Neural Comput & Applic (2017) 28:1329–1342
DOI 10.1007/s00521-016-2567-2

ENGINEERING APPLICATIONS OF NEURAL NETWORKS

Improve credit scoring using transfer of learned knowledge from self-organizing map

Ali AghaeiRad[1] · Ning Chen[2] · Bernardete Ribeiro[1]

Received: 11 January 2016 / Accepted: 17 August 2016 / Published online: 26 August 2016

Abstract Credit scoring is important for credit risk evaluation and monitoring in the accounting and finance domain. For financial institutions, the ability to predict the business failure is crucial, as incorrect decisions have direct financial consequences. A variety of pattern recognition techniques including neural networks, decision trees, and support vector machines have been applied to predict whether the borrowers should be considered a good or bad credit risk. This paper presents a hybrid approach to building the credit scoring model and illustrates how the unsupervised learning based on self-organizing map (SOM) can improve the discriminant capability of feedforward neural network (FNN). Within the hybridization scheme, the knowledge (i.e., prototypes of clusters) found by SOM is transferred as input to the subsequent FNN model. Four real-world data sets are used in the experiments for credit approval problems. By varying the parameters, the experimental results demonstrate the predictive model built by the hybrid approach can achieve better performance than the stand-alone FNN particularly when a limited amount of labeled data is available. This gives some insights on how to construct more accurate predictive models when the data collection is difficult in some financial applications. A complete and unique graphical visualization technique is shown which better outlines the trade-off between distinct metrics and attained performance.

Keywords Credit scoring · Self-organizing map · Feedforward neural network · Hybrid classification · Information visualization

1 Introduction

Credit scoring was introduced in the 1940s for the first time and over the decades has grown up significantly. In the 1960s, with the creation of credit cards, credit scoring in the credit granting process became more important for banks and other credit card issuers. Nowadays credit scoring has been widely used as an aid decision tool for diverse purposes [25] such as approving personal loan applications and predicting the bankruptcy of companies.

The credit scoring problem can be stated as follows: given a number of customers labeled as bad or good credit and a set of financial variables over a given period, build the predictive models to determine the credit score and creditworthiness of the customers.

Credit scoring models have evolved over time based on consumer behavior. Nowadays mortgage crisis and the economic downturn make credit scoring evermore important. Financial industry focuses on developing effective systems that can evaluate and manage the credit. There are different methods of credit scoring regarding how to predict the credit exactly for various purposes with special attention on classification tasks.

✉ Ning Chen
nchenyx@outlook.com

Ali AghaeiRad
ali@dei.uc.pt

Bernardete Ribeiro
bribeiro@dei.uc.pt

[1] CISUC, Department of Informatics Engineering, University of Coimbra, Polo II, Pinhal de Marrocos, 3030 290 Coimbra, Portugal

[2] College of Computer Science and Technology, Henan Polytechnic University, 2001 Century Avenue, Jiaozuo 454003, Henan, People's Republic of China

Before machine learning was introduced to credit scoring, traditional statistical classification methods have been used extensively for tackling this problem. In [9], a statistical model for discrete choice of consumer loan default and credit card expenditure was derived. However, due to the massive increase in applicants which led to the overload of big data, it is often impossible to build such handcraft models. Thus, with the advance in information and computer technology there is an increasing need for machine learning techniques as a popular alternative in credit scoring tasks [19]. K-nearest neighbor (KNN) was proposed in [31] with an adjusted version of Euclidean distance metric for assessing the consumer credit risk. More recently, a thorough comparison of classification performance was performed among several credit scoring models including logistic regression and decision tree [30]. Later in [10], support vector machine was used for credit scoring and received high accuracy.

Artificial neural networks (ANNs) have shown attractive capability in credit scoring compared with other competitive approaches [24]. A variety of ANN methods of different structures have been proposed, including feedforward neural network (FNN), self-organizing map (SOM), learning vector quantization (LVQ), radial basis function network (RBFN), probabilistic neural network (PNN), and support vector machine (SVM). An investigation of supervised neural network models was presented for credit risk evaluation under different learning schemes [15]. Among all neural networks, FNN is most widely used in the examined credit scoring area. For example, FNN is used to predict the credit rating category for Taiwan and US markets [11]. Self-organizing maps (SOMs) have been recognized as a powerful tool to find the intrinsic patterns from complex financial data. The capability of SOMs and their supervised variants was demonstrated in comparison with statistical and other intelligent methods [23]. In [4], SOM was used to discover and visualize the bankruptcy trajectory patterns of companies over several years through a two-step clustering process.

A recent approach in the area of machine learning is the combination of supervised and unsupervised methods to improve the classification accuracy, although it was seldom used in financial risk area. In [28], a comprehensive review was presented concerning the hybrid computing techniques applied to financial credit scoring. Normally in a hybrid system, several approaches are exploited in the analysis and only one is employed for the final prediction. A scheme of hybrid classification is clustering combination mode in which the training data are processed by unsupervised clustering to filter out the nonrepresentative samples, and subsequently fed to a supervised classification model to identify the patterns. In [7, 12, 14], a fusion between supervised classification and unsupervised clustering was used to improve the classification accuracy.

In this paper, a hybrid classification approach is proposed in another way to predict the credit of customers. A SOM is trained in an unsupervised way and then labeled by voted method in a supervised way. For each neuron, the Voronoi set refers to the set of instances which take the underlying neuron as the best-matching unit (BMU) among others. The prototypes of SOM neurons that have nonempty Voronoi set with nonuniform class distribution (i.e., the majority class has more votes than the other) are selected as input to the subsequent FNN along with the original training data. The rational behind using SOM prior to FNN is that the prototypes found by SOM usually represent the center of clusters and therefore contribute to constructing accurate classification models. Three real-world data sets from UCI Machine Learning Repository and a Diane data set of French small and medium enterprises (SME) are used in the experiments for credit approval problems. By varying the parameters (size of training data sets, number of neurons of both FNN and SOM), the results demonstrate the proposed hybrid classification approach produces better performance than the stand-alone FNN using only training data as input. This means the prototypes found by SOM are good supplement as input to FNN particularly when the training data are in small size.

The rest of this paper is organized as follows. Section 2 introduces the background of ANNs with focus on FNN and SOM, and information visualization techniques. Section 3 presents the design of experiments including the data sets under exploration, the framework to build the predictive model in a hybridization scheme, and the performance evaluation. In Sect. 4, some experimental results are reported. The performance of the proposed approach is compared with the baseline FNN in terms of accuracy, sensitivity, and specificity. Lastly, the contributions and future improvements are given in Sect. 5.

2 Research background

We will briefly review in this section the ANN methods, in particular FNN and SOM which are used to construct a hybrid classification model for credit risk scoring tasks in this paper. The widely used information visualization techniques are introduced to display, compare, and analyze the performance results.

2.1 Feedforward artificial neural network

Information is processed by an ANN in a similar way to the human brain. There has been recently a renaissance of neural networks due to their remarkable capabilities in performing tasks related to perception with many layers of processing information. Although deep neural networks are extremely good for computer vision and speech recognition models, shallow networks are able to reduce resources requirement which convey for the problem handled in this work. In fact, a shallow ANN reduces memory requirements by storing a simple transfer function representing output values for multiple nodes. It typically consists of input nodes, output nodes, and hidden nodes.

A node can be referred as a data processing device that receives multiple inputs and generates a single output based on those inputs. During this process, input nodes receive one input, while both hidden nodes and output nodes receive several inputs. The hidden nodes do not receive any input data from sources outside and nor do they send output to any devices outside the ANN.

