

Rule extraction for fatty liver detection using neural networks

Mojtaba Shahabi¹ · Hamid Hassanpour¹ · Hoda Mashayekhi¹

Received: 19 November 2016 / Accepted: 19 June 2017 / Published online: 28 June 2017
© The Natural Computing Applications Forum 2017

Abstract Non-alcoholic fatty liver disease (NAFLD) is one of the most common diseases in the world. Recently the FibroScan device is used as a noninvasive, yet costly method to measure the liver's elasticity as a NAFLD indicator. Other than the cost, the diagnosis is not widely accessible to all patients. On the other hand, early detection of the disease can prevent later risks. In this study, we aim to use learning methods to infer the NAFLD severity level, only based on clinical tests. A dataset was constructed from clinical and ultrasonography data of 726 patients who were diagnosed with different NAFLD severity levels. Artificial neural networks (ANN) were used to model the relationship between NAFLD and the clinical tests. Next, a method was used to analyze the ANN and extract compact and human understandable rules. The derived rules can detect the fatty liver disease with an accuracy above 80%.

Keywords Disease detection · Non-alcoholic fatty liver disease (NAFLD) · Artificial neural networks (ANNs) · Rule extraction

1 Introduction

Liver is the biggest internal part of the human body that performs a big role in human life [1, 2]. Non-alcoholic fatty liver disease (NAFLD) is among the most common diseases related to liver and has several severity levels; it starts with a *simple steatosis* and develops to *cirrhosis*

[3, 4]. Appropriate and early detection of the disease is very important as it can prevent serious later risks [5, 6].

Liver biopsy was introduced as a first method for fatty liver disease detection [6, 7]. But recently, optical methods (or image analyses [2]) have become more popular because of being less invasive and risky for the patient. The FibroScan is the least risky method for measuring the elasticity of liver [8, 9]. The elasticity measure shows the level of the NAFLD severity [1, 4, 10, 11]. Table 1 presents different levels of the NAFLD.

The accuracy of the FibroScan is higher than other NAFLD detection methods. However, it is an expensive and not widely accessible method. Therefore, researchers are seeking to create a system for disease severity detection which requires simpler and less paid tests [12–16]. The main problem of these systems (such as the Forns score system in 2002 [14] and Angulo method in 2007 [13]) is that only certain levels of disease are detectable with accuracy values much lower than FibroScan. In this research, we introduce a new simple and less paid method for NAFLD detection based on artificial neural networks (ANN). In the new method, severity levels of fatty liver disease are induced from clinical features obtained from complete blood count (CBC) and ultrasonography test. The CBC is a common test, and the result contains parameters that explain the conditions of organs through measuring different enzymes in blood (like sugar, cholesterol, and urea).

The dataset used in this research was obtained from patients who visited Sayad Shirazi hospital in the Golestan province, Iran in 2011. The dataset contains results of the blood test, ultrasonography, and FibroScan for the patients. All the participants gave oral informed consent to use these data for scientific purposes, and the study was approved by our Ethical Committee.

✉ Hamid Hassanpour
h.hassanpour@shahroodut.ac.ir

¹ Faculty of Computer Engineering and IT, Shahrood University of Technology, Shahrood, Iran

Table 1 The results of the FibroScan for detecting five levels of NAFLD

Levels	Description
F0	Absence of fibrosis (no fibrosis)
F1	Perisinusoidal or portal involvement (portal fibrosis without septa)
F2	Perisinusoidal and portal/periportal involvement (portal fibrosis and few septa)
F3	Septal or bridging fibrosis (numerous septa without cirrhosis)
F4	Cirrhosis

In this research, after preprocessing, parameters of the blood test and ultrasonography were applied as inputs to a neural network. The known disease level from FibroScan serves as the class label (the network output) in the training phase. The relationship between input and output is attained by training the neural network. Usually a neural network presents a good result in classification of training data, but the network prediction is less interpretable by humans [17–19]. In the literature, different methods have been developed for rule extraction from neural networks [20, 21]. The outcome of such methods is a collection of rules, which can be used as alternative to the original neural network in operation [17]. The rules have the benefit of being more comprehensible by the user. In this paper, a Four-Step Rule Extraction (FSRE) method is introduced to derive the rule set from the neural network. This method has less complexity with an equal or higher accuracy when compared to several other methods. The results of applying the FSRE method in NAFLD severity detection present the efficiency and usefulness of this rule extraction method in comparison with previous work.

This paper is organized as follows. Section 2 reviews previous works in the liver fibrosis diagnosis and rule extraction context. Our rule extraction method from neural network is presented in Sect. 3. Section 4 includes the evaluation and simulation results. Finally, Sect. 5 presents the conclusion.

2 Related works

In this section, we first review the studies related to rule extraction for NAFLD detection, and then discuss the studies on rule extraction using neural networks. The methods whereby the researchers could propose to evaluate a fibrosis diagnosis score using blood test parameters are described.

Forns et al. [14] examined the relationship between laboratory test values and liver fibrosis on 351 patients. They concluded that the parameters age, gamma-glutamyl transpeptidase (GGT), total-cholesterol (T-CHOL), and platelets (PLT) are independent predictors of fibrosis. They created the below formula based on these parameters:

$$\text{Score} = 7.811 - 3.131 \times \ln(\text{PLT}) + 0.781 \times \ln(\text{GGT}) + 3.467 \times \ln(\text{Age}) - 0.014 \times (\text{TCHOL}) \quad (1)$$

This method can attain a cutoff value for detecting only the significant levels of the disease (F2–F4). Wai et al. [16] reported that the ratio between the multiple of the upper limit associated with a normal glutamic oxaloacetic transaminase (GOT) and PLT is useful for assessing liver fibrosis. The formula obtained in their research is shown below.

$$\text{Score} = \frac{\text{GOT}}{\text{upper limited of normal for } \frac{\text{GOT}}{\text{PLT}}} \times 100 \quad (2)$$

The Wai et al. method, similar to the Forns et al. method, can only detect significant levels of fibrosis (F2–F4 levels). Lok et al. [15] used blow formula to predict liver cirrhosis rather than graded assessment of fibrosis:

$$\begin{aligned} &\text{Log odds}(\text{predicted cirrhosis}) \\ &= -5.56 - 0.0089 \times \text{PLT} + 1.26 \times \frac{\text{GOT}}{\text{GPT}} + 5.27 \\ &\quad \times \text{PT_INR}, \end{aligned} \quad (3)$$

in which the PT_INR is the international normalized ratio of prothrombin time.

