Progress in Artificial Intelligence
https://doi.org/10.1007/s13748-021-00246-2

REGULAR PAPER

Credit scoring by leveraging an ensemble stochastic criterion in a transformed feature space

Salvatore Carta[1] · Anselmo Ferreira[1] · Diego Reforgiato Recupero[1] · Roberto Saia[1]

Received: 22 October 2020 / Accepted: 7 May 2021

Abstract
The credit scoring models are aimed to assess the capability of refunding a loan by assessing user reliability in several financial contexts, representing a crucial instrument for a large number of financial operators such as banks. Literature solutions offer many approaches designed to evaluate users' reliability on the basis of information about them, but they share some well-known problems that reduce their performance, such as data imbalance and heterogeneity. In order to face these problems, this paper introduces an ensemble stochastic criterion that operates in a discretized feature space, extended with some meta-features in order to perform efficient credit scoring. Such an approach uses several classification algorithms in such a way that the final classification is obtained by a stochastic criterion applied to a new feature space, obtained by a twofold preprocessing technique. We validated the proposed approach by using real-world datasets with different data imbalance configurations, and the obtained results show that it outperforms some state-of-the-art solutions.

Keywords Credit scoring · Stochastic processes · Ensemble learning · Machine learning · Transformed feature space · Discretization · Meta-features · Heterogeneity · Algorithms

1 Introduction

The significant increase in requests related to consumer credit has made it impossible for credit institutions to use manual approaches to assess the solvency of applicants. Credit Scoring systems [50] are therefore computer-aided statistical approaches proposed to deal with this issue. Such models evaluate the probability that a new instance (*i.e.*, a potential client) is considered reliable (non-default) or unreliable (default), by searching for similarities of that instance with clustered samples that the system learned with previous labeled data. For this reason, credit scoring approaches through machine learning techniques represent the only possible solution nowadays, as they carry out this operation efficiently without any human supervision [61]. Thus, such credit scoring systems offer a great opportunity to financial operators, since they allow the evaluation of a huge number of requests.

To further highlight the importance of efficient credit scoring models, we show in Figs. 1 and 2 a study about consumer credit and spending in the *Eurozone* made by *Trading Economics*[1], which is based on the information provided by the *European Central Bank*[2]. It shows that the consumer loan increment follows the spending (all data are expressed in billions of euros); therefore, it clearly underlines how such requests boosted dramatically over the last years, with this trend being similar to that registered in other world zones such as the *USA* and *Russia*. An effective credit scoring approach must, therefore, evaluate accurately the probability that a user will not (fully or partially) repay a loan, and this type of information is used in order to decide whether to grant or refuse a requested credit, minimizing income losses to the financial operators. The effectiveness of such models is particularly important specially at times of crisis,

✉ Roberto Saia
roberto.saia@unica.it

Salvatore Carta
salvatore@unica.it

Anselmo Ferreira
anselmo.ferreira@gmail.com

Diego Reforgiato Recupero
diego.reforgiato@unica.it

[1] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, 09124 Cagliari, Italy

[1] https://tradingeconomics.com/euro-area/.

[2] https://www.ecb.europa.eu.

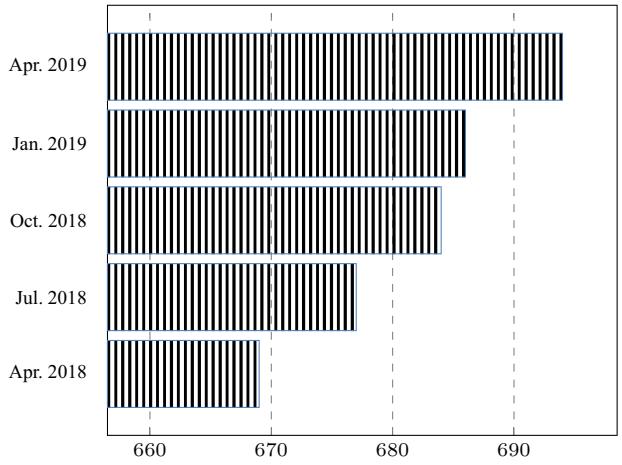

Fig. 1 Consumer credit in the Eurozone (in billions of Euros)

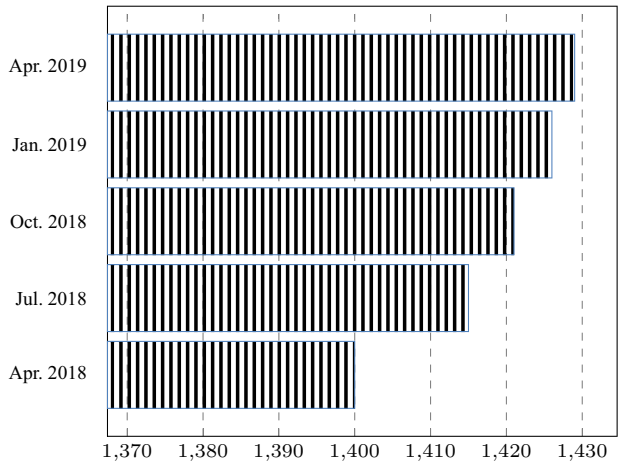

Fig. 2 Consumer spending in the Eurozone (in billions of Euros)

such as when natural disasters and epidemics happen and the market becomes challenging by nature. Notwithstanding, it should be observed how the performance of these techniques is directly affected by misclassification errors [8].

The definition of an evaluation model able to perform effective users rating or classification is not an easy task for a series of limitations, such as the *data heterogeneity* [16], characterized by the information in credit scoring datasets with different semantics and range of values, making it difficult to use them as features to perform credit scoring. Another common issue, considered the most important in credit scoring datasets, is the *data imbalance* [10,41]. This means that the available data about users (from now on denoted as *instances*) are composed of a few examples of *unreliable* cases when compared with the *reliable* ones, preventing the definition of effective evaluation of credit scoring models.

In this paper, we present a credit scoring model that increases credit scoring performance by tackling these above-mentioned limitations in two distinct, but complementary manners: (1) efficient data preprocessing and (2) ensemble of machine learning approaches ruled by a stochastic criterion. As first step, our proposal transforms the feature space through complementary two steps of features discretization and enrichment. The discretization process scales the features, allowing us to reduce data heterogeneity by merging similar patterns. In addition, the enrichment process reduces the loss of information from the previous step by calculating additional meta-information to improve the different characterizations of *reliable* and *unreliable* cases. Additionally, such a transformed feature space is used in the context of an ensemble of classifiers, ruled by a stochastic criterion that minimizes reliable samples misclassification and maximizes unreliable samples classification at the same time, tackling in this way the data imbalance problem. In order to avoid over-fitting issues [37] in our experiments, we evaluate the proposed approach by firstly performing a twofold strategy of data splitting. In more detail, we divide the datasets into an *in-sample* part, which is used for training of the evaluation model, and an *out-of-sample* part, used for final performance evaluation through the *k-fold cross-validation* criterion. We justify such an experimental procedure because the canonical *k-fold cross-validation* criterion for this problem does not allow us obtaining a real separation between the data used to train and assess the evaluation model performance. Such a criterion has been largely used in other domains, such as, for instance, financial market forecasting [38]. In both applications, it is crucial to assess the real effectiveness of a prediction model on data never seen before.

In summary, the main scientific contributions related to this paper are the following:

- exploitation of a data preprocessing approach aimed to improve the performance of the proposed ensemble classifier;
- definition of an ensemble of classifiers ruled by a stochastic criterion and trained in a transformed feature space;
- validation of the proposed approach made by using both the *in-sample/out-of-sample* and the k-fold cross-validation criteria.

The remaining of this paper is organized as follows: We discuss the related work and challenges of this research in Sect. 2. In Sect. 3, we present our proposed approach. Section 4 reports the experimental configuration and results, and finally, Sect. 5 concludes this work and points out future research directions where we are headed to.

2 Related work

The literature presents three different credit scoring models [24] regarding the concept of non-reliable, or *default* cases:

(1) *Probability of Default*, which evaluates the likelihood of a default over a specific time period; (2) *Exposure at Default*, which evaluates the financial exposure of an investor when a loan defaults; and (3) *Loss Given Default*, which evaluates the loss of money of a financial operator when a loan defaults. For the purposes of this paper, we consider only the *Probability of Default* model, which we implement according to a binary classification criterion (*i.e.*, by classifying each new instance as *reliable* or *unreliable*). Different approaches and strategies have been exploited in order to define effective credit scoring approaches. We discuss the most important ones in the following paragraphs.

The first branch of techniques treats the credit scoring problem as a data analysis or data transformation problem, discriminating reliable and unreliable samples by investigating data disturbance or analyzing new transformed spaces. The work of Carta *et al.* [13,54] followed that direction by investigating data entropy before and after an unknown sample is inserted in a dataset in order to measure how it is affected, with this information being helpful to detect default (or unreliable) cases. Fan *et al.* [30] also exploited the entropy criterion in order to face the issues related to imbalanced datasets. Saia and Carta [56] compared features magnitudes in the Fourier space in a test sample and all samples from a dataset. Saia *et al.* [57] analyzed the cosine, features and magnitudes similarities in the Wavelet transform. Similarly, Jaber *et al.* [39] proposed a wavelet-inspired analysis in order to convert the original data into a time-scale domain.