A FNN is a multilayer ANN where the connections between the nodes do not form a directed cycle and are unilateral. In this network, information always moves in one direction from the input nodes, through the hidden nodes and to the output node and never goes backwards [5]. Figure 1 shows an example of FNN composed of n input nodes, m nodes in one hidden layer, and k output nodes. FNN is widely employed as a classification tool in a variety of application domains, including medical analysis, credit scoring, pattern recognition, speech recognition, handwriting recognition, product inspection, drug discovery and development, biological classification, natural language processing, document classification, and network security [6, 8, 20]. In credit risk assessment, FNN is used to develop credit risk models from the historical data and predict the future corporate bankruptcies [1, 2, 27]. FNN is trained by a backpropagation learning algorithm:

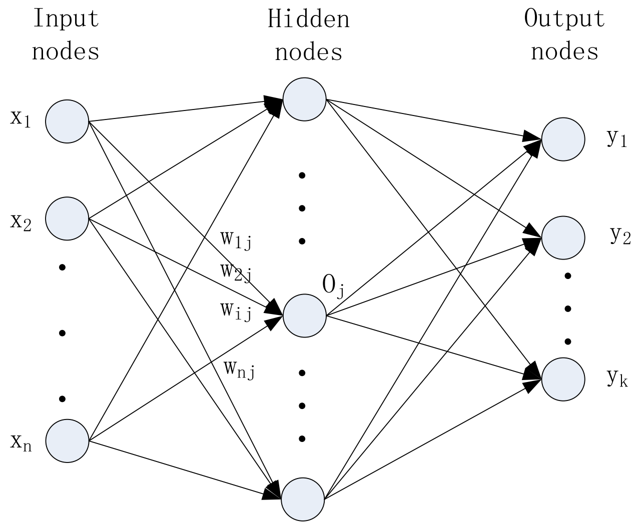

Fig. 1 Example of a feedforward neural network

1. Initialize the connection weights w_{ij} and bias θ_j as small random values.
2. Present the input vector of pattern X_i and the corresponding target vector Y_i to the network.
3. Calculate the output of each node from the first layer to the last layer using an activation function f. For each input node j, $O_j = X_j$; for each node j in hidden layer or output layer,

$$O_j = f\left(\sum_i (w_{ij}O_i + \theta_j)\right).$$

4. Calculate the error for every node in backward order. For an output node j, Err_j is computed by the network output and the target value:

$$\mathrm{Err}_j = O_j(1 - O_j)(Y_j - O_j).$$

For node j in other layers, Err_j is estimated by the errors of nodes in the next layer:

$$\mathrm{Err}_j = O_j(1 - O_j)\sum_k w_{jk}\mathrm{Err}_k.$$

5. Update the connection weights w_{ij} and bias θ_j for node j, where η is the learning rate:

$$w_{ij} = w_{ij} + \eta \mathrm{Err}_j O_i$$
$$\theta_j = \theta_j + \eta \mathrm{Err}_j.$$

6. Repeat from step 2 until the termination condition is satisfied.

2.2 Self-organizing map

A SOM invented by Teuvo Kohonen [17] is a type of ANN where the neurons are set along a grid. SOMs are different from other ANNs in the sense that they use a neighborhood function to preserve the topological properties of the input space. It provides a visual way to understand high-dimensional data in a low-dimensional output space.

SOM is able to reduce the amount of data and simultaneously project the data nonlinearly onto a lower dimensional array. The neurons are distributed on a regular grid of usually two dimensions. Each neuron is related to a reference vector of the same dimension as the input data vectors, reflecting the association strength with the input vectors (see Fig. 2). A neighborhood kernel function $h_{cp}(t)$ describes the topological relation of neurons, and the learning rate $\alpha(t)$ affects the speed of learning. In each iteration of the training process, the reference vectors are updated in such a way that the best-matching neuron and its neighbors on the grid are dragged toward the input. As a result, the neurons are topologically ordered on the grid, where instances that have similar features in the input space will be projected to the

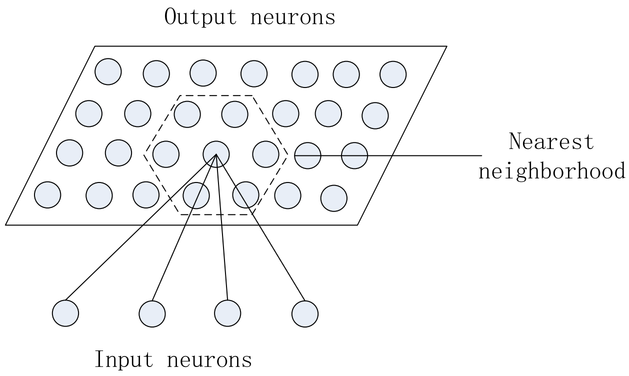

Fig. 2 Example of self-organizing map composed of 4 input neurons and a [7 × 4] output grid

neurons located close to each other in the grid space. Due to capability of preserving the topological properties of the input space, SOMs are usually used for data clustering and visualization in different application domains, including speech recognition, clinical voice analysis, satellite images analysis, document organization and retrieval, DNA sequencing analysis, multimedia and Web mining, and transportation industry [13, 21, 32]. In bankruptcy prediction problems, SOM and its supervised variants can discover and visualize the bankrupt patterns from the financial data [3, 29].

Assume the map consists of m neurons, each associated with a prototype m_{p}, the SOM is learned in an unsupervised and competitive way.

1. For $p = 1, \ldots, m$, initialize the map prototypes as m_{p};
2. For each input vector x, calculate Euclidean distance of x and all neurons and project x to the best-matching unit (BMU):
   $$c = \underset{1 \le i \le m}{\operatorname{argmin}} \, ||x - m_i||$$
3. For $p = 1, \ldots, m$, update the prototypes proportional to the learning rate $\alpha(t)$ and the neighborhood function $h_{c,p}(t)$:
   $$m_{\mathrm{p}}(t+1) = m_{\mathrm{p}}(t) + \alpha(t) h_{c,p}(t)(x - m_{\mathrm{p}})$$
4. Repeat from Step 2 a few iterations until the termination condition is satisfied.

Table 1 Description of four data sets used for credit approval

Data set	#Instances	#Attributes		#Bad	#Good
		Categorical	Numeric		
Australia	690	8	6	383	307
German	1000	7	13	300	700
Japanese	690	9	6	357	296
French	1200	0	30	600	600

Table 2 Financial ratios of Diane database

Variable	Description
x_1	Number of employees previous year
x_2	Capital employed/fixed assets
x_3	Financial debt/capital employed
x_4	Depreciation of tangible assets
x_5	Working capital/current assets
x_6	Current ratio
x_7	Liquidity ratio
x_8	Stock turnover days
x_9	Collection period days
x_{10}	Credit period days
x_{11}	Turnover per employee k EUR
x_{12}	Interest/turnover
x_{13}	Debt period days
x_{14}	Financial debt/equity
x_{15}	Financial debt/cashflow
x_{16}	Cashflow/turnover
x_{17}	Working capital/turnover days
x_{18}	Net current assets/turnover days
x_{19}	Working capital needs/turnover
x_{20}	Export
x_{21}	Added value per employee in k EUR
x_{22}	Total assets turnover
x_{23}	Operating profit margin
x_{24}	Net profit margin
x_{25}	Added value margin
x_{26}	Part of employees
x_{27}	Return on capital employed
x_{28}	Return on total assets
x_{29}	EBIT margin
x_{30}	EBITDA margin

2.3 Information visualization techniques

Information visualization refers to the visual representation and interactive techniques for users to understand the data in an easy way. Information visualization plays an important role in data mining applications. In the preprocessing phase, information visualization is used to explore the distribution and property of the data. In the mining phase, information visualization helps to detect the process of model generation and adjust the parameter setting. In the validation phase, information visualization is able to display the discovered knowledge and compare the performance among various learning algorithms. Due to the complexity of result analysis, there is a growing demand for effective visualization techniques to better understand the information hidden in the research results.

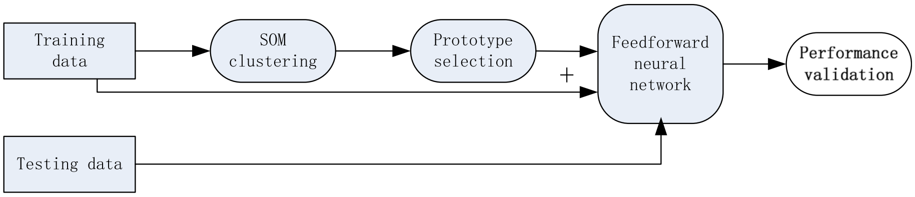

Fig. 3 General framework of the experimental design

The information visualization techniques can be divided into two categories. One is the visualization techniques of original data, including 1D visualization (histogram, box plot etc.), and two- or multiple-dimensional visualization (scatter plot, dendrogram, heatmap, plot matrix, cladogram, graph drawing, hyperbolic tree, parallel coordinates, tree mapping, cobweb, etc.). Among these techniques, cobweb is a radial multilevel pie chart adequate for large data set. It divides a circle chart into equal-width pies whose length represents the numeric data. The cobweb graph has advantages on displaying different levels (or dimensions) of information in one graph. In [22], it was used to visualize and compare the contribution of member SVMs to the ensemble classifier. In this paper, we use cobweb as an information visualization technique to display the research results of different classifiers, data, and metrics in a compact representation. The other is visualization of high-dimensional data including principal component analysis, multidimensional scaling, SOM, manifold learning, which transform the data from a high-dimensional space to a low-dimensional space through a linear or nonlinear projection while approximating the relative distance between samples. These techniques are able to visualize any dimensional data theoretically with the risk of information loss.

