Classifiers consensus system approach for credit scoring

Maher Ala'raj*, Maysam F. Abbod

Department of Electronic and Computer Engineering, Brunel University London, Uxbridge UB8 3PH, United Kingdom

ABSTRACT

Banks take great care when dealing with customer loans to avoid any improper decisions that can lead to loss of opportunity or financial losses. Regarding this, researchers have developed complex credit scoring models using statistical and artificial intelligence (AI) techniques to help banks and financial institutions to support their financial decisions. Various models, from easy to advanced approaches, have been developed in this domain. However, during the last few years there has been marked attention towards development of ensemble or multiple classifier systems, which have proved their ability to be more accurate than single classifier models. However, among the multiple classifier systems models developed in the literature, there has been little consideration given to: 1) combining classifiers of different algorithms (as most have focused on building classifiers of the same algorithm); or 2) exploring different classifier output combination techniques other than the traditional ones, such as majority voting and weighted average. In this paper, the aim is to present a new combination approach based on classifier consensus to combine multiple classifier systems (MCS) of different classification algorithms. Specifically, six of the main well-known base classifiers in this domain are used, namely, logistic regression (LR), neural networks (NN), support vector machines (SVM), random forests (RF), decision trees (DT) and naïve Bayes (NB). Two benchmark classifiers are considered as a reference point for comparison with the proposed method and the other classifiers. These are used in combination with LR, which is still considered the industry-standard model for credit scoring models, and multivariate adaptive regression splines (MARS), a widely adopted technique in credit scoring studies. The experimental results, analysis and statistical tests demonstrate the ability of the proposed combination method to improve prediction performance against all base classifiers, namely, LR, MARS and seven traditional combination methods, in terms of average accuracy, area under the curve (AUC), the H-measure and Brier score (BS). The model was validated over five real-world credit scoring datasets.

Keywords: credit scoring; consensus approach; multiple classifier systems; classifier ensembles; classification

Introduction

1.1. Background

Credit granting to lenders is considered a key business activity that generates profits for banks, financial institutions and shareholders, as well as contributing to the community. However, it can also be a great source of risk. The recent financial crises resulted in huge losses globally and, hence, increased the attention paid by banks and financial institutions to credit risk models. That is, as a result of the crises, banks are now cognisant of the need to adopt rigorous credit evaluation models in their systems when granting a loan to an individual client or a company. The problem associated with credit scoring is that of categorizing potential borrowers into either good or bad. Models are developed to help banks to decide whether to grant a loan to a new borrower or not using their data characteristics (Hand and Henley, 1997). The area of credit scoring has become a widely researched topic by scholars and the financial industry (Kumar and Ravi, 2007; Lin et al. 2012) since the seminal work of Altman in 1968 (Altman, 1968). Subsequently, many models were proposed and developed using statistical approaches, such as logistic regression (LR) and linear discriminate analysis (LDA) (Desai et al., 1996; Baesens et al., 2003). Recently, the Basel Committee on Banking Supervision (Lessmann et al., 2015) requested that all banks and financial institutions to have rigorous and complex credit scoring systems in order to help them improve their credit risk levels and capital allocation.

Despite developments in technology, LR is still the industry-standard baseline model used for building credit scoring models (Lessmann et al., 2015); many studies have demonstrated that artificial intelligence (AI) techniques, such as neural networks (NN), support vector machines (SVM), decision trees (DT), random forests (RF) and naïve Bayes (NB), can be substitutes for statistical approaches in building credit scoring models (Atiya, 2000; Van Gestel et al., 2003; Wang et al., 2012; Verikas et al., 2011; Hsieh and Hung, 2010; Zhou, 2013).

In practice, real historical datasets are used in order to develop credit-scoring models; these datasets might differ in size, nature, and the information or characteristics it holds, whilst individual classifiers might not be able to capture different relationships of these datasets characteristics. As a result, researchers have employed hybrid modelling

* *Corresponding author*. Tel.: +447466925096.
E-mail address: maher.ala'raj@brunel.ac.uk

techniques that can exploit the strength and compensate weaknesses of different classifiers in learning the relationships between data (Sánchez-Lasheras et al., 2012; Zhou et al., 2015; Liang et al., 2015). From hybrid-modelling, researchers have inspired the ensemble modelling, which gives classifiers the opportunity express their ability to learn data on different parts of data and feature space.

However, the research trend has been actively moving towards using single AI techniques in multiple classifier systems (MCS) or ensemble models (Wang et al., 2011; Sun et al., 2014). According to Tsai (2014), the idea of MCS is based on the combination of a pool of diversified classifiers, such that their combination achieves higher performance than single classifiers since each complements the other classifiers' errors. However, in the literature on credit scoring most of the classifier combination techniques take the form of homogenous and heterogeneous classifier ensembles, where the former combine the classifiers of the same algorithm, while the latter combine classifiers of different algorithms (Lessmann et al., 2015; Tsai, 2014). As Nanni and Lumini (2009) pointed out, an MCS is a set of classifiers each of whose decisions are combined using some approach.

1.2. Motivations

Recent studies have shown that MCS or ensemble models perform better than single AI classifiers in credit scoring (Lessmann et al., 2015; Wang et al., 2011; Nanni and Lumini, 2009; Yu et al., 2009). Most of the related work in ensemble studies in the domain of credit scoring have been focused on homogenous ensemble classifiers via simple combination rules and basic fusion methods, such as majority voting, weighted average, weighted vote, reliability-based methods, stacking and fuzzy rules (Wang et al., 2012; Tsai, 2014; Yu et al., 2009; Tsai and Wu, 2008; West et al., 2005; Yu et al., 2008). A few researchers have employed heterogeneous ensemble classifiers in their studies, but still with the above-mentioned combination rules (Lessmann et al. 2015; Wang et al., 2012; Hsieh and Hung, 2010; Tsai, 2014). In ensemble learning, all classifiers are trained independently to produce their decisions, to be combined via a heuristic algorithm to produce one final decision (Zang et al., 2014; Rokach, 2010).

Consequently, the main aim of this paper is improve accuracy by exploring a new combination method in the field of credit scoring by developing a new combination rule whereby the ensemble classifiers can work and collaborate as a group or a team in which their decisions are shared between classifiers. The classifier consensus approach is where classifier ensembles work as a team to interact and cooperate to solve the same problem. Generally speaking, the key idea behind the consensus approach is to build an ensemble of classifiers that can be viewed as a collaborative society. The classifiers share their initial decisions and become involved in a discourse process, until all classifiers come to an agreement on a final optimal decision, which represents the view of all ensemble members. Through this combination process, a more effective and efficient decision-making process can be obtained.

Experimentally, the classifier consensus approach involves combining the decisions of five well-known classifiers that are used as base models, namely, NN, SVM, DT, RF and NB. The predictive performance of the test set is evaluated against four performance measures: average accuracy (ACC), AUC, H-measure and Brier score (BS). Furthermore, the new approach is compared with the other seven combination rules found in the credit scoring literature, in addition to the individual base classifiers. To validate the model, the experiments are carried out over five real-world financial datasets. Finally, the model results are benchmarked to the industry-standard logistic (LR) and multivariate adaptive regression splines (MARS). It is worth noting that the combination rule developed in this paper is based on measurement-level predictions (Suen and Lam, 2000).

The organization of the paper is as follows: Section 2 provides an introduction to multiple classifier systems, with comparison and analysis of the related literature in terms of the datasets, base models, multiple classifier systems, combination methods and evaluation performance measures used. Section 3 explains the classifier consensus approach. Section 4 describes the experimental set-up carried out for the current study. Section 5 presents the experimental results and analysis. Finally, in Section 6 conclusions are drawn and future research trends discussed.

2. Literature review

2.1. Multiple classifier systems (MCS)

In the pattern recognition field and machine learning communities, multiple classifier systems or ensemble models have become an alternative to single classifiers, due to their potential in improving predictive accuracy. These have widely attracted the attention of researchers in the field of credit scoring over the last decade, with many having combined multiple classifier systems in different ways in order to achieve high-prediction-performance classifiers (Tsai, 2014; Nanni and Lumini, 2009). The rationale behind MCS is to combine several classifier predictions in order to achieve classification accuracy better than that of any single base classifier (Tsai and Wu, 2008). Moreover, it can be more difficult

to optimize the design of a single classifier than that of a combination of simple classifiers (Tsai and Wu, 2008; Zhang and Duin, 2010).

Practically, building a multiple classifier systems involves two main steps: 1) classifier generation and 2) classifier fusion or combination (Zhang and Duin, 2010). The first step includes creating base classifiers by training them on different training sets or feature sets. That is, different classifiers are trained on diverse segments of the data, so each trained classifier is generalized in different ways (Tsai and Wu, 2008; Zhang and Duin, 2010). The most popular approaches for modifying training data are bagging and boosting (Wang et al., 2011; West et al., 2005; Marques et al., 2012a). Multiple classifier systems can be built of the same type of classification algorithm, which are termed as homogenous. Alternatively, different classification algorithms can be applied to make the ensemble, with the idea being that the different classifiers have different views on the same data and can complement each other (Lessmann et al., 2015; Zhang and Duin, 2010).

After all classifiers give their decisions, they are pooled in order to combine and fuse them using some rule or method. The most popular fusion methods explored in the field of credit scoring are majority voting, weighted average, and weighted voting, mean, maximum, minimum, and product rules. These fusion methods can be the best option for combining multiple classifiers owing to their simplicity and good performance (Zhang and Duin, 2010). Figure 1 shows the structure of multiple classifier systems. It is worth mentioning that that the fusion methods are widely cited in pattern recognition and machine learning literature (Zhang and Duin, 2015; Canuto et al., 2007). Our proposed combination method, as well as the aforementioned ones, is considered as fixed combiners in the current work.

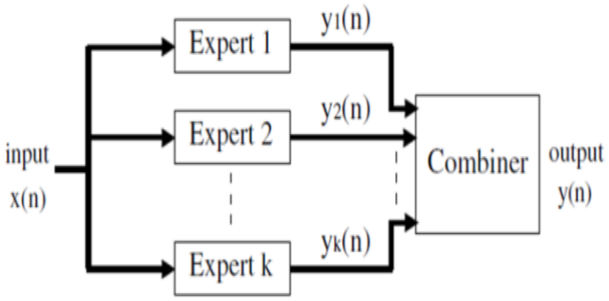

Fig. 1. Structure of multiple classifier systems

2.2. Related work

A comparative analysis of recent literature on multiple classifier systems and ensemble models in credit scoring is presented in this section. For this reason, a summary of studies in the literature on classifier ensembles and multiple classifier systems used for credit scoring from 2005 up until 2015 is given in Table 1. The comparison is made in terms of number of datasets used, whether the developed classifier ensembles are homogeneous or heterogeneous, the combination rules used to fuse the classifier output, the performance measurements employed and whether statistical tests of significance were carried out or not. As can be seen, more than two thirds of the studies involved used from one to three datasets to validate their models, whereas the rest used from five to eight. According to Lessmann et al. (2015a) and Finlay (2011), using data from different companies provides robustness to the model under different environmental conditions. Moreover, real-world datasets used in classifier comparisons should be similar so as to ensure external validity of the empirical results. Regarding the ensemble classifiers, most researchers have opted for homogenous ones, with just two having developed heterogeneous classifiers and three having included both in the same study. In respect of the combination rules used to fuse data, majority vote was the most popular due to its simplicity; weighted average was also used, but less often. Reliability-based methods were only used in four studies, and stacking, which is considered a trainable combiner, was employed in two.

