) and the blue lines shows the class 0 (negative value). Scaling is based on standardized values. The plot shows the median values for each group as a solid line and the quartile values as dotted lines of the same color. Here the blue line shows the median value measured for each variable on class0 and red line shows the median value measured for each variable on class1.

4.1.2 Support Vector Machine Classifier

For applying the SVM classifier on dataset by using MATLAB, have to go to the App part of the software. After A 10-fold cross validation the resultant accuracy was obtained was 69.4%. The time taken to build cross-validation was 26.09 seconds. The following figure stands for confusion matrix by SVM Classifier where the green cell is shown where the predicted result is matched with the true class. And the red cell is shown where the predicted result is not matched with the true class.



(b) Confusion matrix for TPR/FNR



(b) Confusion matrix for PPV/FDR

Figure 4.4: Confusion matrix obtained after training SVM Classifier

This figure for the confusion matrix of the true positive rate(TPR) and false negative rate(FNR). In this figure the model uses dataset which is liver dataset, all rows classes stand for the true class. The columns show the predicted classes. In the top row, 48.0% of the class0 are correctly classified so 48.0% is the true positive rate for correctly classified points in this class and 52.0% are misclassified which is false negative rate. In the next row, 80.2% of the class1 are correctly classified so 80.2% is the true positive rate for correctly classified points in this class and 19.8% are misclassified which is false negative rate. To see results per predicted class, under Plot, select the positive predictive values (PPV)**,** false discovery rates (FDR). The PPV is the proportion of correctly classified observations per predicted class. The FDR is the proportion of incorrectly classified observations per predicted class.

The confusion matrix generated from the 10-fold cross validation by using SVM classifier is shown in Table.

Table 4.2: Accuracy result of SVM classifier

Total no. of Instances	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Percentage of error	Percentage of Accuracy
Class '0'	102	49	53	52.0%	48.0%
Class '1'	202	162	40	19.8%	80.2%
Total	304	211	93	30.6%	69.4%

Form the above table get the values of confusion matrix for (SVM) where class 0 means negative value and class 1 means positive value. In the above table correctly classified instances are 211 and incorrectly classified instances are 93 which leads 69.4% and 30.6% respectively. Following curve shown in figure is for ROC curve for measuring performance.

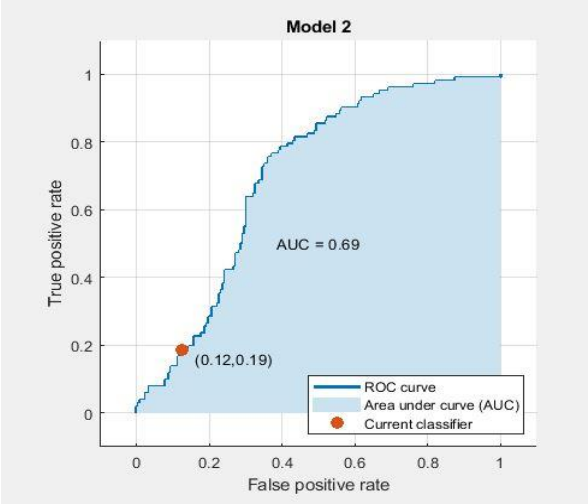


Figure 4.5: ROC curve for measuring performance of SVM

Fig shows Receiver Operating Characteristic (ROC) Curve of SVM classifier. The Y axis is stand for true positive value and X axis is for false positive value. Area under ROC(AUC) for both Class 0 and Class 1 is 0.69. The closer point of the true positive rate is (0.12,0.19).

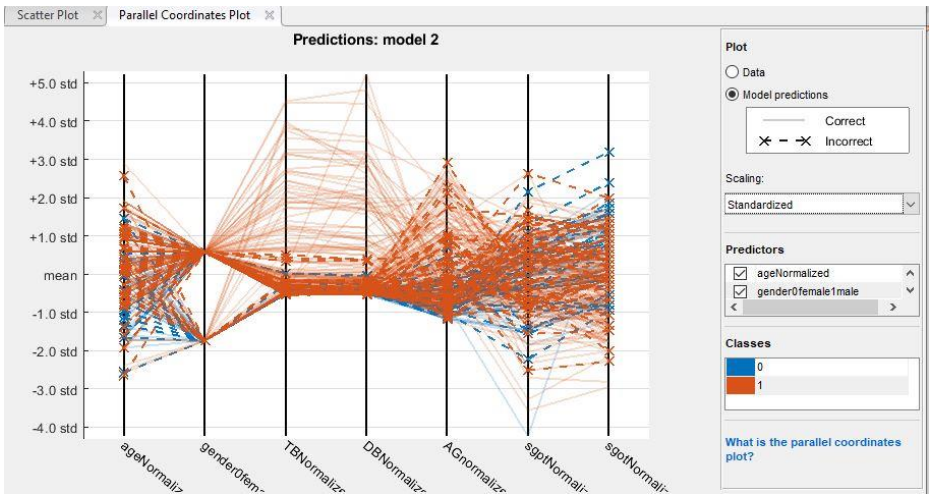


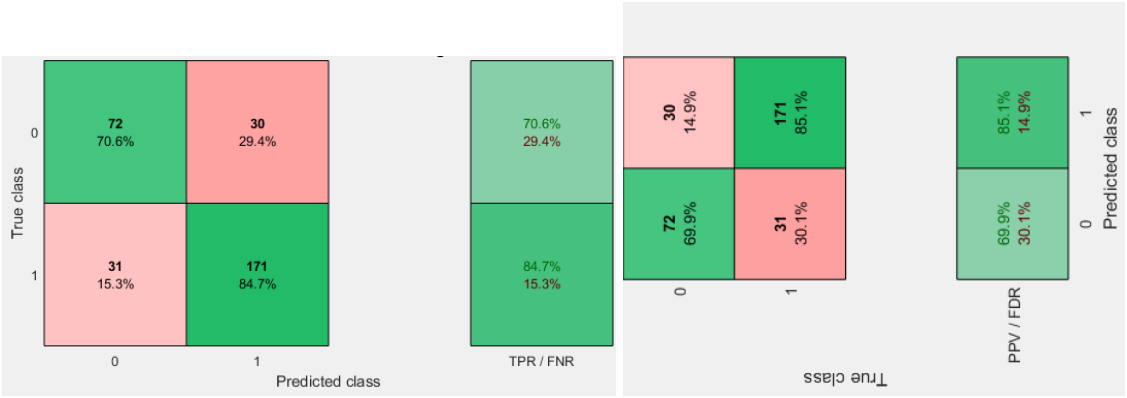
Figure 4.6: Parallel coordinates plot for SVM

Parallel coordinates plot used to visualize high dimensional data, where each observation is represented by the sequence of its coordinate values plotted against their coordinate indices. The resulting plot contains one line for each observation. Here figure shows the

parallel coordinates plot for SVM model. In the figure, red lines for the class 1 (positive value) and the blue lines shows the class 0 (negative value). Scaling is based on standardized values. The plot shows the median values for each group as a solid line and the quartile values as dotted lines of the same color. Here the blue line shows the median value measured for each variable on class0 and red line shows the median value measured for each variable on class1.