Another branch of credit scoring techniques, which is more related to our present work, considers machine learning classifiers in the credit scoring pipeline. For example, Chen *et al.* [19] used the support vector machine (SVM) classifier in order to define a new scoring process, based on historical data on a proprietary dataset. The work of Fan *et al.* [29] considered the same SVM classifier for this task, but optimized by an adaptive mutation partial swarm algorithm. Li *et al.* [44] considered two scenarios to perform credit scoring using a Bayesian optimal filter and a recursive Bayes estimator, whereas Chen *et al.* [18] proposed an advanced Bayesian algorithm for credit assessment. Zhang *et al.* [68] defined a novel random forests (RFs)-based classifier for credit scoring, using feature selection through information entropy and grid search. Damrongsakmethee *et al.* [25] presented a feature selection approach, combining principal component analysis and the ReliefF algorithm, using preprocessed data through these techniques in a decision tree classifier. Arora and Kaur [4] introduced the Bootstrap-Lasso feature selection algorithm, which selects consistent and relevant features from a pool of features. Such an approach is then validated in classification algorithms like random forest, support vector machines, Naive Bayes and K-nearest neighbors.

Other works have considered more efficient machine learning solutions, such as neural networks, to perform credit scoring. Chen *et al.* [21] presented the use of the DeepGBM network, which deals with sparse categorical features and dense numerical features at the same time. Changjian and Peng [15] used an improved Elman neural network to perform credit scoring with particle swarm optimization to initialize network weights. Wang *et al.* [63] considered long short-term memory neural networks with an attention mechanism to perform peer-to-peer credit scoring. Fonseca *et al.* [32] used a two-stage process, involving a fuzzy inference model as input for an artificial neural network. Babaev *et al.* [5] used a recurrent neural network in the pipeline, treating the credit scoring task as a text classification task.

Machine learning techniques can also be combined in order to build hybrid approaches of credit scoring decision support systems as, for instance, the one presented by Feng *et al.* [64], which exploits a two-stage hybrid model with artificial neural networks and a multivariate adaptive regression splines model. Another kind of classifiers combination, commonly known as *ensembles* [27], has also been extensively studied in the literature. The work of Lopez *et al.* [48] used an ensemble of several classifiers, including SVMs and logistic regression, in order to validate a feature selection strategy called *group penalty function*, which penalizes the use of variables from the same source of information in the final features. The work of Zhang *et al.* [67] ensembles five classifiers (logistic regression, support vector machine, neural network, gradient boosting decision tree and random forest) using a genetic algorithm and fuzzy assignment. In the work of Feng *et al.* [31], a set of classifiers are joined in an ensemble according to their soft probabilities. Finally, Tripathi *et al.* [62] proposed a model based on dimensionality reduction and layered ensemble classification with weighted voting on the best five out of seven classification algorithms. Pławiak *et al.* [52] exploited a novel deep genetic cascade ensemble of SVM classifiers to perform credit scoring. The work of Abellan and Castellano [2] presented the findings of a very deep comparative work on ensembling machine learning models for credit scoring. Among their conclusions, they found out that, when employing decision trees as base classifiers, the ensemble schemes have shown to perform better, even though such a classifier does not obtain good results individually.

In addition, the literature offers other hybrid approaches, where different techniques/strategies (e.g., data optimization techniques, genetic algorithms, fuzzy logic, etc.) have been combined in order to improve the credit scoring performance. Some representative examples are the work of Santana *et al.* [59], where the authors combine fuzzy logic, neural networks and a variable population optimization technique, to obtain fuzzy classification rules, and another work from the same authors [40], where they define a method able to reduce the number of classification rules involved in the definition

of the credit scoring predictive model, reducing the system decision time. Other representative works are that of Carta *et al.* [12], where the authors adopt a two-step feature space transforming method, with the aim to improve the credit scoring performance, or the work of Zhang *et al.* [69], where the authors propose a novel sparse multi-criteria optimization classifier based on one-norm regularization, linear and nonlinear programming, for the credit risk evaluation.

Another interesting category of approaches is those that exploit, alone or in combination with other methods, the Auto-Regressive Integrated Moving Average (ARIMA) model, which is largely used in many contexts (e.g., in the e-commerce price forecasting one, as in the work of Carta *et al.* [14], or in the time series forecasting one, as in the work of Domingos *et al.* [28]), in order to perform credit scoring tasks. An example can be found in the work of Jaber *et al.* [39], where this statistical model has been combined with wavelet functions.

It should be observed that most of the presented approaches share some well-known problems that reduce their effectiveness in the credit scoring domain. Two important issues commonly found in credit scoring datasets are: (1) *Class Imbalance*, given by the strong difference in the number of samples related to the *reliable* and *unreliable* cases [11]. Such a limitation reduces the effectiveness of the classification approaches, since they should have a balanced number of samples in order to define a reliable classification model [17], and (2) *Data Heterogeneity*, given by the same information being represented differently in the datasets [16]. Our approach, which will be presented later in this paper, deals with both problems by transforming data and applying stochastic ensemble decisions, maximizing ensemble performance even in an imbalanced scenario.

2.1 Data preprocessing

About discretization approaches, on which part of our proposed approach is based, the literature usually treats them as a preprocessing strategy, aimed to improve the classification algorithms performance. Such technique has been discussed in depth by Kotsiantis *et al.* [42] in their survey, whereas some of its recent developments are taken into account in the work of Börner *et al.* [42]. It works by transforming the feature values from their original quantitative form into a qualitative form, dividing each value according to a discrete number of non-overlapped intervals. In other words, each original continuous or discrete number is mapped into one of such intervals, by following different criteria. For instance, De Sá *et al.* [26] use an entropy-based discretization approach to perform this operation, whereas Sharmin *et al.* [60] propose an approach based on mutual information.

Regardless the adopted discretization criterion, in addition to the improvement in performance that is usually achieved, this process presents further advantages, such as reduction of data dimensionality that produces a faster and accurate learning, as discussed by García *et al.* [33] in the context of big data. Another advantage is related to the better data understandability reached through the discretization process, as discussed in the Luengo *et al.* [47] work. The main disadvantage of a discretization process is related to the loss of information that it yields, since an ideal data discretization represents an *NP-complete*[3] problem.

The addition of meta-features is also exploited in our proposed approach, which the literature classifies as a data enrichment technique. This process, aimed to improve the original data domain by adding additional information, has been discussed by Bilalli *et al.* [9], which is focused on the predictive power of meta-features. The definition of the meta-features is performed by following several and different criteria, such as, for instance, calculating them on the basis of a metric in the context of each single instance of the dataset, or by taking into account the entire dataset. As detailed during the description of the proposed approach, such a technique will be used in order to mitigate the loss of information related to the discretization process we performed.

2.2 Evaluation metrics

The literature indicates several types of metrics able to assess the performance of the credit scoring approaches/strategies. Chen *et al.* [20] discuss them in this domain, as well as Zou *et al.* [70], which focused this evaluation in the context of the classification with an imbalanced class distribution, which usually characterizes the credit scoring scenario.

There are metrics based on the confusion matrix [65], which is a 2×2 matrix that contains the total number of True Negatives (TN), False Negatives (FN), True Positives (TP), and False Positive (FP) as shown in Table 1, where the matrix is contextualized in the credit scoring scenario. Some examples of confusion-matrix-based metrics are the $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, the $Sensitivity = \frac{TP}{TP+FN}$, the $Specificity = \frac{TN}{TN+FP}$ and the $Fallout = \frac{FP}{FP+TN}$. Other metrics widely used in this domain are those based on the the receiver operating characteristic (ROC) curve, such as the area under the ROC curve (AUROC) [2]. The ROC curve plots the *Sensitivity* against the *Fallout*, placing the former on the y-axis and the latter on the x-axis. The ROC is a probability curve, and the AUROC (the area under this curve) gives us information about the capability (separability measure) of a binary classifier to discriminate the two different classes of information (in our context, reliable and unreliable), correctly. The confusion-matrix-based metrics and the

[3] According to the computational complexity theory, a NP-complete problem occurs when its solution needs a restricted class of brute force search algorithms, where NP stands for non-deterministic polynomial.

Table 1 Confusion matrix

		Performed classification	
		Reliable	Unreliable
Real class	*Reliable*	TP	FN
	Unreliable	FP	TN

ROC-based ones are often combined in order to provide a more reliable evaluation of the credit scoring performance.

3 Proposed method

Before discussing our approach, we introduce the adopted formal notation. Given $I = \{i_1, i_2, \ldots, i_X\}$ a set of classified instances, $I^+ = \{i_1^+, i_2^+, \ldots, i_Y^+\}$ a subset of *reliable* instances with $I^+ \subseteq I$, $I^- = \{i_1^-, i_2^-, \ldots, i_W^-\}$ a subset of *unreliable* instances with $I^- \subseteq I$, $\hat{I} = \{\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_Z\}$ a set of unclassified instances, $F = \{f_1, f_2, \ldots, f_N\}$ a set of instance features, $C = \{reliable, unreliable\}$ a set of instance classifications, $A = \{a_1, a_2, \ldots, a_Z\}$ a set of classification algorithms and $P = \{p_1, p_2, \ldots, p_Z\}$ the predictions for a generic $a_i \in A$ with $Z = |P| = |A|$ (*i.e.,* the number of predictors used), the problem faced in this paper is formalized in Eq. 1 as follows:

$$\max_{0 \leq \Theta \leq |\hat{I}|} \Theta = \sum_{j=1}^{|\hat{I}|} eval(\hat{i}_j, I), \tag{1}$$

where the classification of each $\hat{i}$ instance is performed by using the $eval(\hat{i}, I)$ function on the basis of the information in the set I, which gives as output a β binary value (0=*misclassification*, 1=*correct classification*).

Our objective is therefore maximizing Θ, since it represents the sum of the correct classifications. According to the high-level architecture of Fig. 3, the proposed approach has been implemented through the steps described in the following subsections.

3.1 Step #1: feature space transform.

Each feature $f \in F$ that composes the sets I and $\hat{I}$, being continuous or discrete, is discretized in the first step of our approach. In more detail, their original values have been mapped into a discrete range of values $\{0, 1, \ldots, \Delta\} \in \mathbb{Z}$, according to a Δ parameter found through experiments (which will be described later in Sect. 4 of this paper).