The majority of the studies developed multiple classifier systems whereby each classifier gave independent decisions and then combined them into one single output without any collaboration or coordination between the classifiers through the learning process. Conversely, in a study by Yu et al. (2009), heterogeneous ensembles were developed and combined with those using fuzzy rules based on group decision-making that involves classifiers working in a group to reach a consensus on the final output. It was this work that inspired the development of a combination rule based on a consensus between classifier ensembles in order to achieve better performance than that of any individual classifier in the ensemble.

With regard to the performance measurements used, most have been commonly based on average accuracy, Type I and Type II errors. AUC appeared in four studies and in one, by Lessmann et al. (2015), new measures for credit scoring

were used, including Brier score, Kolmogorov–Smirnov, a partial Gini index and the H-measure. Finally, more than half of the studies employed statistical tests to determine whether the performances of their proposed models were statistically significant.

Table 1
Comparisons of related literature

Year	Study	# of data sets	Classifier ensembles		Combination rule	Performance measure	Sig. test
			Homogenous	Heterogeneous			
2005	West et al. (2005)	3	x		Majority vote, weighted average	ACC[1]	x
2006	Lai et al. (2006)	1	x		Majority vote, reliability-based[2]	ACC, Type I & II errors	
2008	Tsai and Wu (2008)	3	x		Majority vote	ACC, Type I & II errors	x
	Yu et al. (2008)	2	x		Majority vote, reliability-based	ACC, Type I & II errors	
2009	Nanni and Lumini (2009)	3	x		Sum rule	ACC, Type I & II errors, AUC[3]	x
	Yu et al. (2009)	3		x	Fuzzy GDM[4]	ACC, Type I & II errors, AUC	x
2010	Hsieh and Hung (2010)	1		x	Confidence-weighted average	ACC	
	Yu et al. (2010)	1	x		ALNN[5]	ACC, Type I & II errors, AUC	x
	Zhang et al. (2010)	2	x		Majority vote	ACC	
	Zhou et al. (2010)	2	x		Majority vote, reliability-based, weights based on samples	ACC, specificity, sensitivity, AUC	
2011	Wang et al. (2011)	3	x	x	Majority vote, weighted average, stacking	ACC, Type I & II errors	
	Finlay (2011)	2	x		Majority vote, weighted average, mean	Classification error rate	x
2012	Wang et al. (2012)	2	x		Majority vote	ACC, Type I & II errors	
	Marqués et al. (2012)	6	x		Majority vote	ACC, Type I & II errors	x
2014	Tsai (2014)	5	x	x	Majority vote, weighted vote	ACC, Type I & II errors	x
	Abellán and Mantas (2014)	3	x		Majority vote	AUC	x
2015	Lessmann et al. (2015)	8	x	x	Majority vote, weighted average, stacking	ACC, AUC, BS[6], KS[7], PG[8], H-measure	x
2016	Zhou et al. (2016)	1	x		Majority vote	ACC, CK[9]	x

[1] ACC: average accuracy. [2] Reliability-based: minimum, maximum, mean, median, product rules. [3] AUC: area under curve. [4] GDM: group decision-making. [5] ALNN: adaptive linear neural network. [6] BS: Brier score. [7] KS: Kolmogorov–Smirnov. [8] PG: partial Gini index. [9] CK: Cohen's Kappa.

3. Classifier consensus approach

The basic idea behind classifier decisions combination is that, when classifiers make a decision, one should not rely only on a single classifier decision, but, rather, require classifiers to participate in the decision-making process by combining or fusing their individual opinions or decisions. Therefore, the core problem that needs to be addressed when combining different classifiers is resolving conflicts between them. In other words, the problem is how to combine the results of different classifiers to obtain better results (Chitroub, 2010; Xu et al., 1992). In this section, a new combination method is introduced in the field of credit scoring based on classifier consensus, where those in the ensemble interact in a cooperative manner in order to reach an agreement on the final decision for each data sample.

Tan (1993) emphasized that agents working in partnership can significantly outperform those working independently. The idea of the consensus approach is not new as it has been investigated in many studies in different fields, such as statistics, remote sensing, geography, classification, web information retrieval and multi-sensory data (Tan, 1993; DeGroot, 1975; Benediktsson and Swain, 1992; Shaban et al., 2002; Basir and Shen, 1993). In this context, the general strategies adopted are those of DeGroot (1975) and Shaban et al. (2002), who proposed a framework that provides a comprehensive and practical set of guidelines on the underpinning constructs of consensus theory where interactions between classifiers are modelled when an agreement between them is needed. It is believed that their strategies can be useful when adopted for the credit scoring domain. Practically, consensus mimics the team-communication processes of a real group of experts, so that each individual expert can modify his/her own opinion according to the opinions of other

experts in the group. The final ranking of the consensus classifier is calculated as a common group decision after equilibrium is reached. When opinions are no longer changing, and in order to reach a consensus between classifiers on each input decision, a set of steps have to be processed. These are discussed below.

Step 1. *Calculating the rankings of all classifiers and building a decision profile*

Consider a group of N agents, denoted by the set $A = A_1.A_2.\ldots.A_N$. When receiving a data sample, A_i chooses an answer from a set of possible answers $\Gamma = (\gamma_1..\ldots\gamma_m)$. For each classifier, consider a ranking function R_i, which associates a nonnegative number for every possible answer from Γ_i. The result of the estimate function R_i is a value in the range of [0, 1] which shows the desirability of the corresponding answer. Prediction of the classifier may be found after finding R_i and applying a threshold to it.

$$\sum_{k=1}^{m} R_i(\gamma_k) = 1 \; \forall i \in \{1..N\} \tag{1}$$

Now, after calculating each classifier ranking, the decision profile can be represented in matrix form as

$$DP = \begin{bmatrix} R_1(e_1) & R_1(e_2) & R_1(e_3) & \ldots & R_1(e_n) \\ R_2(e_1) & R_2(e_2) & R_2(e_3) & \ldots & R_2(e_n) \\ R_3(e_1) & R_3(e_2) & R_3(e_3) & \ldots & R_3(e_n) \\ R_4(e_1) & R_4(e_2) & R_4(e_3) & \ldots & R_4(e_n) \\ R_5(e_1) & R_5(e_2) & R_5(e_3) & \ldots & R_5(e_n) \end{bmatrix} \tag{2}$$

Where n is the number of queries in the training/testing set, ei is the i-th input query and $R_j(e_i)$; j ∈1..5 is the j-th classifier ranking for the i-th input query. So, to evaluate the uncertainty between classifiers we need to process *n* columns of matrix *DP* for testing the set, input by input.

Therefore, our objective is to evaluate the common group ranking $R_G: \Gamma \rightarrow [0,1]$ to aggregate the expected rankings for all classifiers.

Step 2. *Calculating classifier uncertainty*

After building the decision profile for the classifier rankings, the next stage is about finding a function by which each classifier's uncertainty can be computed. The task here is to assign more weight to classifiers that are less certain about their decision, and vice versa. However, the weighting should reflect the contrast in classifiers' decisions. During this stage, uncertainty will be divided into two types: local and global.

Local uncertainty relates to the quality of the classifier's own decision, whereas global uncertainty emerges as the result of collaboration between classifiers taking place in the form of decision profile exchange. At this stage a classifier will be able to review its uncertainty level and modify it given its decision as well as the decisions of others. This shows how a classifier is able to improve its decision when other classifiers' decisions become available. Consequently, $R_i(\gamma_k)$ is the i-th classifier's ranking of answer γ_k, and $R_i(\gamma_k|\Gamma_j)$ is the i-th classifier's ranking of answer γ_k, if it knows the ranking vector of the j-th classifier.

Matrix U is evaluated using equations (3) and (4):

$$U_{ii} = -\sum_{k=1}^{M} R_i(\gamma_k)\log_2(R_i(\gamma_k)) \tag{3}$$

$$U_{ij} = -\sum_{k=1}^{M} R_i(\gamma_k|\Gamma_j)\log_2\big(R_i(\gamma_k|\Gamma_j)\big) \tag{4}$$

Now, knowing that equation (1) is fulfilled, equation (5) can be fulfilled:

$$\sum_{k=1}^{m} R_i(\gamma_k|\Gamma_j) = 1 \forall i \in \{1..N\} \tag{5}$$

in the case of two possible answers (classes): "0", good loan and "1", bad loan. Thus, equations (1) and (5) are converted into

$$R_i(0) + R_i(1) = 1. R_i(0|\Gamma_j) + R_i(1|\Gamma_j) = 1 \tag{6}$$

where, $R_i(1)$ is the i-th classifier ranking of answer "1" and $R_i(0)$ is the i-th classifier ranking of answer "0". Denote $R_i = R_i(1)$ and $R_i(\Gamma_j) = R_i(1|\Gamma_j)$, then $R_i(0) = 1 - R_i$ and $R_i(0|\Gamma_j) = 1 - R_i(\Gamma_j)$ and, hence, equations (3) and (4) are converted into:

$$U_{ii} = -R_i \log_2(R_i) - (1 - R_i)\log_2(1 - R_i) \tag{7}$$

$$U_{ij} = -R_i(\Gamma_j)\log_2(R_i(\Gamma_j)) - (1 - R_i(\Gamma_j))\log_2(1 - R_i(\Gamma_j)) \tag{8}$$

The uncertainty matrix can be presented as follows:

$$U = \begin{bmatrix} U_{11} & U_{12} & U_{13} & \dots & U_{1N} \\ U_{21} & U_{22} & U_{23} & \dots & U_{2N} \\ U_{31} & U_{32} & U_{33} & \dots & U_{3N} \\ U_{41} & U_{42} & U_{43} & \dots & U_{4N} \\ U_{51} & U_{52} & U_{53} & \dots & U_{5N} \end{bmatrix} \tag{9}$$

Where U_{ii}; i $\in$1…5 is the local uncertainty of the i-th classifier, and U_{ij}, I, j$\in$1…5; i $\neq$ j is the global uncertainty of the i-th classifier, when it knows the ranking of the j-th classifier. It is worth mentioning that the reason the uncertainties in the above two equations are evaluated using a logarithm with base 2 is that this can be demonstrated by plotting equation (7) where U_{ii} is a function of parameter R_i:

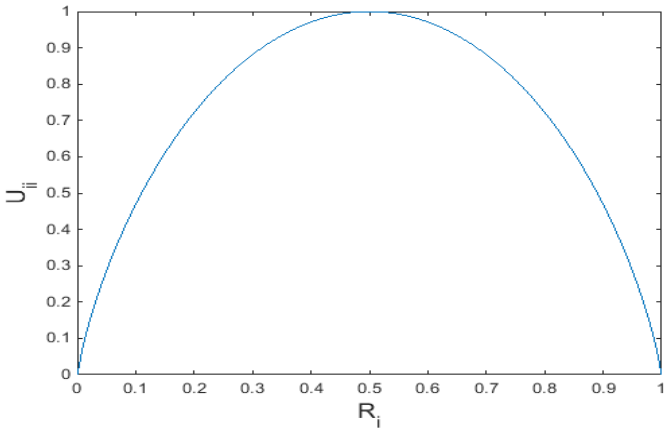

Fig. 2. Uncertainty value U_{ii} as a function of the parameter R_i

From the plot in Figure 2, it is clear that, if the value of the classifier's ranking is close to the edges of the [0,1] interval, uncertainty will be near zero (the classifier is certain about its decision). On the other hand, if the ranking is close to the 0.5 point, uncertainty will be close to the maximal value, which is one.

