Received: July 9, 2023. Revised: August 28, 2023. 286

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

The Impact of Hepatitis Disease in Pre-processing Approach Using Fuzzy with May-Fly Optimization and Ensemble Classification

C. Saranya Jothi[1]* D. Umanandhini[1]

[1]*Department of Computer Science and Engineering,*
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology, Chennai, India
* Corresponding author's Email: saranyajothi22@gmail.com

Abstract: Hepatitis is a disease originating from a viral bacterium that can be avoided by the earlier identification of the disease and also with the proper classification. The proposed methodology is split into two phases: pre-processing technique and classification method. The pre-processing methods are incorporated to improve the accuracy and reduce the complexity of data. In the first phase, the pre-processing method such as (i) First, label encoder with one hot encoder (LE-OHE), is adopted to convert all the categorical data into a numerical column (ii) Then, Fuzzy logic with may fly (FL-MF) is developed to fill the missing values and (iii) Finally, inter-quartile range (IQR) technique is used to detect the outliers. In the second phase, the classification method is refined by following three steps (i) A correlation-based decision tree (CDT) is constructed to interpret the data as simple, (ii) A reduced error pruning tree (REPT) technique is employed to split the data using grid search algorithm and (iii) The different combination of REPT is given as input to the ensemble stacking method that enhances the predictive performance. According to the results, our proposed method outperforms with an average accuracy rate of 96.1%, and time complexity in both datasets (hepatitis and indian liver patient record (ILPR)) when compared with the existing algorithms such as min max scalar-support vector machine (MMS-SVM), mean imputation-random forest (MI-RF), gradient boosting (GB), random forest (RF), ensemble-stacking (ES), extreme gradient boosting (XGBoost), and radial basis function-extreme learning machine (RBF-ELM). Moreover, the proposed method also reduces the mean absolute error (MAE) error rate.

Keywords: Classification, Decision tree, Ensemble, Machine Learning, Pre-processing.

1. Introduction

Hepatitis represents a significant global health challenge, affecting a substantial number of people and imposing a considerable burden on healthcare systems [1]. This infectious disease is characterized by inflammation of the liver, often caused by viral infections. Increasing awareness and knowledge about this disease can improve better prevention. Early detection and effective management, ultimately can reduce the burden of hepatitis on global health [2]. In the realm of healthcare, the integration of machine learning algorithms has opened up new directions for improved patient care, diagnostic accuracy, and treatment outcomes [3].

Machine learning has emerged as a powerful tool in the healthcare domain, analysed, and utilized to enhance patient care and diagnosis. The success of machine learning in healthcare is based on two crucial steps: pre-processing [4-7] and classification techniques [8-13]. In this context, pre-processing involves various techniques such as data cleaning, removing redundancy, detecting outliers, and handling missing values [4]. These processes ensure that the machine learning algorithms receive high-quality input data, improving their ability to make reliable predictions and classifications. Once the data is pre-processed, machine learning algorithms are employed for classification tasks in healthcare. Classification algorithms are used to categorize patient records into distinct groups based on the input features [5]. These algorithms are trained on labeled datasets, where the correct class labels are provided by enabling the model to learn patterns and relationships between input features and target classes.

Pre-processing is an essential step in machine learning tasks, as it involves cleaning, filling in the missing values, and transforming raw text data into a suitable format for analysis and modelling. Generally, medical data often contain missing values, noisy entries, and inconsistent formats. In order to clean the data lot of researchers uses pre-preprocessing techniques to fill the data. Kotsiantis et al. [4] author implemented pre-processing techniques such as irrelevant information from the dataset being removed, outliers detected, and noise data being filtered to improve the model accuracy. This method deletes unwanted information which increases the processing time. It increases the complexity of the model due to the aggressive loss of some data. Haq et al. [5] introduced the hybrid intelligent system to combine multiple machine learning algorithms (MLA) to enhance the accuracy level. Here, min-max scalar (MMS) method is employed to reduce the feature making it more robust and reliable. The hybridized algorithm may increase the time complexity (15.4 sec) in building the framework and also properly training and tuning the hybrid system can be a challenging task. Saxena et al. [6] adopted the mean-imputing (MI) with a random forest (RF) algorithm to enhance interpretability and reduce model complexity. This approach works on the different data distributions and variations, making it suitable for handling large patient populations and data types. Without proper validation and regularization techniques, it reduces the accuracy rate (79.8 %). Yilmaz et al. [7] introduced the RF algorithm used for the analysis of complex patterns in medical data, which leads to higher accuracy compared to traditional diagnostic methods. Early detection allows for timely medical intervention and treatment that can significantly improve patient outcomes and reduce the risk of complications. The effectiveness of machine learning models relies on the data quality. If the data is incomplete, biased, or contains errors, it may impact the accuracy of early detection.

Classification is an important task in machine learning (ML) techniques. That involves categorizing data based on its features. The goal of classification is to build a predictive model that can learn from labeled training data and then predict accurately in the given input. The classification technique with recent trends is discussed in the following papers. Bei et al. [8] described the multiple algorithms to predict the mortality rate of hepatitis diseases. The author explained the real-time dataset that might not be apparent through traditional statistical methods. This can lead to more accurate predictions of patient outcomes, such as mortality rate, in alcoholic hepatitis cases. Moreover, the quality of the dataset is low due to the lack of completeness of the dataset impacts the performance evaluation of area under the roc curve (AUC) which obtained the value of 0.87. Siouda et al. [9] introduced hybrid radial basis function (RBF) and extreme learning machine (ELM) neural networks (NN) to capture complex non-linear relationships in medical data. It can improve classification accuracy compared to the traditional linear models. The ELM model leads to reduces the training time and computational complexity. The RBF algorithm avoids the problem of getting stuck in local minima during the optimization process. The diverse activation functions can increase the time complexity of the approach. Md et al. [10] proposed the pre-processing method for liver disease detection with an ensemble algorithm can enhance the performance of the method. Using ensemble machine learning algorithms, this method combines the strengths of multiple models by reducing the risk of overfitting. Also, it increases the overall performance for the parameters such as f-measure, recall, precision value (PV), and AUC. The structure of the ensemble decisions may be more complex, which can be a concern in critical medical decision-making. The ensemble approach required special techniques to address the imbalanced data. Mehrbakhsh et al. [11] demonstrate the data dimensionality reduction using the non-linear iterative partial least squares (IPLS) method. The neuro-fuzzy model can effectively handle uncertainty and outliers in medical data. While fuzzy logic-based models produce low reliability and ensembling techniques create some level of complexity which makes the overall model perform low. Sikora et al. [12] described the modified ensemble stacking method to address the scaling issues in distributed data mining which combines the meta-model using a genetic algorithm. The usage of several classifiers improves the performance. Here one of the major disadvantages is the execution time for the algorithm is high due to the data pre-processing technique. Azadeh et al. [13] discussed the six machine learning algorithms for the early prediction of liver diseases; among the six algorithms, the author concluded extreme gradient boosting (XGBoost) performance is well. The XGBoost algorithm can handle large datasets and perform well even on high-dimensional feature spaces. This work was not concentrated on pre-processing techniques, the accuracy level is less due to the improper selection of features.

