Journal of Computer Science

Original Research Paper

Diagnosis of Hepatitis Disease with Logistic Regression and Artificial Neural Networks

[1,2]Alaa M. Elsayad, [1,3]Ahmed M. Nassef and [4]Mujahed Al-Dhaifallah

[1]Department of Electrical Engineering, College of Engineering at Wadi Addawaser, Prince Sattam Bin Abdulaziz University, KSA
[2]Department of Computers and Systems, Electronics Research Institute, Giza, 12622, Egypt
[3]Department of Computers and Automatic Control Engineering, Faculty of Engineering, Tanta University, Egypt
[4]Department of Systems Engineering, King Fahd University of Petroleum & Minerals, Dhahran, 31261, KSA

Article history
Received: 11-12-2019
Revised: 21-02-2020
Accepted: 25-03-2020

Corresponding Author:
Alaa M. Elsayad
Department of Electrical Engineering, College of Engineering at Wadi Addawaser, Prince Sattam Bin Abdulaziz University, KSA
Email: a.elsayyad@psau.edu.sa

Abstract: Hepatitis C refers to the inflammatory state of the liver caused by viruses, bacteria, fungi, and exposure to toxins such as alcohol and self-immunity. The diagnosis requires investigating many laboratory tests and comparing the results to those of the former patients with the same conditions. This study presents the results of our experiments to build a hybrid system that combines both neural networks and logistic regression for the diagnosing of the hepatitis dataset using clinical and laboratory test results. The first experiment compared the performances of Multilayer Perceptual Neural Networks (MLPNN) and Radial Basis Function Neural Network (RBFNN) versus the conventional and stepwise Logistic Regression (LR) algorithms, where the results demonstrated the ability of neural networks to deliver better performance than LR models. In the second experiment, the features selected by backward and forward LR models have been evaluated for the improvement of the performances of MLPNN and RBFNN models. The hepatitis dataset was downloaded from the machine-learning repository by the University of California at Ervine. Missing values have been imputed with a separate Classification and Regression Tree (C&RT) for each attribute. Classification models have been evaluated in terms of statistical accuracy, specificity, sensitivity, F1-score and the Area Under the Receiver Operating Characteristic Curve (AUCROC). Experimental results showed that the performances of neural network models have been improved when employing stepwise LR models to select only the predictive attributes. The hybrid system which combined both backward stepwise LR for attribute selection and MLPNN for classification has outperformed other systems in the diagnosis of the hepatitis dataset with 0.973 AUCROC for the training subset and 0.886 for the test one.

Keywords: Hepatitis Dataset, Stepwise Logistic Regression, Attribute Selection, Multilayer Perceptron Neural Networks, Radial Basis Function Neural Networks

Introduction

The liver is the largest and heaviest organ of the human body (Cohen, 1999). Its biological functions are to process nutrients from food, make bile, remove toxins from the body and build proteins. Hepatitis represents the greatest danger that causes chronic liver disease. It refers to an inflammatory condition of the liver, which is commonly caused by at least six different viruses (Daniel, 2018). Blood transfusions, tattoos, and piercing, drug abuse, hemodialysis, health workers, sexual contact with hepatitis carriers are some of the factors that increase the risk of infection (James and Foley, 2018; MayoClinic, 2018). According to the Global Hepatitis Report, it caused approximately 1.34 million deaths in 2015 only and the danger is rapidly growing every year (Taylor, 2003). Infected patients often have some symptoms such as poor appetite, nausea, vomiting, fever, pain in the upper right part of the abdomen (where the liver is located) and jaundice (WHO, 2017). Usually, hepatitis is diagnosed through a routine blood donation or during blood screening. So far, many studies have been performed in the diagnosis of hepatitis diseases, which are mostly done by expert physicians. They have

to conduct several checking strategies including medical and laboratory tests and then compare the results to other past patients with the same conditions. From the literature, it has been found that data mining algorithms can assist physicians to improve their medical decisions. Recently, some researchers wonder about the possibility of substituting normal human physicians with artificial intelligence tools in the near future (Jiang *et al*., 2017).

The data mining algorithm refers to the automated extraction of hidden, unknown and hypothetically valuable information from a large dataset. Different data mining models work to fetch and interpret the valuable information based on multidisciplinary fields such as statistics, artificial intelligence, machine learning, database management, etc., Gullo (2015). In general, different predictive models have different prediction capabilities that depend on the type of data and how it is preprocessed (Patel *et al*., 2009). This study aimed to develop a hybrid model that combines both statistical Logistic Regression (LR) and advanced Neural Network (NN) techniques for the diagnosis of hepatitis dataset as "Die" and "Live" state. Both LR and NN have many great characteristics that led to their widespread and use in many bioinformatics and biomedical applications. The purpose was to combine both methods to build a hybrid system for the purposes of attribute selection and record classification of hepatitis data. At first, the efficiency of conventional and stepwise LR models, as well as two commonly used neural networks; namely Multilayer Perceptron Neural Networks (MLPNN) and Radial Basis Function Neural Networks (RBFNN), were evaluated and compared on the hepatitis dataset. Then both forward and backward stepwise LR models were employed to select only the sets of predictive attributes to construct efficient neural network models.

LR algorithm models the probability of an event in terms of suitable explanatory attributes with no prior assumptions about the distributions of these attributes (Sperandei, 2014). Stepwise attribute selection is an approach to construct regression models in which the selection of predictive attributes' subset is carried out using an iterative procedure during the learning process. In each iteration, an attribute is considered for addition to or removal from the prediction process based on some prespecified criterion. Stepwise regression provides the ability to handle large amounts of possible attributes and to configure the model precisely (Hosmer and Lemeshow, 2000; Maxwell and Obinna, 2018). The order in which the attributes are added or removed can provide valuable information about their significances. In general, LR models are well-established statistical model and their coefficients can have clear clinical explanations. On the other hand, Artificial Neural Networks (ANNs) are prediction tools based on nonlinear models that are trying to mimic the brain's neurons network concept (Haykin, 2009). They gain increasing popularity for their flexibility and high accuracy in complex data modeling. They are currently occupying the second rank among the most widely used methods in medical applications (Jiang *et al*., 2017; Lancashire *et al*., 2009; Amato *et al*., 2013). ANNs were found to outperform traditional statistical regression methods in different biomedical applications. In a recent study published in 2019, the performances of ANN and LR have been evaluated for the diagnostic prediction of giant cell arteritis where ANN had higher sensitivity and accuracy than the LR, with a 17% lower FN rate (Ing *et al*., 2019).

In general, they are parallel algorithms, which consist of small processing units (neurons) organized in layers and connected through several links (weights). Haykin (2009) introduced in detail the most common ANN methods in terms of their structures, mathematics and potential benefits. Multilayer Perceptron Neural Network (MLPNN) and Radial Basis Function Neural Network (RBFNN) is a commonly used Artificial Neural Network (ANN) method (Lancashire *et al*., 2009). Both methods are feed-forward networks. However, they process the data with different mechanisms. RBFNN clusters the data into hyperspheres; while the MLPNN arbitrarily shapes the data into hypersurfaces. This study compared the predictive performance of both networks in the prediction of the hepatitis dataset.

The remainder of the paper is organized as follows: Section 2 presents all the hepatitis dataset in detail as well as a literature review. Section 3 introduces the mathematical models of LRs and ANNs. Section 4 illustrates the statistical measures used to evaluate the classification performance. In section 5, all the experimental results and discussion are introduced. Finally, the conclusions and acknowledgments are presented.