4.1.3 K- Nearest Neighbor Classifier

For applying the K-NN classifier on dataset by using MATLAB, have to go to the App part of the software. After A 10-fold cross validation the resultant accuracy was obtained was 79.9%. The time taken to build cross-validation was 13.60 seconds.



(a) Confusion matrix for TPR/FNR

(b) Confusion matrix for PPV/FDR

Figure 4.7: Confusion matrix obtained after training K-NN Classifier

The above figure is stand for confusion matrix by K-NN Classifier where the green cell is shown where the predicted result is matched with the true class. And the red cell is shown where the predicted result is not matched with the true class.

This figure for the confusion matrix of the true positive rate(TPR) and false negative rate(FNR). In this figure the model uses dataset which is liver dataset, all rows classes stand for the true class. The columns show the predicted classes. In the top row, 70.6% of the class0 are correctly classified so 70.6% is the true positive rate for correctly classified points in this class and 29.4% are misclassified which is false negative rate. In the next row, 84.7% of the class1 are correctly classified so 84.7% is the true positive rate for

correctly classified points in this class and 15.3% are misclassified which is false negative rate. To see results per predicted class, under Plot, select the positive predictive values (PPV), false discovery rates (FDR). The PPV is the proportion of correctly classified observations per predicted class. The FDR is the proportion of incorrectly classified observations per predicted class.

The confusion matrix generated from the 10-fold cross validation by using K-NN classifier is shown in Table:

Table 4.3: Accuracy result of K-NN classifier

Total no. of Instances	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Percentage of error	Percentage of Accuracy
Class '0'	102	72	30	29.4%	70.6%
Class '1'	201	171	31	15.3%	84.7%
Total	304	247	57	20.1%	79.9%

Form the above table get the values of confusion matrix for K-NN where class 0 means negative value and class 1 means positive value. Left side column is for the actual value and the first row is for predicted value. Following curve shown in figure is for ROC curve for measuring performance.

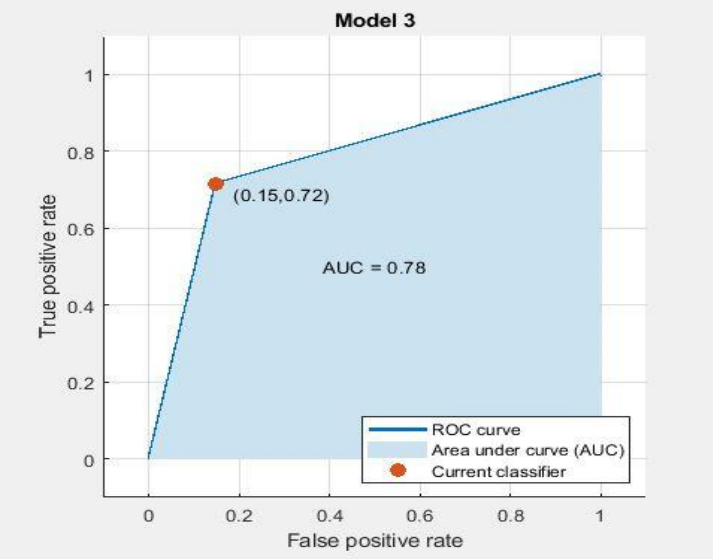


Figure 4.8: ROC curve for measuring performance of K-NN

Fig shows Receiver Operating Characteristic (ROC) Curve of K-NN classifier. The Y axis is stand for true positive value and X axis is for false positive value. Area under ROC(AUC) for both Class 0 and Class 1 is 0.78. The closer point of the true positive rate is (0.15,0.72).

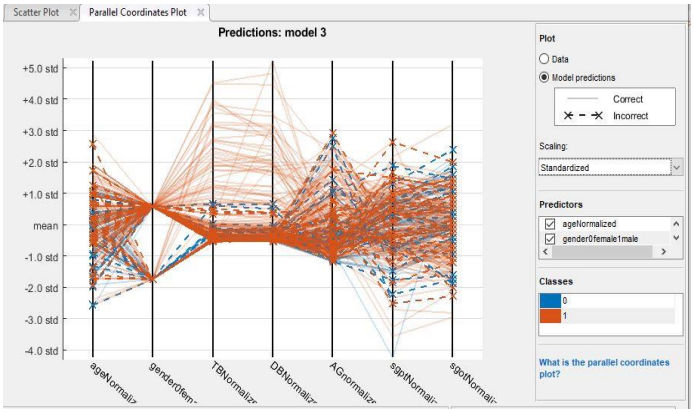


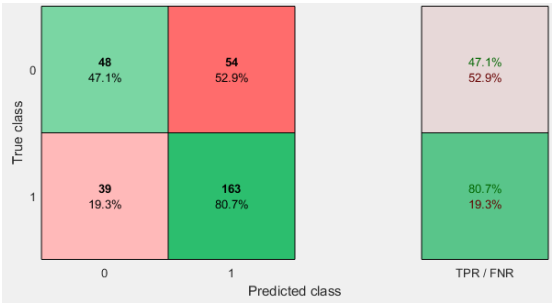
Figure 4.9: Parallel coordinates plot for K-NN

Parallel coordinates plot used to visualize high dimensional data, where each observation is represented by the sequence of its coordinate values plotted against their coordinate indices. The resulting plot contains one line for each observation. Here figure shows the parallel coordinates plot for K-NN model. In the figure, red lines for the class 1 (positive value) and the blue lines shows the class 0 (negative value). Scaling is based on standardized values. The plot shows the median values for each group as a solid line and the quartile values as dotted lines of the same color. Here the blue line shows the median value measured for each variable on class0 and red line shows the median value measured for each variable on class1.