Below are some of the drawbacks of traditional pre-processing with the classification approach: Pre-processing techniques like data cleaning [4] and dimensionality reduction [4] can lead to the loss of valuable information present in the original dataset.

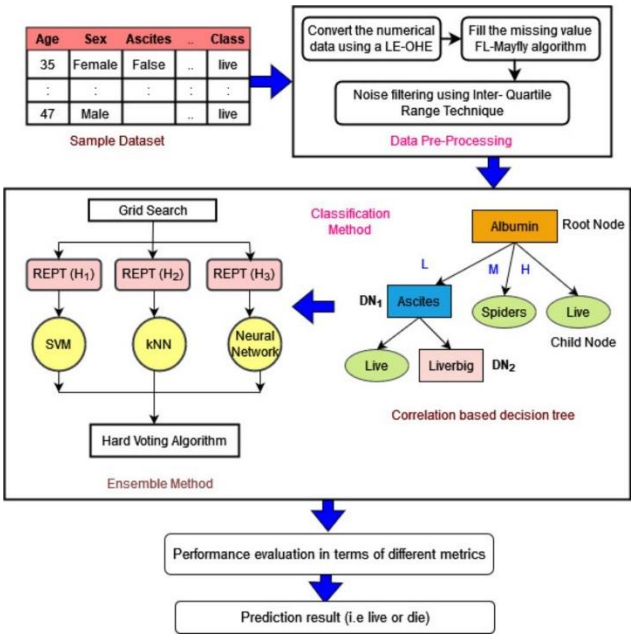

Figure. 1 Proposed framework

Some pre-processing techniques, such as feature engineering and feature selection, may inadvertently cause overfitting, where the model performs well on the training data but fails to generalize to new, unseen data. Many pre-processing techniques have hyperparameters that consume more time in tuning the process. While the separation of classes, the class imbalance is generated when the number of instances in different classes is significantly unequal. This can lead the classifier to be biased resulting in poor performance. To solve the above issues, the proposed method based on pre-processing techniques such as label encoder with one hot encoder (LE-OHE), a hybrid Fuzzy logic with may fly (FL-MF) optimization algorithm, and the inter-quartile range (IQR) method is implemented. Classification techniques such as the correlation-based decision tree (CDT), reduced error pruning tree (REPT), and Ensemble method are adopted to improve the classification accuracy.

1.1 Contribution of the work

The proposed method uses preprocessing and classification techniques to detect the diseases earlier and the major contributions of the work are listed below:

- The hybrid LE-OHE technique contains two methods. The first LE method classifies the attributes into a label based on weights and the OHE method further converts the attributes into rows and columns which improves the efficiency in terms of precision and processing time.
- The novel FL-MF optimization algorithm is developed to choose the appropriate missing values carefully that reach the optimal solution.
- Moreover, the IQR method is adopted to detect the outliers and boost the accuracy level.
- By constructing the CDT we can easily interpret the data and reduce the error rate for data preparation.
- According to the REPT process, it minimizes the misclassification error that overcomes the overfitting problems and also the grid search used to find the optimal combination of hyperparameters for a REPT model.
- To achieve better accuracy, the ensemble bagging mechanism is applied for classification techniques such as support vector machine (SVM), k-nearest neighbor (kNN), and NN to predict the multiple models via a hard voting algorithm.

2. Proposed framework

In order to forecast the diseases in their early stages, the proposed methodology implements pre-processing and classification techniques which are divided into two sub-sections. Initially, the sample dataset is given as the input for the pre-processing technique as shown in Fig. 1. section 2.1 describes the data preprocessing technique, to convert the categorical column into a numeric column via the LE-OHE method has been employed in section 2.1.1. Section 2.1.2 introduces a new hybrid FL-MF algorithm to fill the missing values and for cleaning the unnecessary data IQR technique is implemented to remove the noise as discussed in section 2.1.3. Section 2.2 explains the classification method which has three subsections in the first subsection 2.2.1, a CDT for better categorizing is illustrated. In the next subsection 2.2.2, REPT with hyperparameter tuning along a grid search mechanism is applied to build the different REPT. Finally, in the last subsection 2.2.3, the input of REPT is given to the ensemble technique, which merges the distinct classifier to increase the performance.

2.1 Data pre-processing

Data pre-processing is one of the most important methods in ML techniques. The pre-processing technique aims to reduce the time complexity in the training phase and also to improve the performance of the data. In our work, we focus on the pre-processing technique which is categorized into three phases. First,

converting the dataset into a numerical column using a LE-OHE is applied in section 2.1.1. Next, fill the missing values by using a hybrid FL-MF algorithm has been developed which is illustrated in section 2.1.2. Finally, identifying the outliers via the IQR technique is displayed in section 2.1.3.

2.1.1. Convert the dataset into a numerical column using a LE-OHE

Initially, consider the hepatitis dataset as the input which is taken from the UCI repository [14]. The dataset is based on 19 features that are split into the numerical column (NC) and categorical column (CC). The NC consists of 6 features and CC has 14 features. After identifying the columns, the novel hybrid LE-OHE method is applied to convert all the CC into NC which can reduce the time complexity in the training phase is briefly described in the following steps:

(i) The LE method is applied to change the CC into NC. This method classifies the attributes based on weight and encoder. The feature 'class' contains two attributes namely 'live' and 'die'. These attributes can classify the patient, 'live' is labelled as '0' and 'die' is labelled as '1'. Similarly, some of the features contain two attributes 'true' or 'false'. The 'true' represents the diseases present that are labelled as '1' and the 'false' are labelled as '0' based on the weight category it classifies the attributes. Next, classify based on the encoder which contains the feature 'sex' has two attributes (i.e. 'female' and 'male'). Here, classifying the attributes using weight is not possible because the preference of the attributes is the same. This is considered the drawback of the LE method. To overcome the issue, the OHE method is introduced to convert into the NC.