Regarding the estimation of classifier uncertainties, calculating local uncertainty is straightforward as the classifier is assessing its own decision. On the other hand, when it comes to evaluating global uncertainty, it is unknown how classifier *i* is influenced by the decision of classifier *j*. Hence, equation (10) is proposed to estimate this influence:

$$c_{ij} = \frac{\tanh\left((A_j - A_i)\cdot\phi_i\right)+1}{2} \tag{10}$$

Where A_i is the general accuracy of the i-th classifier, evaluated on the training set. Using hyperbolic functions is useful for solving classification problems then combining different classifiers using neural networks (Toh and Yau, 2004; Toh et al., 2007). This function is useful in this case because, if the difference between $A_j - A_i$ is increasing, the ratio of impact of the j-th classifier to the impact of the i-th classifier $\frac{c_{ij}}{1-c_{ij}}$ is increased exponentially with regard to k. In other words, if the difference $A_j - A_i$ changes from x to $k \cdot x$, then the ratio of impact of the j-th classifier to the impact of the i-th classifier $\frac{c_{ij}}{1-c_{ij}}$ will change from $e^{2\phi_i x}$ to $\left(e^{2\phi_i x}\right)^k$. The parameter ϕ_i is defined for all datasets during training process which varies from 0.5 to 1.

Another reason to use equation (10) is that

- It is defined for all real numbers.
- It is monotonous, so an increasing $A_j - A_i$ will always increase the impact of the j-th classifier.
- The values of equation (10) lie between 0 and 1, so it is convenient to use the values of this function in linear combination.

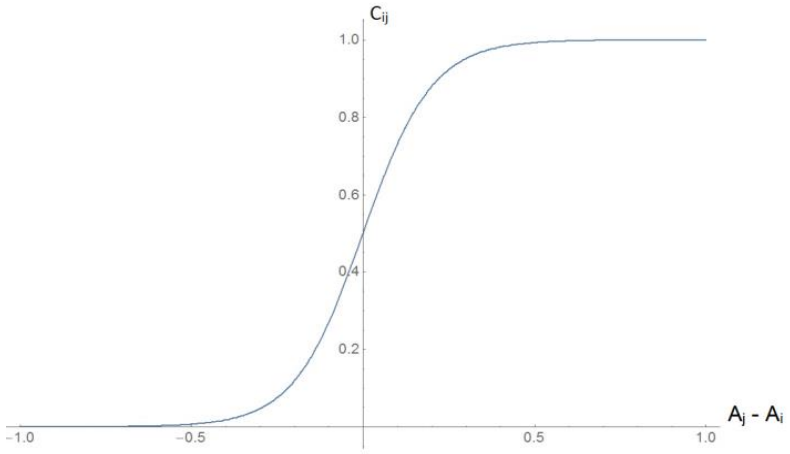

Fig. 3. Illustrative plot of equation (10), $\phi_i = 5$

It can be seen from equation (10) in Figure 3 that, if $c_{ij} = 1/2$, the i-th and j-th classifier accuracies are the same; if the i-th classifier accuracy is far greater than that of the j-th, then it tends to zero, and otherwise it tends to one. Then, $R_i(\gamma_k|\Gamma_j)$ is evaluated as a linear combination of the i-th and j-th experts' ranking of γ_k:

$$R_i(\gamma_k|\Gamma_j) = R_j(\gamma_k) \cdot c_{ij} + R_i(\gamma_k) \cdot (1 - c_{ij}) \tag{11}$$

The derived equations (8) and (10) are new, since the global ranking for each classifier is not available, and needs to be estimated somehow. The non-linear function $\frac{tanh(\alpha x)+1}{2}$ is used, which tends to 0 with large negative x, and to 1 with large positive x. So, when the i-th classifier accuracy is far greater than the j-th's accuracy, then the i-th ranking will remain almost unchanged.

Step 3. *Evaluating the weights of classifiers*

After evaluating the uncertainties of the classifiers in matrix U, matrix w will be evaluated via

$$w_{ij} = \frac{1}{U_{ij}^2 \sum_{k \in A} U_{kl}^{-2}} \tag{12}$$

Step 4. *Evaluating vector* π

Vector π is evaluated as an approximate solution for the following equation:

$$\begin{pmatrix} \pi \cdot W & = \pi \\ \sum_{i=1}^{N} & \pi = 1 \end{pmatrix} \tag{13}$$

Here, the weights of the matrix are considered as the transition matrix of a Markovian chain with single classifiers as stated in Shaban et al. (2002). Then, the stationary distribution π of this chain is evaluated using a system of equations. This system can be converted into the form of

$$\begin{pmatrix} \widetilde{W}\pi & = 0 \\ \sum_{i=1}^{N} & \pi = 1 \end{pmatrix} \tag{14}$$

where $\widetilde{W} = (W - E)^T$, the sum of the elements for each column of matrix $\widetilde{W}$ is equal to 0, and matrix $\widetilde{W}$ is singular; hence, if

$$rank(\widetilde{W}) = N - 1 \tag{15}$$

then equation (14) has a single exact solution. By changing the parameters ϕ_i in equation (10), equation (15) can be achieved and used to solve equation (13), hence the least squares method is proposed. By using this method, there is no need to worry about vector π convergence, because the results of the approximate solution of (14) when (15) is fulfilled will be the same as using DeGroot's (1975) iterative method $\pi^{i+1} = \pi^i W$ with normalization at each step.

Step 5. *Aggregating consensus rankings*

This step comes when all classifiers reach a consensus about their decisions and there is no room for decision updates. Here, the aggregate consensus ranking is evaluated using the following equation:

$$R_G(\gamma_k) = \sum_{i=1}^{N} R_i(\gamma_k) \cdot \pi_i \tag{16}$$

Vector π is the importance weights for each single classifier, and the sum of all its elements equals 1. So, the aggregate consensus ranking can be evaluated as a linear combination of a single classifier's rankings.

Step 6. *Group consensus final answer*

Length of vector R_G is equal to the size of the set of the possible answers, and the sum of all element of R_G is equal to 1. The final prediction of the group using the consensus method is the answer γ_*, for which $R_G(\gamma_*)$ reaches the maximum value. Thus, using formal language, the final answer of the group can be specified as

$$\gamma_* = Arg \max_{a \in (\gamma_1, \ldots, \gamma_m)} R_G(a) \tag{17}$$

Figure 4 shows a flowchart of the pseudo-code for the consensus approach, based on generating a common group ranking for one data sample or input:

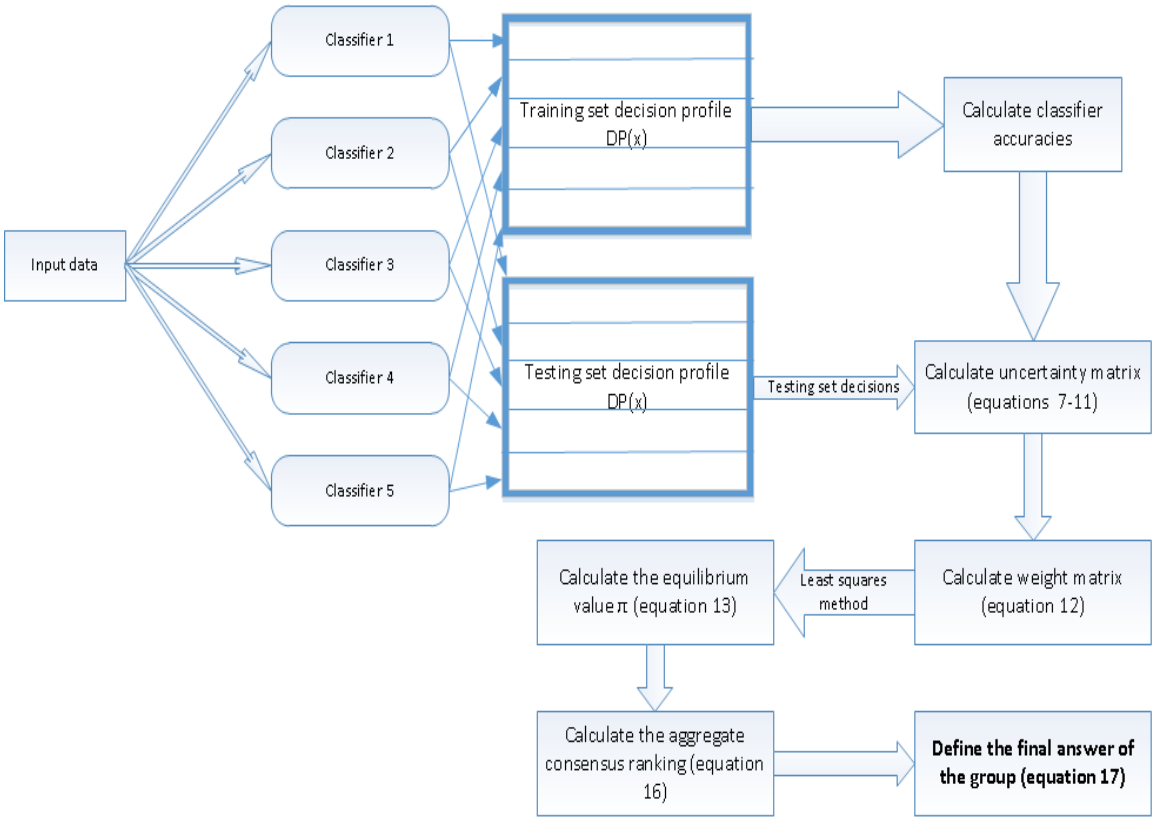

Fig. 4. The process of the classifier consensus approach

Input: R_i – ranking of answer "1" for each classifier, $i = 1..5$ and A_i– is the accuracy of each agent.
Output:
1. **for** i = 1 to N **do**
2. **for** j = 1 to N **do**
3. **if** (i==j) **then** $U_{ii} =$ (computed by equation (3)) **else**
4. c_{ij} = (computed by equation (10))
5. $R_i(\gamma_k|\Gamma_j)$ = (computed by equation (11))
6. U_{ij} =(computed by equation (8))
7. **end if**
8. **end for**
9. **end for**
10. $\forall i, j \in \{1..5\}$ $w_{ij} =$ (computed by equation (12))
11. Compute $\widetilde{W} = (W - E)^T$
12. Compute π =(computed by equation (14))
13. Compute aggregate consensus $R_G(\gamma_k)$ using equation (16)
14. Define aggregate group answer using equation (17).

Procedure of the consensus approach

4. Experimental set-up

4.1. Real-world credit datasets

Referring to Table 2 the majority of the previous studies used from one to three datasets in order to evaluate the predictive performance of the proposed model against other models and to reach a trustworthy conclusion, whereas here five real-world credit datasets are used for model validation. Three credit scoring datasets from the UCI repository (Asuncion and Newman, 2007), namely, German, Australian and Japanese, were employed. In addition, corporate and bankruptcy datasets were used for extra validation. The Iranian dataset, which consists of corporate client data from a small private bank in Iran, has been used in several studies (Sabzevari et al., 2007; Kennedy, 2012; Marqués et al., 2012). The Polish dataset contains information on bankrupted Polish companies recorded over two years (Pietruszkiewicz, 2008; Kennedy, 2012; Marqués et al., 2012). A summary of all the datasets is illustrated in Table 2.