More formally, denoting $f \xrightarrow{\Delta} d$ the discretization process, we map each single feature value $f \in F$ into one of the discrete integer values in $\{d_1, d_2, \ldots, d_\Delta\}$. Such a process leads toward a reduction of instance patterns, merging similar patterns as formalized in Eq. 2 as follows.

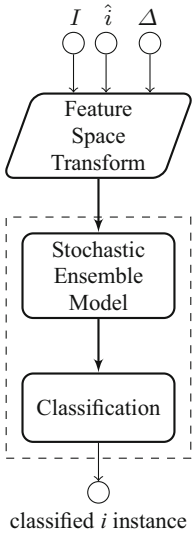

Fig. 3 High-level architecture of our proposed approach

$$\begin{matrix} \{f_1,\ f_2,\ \ldots,\ f_N\} \xrightarrow{\Delta} \{d_1, d_2, \ldots, d_N\}, & \forall\ i \in I \\ \{f_1,\ f_2,\ \ldots,\ f_N\} \xrightarrow{\Delta} \{d_1, d_2, \ldots, d_N\}, & \forall\ \hat{i} \in \hat{I} \end{matrix}. \tag{2}$$

Subsequently, we process each discretized vector of features $\{d_1, d_2, \ldots, d_\Delta\}$ in the sets I and $\hat{I}$ by adding a series of meta-features μ. Such meta-features have been calculated in the new discretized space, and they are the *minimum* (m), *maximum* (M), *average* (A) and the *standard deviation* (S). The result of such operations is four new features for each vector (*i.e.*, $\mu = \{m, M, A, S\}$), which face through an improved characterization of each instance the loss of information related to the previous discretization process. The new meta-features are calculated as shown in Eq. 3.

$$\mu = \begin{cases} m = min(d_1, d_2, \ldots, d_N) \\ M = max(d_1, d_2, \ldots, d_N) \\ A = \frac{1}{N} \sum_{n=1}^{N} (d_n) \\ S = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (d_n - \bar{d})^2} \end{cases} \tag{3}$$

The new feature space ϕ represents the transformation of the original feature space according to a Δ value of discretization, which has been followed by adding four new meta-features $\mu = \{m, M, A, S\}$. To simplify, we show in Eq. 4 such an operation only in the set I (which is the same for the set $\hat{I}$).

$$\phi(I) = \begin{pmatrix} d_{1,1} & d_{1,2} & \ldots & d_{1,N} & m_{1,N+1} & M_{1,N+2} & A_{1,N+3} & S_{1,N+4} \\ d_{2,1} & d_{2,2} & \ldots & d_{2,N} & m_{2,N+1} & M_{2,N+2} & A_{2,N+3} & S_{2,N+4} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{X,1} & d_{X,2} & \ldots & d_{X,N} & m_{X,N+1} & M_{X,N+2} & A_{X,N+3} & S_{X,N+4} \end{pmatrix}. \tag{4}$$

With such steps of data transform, we tackle data heterogeneity present in credit scoring datasets, putting different users information in the same transformed feature space.

3.2 Step #2: Stochastic ensemble model.

As discussed in the previous section, ensemble machine learning methods were largely been used in the literature, with the aim to improve single classifiers performance. Ensembling machine learning algorithms means that, instead of using a single classification algorithm, several algorithms are taken into account to classify an event, and such a classification depends on all the single results according to a certain criterion (*e.g.*, full agreement, majority voting, weighted voting, among others). Although an ensemble method commonly improves the single used algorithms, it should be observed how this could be not true in certain cases [34]. An ensemble configuration can use a *dependent* or *independent* model [53]. In the first case, the result of an algorithm depends on the result of the others, whereas, in the second case, the result of an algorithm is independent of the results of the others. The approach proposed in this paper is an *independent* ensemble model.

The stochastic model that we used to drive the ensemble approach classification relies on probabilities models, since such models are able to evaluate a binary response given by a series of independent predictors in terms of probability. In more detail, we use these models in order to assess the probability (or confidence) that an instance $\hat{i} \in \hat{I}$ belongs to one of the classes in C, by mapping the related probabilities through a sigmoid σ function[4]. This function is largely used in machine learning to connect predictions to probabilities, mapping any real value into a binary value [0,1]. More formally, Eq. 5 shows this process

$$\sigma(a_z(p)) = \frac{1}{1+e^{-p}}, \tag{5}$$

where the probability estimated for a prediction p, performed by using a machine learning algorithm a_z and expressed in the range [0, 1] is denoted by $\sigma(a_z(p))$, and e represents the base of natural log. Such a probability can be calculated, in the credit scoring context, for both *reliable* (σ_r) and *unreliable* (σ_u classes), where $\sigma_r = 1 - \sigma_u$.

On the basis of Eq. 5, our proposed ensemble model operates by excluding from the decision process the algorithms that make predictions with a low level of probability, performing the ensemble classification on the basis of the predictions from stronger algorithms, according to the following weighted probabilistic criterion.

$$\begin{aligned}
&\overline{\sigma_r} = \tfrac{1}{Z} \cdot \sum_{z=1}^{Z} \sigma_r(a_z(p)) \\
&w_1 = w_1 + 1 \; if \; \sigma_r(a_z(p)) > \overline{\sigma_r} \; \wedge \; a_z(p) = \textit{reliable} \\
&w_2 = w_2 + 1 \; if \; \sigma_r(a_z(p)) < \overline{\sigma_r} \; \vee \; a_z(p) = \textit{unreliable} \\
&c(\hat{i}) = \begin{cases} \textit{reliable}, & \text{if } w_1 > w_2 \\ \textit{unreliable}, & \textit{otherwise}. \end{cases}
\end{aligned} \tag{6}$$

The first step of the proposed stochastic ensemble approach is, therefore, defining what is a low level of probabilities. The approach does this sample-wise, by calculating the mean probability $\overline{\sigma_r}$ that a given test sample is classified as *reliable* in a set of classifiers in the ensemble Z. Then, the approach starts weighting the classes in order to avoid misclassification of reliable samples (the class more frequent in credit scoring datasets) and, at the same time, maximizing the classification of the less frequent classes (unreliable samples).

To perform such a task, the proposed ensemble criterion requires the comparison of two variables w_1 and w_2, as detailed in Eq. 6. Both variables are initialized to zero and are updated for each sample depending on the classification of that sample for the different classifiers of the ensemble. The first assumption, denoted as w_1, is activated when a classifier confidence that the sample is reliable is higher than the mean confidence $\overline{\sigma_r}$ considering all the classifiers in that ensemble. As in credit scoring datasets the reliable samples are more present, we need a strong confidence to activate such assumption as classifiers had more data to train for that particular class. Therefore, w_1 is considered our first step to tackle data imbalance, which is stating that most of classifiers in an ensemble need higher confidence on classifying the more frequent samples. Such requirement reduces misclassifications for that particular class.

The second assumption of our stochastic ensemble criterion, denoted as w_2, deals with unreliable samples and states that the classifier decision of unreliable is always respected, no matter how high σ_u is. Additionally, w_2 is activated if w_1 assumption is false. Therefore, the second step in dealing with data imbalance is respecting the classification of unreliable samples and being more rigorous regarding reliable samples classification (w_1), increasing therefore the classification of unreliable samples. In the end, our approach compares the sum of activations of w_1 and w_2 to make a final decision, dealing with data imbalance issues in the ensemble criterion itself, not working with the data directly.

3.3 Step #3: classification.

The new feature space and the stochastic model previously formalized have been used in the context of classification in Algorithm 1, with the aim to classify each instance $\hat{i} \in \hat{I}$. It takes as input the set of classification algorithms Z, the

[4] A mathematical function has a characteristic *sigmoid curve*.

set of classified instances I and the instance to evaluate $\hat{i}$, returning as output the classification (*reliable* or *unreliable*) of the instance $\hat{i}$. The algorithm starts by discretizing and enriching features in lines 4 and 5, respectively. Then, the training of the models of the ensemble in the transformed space is started in line 6, and the testing of an unevaluated (testing) sample is done in line 7. Line 8 of the algorithm calculates the mean probability of reliable in all the models, and lines 9-15 calculate the stochastic ensemble result for each class according to Eq. 6 requirements. Finally, lines 16-21 work by checking the class with highest weight, and finally, the final classification for that sample is returned.

Algorithm 1 Instance classification

Require: Z=Ensemble algorithms set, I=Classified instances set, $\hat{i}$=Unevaluated instance
Ensure: c=Classification of the $\hat{i}$ instance
1: **procedure** CLASSIFICATION(A, I, $\hat{i}$)
2: $w_1 \leftarrow 0$
3: $w_2 \leftarrow 0$
4: $I \leftarrow getTransformedFeatureSpace(I)$
5: $\hat{i} \leftarrow getTransformedFeatureSpace(\hat{i})$
6: $models = trainModels(Z, I)$
7: $predictions = getPredictions(models, \hat{i})$
8: $\overline{\sigma} \leftarrow getMeanPredictionsProbability(predictions)$
9: **for each** p **in** $predictions$ **do**
10: **if** $getProbability(p) > \overline{\sigma} \ \wedge \ p == reliable$ **then**
11: $w_1 \leftarrow w_1 + 1$
12: **else**
13: $w_2 \leftarrow w_2 + 1$
14: **end if**
15: **end for**
16: **if** $w_1 > w_2$ **then**
17: $c \leftarrow reliable$
18: **else**
19: $c \leftarrow unreliable$
20: **end if**
21: **return** c
22: **end procedure**

3.3.1 Computational complexity analysis

Without claiming to carry out an exhaustive and detailed analysis of the computational complexity of the proposed Algorithm 1, in this section we provide some indication with regard to its complexity in the context of the most used state-of-the-art algorithms and available real-world datasets in this domain.