(ii) The OHE method converts the attributes into columns. Ex: male and female are the attributes that are converted into two columns. The first column is the female and the second column is the male. Fill the values by '1' or '0' depending on the attribute present. Assume, the dataset contains the first instance as female. The first female column is filled as 1 and the next male is filled as 0. Finally, all columns are converted into binary values.

2.1.2. Fill the missing values using hybridized fuzzy logic with the MF algorithm

After converting all the columns to a binary value, identify the missing values. The missing values are filled via FL to improve the accuracy level for prediction. The FL categorizes the attribute into four types of ranges namely low (L), medium (M), high (H), and very high (VH) as displayed in the Table. 1. The missing values are filled by a novel hybrid method FL-MF algorithm. It lies under four different cases given below.

Table. 1 The ranges lie under the category

CT	A	BI	AP	S	AB	P
L	7-24	0.3-1.9	26-96	14-98	2.1-3.5	0-39
M	25-45	2-3.9	96.1-147	98.1-182	3.6-4.8	39.1-58
H	46-65	4-5.9	147.1-194	182.1-278	4.9-6.4	58.1-78
VH	>65	6-8	194.1-295	278.1-648	>6.4	78.1-100

Case 1: All the feature value falls under any one category (i.e., 'L' or 'M' or 'H' or 'VH').

Consider all the features (i.e., age falls under the category 'L', Bilirubin (BI) falls under the category 'L', Alk-phosphate (AP) falls under the category 'L', Sgot (S) falls under the category 'L', albumin (AB) falls under the category 'L') falls under any one category and the remaining one feature fall under Null Value (NV) (i.e. Protime (P) falls under the category 'NV'). Based on the remaining feature value the missing null value is decided (i.e., the protime value null value is changed to 'L' category). After Identifying the range from the Table 1, (i.e. protime range lies between 0 to 39) then calculate the mean value to fill the null value using Eq. (1).

$$x' = \frac{\sum x_i}{n} \tag{1}$$

Where x' represents the mean value, $\sum x_i$ denotes the sum of all data points and *n* number of instances.

Case 2: Some of any feature values fall under the majority category

If the maximum column lies under the same category, then the null value can be replaced by the same category. E.g., Age contains 'M', Bilirubin contains 'L', Alk-phosphate contains 'M', Sgot contains 'null value', albumin contains 'M', and Protime contains 'H' etc. From the prediction, the number of the 'M' column is 3, the number of the 'L' column is 1 and the number of the 'H' column is 1. The above example concludes that the maximum column lies under the 'M'. After Identifying the range from Table. 1, (i.e., the protime range lies between 39.1 to 58) then calculate the mean value to fill the null value using Eq. (1).

Case 3: Except for the values of one feature remaining features are Null

In some cases, the age feature is alone present and the remaining features are null. Then, based on Table. 1 remaining feature is filled.

Case 4: All the feature value is scattered which is divided into two types: (i) If the missing value contains an equal category (i.e. two 'L' and two 'H') and (ii) If the missing value contains different category (i.e. one 'L', one 'H', one 'M' and two 'VH'). For the above two cases apply the MF algorithm to fill the null values. The MF algorithm is based on the following steps:

Step 1: Initially, the row population size is taken as *(u)* and its velocity (v_i) of a male is considered MF (x_i). Where similarly, the column population size is taken as *(n)* and its velocity (v_j) is considered a female MF (y_j). Where *(i=1,2,3,…,u)* and *(j=1,2,3,…,n)*.

Step 2: Evaluate the fitness function using Eq. (2).

$$fitness = \sqrt{\sum_{i,j}^{u,n}(x_i - y_j\ )^2} \tag{2}$$

Step 3: Find the global best MF among all (i.e.) (g_{best}) in the following Eq. (3).

$$g_{best_{ij}} = \min\ (fitness) \tag{3}$$

Step 4: Update the position for male and female MF xy_i^{t+1} . Adding the velocity $v_{i\,j}^{t+1}$ in the current position in Eq. (4).

$$xy_i^{t+1} = xy_{ij}^t + \ v_{i\,j}^{t+1} \tag{4}$$

Where, *t* denotes the number of iterations *(t=1,2,3,….20).*

Step 5: The current position is updated using the velocity $v_{i\,j}^{t+1}$ is substituted in Eq. (5).

$$v_{ij}^{t+1} = v_{ij}^t + \alpha_1\left(g_{best_{ij}} - xy_{ij}^t\right) \tag{5}$$

The α_1 is a positive attractive constant, that takes a random value from *0 to 1*.

Step 6: Compare the first and second iteration minimum value is considered as the g_{best}. Repeat steps 3 to 5 until reach the optimal solution. Finally, fill in the null value using g_{best} solution.

2.1.3. Noise filtering

The outlier is the process of grouping the values in the range (i.e., upper fence and lower fence) and ignoring the values which cannot be grouped. Identifying the outliers helps to improve the processing time in training the model which in turn increases the accuracy level. In this paper for predicting the outliers IQR method is employed to calculate the statistical dispersion of the data. The following steps are given below.

Step 1: After the process of pre-processing, the dataset contains filled rows and columns. First, take the column data and arrange them in ascending order.

Step 2: IQR process divides the column data into three quartiles R_1, R_2, and R_3. The first lowest quartile part is R_1, the second median quartile part is R_2 and the third highest quartile part is R_3.

Step 3: Calculate the *IQR* range using Eq. (6).

$$IQR = R_3 - R_1 \tag{6}$$

Here, the R_1 value is calculated using the first-half median value and R_3 is represented by the second-half median value.

Step 4: After finding the *IQR* range, then calculate the upper fence boundary around the third quartile. The exceeding values are denoted by outliers in Eq. (7).

$$Upper\ Fence = R_3 + (1.5 \times IQR) \tag{7}$$

Step 5: Next, find the lower fence around the boundary of the first quartile. The value less than the boundary value is taken as an outlier and is calculated using Eq. (8).

$$Lower\ Fence = R_1 - (1.5 \times IQR) \tag{8}$$

Step 6: The exceeding range values in the upper fence and lower fence are identified and removed.