3 Methodology

In this section, we will describe the data sets used in the experiments, the general framework of hybrid classification, and the metrics for performance evaluation.

3.1 Data set description

In the experiments described in Sect. 4, we used three real-world data sets: Australia, German, and Japanese taken from UCI Machine Learning Repository for credit approval [18]. For each data set, the class composition of instances and the attributes are characterized in Table 1. The bad class indicates the refused credit and the good class indicates the granted credit. The instances containing missing values are deleted from Japanese data set in the preprocessing. Diane is a database of French small to middle companies over the years from 2002 to 2006. It contains complete information about the financial ratios and the status (bankrupt or normal) in the year 2007. The financial ratios are described in Table 2. A balanced subset of 1200 samples is used in the experiment. For all data sets, the categorical attributes are converted to continuous numbers, and all attributes are normalized to zero mean and unit variance. Finally we convert the bad credit to 1 and the good credit to 0.

3.2 Experimental design

The proposed hybrid approach is implemented based on Neural Network toolbox and SOMToolbox [16] in MATLAB environment. The general framework of the experimental design is outlined in Fig. 3. In details, the experiments are performed in the following steps:

Step 1 A parameter *TrRate* is set as the percent of samples selected from the original data set. By varying the value of *TrRate* from 5 to 60 %, we can get a series of training data sets in different sizes.

Step 2 The data set is divided into two subsets randomly dependent on the value of *TrRate*. The training data set is used for model training, and the testing data set is used for model validation.

Step 3 In each trial, SOM is applied to the training data set in an unsupervised manner and then labeled in a supervised manner. The training samples are projected to the best-matching unit (BMU), i.e., the winner neuron with the closest reference vector to the input, that results into a number of Voronoi sets. Each Voronoi set is composed of the instances which take the underlying neuron as the BMU.

Step 4 By detecting the composition of Voronoi sets, the neurons are selected which have nonempty Voronoi set with nonuniform class distribution (i.e., the majority class has more votes than the other) and assigned by the majority vote method.

Table 3 Contingency matrix of prediction results

Real class	Predicted class	
	Positive	Negative
Positive	tp	fn
Negative	fp	tn

Positive class: bad credit, negative class: good credit

The prototypes as well as the label serve as a supplement to the training data in the next step.

Step 5 A FNN is trained taken as input the training data and the prototypes.

Step 6 In the validation, each sample of the testing data set is fed to the learned FNN. Then the accuracy, sensitivity, and specificity are calculated by comparing the predicted class with the real class.

Step 7 For each configuration of parameter *TrRate*, 10 trials are performed with random division of training data and testing data. For each training data set, 10 models are trained with different

Table 4 Performance comparison between the two methods using Australian data set (mean values are in percent and significance is at 5 % level)

TrRate (%)	n_s	n_p	Accuracy			Sensitivity			Specificity		
			FNN	SOM+FNN	p value	FNN	SOM+FNN	p value	FNN	SOM+FNN	p value
5	31	19	73.57	76.95	$8e^{-6}$*	76.66	78.81	$3.9e^{-3}$*	71.85	75.91	$3.7e^{-5}$*
10	40	28	75.81	77.66	$4.1e^{-4}$*	77.99	80.06	$6.5e^{-4}$*	74.26	75.44	0.03*
20	54	40	81.07	82.47	$1.1e^{-3}$*	85.44	86.58	0.0234*	77.45	78.45	0.03*
40	83	62	83.90	84.67	0.0178*	85.82	86.22	0.2083	82.44	83.05	0.09
60	104	87	85.76	85.88	0.1416	87.27	87.09	0.192	84.21	84.60	0.038*

Table 5 Performance comparison between the two methods using German data set (mean values are in percent and significance is at 5 % level)

TrRate (%)	n_s	n_p	Accuracy			Sensitivity			Specificity		
			FNN	SOM+FNN	p value	FNN	SOM+FNN	p value	FNN	SOM+FNN	p value
5	35	22	62.76	64.82	$6.2e^{-4*}$	74.16	75.51	$4e^{-6*}$	40.3	42.35	$1.5e^{-3*}$
10	48	32	64.34	65.65	$4.7e^{-4*}$	73.92	74.73	$2.8e^{-3*}$	40.74	43.47	$7.5e^{-5*}$
20	71	51	66.61	67.74	$2.5e^{-3*}$	76.42	77.09	0.019*	44.34	46.28	$2e^{-4*}$
40	99	75	69.66	71.23	$1e^{-6*}$	77.56	78.23	$2.7e^{-3*}$	48.63	52.00	$6.7e^{-5*}$
60	120	90	71.48	71.79	0.025*	77.26	77.50	0.045*	54.93	55.16	0.583

Table 6 Performance comparison between the two methods using Japanese data set (mean values are in percent and significance is at 5 % level)

TrRate (%)	n_s	n_p	Accuracy			Sensitivity			Specificity		
			FNN	SOM+FNN	p value	FNN	SOM+FNN	p value	FNN	SOM+FNN	p value
5	31	17	71.35	74.47	$1.6e^{-4}$*	74.37	77.25	$7.5e^{-4}$*	69.96	72.74	$1.6e^{-4}$*
10	40	24	76.18	78.29	$2.2e^{-4}$*	79.36	81.15	$6.7e^{-4}$*	73.73	75.99	$1.2e^{-3}$*
20	54	37	80.46	81.55	0.036*	83.7	83.92	0.704	78.22	79.51	0.02*
40	81	56	83.86	84.09	0.126	86.33	86.54	0.472	81.48	81.7	0.127
60	102	75	84.69	84.76	0.608	87.69	87.71	0.932	81.78	81.88	0.455

Table 7 Performance comparison between the two methods using Diane data set (mean values are in percent and significance is at 5 % level)

TrRate (%)	n_s	n_p	Accuracy			Sensitivity			Specificity		
			FNN	SOM+FNN	p value	FNN	SOM+FNN	p value	FNN	SOM+FNN	p value
5	41	28	78.24	80.8	$2.1e^{-4}$*	78.57	80.28	$1.1e^{-3}$*	80.85	82.78	0.03*
10	55	44	82.11	83.75	$9.6e^{-5}$*	80.76	82.22	$7.1e^{-4}$*	85.46	86.39	0.1549
20	84	72	87.05	87.53	0.021*	85.74	89.08	0.041*	89.17	85.59	0.27
40	112	101	89.00	89.32	0.005*	87.95	88.33	0.014*	90.45	90.59	0.499
60	136	129	90.27	90.56	0.0005*	88.99	89.26	0.044*	91.86	92.07	0.299

initializations of FNN. That is to say, the experiment was conducted a total of 100 runs for each value of *TrRate*. The final performance is obtained by calculating the average results on distinct trials.

The default configuration is used to create a SOM; that is to say, the topology lattice is 'hexa' and shape is 'sheet.' The number of map neurons is determined by a heuristic formula of munits $= 5\sqrt{N}$ where N is the number of training instances. Afterward, the ratio between sidelengths of the map grid is set to the square root of the ratio between the two biggest eigenvalues of the training data. Thus the sidelengths are set so that their product is as close to the desired number of map units as possible [16].