It has been performed by analyzing the theoretical complexity (adopting the Big $\mathcal{O}$ notation [22], which defines the algorithm upper bound, bounding its complexity only from above) of the instance classification and the data features transformation we applied, without taking into account aspects such as the number of features, the number of algorithms and their different complexities during the training phase, considering in these contexts $\mathcal{O}(n)$ as worst case, according to the most common algorithms and datasets scenarios in the credit scoring literature.

In light of the above, generalizing the total number of instances to process as n, we can made the following consideration about the algorithm Big $\mathcal{O}$ complexity:

- The steps *2* and *3* refer to two operations of assignment performed in constant time, and the final complexity is then $\mathcal{O}(1)$;
- the steps *4* and *5* are related to the proposed data transformation (*i.e.*, discretization and meta-features addition) that are applied, respectively, on each instance of the dataset I and an unevaluated instance $\hat{i}$. These operations have a required running time that increases linearly with the number of instances, which in the worst case leads to a $\mathcal{O}(n)$ complexity;
- the steps *6* and *7* refer to the training and classification phases of the five machine learning algorithms that compose the ensemble, whose complexity in the worst case is $\mathcal{O}(n)$;
- the step *8* performs a mean value calculation in a time that increases linearly with the number of instances, then with an $\mathcal{O}(n)$ complexity;
- the steps *9* to *15* classify the $\hat{i}$ instance on the basis of the ensemble algorithms predictions, involving only comparison and assignment operations, since the prediction probability information (step *10*) is obtained at step *7*, and then, the worst-case complexity is $\mathcal{O}(1)$;
- the steps from *16* to *21* are related to an if-statement and the return of the algorithm, both characterized by an $\mathcal{O}(1)$ complexity, and then, the final complexity is $\mathcal{O}(1)$;
- on the basis of the aforementioned considerations, made according to the Big $\mathcal{O}$ notation that defines the upper bound complexity, the complexity related to Algorithm 1 is $\mathcal{O}(n)$.

It should be observed that the computational load of the proposed Algorithm 1 can be tackled and reduced by parallelizing the process by using large-scale distributed computing models, such as, for instance, the Hadoop MapReduce [36] or the Apache Spark [49] frameworks.

4 Experiments

In this section, we evaluate the benefits of our proposed approach in an experimental scenario that involves the use of real-world data. For that, we start firstly describing the experimental setup we use and then we report results in the two following subsections.

Considering that the proposed approach is based on the preliminary transformation of the feature space, and subsequently on the exploitation of an ensemble of algorithms for the classification task, we validate it using as competitors the most widespread and effective state-of-the-art baseline

Table 2 Datasets characteristics

Dataset name	Total instances	Reliable instances	Unreliable instances	Feature number
AC	690	307	383	15
GC	1,000	700	300	21
DC	30,000	23,364	6,636	24

Table 3 AC dataset features

	Feature		Feature
01	Categorical feature	08	Categorical feature
02	Continuous feature	09	Categorical feature
03	Continuous feature	10	Continuous feature
04	Categorical feature	11	Categorical feature
05	Categorical feature	12	Categorical feature
06	Categorical feature	13	Continuous feature
07	Continuous feature	14	Continuous feature

algorithms used in this domain, both in single and in ensemble configuration. This choice derives from the consideration that the benefits measured in this context will be also measurable in the context of other more sophisticated and complex approaches that include such algorithms.

4.1 Experimental setup

We now focus our attention on explaining the experimental setup chosen to validate our proposed approaches against the baselines. We report in the following subsections the datasets, metrics, the methodology considered for the experiments and implementation details of our proposed approach.

4.1.1 Datasets

The validation of the proposed approach has been performed by using three real-world datasets that are publicly available[5] and widely used in the literature. They are the *Australian Credit Approval* (AC), the *German Credit* (GC) and the *Default of Credit Card Clients* (DC) datasets. Such datasets, whose characteristics are summarized in Table 2, allow us to evaluate the approach performance on different real-world credit scoring scenarios.

The AC dataset contains 690 instances composed by 307 reliable cases and 387 unreliable ones. Each instance is characterized by 14 features described in Table 3, and a label of those features characterizing them as being from a reliable or unreliable instance. For confidentiality reasons, all the feature names and all the values in this dataset have been modified to meaningless symbols.

[5] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/statlog/.

Table 4 GC dataset features

	Feature		Feature
01	Status of checking account	11	Present residence since
02	Duration	12	Property
03	History of credit	13	Age
04	Purpose	14	Other installment plans
05	Amount of credit	15	Housing
06	Savings account/bonds	16	Other credits
07	Current employment since	17	Job
08	Installment rate	18	Maintained people
09	Status and gender	19	Telephone
10	Other debtors/guarantors	20	Foreign worker

The GC dataset contains 1000 instances composed by 700 reliable cases and 300 unreliable ones. Each instance is characterized by 20 features described in Table 4, and a label of those features characterizing them as being from a reliable or unreliable instance. The dataset version we adopted is that composed by only numerical attributes, provided by the University of Strathclyde, Glasgow[6].

The DC dataset contains 30,000 instances, composed by 23,364 reliable cases and 6636 unreliable ones. Each instance is characterized by 23 features described in Table 5, together with a label that characterizes them as from a reliable or unreliable instance.

4.1.2 Metrics

To evaluate the performance of the considered algorithms for credit scoring, we selected some of the metrics discussed in Sect. 2.2. We discuss them as follows:

Sensitivity: it provides information on the *true positive rate*, evaluating the capability of a credit scoring approach to classify *unreliable* instances correctly [7]. It is formalized in Eq. 7, where *TP* and *FN* are, respectively, the number of instances correctly classified as *unreliable* and incorrectly classified as *reliable*.

$$Sensitivity(\hat{I}) = \frac{TP}{(TP + FN)}. \tag{7}$$

Area Under the Receiver Operating Characteristic curve (AUROC): it is a metric able to assess the predictive capability of an evaluation model even in the presence of imbalanced data. The literature indicates it as a reliable metric to evaluate the performance of a *credit scoring* model [2]. The receiver operating characteristic (ROC) curve is firstly built by plotting the true positive rate (also known as sensitivity) and the false positive rate (also known as fallout) at different clas-

[6] https://www.strath.ac.uk/.

Table 5 DC dataset features

	Feature		Feature
01	Credit amount	13	Bill statement August 2005
02	Gender	14	Bill statement July 2005
03	Education	15	Bill statement June 2005
04	Marital status	16	Bill statement May 2005
05	Age	17	Bill statement April 2005
06	Repayments September 2005	18	Amount paid September 2005
07	Repayments August 2005	19	Amount paid August 2005
08	Repayments July 2005	20	Amount paid July 2005
09	Repayments June 2005	21	Amount paid June 2005
10	Repayments May 2005	22	Amount paid May 2005
11	Repayments August 2005	23	Amount paid April 2005
12	Bill statement September 2005		

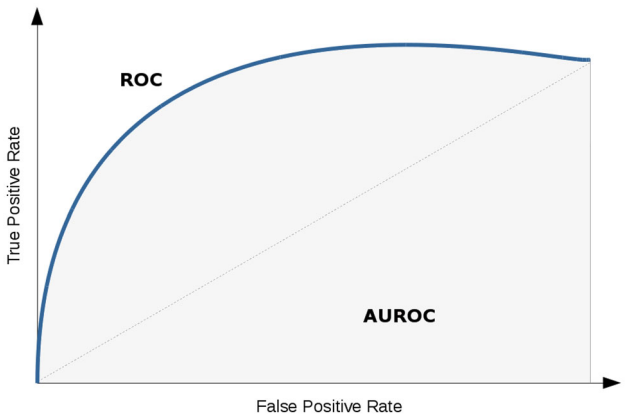

Fig. 4 ROC curve and AUROC area

sification thresholds (as shown in Fig. 4), and subsequently, the area under that curve is calculated. The AUROC value is in the range [0, 1], where 1 indicates the best performance.

4.1.3 Experimental methodology

As a preliminary step, we converted (if necessary) each instance classification in the datasets from its categorical or numerical former value into the canonical binary form, using 0 to classify the reliable instances and 1 to classify the unreliable ones, according to the simple criterion shown in Eq. 8.

$$c = \begin{cases} 1, & if\ c = reliable \\ 0, & if\ c = unreliable \end{cases} \qquad \forall c \in C. \tag{8}$$

Subsequently, to perform the experiments we divided each dataset into two slices: (1) an *in-sample* part, in order to detect the optimal parameters of the ensembled algorithms and the discretization value Δ, and (2) an *out-of-sample* part for performance evaluation, applying the *k-fold cross-validation* in context of these subsets. Such a criterion, largely used in many crucial data domains, avoids the *over-fitting* [37] problem that occurs by adopting the canonical *k-fold cross-validation* approach only, since it does not make a real separation between the data used to define the evaluation model and the ones used to evaluate its performance.

In more detail, in our approach, each dataset has been divided into 50% for *in-sample* and 50% for the out-of-sample part, and in each of these parts, we then apply a *k-fold cross-validation* procedure with $k = 5$. To perform such an experimental setup, we firstly perform the common k-fold cross-validation in the *in-sample data*, finding the parameters that maximize mean performance metrics in the validation data. Then, the found model is used for final evaluation. To do that, the *out-of-sample* data are also divided into $k-$folds, with each of them tested in that model. We report the mean metrics as final results. This way, we get the average of several and different configurations of testing data that were never seen before in the *k-fold cross-validation* trained models.