Rule extraction using artificial neural networks is an appropriate method to extract suitable rules for disease detection or classification. For rule extraction, first a neural network is selected according to the problem in hand. Second, the training process yields the relation between input and output features. Finally, this relation is expressed in the form of a set of limited, simple, and efficient rules. Different algorithms exist for rule extraction from neural networks which can be classified in different aspects.

The type of attributes in the database is an important issue for a rule extraction method. Some methods are suitable for datasets only containing *continuous* attributes (e.g., temperature, height, weight) [22]. Some other methods only support dataset containing *discrete* attributes (e.g., sex, education level, and religion) [23, 24]. There are some methods that can deal with attributes containing *both* discrete and continuous types [21, 25].

The rule extraction methods can also be classified based on the form of the neural network's output. Some methods

express results in the form of an M-of-N rule set. In this case, N conditions (without any priority or ordering) are considered for an output class. An input sample is a member of this class if it satisfies at least M conditions. For example the M-of-N rule for two-bit XOR problem may be expressed as follows: “if (exactly 1 of 2 inputs is true) then Odd parity, else Even parity” [23]. However, majority of available methods are based on IF–THEN rules. In this case, each rule declares a condition of belonging to a special output class. For example the IF–THEN rule for two bits XOR problem is: “if ($X1 = 1$ and $X2 = 0$) or ($X1 = 0$ and $X2 = 1$) then Odd parity, else Even parity” [26, 27]. There are some other methods that employ fuzzy conditions instead of crisp ones. In this case fuzzification and defuzzification operations will be required [21, 28, 29].

Some of the rule extraction methods use the self-organizing map (SOM) neural networks [30, 31]. Generally, the outcome of such methods is a very simple rule set that can be applied in simple datasets. As this type of neural networks is only used for data representation and deriving relations between neighborhoods, they are not generally useful for classification problems. Malone et al. [31] applied a SOM neural network on the Iris dataset to show the differences among samples of different classes and also their relations to the nearby samples in the same classes. Then they derived the classification rules. Some of the methods are useful for radial-basis function (RBF) neural networks. These methods can be used for more complex classification/clustering problems. Also the obtained results from these methods have a higher accuracy [32]. Most existing rule extraction methods leverage multilayer perceptron (MLP) neural networks. These networks are useful for both simple and complex classification/clustering problems and generally present higher accuracy values [27, 33].

Most existing rule extraction methods are used for *classification*. In these issues, the desired output includes discrete values that represent several distinct categories. In this case, the number of neurons in the output layer of the neural network accommodates with the number of available categories [26, 34, 35]. There are also small numbers of procedures that can be used for *regression* problems. In these studies, unlike classification, the desirable output is continuous. In this case, there is only one neuron in the output layer of the neural network which produces the result [36, 37].

There are three types of overall approaches for rule extraction: the *Pedagogical approach* considers the entire neural network as a black box, and extracts the desired rules regardless of the operations within the network, and only with respect to its inputs and outputs [27, 38]. In the *Decompositional approach*, the rules are obtained by decomposing some parts of the neural network (input,

hidden, and output) and considering the conducted operations in each section [39–41]. And finally the *Eclectic approach* uses a combination of compositional and pedagogical approaches [20].

3 Rule extraction using neural networks

The existing rule extraction methods are useful for simple problems and can only support less complex in datasets. Although a neural network is capable of classifying more complex data, but rule extraction from this data leads either to a very large number of rules or rules with low accuracy results. On the other side, the complexity in fatty liver severity detection based on clinical data is significant, and existing methods cannot achieve the appropriate rule set. We use a Four-Step Rule Extraction algorithm which can derive rules from ANN in complex problems. This method which is a decompositional approach is applicable to multilayer perceptron (MLP) neural networks and is capable of extracting IF–THEN rules for classification problems.

FSRE has four main phases. The first phase is the *Data preprocessing/representation* phase in which the input data are normalized using data mining techniques, the second phase is the *Model learning* in which the main classification operation is performed by training an MLP neural network. The *Pruning* phase comes next in which some of the less importance connections and neurons in the network are pruned. Finally, the last phase is *Rule extraction* which derives rules from the pruned neural network. The details of the process will be explained in the next subsections.

3.1 Data preprocessing/representation

In most problems, raw data in its initial form may not lead to ideal classification results, and usually some preprocessing techniques are required for simplification and increasing the accuracy. The goal of this step is to process the initial data such that the neural network can classify the data with a maximum accuracy. To this aim, some of the data mining techniques may be used, including aggregation, sampling, dimensionality reduction, feature subset selection, feature creation (extraction, mapping, and construction), discretization, and feature transformation [42]. Later in Sect. 4, we describe the specific operations performed on the dataset of this study.

3.2 Model learning

The second phase includes creation, training, and validation of a learner model. The method uses an MLP neural network with a standard structure (only one hidden layer)

as a learner model for data classification. A three-layer (input, hidden, and output layers) fully connected neural network is constructed. The structure of this neural network is determined according to the dataset. The number of neurons in the input and output layers is equal to the number of features and the number of output classes, respectively. The hidden layer size is gradually incremented until the desired accuracy is obtained based on the validation subset of data. All weights (edges value) in the neural network are initialized to small random values. The value of bias is equal to 1, and only the hidden layer contains a bias node. The tangent hyperbolic and linear functions are used as the activation functions of the hidden and output layers, respectively.