Hepatitis Dataset and Literature Review

The hepatitis dataset is downloaded from the UCI machine learning repository (Blake and Merz, 1996). This data can be employed to construct predictive models to classify whether a hepatitis patient "Live" or "Die". There are 155 observations; 32 cases belong to the ''Die'' class while the remaining 123 belong to the ''Live'' one with a class distribution ratio of 1:4 approximately. This biasing distribution increases the prediction challenge towards effective model construction. Each observation has 19 input attributes aside with them the corresponding output label. The list of attributes is shown in Table 1. There are 13 attributes with binary values and 6 have enumerated values. The dataset contains a lot of missing values. Table 1 shows the percentage of missing values for every attribute. The PROTIME attribute is the one that has the highest missing data values of 43.23%.

Table 1: Descriptions of hepatitis attributes (Cohen, 1999; Daniel, 2018; Blake and Merz, 1996)

Attribute	Description	Values	Percentage of missing data
AGE	Patient (case) Age	10, 20, 30, 40, 50, 60, 70, 80	0.00%
SEX	Gender	Male, Female	0.00%
STEROID	Response to corticosteroids treatment	Yes, No	0.65%
ANTIVIRALS	Response to antiviral treatment	Yes, No	0.00%
FATIGUE	Fatigue is the most commonly encountered symptom in patients with liver disease	Yes, No	0.65%
MALAISE	A general feeling of discomfort.	Yes, No	0.65%
ANOREXIA	An eating disorder.	Yes, No	0.65%
LIVER BIG	Enlarge liver	Yes, No	6.45%
LIVER FIRM	The liver is typically palpable and firm, with a blunt edge	Yes, No	7.10%
SPLEEN PALPABLE	When a spleen is felt via external examination.	Yes, No	3.23%
SPIDERS	type of telangiectasis (swollen blood vessels) found slightly beneath the skin surface,	Yes, No	3.23%
ASCITES	Abnormal buildup of fluid in the abdomen	Yes, No	3.23%
VARICES	Large blood vessels in the esophagus.	Yes, No	3.23%
BILIRUBIN	Level of bilirubin in the blood	0.39, 0.80, 1.20, 2.00, 3.00, 4.0	3.87%
ALK PHOSPHATE	Level of Alkaline phosphatase enzyme	33, 80, 120, 160, 200, 250	18.71%
SGOT	Level of Serum glutamic-oxaloacetic transaminase "liver enzymes"	13, 100, 200, 300, 400, 500,	2.58%
ALBUMIN	Level of Albumin protein made by the liver.	2.1, 3.0, 3.8, 4.5, 5.0, 6.0	10.32%
PROTIME	Prothrombin time in seconds	10, 20, 30, 40, 50, 60, 70, 80, 90	43.23%
HISTOLOGY		Yes, No	0.00%

The majority of published literature included the removal of records with missing values and only a few studies attempted to impute these values (Kaya and Uyar, 2013). However, some recent studies employed both methods. They removed the records that contain a lot of missing data and imputed those contain few ones. In (Borah and Nath, 2018), the authors firstly removed the PROTIME attribute and then removed all observations with more than 25% missing values and finally they imputed the rest with the mode value of the respective attribute. They used a rare association rule to build a medical diagnosis system by studying the infrequent correlations between dissimilar patient characteristics and diseases. Support Vector Machine (SVM) models have been used several times to classify the hepatitis dataset (Kaya and Uyar, 2013; Afif *et al*., 2013; Sartakhti *et al*., 2012; Chen *et al*., 2011). There are two differences between these studies in terms of the attribute selection methods and optimization methods. For example, in (Sartakhti *et al*., 2012), the authors used the simulated annealing algorithm to optimize the SVM to find the best box constraint parameter and sigma. They also compared their proposed system to other published classification methods. In (Mitra and Samanta, 2015; Çetin *et al*., 2015), artificial neural networks were applied for the diagnosis of hepatitis. All of them were using MLPNNs. However, in (Ansari *et al*., 2011), the performances of MLPNN are compared to the Generalized Regression neural network (GNRR) and Self-Organizing Feature Map neural network (SOFM). They reported that MLPNN and GNRR have diagnosed the dataset efficiently as compared to SOFM. Very recent works published in (Bhargav *et al*., 2018; Nilashi *et al*., 2019). In (Bhargav *et al*., 2018) the authors compared the performances of four classification algorithms namely; logistic regression, decision tree, linear support vector, and naive Bayes applied to classify the hepatitis dataset in terms of accuracy, precision, recall, and F1-score. They concluded that the ordinary logistic regression algorithm achieved the best accuracy of 87.17%. To the best of the authors' knowledge, stepwise logistic regression models have used neither for the diagnosis nor for the attribute selection of the hepatitis dataset. Nilashi *et al*. (2019), a predictive method for hepatitis diagnosis has been developed using neuro-fuzzy techniques.

Modeling Algorithms

Logistic Regression (LR)

LR is considered a parametric model that able to capture only a linear decision boundary between two classes (Sperandei, 2014). It uses the maximum log-likelihood principle to build a binary classification model that maps a d-dimensional input attributes vector x to one of two possible classes, i.e., $y \in \{0,1\}$. The attribute can be either nominal, ordinal, interval or ratio scale, while the output response can only take a binary value. In this modeling scheme, the input-output relationship is nonlinear. LR is considered as a special case of the generalized linear model (Hosmer and Lemeshow, 2000). That is since the output response is binary, the conditional distribution does not follow the Gaussian distribution but it follows Bernoulli distribution instead. The main task of LR modeling algorithm is to find a linear decision boundary (hyperplane) that partitions the attributes' space at the point where the probability of the outcome ($y = 1$) equals 50%, i.e:

$$p(y=1|x)=p(y=0|x) \tag{1}$$

The model uses the linear regression optimization algorithm to fit the logarithmic (sigmoid) form of the odds ratio as follows:

$$\log\frac{p(y=1|x)}{p(y=0|x)}=\log\frac{p(y=1|x)}{1-p(y=1|x)}=\beta^T x \tag{2}$$

where β^T is the transpose of the coefficients vector.

The log of the odds ratio is called the *logit* function which linking the output of the linear model to the actual outcome. The use of *Logit* function is computationally easier than the use of normal distributions (Sperandei, 2014). Equation (2) can be reformulated to find the probability of the outcome directly using the logistic (*Sigmoid*) function:

$$p(y=1|x)=\frac{e^{\beta^T}x}{1+e^{\beta^T}x} \tag{3}$$

The decision hyperplane is described by a linear function with the coefficient vector β. Then, the LR algorithm searches for the optimal d-dimensional vector $\beta = [\beta_0, \beta_1,\ldots, \beta_d]$ that best fits the training observations such that:

$$\begin{aligned}&logistic\ function\ \phi(z)=\frac{1}{1+e^{-z}}\\&Where\ z=\beta_0+\beta_1 x_1+\cdots+\beta_d x_d,\\&y=\begin{cases}1, & z>0\\0, & otherwise\end{cases}\end{aligned} \tag{4}$$

The schematic diagram of the LR algorithm and its logistic function (sigmoid) are shown in Fig. 1.