4.1.4 Implementation details

All the involved code has been developed in *Python* using numpy and *scikit-learn* (http://scikit-learn.org) libraries. For the discretization process, we used the $np.digitize()$ function, which converts the features to a discrete space according to where each feature value is located in an interval of bins. Such bins are defined as $bins = \{0, 1, \ldots, \Delta - 2, \Delta - 1\}$, where Δ is calculated experimentally. (We show how we find Δ later in this section.) The experiments reproducibility has been granted by fixing the *pseudo-random number generator* seed to 1 for randomly chosen parameters.

According to the credit scoring literature and a series of experiments aimed to select the best algorithms to use in our ensemble, we chose and tuned (by using the grid search method in the *in-sample* part of each dataset) five algorithms reported in Table 6, which reports the best algorithms param-

Table 6 In-sample algorithms parameters tuning

Algorithm	Parameter	Tuned value AC	GC	DC
Gradient boosting	*n_estimators*	25	25	25
(*GBC*)	*learning_rate*	0.01	0.1	0.1
	max_depth	2	2	4
Adaptive boosting	*n_estimators*	10	100	50
(*ADA*)	*learning_rate*	0.001	0.1	0.001
Random forests	*n_estimators*	20	20	20
(*RFC*)	*max_depth*	5	5	5
	min_samples_split	0.4	0.2	0.2
Multilayer perceptron	*alpha*	0.0001	0.001	0.0001
(*MLP*)	*max_iter*	50	100	50
	solver	*sgd*	*lbfgs*	*lbfgs*
Decision tree	*min_samples_split*	2	2	2
(*DTC*)	*max_depth*	1	1	5
	min_samples_leaf	1	1	1

eters found for the *AC*, *GC* and *DC* datasets. We define them as the ones used in our ensemble method through the rest of this paper.

4.2 Results

We start the experiments by performing a search in a large range of values for the optimal Δ that defines the upper limit of the discretization range in our approach (*i.e.*, $\{1, \ldots, \Delta\}$). The optimal Δ has been selected by taking into account the *average value* (α) of all the adopted metrics, calculated as follows:

$$\alpha = \frac{Sensitivity + AUROC}{2}. \tag{9}$$

Considering that the ROC curve plots the *true positive rate* (also known as *sensitivity*) versus the *false positive rate* (also known as *fallout*), parametrically, and the area under that curve represents the AUROC, the adoption of the α value in this process allows us to privilege the capability to detect the positive cases.

According to our validation strategy, such a process has been performed by using only the *in-sample* part of each dataset. As shown in Fig. 5, Figs. 6, and 7, which, respectively, refer to the α at each Δ value for each dataset (in order to simplify, it shows only the most significant range of values), the optimal Δ value is *10* for the *AC* dataset, *52* for the *GC* dataset and *9* for the *DC* dataset.

The second set of experiments are aimed to compare the performance of the proposed approach with those of the single and ensemble solutions that operate in the original feature space. Table 7 shows the performance metrics related to each single algorithm, the canonical ensemble configurations

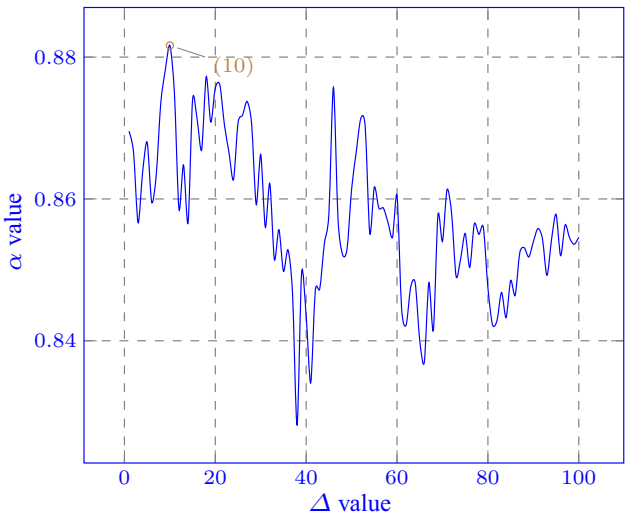

Fig. 5 In-sample discretization value tuning AC dataset according to the average value metric

based on the *full agreement ensemble* (FAE), *majority voting ensemble* (MVE) and *weighted voting ensemble* (WVE) strategies and the proposed approach (grayed rows) based on an ensemble regulated by a stochastic criterion that operates in the original feature space (SEO) and in our transformed feature space (SET). The best values are indicated in bold. Considering that we need to classify each instance, but a *full agreement* ensemble strategy can perform this operation only when all the algorithms agree, instead of excluding this strategy we decide to classify an event as *unreliable* in the absence of an agreement, according to a prudential criterion.

A consideration must be made regarding the performance evaluation process, since we have chosen not to take into account complex approaches/strategies in order to assess

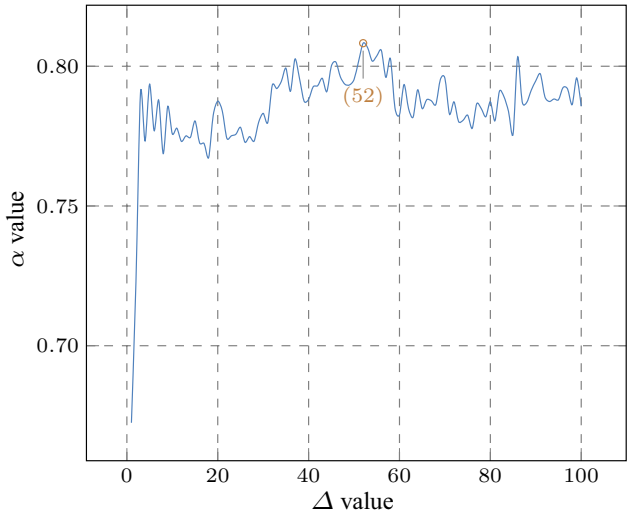

Fig. 6 In-sample discretization value tuning GC dataset according to the average value metric

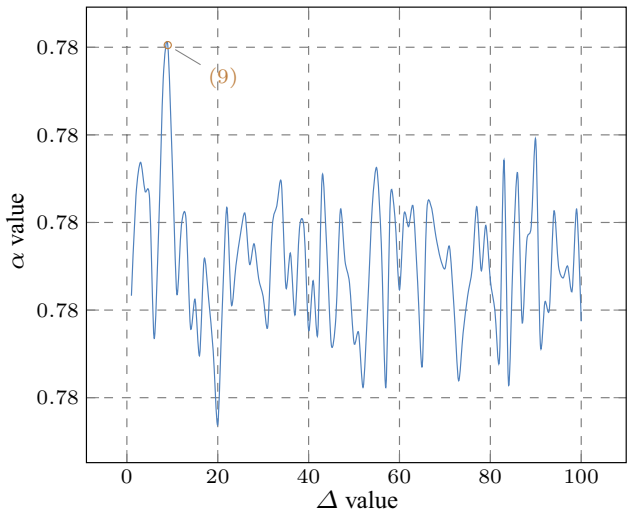

Fig. 7 In-sample discretization value tuning DC dataset according to the average value metric

Table 7 Out-of-sample performance: (top) results for AC dataset; (middle) results for GC dataset; and (bottom) results for DC dataset. Best results per metric are highlighted in bold

Approach	Type	Dataset	Sensitivity	AUROC
GBC	Algorithm	AC	0.850	0.768
ADA	Algorithm	AC	0.784	0.867
RFC	Algorithm	AC	0.825	0.842
MLP	Algorithm	AC	0.586	0.655
DTC	Algorithm	AC	0.784	0.867
FAE	Ensemble	AC	0.885	0.722
MVE	Ensemble	AC	0.796	0.869
WVE	Ensemble	AC	0.796	0.869
SEO	Ensemble	AC	**0.966**	0.630
SET	Ensemble	AC	0.915	**0.876**
GBC	Algorithm	GC	0.701	0.557
ADA	Algorithm	GC	0.747	0.635
RFC	Algorithm	GC	0.686	0.530
MLP	Algorithm	GC	0.773	0.669
DTC	Algorithm	GC	0.672	0.500
FAE	Ensemble	GC	0.785	0.679
MVE	Ensemble	GC	0.702	0.560
WVE	Ensemble	GC	0.701	0.559
SEO	Ensemble	GC	0.838	0.679
SET	Ensemble	GC	**0.843**	**0.700**
GBC	Algorithm	DC	0.849	0.670
ADA	Algorithm	DC	0.842	0.655
RFC	Algorithm	DC	0.837	0.644
MLP	Algorithm	DC	0.782	0.500
DTC	Algorithm	DC	0.846	0.663
FAE	Ensemble	DC	0.854	0.681
MVE	Ensemble	DC	0.846	0.663
WVE	Ensemble	DC	0.840	0.651
SEO	Ensemble	DC	**0.919**	0.696
SET	Ensemble	DC	0.889	**0.724**

the effectiveness of our approach in the context of baseline algorithms, even in ensemble configurations. This type of experimental approach finds its reason in the deduction that a performance improvement measured in these simple configurations will be inherited from their more sophisticated configurations. This idea is supported by various theories in the literature, such as, for instance, the ensemble learning one, as discussed in the work of Livieris *et al.* [46], or in that of Pintelas *et al.* [51], both of them indicating the performance of an ensemble of baseline algorithms as better than those of each single algorithm, or in [23], where Costa *et al.* discuss the problem of combining individual classifiers in ensemble in order to get a more accurate classifier. It should also be noted how the aforementioned literature concept, which indicates the improvement in terms of performance related to the adoption of multiple classifiers with respect to any single one of them, usually goes hand in hand with the concept related to the difficulty of optimizing a single classification algorithm compared to the optimization of a set of them. This has been widely discussed by Ala'raj *et al.* [3] and Zhang *et al.* [66] in their works.