The back-propagation algorithm is used for training the neural network, and the network performance is calculated using the Mean Square Error (MSE) measure. The generalized error is calculated using unseen samples of the test set.

3.3 Pruning

Usually the neural networks are fully connected, but not all of these connections are significant. Some may even be ineffective for predicting the neural network output [19]. Rule extraction from a fully connected neural network is very difficult (or impossible), and the final result has low comprehensibility. The purpose of neural network pruning is to remove the connections and nodes that are less useful in predicting the output. After pruning, the neural network structure will be simpler and contains only the most dominant connections and nodes.

First the connections between the hidden and output layers are pruned. This process specifies the effective hidden neurons for each output class. Afterward, the ineffective input edges for each hidden neuron are removed. This process will be continued until the final neural network efficiency is acceptable.

The continuous output values of the hidden neurons are one of the critical issues in the rule extraction methods that complicates perception of neural network calculations. The output values of hidden neurons are in the interval $[-1, 1]$ because of applying the tangent hyperbolic activation function. The outputs of hidden neurons per each input data are converted to discrete values using one or more threshold values (Such as Fig. 1).

3.4 Rule extraction

The last phase of the FSRE method is rule extraction. In this phase, a set of simple rules is extracted from structure of the pruned neural network. This phase contains four steps described below.

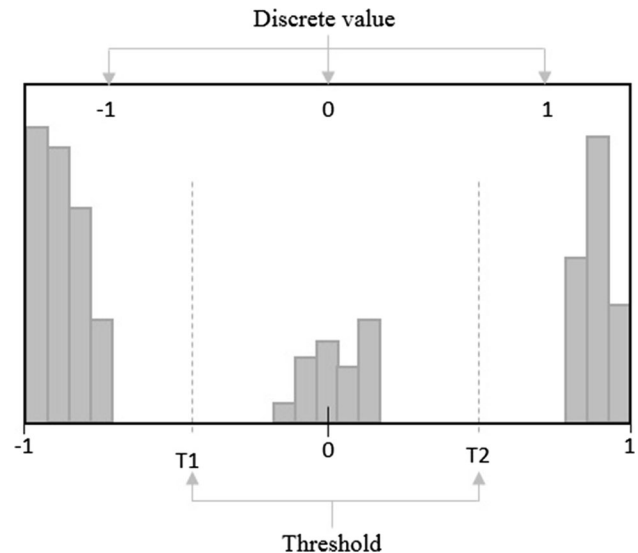


Fig. 1 Discretization of the output value for one hidden neuron according to the threshold values

Step 1 Determining hidden layer patterns of input samples: input samples are applied to the network to obtain the discrete output values from the hidden layer. These discrete values constitute the hidden layer *pattern of the sample*. In Fig. 2, an example of pattern generation is shown for input samples in a pruned neural network. In this example, the input samples have three distinct features and belong to output classes. The trained neural network used for this problem initially had four neurons in the hidden layer, but only two neurons (H2 and H4) remained after pruning. Moreover, the second feature of the data was also pruned and was not further used in later steps of rule extraction. Discretization of the hidden layer outputs created two and four discrete values for neurons H2 and H4, respectively. By applying each sample to this neural network, the pattern of that sample was obtained and shown in the rightmost table of Fig. 2.

Step 2 Sample classification: Each collection of samples with identical hidden layer patterns constitutes a category. Each category is associated with one output class according to the majority of network predicted outputs for samples of that category. Hence, input samples with the same hidden layer pattern would belong to the same class. Notably, several different categories may be assigned to one particular class.

Figure 3 shows the classification of the samples in the previous example based on the pattern obtained from the neural network. In this example, it is seen that three distinct patterns are created for the samples. Samples with the same pattern belong to the same category. The categories (and the respective samples) are associated with output classes,

Input Sample	Output
1 5 7	C1
6 4 3	C2
5 1 3	C2
3 2 4	C2
2 7 1	C1
6 8 3	C2
2 6 7	C1

Hidden Neuron	Discrete Values
H1	pruned
H2	-1 1
H3	pruned
H4	-1 0 1

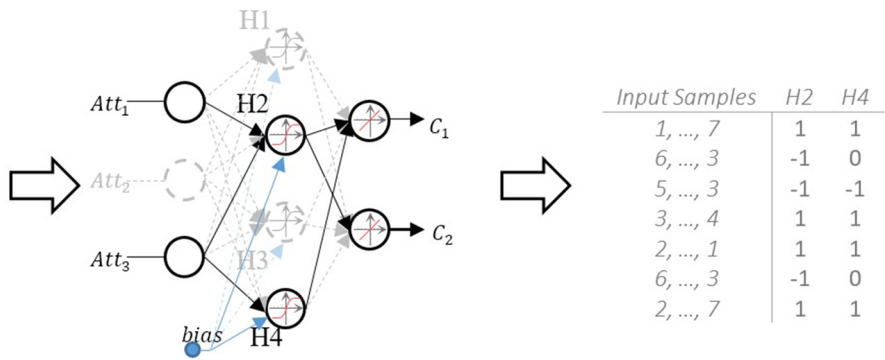
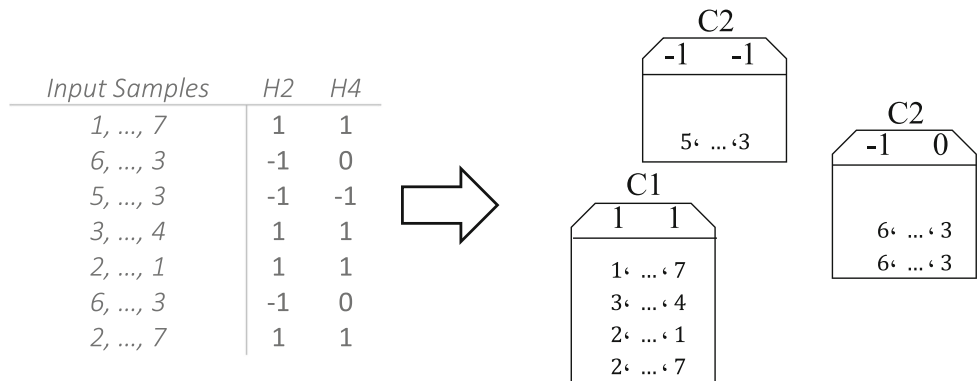