The decision boundary can be found using all data points $X\in\mathbb{R}^d$ such that:

$$\beta_0+\beta_1 x_1+\cdots+\beta_d x_d=0 \tag{5}$$

The LR classifier assigns x to class (1) if $\beta^T x$ is positive otherwise it is assigned to the other class (0) and the LR classification rule becomes:

$$y=1_{p(x,\beta)>0.5} \tag{6}$$

The values of βs are determined by maximizing the Log-likelihood function with an iterative numerical algorithm (Hosmer and Lemeshow, 2000).

Forward and backward stepwise selection methods are two different algorithms used to provide reliable LR models as well as to reduce the model complexity (Maxwell and Obinna, 2018). In the forward stepwise algorithm, the model is built by moving forward iteratively where the initial model has only the constant term, β_0. Then, the attributes are added one-by-one to the model. At each iteration, the attributes that are still not used in building the model are evaluated and the one that gives the best improvement is added. This iterative process is continued until the best candidate attribute does not produce any significant difference and the final model is generated. On the other hand, the backward stepwise algorithm is an elimination method, where the model is built first with all attributes. Then, iteratively, the algorithm removes the attributes that do not influence the improvement of model performance. The process continues until no more attributes can be eliminated and the final model is generated.

Multilayer Perceptron Neural Network (MLPNN)

MLPNN is one of the most popular modeling tools in classification and regression applications. It has the ability to approximate complex and nonlinear functions effectively with no prior assumptions to the model characteristics or data distribution (Haykin, 2009). Figure 2 shows the structure of the MLPNN model with only one hidden layer. It is organized in a feed-forward structure where the stream flows from inputs, forwards through hidden layers and finally reaching the output layer. The input layer delivers the weighted input data to the hidden neurons. Hidden and output neurons accumulate their input data after multiplying them by the appropriate strengths of the respective connection weights. Then, the neuron fires its output using the activation function. The mathematical description of an artificial neuron can be written as in Equation (7):

$$y_i = f\left(\sum w_{ij}x_i\right), \tag{7}$$

Where:
y_i = The output
f = The activation function
x_i = The input and
w_{ij} = The weight

The activation function f may have a linear or nonlinear form. Back-Propagation (BP) is a popular learning algorithm for MLPNN (Lancashire *et al*., 2009). BP works by feeding training samples one-by-one to the network and then finds the squared difference between the real output (desired) and the network's output (estimated) as follows:

$$E=\frac{1}{2}\sum_j\left(y_{dj}-y_j\right)^2 \tag{8}$$

where y_{dj} and y_j are the desired and estimated output values of output neuron j.

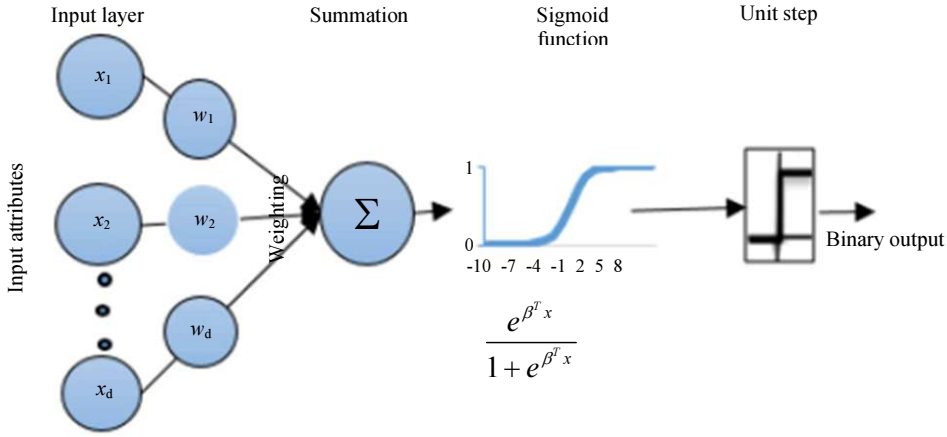

Fig. 1: Schematic diagram of the LR model and logistic function

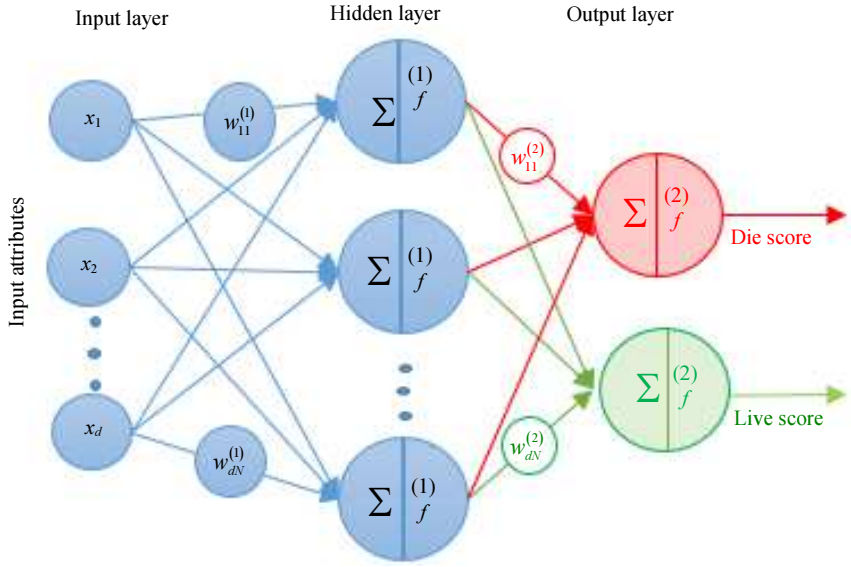

Fig. 2: Graphical representation of MLPNN with one hidden layer for hepatitis dataset

BP corrects the network weights iteratively to reduce the summation of all squared errors, *E*. A network with high generalization characteristics will be able to reduce any future errors. However, such an optimal structure is still a challenging task. On one side, the structure has to be relatively small to increase the generalization. On the other side, the network has to classify the training samples efficiently. Some training algorithms start with a large network structure and then apply a pruning method to remove redundant and weak connections (Amato *et al*., 2013).

Radial Basis Function Neural Network (RBFNN)

RBFNN consists of only three fully interconnected layers: Input, hidden and output layer as shown in Fig. 3 (Haykin, 2009). The input layer delivers the input data to the hidden neurons. The number of neurons in the hidden layer (N_H) is determined during the training process. Every hidden neuron represents a basis function with equal dimensions to the input observations. They represent particular points in the input space and their responses depend on the distances between them and the input observations. The appropriate activation functions for RBFNN have to be strictly positive and radially symmetric with their corresponding unique maxima at their centers (Prez-Godoy *et al*., 2014). That is the closer the observation is to a given hidden neuron's center, the stronger is its response. Gaussian activation functions are common in RBFNN (Haykin, 2009). They are characterized by their mean vectors, m_i and spreads, σ_i, where $i = 1, 2, \ldots, N_H$. The activation function, g_i, of the i^{th} hidden neuron for an observation x_j is given by:

$$g_i\left(x_j\right) = e^{\left(\frac{-\|x_j - m_i\|^2}{2\sigma_i^2}\right)} \quad (9)$$