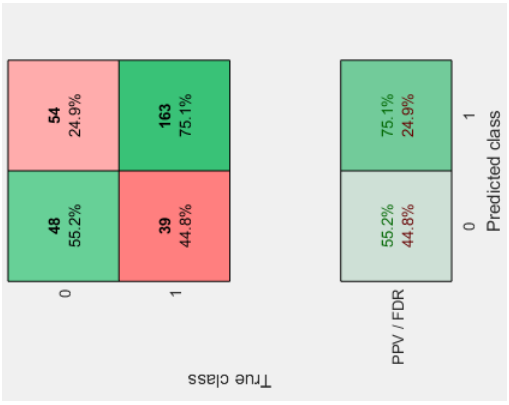
4.1.4 Fine Tree Classifier

For applying the Fine Tree classifier on dataset by using MATLAB, have to go to the App part of the software. After A 10-fold cross validation the resultant accuracy was obtained was 69.4%. The time taken to build cross-validation was 34.14 seconds. The following figure stands for confusion matrix by Fine Tree Classifier where the

green cell is shown where the predicted result is matched with the true class. And the red cell is shown where the predicted result is not matched with the true class.



(b) Confusion matrix for TPR/FNR



(b) Confusion matrix for PPV/FDR

Figure 4.10: Confusion matrix obtained after training Fine Tree Classifier

This figure for the confusion matrix of the true positive rate(TPR) and false negative rate(FNR). In this figure the model uses dataset which is liver dataset, all rows classes stand for the true class. The columns show the predicted classes. In the top row, 47.1% of the class0 are correctly classified so 47.1% is the true positive rate for correctly classified points in this class and 52.9% are misclassified which is false negative rate. In the next row, 80.7% of the class1 are correctly classified so 80.7% is the true positive rate for correctly classified points in this class and 19.3% are misclassified which is false negative rate. To see results per predicted class, under Plot, select the positive predictive values (PPV)**,** false discovery rates (FDR). The PPV is the proportion of correctly classified observations per predicted class. The FDR is the proportion of incorrectly classified observations per predicted class.

The confusion matrix generated from the 10-fold cross validation by using Fine Tree classifier is shown in Table:

Table 4.4: Accuracy result of Fine Tree classifier

Total no. of Instances	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Percentage of error	Percentage of Accuracy
Class '0'	102	48	54	52.9%	47.1%
Class '1'	201	163	39	19.3%	80.7%
Total	304	247	57	30.6%	69.4%

In the above table correctly classified instances are 211 and incorrectly classified instances are 93% which leads 69.4% and 30.6% respectively. After performing 10-fold Cross-Validations Table of Confusion matrix is given bellow:

Form the above table get the values of confusion matrix for Fine Tree where class 0 means negative value and class 1 means positive value. Left side column is for the actual value and the first row is for predicted value. Following curve shown in figure is for ROC curve for measuring performance.

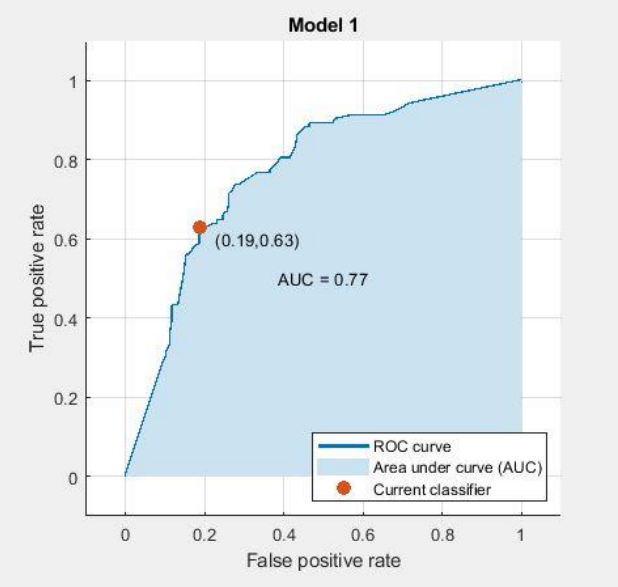


Figure 4.11: ROC curve for measuring performance of Fine Tree

Fig shows Receiver Operating Characteristic (ROC) Curve of Fine Tree classifier. The Y axis is stand for true positive value and X axis is for false positive value. Area under ROC(AUC) for both Class 0 and Class 1 is 0.77. The closer point of the true positive rate is (0.19,0.63).

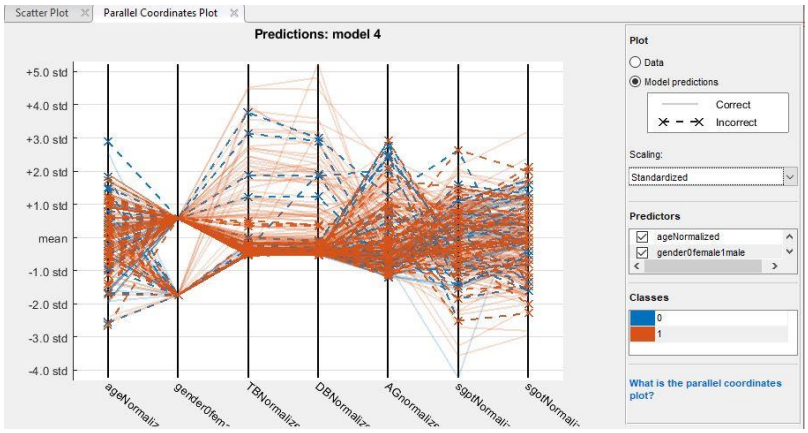


Figure 4.12: Parallel coordinates plot for Fine Tree

Parallel coordinates plot used to visualize high dimensional data, where each observation is represented by the sequence of its coordinate values plotted against their coordinate indices. The resulting plot contains one line for each observation. Here figure shows the parallel coordinates plot for Fine Tree model. In the figure, red lines for the class 1 (positive value) and the blue lines shows the class 0 (negative value). Scaling is based on standardized values. The plot shows the median values for each group as a solid line and the quartile values as dotted lines of the same color. Here the blue line shows the median value measured for each variable on class0 and red line shows the median value measured for each variable on class1.