A FNN of 10 hidden nodes and one output node is constructed in the experiments. The number of input nodes

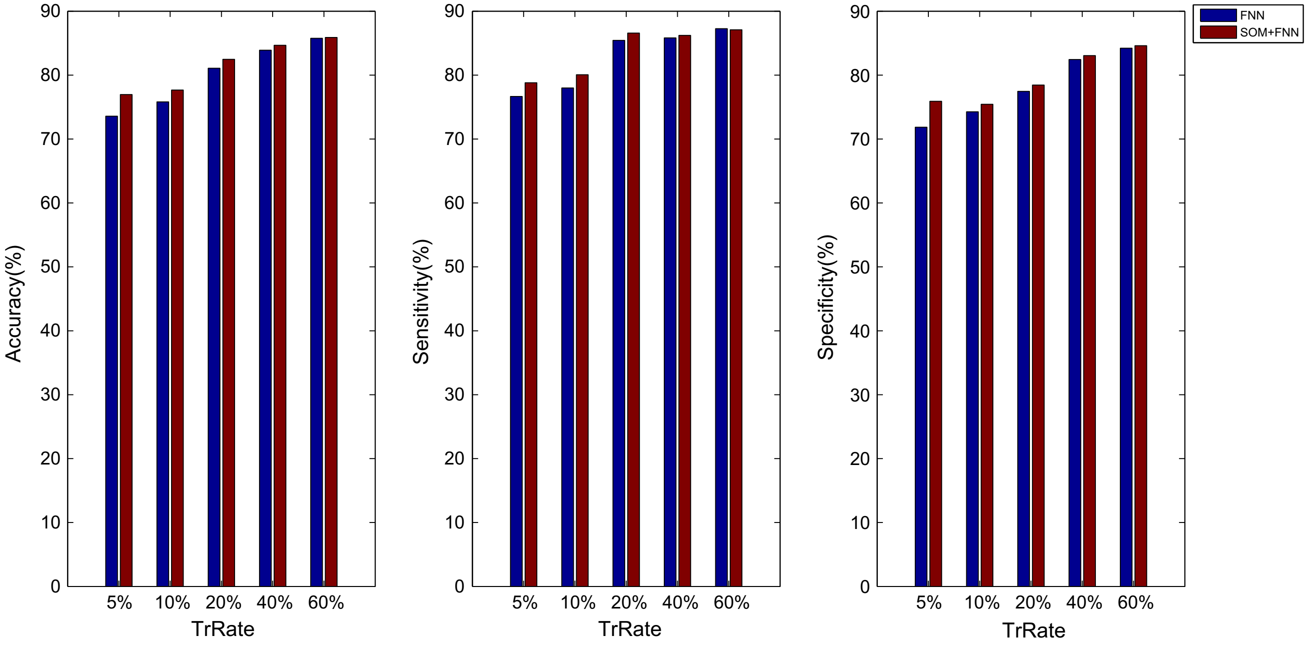

Fig. 4 Performance comparison of Australian data set

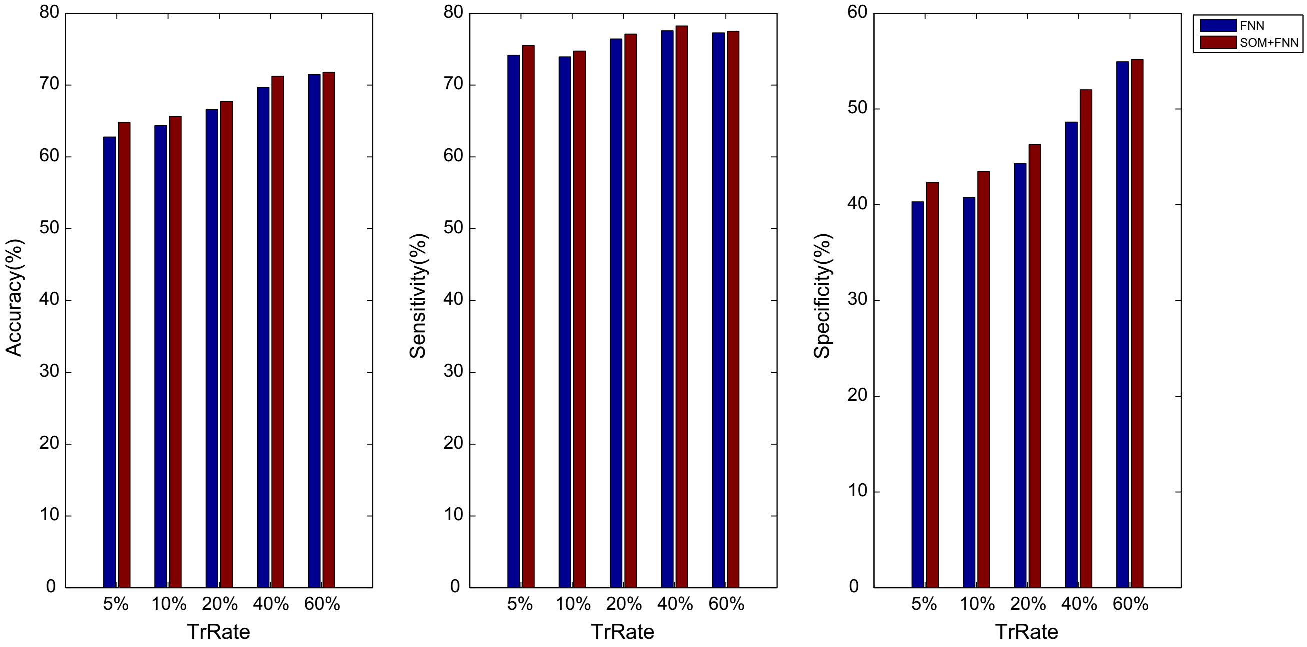

Fig. 5 Performance comparison of German data set

is determined by the examined data set. Other parameters are set by the default values of the MATLAB function, e.g., the default transfer function is 'transig' for hidden layer and 'purelin' for output layer. During the learning of FNN, each pair of input–output values is fed into the network for a number of epochs until the network learns the relationship between the input and output. Levenberg–Marquardt backpropagation numerical optimization algorithm is used to optimize the performance function. The output layer of FNN produces a value between 0 and 1, so with a threshold 0.5 we can get a binary class in the following way:

- If the output >0.5, assign the label 'bad credit' to the input;
- Otherwise, assign the label 'good credit' to the input.