Fig. 2 Determining the pattern for each input sample

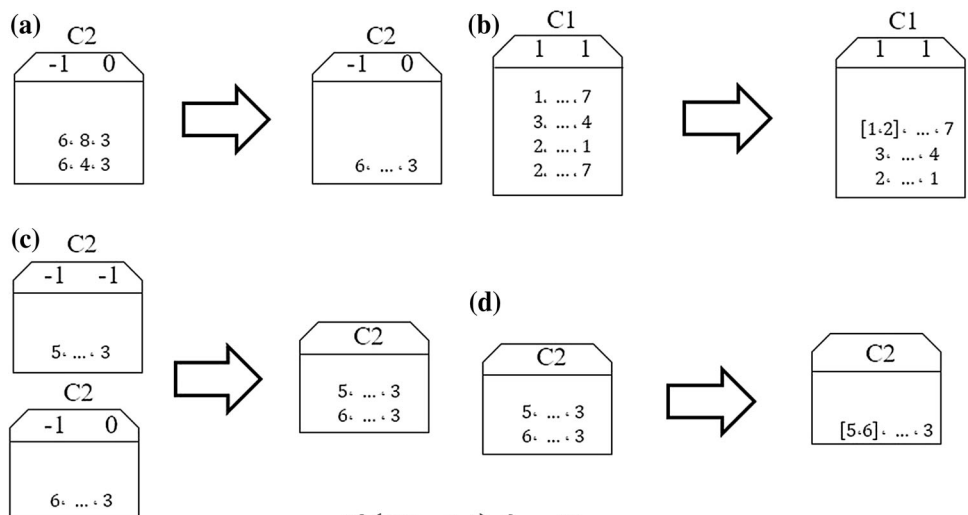
Fig. 3 Classifying input samples according to their pattern



Step 3 Pruning and combining: Several categories may be associated to the same class, and also because of pruning input features, duplicate samples may exist in some categories. Therefore, we need to handle these issues, as stated below, to create the final rule set:

- Removing the duplicate samples in each category. In Fig. 4a, samples [6,8,3] and [6,4,3] are similar because of pruning the second feature. In this case, one of the samples is deleted.
- Combining similar samples to create a more general case (or rule). In Fig. 4b, samples [1, ..., 7] and [2, ..., 7] have

Fig. 4 Management of classes for the production of final rules



if ($Att_1 \leq 4$) then C1
 if ($Att_1 > 4$) then C2

been combined together due to having same value in the third feature and successive values in the first feature.

- Combining categories that belong to the same output class. In Fig. 4c, the two created categories in class C2 are combined together.
- If similar cases (or rules) exist in the new categories, they are combined/pruned to create more general cases (or rules). In Fig. 4d, samples $\{5, \dots, 3\}$ and $\{6, \dots, 3\}$ produced in the new category have been combined together for having the same values in the third, and successive values in the first feature.
- Presenting the classification rule set for each output class. In the presented example, final rules for the classification of samples are produced according to the categories obtained for class C1 (Fig. 4b) and the category produced for Class C2 (Fig. 4d). In this example, values of the first feature for the two output classes were separated, and there was no need to use the third feature in the production of rules (Fig. 4e).

Step 4 Default rule: among the classes, the one with the largest number of samples, or the largest number of rules, or most classification error value is identified. This output class is considered as a default response (rule) for input samples. If an input sample is not supported by any other rules, the default rule is applied. This process reduces the number of rules and improves the results.

Finally, the rule set is constructed and can be applied to unseen (test) data. This rule set can be used on new data instead of the neural network if its efficiency is acceptable. We further analyze the performance of FSRE on two common datasets and compare its accuracy with other methods.

4 Evaluation results

In this research, the proposed method is used for NAFLD severity detection based on the clinical parameters. The FibroScan result is considered as the desired output for each sample (patient). This result consists of five severity levels introduced previously in Table 1. Four different neural networks were used to detect the four severity levels (F1–F4). If an input sample is not classified as belonging to any of the four levels, it is labeled as “healthy” and belongs to the default level (F0). In this section we present the evaluation and the obtained results.

4.1 Dataset

The collected dataset contains 726 samples (patients). The following 17 features were recorded for each sample: *personal information* (sex, age, height, and weight), *blood*

test parameters (total-cholesterol (T-CHOL), high-density lipoprotein (HDL), low-density lipoprotein (LDL), Triglyceride (TG), blood creatinine (CRE), glomerular filtration rate (GFR), blood urea (BU), alkaline phosphatase (ALP), gamma-glutamyl transpeptidase (GGT), fasting blood sugar (FBS), white blood cell (WBC), and platelets (PLT)), and *ultrasonography result* (Ultra_Score). Each patient is classified to one of the five severity levels. Table 2 presents the output classes and number of samples in each class.

4.2 Efficiency parameters

The accuracy, sensitivity, and specificity metrics are used to measure efficiency of the propose method. The metrics are calculated as follows:

$$\text{Accuracy} = \frac{(\text{TP} + \text{TN})}{(\text{TP} + \text{TN} + \text{FP} + \text{FN})} = \frac{\text{Number of Correctly Classified samples}}{\text{Total number of samples}} \quad (4)$$

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \quad (5)$$

$$\text{Specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \quad (6)$$

In Eqs. (4)–(6), the TP, TN, FP, and FN stand for number of true positive, true negative, false positive, and false negative results, respectively.