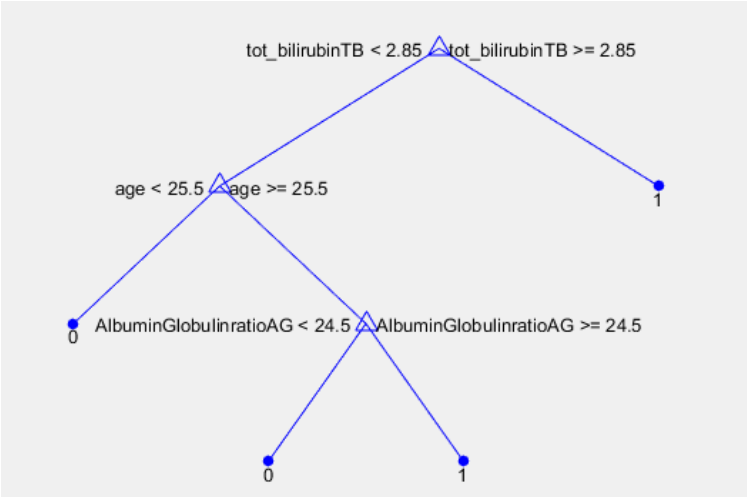


Figure 4.13: Graphical view of the Fine Tree model

For generating the tree view of the Fine Tree model, the whole code and the compact model both have to be exported in the working space. Then the function has to write to generate the graph of the Tree. In the command line, for graphical view have to write the line that is view (M, 'Mode', 'Graph') where the M stands for the model name which is exported after model build.

4.2 COMPARISON WITH PREVIOUS RELATED WORK

Anju Gulia, Dr. Rajan Vohra, Praveen Rani in their paper "Liver Patient Classification Using Intelligent Techniques" showed Random Forest gives 71.8696% accuracy [2]. The author Onwodi Gregory in his paper "Prediction of Liver Disease (Biliary Cirrhosis) Using Data Mining Technique" showed FT Tree algorithm gives 78% accuracy [4]. Ebenezer Obaloluwa Olaniyi, Khashman Adnan in their paper "Liver Disease Diagnosis Based on Neural Networks" showed Radial basis function neural network 70% accuracy [5]. Tapas Ranjan Baitharu, Subhendu Kumar Pani in their research "Analysis of Data Mining Techniques for Healthcare Decision Support System Using Liver Disorder Dataset" showed Multilayer Perceptron 71.59% accuracy [6]. On the other hand, in this proposed paper, it shows 81.30% in Bagged Trees. The following table shows the comparison with previous related work.

Table 4.5: Comparison with Previous Related Work

Sl No.	Title	Author	Year	Methods Techniques	Accuracy (%)	Accuracy of this work (%)
1.	Liver Patient Classification Using Intelligent Techniques	Anju Gulia, Dr. Rajan Vohra, Praveen Rani	2014	SVM, Bagged Tree	Bagged Tree 70.32 %, SVM 71.35%	Bagged Tree 81.3%, SVM 69.4%
2.	A Critical Study of Selected Classification	B. Venkata Ramana1, M. S. P. Babu and	2011	KNN, NB, DT, MLP	Decision Tree 69.4%, KNN	Decision Tree 69.4%, KNN

	Algorithms for Liver Disease Diagnosis	N.B. Venkateswarlu			65.3%	79.9%
3.	Prediction of Liver Disease (Biliary Cirrhosis) Using Data Mining Technique	Onwodi Gregory	2015	eleven data mining classification algorithms	FT Tree algorithm gives 78% accuracy	FT Tree 69.4%,
4	Disease prediction with different types of neural network classifiers	CH. Weng, TCK. Huang, RP. Han	2016	Ensemble Classifier	Ensemble Classifier 68.49%	Ensemble Classifier (Bagged Tree) 81.30%

4.3 COMPARATIVE ANALYSIS AND FINAL SELECTION

The paper carried out experiments in order to evaluate the performance and usefulness of different classification algorithms for predicting Liver disease. After performing different classification algorithms, values of different term for each classification algorithm are listed in a table to measure and investigate the performance on the classification methods. The Table shows the performance measurement for the selected classifiers. In this simple experiment, from Table it can be said that a Bagging required 50.27 seconds, SVM requires around 16.178 seconds compared to K-NN which requires around 10.942 second and provides better building time. Comparison is made among these classification algorithms out of which Bagged Trees classifier is considered as the better performance algorithm. Because it provides the better accuracy 81.3% than others classifiers. SVM and Fine Tree provide lower accuracy 69.4%. The

following table shows accuracy rate and shows build time of different classifier for dataset.

Table 4.6: Performance measurement of all classifiers

Evaluation Criteria	Bagged Trees	SVM	K-NN	Fine Tree
Timing to build model (in Sec)	50.27	16.178	10.942	24.267
Correctly classified instances	247	211	243	211
Incorrectly classified instances	57	93	61	93
Error (%)	18.7%	30.6%	20.1%	30.6%
Accuracy(%)	81.3%	69.4%	79.9%	69.4%
ROC Area	0.860	0.720	0.760	0.790

In the table above table shows the values in some evaluation criteria- time to build model, number of correctly classified instances, incorrectly classified instances and accuracy of all the model which is developed in MATLAB. The following figure shows in bar chart form of accuracy of all the models.

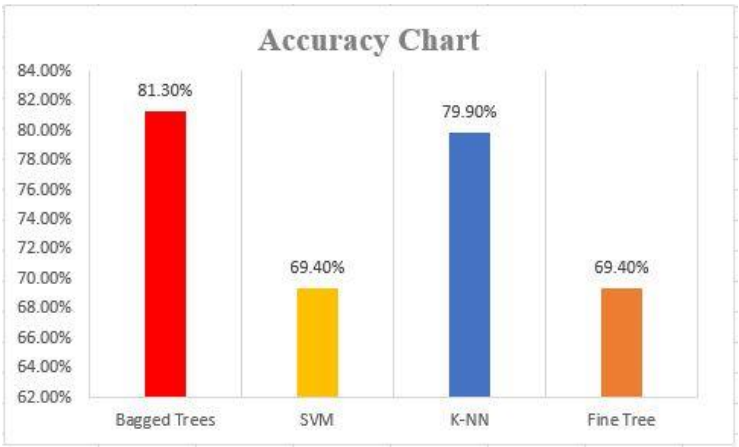


Figure 4.14: Accuracy comparison chart of different classifiers

From the above figure shows that Bagged Trees, SVM, K-NN and Fine Tree model give accuracy in percentage 81.30%, 69.40%, 79.90% and 69.40% respectively. It can visibly show that Bagged Trees is highest in accuracy measurement which is 81.30%.

This bar chart is built in excel sheet. The following figure shows in bar chart form of the time to build the models.