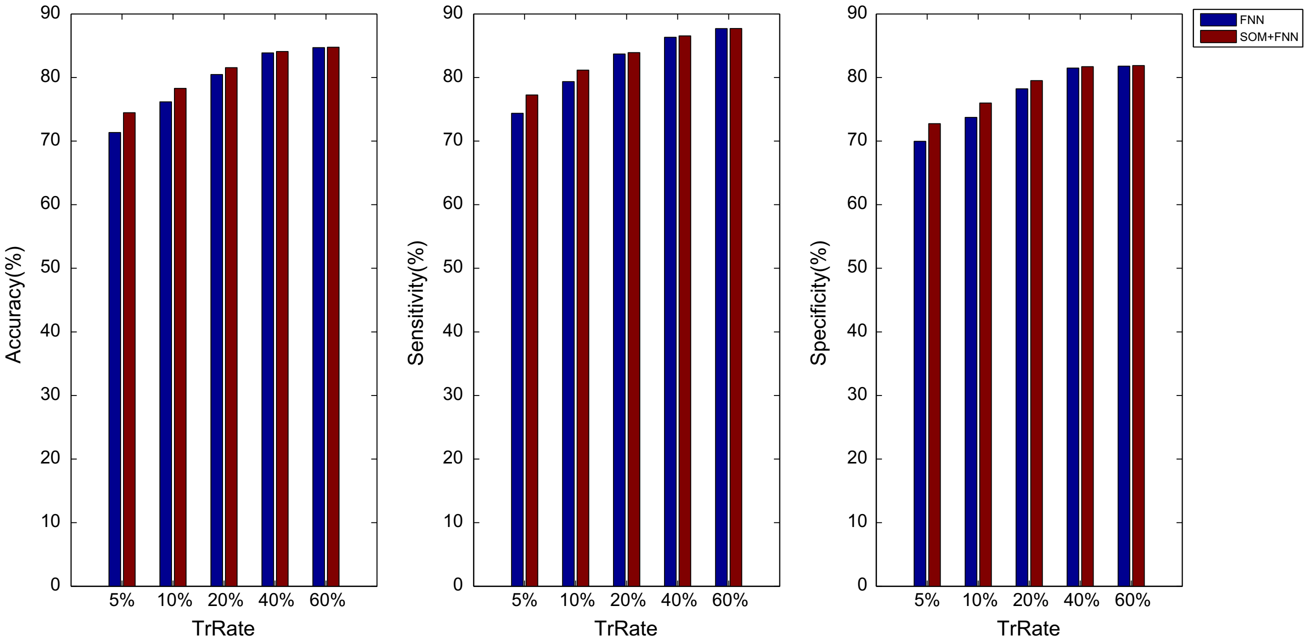

Fig. 6 Performance comparison of Japanese data set

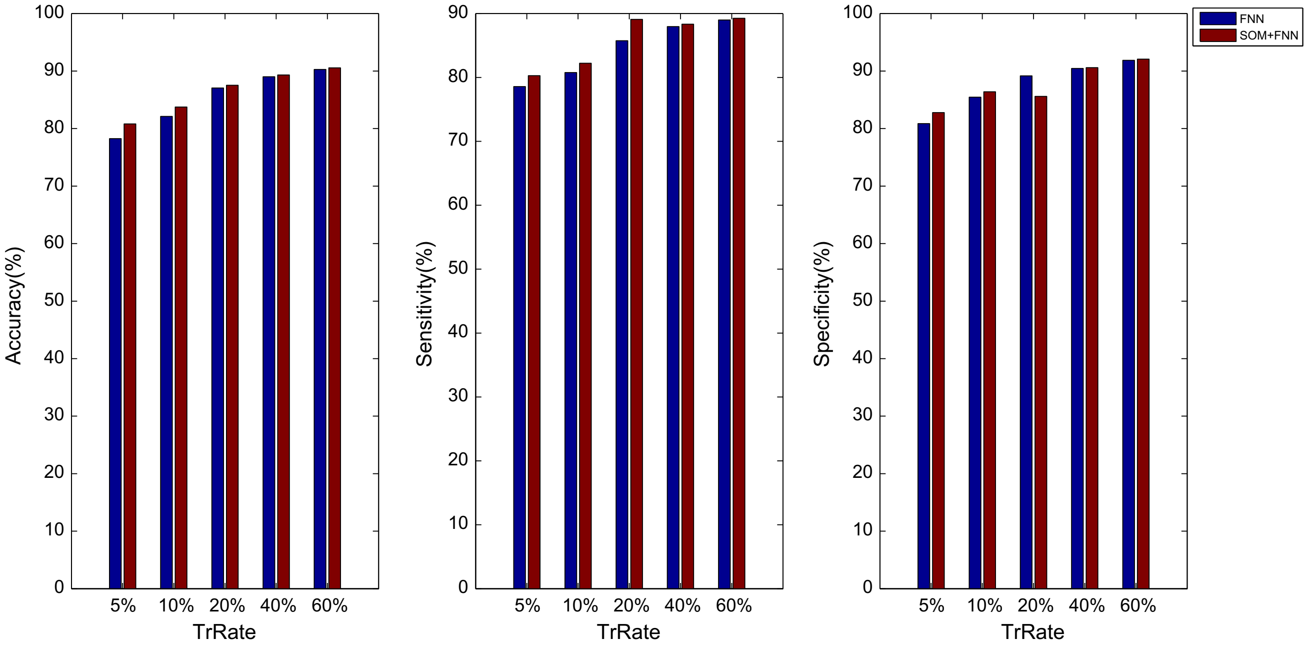

Fig. 7 Performance comparison of French data set

3.3 Performance evaluation

In order to evaluate a binary decision task, we define a contingency matrix representing the outcome of the classification, as given in Table 3. In this table, fn (false negative error) denotes the misclassification errors of bad credit credits as good credits, fp (false positive error) denotes the misclassification errors of good credits as bad credits, tp (true positive) denotes the correct predictions of positive samples, and tn (true negative) denotes the correction predictions of negative samples. Based on the contingency matrix, three performance metrics are defined:

Accuracy $= \frac{\mathrm{tp}+\mathrm{tn}}{\mathrm{tp}+\mathrm{fn}+\mathrm{tn}+\mathrm{fp}}$ denotes the proportion of correct predictions out of the total samples;

Sensitivity $= \frac{\mathrm{tp}}{\mathrm{tp}+\mathrm{fn}}$ denotes the fraction of true positives that are actually positive;

Specificity $= \frac{\mathrm{tn}}{\mathrm{tn}+\mathrm{fp}}$ denotes the fraction of true negatives that are actually negative.

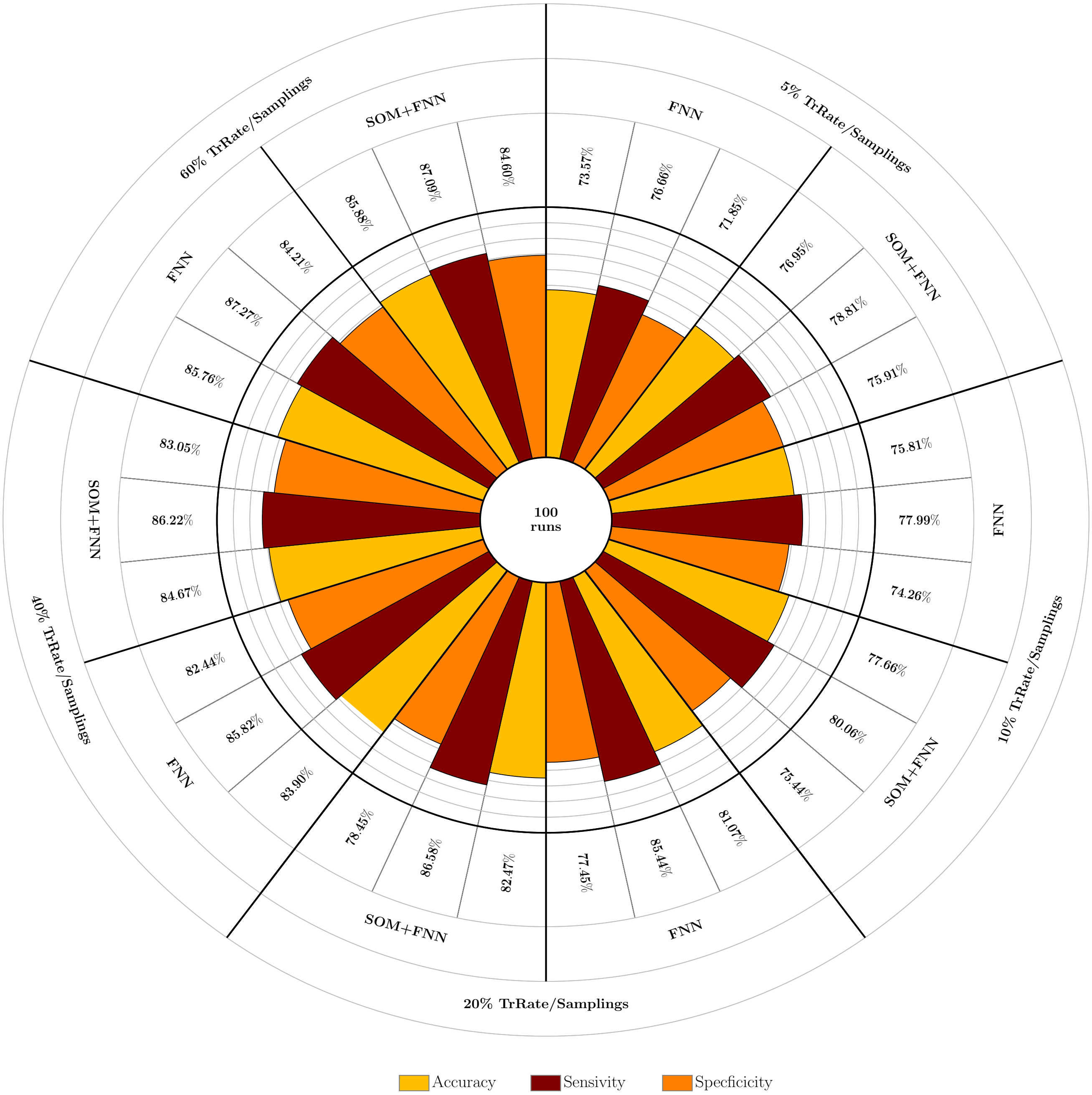

Fig. 8 Accuracy, sensitivity, and specificity of Australian data set

4 Results and discussion

In this experimental work, we carry out an empirical study of the proposed hybrid classification method (SOM+FNN) for credit risk scoring with comparison of the baseline FNN. In the former, the training instances along with the prototypes found by SOM are input to FNN, while in the latter only the training instances are used as input to FNN.