4.2.1 Data heterogeneity analysis

Given that the proposed data preprocessing approach combines the discretization and enrichment processes in order to improve the events characterization and reducing, thus, the data heterogeneity, we performed an evaluation about the effective impact of such a heterogeneity reduction. This has

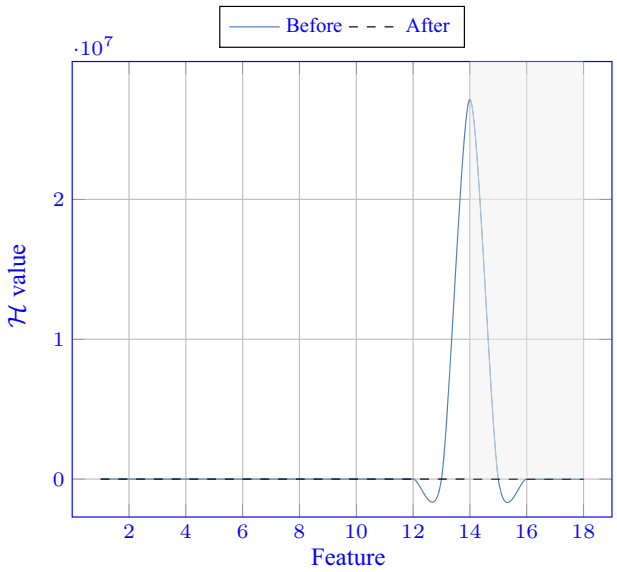

Fig. 8 Heterogeneity related to the AC dataset, evaluated in terms of feature variance, before and after the feature space transformation

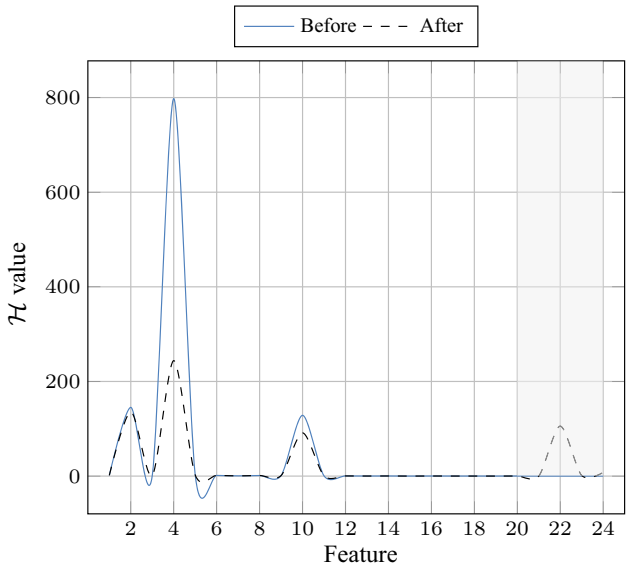

Fig. 9 Heterogeneity related to the GC dataset, evaluated in terms of feature variance, before and after the feature space transformation

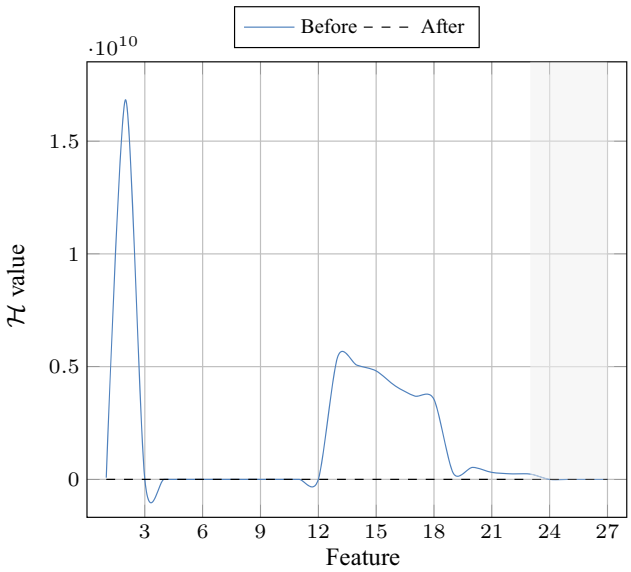

Fig. 10 Heterogeneity related to the DC dataset, evaluated in terms of feature variance, before and after the feature space transformation

been done by considering that an huge number of potential patterns lead toward a big computational complexity regardless the involved classification algorithms, with this being a potential problem in the context of credit scoring systems that operate in real time.

To perform such study, we measured the data heterogeneity in terms of feature variance, as shown in Fig. 8, Figs. 9, and 10. The variance gives us a measure on how far the feature values spread out with regard to their average value. Premising that, in statistics, the term *population* indicates the entire group of samples to study, and that in our context, such a term indicates the group of values that characterize each dataset feature, the formalization of the variance is shown in Eq. 10 as follows

$$\sigma^2 = \frac{\sum_{i=1}^{N} (x_i - \mu)^2}{N}, \tag{10}$$

where σ^2 denotes the feature variance (population variance), x_i is the value of the feature at the i^{th} instance, μ is the average feature value (population mean), and N is the number of instances (data points).

If we transpose this canonical formalization of the variance as a measure of the data heterogeneity (denoted as $\mathcal{H}$ in the following) in our credit scoring domain, we obtain Eq. 11.

$$\mathcal{H} = \frac{\sum_{x=1}^{X} \left(i_x^{f_n} - \frac{i_x^{f_n}}{X} \right)^2}{X}, \tag{11}$$

Starting from the canonical formalization of the variance shown in Eq. 10, we reformulated it in order to formalize the data heterogeneity, with regard to the set $I = \{i_1, i_2, \ldots, i_X\}$ (but this applies equally to the set $\hat{I} = \{\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_Z\}$).

In more detail: we replaced σ^2 with $\mathcal{H}$ to denote the data heterogeneity instead of the variance; we replaced N with X to denote the size of set I (since $I = \{i_1, i_2, \ldots, i_X\}$); we replaced x_i with $i_x^{f_n}$ to denote a single feature f_n of the instance i_x in the set I; and we replaced μ with $\frac{i_x^{f_n}}{X}$ to denote the mean value of the feature f_n in the entire dataset I.

In the statistics theory, especially in the extreme value analysis branch [6], when a dataset is characterized by a lower variability, its values are more consistent, whereas when the

variability is higher, its data points are more different and the extreme values become more likely. For this reason, a reduction of such a variability in the dataset can lead toward better classification performance. Therefore, it can be seen in Fig. 9 that, for the GC dataset, the proposed feature space, defined by discretizing the original feature values followed by adding four new features, reduces the feature heterogeneity (expressed in terms of variance) in a substantial way. This reduction is much more evident in the AC and DC datasets, as shown in Fig. 8 and Fig. 10. It should also be noted that the increase in variance measured in the last four features (areas highlighted in gray) is due to the introduction of them (that were not there before), according to the proposed approach.

4.2.2 Discussion

The obtained results lead us toward the following considerations:

- As shown in Table 7, in each dataset the proposed approach outperforms all the single algorithms and all the ensemble canonical strategies in terms of *Sensitivity* and AUROC metrics. It should also be observed how the proposed ensemble approach outperforms its competitors with and without data preprocessing, but best AUROC results are achieved by using the new feature space. Finally, it can be seen that only in the *AC* and *DC* datasets, our ensemble approach gets better sensitivity results without the data preprocessing, but at a cost of a lower AUROC score. This means that the most effective approach is the one that operates in the new feature space, as the AUROC better describes the classification performance in imbalanced datasets;
- The combination of the *in-sample/out-of-sample* and *k-fold cross-validation* criteria allowed us obtaining results not influenced by over-fitting;
- Regarding the ensemble baselines, it could be noticed that only the full agreement ensemble enhanced the performance of the best individual approach of that ensemble, being the best state-of-the-art competitor. This is mainly due to the additional prudential criterion we adopted (a sample is unreliable if there is not an agreement);
- Most of the individual approaches showed very unstable results. For example, the multilayer perceptron (MLP) algorithm was the best individual classifier for the DC dataset when considering the AUROC metric, and gradient boosting (GBC) was the best for the DC dataset. Both classifiers showed to be very good for one dataset, but very bad for the other, which means that they are affected by different levels of imbalance and heterogeneity in both datasets. These issues further highlight our approach contributions.