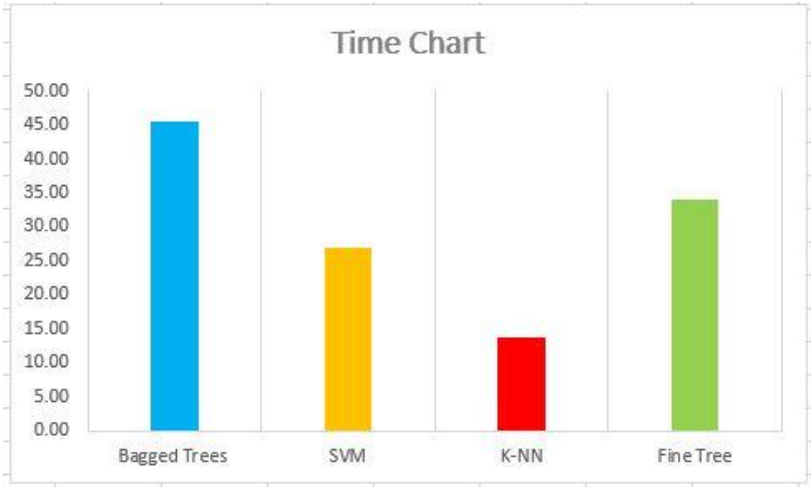


Figure 4.15: Time comparison chart of different classifiers

From the above figure it can visibly shows that Bagged Trees model is highest in build in time measurement which is 50.27 in seconds. Which highest then the others as its process in ensemble of a number of decision tree classifier. Although, all the time measurement of the model creation is varied from machine to machine.

From the above figure, it can be shown that Bagged Trees model shows the highest accuracy 81.30%. The others three SVM 69.40%, K-NN 79.90% and 69.40% accuracy respectively. Although, all the time measurement of the model creation is varied from machine to machine. After comparing with all the aspects of all the models, it visibly shows that Bagged Trees model is the best model with respect to the accuracy.

4.4 TESTED WITH NEW SAMPLES

After whole model development and performance analysis with different classification algorithm, the values of different models are investigated with the performance on the classification methods. Comparison is made among these classification algorithms and finally Bagged Trees classifier is considered as the better performance algorithm. Because it provides the better accuracy 81.3% than others

classifiers. Now Bagged Tree model is used for prediction process. The steps of the prediction process are given bellow.

I. Test the dataset with new 10 instances. The following table is new instances table.

Table 4.7: Data samples for prediction

Age normalized	gender	TB normalized	DB normalized	A/G normalized	Sgpt normalized	Sgot normalized
0.01320	0	0.01320	0.010204	0.02247	0.66153	0.54347
0.27380	1	0.01320	0.005102	0.01123	0.73846	0.58695
0.36904	0	0.00990	0.005102	0.29134	1	0.80434
0.40476	1	1	0.928571	0.09662	0.8	0.39130
0.47619	0	0.00990	0.005102	0.33707	0.50769	0.43478
0.38095	1	0.02970	0.015306	0.00224	0.50769	0.45652
0.23809	0	0.00330	0.005102	0.24719	0.46153	0.32608
0.46428	0	0.01320	0.010204	0.25842	0.6	0.65217
0.52380	1	0.019801	0.010204	0.10112	0.67692	0.78260
0.52380	1	0.072607	0.076530	0.06292	0.8	0.4782

II. For prediction process, after training process export the selected whole model from software APP in the working space.

III. Then import the new sample dataset which is also normalized. In the dataset, all the attribute field will be as same as the previous full dataset for training purpose. Just the values of the target class are not being included.

IV. In the working window, have to write a specific function for all the trained model which is exported and it is ‘yfit = trainedmodel.predictFnc(T)’. Here trainedmodel is the compact model name and T is the name of the test dataset.

V. Run the test dataset in MATLAB. Then apply different classifier algorithms for testing purpose. As a result, different algorithm shows different predictive result. As for example, in figure includes output of algorithm which is Bagged Trees.

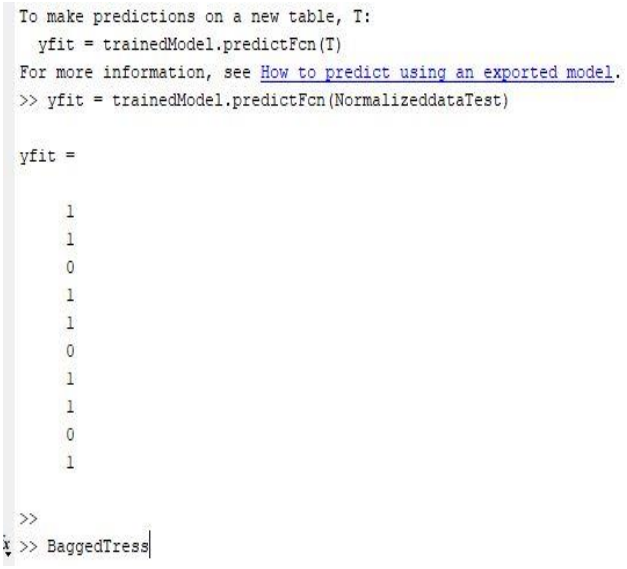


Figure 4.16: Prediction of new samples

One of the main task of the research is to show the prediction with new samples. As the main goal of the research is to find out the best algorithm which will give better accuracy in early prediction of liver disease to minimize the death rate. In this purpose, after follow the 4.3 section process, get the output of the prediction. In the following figure shows the prediction results of the all of the experimental models.

Table 4.8: Prediction results of new samples

Is_patient (yes=1, no=0)	BaggedTrees PredictedResult	SVM PredictedResult	KNN PredictedResult	Fine Tree PredictedResult
1	1	1	1	1
0	1	1	0	1
0	0	1	0	1
1	1	1	1	1
1	1	1	0	1
0	0	1	0	0
1	1	1	1	1
1	1	0	1	1
0	0	0	0	0
1	1	1	1	1

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

This research focus on the use of different supervised classification techniques to enhance the early detection Liver disease. Liver disease is a fatal disease which may cause life threatening complication such as death. Here used real time dataset of liver patients which are collected from hospital and clinic. In proposed model, all the experiment is based on real dataset and MATLAB software is used for experiments. This research work used classification algorithms namely Bagged Trees, SVM, K-NN, Fine Tree. Each classification algorithm's outcome is used to determine the better classifiers among them. Comparisons of these selected algorithms are based on the performance factors. By the evaluation with all algorithms Bagged Trees gives 81.30%, SVM gives 69.4%, K-NN gives 79.90%, Fine Tree gives 69.4%. From the experimental results, this work concludes that, Bagged Trees classifier can be considered as a best algorithm because of its highest classification accuracy 81.30%. As Bagged Trees gives better accuracy it can be used for early Liver disease prediction.