4.2. Baseline model development

The baseline models developed to be part of the multiple classifier systems are NN, SVM, DT, RF and NB. Furthermore, LR and MARS, which are the reference points to our final model, are used. Below is the theoretical background of the classifiers are described.

- *NN* is an AI technique that mimics human brain function, which is made up of interconnected neurons that process information concurrently, and consists of three layers: input, hidden and output. The NN model for credit scoring problems starts by passing the attributes of each applicant to the input layer processing; then they are transferred to the hidden layer for further processing, and the values are sent to the output layer, which gives the final answer to the problem: either to give or not to give a loan. The output is calculated using weights. These are assigned to each attribute according to its relative importance; then all weighted attributes are summed together and fed to a transfer function (i.e., sigmoid, tangent-sigmoid) to create output. This gives a final result based on adjusting weights iteratively by minimizing the error between the predicted output and the actual targets (Jensen, 1992; Haykin, 1999; Malhotra and Malhotra, 2003).

- *SVM* is another AI technique used in classification and credit scoring problems. Assume an input training vector (x_1,y_1), (x_2,y_2),…, (x_n,y_n) exists, where $x \in R^d$, which is a vector in d-dimensional feature space, and $y_i \in \{1,+1\}$ is the class label. Here, the SVM tries to find an optimal separating hyperplane that separates data correctly in two classes (good and bad loans), so the margin width between the optimal hyperplane and the support vectors (training data on the margin) is maximized to fit the data. Based on the features of the support vectors, which applicant belongs to a good or bad credit rating can be predicted. In case the data are not linear, the SVM uses different types of kernel functions, which transform the data in higher dimensional space. Linear, radial basis function (RBF) and polynomial are types of SVM kernels (Zhou et al., 2010; Teng et al., 2010).

- *DT* is commonly used for classification purposes in credit scoring applications, which uses graphical tools. The node is shown in a box with lines to indicate possible events and their consequences, until the best and optimal outcome is reached. The idea behind the decision tree is to classify into two classes of credit good and bad. It begins with a root node that contains the two types of classes; then the node splits into two subsets representing possible events. The decision algorithm loops on all the splits to find the optimal one and selects the winning sub-tree that gives the best partitioning of mostly good and bad credit based on the overall error rate and lowest misclassification cost (Breiman et al., 1984; Hand and Henley, 1997).

- *RF:* these are defined as a group of unpruned classification or regression trees, trained on bootstrap samples of the training data using random selected variables or features in the process of tree generation. After a large number of trees have been generated, each tree votes for the most popular class. These tree voting procedures are collectively defined as random forests and for their classification technique two parameters require tuning: the number of trees and the number of attributes used to grow each tree (Breiman, 2001).

- *NB:* these are statistical classifiers that predict a particular class (good or bad loan). Bayesian classification is based on Bayesian theory. For example, assume, given the training sample set $D = \{d_1, d_2,....,d_n\}$, that the task of the classifier is to analyse these training set instances and determine a mapping function $f: \{x_1,....,x_n\} \rightarrow C$, which can decide the label of an unknown ensample $x = (x_1,....., x_n)$. Bayesian classifiers choose the class that has the greatest posterior probability $P(c_j \mid x_1,....., x_n)$ as the class label, according to a minimum error probability criterion or a maximum posterior probability criterion. That is, if $P(c_i \mid x) = max\ P(c_j \mid x)$, then assigning x to a particular class c_i can be determined. The NB classifier assumes that the variables of samples are independent. In practice, however, dependences can exist between the variables (Bishop, 2006; Antonakis and Sfakianakis, 2009).

Table 2

Description of the five datasets used in the study

Dataset	#Loans	#Attributes	Training set size	Testing set size	Good/ Bad
German	1000	20	800	200	700/300
Australian	690	14	552	138	307/383
Japanese	690	15	552	138	296/357
Iranian	1000	27	800	200	950/50
Polish	240	30	192	48	128/112

Practically, a few parameters need to be set up before classifier construction for NN, SVM and RF. However, the intention is to make a unique model for all datasets. For the NN model, a feed-forward back-propagation is constructed based on one hidden layer of 40 neurons, which is established by a trial and error process. Furthermore, the training epochs were 1000, and the activation function was "pure-linear". Regarding SVM, an RBF kernel was used with two parameters to tune C and gamma. The former controls the trade-off between errors of the SVM on training data and margin maximization, while the latter handles non-linear classification. Regarding this, C is set to 2 and gamma is set to 2^{-3}. In random forests the most important parameters are the number of trees and attributes used to build a tree. 60 trees are built and the number of features used varied from 15 for the German set to 11 each for the Australian and Japanese sets, while 20 and 22 are employed for the Iranian and Polish datasets respectively.

4.3. Benchmark model development

In order to measure how well the consensus approach has performed, the results of the proposed model are compared with two benchmark models, namely, LR and MARS. LR is the industry standard for developing credit scoring models (Crook et al., 2007; Lessmann et al., 2015b). However, Lessmann et al. (2015b) have stated that it is beneficial to compare a new method with the standard one as well as other established techniques. On the other hand, MARS has been used in several studies as a benchmark model (de Andrés et al., 2011; Sánchez-Lasheras et al., 2012).

- *LR*: this has been considered as the industry standard until now in credit scoring model development (Lessmann et al., 2015). It is an extensively used statistical technique that is popular for solving classification and regression problems. LR is used to model a binary outcome variable, usually represented by 0 or 1 (good and bad loans). The LR formula is expressed as (Atiya, 2000)

$$log[p(1-p)] = \beta_0+\beta_1X_1+\beta_2X_2+....+\beta_nX_n. \quad (18)$$

where p is the probability of the loan status result, either by (0/1), β_0 is the intercept term, βi represents the coefficient related to the independent variables *Xi (i=1… n)* and *log [p (1-p)]* is the dependent variable, which is the logarithm of the ratio of two probability outcomes of interest. The objective of LR in credit scoring is to determine the conditional probability of a specific input (the characteristics of the customer or features) belonging to a certain class.

- *MARS* is a non-parametric and non-linear regression technique developed by Friedman (1991), which models the complex relationships between the independent input variables and the dependent target variable. MARS is built using piece-wise linear regression by modelling a sequence of linear regressions on different intervals or splines (Briand et al., 2004). Each spline should be specified by finding the suitable independent variable to fit. MARS is built in the form of

$$y = c_0 + \sum_{i=1}^{k} c_i B_i(x) \quad (19)$$

where c_0 is a constant coefficient, $B_i(x)$ is the basis function and c_i is a coefficient for the basis function. MARS uses what is called the basis function where it takes numerous forms of independent variables' interactions; the common functions used are the hinge functions that are used to find variables which are selected as knots, hence the function takes the following form (Friedman, 1991):

$$max\ (0, X - c) \text{ or,} \quad (20)$$

$$max\ (0, c - X) \quad (21)$$

Where c is a constant, threshold or knot location, X is the independent variable. The goal behind the basis functions is to transform the independent variable X into a new variable (e.g., X^*). Based on equations (19) and (20), X^* will take the value of X if X is greater than c and it will take the value of zero if the value of X is less than c (Briand et al., 2004). MARS refits the model after all terms involving the variable are removed to be assessed and calculates the reduction in the model's error, and then all variables are categorized according to their influence on the performance of the model; the optimal MARS model is based on the lowest generalized cross-validation (GCV) measure (Briand et al., 2004). For more insight into MARS modelling, please refer to Friedman (1991) and Hastie et al. (2001).

4.4. Performance indicator measures

To validate our model and to reach a reliable and robust conclusion on its predictive accuracy, four performance indicator measures are adopted, namely, ACC, AUC, the H-measure and BS. These are chosen because they are popular in credit scoring and they cover all aspects of model performance. The ACC stands for the proportion of correctly classified good and bad loans, which measures the predictive power of the model. As such, this is a criterion that measures the discriminating ability of the model (Lessmann et al., 2105a). An alternative interpretation of the AUC is that it is a measurement used in binary classification analysis to determine which of the models used predict the classes best. According to Hand (2009), the AUC is used to estimate the models' performance without any prior information about the error costs. However, it assumes different cost distribution among classifiers depending on their actual score distribution, which prevents them from being compared effectively. As a result, Hand (2009) proposed the H-measure as an alternative measure to the AUC for measuring classification performance, because the H-measure assumes different cost distribution between classifiers without depending on their scores. In other words, this measure finds a single threshold distribution for all classifiers. Finally, BS measures the accuracy of the probability predictions of the datasets and the difference between it and ACC is that it directly takes the probabilities into the account, while ACC transform these probabilities into 0 or 1 based on a determined cut-off score. Subsequently, the lower the BS the better the predictions that are calibrated.

4.5. Statistical tests of significance

According to García et al. (2015), it is not sufficient to prove that one model achieves results better than another, because of the different performance measures or splitting techniques used. For a complete performance evaluation, it would seem appropriate to implement some hypothesis testing to emphasize that the experimental differences in performance are statistically significant, and not just due to random splitting effects. Choosing the right test for specific experiments depends on factors such as the number of datasets and the number of classifiers to be compared.

According to Demšar (2006), statistical tests can be parametric (e.g., paired t-test) and non-parametric (e.g., Wilcoxon, Friedman test) However, Demšar advised that non-parametric tests are preferable to parametric tests as they can be conceptually inappropriate and statistically unsafe. Non-parametric tests may be more appropriate and safer than parametric tests since they don't assume normality of data or homogeneity of variance (Demšar, 2006). Accordingly, in

this study, we embraced the Friedman test to compare the ranking performance of all the models measured across all datasets.

The Friedman (1940) test is a non-parametric test that ranks the classifiers for each dataset separately. The best-ranking classifier is given a rank of one, the second-best classifier ranked second and so on. Under the null hypothesis of Friedman, which test that all classifiers from this group perform identically and all differences are only random fluctuations. The Friedman statistic x_F^2 is distributed according to x_F^2 with K – 1 degrees of freedom when N (number of datasets) and K (number of classifiers) are big enough (Demšar, 2006). If the null hypothesis of the Friedman test is rejected, then it is possible to proceed to a post hoc test in order to find the particular pair-wise comparisons that produce significant differences.

For instance, the Bonferroni–Dunn (1961) test can be used when all classifiers are compared with a control model (Demšar, 2006; Marqués et al., 2012). With this test, the performance of two or more classifiers is significantly different if their average ranks differ by at least the critical difference (CD), as follows:

$$CD = q_{\propto} \sqrt{\frac{k(k+1)}{6N}} \tag{22}$$

Where $q_{\propto}$ is calculated as a studentised range statistic with a confidence level $\propto/(k\text{-}1) = \propto/14$ divided by $\sqrt{2}$. Also, here k = 15 (number of classifiers), N = 5 (number of datasets).

5. Experimental results

In this section, a multiple classifier system based on a consensus approach is proposed, along with seven traditional combination methods, validated over the above-described five real-world credit datasets. To ensure the diversity of the classifiers to be used, which is the main element in the ensemble models, it has been decided to train each classifier independently using the bagging sampling method, where each classifier is trained on different parts of the training data with resampling, and subsequently each classifier's final predictions are averaged. Finally, the averaged predictions of each will be ready to be fused using the consensus approach. To minimize the effect of the variability and to reach a reliable conclusion the experiments will be reiterated 100 times based on different training and testing sets at each run, but all the models are trained and tested on an identical partition of the dataset.

5.1. Data pre-processing and partitioning

The crucial step before building the model is to prepare the data for training. First, a check is done for any missing variables, which are replaced via an imputation approach by replacing the missing variable by the average or mean value of the entries.