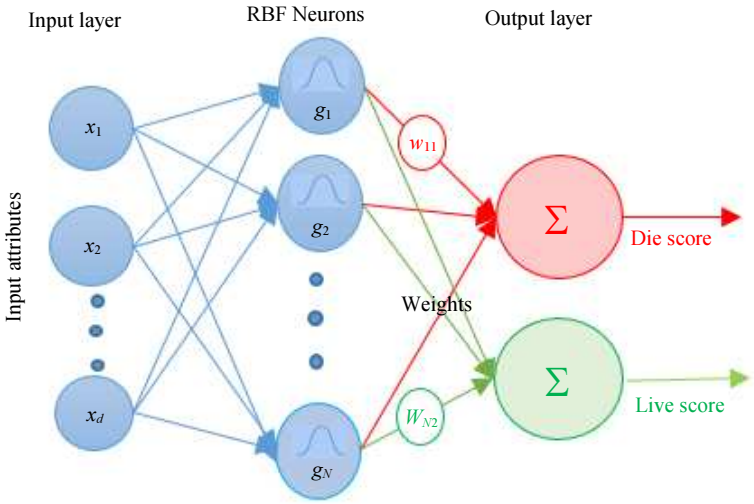

Fig. 3: RBFNN with Gaussian basis function for hepatitis dataset

The hidden layer neurons are fully connected to the output layer neurons through weights w_{ik}. The number of output neurons equals the number of classes. The output of the k^{th} neuron of the output layer for an observation x_j can be calculated as:

$$y_k\left(x_j\right)=\sum_{i=0}^{N_H} w_{ik} g_i\left(x_j\right) \tag{10}$$

where $g_0(x_j) = 1$.

Equation (10) shows that the RBFNN can be considered as an approximation of the output y_k by the weighted sum of non-orthogonal Gaussian basis functions (Venkatesan and Anitha, 2006). The learning algorithm has to predefine the centers as well as the spreads of the Gaussians in the hidden layer. Then, it uses the training subset to adjust the weights of the output layer to minimize the classification error.

Performance Measures

In this work, to assess the modeling performance, some common measures have been used namely: Overall accuracy, sensitivity, specificity, F1-score, and AUCROC. These measures are derived from the confusion matrix, which records the correctly and incorrectly classified observations for each class. Table 2 shows the confusion matrix for a binary classification problem. True Positive (TP) is the number of positive observations predicted correctly, False Positive (FP) is the number of negative observations classified as positives incorrectly, True Negative (TN) is the number of negative observations predicted correctly and finally, False Negative (FN) is the number of positive observations classified as negatives incorrectly.

Accuracy quantifies the fraction of the correctly predicted observations to all observations; sensitivity quantifies the fraction of accurately predicted positive observations; specificity quantifies the correctly predicted negative observations (Hossin and Sulaiman, 2015). These metrics are computed as follows:

$$Accuracy=\frac{TP+TN}{N} \tag{11}$$

$$Sensitivity=\frac{TP}{TP+FN} \tag{12}$$

$$Specificity=\frac{TN}{TN+FP} \tag{13}$$

The dataset has imbalanced class distribution; it has more negative observations than the positives. Therefore, the overall accuracy is greatly affected by this large number of true negatives, which gives a misleading percentage. Typically, there are two different approaches to deal with such unbalanced data. The first approach works to achieve the balancing in the training data using different techniques whether by oversampling the minority records or undersampling the majority ones (Pourhabib, 2019). The second approach relies on the use of specific performance measures other than the general accuracy.

Table 2: The confusion matrix for binary classification

Value predicted	Actual value positives	Negatives
Positives	True Positive **TP**	False Positive **FP**
Negatives	False Negative **TN**	True Negative **FN**

In this study, we applied the F1-score and AUCROC performance metrics that are not affected by such uneven distribution. F1-score equals the weighted harmonic mean between precision and recall as shown in Equation 15. Precision represents the fraction of true positives among the predicted positive ones, while recall is the same as sensitivity. The greater the F1-score the better the performance of the model (Hossin and Sulaiman, 2015; Marina *et al*., 2006):

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{15}$$

The Receiver Operating Characteristic (ROC) curve is a graphical general performance metric. It plots the sensitivity against one minus the specificity for different values of the threshold (Marina *et al*., 2006). This curve evaluates the model's ability to identify the positive observations from the negative ones. The area under this curve is called AUCROC, which represents the probability that the model will rank a randomly chosen positive observation higher than a randomly chosen negative one. AUCROC is commonly used in many applications to measure the performance of any model across all possible thresholds. Higher AUC indicates better model prediction power. However, it is not recommended for extremely imbalanced data which contains very rare events. In such situations, the area under the precision-recall curve is more appropriate (Sofaer *et al*., 2019). As stated by Equation (14), precision is a measure of result relevancy, while recall is the same as sensitivity. The precision-recall curve illustrates the compromise between precision and recall for different threshold values.

Experimental Results

IBM® SPSS modeler data mining workbench has been used to implement and validate the LR and ANN predictive models (IBM, 2016). Figure 4 shows the sequential processes for the analysis and diagnosis of the hepatitis dataset. The stream starts with the *InputData* node to read the data file. Then the subsequent nodes are used to define the attribute types, audit the data, check the distribution, explore the basic statistics and inspect the quality of all observations. Hence, the stream imputes the missing values, partitions the observations into training and testing subsets, develops all predictive models with the training subset, validates the trained models and finally scores all observations.

Data Exploration and Preprocessing

The data has been explored to evaluate the percentage of missing data among the attributes as well as the observations. It has been found that the "PROTIME" attribute has 43.23% percentage of missing values. It was totally deleted from any further processing. In addition, all observations with more than two missing values were excluded. The remaining dataset contained 141 observations with only 18 predictive attributes associated with their corresponding output classes. The *type* node specifies the properties of all attributes and defines the metadata whether it is for input attribute or output class. The *DataAudit* node is used to examine the basic statistics of each attribute, checks the quality of the whole dataset and finally generate the *Imputation* node to fill the missing values with a separated C&RT algorithm for each attribute (IBM, 2016; Buuren, 2018). The multicollinearity among the input attributes has been calculated using Variance Inflation Factors (VIF) (Midi *et al*., 2013). Multicollinearity can lead to biased estimates and inflated standard errors. VIF measures how much the variance of a coefficient increases due to collinearity. A VIF value of greater than 5 is generally considered as evidence of multicollinearity (Midi *et al*., 2013). In this study, collinearity diagnostics showed that all attributes have VIFs below 5. Therefore, all attributes have been used in the LR analysis. The stream continues by partitioning the input data by the *Partition* node into 70% for training and 30% for testing subsets.

Models Construction

LR Models

Three LR models have been constructed and studied: traditional, forward stepwise and backward stepwise. The singularity tolerance, the maximum iterations, the Log-likelihood convergence and the parameter of convergence are the iteration process stopping criteria and their values were set to 1.0E05, 30, 1.0E-3 and 1.0E-4, respectively. The entry and removal probability values of the forward and backward stepping methods were set to 0.01 and 0.1, respectively. These values were selected to avoid including less important attributes or removing more important ones. The iterative process continues until one of the above stopping criteria is reached. In the stream shown in Fig. 4, *LR_Enter* node was used to train the traditional LR model with the block entry of all attributes directly to predict the class outcomes. On the other hand, the *LR_Forward and LR_Backward* nodes represented forward and backward stepping classifiers. The classifiers' outputs for the training and testing data were compared with the target classes for identifying the confusion matrix as illustrated in Table 3.