This research paper's goal is to provide better performance and prediction. For calculating performance, here used classifier algorithms. Then calculated the performance and get predictive class. In future, there is a possibility to generate a new algorithm or a highbred one which will provide better accuracy than the existing classifier algorithms.

REFERENCES

[1] G. S. R. P. Rajeswari, "Analysis of Liver Disorder Using Datamining Algorithm," *Global Journal of Computer Science and Technology,* vol. 10, no. 14, pp. 48-52, 2010.

[2] D. R. V. P. R. Anju Gulia, "Liver patient classification using Intelligent Techniques," *International Journal of Computer Science and Information Technologies,* vol. 5, no. 4, pp. 5110-5115, 2014.

[3] J. V. A. S. Aneeshkumer, "A novel approach for liver disorder classification using datamining techniques," *The Journal of Scientific and Engineering Research,* vol. 2, no. 1, pp. 15-18, 2015.

[4] O. Gregory, "Prediction of liver disease (biliary cirrhosis) using datamining technique," *International Journal of Engineering Technology and Research,* vol. 1, no. 1, pp. 2347-6079, 2015.

[5] E. O. O. K. Aadnan, "Liver disease diagnosis based on neural networks," in *Advances in Computational Intelligence*, 2015.

[6] S. K. P. Tapas Ranjan Baitharua, "Analysis of datamining techniques for healthcare decision support system using liver disorder dataset," in *International Conference on Computational Modeling and Security*, 2016.

[7] P. M. S. P. B. a. P. N. B. V. Bendi Venkata Ramana, "A critical study of selected classification algorithms for liver disease diagnosis," *International Journal of Database Management Systems,* vol. 3, no. 2, pp. 101-111, 2011.

[8] S. Karthik, "Classification and rule extraction using rough set for diagnosis of liver disease and its types," *Advances in Applied Science Research,* vol. 2, no. 3, pp. 334-345, 2011.

[9] E. E. A. Y. K. A. K. A. F. Otoom, "Effective diagnosis and monitoring of heart disease," *International Journal of Software Engineering and Its Applications,* vol. 9, no. 1, pp. 143-156, 2015.

[10] M. S. D. Dr. S. Vijiayarani, "Liver disease prediction using SVM and Naive Bayes algorithms," *International Journal of Science, Engineering and Technology Research,* 2015.

[11] S. J. R. S. A. Iyer, "Diagnosis of diabetes using classification mining techniques," *International Journal of Data Mining and Knowledge Management Process,* vol. 5, no. 1, pp. 114-125, 2015.

[12] T. H. R. H. CH. Weng, "Disease prediction with different types of neural network classifiers," *Telematics Information,* vol. 33, no. 2, pp. 277-292, 2016.

[13] S. K. J. K. H. Jin, "Decision factors on effective liver patient data prediction," *International Journal of Bioscience and Biotechnology,* vol. 6, no. 4, pp. 167-178, 2014.

[14] M. B. N. V. B.V. Ramana, "Liver classification using modified rotation forest," *International Journal of Engineering Research and Development,* vol. 1, no. 6, pp. 17-24, 2012.

[15] S. Dhamodharan, "LIver disease prediction using bayesian classification," in *National Conference on Advanced Computing Appications and Technologies*, 2014.

[16] T. M. S.N.N. Alfisahrin, "Datamining techniques for optimization of liver disease classification," in *International Conference on Advanced Computer Science Applications and Technologies*, 2013.

[17] B. V. R. a. P. M. S. P. Babu, "Liver classification using modified rotational forest," *International Journal of Engineering Research and Development,* vol. 1, no. 6, pp. 17-24, 2012.

[18] D. C. J. V. A.S. Aneeshkumar, "A novel approach for liver disorder classification using datamining techniques," *Engineering and Scientific International Journal,* vol. 2, no. 1, pp. 2394-7187, 2015.

[19] R. J. P. D. Sindhuia, "A Survey on classification techniques in datamining for analyzing liver disease disorder," *International Journal of Computer Science and Mobile Computing,* vol. 5, no. 5, pp. 1-5, 2016.

[20] V. S. C. K. T. V. V. Shankar Sowmien, "Diagnosis of hepatitis using decision tree algorithm," *International Journal of Engineering and Technology,* vol. 8, no. 3, pp. 2319-8613, 2016.

[21] H. M. H. F. M. Ba-Alwi, "Comparative study for analysis the prognostic in hepatitis data: datamining approach," *International Journal of Science and Engineering Research,* vol. 4, no. 8, pp. 680-685, 2013.

[22] S. S. Y. C. Kim YS, "Screening test data analysis for liver disease prediction model using growth curve," *Biomedicine and Pharmacotherapy,* vol. 57, no. 8, p. 482, 2003.

[23] Y. O. H. N. S. H. Hiroshi Nakano, "Application of neural network to the interpretation of laboratory data for the diagnosis of two forms of chronic active hepatitis," *International Hepatology Communications,* vol. 5, no. 3, pp. 160-165, 1996.

[24] P. K. G. S. A. A. Comak E, "A new medical decision making system: least square support vector machine (LSSVM) with fuzzy weighting pre-processing," *Expert Systems with Applications,* vol. 32, p. 409—414, 2007.

[25] C.-C. W. Md.Mohaimenul Islam, "Applications of Machine Learning in Fatty Live Disease Prediction," *European Federation for Medical Informatics (EFMI),* vol. 5, no. 247, p. 166, 2018.

[26] S. M. T. A. S. P. A. R. S. Mehdi Birjandi, "Prediction and diagnosis of non-alcoholic fatty liver disease (NAFLD) and identification of its associated factors using the classification tree method," *Iranian Red Crescent Medical Journal,* vol. 18, no. 11, p. 5812, 2016.

[27] C.-C. Wua, "Prediction of fatty liver disease using machine learning algorithms," *Computer Methods and Programs in Biomedicine,* vol. 170, pp. 23-29, 2019.

[28] A. M. M. E. S. Bahramirad, "Classification of liver disease diagnosis: a comparative study," in *Second International Conference on Informatics and Applications*, 2013.

[29] M. ABDAR, "A Survey and Compare the Performance of IBM SPSS Modeler and Rapid Miner Software for Predicting Liver Disease by Using Various Data Mining Algorithms," *Cumhuriyet Science Journal ,* vol. 36, no. 3, pp. 3230-3241, 2015.