Each variable in the datasets contains values that differ in range. In order to avoid bias and build the models with data within the same interval, they should be transformed from different scale values to common ones. To accomplish this, the dataset attributes are normalized to values in the range between 0 and 1. These transformations are performed by taking the maximum value within each attribute and dividing all of the values for each by its maximum value.

The main idea behind data partitioning is to break the data into two parts: one for learning and the other for evaluating. In the credit scoring literature, different splitting methods were used: the most common is to partition the dataset into training (learning) and testing (evaluation) sets (Wang et al., 2012; Tsai, 2014; Nanni and Lumini, 2009). So, in accordance with common practice, each dataset is divided into 80% training to build the model and 20% for evaluating. According to García et al. (2015), despite there being various splitting methods available and the factors that affect their use, such as data size, partitioning the dataset depends on the preference of each author. All the experiments for this study are performed using MATLAB, 2014b version, on a PC with 3.4 GHz, Intel CORE i7 and 8 GB RAM, using the Microsoft Windows 7 operating system.

5.2. Classification results

Our aim in this empirical evaluation is to demonstrate that combining heterogeneous multiple classifier systems through the consensus approach can lead to better performance than any base classifier in the system, traditional combination method and the industry-standard LR and MARS against different performance measures. To validate our approach and to reach a reliable conclusion, Tables 3–7 review the performance indicator measures of the five base classifiers that make up the ensemble, seven traditional combination methods and the proposed consensus approach on the five datasets. As can been seen in Tables 3–7, the proposed consensus combination approach exhibits better performance in all four performance measurements across all the datasets, except in the Iranian case.

Starting with the German dataset regarding ACC, the consensus approach achieves 77.72%, beating the best traditional combination method (majority voting) and best base classifier (RF) by 1.75% and 1.51% respectively. The AUC of consensus achieves 80.23%, which indicates its separation ability between classes. In addition, RF in the base classifier comes second with 78.47% and among the traditional combiners, mean rule is the best, scoring 78.46%. The H-measure proves that consensus is better at dealing with cost assumptions between classes, as it achieved 30.95%, which is 3.44% better than RF, coming in second place. Finally, in terms of the BS, consensus achieves the best accuracy of the probabilities: 15.77% is below mean rule, in second place with 16.52%, and the lower the score the better. Finally, it is found that with the German dataset the RF comes second to the consensus approach on all the performance measures, while mean rule is the best among the traditional combination methods.

Table 3
Classifier results for the German dataset over the different performance measures

	Base classifiers					Traditional combination methods							Proposed
Performance measure	NN	SVM	RF	DT	NB	Min.	Max.	Product	Mean	Majority voting	Weighted average	Weighted voting	Consensus approach
ACC	0.7555	0.7216	0.7621	0.7242	0.7263	0.7401	0.7195	0.7077	0.7591	0.7597	0.7201	0.7430	**0.7772**
AUC	0.7788	0.6696	0.7847	0.7290	0.7426	0.7016	0.7661	0.6760	0.7846	0.7447	0.7121	0.7121	**0.8023**
H-measure	0.2680	0.1506	0.2751	0.1799	0.2680	0.1853	0.2434	0.1855	0.2714	0.2542	0.1787	0.1389	**0.3095**
Brier score	0.2440	0.2847	0.2269	0.2276	0.2440	0.2675	0.2376	0.2918	0.1652	0.1857	0.2221	0.2464	**0.1577**

Table 4
Classifier results for the Australian dataset over the different performance measures

	Base classifiers					Traditional combination methods							Proposed
Performance measure	NN	SVM	RF	DT	NB	Min.	Max.	Product	Mean	Majority voting	Weighted average	Weighted voting	Consensus approach
ACC	0.8590	0.8133	0.8707	0.8486	0.8058	0.8219	0.8466	0.8034	0.8545	0.8641	0.8622	0.8707	**0.8798**
AUC	0.9278	0.8944	0.9351	0.9111	0.8335	0.8718	0.9132	0.8484	0.9273	0.9196	0.9130	0.9130	**0.9404**
H-measure	0.6419	0.5788	0.6597	0.5813	0.4754	0.5317	0.6079	0.5264	0.6256	0.6349	0.6158	0.6128	**0.6719**
Brier score	0.3669	0.3916	0.0935	0.3218	0.4578	0.2582	0.1571	0.3479	0.1036	0.1036	0.1204	0.1292	**0.0920**

Table 5
Classifier results for the Japanese dataset over the different performance measures

	Base classifiers					Traditional combination methods							Proposed
Performance measure	NN	SVM	RF	DT	NB	Min.	Max.	Product	Mean	Majority voting	Weighted average	Weighted voting	Consensus approach
ACC	0.8485	0.8194	0.8697	0.8378	0.7769	0.8396	0.7927	0.8229	0.8557	0.8669	0.8357	0.8664	**0.8788**
AUC	0.8981	0.8773	0.9294	0.9067	0.8179	0.8781	0.8837	0.8795	0.9162	0.9130	0.8705	0.8705	**0.9328**
H-measure	0.5928	0.5533	0.6418	0.5614	0.3711	0.5535	0.5598	0.5744	0.6108	0.6205	0.5311	0.5969	**0.6653**
Brier score	0.3920	0.4491	0.0979	0.3259	0.3955	0.3049	0.1624	0.4091	0.1120	0.1076	0.1378	0.2309	**0.0946**

Table 6
Classifier results for the Iranian dataset over the different performance measures

	Base classifiers					Traditional combination methods							Proposed
Performance measure	NN	SVM	RF	DT	NB	Min.	Max.	Product	Mean	Majority voting	Weighted average	Weighted voting	Consensus approach
ACC	0.8188	0.9449	0.9506	0.9496	0.2645	0.9498	0.2382	0.9496	0.9263	0.9490	0.9111	0.9506	**0.9569**
AUC	0.5404	0.7185	**0.7798**	0.7274	0.5427	0.5848	0.5529	0.5357	0.6772	0.6022	0.7407	0.7407	0.7761
H-measure	0.0160	0.2592	0.2833	0.2039	0.0141	0.1386	0.0240	0.1476	0.1297	0.0629	0.2572	0.0673	**0.3342**
Brier score	0.0534	0.066	**0.0437**	0.0558	0.7634	0.0489	0.7628	0.0502	0.0928	0.0747	0.0476	0.0478	0.0580

Table 7
Classifier results for the Polish dataset over the different performance measures

	Base classifiers					Traditional combination methods							Proposed
Performance measure	NN	SVM	RF	DT	NB	Min.	Max.	Product	Mean	Majority voting	Weighted average	Weighted voting	Consensus approach
ACC	0.6169	0.6717	0.7596	0.7573	0.6879	0.7239	0.6852	0.6439	0.7563	0.7575	0.7406	0.7596	**0.7681**
AUC	0.6719	0.7979	0.8322	0.8273	0.7612	0.7983	0.7678	0.8055	0.8261	0.8209	0.8051	0.8051	**0.8406**
H-measure	0.1144	0.3173	0.3735	0.3607	0.2391	0.3215	0.2622	0.3472	0.3585	0.3541	0.3234	0.3091	**0.3869**
Brier score	0.3342	0.3748	0.1672	0.3205	0.3910	0.4048	0.2871	0.4576	0.1721	0.1769	0.2142	0.2401	**0.1623**

In the Australian dataset, the consensus approach provides enhancement over the best base classifier (RF) and best traditional combiner (weighted voting) by 0.91%. The AUC of consensus reaches 94.04%, which indicates its separation ability between classes. Once again, the RF comes second in the base classifiers, with 93.51%, and the best of the

traditional combiners is mean rule, scoring 92.73%. The H-measure verifies that consensus is better at dealing with cost assumptions between classes, as it scores 67.19%, better than the RF (in second place) by 1.22%. Finally, for the BS the consensus achieves the best accuracy of the probabilities, at 9.2%, with RF second at 9.35%. The reason why the H-measure and BS deliver better results than those in the German set might be that the Australian set is more balanced. Finally, RF presents a challenge to our consensus approach in all performance measures, while majority voting performance is the most stable among the traditional combination methods.

Looking at the Japanese dataset, consensus improves the model performance over the best base classifier (RF) and best traditional combiner (majority voting) by 0.91% and 1.19% respectively. Regarding the AUC, consensus achieves 93.28%. Once again, RF comes second in the base classifiers with 92.94%, and the best of the traditional combiners is mean rule, with 91.62%. The H-measure of the consensus approach achieves 67.19%, which is superior to RF, in second place with 64.18%. Lastly, the Brier score of the consensus method attains the best accuracy of the probabilities with 9.46% and RF, once again, is a close challenger on 9.79% (lower scores being preferred). This again provides evidence that balanced datasets are the reason why the H-measure and Brier score of the Japanese and Australian datasets are higher than for the German. Overall, regarding the Japanese dataset, the RF results again indicate that it is a close rival to consensus for all performance measures, while again majority voting performance is better, on average, among the traditional combination methods.

To summarize, regarding the three credit scoring datasets, the consensus method performs significantly better than other approaches for the German dataset and the performance for both the Australian and Japanese datasets is, if not as notable, still an improvement. The surprising thing is that the pattern of improvement among the classifiers in the three datasets is almost the same, in that RF achieves the highest performance after the consensus method across all the performance measures, NN comes after RF in performance among the base classifiers, and NB is the poorest performer for everything. Regarding the traditional combination methods, it has emerged that the prediction accuracy of mean rule does well in semi-balanced datasets, while those of majority voting and weighted average perform well in balanced ones.

In the last two datasets, which relate to bankruptcy, first the Iranian data is considered, which is very highly imbalanced towards positive classes. Regarding predictive accuracy, the consensus approach outperforms RF, the best base classifier, by 0.63%, while weighted voting was the best of the traditional combiners and achieved the same accuracy as RF. Concerning AUC, RF shows a powerful performance against the consensus approach, beating it by 0.37%, thus showing it has better separation ability between classes This could be due to there being very few negative classes or bad loans in the testing set, and RF is able to classify them better than the consensus method, and the base classifiers within the latter could not reach agreement on these points. In addition, it can be seen that the AUC values for NN and NB are low compared to all the other classifiers. In the H-measure, consensus achieves 33.42%, heading RF by 0.51%. Finally, for the BS, again, RF attains the better performance when compared to the consensus method. The reason might be that the accuracy of the probability of bad loans of RF is higher than that of consensus, hence the better score for the former. One surprising finding is that NB's predictive performance in severely imbalanced datasets is terrible, as it achieves an ACC of only 26.45%. Regarding traditional combiners, weighted voting does well in ACC and mean rule does so for the rest of the performance measures.

The results for the Polish dataset reveal the superiority of the consensus approach over the other classifiers for all the performance measurements. The consensus average accuracy reaches 76.81%, which is 0.85% better than RF and weighted voting, which attain the same accuracy of 75.96%. The consensus approach's AUC achieves 84.06%, showing better separation ability of classes than all the other classifiers. The H-measure and Brier score are 38.69% and 16.23% respectively. As in the Iranian dataset, weighted voting does well in ACC and mean rule for the rest of the performance measures.

It can be seen from the Iranian and Polish datasets that, on average, consensus does better in both datasets, except for AUC and the Brier score in the former. RF emerges as a strong classifier, being the best classfier across all the performance measures, and weighted voting prediction accuracy is good on both datasets.

To summarize, according to the experimental results the following conclusions can be drawn.