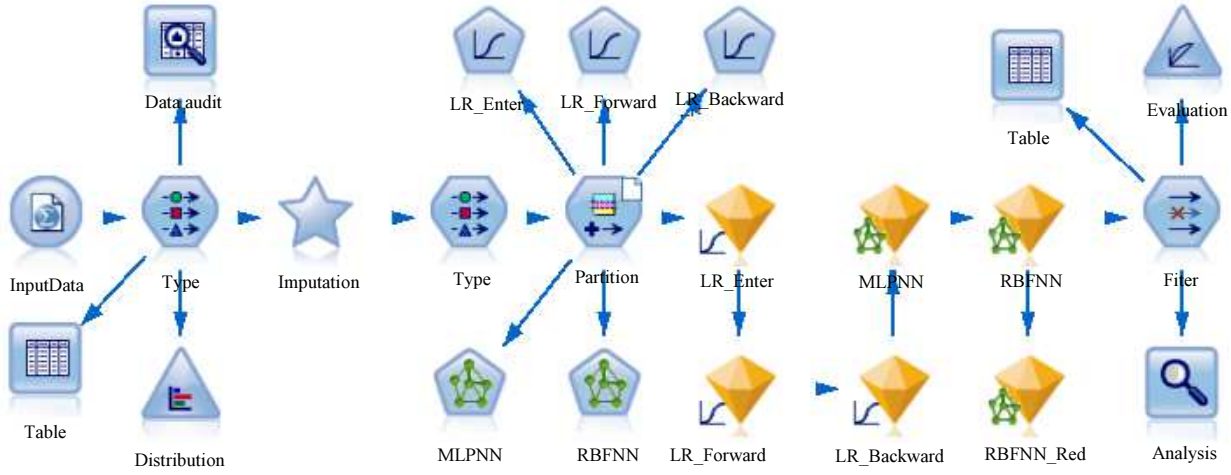

Fig. 4: Stream of analysis and prediction models for the diagnosis of hepatitis dataset

Table 3: The confusion matrix for the three LR models: Enter, Forward and |Backward

	Training				Testing			
Model	TP	TN	FP	FN	TP	TN	FP	FN
LR_Enter	16	77	1	3	16	77	1	3
LR_Forward	14	76	2	5	14	76	2	5
LR_Backward	16	76	2	3	16	76	2	3

The MLPNN model was built with only one hidden layer. The input layer consists of 18 neurons to accept the input attributes while the output layer has two neurons for the class outputs. To end up with the best structure, the hidden layer was tested with a different number of neurons, then the performance was evaluated for each case. The hidden layer was tested for the number of neurons from 1 to 12. The maximum number of learning cycles was set to 1000 and the learning process was unleashed to achieve 100% training accuracy. Ten percent of the training subset has been reserved for validation to reduce the over-fitting and hence to increase the generalization. The network weights were not adjusted with this validation data set, but this set of data has been used to verify that any increase in classification accuracy over the training data set, on which the model was built, actually increases the accuracy of data that has not been used in the training process (i.e., our validation data set). However, if the model gets an increase in the accuracy over the training data set, while the value of the accuracy over the validation data set continues as is or decreased, then the network will be overfitted with low generalization capability and the training process must be stopped. Table 4 shows the resulting confusion matrix of MLPNN with 1, 3, 5, 7 and 9 neurons in the hidden layer.

RBFNN Model

In the case of the RBFNN model, the model was tested for a different number of RBF hidden neurons in each run. The network was built with a number of neurons from 6 to 17. Table 5 shows the confusion matrix of the results with 7, 9, 11 and 13 neurons in the hidden layer. The data set was divided in the same way as in the case of MLPNN with 10% of the training samples has been reserved for validation of the network.

Analysis and Discussion

The confusion matrices of different models have been used to compute the statistical metrics, which were discussed in section 4. The performance measures were calculated individually for each run and their values for training and testing subsets. Based on the results of the classification of testing data, MLPNN achieved the best F1-score with 7 neurons, while the RBFNN the best F1-score with 11 neurons. Tables 6 and 7 as well as Fig. 5a and 5b illustrate the values and graphical representation of the performance metrics for training and testing data, respectively.

The overall accuracy is an indicator of the model's effectiveness. It is calculated in terms of the probability of the true positives and the true negatives of the predicted classes. Table 6 and Fig. 5a shows that the LR_Enter classifier has the best training accuracy, which is reached 95.88%. However, the RBFNN and LR_Forward classifiers have the worst training accuracy of 92.78%. Similarly, Table 7 and Fig. 5b illustrates that the RBFNN classifier achieves the best testing accuracy of 86.36% while the LR_Enter and LR_Backward have the worst value of 75.0%. Hence, the LR models are suffering much more from the overfitting problem.

Sensitivity and specificity approximate the ability of the model to correctly classify the data observations to "Die" and "Live" classes, respectively. Referring to Table 6 and Fig. 5a, it can be observed that the sensitivity of both LR_Enter and LR_Backward classifiers of the training data are better than the others. They have the same sensitivities which reach 84.21%. However, for the testing subset, the LR_Enter and RBFNN models perform better than the others. The specificity of LR_Enter, MLPNN and RBFNN models are found to be equal and reach 98.72% for training data. However, for the testing observations, both MLPNN and RBFNN achieve the same specificity value of 88.57%.

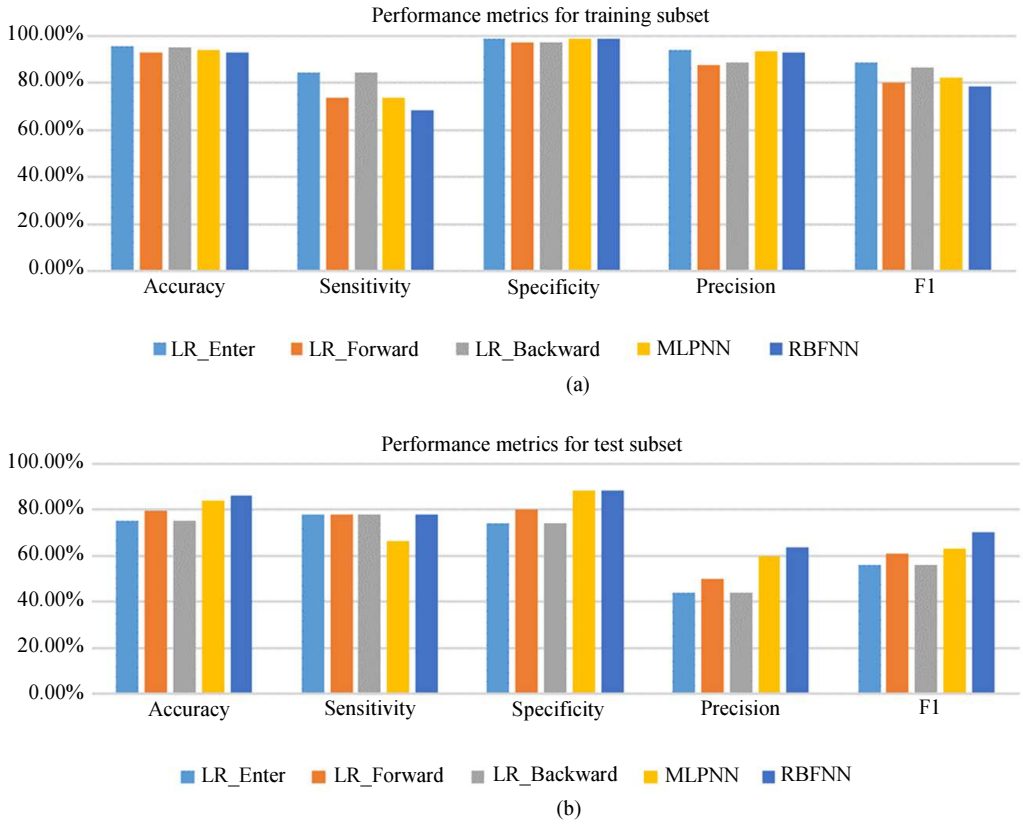

Fig. 5: Performance metrics of classification models for; (a) training data; (b) test data; MLPNN and RBFNN have 7 and 11 hidden layer's neurons, respectively