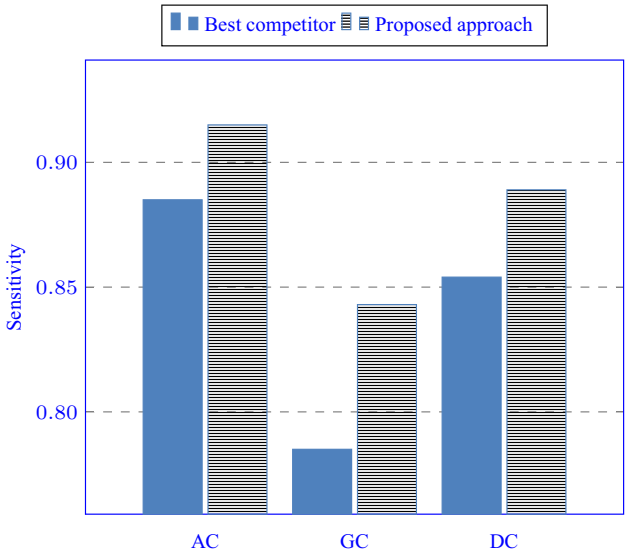

Fig. 11 Out-of-sample sensitivity performance overview

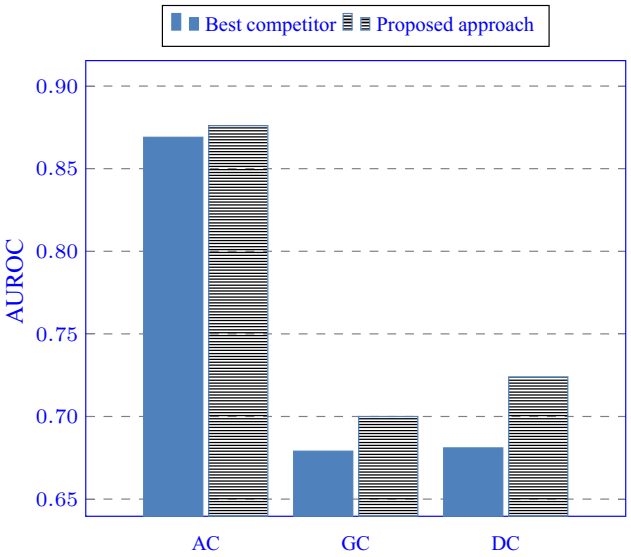

Fig. 12 Out-of-sample AUROC performance overview

As last consideration, we refer to the data heterogeneity analysis performed in Sect. 4.2.1, which shows how the number of potential instance patterns has been dramatically reduced by preprocessing the feature space according to the proposed approach. This represents an advantage in terms of computational complexity, since a minor number of involved patterns trivially lead toward a less complex process of definition of the evaluation model. It can be considered an important milestone, considering that we get better classification performance in our transformed feature space.

Finally, the performance overview in terms of sensitivity and AUROC metrics, respectively, shown in Figs. 11 and 12, indicates how the proposed combination of a stochastic ensemble classifier with a discretized/enriched feature space

process is effective in credit scoring, since such a combination outperforms the best state-of-the-art competitor with a large margin.

In summary, our approach is effective for credit scoring because, in our stochastic ensemble, we define a very rigorous classification procedure for the most frequent class in credit scoring datasets. Our proposed weighting scheme underweights individual reliable classifications with low probability and, at the same time, does not affect classifiers with unreliable classifications, no matter how high their confidences are. This can be better seen from Eq. 6, where reliable classifications with less confidence than the mean confidence of all models of the ensemble are declared unreliable. This way, a twofold effect can be seen in the final credit scoring ensemble algorithm: (1) minimization of the misclassification of reliable samples (most frequent class) in the datasets, as can be seen in requirement w_1 in Eq. 6, and (2) maximization of correct classification of unreliable samples (the less frequent class), as can be seen in requirement w_2 in the same equation. Therefore, the approach deals efficiently with imbalanced datasets, as it deals with noise from the individual classifications regarding the reliable (most frequent) cases, classifying a reliable sample only if the individual models of the ensemble have high levels of confidence in their predicted labels. Fusing such stochastic ensemble with an enhanced input feature space, as also proposed in this paper, can boost the credit scoring performance.

5 Conclusions and future work

The development of effective credit scoring tools is becoming more crucial in this era dominated by consumer credit, with this scenario leading to an increasing number of researchers spending efforts to define new approaches able to overcome open problems in this domain. Notwithstanding, issues like *data heterogeneity* and *data imbalance* are still affecting real-world credit scoring and, thus, require more complex solutions.

The proposed approach faces such limitations in credit scoring by acting on two different fronts, data preprocessing and rule-based ensembling. The data preprocessing step transforms the original feature space by discretizing the feature values, enriching such a space by adding meta-information. The discretization process allows us to merge similar patterns, reducing the data heterogeneity by merging similar patterns, whereas the enrichment process counteracts the loss of information related to the discretization process, improving the instance characterization in one of two possible classes (i.e., *reliable* or *unreliable*). Subsequently, we apply on the transformed feature space a stochastic ensemble criterion able to minimize the reliable samples misclassification, maximizing the unreliable samples classification, facing at the same time the data imbalance problem. In order to produce more valuable results, the validation process of the proposed approach has been performed by using several real-world credit scoring datasets, adopting in this context a double criterion: *k-fold cross-validation* and *complete data separation*. This is because the use of a single *k-fold cross-validation* criterion does not grant a real separation between the data used to train the evaluation model, and the data used to assess its performance. In addition, the aforementioned double criterion offers results not biased by over-fitting. According to this, each dataset has been divided into two parts, an *in-sample* part devoted to the training process of the evaluation model and an *out-of-sample* part used to assess the performance, continuing to maintain in each involved process a *k-fold cross-validation* criterion with $k=5$.

As future work, we are already working to build a general domain big data infrastructure using the Amazon cloud, Apache Spark, Terraform and other tools to automatically create as many Amazon instances as desired and run the entire computation in a distributed fashion to reduce computational complexity, in accordance with what has been discussed in Sect. 3.3.1. Moreover, we plan to extend the proposed approach to other data domains characterized by similar problems of those that are present in the credit scoring, such as security (*e.g.*, telecommunication fraud detection [1], financial fraud detection [55] and network intrusion detection [58]), medical applications (*e.g.*, rare diseases diagnosis [43] and cancer gene expressions [45]) and other class-imbalanced data domains [35].

Acknowledgements This research was partially funded and supported by the "Bando 'Aiuti per progetti di Ricerca e Sviluppo'– POR FESR 2014-2020 – Asse 1, Azione 1.1.3. Project IntelliCredit: AI-powered digital lending platform."

Declarations

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

References

1. Ab Raub, R., Hamzah, A.H.N., Jaafar, M.D., Baharim, K.N.: Using subscriber usage profile risk score to improve accuracy of telecommunication fraud detection. In: 2016 International Conference on Computational Intelligence and Cybernetics, pp. 127–131. IEEE (2016)
2. Abellán, J., Castellano, J.G.: A comparative study on base classifiers in ensemble methods for credit scoring. Expert Syst. Appl. **73**, 1–10 (2017)
3. Ala'raj, M., Abbod, M.F.: Classifiers consensus system approach for credit scoring. Knowl.-Based Syst. **104**, 89–105 (2016)