1. The consensus approach has proven to be a reliable and efficient combination or fusion method when combining heterogeneous classifiers across several performance indicator measures and several dataset distributions.

2. RF is an efficient classifier in credit scoring, achieving better results than all the other base classifiers, including NN and SVM. However, using RF as a classification can produce very efficient and competitive scoring models (Lessmann et al., 2015b).

3. No traditional combination method attained better results than the best base classifier.

4. Weighted voting has emerged as being a good combination technique as it can achieve better predictive accuracy than the best base classifier (RF).

In order to reach consistent conclusions on how well the consensus approach is performing, a comparison is carried out with the benchmark techniques LR and MARS using all performance measures across the five datasets. Table 8 illustrates the comparison results.

Table 8
Results of the consensus approach, LR and MARS across five datasets

Logistic Regression (LR)					
	German	Australian	Japanese	Iranian	Polish
ACC	0.7555	0.8594	0.8623	0.9473	0.7208
AUC	0.7794	0.9296	0.9112	0.7591	0.7997
H-measure	0.2708	0.6461	0.6267	0.2448	0.3043
Brier score	0.2269	0.2754	0.2830	0.1044	0.2718
MARS					
	German	Australian	Japanese	Iranian	Polish
ACC	0.7598	0.8681	0.8688	0.948	0.7218
AUC	0.77768	0.9345	0.9335	0.7364	0.7972
H-measure	0.1691	0.0989	0.0981	0.0471	0.1870
Brier score	0.2648	0.6533	0.6501	0.2005	0.3012
Consensus approach					
	German	Australian	Japanese	Iranian	Polish
ACC	**0.7772**	**0.8798**	**0.8788**	**0.9569**	**0.7681**
AUC	**0.8023**	**0.9404**	**0.9328**	**0.7761**	**0.8406**
H-measure	**0.3095**	**0.6719**	**0.6653**	**0.3342**	**0.3869**
Brier score	**0.1577**	**0.0920**	**0.0946**	**0.0580**	**0.1623**

From Table 8, it is clear that the consensus approach is superior to LR and MARS for all the performance measurements and across the five datasets. The improvement of consensus over LR and MARS in predictive accuracy varies from 0.96% to 4.73% and from 0.89% to 4.63% respectively across the datasets. Its separation ability is better, as shown by the AUC figures. H-measure values are better for consensus than LR and MARS, except for the Japanese dataset which indicates that at some thresholds MARS is slightly better than consensus. Despite AUC values indicate that the consensus approach has the ability to handle cost distributions across classifiers independently from their scores. Finally, the Brier score shows that the consensus probability accuracies outperform LR and MARS for both classes. Moreover, the results for RF support the conclusion by Lessmann et al. (2015b) that RF is an efficient and promising classifier for developing credit scoring models as it has superior predictive performance when compared to LR and MARS.

5.3. ROC curve analysis

To demonstrate the separation and discrimination ability of the models and to assess their performance from a different angle as well as measuring their sensitivity (correctly classifying good loans) and specificity (correctly classifying bad loans) over various thresholds, ROC curve plots are executed for the consensus approach, best base classifier and best traditional combining method for each dataset. Figures 5–9 display the ROC curves for the aforementioned models across all datasets.

Each classifier gives some prediction or ranking value in response to the input loan data. Usually, if the value is less than 0.5 the good loans are considered as (0) and otherwise seen as bad (1). Sometimes, the calculated values of sensitivity and specificity for classifiers are insufficient, because of false positive (misclassifying bad loans) or false negative errors (misclassifying good loans), which can be expressed as financial and opportunity costs or losses. One way of increasing one of these parameters is to consider the value 0.5 as a threshold and change it. Increasing this value will lead to an increase in sensitivity, while specificity will decrease. Decreasing the threshold value has the opposite effect, so the cost of increasing one of the parameters is that of decreasing another. The ROC curve is obtained as follows:

1. For each cut-off threshold, which varies from 0 to 1 with the increment of a threshold each time (e.g., 0.01), sensitivity and specificity values are calculated.
2. The ROC curve is plotted with sensitivity along the y-axis and false positive along the x-axis.

The German dataset is considered first, regarding which it can be seen that the consensus ROC curve lies above all the other curves for all threshold values. This means that for this dataset the consensus method is the best for all the required values of sensitivity and specificity. Random forest also has a convex circle-like shape with an optimal threshold value near to 0.5. In the Australian dataset, when compared to the German one, the ROC curves of consensus and random forest are higher, which means lower rates of false negative and false positive errors for all classifiers. The random forest ROC

curve lies below consensus, but for almost all threshold values it is above all the other classifiers, which means it is the second-best performer. Regarding the Japanese dataset, the same conclusions can be drawn as for the Australian one.

The Iranian dataset ROC curves for all classifiers are skewed, which means that the increase in specificity leads to a huge decrease in sensitivity. This can be affected by the small number of bad loan entries in this dataset, which leads to classifiers not being able to learn about their patterns. For optimal cut-off, most of the classifiers have large values for sensitivity, but relatively small specificity values, which mean that these classifiers can't recognize bad loans with sufficient accuracy. Finally, for the Polish dataset ROC curve, the RF is not always convex; this means that it is possible to update this classifier a bit. If the ROC curve is not convex in the range from threshold t_0 to threshold t_1, and the classifier's ranking lies between t_0 and t_1, this ranking is assigned to t_1. In other words, if ranking t_0 < ranking ≤ t_1, then the assigned ranking = t_1. This procedure, made for all entries for the dataset, will change the classifier's range such that the ROC curve will form a straight line from t_0 to t_1.

In conclusion, it is notable that the consensus approach beats all single and combined classifiers for all the datasets. Among the single classifiers, random forest shows the best results, which is because it is not, actually, a single classifier, but, rather, a number of decision trees, which produce ranking using a voting procedure.

5.4. Significance test results

Friedman ranking test outcomes (accuracy rankings) are provided for all single classifiers, all traditional combiners and the consensus approach. Subsequently, to discover any significance differences in the accuracy results, a Bonferroni–Dunn test was carried out. Now, to evaluate the critical values at the significance levels $\propto = 0.05$ and $\propto = 0.1$ were the Bonferroni–Dunn two-tailed test is used. With reference to equation (18), values of $q_{0.05} = 2.8905$ and $q_{0.1} = 2.6653$ were obtained (Demšar, 2006). Hence, the critical difference (CD) values at the 0.05 and 0.1 significance levels are 7.64 and 7.05 respectively. The two horizontal lines, which are at a height equal to the sum of the lowest rank and the critical difference computed by the Bonferroni–Dunn test, represent the threshold for the best-performing method at each significance level ($\propto = 0.05$ and $\propto = 0.1$). The obtained results clearly show that the consensus method is the best, over the base classifiers, LR, MARS and traditional combination methods, across all five datasets. Random forest has good stable results, holding second position for all datasets, while LR, despite being good, performs worse than some of the classical combiners. MARS performs well, beating LR and many base classifiers and traditional combiners. Based on the evaluated critical values, it can be concluded that SVM, product rule, naïve Bayes, maximum rule, weighted average and neural networks are significantly worse than the consensus method at the $\propto = 0.05$ and $\propto = 0.1$ significance levels.

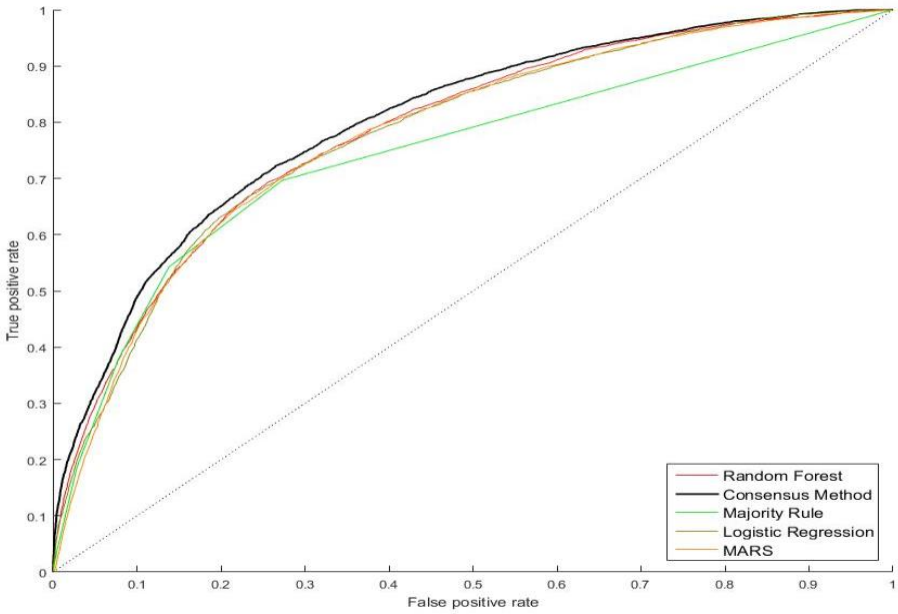

Fig 5. ROC curve comparing the performance of the consensus approach, benchmark classifiers, best base classifier and best traditional combination method on the German dataset

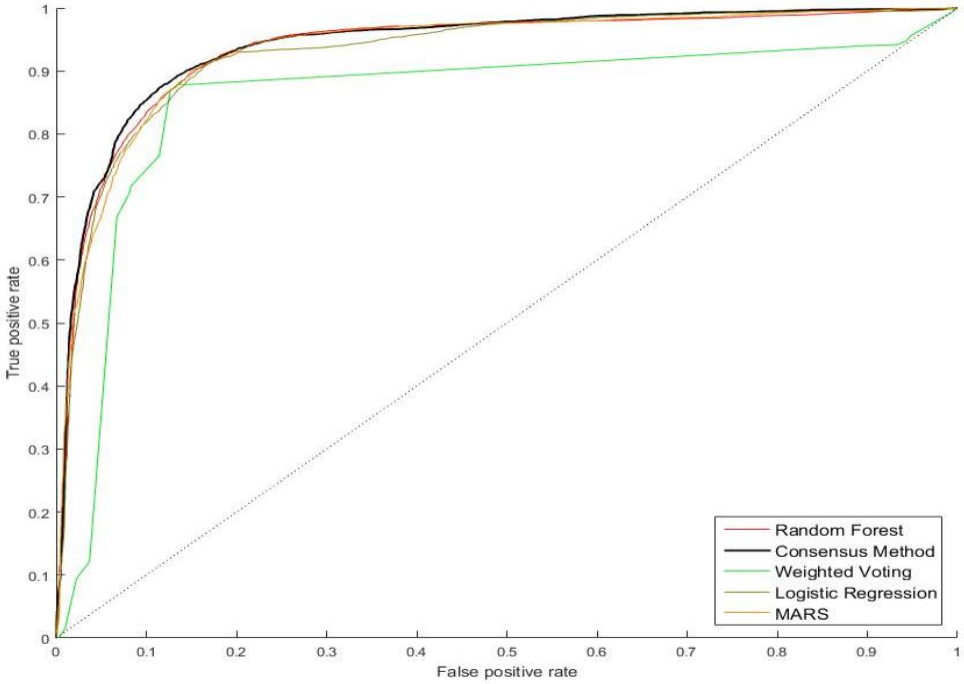

Fig 6. ROC curve comparing the performance of the consensus approach, benchmark classifiers, best base classifier and best traditional combination method on the Australian dataset