2.2 Classification technique

After the completion of the pre-processing stage, all the missing data are filled in. In order to increase the precision rate, CDT is built. Further, the REPT pruning process is adopted with different hyperparameter tuning using a grid search technique. This is given as the input for the ensemble strategy. The ensemble method combines multiple classification models for more accuracy and reliable predictions.

2.2.1. Correlation-based decision tree

The noise-filtered dataset is given as the input for the decision tree (DT). To build the DT correlation coefficient weight is calculated for all features which lie from the range -1 to 1. The highest correlation value is taken as a root node, from the root node all

the possible attributes are taken as the child node. Then again calculate the correlation coefficient for each feature and fix the highest value as the root node. This process is repeated until it reaches the leaf node.

$$cc_{x_i z_j} = \frac{\sum(x_i - x_i')(z_j - z_j')}{\sqrt{\sum(x_i - x_i')^2 \sum(z_j - z_j')^2}} \quad (9)$$

Here x_i act as the feature value and x_i' represents the mean of the features, z_j denotes the class value, z_j' stands for the mean of the class. where $i = 1,2,3,\ldots 19$ and $j=1,2$. Consider, the sample input for calculating correlation coefficient weight is explained below:

Step 1: Initially, find the root rode using the correlation coefficient $cc_{x_i y_j}$ which is projected in Eq. (9). Where $x_1 = age,\ z_1 = class$. Apply the values in Eq. (9) $cc_{(age,class)} = \frac{(30-41.2)\times(2-1.79)}{\sqrt{(30-41.2)^2}\times\sqrt{(2-1.79)^2}} + \ldots \frac{(39-41.2)\times(2-1.79)}{\sqrt{(39-41.2)^2}\times\sqrt{(2-1.79)^2}} = --0.21$. Similarly, repeat this process for different $cc_{x_i z_j}$. Where, x_i= (Sex, Setorid, Antivirals, Fatigue, Malaise, Anorexia, Liver_big, Protime, Varices, Histology, Liver_firm, Spleen-palpable, Spiders, Ascites, Bilirubin, Alk_phosphate, Sgot, Albumin) and z_j = class. From the correlation weight, the highest weight is considered the root node. Here, albumin features are taken as the high value as shown in Fig. 2.

Step 2: In Fig. 2 the root node is fixed as the albumin and the square bracket represents the instance. Here, the albumin includes a total of 155 instances in which 123 instances denote the live class and 32 instances represent the die class. After finding the root node albumin, which contains the attributes such as L, M, H, and VH. Based on albumin range 2.1 to 3.5 the attribute instance value is fixed as low. From the dataset, the low attribute has a total of 42 instances, in that 22 instances are live and 20 instances are dying, class. Similarly, all other attributes such as high and medium are considered in Fig 2. In the dataset, the high attribute incorporates 4 live classes and no die class. In that scenario, the high attribute is fixed as the leaf node because there is no error. This data cannot split as the decision node.

Step 3: After the root node selection process, again the Correlation Coefficients for every attribute are calculated to fix the next decision node is assigned as DN_1 and DN_2. The CC of the highest value is recognized as the decision node and denoted as the subtree-1. The decision node DN_1 ascites which contain two attributes such as true and false are displayed in Fig. 2. Similarly, spiders DN_2 it also contains two attributes such as true and false values.

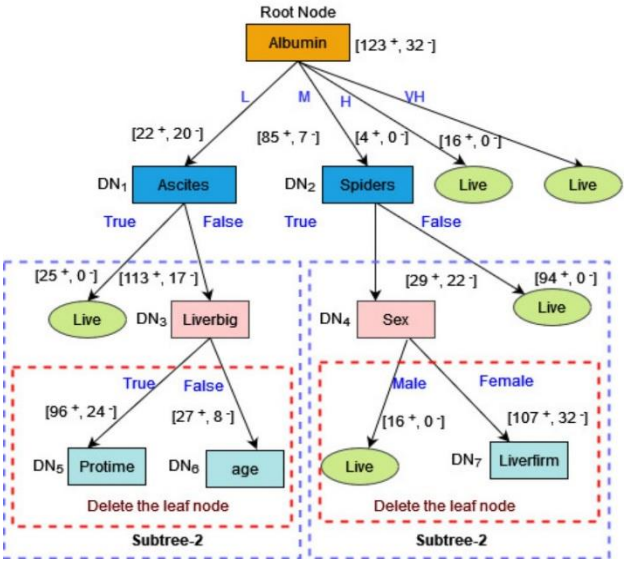

Figure. 2 Correlation-based decision tree

Step 4: The subtree-2 decision node DN_3 is fixed as the liver big that has two attributes such as true and false. Here the blue colour dotted box denotes the subtree-2 and the red colour dotted box represents the delete the leaf node. After finding a correlation, for true attribute highest value is set as the protime decision node DN_5 and for false attributes highest value is taken as the age. This process is repeated until it finds the leaf node. Similarly, for sex subtree- 2 decision node DN_4, it also encloses two attributes such as male and female values. The male contains only positive values (i.e. live) put as leaf nodes. After finding a correlation, for female attributes the highest value is chosen as the live firm decision node DN_7.

Step 5: Repeat steps 1 to 4 until all the decision nodes are converted as the leaf node.

2.2.2. REPT to minimize the error

RERT is the process to adjust the DT in such a way it minimizes the misclassification error and solves the overfitting problem. The REPT uses a pruning set which is assigned into rows and columns with classes. The goal of the REPT is to check the error estimate of the training data compared with the pruning data. The optimistic error estimate *e(g)* is given in Eq. (10) below.

$$e(g) = \frac{T}{n} \quad (10)$$

Where, *T* represents the total number of errors in the decision tree (i.e., Misclassified data) and *n* denotes the total number of instances. In REPT process takes place only if the error of the parent node is less than the error of the child node else it does not occur as shown in Eq. (11).

$$REPT = \begin{cases} pruning\ occurs\ (p), & if\ \ e(pn) < e(cn) \\ No, & otherwise \end{cases} \quad (11)$$