Table 4: The confusion matrix for the MLPNN for different number of neurons

	Training				Testing			
Neurons	TP	TN	FP	FN	TP	TN	FP	FN
1	13	77	1	6	7	29	6	2
3	15	77	1	4	6	29	6	3
5	13	76	2	6	6	29	6	4
7	14	77	1	5	6	31	4	3
9	15	77	1	4	6	29	6	3

Table 5: The confusion matrix for the RBFNN for different number of neurons

	Training				Testing			
Clusters	TP	TN	FP	FN	TP	TN	FP	FN
7	11	77	1	8	5	30	5	4
9	12	77	1	7	6	30	5	3
11	13	77	1	6	7	31	4	2
13	13	77	1	6	6	31	4	3

Table 6: The values of the performance measures for the training subset

Model	Accuracy	Sensitivity	Specificity	Precision	F1-score
LR_Enter	95.88	84.21	98.72	94.12	88.89
LR_Forward	92.78	73.68	97.44	87.50	80.00
LR_Backward	94.85	84.21	97.44	88.89	86.49
MLPNN	93.81	73.68	98.72	93.33	82.35
RBFNN	92.78	68.42	98.72	92.86	78.79

Table 7: The values of the performance measures for the testing subset

Model	Accuracy	Sensitivity	Specificity	Precision	F1-score
LR_Enter	75.00	77.78	74.29	43.75	56.00
LR_Forward	79.55	77.78	80.00	50.00	60.87
LR_Backward	75.00	77.78	74.29	43.75	56.00
MLPNN	84.09	66.67	88.57	60.00	63.16
RBFNN	86.36	77.78	88.57	63.64	70.00

Precision is the ratio between the correctly predicted positive observations of the total predicted positive ones. This metric answers this question: How many patients already died from those who are predicted as "Die"? High precision is related to the low false-positive rate. In this study, the LR_Enter classifier has the best precision for the training subset with 94.12% which is pretty good. In the testing subset, the RBFNN achieved the best with 63.64%.

The F1-score is a combined metric that balances precision and sensitivity. It is preferred to use this metric when the dataset has biased distribution as in the current hepatitis dataset. In comparison, the classifier with a high F1-score is considered as superior to the others. Moreover, the classifier with a low F1-score value should be ignored. In the training dataset, the F1-score of the LR_Enter classifier achieved the highest value of 88.89% but it attained the worst value for the testing data. Among all classifiers, RBFNN achieved the best F1-score with 70.00% for testing data.

The ROC curves show the relation between the sensitivity and specificity of predictive algorithms, which give summaries of performances over the whole range of values. The areas under these ROC curves (AUCROCs) are independent measures, which weight the sensitivity and specificity in proportion to their occurrences. Figure 6 shows the ROC plots of the models under consideration. The higher lines indicate better models, especially on the left side of the chart.

The AUCROCs of all models for the training and testing subsets are presented in Fig. 7. These plots illustrate that the LR-Enter that uses all attributes achieved the best performance on the prediction of the training set. Furthermore, it has comparable performance to both MLPNN and RFNNN when classifying the testing observations. Both MLPNN and RBFNN achieved different performances according to AUCROCs. MLPNN achieved 0.883 while RBFNN achieved 0.873.

LR Attribute Selection

The LR models have been proved to be very useful for understanding the effect of every input attribute on the output response. The influence of each attribute can easily be captured from Equation 4. The magnitude of a coefficient β, assigned to a certain attribute on the *Logit* function, represents the importance degree of this attribute. On the other hand, the stepwise LR modeling is an iterative process that involves the inclusion or removal of attributes to or from the model during the learning process (Maxwell and Obinna, 2018). At every iteration, a specified fraction of attributes is included or eliminated based on the ranking of their weights, until the required number of attributes are left or the prediction errors do not decrease. This study employed forward and backward stepwise LR models to select the relevant attributes and remove the irrelevant ones. Figure 8 shows the relative importance of different attributes performed by different LR algorithms. Forward LR employed four attributes, which are added iteratively as follows ASCITES, BILIRUBIN, SPIDERS, and SEX. While Backward LR used eight ones which are ASCITES, BILIRUBIN, SPIDERS, ALBUMIN, SGOT, SEX, STEROID, and LIVER_BIG. From Fig. 8, it is shown that three attributes have the greatest importance among the three models, namely: ASCITES, BILIRUBIN, and SPIDERS.

MLPNN and RBFNN classifiers have been re-applied again two times; first by using the attributes selected by the forward LR approach and second by using the attributes chosen by the backward one. The resulting AUCROCs are plotted for comparison in Fig. 9. It is shown that the attributes selected by the forward approach improved the performance of MLPNN on the training observations from 0.906 to 0.928 and worsen the performance of the testing data from 0.883 to 0.854. On the other hand, the RBFNN classifier has improved significantly in both training and testing datasets. However, the attributes selected by the Backward approach improved the performances of both MLPNN and RBFNN on training and testing subsets.

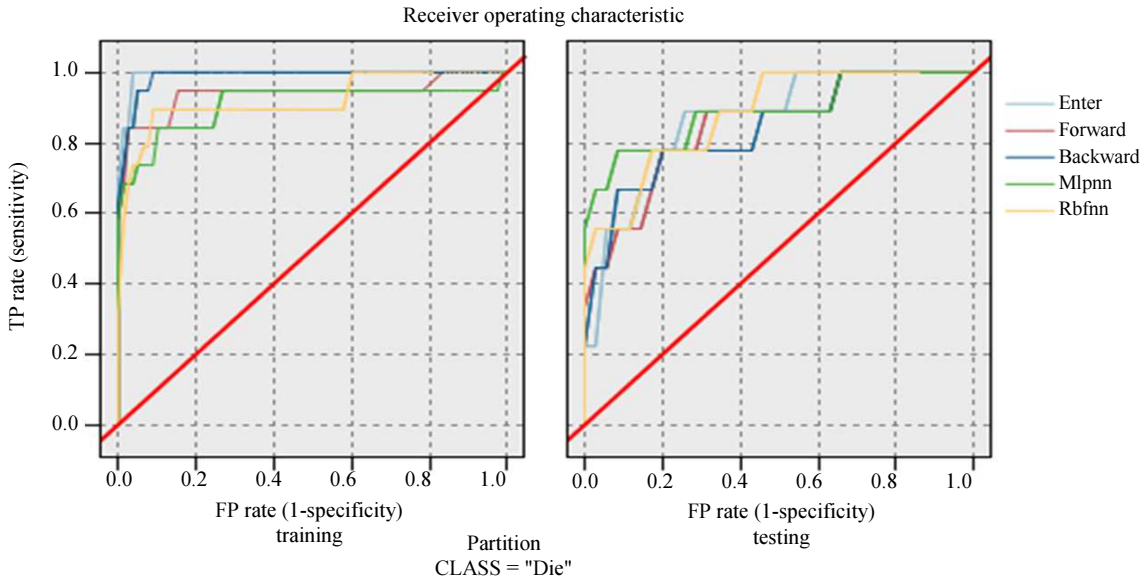

Fig. 6: ROC curves of three LR and two ANN predictive models for training and test subsets