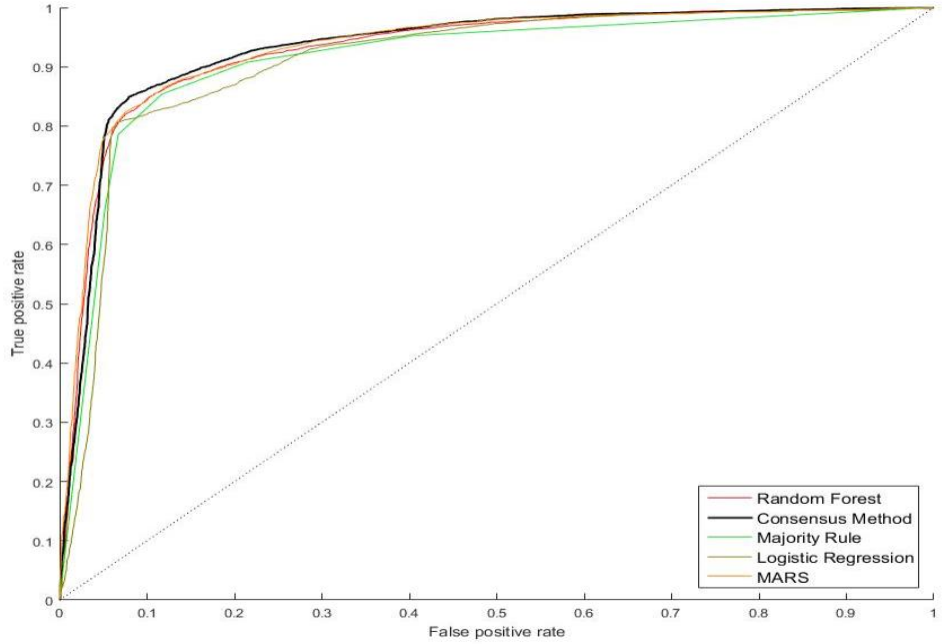

Fig 7. ROC curve comparing the performance of the consensus approach, benchmark classifiers, best base classifier and best traditional combination method on the Japanese dataset

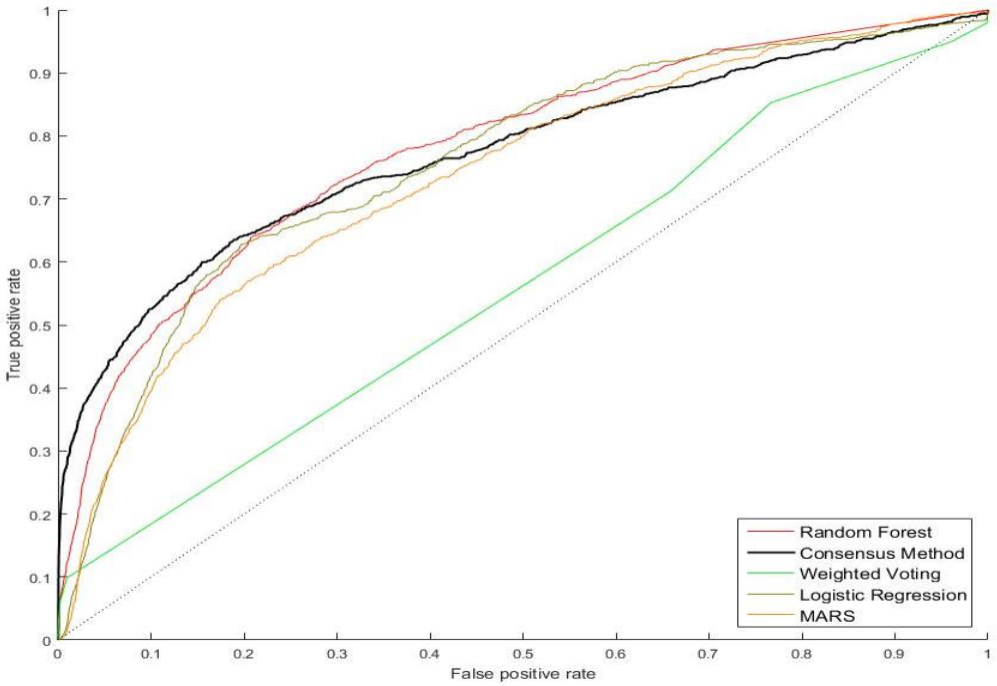

Fig 8. ROC curve comparing the performance of the consensus approach, benchmark classifiers, best base classifier and best traditional combination method on the Iranian dataset

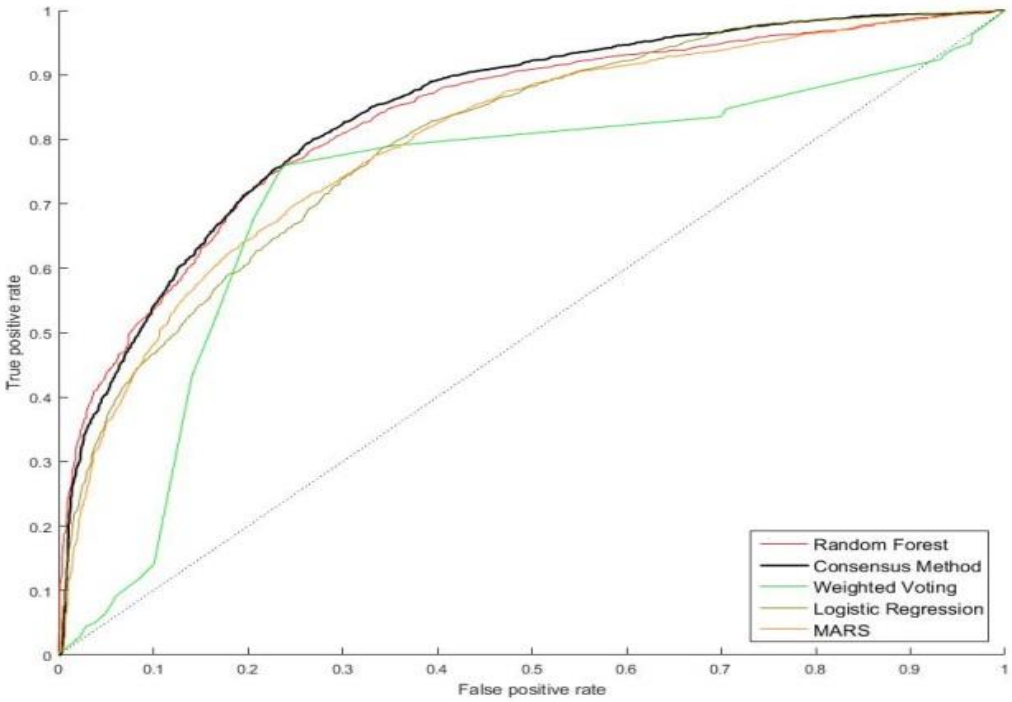

Fig 9. ROC curve comparing the performance of the consensus approach, benchmark classifiers, best base classifier and best traditional combination method on the Polish dataset

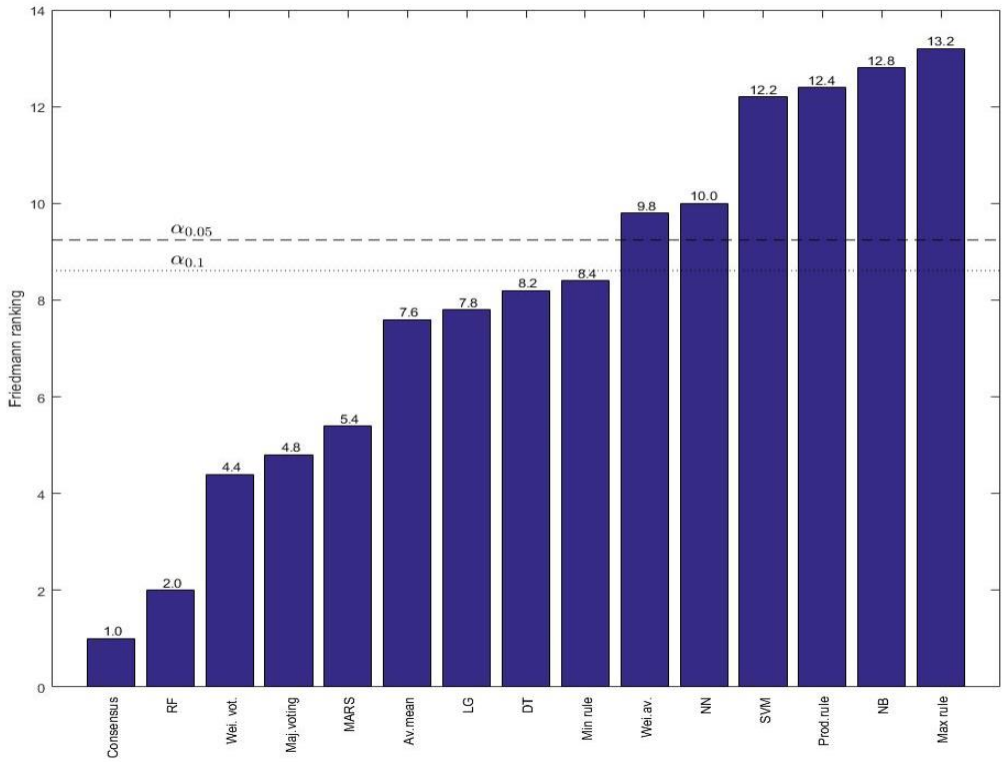

Fig 10. Significance ranking for the Bonferroni–Dunn two-tailed test for the consensus approach, benchmark classifiers, base classifiers and traditional combination methods with $\propto$ = 0.05 and $\propto$= 0.10

6. Conclusions

The main advantage of the consensus method when compared to traditional combiners is that it creates a group ranking as a fusion of individual classifiers' rankings, instead of merging them using arithmetical, logical or other mathematical functions. The method simulates real expert group behaviour: the members continuously exchange opinions and change their measurements of possible answers, being influenced by other experts. The process continues until they come up with a group decision on which they all agree. Sometimes, however, experts cannot achieve this and, hence, the consensus method will fail. To prevent this situation, the least squares method is used instead of an iterations procedure to solve equation (12). Another problem is unknown conditional ranking values, which have been evaluated as a linear combination of two classifier rankings. Moreover, the better the accuracy of the classifier, the more impact it has on others. In other words, R(i|j), which is the conditional ranking of the i-th classifier knowing the ranking of j-th classifier, is close to R(j) if the accuracy of the j-th classifier is greater than that of the i-th classifier; otherwise, it will be close to R(i). The consensus approach is tested on five real-world datasets using four different performance indicator measures, with the goal being to predict the loan quality of the client (0 = good loan, 1 = bad loan) for every focal dataset. Comparison with single classifiers and traditional combiners has shown the superiority of the consensus method in terms of predictive performance. It is worth mentioning that the accuracy of traditional combiners was often found to be better than that of the best base classifier, which demonstrated the futility of blind combination of classifier outcomes.

Classifiers using traditional combination methods usually achieve accuracy worse than that of good classifiers and better than that of bad ones (medium accuracy). In contrast, consensus shows the relationship between single classifiers in terms of how the ranking of each classifier affects the others. If the majority of classifiers at a given entry make the wrong prediction, the traditional combiners have a high probability of doing the same. However, with the consensus approach, using the relationship between classifiers can result in correct predictions. Some interesting future research directions would be to 1) analyse another evaluation or estimation method of conditional rankings $R_i(\gamma_k|\Gamma_j)$ to discover any potential enhancement of the consensus procedure, 2) investigate combining homogenous classifiers

or different numbers of heterogeneous classifiers to see to what extent the consensus approach results can change, and 3) conduct a pre-processing stage for the datasets, such as feature selection or data filtering, and see how this could reflect on the consensus approach results.