Here, p act as the change of the parent node into the leaf node, $e(pn)$ denotes the error rate of the parent node and $e(cn)$ refers to the error rate of the child node. In order to build the ensemble classification model, input is taken from the different REPTs using a grid search technique. The grid search is a hyperparameter tuning strategy used to find the best combination of hyperparameter values that are used for the input of different classification models. It exhaustively searches the tree values through a predefined set of hyperparameter values to generate the optimal hyperparameters. The ML model MLP_H is represented as Eq. (12).

$$MLP\_H = M(r, h) \quad (12)$$

Where M denotes the model, r is the input data of the REPT, h represents the hyperparameters of the model $(h_1, h_2, ..., h_N)$ with a corresponding list of possible values, N is the number of hyperparameters in the grid search. It avoids manual tuning and saves time in finding the optimal hyperparameters. Consider the example, the first hyperparameter h_1 denotes the maximum depth tree size=2 and the minimum sample split=2. Similarly, the second hyperparameter h_2 =8 and the minimum split=4. And the third hyperparameter is h_3=5 and the minimum split = 3. The values of each parameter vary based on classification output for each iteration to reach a better optimal solution.

2.2.3. Ensemble classification

Ensemble classification involves multiple models (e.g., SVM, kNN, NN) combines to predict the final decision. In this work, the bagging method is implemented to improve the accuracy and reduce the variance of multiple classifier models by voting predictions. The trimmed REPT h_1 is given as the input for the SVM classification process is projected in Fig 3. It helps in determining the ideal hyperplane to maximize the margin by separating the live class and die class. Here, moving the support vector plane makes the decision boundary for better classification. The L_1, L_2, and L_3 are the marginal hyperplanes, and a vector line perpendicular to three lines (i.e., L_1, L_2, L_3) is denoted as w and is called the weight vector. The centreline of the hyper-line is denoted in Eq. (13). Here b is the bias and x is the data points.

$$c = (w \times x) + b = 0 \quad (13)$$

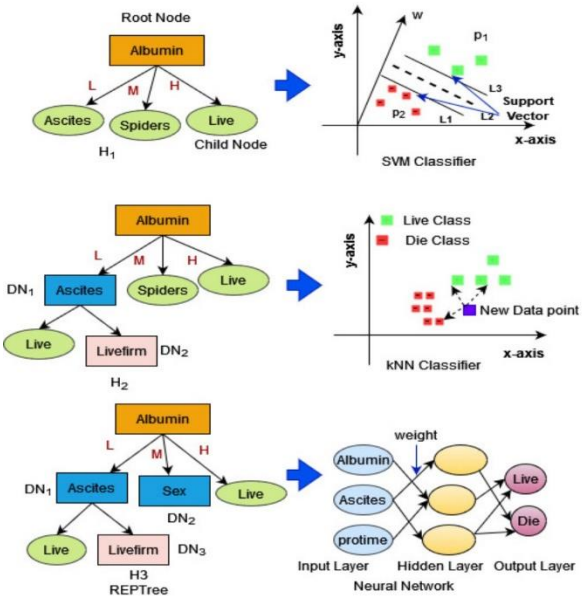

Figure. 3 Different REPT to ensemble classifier

If the expression c is equal to or greater than zero it is called a positive sample (i.e.) live otherwise it is called a negative sample (i.e.) die.

The trimmed REPT (h_2) is given as the input of the kNN classifier is represented in Fig 3. It is a simple classification algorithm that uses the majority class among the kNN to classify new data points. In a labeled training dataset with N data points such as (q_1, z_1), $(q_2, z_2), \ldots (q_n, z_n)$. Each q_i represents a feature vector, and z_i is the corresponding class label for each input q_i. The test data point which calculates the distance between the x_{query} and all input data points using the distance metric such as Euclidean distance is shown in Eq. (14).

$$\text{distance}(\text{x}_{\text{query}}, x_i) = \sqrt{\sum(\text{x}_{\text{query}} - x_i)^2} \quad (14)$$

Then select the k data points with the smallest distance to x_{query} and count the occurrences of each class label among the k selected data points. Finally, assign the class label of x_{query} based on majority voting. The trimmed REPT h_3 is given as the input of a NN classifier shown in Fig. 3. It is a powerful ML algorithm. Each neuron processes input data applies an activation function, and passes the result to the hidden layer to learn more complex patterns and relationships. The network's parameters, such as weights and biases, are learned through training using optimization techniques like backpropagation that decides the classification. The ensemble combines the strengths of both the REPT and NN, potentially leading to better overall performance.

2.2.3.1. Hard voting algorithm

After the classification process, the prediction of multiple individual classifiers (i.e.h_1, h_2, ... h_n) such as SVM, kNN, and NN combined to make an optimal solution using a hard voting algorithm. Each classifier takes an input of REPT and predicts a class label (z_i), where i is the index of the classifier. The predicted class labels can be either 0 or 1 (i.e.) live or die. Hard voting is an ensemble learning technique that predicts the output by majority voting that is chosen as the final prediction is represented as y_{final} using Eq. (15). Here, mode () represents the most frequent class label among the predictions. Table 2 summarizes the glossary of terms.

$$y_{final} = mode(h_1, h_2, \ldots, h_n) \quad (15)$$

3. Experimental results with discussion

In this paper, novel pre-processing techniques were introduced such as LE-OHE, a hybrid FL-MF optimization algorithm, and the IQR method was implemented with a classification technique containing techniques such as CDT, REPT using grid search algorithm, and Ensemble strategy. These methods are used to predict diseases in the early stages to decrease the mortality rate. The two input datasets are considered for experimental purposes; they are (i) the hepatitis dataset [14], and (ii) ILPR datasets [15] are taken from the UCI repository. The hepatitis dataset has a total number of 155 instances and 19 attributes. The ILPR dataset contains 583 represents the patient's records which have some missing values, redundant data, and outliers presented. To overcome these issues, the LE-OHE, a hybrid FL-MF optimization algorithm, and the IQR method are introduced. In order to classify the data easily, the decision tree representation is used. Moreover, to reduce the error rate and processing time REPT is involved. To the extent of tuning the hyperparameter, split the REPT tree using a grid search algorithm that is given as the input to the ensemble mechanism for further classification.