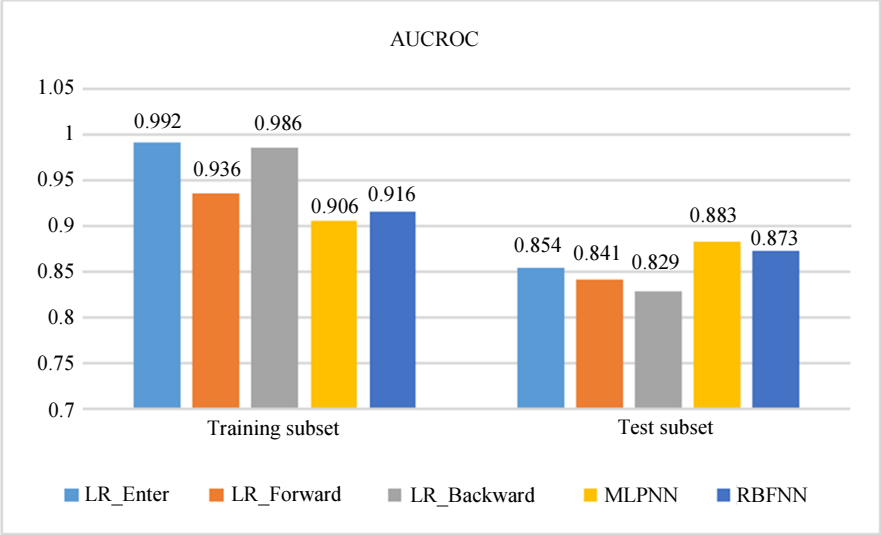

Fig. 7: The area under the receiver operating characteristic curves of all predictive models for training and test subsets

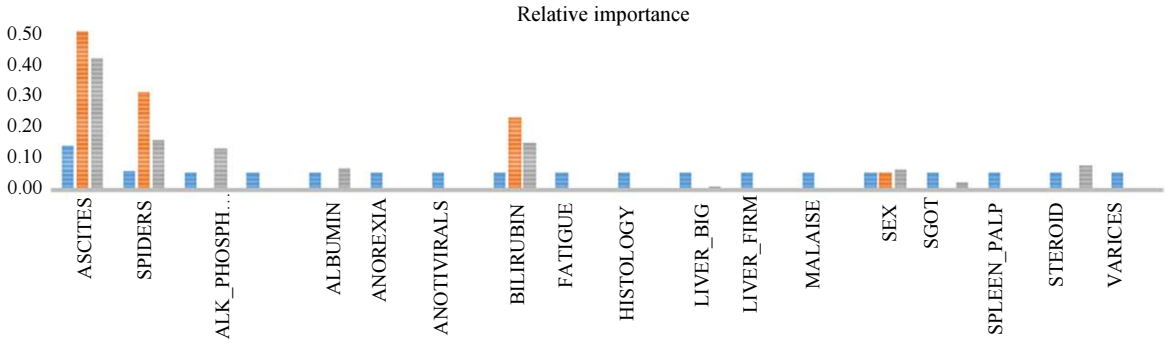

Fig. 8: The relative importance of all attributes for the enter, forward and backward LR models

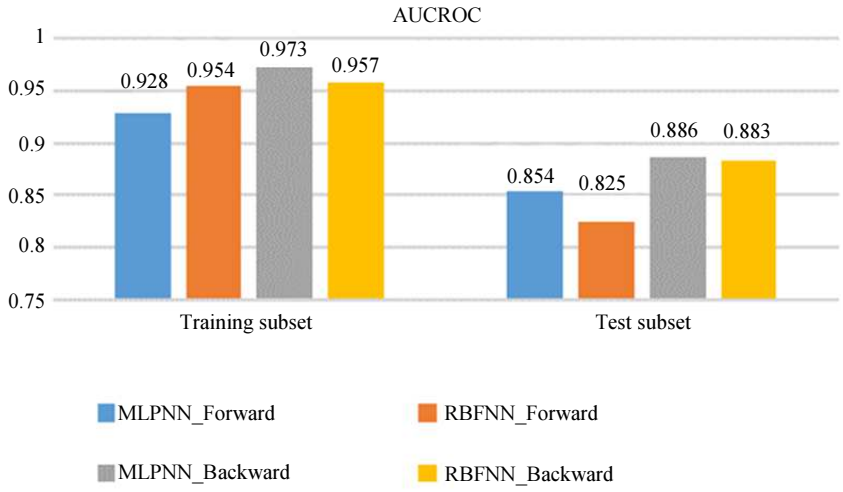

Fig. 9: AUCROC values of MLPNN and RBFNN with selected attributes by forward and backward LRs

The comparisons between Fig. 7 and 9 demonstrate that the stepwise LR models provide efficient methods for examining potential underlying relationships between the input attributes and the output response. The feature selection with Backward LR combined with classification with MLPNN outperforms other algorithms with AUCROC 0.973 for training subset and 0.886 for test one respectively. These results confirm the ability of LR models in selecting attributes. It is clear that the approach of the LR modeling does not assume any distribution assumptions neither on the explanatory attributes nor on the dependent variable. It is a direct probability model without any intermediate tool such as the Bayes rule for converting results into probabilities. Consequently, LR coefficients can be used directly to rank and select attributes for artificial intelligence applications.

The resulting accuracies are comparable to those published in the most recent paper in (Nilashi *et al*., 2019) where the maximum AUCROC the author got is 0.9456. However, here in our study, we did not discard records with missing values but, on the contrary, they have been imputed using a separate Classification and Regression Tree (C&RT) for each attribute. Nevertheless, the use of feature reduction and structure tuning of neural networks were effective techniques in improving the classification accuracies. There is an opportunity for future work to develop a more effective neural network based on the techniques presented in the literature especially using bagging, boosting and other augmentation techniques.

Conclusion

Both ANN and LR are powerful tools to help physicians examining medical data, making decisions and diagnose correctly. They make the diagnosis more reliable and increase patient satisfaction. The purpose of this study is to combine both methods for the analysis and classification of the hepatitis data. The importance of this study comes from the increase in the yearly mortality rate due to hepatitis, which has become a major concern around the world. This paper presents LR models and MLPNN as well as RBFNN. Moreover, it introduces and analyzes the applications of these algorithms in predicting the status of hepatitis patients as “Die” or “Live” with the use of clinical and laboratory test results. The dataset has been imputed and preprocessed before the modeling phase. Besides the traditional LR, two stepwise LR classification models have been investigated: the forward and backward ones. These stepwise methods selected the predictive attributes using an iterative procedure during the learning process. The selected attributes have been used to build the MLPNN and RBFNN models. They succeeded to produce more efficient and generalized models. Particularly, the attributes selected by the backward LR algorithm have proved to be effective in improving the performance of both neural network models. The dataset has been divided into training, validation and test subsets where the role of the validation subset (10% of the training data) was to reduce the over-fitting and increase the generalization capacity of the resulting neural models. The experimental results demonstrated that the MLPNN with the attributes selected by the backward LR model resulted in the best performance with AUCROCs 0.973 for training and 0.886 for test subsets. These results confirmed that better performance could be obtained based on hybrid algorithms that take advantage of the good characteristics of different prediction models.