4.1 Performance comparison

Table 4 shows the results of Australian data set in terms of accuracy, sensitivity, and specificity by varying the percent of training samples from 5 to 60 %. In this table, n_s denotes the average number of SOM neurons across all trials and n_p denotes the average number of selected prototypes. A t test is employed to evaluate the statistical significance of observed difference in means of results. If the p value is

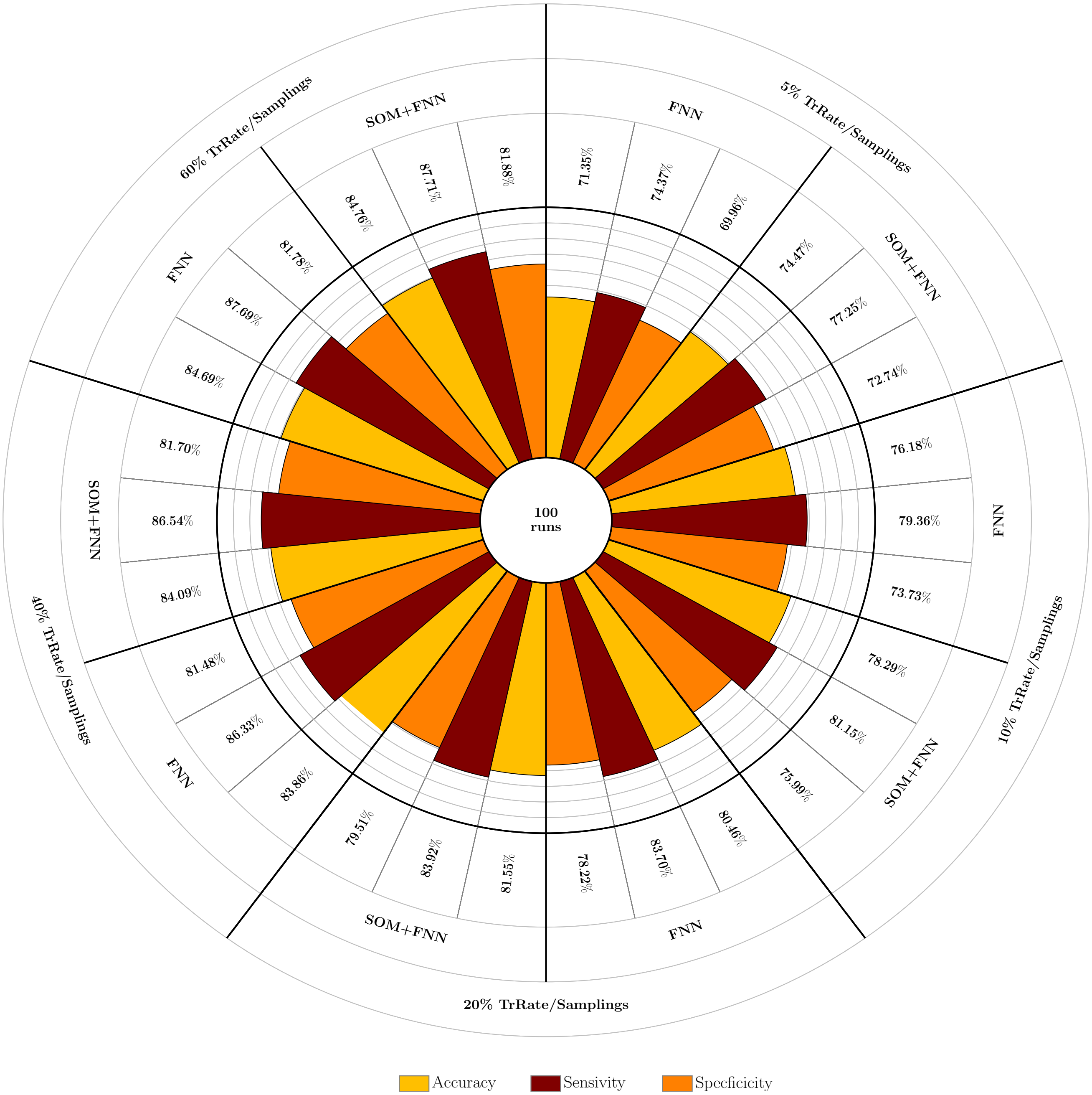

Fig. 9 Accuracy, sensitivity, and specificity of Japanese data set

less than 0.05 (marked by * in the table), there is significant difference between the two methods. As was shown, the hybrid classification method always produces better performance in all metrics than the stand-alone FNN although the benefit is not significant in some cases. Particulary the superiority tends to be more significant (p value <0.05) when the size of training data decreases and the classification becomes more difficult. This means the prototypes found by SOM are good supplement to the training data for improving the discriminant ability of FNN. The experimental results of the other three data sets are given in Tables 5, 6, and 7, respectively, also demonstrating the benefit of the hybridization approach.

The performance comparison between the two algorithms in terms of accuracy, sensitivity, and specificity is shown Figs. 4, 5, 6, and 7, respectively. As was disclosed, for both algorithms the classification performance tends to be better when using more samples for training, on the other hand, the hybrid classifier (SOM+FNN) always produces better results in three metrics than the stand-alone FNN, particularly in the case of small-scaled training data set.

We have generated a cobweb graph for each data set that allows to compare the performance of the two methods. Compared to the previous graphs, cobweb provides a compact representation which includes all the parameter configurations and metrics in one meaningful graphical visualization. For the lack of space, we only show the cobweb graph of Australian and Japanese data sets in Figs. 8 and 9. The performance results are presented in a visual way in terms of accuracy, sensitivity, and specificity, obtained, respectively, for Australian and Japanese data sets. Note that, in general, higher value of *TrRate* corresponds to better results as a more accurate predictive model can be constructed based on more training samples. Moreover under the same configuration, SOM+FNN algorithm always achieves the better performance than the stand-alone FNN algorithm.

SOM is a useful multivariate visualization method able to display the information hidden in multidimensional data on a two-dimensional map. Figure 10, taken as an example, shows the visualization of SOM output using a subset of Japanese data set with 10 % instances as the training data. A SOM of [8 × 5] grid size is used to train the 67 samples, which are then projected to the winner neuron via the best-matching manner. Various information is shown in the place of each map neuron, including (a) the number of hits taken the underling neuron as the winner; (b) the label by majority vote method; (c) the class distribution in pie chart of Voronoi set; (d) the class histogram of Voronoi set. As is shown, the map neurons separate the training instances into a number of disjoint clusters in a good way. The prototype is the reference vector of map neuron, representing the center of the corresponding clusters. After discarding the 10 neurons with empty Voronoi set and 3 neurons whose Voronoi set has uniform class distribution (e.g., the number of positive and negative instances is the same), we get 27 prototypes which will be used as input to FNN.

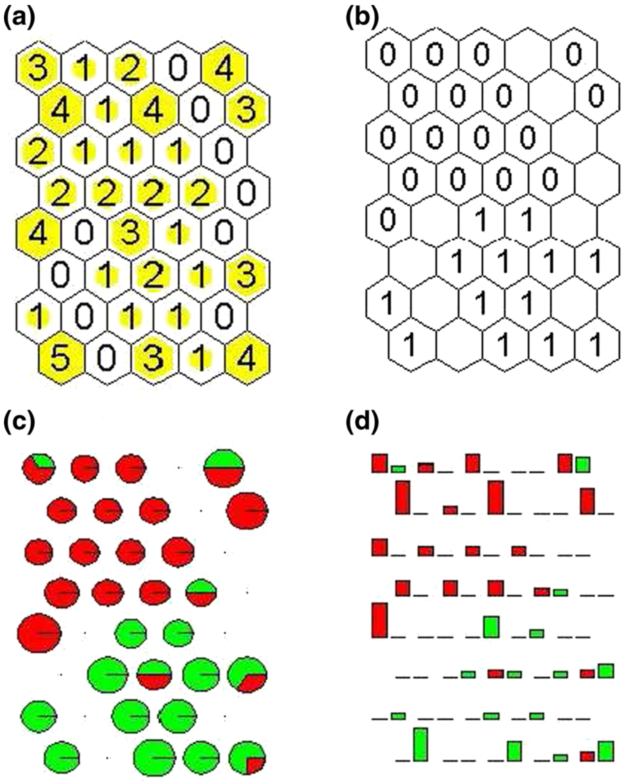

Fig. 10 SOM visualization of Japanese data set: **a** number of hits; **b** label of neurons; **c** class distribution of each neuron; **d** class histogram of each neuron