4.3 Preprocessing

The dataset includes 17 basis features in which some of them may be irrelative for the classification. So the importance and priority of these features are initially examined, then the appropriate features are selected. We used information gain (IG) [43] to rank the features. The information gain is a statistical property based on entropy. Once the entropy of a feature varies widely for different samples in a category, it indicates that the feature is unstable; hence, it is not appropriate for identifying that category. Table 3 represents the information gain results, and the priorities of various features in different categories.

The results presented in Table 3 show that the superior features have IG scores near to zero (<0.05). Evaluation of the method using these features shows low sensitivity values in classification (Table 4).

Therefore, solely using these features is not suitable for disease severity detection. To increase the efficiency of the system, another feature named *Forns_Score*, proposed by Forns et al. [14], is employed as below:

Table 2 Output classes and number of samples in each class

Output class	Class F0	Class F1	Class F2	Class F3	Class F4
Number of samples in each class	415	151	132	23	5

Table 3 Information gain score for some superior features

Feature number	F1	F2	F3	F4
1	LDL (0.007)	FBS (0.025)	HDL (0.012)	HDL (0.008)
2	Weight (0.005)	Ultra-Score (0.024)	T-CHOL (0.009)	Ultra-Score (0.006)
3	HDL (0.004)	HDL (0.016)	Weight (0.006)	Weight (0.004)
4	Ultra-Score (0.003)	Weight (0.012)	LDL (0.006)	LDL (0.004)
5	Height (0.002)	T-CHOL (0.011)	Ultra-Score (0.003)	T-CHOL (0.004)
⋮	⋮	⋮	⋮	⋮

Table 4 Result of the neural network on the 17's basic features

Classes	Number of hidden neuron	Sensitivity (%)	Specificity (%)
F1	23	78.14	89.73
F2	20	62.12	92.24
F3	19	52.17	98.86
F4	10	60.00	99.58

$$\begin{aligned}
 \text{Forns_Score} = & 7.811 - 3.131 \times \ln(\text{PLT}) + 0.781 \\
 & \times \ln(\text{GGT}) + 3.467 \times \ln(\text{Age}) - 0.014 \\
 & \times (\text{TCHOL})
 \end{aligned}
 \tag{7}$$

The research conducted in [14] designed a noninvasive method aimed to discriminate between patients with and without significant liver fibrosis (Level F2–F4 vs. F0–F1). Using the *Forns_Score* on the dataset, 82.35% of patients in the significant levels (F2–F4) can be successfully detected. Also the *Body Mass Index (BMI)* feature was employed by the following formula and added to dataset.

$$\text{BMI} = \frac{\text{Weight}_{\text{kg}}}{\text{Height}_{\text{m}}^2}
 \tag{8}$$

The F0 class was considered as the default answer, because it has more number of samples among the classes. So, it is assumed that the input sample by default belongs to this class if it does not satisfy any other rule. The four separate detector systems are considered for the four disease levels (based on Table 1). The seven more effective features selected by the information gain score are represented in Table 5.

After choosing the features, an effective method is needed to convert the continuous values to discrete values. Many methods were presented for this purpose in recent years (e.g., CAIM [44], Fast-CAIM [45], Modified-CAIM [46], and Ur-CAIM [47]). In this research, the ur-CAIM method is used because of its performance results compared to the other

methods. The discretization operation is done separately for each detector. Afterward, the new features values are normalized to the unit interval. The general normalization method is used to value normalization.

$$\text{NormData} = \frac{(\text{Data} - \mu_{\text{Data}})}{\sigma_{\text{Data}}}
 \tag{9}$$

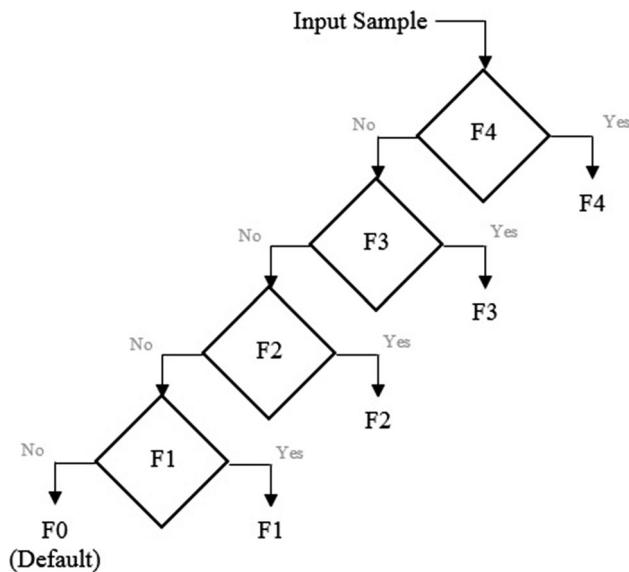
After the above-mentioned preprocessing operations, the data are ready to be applied to the neural network. For the implementation, 70% of the dataset is used for training, 15% is used for validation, and the rest are used for testing. The proportion of the different classes in each subset is consistent with the proportion of these classes in the dataset.

4.4 Results

In this stage, four detector systems are considered to the four existing disease levels (except for F0 level). The structure of the neural network for each detector is decided independently. The parameters of these neural networks will be assigned according to the method mentioned in Sect. 2. Each detector system will solely decide on its specified class (one vs. all). After the training phase, the efficiency value is calculated using the whole data, then all of the additional connections and neurons are pruned. The input sample will be applied to the detectors in a hierarchical manner (and not simultaneous). The order of applying a sample to these detectors is based on the probability of each class as shown in Fig. 5.