[30] R. V. J. M. S. P. Jankisharan Pahareeya, "Liver Patient Classification using Intelligence Techniques," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, no. 2, p. 2, 2014.

[31] I. P. M. T. Ahmad Ashari, "Performance comparison between naïve bayes, decision tree and k -nearest neighbor in searching alternative design in an energy simulation tool," *International Journal of Advanced Computer Science and Applications,* vol. 4, no. 11, pp. 33-39, 2013.

[32] R. J. T. Bradley Efron, "An introduction to the bootstrap," in *Chapman & Hall*, New York.

[33] M. G. N. G. Esteban Alfaro, "An R package for classification with boosting and bagging," *Journal of Statistical Software,* vol. 54, no. 32, p. 1135, 2013.

[34] M. Dr. S. Vijayarani, "DATA MINING CLASSIFICATION ALGORITHMS FOR KIDNEY DISEASE PREDICTION," *International Journal on Cybernetics & Informatics,* vol. 4, no. 4, pp. 12-25, 2015.

[35] S. R. F. K. K. F. S. A. Mamun Al Mahtab, "Epidemiology of hepatities B virus in Bangladesi general population," *Hepatobiliary Panceat Dis Int,* vol. 7, no. 6, pp. 595-600, 2008.

APPENDIX A

Sample of Collected Data

Age	gender	TB normalized	DB normalized	A/G normalized	Sgpt normalized	Sgot normalized	Is Patient
38	0	2.8	1.3	5.6	5.7	3.1	0
35	1	1.6	0.4	2.4	7.2	3.4	0
32	1	12.1	6	1.2	6.6	2.4	1
32	1	3.7	1.6	1.8	6.2	1.9	1
57	0	0.7	0.1	2.4	7	4.2	0
61	1	0.7	0.2	1.8	6.3	2.8	0
16	1	2.6	1.2	1.0	5.4	2.6	1
16	1	7.7	4.1	1.68	7.1	4	1
20	0	16.7	8.4	1.1	6.9	3.5	1
52	1	2.7	1.4	4.0	6	1.7	1
66	1	16.6	7.6	3.84	6.9	2	1
62	1	0.7	0.2	1.08	4.9	2.7	0
26	1	42.8	19.7	1.38	7.5	2.6	1
66	1	16.6	7.6	3.84	6.9	2	1
62	1	0.7	0.2	1.08	4.9	2.7	0
26	1	42.8	19.7	1.38	7.5	2.6	1
51	1	4	2.5	3.30	7.5	4	1
61	1	5.8	3.1	1.03	6.6	3.5	0
50	1	2.2	0.6	5.2	6.6	3.6	0
65	0	0.7	0.1	1.8	6.8	3.3	1
62	1	10.9	5.5	1.00	7.5	3.2	1
62	1	7.3	4.1	6.8	7	3.3	1
61	1	1.5	0.6	8.2	6.7	3.6	0
72	1	3.9	2	5.9	7.3	2.4	1
46	1	1.8	0.7	1.4	7.6	4.4	1
26	0	0.9	0.2	1.2	7	3.5	1

APPENDIX B

Sample of Normalized Data

Age	gender	TB normalized	DB normalized	A/G normalized	Sgpt normalized	Sgot normalized	Is Patient
0.404761905	0	0.01320132	0.010204082	0.114606742	0.523076923	0.347826087	1
0.5	0	0.00990099	0.647959184	0.168539326	0.538461538	0.369565217	1
0.488095238	0	0.01320132	0.005102041	0.191011236	0.507692308	0.456521739	1
0.178571429	0	0.00330033	0	0.460674157	0.430769231	0.347826087	1
0.642857143	1	0.191419142	0.158163265	0.015730337	0.584615385	0.608695652	0
0.547619048	1	0.02640264	0.015306122	0.168539326	0.507692308	0.47826087	1
0.642857143	1	0.062706271	0.025510204	0.494382022	0.461538462	0.217391304	1
0.130952381	1	0.01320132	0.005102041	0.393258427	0.692307692	0.673913043	0
0.571428571	1	0.00660066	0.005102041	0.123595506	0.430769231	0.391304348	1
0.702380952	1	0.240924092	0.214285714	0.685393258	0.507692308	0.456521739	1
0.345238095	1	0.00990099	0.005102041	0.168539326	0.676923077	0.47826087	1
0.619047619	1	0.01320132	0.005102041	0.280898876	0.692307692	0.565217391	1
0.476190476	1	0.00660066	0.005102041	0.146067416	0.676923077	0.456521739	0
0.380952381	1	0.01650165	0.010204082	0.157303371	0.707692308	0.760869565	0
0.297619048	1	0.02640264	0.020408163	0.247191011	0.630769231	0.543478261	1
0.547619048	1	0.00990099	0.005102041	0.438202247	0.569230769	0.5	1
0.404761905	1	0.00330033	0	0.235955056	0.507692308	0.5	1
0.642857143	1	0	0	0.269662921	0.492307692	0.152173913	0
0.607142857	0	0.00660066	0	0.157303371	0.661538462	0.717391304	0
0.654761905	1	0.00660066	0.005102041	0.08988764	0.553846154	0.413043478	0
0.119047619	1	0.069306931	0.056122449	0.898876404	0.415384615	0.369565217	1
0.119047619	1	0.237623762	0.204081633	0.076404494	0.676923077	0.673913043	1
0.166666667	0	0.534653465	0.423469388	0.001123596	0.646153846	0.565217391	1
0.547619048	1	0.072607261	0.066326531	0.337078652	0.507692308	0.173913043	1
0.523809524	0	0.01650165	0.010204082	0.235955056	0.784615385	0.673913043	0
0.44047619	1	0.726072607	0.596938776	0.048314607	0.6	0.260869565	1

APPENDIX C

Algorithms

Algorithm of Bagged Trees

Input: training set S, Inducer Ғ, integer T (number of bootstrap samples)

i. For i = 1 to T {
ii. Si= bootstrap sample from S (sample with replacement).
iii. $C_i = Ғ(Si)$
iv. }
v. $C^*(x) = argmax_{y \epsilon Y} \sum_{i:Ci(x)=y} 1$ (the most often predicted label y)
vi. Output: classifier C^*

Algorithm of SVM

Step 1: Let's assume a supervised binary classification problem. Let us consider that the training set consists of N vectors from the dimensional feature space.

$$x_i \in \Re^d \ (i = 1,2,...., N) \qquad 3.3$$

Step 2: A target $y_i \in \{-1,+1\}$ is associated to each vector x_i .