4.2 Parameter discussion

There are some parameters of the proposed hybrid algorithm, such as the number of neurons, neighborhood, learning rate, and initialization for SOM, and the number of neurons, number of training cycles, and parameters of learning algorithms for FNN. In the following experiments, we take into consideration some important parameters: TrRate, #units, and #nodes. The notation of these parameters is given in Table 8. For each training data set, a FNN

Table 8 Notation: parameters of SOM+FNN

Parameters	Meaning	Values
TrRate	Percentage of samples as training data	5 %, 10 %, 20 %, 40 %, 60 %
#Units	Number of units for SOM	[10, 100]
#Nodes	Number of hidden neurons for FNN	[5, 30]

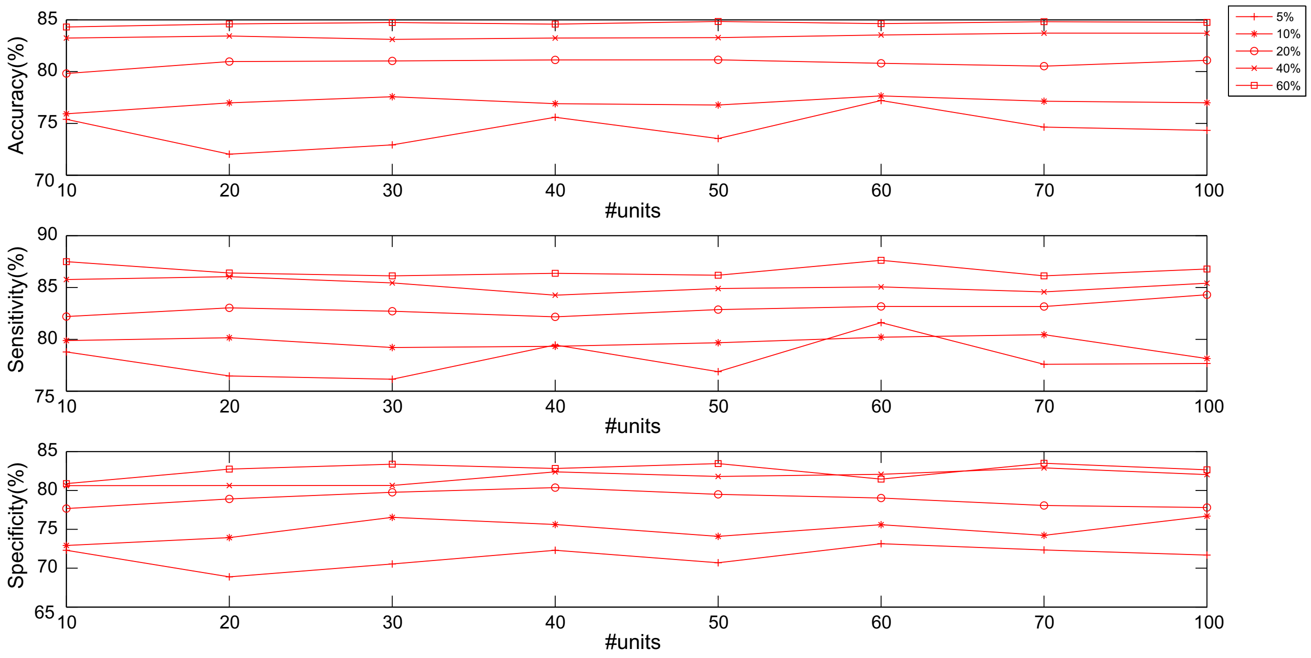

Fig. 11 Performance of SOM+FNN using Australian data set

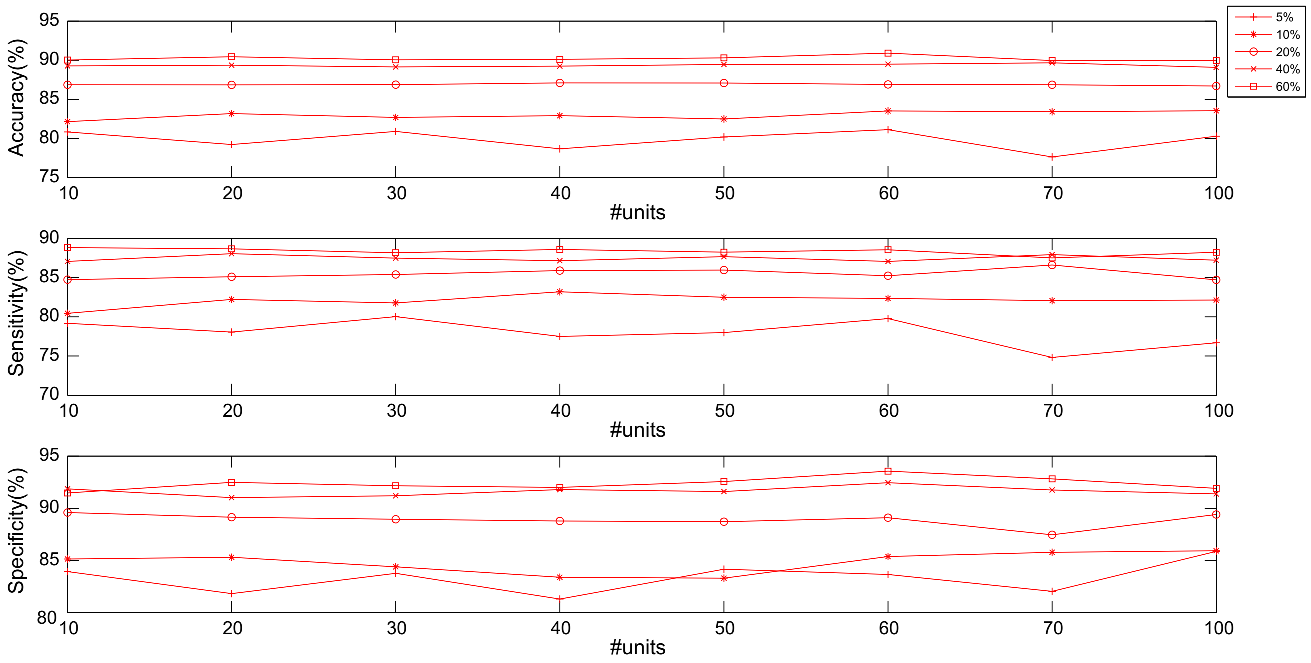

Fig. 12 Performance of SOM+FNN using Diane data set

is created with a random number between 5 and 30 of hidden nodes and then combined with a SOM by varying the number of units from 10 to 100. As expected, in all cases the hybrid algorithm receives better classification results than the stand-alone FNN, demonstrating the superiority of SOM+FNN regardless of the parameters. Additionally, the number of SOM neurons determines the cluster prototypes found by SOM and thus impacts the performance of hybrid classification. Figures 11, 12, and 13 show the performance of SOM+FNN on Australian, Diane, and Japanese data set, respectively, by varying the number of map neurons from 10 to 100 with a randomly selected hidden node of FNN. The size of training data set is varying from 5 to 60 % of the original data set. It was observed the value of #units has relatively more impact on the discriminative ability of FNN in the case of small-