Table 5 Features applied to the neural network rule extraction

No.	Feature	Complete name
1	Ultra_Score	Ultrasound test result
2	HDL	High-density lipoprotein
3	LDL	Low-density lipoprotein
4	TG	Triglyceride
5	FBS	Fasting blood sugar
6	BMI	Body mass index
7	Frons_Score	Frons et al. [14] detector for significant liver fibrosis (F2–F4 vs. F0–F1)

**Fig. 5** Applying the input sample to the detector systems as hierarchical form

The advantage of the hierarchical scheme over the simultaneous one is that there is no uncertainty and subscription in the detector systems' decisions. The errors will also be reduced in this manner, because each sample is exactly member of a special class. Table 6 represents the structure of neural networks and the results of training and pruning phases. At the end, the discrete models are created for the samples, and classification rules are obtained for each class (see the results in Table 7).

The results in Table 6 indicate that the neural networks (used as detector systems) can detect the data during the training phase very well. Also, these neural networks will obtain more ideal results with a simpler structure for serious cases of the disease (i.e., F3 and F4). The classification accuracies for classes F1 and F2 are above 80% which are better than the previous works. The results of the pruning phase also indicate that the sensitivity of the detectors has increased; even with the elimination of some connections, the obtained results are congruous and acceptable.

In Table 7, it is shown that the neural networks with a simpler structure lead to the production of a lower number of rules, and the received results of these rules are almost equal to the results of pruning phase (Table 6). These results are ideal for effective levels of disease (F2, F3, and F4) and are acceptable for the early stage of disease (F1). The set of extracted rules are as follows:

- Class F4:
if(LDL < 216) and (FornsScore \geq 115.01) then ClassF4
- Class F3:
if(UltraScore = [1or7]) and (LDL < 178.5) and (FornsScore \geq 109.035) then ClassF3
if(UltraScore = [1or3or5or7]) and (LDL \geq 178.5) and (69.8 \leq FBS < 220.7) and (FornsScore \geq 109.035) then ClassF3
if(2 \leq UltraScore \leq 6) and (LDL < 178.5) and (69.8 \leq FBS < 220.7) and (FornsScore \geq 109.035) then ClassF3
- Class F2:
if(LDL \geq 71.5) and (FornsScore \geq 106.485) then ClassF2
if(3 \leq UltraScore \leq 5) and (LDL < 65.5) and (FornsScore \geq 106.485) then ClassF2
if(UltraScore = 1) and (65.5 \leq LDL < 71.5) and (FornsScore \geq 106.485) then ClassF2
- Class F1:
if(LDL \geq 105.5) and (TRIG < 275.5) and (BMI < 32.108) and (105.085 \leq FronsScore < 106.725) then ClassF1
if(LDL < 104.5) and (TRIG < 263) and (BMI < 32.108) and (105.085 \leq FronsScore < 106.725) then ClassF1
if(LDL < 104.5) and (TRIG \geq 275.5) and (BMI < 32.108) and (105.085 \leq FronsScore < 106.725) then ClassF1
if(LDL < 104.5) and (TRIG < 241.5) and (BMI \geq 32.667) and (105.085 \leq FronsScore < 106.725) then ClassF1
- Class F0:
Default is Class F0

Table 6 The obtained results after training and pruning phases

	Number of hidden unit	Training phase result		Pruning phase result	
		Sensitivity (%)	Specificity (%)	Sensitivity (%)	Specificity (%)
Class F1	5	80.79	82.78	84.76	81.21
Class F2	3	94.96	93.60	93.93	93.77
Class F3	3	100	99.28	100	99.28
Class F4	2	100	100	100	100

Table 7 The obtained results from extracted rules

	Number of rules	Sensitivity (%)	Specificity (%)
Class F1	5	99.34	75.65
Class F2	3	93.93	93.77
Class F3	3	100	99.28
Class F4	1	100	100

Table 8 Accuracy of the different methods for NAFLD severity diagnosis

	FSRE (%)	Forns_Score (%)	JRip (%)	J48 (%)
Class F1	80.58	70.11	76.72	67.07
Class F2	93.94	82.51	79.75	72.17
Class F3	99.31	91.32	94.76	93.52
Class F4	100	99.31	98.76	98.62

4.5 Discussion

This section compares experimental results of FSRE with the results of other works. Table 8 compares FSRE results of NAFLD severity detection with those produced using Forns_Score [14], JRip Rule Induction [48], and J48 [49] algorithms on the dataset.

Breast cancer and Wine datasets from UCI database [50] are used to test the proposed method and to evaluate its performance. The obtained results are compared with other popular rule extraction methods including SV-DT [51],

RGANN [52], Rex-P [53], and Rex-M [53]. Considering the results shown in Tables 9 and 10, the FSRE is able to produce a rule set with a higher accuracy and comprehensibility compared to the existing methods.

5 Conclusion

In this research, a new technique was introduced for liver fibrosis diagnosis using neural networks. This technique consists of four main phases, namely data preparation, constructing and training the detectors (neural networks) for various severity levels of the disease, pruning, and rule extraction. The method was used to determine the relationship between clinical parameters and the severity of NAFLD (non-alcoholic fatty liver) disease. After training and eliminating the unnecessary connections in these detectors, the relationships between clinical parameters and levels of NAFLD disease were obtained in the form of a set of rules. In this research, a hierarchical system has been used to appropriately detect the various severity levels of the disease. The ultimate accuracy of these rules for severity levels of F1, F2, F3, and F4 is, respectively, equal to 80.58, 93.94, 99.31, and 100%. The results are superior compared with the existing scoring systems.