Acknowledgment

This project was supported by the Deanship of Scientific Research at Prince Sattam Bin Abdulaziz University under the research project 2017/01/8262.

Author's Contributions

All authors contributed equally to this manuscript.

Ethics

No ethical issues that may arise after the publication of this manuscript.

References

Afif, M.H., A.R. Hedar, T.H.A. Hamid and Y.B. Mahdy, 2013. SS-SVM (3SVM): A new classification method for hepatitis disease diagnosis. Int. J. Adv. Comput. Sci. Applic., 4: 53-58.

Amato, F., A. Lṕez, E.M. Peǎ-Méndez, P. Vaňhara and A. Hampl *et al*., 2013. Artificial neural networks in medical diagnosis. J. Applied Biomed., 11: 45-58.

Ansari, S., I. Shafi, A. Ansari, J. Ahmad and S.I. Shah, 2011. Diagnosis of liver disease induced by hepatitis virus using artificial neural networks. Proceedings of the 14th International Multitopic Conference, Dec. 22-24, IEEE Xplore Press, Karachi, Pakistan, pp: 8-12. DOI: 10.1109/INMIC.2011.6151515

Bhargav, K.S., D.S.S.B. Thota, T.D. Kumari and B. Vikas, 2018. Application of machine learning classification algorithms on hepatitis dataset. Int. J. Applied Eng. Res., 13: 12732-12737.

Blake, C.L. and C.J. Merz, 1996. UCI machine learning repository.

Borah, A. and B. Nath, 2018. Identifying risk factors for adverse diseases using dynamic rare association rule mining. J. Exp. Syst. Applied, 113: 233-263.

Buuren, S.V., 2018. Flexible Imputation of Missing Data. 2nd Edn., Chapman and Hall/CRC, ISBN-10: 1138588318, pp: 415.

Çetin, O., F. Temurtaş and Ş. Gülgönül, 2015. An application of multilayer neural network on hepatitis disease diagnosis using approximations of sigmoid activation function. Dicle Med. J., 42: 150-157.

Chen, H.L., D.Y. Liu, B. Yang, J. Liu and G. Wang, 2011. A new hybrid method based on local fisher discriminant analysis and support vector machines for hepatitis disease diagnosis. Exp. Syst. Applic., 38: 11796-11803.

Cohen, J., 1999. The scientific challenge of hepatitis C. Science, 285: 26-30.

Daniel, C., 2018. http://hepatitis.about.com/od/overview/a/numbers.htm

Gullo, F., 2015. From patterns in data to knowledge discovery: What data mining can do. Phys. Proc., 62: 18-22.

Haykin, S., 2009. Neural Networks and Learning Machines. 3rd Ed., NJ: Prentice-Hall, ISBN-10: 0131293761, pp: 934.

Hosmer, D.W. and S. Lemeshow, 2000. Applied Logistic Regression. 2nd Edn., John Wiley and Sons, ISBN-10: 0471356328, pp: 373.

Hossin, M. and M.N. Sulaiman, 2015. A review on evaluation metrics for data classification evaluations. Int. J. Data Mining Knowl. Manage. Process, 5: 1-11.

IBM, 2016. IBM SPSS modeler 18.0, algorithms guide.

Ing, E.B., N.R. Miller, A. Nguyen, W. Su and L.L.C.D. Bursztyn *et al*., 2019. Neural network and logistic regression diagnostic prediction models for giant cell arteritis: Development and validation. Clin. Ophthalmol., 13: 421-421.

James, S. and N. Foley, 2018. https://askanaturopath.com/faqs/liver-function-test/p/467

Jiang, F., Y. Jiang, H. Zhi, Y. Dong and H. Li *et al*., 2017. Artificial intelligence in healthcare: Past, present and future. Stroke Vascular Neurol., 2: 230-243.

Kaya, Y. and M. Uyar, 2013. A hybrid decision support system based on rough set and extreme learning machine for diagnosis of hepatitis disease. J. Applied Soft Comput., 13: 3429-3438.

Lancashire, L.J., C. Lemetre and G.R. Ball, 2009. An introduction to artificial neural networks in bioinformatics-application to complex microarray and mass spectrometry datasets in cancer studies. Briefings Bioinform., 10: 315-329.

Marina, S., J. Nathalie and S. Stan, 2006. Beyond accuracy, f-score and ROC: A family of discriminant measures for performance evaluation. Proceedings of the 19th Australian joint conference on Artificial Intelligence: Advances in Artificial Intelligence Hobart, Dec. 4-8, Australia.

Maxwell, I.A. and N. Obinna, 2018. A comparative study of some variable selection techniques in logistic regression. Eur. J. Math. Comput. Sci., 5: 1-20.

MayoClinic, 2018. https://www.mayoclinic.org/diseases-conditions/hepatitis-c/symptoms-causes/syc-20354278

Midi, H., S.K. Sarkar and S. Rana, 2013. Collinearity diagnostics of binary logistic regression model. J. Int. Math., 13: 253-267.

Mitra, M. and R.K. Samanta, 2015. Hepatitis disease diagnosis using multiple imputation and neural network with rough set feature reduction. Proceedings of the 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications, (CTA' 15), pp: 285-293. DOI: 10.1007/978-3-319-11933-5_31

Nilashi, M., H. Ahmadi, L. Shahmoradi, O. Ibrahim and E. Akbari, 2019. A predictive method for hepatitis disease diagnosis using ensembles of neuro-fuzzy technique. J. Infec. Public Health, 12: 13-20. DOI: 10.1016/j.jiph.2018.09.009

Patel, V.L., E.H. Shortliffe, M. Stefanelli, P. Szolovits and M.R. Berthold *et al*., 2009. The coming of age of artificial intelligence in medicine. Artificial Intell. Med., 46: 5-17.

Pourhabib, A., 2019. Empirical similarity for absent data generation in imbalanced classification. Proceedings of the Future of Information and Communication Conference, Springer, Cham, pp: 1010-1030. DOI: 10.1007/978-3-030-12388-8_70

Prez-Godoy, M.D., A.J. Rivera, C.J. Carmona and M.J.D. Jesus, 2014. Training algorithms for radial basis function networks to tackle learning processes with imbalanced data-sets. Applied Soft Comput., 25: 26-39.

Sartakhti, J.S., M.H. Zangooei and K. Mozafari, 2012. Hepatitis disease diagnosis using a novel hybrid method based on Support Vector Machine and Simulated Annealing (SVM-SA). Comput. Meth. Programs Biomed., 108: 570-579.

Sofaer, H.R., J.A. Hoeting and C.S. Jarnevich, 2019. The area under the precision-recall curve as a performance metric for rare binary events. Meth. Ecol. Evolut., 10: 565-577.

Sperandei, S., 2014. Understanding logistic regression analysis. Biochem. Med., 24: 12-18.

Taylor, J.M., 2003. Replication of human hepatitis delta virus: Recent developments. Trends Microbiol., 11: 185-90.

Venkatesan, P. and S. Anitha, 2006. Application of a radial basis function neural network for diagnosis of diabetes mellitus. Curr. Sci., 91: 1195-1199.

WHO, 2017. Global hepatitis report. World Health Organization.