4. Arora, N., Kaur, P.D.: A bolasso based consistent feature selection enabled random forest classification algorithm: an application to credit risk assessment. Appl. Soft Comput. **86**, 105936 (2020)
5. Babaev, D., Savchenko, M., Tuzhilin, A., Umerenkov, D.: E.t.-rnn: Applying deep learning to credit loan applications. In: SIGKDD International Conference on Knowledge Discovery & Data Mining, p. 2183–2190. ACM, New York, NY, USA (2019)
6. Benstock, D., Cegla, F.: Extreme value analysis (eva) of inspection data and its uncertainties. NDT E Int. **87**, 68–77 (2017)
7. Bequé, A., Lessmann, S.: Extreme learning machines for credit scoring: An empirical evaluation. Expert Syst. Appl. **86**, 42–53 (2017)
8. Bijak, K., Mues, C., So, M.C., Thomas, L.: Credit card market literature review: Affordability and repayment (2015)
9. Bilalli, B., Abelló, A., Aluja-Banet, T.: On the predictive power of meta-features in openml. Int. J. Appl. Math. Comput. Sci. **27**(4), 697–712 (2017)
10. Bischl, B., Kühn, T., Szepannek, G.: On class imbalance correction for classification algorithms in credit scoring. In: Operations Research Proceedings 2014, pp. 37–43. Springer (2016)
11. Brown, I., Mues, C.: An experimental comparison of classification algorithms for imbalanced credit scoring data sets. Expert Syst. Appl. **39**(3), 3446–3453 (2012)
12. Carta, S., Fenu, G., Ferreira, A., Recupero, D.R., Saia, R.: A two-step feature space transforming method to improve credit scoring performance. In: International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, pp. 134–157. Springer (2019)
13. Carta, S., Ferreira, A., Recupero, D.R., Saia, M., Saia, R.: A combined entropy-based approach for a proactive credit scoring. Eng. Appl. Artif. Intell. **87**, 103292 (2020)
14. Carta, S., Medda, A., Pili, A., Reforgiato Recupero, D., Saia, R.: Forecasting e-commerce products prices by combining an autoregressive integrated moving average (arima) model and google trends data. Future Internet **11**(1), 5 (2019)
15. Changjian, L., Peng, H.: Credit risk assessment for rural credit cooperatives based on improved neural network. In: International Conference on Smart Grid and Electrical Automation (ICSGEA), pp. 227–230. IEEE, Changsha, China (2017)
16. Chatterjee, A., Segev, A.: Data manipulation in heterogeneous databases. ACM SIGMOD Rec. **20**(4), 64–68 (1991)
17. Chawla, N.V., Japkowicz, N., Kotcz, A.: Special issue on learning from imbalanced data sets. ACM Sigkdd Explor. Newslett. **6**(1), 1–6 (2004)
18. Chen, H., Jiang, M., Wang, X.: Bayesian ensemble assessment for credit scoring. In: 2017 4th International Conference on Industrial Economics System and Industrial Security Engineering (IEIS), pp. 1–5. IEEE (2017)
19. Chen, M., Dautais, Y., Huang, L., Ge, J.: Data driven credit risk management process: A machine learning approach. In: Proceedings of the 2017 International Conference on Software and System Process, ICSSP 2017, p. 109–113. Association for Computing Machinery, New York, NY, USA (2017). 10.1145/3084100.3084113
20. Chen, N., Ribeiro, B., Chen, A.: Financial credit risk assessment: a recent review. Artif. Intell. Rev. **45**(1), 1–23 (2016)
21. Chen, X., Liu, Z., Zhong, M., Liu, X., Song, P.: A deep learning approach using deepgbm for credit assessment. In: Proceedings of the 2019 International Conference on Robotics, Intelligent Control and Artificial Intelligence, RICAI 2019, p. 774–779. Association for Computing Machinery, New York, NY, USA (2019). 10.1145/3366194.3366333
22. Chivers, I., Sleightholme, J.: An introduction to algorithms and the big o notation. In: Introduction to Programming with Fortran, pp. 359–364. Springer (2015)
23. Costa, V.S., Farias, A.D.S., Bedregal, B., Santiago, R.H., Canuto, A.M.D.P.: Combining multiple algorithms in classifier ensembles using generalized mixture functions. Neurocomputing **313**, 402–414 (2018)
24. Crook, J.N., Edelman, D.B., Thomas, L.C.: Recent developments in consumer credit risk assessment. Eur. J. Oper. Res. **183**(3), 1447–1465 (2007)
25. Damrongsakmethee, T., Neagoe, V.E.: Principal component analysis and relieff cascaded with decision tree for credit scoring. In: Computer Science On-line Conference, pp. 85–95. Springer (2019)
26. De Sá, C.R., Soares, C., Knobbe, A.: Entropy-based discretization methods for ranking data. Inf. Sci. **329**, 921–936 (2016)
27. Dietterich, T.G.: Ensemble methods in machine learning. In: Multiple Classifier Systems. Lecture Notes in Computer Science, vol. 1857, pp. 1–15. Springer, United States of America (2000)
28. Domingos, S.D.O., de Oliveria, J.F., de Mattos Neto, P.S.: An intelligent hybridization of arima with machine learning models for time series forecasting. Knowl.-Based Syst. **175**, 72–86 (2019)
29. Fan, Q., Liu, X., Zhang, Y., Bao, F., Li, S.: Adaptive mutation pso based svm model for credit scoring. In: Proceedings of the 2nd International Conference on Computer Science and Application Engineering, CSAE '18. Association for Computing Machinery, New York, NY, USA (2018)
30. Fan, Q., Wang, Z., Li, D., Gao, D., Zha, H.: Entropy-based fuzzy support vector machine for imbalanced datasets. Knowl.-Based Syst. **115**, 87–99 (2017)
31. Feng, X., Xiao, Z., Zhong, B., Qiu, J., Dong, Y.: Dynamic ensemble classification for credit scoring using soft probability. Appl. Soft Comput. **65**, 139–151 (2018)
32. Fonseca, D.P., Wanke, P.F., Correa, H.L.: A two-stage fuzzy neural approach for credit risk assessment in a brazilian credit card company. Appl. Soft Comput. (2020). https://doi.org/10.1016/j.asoc.2020.106329
33. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J.M., Herrera, F.: Big data preprocessing: methods and prospects. Big Data Anal. **1**(1), 9 (2016)
34. Gomes, H.M., Barddal, J.P., Enembreck, F., Bifet, A.: A survey on ensemble learning for data stream classification. ACM Comput. Surv. **50**(2), 1–36 (2017)
35. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: review of methods and applications. Expert Syst. Appl. **73**, 220–239 (2017)
36. Hashem, I.A.T., Anuar, N.B., Gani, A., Yaqoob, I., Xia, F., Khan, S.U.: Mapreduce: review and open challenges. Scientometrics **109**(1), 389–422 (2016)
37. Hawkins, D.M.: The problem of overfitting. J. Chem. Inf. Comput. Sci. **44**(1), 1–12 (2004)
38. Henrique, B.M., Sobreiro, V.A., Kimura, H.: Literature review: machine learning techniques applied to financial market prediction. Expert Syst. Appl. **124**, 226–251 (2019)
39. Jaber, J.J., Ismail, N., Ramli, S., Al Wadi, S., Boughaci, D.: Assessment of credit losses based on arima-wavelet method. J. Theor. Appl. Inf. Technol. **98**(09), 1379–392 (2020)
40. Jimbo Santana, P., Villa Monte, A., Rucci, E., Lanzarini, L.C., Fernández Bariviera, A.: Analysis of methods for generating classification rules applicable to credit risk. Journal of Computer Science & Techonology (2017)
41. Khemakhem, S., Ben Said, F., Boujelbene, Y.: Credit risk assessment for unbalanced datasets based on data mining, artificial neural network and support vector machines. J. Modell. Manag. **13**(4), 932–951 (2018)
42. Kotsiantis, S., Kanellopoulos, D.: Discretization techniques: a recent survey. GESTS Int. Trans. Comput. Sci. Eng. **32**(1), 47–58 (2006)
43. Lei, W., Zhang, R., Yang, Y., Wang, R., Zheng, W.S.: Class-center involved triplet loss for skin disease classification on imbalanced

data. In: 2020 IEEE 17th International Symposium on Biomedical Imaging (ISBI), pp. 1–5. IEEE (2020)
44. Li, Y., Wang, X., Djehiche, B., Hu, X.: Credit scoring by incorporating dynamic networked information. Eur. J. Oper. Res. **286**, 1103–1112 (2020)
45. Liu, Z., Tang, D., Cai, Y., Wang, R., Chen, F.: A hybrid method based on ensemble welm for handling multi class imbalance in cancer microarray data. Neurocomputing **266**, 641–650 (2017)
46. Livieris, I.E., Kiriakidou, N., Kanavos, A., Tampakas, V., Pintelas, P.: On ensemble ssl algorithms for credit scoring problem. In: Informatics, vol. 5, p. 40. Multidisciplinary Digital Publishing Institute (2018)
47. Luengo, J., García-Gil, D., Ramírez-Gallego, S., García, S., Herrera, F.: Big Data Discretization. Big Data Preprocessing, pp. 121–146. Springer, Berlin (2020)
48. López, J., Maldonado, S.: Profit-based credit scoring based on robust optimization and feature selection. Inf. Sci. **500**, 190–202 (2019)
49. Meng, X., Bradley, J., Yavuz, B., Sparks, E., Venkataraman, S., Liu, D., Freeman, J., Tsai, D., Amde, M., Owen, S., et al.: Mllib: machine learning in apache spark. J. Mach. Learn. Res. **17**(1), 1235–1241 (2016)
50. Mester, L.J., et al.: What's the point of credit scoring? Bus. Rev. **3**, 3–16 (1997)
51. Pintelas, P., Livieris, I.E.: Special issue on ensemble learning and applications (2020)
52. Pławiak, P., Abdar, M., Acharya, U.R.: Application of new deep genetic cascade ensemble of svm classifiers to predict the australian credit scoring. Appl. Soft Comput. **84**, 105740 (2019)
53. Sagi, O., Rokach, L.: Ensemble learning: a survey. WIREs Data Min. Knowl. Discov. **8**(4), e1249 (2018)
54. Saia, R., Carta, S.: An entropy based algorithm for credit scoring. In: International Conference on Research and Practical Issues of Enterprise Information Systems, pp. 263–276. Springer (2016)
55. Saia, R., Carta, S.: Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach. In: SECRYPT, pp. 335–342 (2017)
56. Saia, R., Carta, S.: A fourier spectral pattern analysis to design credit scoring models. In: 1st International Conference on Internet of Things and Machine Learning, p. 18. ACM, United Kingdom (2017)
57. Saia, R., Carta, S., Fenu, G.: A wavelet-based data analysis to credit scoring. In: Proceedings of the 2nd International Conference on Digital Signal Processing, pp. 176–180. ACM, Tokyo, Japan (2018)
58. Saia, R., Carta, S., Recupero, D.R.: A probabilistic-driven ensemble approach to perform event classification in intrusion detection system. In: KDIR, pp. 139–146. SciTePress (2018)
59. Santana, P.J., Lanzarini, L., Bariviera, A.F.: Fuzzy credit risk scoring rules using frvarpso. Int. J. Uncertain. Fuzziness Knowl.-Based Syst. **26**(Suppl. 1), 39–57 (2018)
60. Sharmin, S., Shoyaib, M., Ali, A.A., Khan, M.A.H., Chae, O.: Simultaneous feature selection and discretization based on mutual information. Pattern Recogn. **91**, 162–174 (2019)
61. Siddiqi, N.: Intelligent Credit Scoring: Building and Implementing Better Credit Risk Scorecards. John Wiley & Sons, United States of America (2017)
62. Tripathi, D., Edla, D.R., Cheruku, R.: Hybrid credit scoring model using neighborhood rough set and multi-layer ensemble classification. J. Intell. Fuzzy Syst. **34**(3), 1543–1549 (2018)
63. Wang, C., Han, D., Liu, Q., Luo, S.: A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism lstm. IEEE Access **7**, 2161–2168 (2019)
64. Wang, G., Hao, J., Ma, J., Jiang, H.: A comparative assessment of ensemble learning for credit scoring. Expert Syst. Appl. **38**(1), 223–230 (2011). https://doi.org/10.1016/j.eswa.2010.06.048
65. Zeng, G.: On the confusion matrix in credit scoring and its analytical properties. Commun. Stat.-Theory Methods **49**(9), 2080–2093 (2020)
66. Zhang, D., Zhou, X., Leung, S.C., Zheng, J.: Vertical bagging decision trees model for credit scoring. Expert Syst. Appl. **37**(12), 7838–7843 (2010)
67. Zhang, H., He, H., Zhang, W.: Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. Neurocomputing **316**, 210–221 (2018)
68. Zhang, X., Yang, Y., Zhou, Z.: A novel credit scoring model based on optimized random forest. In: Annual Computing and Communication Workshop and Conference (CCWC), pp. 60–65. IEEE, Las Vegas, USA (2018)
69. Zhang, Z., He, J., Gao, G., Tian, Y.: Sparse multi-criteria optimization classifier for credit risk evaluation. Soft Comput. **23**(9), 3053–3066 (2019)
70. Zou, Q., Xie, S., Lin, Z., Wu, M., Ju, Y.: Finding the best classification threshold in imbalanced classification. Big Data Res. **5**, 2–8 (2016)