In this study, only a few clinical parameters were used for NAFLD diagnostic, and other parameters are ignored. These parameters will be investigated in future works. Also using other base networks and training models such as ELM/Kernel ELM instead of MLP to achieve a better and

Table 9 Comparison of the results obtained for Breast cancer problem

	FSRE	SV-DT [51]	RGANN [52]	Rex-P [53]	Rex-M [53]
No. of rules	2	4	2	3	15.6
Avg. no. of condition	3	2.25	3	–	–
Rule accuracy (%)	96.56	92.98	96.28	95.97	93.43

Table 10 Comparison of the results obtained for Wine problem

	FSRE	SV-DT [51]	RGANN [52]	Rex-P [53]	Rex-M [53]
No. of rules	3	18	3	4	13.8
Avg. no. of condition	2	6.28	3	–	–
Rule accuracy (%)	95.50	81.36	91.01	90.98	92.22

more helpful rule set in real clinical environments can be another direction of future work.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

References

- Stoean C, Stoean R, Lupsor M, Stefanescu H, Badea R (2011) Feature selection for a cooperative coevolutionary classifier in liver fibrosis diagnosis. *Comput Biol Med* 41(4):238–246
- Acharya UR, Raghavendra U, Fujita H, Hagiwara Y, Koh JE, Hong TJ, Sudarshan VK, Vijayanathan A, Yeong CH, Gudigar A (2016) Automated characterization of fatty liver disease and cirrhosis using curvelet transform and entropy features extracted from ultrasound images. *Comput Biol Med* 79:250–258
- Sug H (2012) Improving the prediction accuracy of liver disorder disease with oversampling. In: *Proceeding of the 6th WSEAS international conference on computer engineering and application, and American conference on applied mathematics*, Cambridge, pp 331–335
- Jiang ZG, Tapper EB, Connelly MA, Pimentel CM, Feldbrugge L, Kim M, Krawczyk S, Robson SC, Herman M, Otvos JD (2016) Steatohepatitis and liver fibrosis are predicted by the characteristics of very low density lipoprotein in nonalcoholic fatty liver disease. *Liver Int* 36(8):1213–1233
- Gorunescu F, Belciug S, Gorunescu M, Badea R (2012) Intelligent decision-making for liver fibrosis stadialization based on tandem feature selection and evolutionary-driven neural network. *Expert Syst Appl* 39(17):12824–12832
- Goceri E, Shah ZK, Layman R, Jiang X, Gurcan MN (2016) Quantification of liver fat: a comprehensive review. *Comput Biol Med* 71:174–189
- Siddiqui MS, Patidar KR, Boyett S, Luketic VA, Puri P, Sanyal AJ (2015) Performance of non-invasive models of fibrosis in predicting mild to moderate fibrosis in patients with nonalcoholic fatty liver disease (NAFLD). *Liver Int* 36:572–579
- Afdhal NH (2012) Fibroscan (transient elastography) for the measurement of liver fibrosis. *Gastroenterol Hepatol* 8(9):605
- Gaia S, Campion D, Evangelista A, Spandre M, Cosso L, Brunello F, Ciccone G, Bugianesi E, Rizzetto M (2015) Non-invasive score system for fibrosis in chronic hepatitis: proposal for a model based on biochemical, FibroScan and ultrasound data. *Liver Int* 35(8):2027–2035
- Bril F, Ortiz-Lopez C, Lomonaco R, Orsak B, Freckleton M, Chintapalli K, Hardies J, Lai S, Solano F, Tio F (2015) Clinical value of liver ultrasound for the diagnosis of nonalcoholic fatty liver disease in overweight and obese patients. *Liver Int* 35(9):2139–2146
- Cales P, Boursier J, Chaigneau J, Laine F, Sandrini J, Michalak S (2010) Diagnosis of different liver fibrosis characteristics by blood tests in non-alcoholic fatty liver disease. *Liver Int* 30(9):1346–1354. doi:10.1111/j.1478-3231.2010.02314.x
- Fujiwara Sh, Hongou Y, Miyaji K, Asai A, Tanabe T, Fukui H (2007) Relationship between liver fibrosis noninvasively measured by fibro scan and blood test. *Bull Osaka Med Coll* 35(2):93–105
- Angulo P, Hui JM, Marchesini G, Bugianesi E, George J, Farrel GC (2007) The NAFLD fibrosis score: a noninvasive system that identifies liver fibrosis in patients with NAFLD. *Hepatology* 45(4):846–854
- Forns X, Ampurdanes S, Llovet JM, Aponte J, Quinto L, Martinez-Bauer E (2002) Identification of chronic hepatitis C patients without hepatic fibrosis by a simple predictive model. *Hepatology* 36(4):986–992
- Lok A, Ghany MG, Goodman ZD, Wright EC, Everson GT, Sterling RK (2005) Predicting cirrhosis in patients with hepatitis C based on standard laboratory test: results of the Halt-C cohort. *Hepatology* 24(2):282–292
- Wai Ch, Greenon JK, Fontana RJ, Kalbfleisch JD, Marrero JA, Conjeevaram HS (2003) A simple noninvasive index can predict both significant fibrosis and cirrhosis in patients with chronic hepatitis C. *Hepatology* 38(2):518–526
- Augasta MG, Kathirvalavakumar T (2012) Rule extraction from neural networks—a comparative study. In: *Proceeding of the international conference on pattern recognition, informatics and medical engineering*, Salem, Tamil Nadu. IEEE, pp 404–408. doi:10.1109/ICPRIME.2012.6208380
- Kahramanli H, Allahverdi N (2009) Extracting rules for classification problems: AIS based approach. *Expert Syst Appl* 36(7):10494–10502
- Kamruzzaman SM, Sarkar AM (2011) A new data mining schema using artificial neural networks. *Sensors* 11(5):4622–4647. doi:10.3390/s110504622
- Chorowski J, Zurada JM (2011) Extracting rules from neural networks as decision diagrams. *IEEE Trans Neural Networks* 22(12):2435–2446. doi:10.1109/TNN.2011.2106163
- Kulluk S, Özbaki L, Baykasoglu A (2013) Fuzzy DIFACONN-miner: a novel approach for fuzzy rule extraction from neural networks. *Expert Syst Appl* 40(3):938–946. doi:10.1016/j.eswa.2012.05.050
- Setiono R (1997) Extracting rules from neural network by pruning and hidden-unit node splitting. *Neural Comput* 9(1):205–225
- Setiono R (2000) Extracting M-of-N rules from trained neural networks. *IEEE Trans Neural Networks* 11(2):512–519. doi:10.1109/72.839020
- Tsukimoto H (2000) Extracting rules form trained neural networks. *IEEE Trans Neural Networks* 11(2):377–389. doi:10.1109/72.839008
- Fu X, Wang L (2002) Rule extraction using a novel gradient-based method and data dimensionality reduction. In: *Proceeding of the international joint conference on neural networks*, Honolulu, HI, IEEE, pp 1275–1280. doi:10.1109/IJCNN.2002.1007678
- Kamruzzaman SM, Islam MD (2006) An algorithm to extract rules from artificial neural networks for medical diagnosis problems. *Int J Inf Technol* 12(8):41–59
- Tewary G (2015) Effective data mining for proper mining classification using neural networks. *Int J Data Min Knowl Manag Process* 5(2):65–82
- Yu S, Guo X, Zhu K, Du J (2010) A neuro-fuzzy GA-BP method of seismic reservoir fuzzy rules extraction. *Expert Syst Appl* 37(3):2037–2042
- Wang J, Lim CP, Creighton D, Khorsavi A, Nahavandi S, Ugon J, Vamplew P, Stranieri A, Martin L, Freischmidt A (2015) Patient admission prediction using a pruned fuzzy min–max neural network with rule extraction. *Neural Comput Appl* 26(2):277–289
- Korosec M (2007) Technological information extraction of free form surfaces using neural networks. *Neural Comput Appl* 16(4–5):453–463
- Malone J, McGarry K, Wermter S, Bowerman Ch (2006) Data mining using rule extraction from kohonen self-organising maps. *Neural Comput Appl* 15(1):9–17
- Oh SK, Kim WD, Pedrycz W, Park BJ (2010) Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization. *Fuzzy Sets Syst* 163(1):54–77. doi:10.1016/j.fss.2010.08.007