Step 3: Let us consider that the two classes are linearly separable. This points that it is possible to discovery at least one hyperplane (linear surface) defined by a vector $w \in \Re^d$ (normal to the hyperplane) and a bias $b \in \Re$ that could separate two classes without errors.

Step 4: The membership decision rule can be based on the function sgn [f(x)], where f(x) is the discriminant function associated with the hyperplane and defined as

$$f(x) = w.x + b \qquad 3.4$$

In case to find such a hyperplane, one should estimate w and so that

$$yi\ (w.xi + b) > 0 \text{ , with } i = 1,2,..., N \qquad 3.5$$

Step 5: The SVM approach involves in discovering the optimal hyperplane that increases the distance between the neighboring training sample and the splitting hyperplane. It is possible to express this distance as equal to 1/ || w|| with a simple rescaling of the hyperplane parameters w and b such that

$$\min_{l=1,2,....N} yi(w.xi + b) \geq 1 \qquad 3.6$$

Step 6: Consequently, it changes the optimal hyperplane which can be controlled by the following solution of convex quadratic programming problem

$$\text{Min} : \frac{1}{2} \|w\|^2 \quad 3.7$$

Step 7: This traditionally linear constrained optimization problem can be interpreted (using a Lagrangian formulation) into the following dual problem

$$\text{Max: } \sum_{i=1}^{N} \alpha i - \frac{1}{2} \sum_{i=1}^{N} \quad \sum_{j=1}^{N} \alpha i \alpha j y i y j (xi.xj) \quad 3.8$$

Step 8: The Lagrange formulizers αi 's (i =1,2,...,N) represented in above can be assessed using quadratic programming (QP) methods. The discriminant function associated with the optimal hyperplane becomes an equation depending both on the Lagrange multipliers and on the training samples, i.e.

$$f(x) = \sum_{i \epsilon s} \alpha i y i (xi.x) + b \quad 3.9$$

Where s is the subset of training samples corresponding to the nonzero Lagrange multiplier's. It is worth noting that the Lagrange multipliers effectively weight each training sample according to its importance in determining the discriminant function. The training samples associated to nonzero weights are called support vectors. These lie at a distance exactly equal to 1/ || w|| from the optimal separating hyperplane.

Algorithm of KNN

i. Determine the parameter K i.e., number of nearest neighbors beforehand.
ii. Calculate the distance between test data and each row of training data with the help of any of the method namely: euclidean distance measure algorithm.
    a. d(p,q) = $\sqrt{(p+q)^2}$
    3.10
iii. Distances for all the training samples are sorted and nearest neighbor based on the K-th minimum distance is determined.
iv. Since the K-NN is supervised learning, get all the Categories of your training data for the sorted value which fall under K.
v. The prediction value is measured by using the majority of nearest neighbors.

Algorithm of Fine Tree

Step 1: create a node N

Step 2: if samples are all of the same class, C then

Step 3: return N as a leaf node labeled with the class C;

Step 4: if attribute-list is empty then

Step 5: return N as a leaf node labeled with the most common class in samples;

Step 6: select test-attribute, the attribute among attribute-list with the highest information gain;

Step 7: label node N with test-attribute;

Step 8: for each known value ai of test-attribute

Step 9: grow a branch from node N for the condition test-attribute= ai;

Step 10: let s_i be the set of samples for which test-attribute= ai;

Step 11: if s_i is empty then

Step 12: attach a leaf labeled with the most common class in samples;

Step 13: else attach the node returned by generate decision tree (si, attribute-list, and test-attribute)

Counting Gain This process uses the "Entropy" which is a measure of the data disorder. The Entropy of is calculated by

$$Entropy(y) = \sum_{i=1}^{n} \frac{|yi|}{|y|} \log \left(\frac{|yi|}{|y|}\right) \quad 3.11$$

$$Entropy(i|y) = \frac{|yi|}{|y|} \log \left(\frac{|yi|}{|y|}\right) \quad 3.12$$

And Gain is

$$Gain(y, i) = Entropy(y - Entropy(i|y)) \quad 3.13$$

APPENDIX D

Code

Code of Bagged Trees

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitensemble(predictors, response, 'Bag', 200, 'Tree', 'Type',
'Classification', 'PredictorNames', {'ageNormalized' 'gender0female1male'
'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized' 'sgotNormalized'},
'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1]);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

Code of SVM

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitcsvm(predictors, response, 'KernelFunction', 'linear',
'PolynomialOrder', [], 'KernelScale', 'auto', 'BoxConstraint', 1, 'Standardize', 1,
'PredictorNames', {'ageNormalized' 'gender0female1male' 'TBNormalized'
```

```
'DBNormalized' 'AGnormalized' 'sgptNormalized' 'sgotNormalized'}, 'ResponseName',
'is_patientyes1no0', 'ClassNames', [0 1]);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

Code of KNN

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitcknn(predictors, response, 'PredictorNames', {'ageNormalized'
'gender0female1male' 'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized'
'sgotNormalized'}, 'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1], 'Distance',
'Euclidean', 'Exponent', '', 'NumNeighbors', 1, 'DistanceWeight', 'Equal',
'StandardizeData', 1);
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```

Code of Fine Tree

```
function [trainedClassifier, validationAccuracy] = trainClassifier(datasetTable)
% Extract predictors and response
predictorNames = {'ageNormalized', 'gender0female1male', 'TBNormalized',
'DBNormalized', 'AGnormalized', 'sgptNormalized', 'sgotNormalized'};
predictors = datasetTable(:,predictorNames);
predictors = table2array(varfun(@double, predictors));
```

```
response = datasetTable.is_patientyes1no0;
% Train a classifier
trainedClassifier = fitctree(predictors, response, 'PredictorNames', {'ageNormalized'
'gender0female1male' 'TBNormalized' 'DBNormalized' 'AGnormalized' 'sgptNormalized'
'sgotNormalized'}, 'ResponseName', 'is_patientyes1no0', 'ClassNames', [0 1],
'SplitCriterion', 'gdi', 'MaxNumSplits', 4, 'Surrogate', 'off');
% Perform cross-validation
partitionedModel = crossval(trainedClassifier, 'KFold', 10);
% Compute validation accuracy
validationAccuracy = 1 - kfoldLoss(partitionedModel, 'LossFun', 'ClassifError');
```