In this experiment, the existing methods namely min max scalar- support vector machine (MMS-SVM) [5], mean imputation-random forest (MI-RF) [6], gradient boosting (GB) [8], random forest (RF) [7], ensemble-stacking (ES) [12], extreme gradient boosting (XGBoost) [13], and radial basis function-extreme learning machine (RBF-ELM) [9] are compared with the proposed method using 10 different metrics. The various metrics used for evaluating the performance to predict the algorithm in the machine learning approach are given below in Eq. (16-22). The metrics are accuracy [1], precision value (PV) [2], recall [4], f-measure [1], AUC [5], processing time [5], mean absolute error (MAE) [6], root mean squared error (RMSE) [2], relative absolute error (RAE) [7], and root relative squared error (RRSE) [11]. It also includes the attribute of true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN). The accuracy measures the error difference between the predicted instance and with actual instance (i.e.) correctly classified instance divided by the total number of instances is calculated using the formula in Eq. (16).

$$Accuracy = \frac{TP}{TP+TN} \times 100 \quad (16)$$

Table 2. Glossary of terms

Symbol	Description
n	Total number of instances
U	Row population size (Total features)
v_i	Velocity
x_i	Row data values where *(i=1,2,3,….u)*
y_j	Column data values where *(j=1,2,3,…,n)*
x_i'	Mean of the features
$g_{best_{ij}}$	Global best value of Mayfly algorithm
t	Number of iterations where (*t=1,2,..20*)
xy_i^{t+1}	Update the new position
α_1	positive attractive constant
R_1	The first lowest quartile par
R_2	The second median quartile part
R_3	The third-highest quartile part
z_j	Class Value where(*j=1,2)*
z_j'	Mean of the class
$cc_{x_iy_j}$	Correlation Coefficient
$e(g)$	Error Estimate
P	Change of parent node into the leaf node
$e(pn)$	The error rate of the parent node
$e(cn)$	The error rate of the child node.
M	Model
r	Input data of the REP tree
H	Hyperparameters of the model where *(h_1, h_2, ..., h_N)* corresponding list
N	Number of hyperparameters in the grid search
b	Bias in the SVM classifier
c	The centreline of the hyper-line in SVM
W	Weight vector in the SVM
q_i	Feature vector where (i=1,2,….n)
x_{query}	New data point in the kNN classifier
K	The smallest distance to the new points
e_i	Predicted Value
f_i	Observed Value
o_i	Actual Value
f_i'	Mean of the observed value
y_{final}	Final predicted value

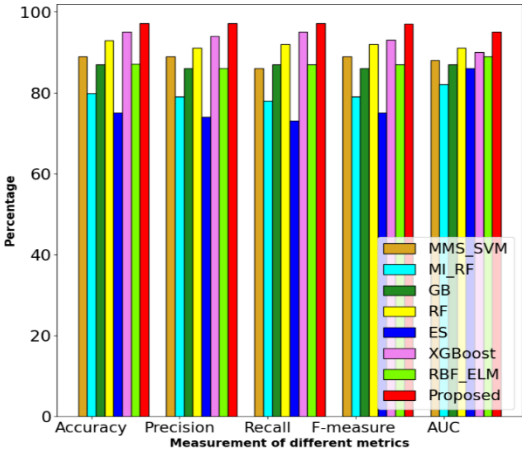

Figure. 4 Performance evaluation with different metrics using the hepatitis dataset

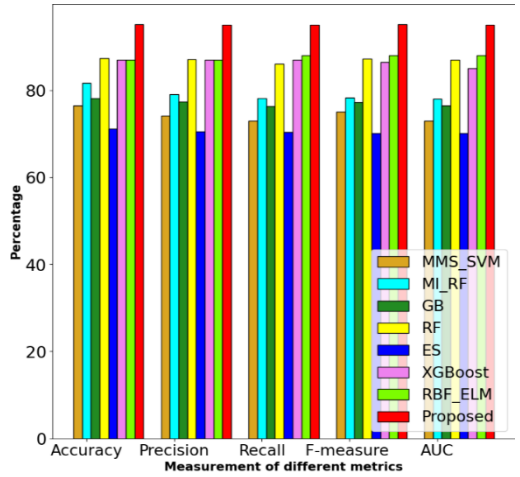

Figure 5. Performance evaluation with different metrics using the ILPR dataset

Here, the TP denotes the number of positive patient records correctly identified and TN represents the correctly predict negative class.

The precision parameter is used to predict the proportion of positive classes with predicted positive classes in the patient record as described in Eq. (17)

$$Precision = \frac{TP}{TP+FP} \tag{17}$$

Where FP acts as the number of patient records incorrectly classified.

The recall parameter is described as some positive patient records correctly identified divided by the total number of positive classes as shown in Eq. (18)

$$Recall = \frac{TP}{TP+FN} \tag{18}$$

Where FN acts as the predicted wrongly in the dataset. The parameter F-measure is estimated using the equally weighted with precision and recall presented in the given Eq. (19)

$$F_{measure} = \frac{2}{\text{Precision}^{-1}+\text{Recall}^{-1}} \tag{19}$$

This work is compared with existing work such as MMS-SVM [5], MI-RF [6], GB [8], RF [7], ES [12], XGBoost [13], and RBF-ELM [9]. In Fig. 4 the different metrics namely accuracy [1], PV [2], recall [4], F-measure [1], and AUC [5] are used to evaluate the performance. The MMS-SVM algorithm [5], is used to minimize the feature in order to make it more durable and reliable. The hybridized approach may boost the time complexity of the framework's construction, and also training and modifying the hybrid system might be difficult. The metrics accuracy (89%), precision (89%), recall (86%), f-measure (89%), and AUC performance values are less due to the turning parameter values c=100, g=0.0001 being set low. The GB [8] has various strategies for predicting hepatitis disease death rates. The dataset's quality is low due to its incompleteness, resulting in an impact on the performance evaluation of AUC, which obtained a minimum average accuracy score level of 87 %. The MI-RF [6] boosts interpretability and reduces the model complexity. This method works with various data distributions and variances, making it suited for dealing with huge patient populations. The improper validation and regularization approaches reduce the accuracy rate. The ES [12] approach, integrates the meta-model by utilizing a genetic algorithm that will overcome the scaling challenges. The usage of many classifiers improves performance. One of the key limitations is that the algorithm's execution time is too long due to the improper pre-processing technique. The above two techniques [6] [12] accuracy level (79.8% and 75.1%) is low due to the elimination of the prominent features. The XGBoost [13] method is capable of handling big datasets and performing well in high-dimensional feature spaces. Although this work did not focus on pre-processing approaches, the accuracy level is low due to the incorrect feature selection. In RF [7] algorithms complicated patterns in medical data, resulting in higher accuracy than standard diagnostic approaches. If the data is inadequate, biased, or contains errors, the accuracy of early detection may suffer. The XGBoost [13] and RF [7] algorithms achieve moderate performance in the following metrics: accuracy (95% and 92.9%), PV (94% and 91%), recall (95% and 92%), f-measure (93% and 92%), and AUC (90%) and (91%) thereby not using the prominent pre-processing technique. The proposed method adopts pre-processing techniques to clean the data efficiently by using the techniques such as LE-OHE, FL-MF,