33. Karabatak M, Ince MC (2009) An expert system for detection of breast cancer based on association rules and neural network. *Expert Syst Appl* 36(2):3465–3469
34. Chan KY, Ling S-H, Dillon TS, Nguyen HT (2011) Diagnosis of hypoglycemic episodes using a neural network based rule discovery system. *Expert Syst Appl* 38(8):9799–9808
35. Karthik S, Priyadarishini A, Anuradha J, Tripathy BK (2011) Classification and rule extraction using rough set for diagnosis of liver disease and its types. *Adv Appl Sci Res* 2(3):334–345
36. Fortuny EJ, Martens D (2012) Active learning based rule extraction for regression. Paper presented at the 12th international conference on data mining workshops (ICDMW), Brussels
37. Widodo A, Shim M-C, Caesarendra W, Yang B-S (2011) Intelligent prognostics for battery health monitoring based on sample entropy. *Expert Syst Appl* 38(9):11763–11769
38. Young WA II, Weckman GR (2010) Using a heuristic approach to derive a grey-box model through an artificial neural network knowledge extraction technique. *Neural Comput Appl* 19(3):353–366
39. Kamruzzaman SM, Islam M (2007) Extraction of symbolic rules from artificial neural networks. *Int J Comput Inf Sci Eng* 1(10):3022–3028
40. Heh JS, Chen JC, Chang M (2008) Designing a decompositional rule extraction algorithm for neural networks with bound decomposition tree. *Neural Comput Appl* 17(3):297–309
41. Plikynas DSL, Rasteniene A (2005) Portable rule extraction method for neural network decisions reasoning. *Syst Cybern Inform* 3(4):79–84
42. Siraj F, Omer EA, Hassan R (2012) Data mining and neural networks: the impact of data representation. In: A Karahoca (ed) *Advances in data mining knowledge discovery and applications*. InTech, Rijeka, pp 463–470. doi:10.5772/51594
43. Roobeart D, Karakoulus G, Chawla NV (2006) Information gain, correlation and support vector machine. In: Guyon I, Nikravesh M, Gunn S, Zadeh LA (eds) *Feature extraction. Studies in fuzziness and soft computing*, vol 207. Springer, Berlin, pp 463–470. doi:10.1007/978-3-540-35488-8_23
44. Kurgan LA, Cios KJ (2004) CAIM discretization algorithm. *IEEE Trans Knowl Data Eng* 16(2):145–153
45. Kurgan LA, Cios, KJ (2003) Fast class-attribute interdependence maximization (CAIM) discretization algorithm. In: *Proceedings of the 2003 international conference on machine learning and applications*, Los Angeles, California, USA, 2003. CSREA Press, pp 30–36
46. Vora Sh, Mehta RG (2012) MCAIM: Modified CAIM discretization algorithm for classification. *Int J Appl Inf Syst IJAIS* 3(5):42–50
47. Cano A, Nguyen DT, Ventura S, Cios KJ (2014) ur-CAIM: Improved CAIM discretization for unbalanced and balanced data. *Soft Comput*. doi:10.1007/s00500-014-1488-1
48. Cohen WW (1995) Fast effective rule induction. In: *Proceedings of the twelfth international conference on machine learning (1995)*, pp 115–123 Key: citeulike:3157878, Tahoe City, California, USA, 1995. pp 115–123
49. Quinlan JR (1992) *Learning with continuous classes*. In: 5th Australian joint conference on artificial intelligence, Singapore, pp 343–348
50. Lichman M (2013) UCI machine learning repository. Irvine, CA, University of California, School of Information and Computer Science. <http://archive.ics.uci.edu/ml>
51. Farquad MAH, Sultana J, Nagalaxmi G, Savankumar G (2014) Knowledge discovery from data: comparative study. *Trans Eng Sci* 2(6):72–75
52. Kamruzzaman SM (2007) RGANN: An efficient algorithm to extract rules from ANNs. *J Electron Comput Sci* 8:19–30
53. Kaczmar UM, Trelak W (2005) Fuzzy logic and evolutionary algorithm—two techniques in rule extraction from neural networks. *Neurocomputing* 63:359–379