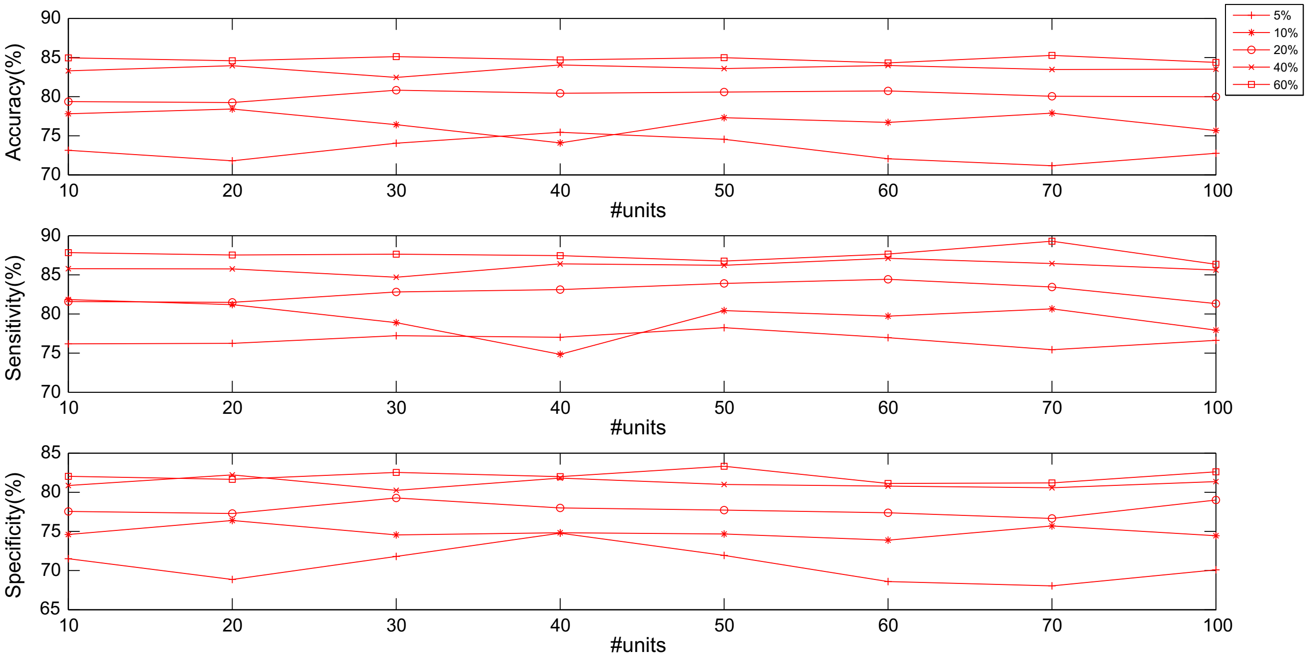

Fig. 13 Performance of SOM+FNN using Japanese data set

scaled training data set. When more training samples are used, the importance of cluster prototypes tends to decrease. On the other hand, increasing the number of map neurons does not always contribute to the discriminative ability of FNN. Too small number of neurons is difficult to separate the instances in a good way, while too big number of neurons results in many empty and small clusters that are probably discarded in prototype selection phase. A well-defined SOM is able to improve the functioning of the subsequent FNN.

5 Conclusion

Nowadays the combination of supervised and unsupervised methods receives increasing attention in credit scoring tasks to improve the classification accuracy. SOMs have been widely used in data clustering as a valuable tool due to the unique properties of data summarization and visualization. In this paper, we present a hybrid approach to transferring the learned knowledge from SOM to FNN. Thus, the FNN inputs are the prototypes found by SOM along with the original training data. Four real-world data sets are used in the experiments for credit approval problems. From the experimental results, some conclusions can be obtained. The hybridization technique actually contributes to boosting the performance of FNN for credit scoring; however, the benefit is closely related to the number of training instances. The benefit becomes more significant when the training data are in small size. This gives some insights on how to construct weakly supervised learning models [26] in the sense that with limited amount of labeled data the transfer of learning is possible and the performance improves. Moveover the superiority of the hybrid classifier remains stable when varying the parameters of the algorithm. A complete and unique graphical visualization technique is shown to better outline the trade-off between distinct metrics and attained performance.

In the future work, some limitations will be addressed as the research directions. Firstly, the parameters of both SOM and FNN will be deeply investigated and optimized to advance the classification performance. Additionally, the presented study is not limited to SOM and FNN to build the hybrid classification model. The hybridization of other unsupervised learning methods and supervised methods might result into valuable tools for credit scoring.

Acknowledgments This work was supported by national funds through the Portuguese Foundation for Science and Technology (FCT), National Natural Science Foundation of China (Contact No. 11601129), and the European Regional Development Fund (FEDER) through COMPETE 2020—Operational Program for Competitiveness and Internationalization (POCI).

References

1. Atiya AF (2001) Bankruptcy prediction for credit risk using neural networks: a survey and new results. IEEE Trans Neural Netw 12(4):929–935
2. Bapat V, Nagale A (2014) Comparison of bankruptcy prediction models: evidence from india. Account Finance Res 3(4):91–98

3. Burda A, Pancerz K (2014) Clustering and visualization of bankruptcy patterns using the self-organizing maps. Barometr Regionalny 12(3):133–138
4. Chen N, Ribeiro B, Vieira A, Chen A (2013) Clustering and visualization of bankruptcy trajectory using self-organizing map. Expert Syst Appl 40(1):385–393
5. Davidian D (1995) Feed-forward neural network. US, US 5438646 A
6. Dharamkar B, Singh RR (2014) A review of cyber attack classification technique based on data mining and neural network approach. Int J Comput Trends Technol 7(2):100–105
7. Dietterich TG, Bakiri G (1995) Solving multiclass learning problems via error-correcting output codes. J Artif Intell Res 2(2):263–286
8. Dreiseitl S, Ohno-Machado L (2002) Logistic regression and artificial neural network classification models: a methodology review. J Biomed Inform 35(5–6):352–359
9. Greene WH (1992) A statistical model for credit scoring. Technical Report EC-92-29, NYU Working Paper
10. Harris T (2015) Credit scoring using the clustered support vector machine. Expert Syst Appl 42(2):741–750
11. Huang Z, Chen H, Hsu CJ, Chen WH, Wu S (2010) Credit rating analysis with support vector machines and neural networks: a market comparative study. Decis Support Syst 37(4):543–558
12. Japkowicz N (2002) Supervised learning with unsupervised output separation. In: International conference on artificial intelligence and soft computing, pp 321–325
13. Kalinic H, Mihanovic H, Cosoli S, Vilibic I (2015) Sensitivity of self-organizing map surface current patterns to the use of radial vs. cartesian input vectors measured by high-frequency radars. Comput Geosci 84:29–36
14. Karem F, Dhibi M, Martin A (2012) Combination of supervised and unsupervised classification using the theory of belief functions. Springer, Berlin
15. Khashman A (2010) Neural networks for credit risk evaluation: investigation of different neural models and learning schemes. Expert Syst Appl 37(9):6233–6239
16. Kohonen T SOM toolbox. Technical report, Laboratory of computer and information sciences and Neural networks research center, Helsinki University of Technology. http://www.cis.hut.fi/projects/somtoolbox/
17. Kohonen T (1982) Self-organized formation of topologically correct feature maps. Biol Cybern 43:59–69
18. Lichman M (2013) UCI machine learning repository. Technical report, University of California, Irvine, School of Information and Computer Sciences. http://archive.ics.uci.edu/ml
19. Lin WY, Hu YH, Fong TC (2012) Machine learning in financial crisis prediction: a survey. IEEE Trans Syst Man Cybern Part C Appl Rev 42(4):421–436
20. Mahmon NA, Ya'acob N (2014) A review on classification of satellite image using artificial neural network (ANN). In: 2014 IEEE 5th control and system graduate research colloquium (ICSGRC), pp 153–157
21. Pandey M, Pandey AK, Mishra A, Tripathi BD (2015) Application of chemometric analysis and self organizing map-artificial neural network as source receptor modeling for metal speciation in river sediment. Environ Pollut 204:64–73
22. Silva C, Ribeiro B (2016) Multiclass ensemble of one-against-all SVM classifier. In: International symposium on neural networks (ISNN), LNCS, Russia
23. Merkevicius E, Garsva G, Simutis R (2004) Forecasting of credit classes with the self-organizing maps. Inf Technol Control 33(4):61–66
24. Tamura M, Kitamura N, Watarai S (2016) Artificial neural networks in business: two decades of research. Appl Soft Comput 38:788–804
25. Thomas LC (2000) A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. Int J Forecast 16(2):149–172
26. Torresani L (2014) Weakly supervised learning. Springer, New York
27. Tsai CF, Wu JW (2008) Using neural network ensembles for bankruptcy prediction and credit scoring. Expert Syst Appl 34(4):2639–2649
28. Verikas A, Kalsyte Z, Bacauskiene M, Gelzinis A (2010) Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: a survey. Soft Comput A Fusion Found Methodol Appl 14(9):995–1010
29. Vieira JC, Neves A (2006) Improving bankruptcy prediction with hidden layer learning vector quantization. Eur Account Rev 15(15):253–271
30. Yap BW, Ong SH (2011) Using data mining to improve assessment of credit worthiness via credit scoring models. Expert Syst Appl 38(10):13274–13283
31. Yu L, Wang S, Lai KK (2008) Credit risk assessment with a multistage neural network ensemble learning approach. Expert Syst Appl 34(2):1434–1444
32. Zeng X, Li Y, He R, Yin Y (2015) Clustering of loop current patterns based on the satellite-observed sea surface height and self-organizing map. Remote Sens Lett 6(1):11–19