Table 3. Reveals the error prediction using hepatitis and ILPR datasets

	MAE		RMSE		RAE		RRSE	
Algorithm	**Hepatitis**	**ILPR**	**Hepatitis**	**ILPR**	**Hepatitis**	**ILPR**	**Hepatitis**	**ILPR**
MMS-SVM [5]	0.21	0.35	0.21	0.37	60.37	78.13	80	85
MI-RF [6]	0.19	0.24	0.43	0.59	59.92	74.6	107.88	109.6
GB [8]	0.21	0.20	0.32	0.45	66.37	86.3	81	90
RF [7]	0.18	0.19	0.32	0.39	59.37	74.2	78	86
ES [12]	0.2	0.23	0.37	0.43	25.57	35.4	75.09	95
XGBoost [13]	0.36	0.18	0.28	0.35	68.25	86.6	116.89	120.5
RBF-ELM [9]	0.18	0.19	0.32	0.45	54.64	72.6	79.46	92.6
Proposed	0.02	0.16	0.19	0.30	14.25	35.4	55.26	72.5

and IQR methods. Moreover, the bagging ensemble technique gives a better accuracy rate of 97.1%.

In Fig. 5, MMS-SVM [5] algorithm generates low accuracy (76.51%) compared to the other algorithm due to the poorly selected pre-processing technique. The normalization method ensures that all feature weight values are equally given to the SVM model to avoid dominance of the feature. The MI-RF [6] algorithm gives moderate accuracy (81.65%) when the raw data is handled via hyperparameter optimization. The GB [8] algorithm which contains accuracy (80%) due to imputing the missing values using the multivariate imputation by chained equations (MICE) method which created data leakage from training samples to testing MICE is applied. The RF [7] algorithm shows (87.35%) accuracy because it works on different decision trees as it predicts the label using priority wise in training data. ES [12] algorithm which shows lower than the other algorithm as it needs tuning parameters to achieve optimal performance. XGBoost [13] algorithm contains accuracy (87%) that trains to build the multiple base models to predict the performance. The weighted quantile sketch efficiently handles the missing data and decreases the iterative process. The RBF-ELM [9] algorithms (87.1%) give moderate performance due to the usage of kernel functions to map the input data into a higher-dimensional space which makes it to learn the complex patterns input. Thereby, usage of traditional complex kernel structure may lead the process tedious. Our proposed work is superior to an existing algorithm due to utilizing the optimization algorithm (i.e.) the MF algorithm which produces the optimal filling process. Moreover, it includes fuzzy logic to predict the missing values that improve the accuracy level (95.1%) to a greater extent.

3.1 Error prediction

The MAE [6], RMSE [2], RAE [7], and RRSE [11] are used to measure the error rate of the model. The MAE [6] calculates the mistakes by observing data with the expected values. Where e_i acts as the predicted value, f_i denotes the observed value, and n represents the total number of instances (i.e., patient records) as shown in Eq. (20).

$$MAE = \frac{\sum_{i=1}^{n}|e_i - f_i|}{n} \tag{20}$$

Another error metric RMSE [2] is used to measure the model prediction error. Here c_i represents the actual value shown in Eq. (21).

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(e_i - c_i)^2} \tag{21}$$

The RAE [7] metric calculates the error in different units projected in Eq. (22). Where, $f_{i'}$ is the mean of the observed value.

$$RAE = \frac{\sum_{i=1}^{n}|e_i - f_i|}{\sum_{i=1}^{n}|f_{i'} - f_i|} \tag{22}$$

MMS-SVM [5], in both the dataset are error rates in the parameter MAE [6], RMSE [2], RAE [7], and RRSE [11] higher due to the irrelevant feature selection and finding the best hyperplane create a challenging task. The MI-RF [6], RBF-ELM [9], and RF [7] techniques generate an average error rate in all datasets when the model learns to perform well on the training data but fails to predict unknown data-id shown in Table 3. Also, it produces the average error rate because of the improper detection of outliers in the dataset. The XGBoost [13] algorithm gives a high error rate because there is no proper tuning parameter, it causes the overfitting problem. Also, the error rate increases due to the lack of pre-processing technique. The ES [12] algorithm yields a moderate error rate due to data preprocessing steps such as scaling, and imputation is applied to the entire dataset before splitting it into training and testing sets. In the proposed technique CDT and REPT cut down

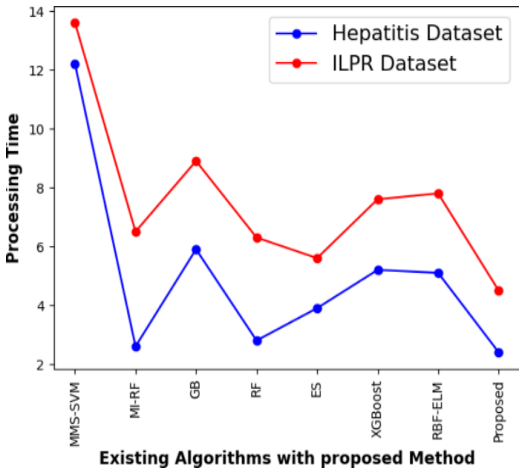

Figure. 6 Time complexity using hepatitis with ILPR datasets

the error rate when compared to the existing algorithm. Moreover, the ensemble bagging classifier reduces variance and overfitting which makes the dataset more reliable.