References

Abellán, J., & Mantas, C. J. (2014). Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 41*(8), 3825–3830.

Altman, E. I. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance, 23*(4), 589–609.

Antonakis, A., & Sfakianakis, M. (2009). Assessing naive Bayes as a method for screening credit applicants. *Journal of Applied Statistics, 36*(5), 537–545.

Asuncion, A., & Newman, D. (2007). *UCI Machine Learning Repository,*

Atiya, A. F., & Parlos, A. G. (2000). New results on recurrent network training: Unifying the algorithms and accelerating convergence. *Neural Networks, IEEE Transactions on, 11*(3), 697–709.

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635.

Basir, O. A., & Shen, H. C. (1993). New approach for aggregating multi-sensory data. *Journal of Robotic Systems, 10*(8), 1075–1093.

Benediktsson, J. A., & Swain, P. H. (1992). Consensus theoretic classification methods. *Systems, Man and Cybernetics, IEEE Transactions on, 22*(4), 688–704.

Bishop, C. M. (2006). *Pattern recognition and machine learning*. Springer.

Briand, L. C., Freimut, B., & Vollei, F. (2004). Using multiple adaptive regression splines to support decision making in code inspections. *Journal of Systems and Software, 73*(2), 205–217.

Breiman, L. (2001). Random forests. *Machine-learning, 45*(1), 5-32.

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). *Classification and regression trees*. Wadsworth. Belmont, CA.

Canuto, A. M., Abreu, M. C., de Melo Oliveira, L., Xavier, J. C., & Santos, A. d. M. (2007). Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles. *Pattern Recognition Letters, 28*(4), 472–486.

Chitroub, S. (2010). Classifier combination and score level fusion: Concepts and practical aspects. *International Journal of Image and Data Fusion, 1*(2), 113–135.

Crook, J. N., Edelman, D. B., & Thomas, L. C. (2007). Recent developments in consumer credit risk assessment. *European Journal of Operational Research, 183*(3), 1447–1465.

De Andrés, J., Lorca, P., de Cos Juez, Francisco J., & Sánchez-Lasheras, F. (2011). Bankruptcy forecasting: A hybrid approach using fuzzy c-means clustering and multivariate adaptive regression splines (MARS). *Expert Systems with Applications, 38*(3), 1866–1875.

DeGroot, M. H. (1974). Reaching a consensus. *Journal of the American Statistical Association, 69*(345), 118–121.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *The Journal of Machine Learning Research, 7*, 1–30.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Dunn, O. J. (1961). Multiple comparisons among means. *Journal of the American Statistical Association, 56*(293), 52–64.

Finlay, S. (2011). Multiple classifier architectures and their application to credit risk assessment. *European Journal of Operational Research, 210*(2), 368–378.

Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics,* 1–67.

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics, 11*(1), 86–92.

García, V., Marqués, A. I., & Sánchez, J. S. (2015). An insight into the experimental design for credit risk and corporate bankruptcy prediction systems. *Journal of Intelligent Information Systems, 44*(1), 159–189.

Hand, D. J., & Henley, W. E. (1997). Statistical classification methods in consumer credit scoring: A review. *Journal of the Royal Statistical Society. Series A (Statistics in Society),* 523–541.

Hand, D. J. (2009). Measuring classifier performance: A coherent alternative to the area under the ROC curve. *Machine Learning, 77*(1), 103–123.

Hastie, T., Tibshirani, R., Friedman, J., & Franklin, J. (2005). The elements of statistical learning: Data mining, inference and prediction. *The Mathematical Intelligencer, 27*(2), 83–85.

Haykin, S. (1999). Adaptive filters. *Signal Processing Magazine, 6.*

Hsieh, N., & Hung, L. (2010). A data driven ensemble classifier for credit scoring analysis. *Expert Systems with Applications, 37*(1), 534–545.

Jensen, H. L. (1992). Using neural networks for credit scoring. *Managerial Finance, 18*(6), 15–26.

Kennedy, K., Mac Namee, B., & Delany, S. J. (2012). Using semi-supervised classifiers for credit scoring. *Journal of the Operational Research Society, 64*(4), 513–529.

Kumar, P. R., & Ravi, V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques – A review. *European Journal of Operational Research, 180*(1), 1–28.

Lai, K. K., Yu, L., Zhou, L., & Wang, S. (2006). Credit risk evaluation with least square support vector machine. *Rough sets and knowledge technology* (pp. 490–495). Springer.

Lessmann, S., Baesens, B., Seow, H., & Thomas, L. C. (2015). Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. *European Journal of Operational Research, 247*(1), 124–136.

Lessmann, S., Seow, H., Baesens, B., & Thomas, L. (2013). Benchmarking state-of-the-art classification algorithms for credit scoring: A ten-year update, credit scoring conference CRC, Edinburgh. *Www. Business-School. Ed. Ac. uk/crc/conferences/, Access,* 09-23.

Liang, D., Tsai, C. F., & Wu, H. T. (2015). The effect of feature selection on financial distress prediction. *Knowledge-Based Systems, 73*, 289-297.

Lin, W., Hu, Y., & Tsai, C. (2012). Machine learning in financial crisis prediction: A survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 42*(4), 421–436.

Malhotra, R., & Malhotra, D. (2003). Evaluating consumer loans using neural networks. *Omega, 31*(2), 83–96.
Marqués, A., García, V., & Sánchez, J. S. (2012). Two-level classifier ensembles for credit risk assessment. *Expert Systems with Applications, 39*(12), 10916–10922.
Nanni, L., & Lumini, A. (2009). An experimental comparison of ensemble of classifiers for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 36*(2), 3028–3033.
Pietruszkiewicz, W. (2008). Dynamical systems and nonlinear Kalman filtering applied in classification. *Cybernetic Intelligent Systems, 2008. CIS 2008. 7th IEEE International Conference on,* 1–6.
Rokach, L. (2010). Ensemble-based classifiers. *Artificial Intelligence Review, 33*(1–2), 1–39.
Sabzevari, H., Soleymani, M., & Noorbakhsh, E. (2007). A comparison between statistical and data mining methods for credit scoring in case of limited available data. *Proceedings of the 3rd CRC Credit Scoring Conference, Edinburgh, UK,*
Sánchez-Lasheras, F., de Andrés, J., Lorca, P., & de Cos Juez, Francisco J. (2012). A hybrid device for the solution of sampling bias problems in the forecasting of firms' bankruptcy. *Expert Systems with Applications, 39*(8), 7512–7523.
Shaban, K., Basir, O., Kamel, M., & Hassanein, K. (2002). Intelligent information fusion approach in cooperative multiagent systems. *Automation Congress, 2002 Proceedings of the 5th Biannual World, 13*, 429–434.
Suen, C. Y., & Lam, L. (2000). Multiple classifier combination methodologies for different output levels. *Multiple classifier systems* (pp. 52–66). Springer.
Sun, J., Li, H., Huang, Q. H., & He, K. Y. (2014). Predicting financial distress and corporate failure: A review from the state-of-the-art definitions, modeling, sampling, and featuring approaches. *Knowledge-Based Systems*, *57*, 41-56.
Tan, M. (1993). Multi-agent reinforcement learning: Independent vs. cooperative agents. *Proceedings of the Tenth International Conference on Machine Learning*, 330–337.
Teng, S., Du, H., Wu, N., Zhang, W., & Su, J. (2010). A cooperative network intrusion detection based on fuzzy SVMs. *Journal of Networks, 5*(4), 475–483.
Toh, K., Tran, Q., & Srinivasan, D. (2007). Hyperbolic function networks for pattern classification. *Trends in neural computation* (pp. 1–33). Springer.
Toh, K., & Yau, W. (2004). Combination of hyperbolic functions for multimodal biometrics data fusion. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on, 34*(2), 1196–1209.
Tsai, C. (2014). Combining cluster analysis with classifier ensembles to predict financial distress. *Information Fusion, 16*, 46–58.
Tsai, C., & Wu, J. (2008). Using neural network ensembles for bankruptcy prediction and credit scoring. *Expert Systems with Applications, 34*(4), 2639–2649.
Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert Systems with Applications, 38*(1), 223–230.
Wang, G., & Ma, J. (2012). A hybrid ensemble approach for enterprise credit risk assessment based on support vector machine. *Expert Systems with Applications, 39*(5), 5325–5331.
Wang, G., Ma, J., Huang, L., & Xu, K. (2012). Two credit scoring models based on dual strategy ensemble trees. *Knowledge-Based Systems, 26*, 61–68.
West, D., Dellana, S., & Qian, J. (2005). Neural network ensemble strategies for financial decision applications. *Computers & Operations Research, 32*(10), 2543–2559.
Xu, L., Krzyżak, A., & Suen, C. Y. (1992). Methods of combining multiple classifiers and their applications to handwriting recognition. *Systems, Man and Cybernetics, IEEE Transactions on, 22*(3), 418–435.
Yu, L., Wang, S., & Lai, K. K. (2008). Credit risk assessment with a multistage neural network ensemble learning approach. *Expert Systems with Applications, 34*(2), 1434–1444.
Yu, L., Wang, S., & Lai, K. K. (2009). An intelligent-agent-based fuzzy group decision making model for financial multicriteria decision support: The case of credit scoring. *European Journal of Operational Research, 195*(3), 942–959.
Yu, L., Yue, W., Wang, S., & Lai, K. K. (2010). Support vector machine based multiagent ensemble learning for credit risk evaluation. *Expert Systems with Applications, 37*(2), 1351–1360.
Van Gestel, I. T., Baesens, B., Garcia, I. J., & Van Dijcke, P. (2003, January). A support vector machine approach to credit scoring. In *FORUM FINANCIER-REVUE BANCAIRE ET FINANCIAIRE BANK EN FINANCIEWEZEN*- UNKNOWN, 73-82.
Verikas, A., Gelzinis, A., & Bacauskiene, M. (2011). Mining data with random forests: A survey and results of new tests. *Pattern Recognition, 44*(2), 330-349.
Zang, W., Zhang, P., Zhou, C., & Guo, L. (2014). Comparative study between incremental and ensemble learning on data streams: Case study. *Journal of Big Data, 1*(1), 1–16.
Zhang, C., & Duin, R. P. (2011). An experimental study of one-and two-level classifier fusion for different sample sizes. *Pattern Recognition Letters, 32*(14), 1756–1767.
Zhang, D., Zhou, X., Leung, S. C., & Zheng, J. (2010). Vertical bagging decision trees model for credit scoring. *Expert Systems with Applications, 37*(12), 7838–7843.
Zhou, L. (2013). Performance of corporate bankruptcy prediction models on imbalanced dataset: The effect of sampling methods. *Knowledge-Based Systems*, *41*, 16-25.
Zhou, L., Lai, K. K., & Yu, L. (2010). Least squares support vector machines ensemble models for credit scoring. *Expert Systems with Applications, 37*(1), 127–133.
Zhou, L., Lu, D., & Fujita, H. (2015). The performance of corporate financial distress prediction models with features selection guided by domain knowledge and data mining approaches. *Knowledge-Based Systems*, *85*, 52-61.
Zhou, L., Tam, K. P., & Fujita, H. (2016). Predicting the listing status of Chinese listed companies with multi-class classification models. *Information Sciences*, *328*, 222-236.