3.2 Processing time

In this section, the computational time of existing techniques is compared with the proposed work. Fig. 6 shows the different processing times using existing algorithms namely MMS-SVM [5], MI-RF [6], GB [8], RF [7], ES [12], XGBoost [13], and RBF-ELM [9]. In MMS-SVM [5] algorithm consumes more processing time of 12.2 sec in the hepatitis dataset. For the same existing algorithm, the ILPR dataset achieves a processing time of 13.6 sec for the selection of the right kernel, and tuning hyperparameters is a challenging task that increases the time for execution. Moreover, the redundant values are not detected, and also improper pre-processing technique is selected may higher the processing time. In GB [8], XGBoost [13], and RBF-ELM [9] it takes an average processing time of 5.9 sec, 5.2 sec, and 5.1 sec in the hepatitis dataset. Also, the same existing algorithms yield the highest processing time of 8.9 sec, 7.6 sec, and 7.8 sec in the ILPR dataset. Here, the input is split into multiple numbers of trees that are given training to the base model for classifying the classes. Thereby, each tree built sequentially enlarges the time complexity. In MI-RF [6], RF [7], and ES [12] algorithms produce low processing times approximately 2.1 sec, 2.8 sec, and 3.9 sec. For the same existing algorithm, the ILPR dataset performs 6.5 sec, 6.3 sec, and 5.6 sec because building the multiple trees and improper selection of outliers has a higher processing time. The initial data set is converted to binary data and redundancy data from the dataset is removed for classification. Further, CDT and REP tree techniques are employed to make the classification process simple thereby reducing the data set size which in turn reduces the processing time of 2.4 sec for the hepatitis dataset and 4.5 sec for the ILPR dataset.

4. Conclusion

This work focused on data pre-processing with classification methods implemented to enhance performance. Initially pre-processing approaches such as LE-OHE, FL-Mayfly algorithm, and IQR methods are implemented to fill in missing values and detect the outliers. Consequently, a correlation-based decision tree is built based on the weight makes performing the classification process much easier. Finally, the REPT process is employed to separate different combinations using a grid search algorithm. The splitted REPT is given the input of ensemble classifiers such as SVM, kNN, and NN to minimize the error rate and avoid the overfitting problem. The different metrics used for evaluation are accuracy, PV, MAE, RMSE, and RAE. The performance of the proposed technique outperforms well when compared with the other existing techniques such as MMS-SVM, MI-RF, GB, RF, ES, XGBoost, and RBF-ELM. The proposed method achieves better results in the accuracy of the hepatitis dataset 97.1% and the ILPR dataset 95.1%. The average precision and MAE error rate of the two metrics are 96% and 18%. Moreover, the proposed method has less execution time for both datasets (i.e., hepatitis dataset is 2.4 sec and for ILPR dataset is 4.5 sec). In the future, feature selection methods with different types of hepatitis datasets will be employed to reach the optimal solution.

Conflicts of interest

The authors declare no conflict of interest.

Author contributions

Conceptualization, methodology, software, formal analysis, resources, data curation, and writing-original draft preparation, writing-review, editing: C. Saranya Jothi, and Supervision: D. Umanandhini

Reference

[1] M. Nilashi, H. Ahmadi, L. Shahmoradi, O. Ibrahim, and E. Akbari, “A predictive method for hepatitis disease diagnosis using ensembles of neuro-fuzzy technique”, *Journal of Infection and Public Health*, Vol. 12, No 1, pp. 13-20, 2019.

[2] Sartakhti, J. Salimi, M. H. Zangooei, and K. Mozafari, “Hepatitis disease diagnosis using novel hybrid method based on support vector machine and simulated annealing (SVM-SA)”,

Computer Methods and Programs in Biomedicine, Vol. 108, No. 2, pp. 570-579, 2012.
[3] B. Gao, T. C. Wu, S. Lang, L. Jiang, Y. Duan, D. E. Fouts, X. Zhang, X. M. Tu, and B. Schnabl, "Machine learning applied to omics datasets predicts mortality in patients with alcoholic hepatitis", *Metabolites*, Vol. 12, No. 1, p. 41, 2022.
[4] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data Preprocessing for Supervised Learning", *International Journal of Computer Science*, Vol. 1, No. 2, pp. 111-117, 2006.
[5] A. U. Haq, J. P. Li, M. H. Memon, S. Nazir, and R. Sun, "A Hybrid Intelligent System Framework for the Prediction of Heart Disease Using Machine Learning Algorithms", *Mobile Information Systems*, pp. 1-21, 2018.
[6] R. Saxena, S. K. Sharma, M. Gupta, and G. C. Sampada, "A Novel Approach for Feature Selection and Classification of Diabetes Mellitus: Machine Learning Methods", *Computational Intelligence and Neuroscience*, 2022.
[7] R. Yilmaz and F. H. Yagin, "Early detection of coronary heart disease based on machine learning methods", *Medical Records*, Vol. 4, No.1, pp. 1-6, 2022.
[8] B. Gao, T. C. Wu, S. Lang, L. Jiang, Y. Duan, D. E. Fouts, X. Zhang, X. M. Tu, and B. Schnabl, "Machine learning applied to omics datasets predicts mortality in patients with alcoholic hepatitis", *Metabolites*, Vol. 12, No. 1, p. 41, 2022.
[9] R. Siouda, M. Nemissi, and H. Seridi, "Diverse activation functions based-hybrid RBF-ELM neural network for medical classification", *Evolutionary Intelligence*, pp. 1-17, 2022.
[10] M. A. Quadir, S. Kulkarni, C. J. Joshua, T. Vaichole, S. Mohan, and C. Iwendi, "Enhanced Preprocessing Approach Using Ensemble Machine Learning Algorithms for Detecting Liver Disease", *Biomedicines*, Vol. 11, No. 2, p. 581, 2023.
[11] M. Nilashi, H. Ahmadi, L. Shahmoradi, O. Ibrahim, and E. Akbari, "A predictive method for hepatitis disease diagnosis using ensembles of neuro-fuzzy technique", *Journal of Infection and Public Health*, Vol. 12, No. 1, pp. 13-20, 2019.
[12] R. Sikora, "A modified stacking ensemble machine learning algorithm using genetic algorithms", *Handbook of Research on Organizational Transformations Through Big Data Analytics*, IGi Global, pp. 43-53, 2015.
[13] A. Alizargar, Y. L. Chang, and T. H. Tan, "Performance comparison of machine learning approaches on Hepatitis C prediction employing data mining techniques", *Bioengineering*, Vol. 10, No. 4, p. 481, 2023.
[14] https://archive.ics.uci.edu/ml/datasets/hepatitis
[15] https://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset