A Two-Step Feature Space Transforming Method to Improve Credit Scoring Performance

Salvatore Carta, Gianni Fenu, Anselmo Ferreira, Diego R. Recupero, and Roberto Saia

Department of Mathematics and Computer Science, University of Cagliari, Italy,
{salvatore,fenu,anselmo.ferreira,diego.reforgiato,roberto.saia}@unica.it

Abstract. The increasing amount of credit offered by financial institutions has required intelligent and efficient methodologies of credit scoring. Therefore, the use of different machine learning solutions to that task has been growing during the past recent years. Such procedures have been used in order to identify customers who are reliable or unreliable, with the intention to counterbalance financial losses due to loans offered to wrong customer profiles. Notwithstanding, such an application of machine learning suffers with several limitations when put into practice, such as unbalanced datasets and, specially, the absence of sufficient information from the features that can be useful to discriminate reliable and unreliable loans. To overcome such drawbacks, we propose in this work a Two-Step Feature Space Transforming approach, which operates by evolving feature information in a twofold operation: (i) data enhancement; and (ii) data discretization. In the first step, additional meta-features are used in order to improve data discrimination. In the second step, the goal is to reduce the diversity of features. Experiments results performed in real-world datasets with different levels of unbalancing show that such a step can improve, in a consistent way, the performance of the best machine learning algorithm for such a task. With such results we aim to open new perspectives for novel efficient credit scoring systems.

Keywords: Business Intelligence · Credit Scoring · Machine Learning· Algorithms · Transforming

1 Introduction

A report from *Trading Economics* [1, 2], which is based on the information provided by the *European Central Bank*[1] data, has shown that credit for consumers has been regularly increasing over the last years. Such behavior in the Euro zone, which can be seen in Fig. 1, is also noticed in other markets such as Russia and USA. This increasing phenomenon has forced *Credit Rating Agencies* (CRAs), also known as *ratings services*, to define and establish intelligent strategies to offer credit for the right customers, minimizing financial losses due to bad debts.

Nowadays, CRAs have been using credit scoring systems coupled with machine learning solutions in order to perform credit scoring. Such approaches take into account the big data nature of credit datasets, which can enable machine learning models that can understand credit information from clients and, consequently, discriminate them

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

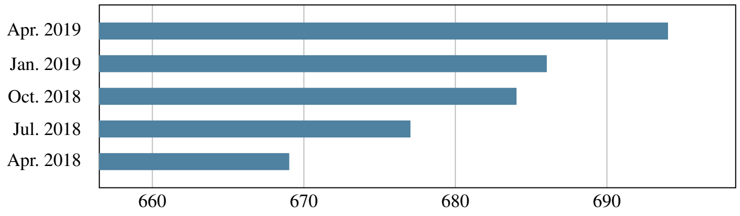

Fig. 1. Euro zone consumer credit in billions of euros

into *reliable* or *non-reliable* users. Credit scoring systems have been vital in many financial areas [3], as they avoid human interference and eliminate biased analyzes of people information who request financial services, such as a loan. Basically, most of these approaches can be considered probabilistic approaches [4], performing credit scoring by calculating in real time the probability of the loan being repaid, partially repaid and even not repaid based on the given information (*e.g.*, age, job, salary, previous loans status, marital status, among others) in credit scoring datasets, helping the financial operator in the decision of grating or not a financial service [5].

Notwithstanding, credit scoring systems are still limited as a solution for defining loans for three main reasons. The fist one lies in the dataset nature of the problem itself. Similar to other problems, namely fraud or intrusion detection [6–8], the source of data typically contains different distributions of classes [9, 10], which, in the specific case of credit scoring, is more favorable to the reliable instances rather than to the unreliable ones [11]. Such a behavior can seriously affect the performance of classification algorithms, once they can be often biased to classify the most frequent class [11, 12]. The second problem comes from the fact that some datasets face the *cold start* issue, on which the unreliable cases do not even exist. Such an issue has motivated several *proactive methods* in the literature to deal with such a problem [13–15]. The last problem, which motivates our solution presented in this work, arises from the heterogeneity of the data. Such limitation highlights the fact that the data, the way they are disposed in datasets, are not enough to describe the different instances. Such information is characterized by features that are very different from each other, even thought they belong to the same class of information. Therefore, further feature transformations are still needed to provide insightful credit scoring.

Based on our previous experience [13–18] to deal with credit scoring, we present in this work a solution for data heterogeneity in credit scoring datasets. We do that by assessing the performance of a Two-Step Feature Space Transforming (TSFST) method we previously proposed in [19] to improve credit scoring systems. Our approach to improve features information has a twofold process, composed of (i) enrichment; and (ii) discretization phases. The enrichment step adds several meta-features in the data, in order to better spread the different instances into separated clusters in the D dimensional space, whereas the discretization process is done to reduce the number of feature patterns. For the sake of avoiding the risk of overfitting [20] associated to our method and also highlight its real advantages, we adopted an experimental setup that aims at assessing the real performance of financial systems [21]. Such a methodology considers our TSFST method evaluated on data never seen before, which we name *out-of-sample*, and trained on known and different data, which we name *in-sample* data. Experiments considering different classifiers dealing with such feature improvement method spot the effectiveness of such approach, which can mitigate even the data unbalance nature of such datasets.

In summary, the main contributions provided through this work are:

1. The establishment of the Two-Step Feature Space Transforming approach, which enriches and discretizes the original features from credit scoring datasets in order to boost machine learning classifiers performance when using these features.
2. The assessment of the best classifier to be used with the proposed method, done after a series of experiments considering the *in-sample* part of the datasets.
3. The analysis of the method performance considering the *out-of-sample* part of each dataset, adding a comparison with the same canonical approach but without considering our proposed method.

The work presented in this paper is based on our previously published one [19]. Notwithstanding, it has been extended in such a way to add the following new discussions and contributions:

1. Extension of the *Background and Related Work* section by discussing more relevant and recent state-of-the-art approaches, extending the information related to this research field with the aim to provide the readers a quite exhaustive overview on the credit scoring scenario.
2. We changed the order of operations reported in our previous work [19], as we realized that it achieves better results.
3. Inclusion of a new real-world dataset, which allows us to evaluate the performance using a dataset characterized by a low number of instances (690, which is lower than 1000 and 30000 from the other datasets) and features (14, which is also a low number if compared to 21 and 23 from the other datasets).
4. We better discuss the composition of the *in-sample* and *out-of-sample* datasets in terms of number of involved instances and classes. Our choice is based on better providing details about the data imbalance that is present during both the definition of the model (done with the *in-sample* dataset) and the evaluation of its performance (done with the *out-of-sample* dataset).
5. We perform an analysis of the asymptotic time complexity related to the proposed algorithm, which adds valuable information in the context of considering *real-time* credit scoring systems.

The rest of this paper has been structured as follows: Section 2 provides information about the background and the related work of the credit scoring domain. Section 3 introduces the formal notation used and provides the formalization of our proposed method. Section 4 describes the experimental environment considered. Section 5 reports and discusses the experimental results in the credit scoring environment and, finally, Section 6 makes some concluding remarks and points out some directions for future works.

2 Related Work

In the past few years, it has been witnessed an increasing investment and research over the credit scoring applications with the aim of performing efficient credit scoring. The literature [22] describes several kinds of credit risk models in respect to the *unreliable* cases, which are commonly known as *default* cases. Such models are divided into: (i) *Probability of Default* (PD) models, which investigate the probability of a default in a period; (ii) *Exposure at Default* (EAD) models, which analyse the value the financial operator is exposed to if a default happens; and (iii) *Loss Given Default* (LGD) models, which evaluate the amount of money the operator loses after a default happens. In this section, we discuss the related work of the first kind of models (PD) only, as they are related to our proposed method. Further details of EAD and LGD models can be found in several surveys in the literature [23–25].

The related work in PD models can be strictly divided into six main branches. The first branch of research is based on statistical methods. For instance, the work in [26] applies Kolmogorov-Smirnov statistics in credit scoring features to discriminate default and non-default users. Other methods, such as the *Logistic Regression* (LR) [27] and *Linear Discriminant Analysis* [28] are also explored in the literature to predict the probability of a default. In [29], the authors propose to use Self Organized Maps and fuzzy k-Nearest Neighbors for credit scoring.

The second branch of research aims to explore data features transformed into other feature domains. The work in [18] processes data in the wavelet domain with three metrics used to rate customers. Similarly, the approach in [17] uses differences of magnitudes in the frequency domain. Finally, the approach in [13] performs comparison of non-square matrix determinants to allow or deny loans.

The third branch of approaches, which is among the most popular ones in credit scoring management, is based on machine learning models. In this topic, the work in [30] considers a Random Forest on preprocessed data. A three-way decision methodology with probability sets is considered in [31]. In [32], a deep learning Convolutional Neural Network approach is applied to pre-selected features that are converted to images. A specific Support Vector Machines with kernel-free fuzzy quadratic surface is proposed in [33]. The work in [34] reports the beneficial use of bagging, boosting and Random Forest techniques to plan and evaluate a housing finance program. An extensive work with machine learning is done in [25], where forty-one methods are compared when applied to eight Credit Scoring datasets.

In the fourth branch of research, approaches based on general artificial intelligence such as neural networks have been explored. For example, authors in [35] present the application of artificial intelligence in the credit scoring area. In [36], the authors use a novel kind of artificial neural network called extreme learning machines. The work in [37] reports credit score prediction using the Takagi-Sugeno neuro-fuzzy network. Finally, the work in [38] performs a benchmark of different neural networks for credit scoring.

The fifth branch of research considers hybrid approaches, where more than one model is used to perform a final decision of credit scoring. The work in [39] used Gabriel Neighbourhood Graph and Multivariate Adaptive Regression Splines together with a new consensus approach. Authors in [40] used seven base different classifiers in dimensionality reduced data with Neighborhood Rough Set. The authors propose a novel ranking technique used to decide the top-5 best classifiers to be part of a layered ensemble. The work in [41] uses several classifiers to validate a feature selection approach called *group penalty function*. In [42], a similar procedure is done, but including normalization and dimensionality reduction preprocessing steps and an ensemble of five classifiers optimized by a Bayesian algorithm. The same number of classifiers is used in [43], but with genetic algorithm and fuzzy assignment. In [44], ensembles are done according to classifier soft probabilities and, in [45], an ensemble with feature clustering-based feature is done in a weighted voting approach.

The last set of models consider specific features of the problem, such as *user profiling* in social networks [46–49], news from media [50], data entropy [16] , linear-dependence [13, 15], among others. One interesting research in this topic is considering *proactive methods* [13–15], which previously assume that the credit scoring datasets are biased and alleviate such a problem before they happen.

Although several approaches have been proposed in literature, there are still many challenges in credit scoring research. All these issues reduce in a significant way the performance of Credit Scoring systems, specially when applied to real-world credit risk management. Such challenges can be enumerated as follows:

1. *Lack of Datasets*, caused mainly by privacy, competition, or legal issues [51].
2. *Non-adaptability*, commonly known as *overfitting*, where Credit Scoring models are unable to correctly classify new instances.
3. *Cold-start*, when the datasets used to train a model do not contain enough information about default and non default cases [52–56].
4. *Data Unbalance*, where an imbalanced class distribution of data [57, 58] is found, being typically beneficial to the non-default class.
5. *Data Heterogeneity*, where the same information is represented differently in different data samples [59].

Our approach differs from the previous ones in the literature as it deals with the *Data Heterogeneity* problem in a two step process. To do that, we perform a series of transforming steps in order to make the original heterogeneous data better discernible and separable, which can boost the performance of any classifier. More details of our approach are discussed in the next section.

3 The Two-step Feature Space Transforming Approach

Before discussing our approach in details, let us define the formal notation used from this section to the rest of this work. Given a set $S = \{s_1, s_2, \ldots, s_X\}$ of samples (or instances) already classified in another set $C = \{reliable, unreliable\}$, we then split S into subsets $S^+ \subseteq S$ of *reliable* or *non default* cases, and another subset $S^- \subseteq S$ of *unreliable* cases, where $S^+ \cap S^- = \emptyset$. Lets also consider another set $P = \{p_1, p_2, \ldots, p_X\}$ as the labels (or predictions) given by a credit scoring system for each sample that will split S as discussed before, and $Y = \{y_1, y_2, \ldots, y_X\}$ their true labels where $P \in C$, $Y \in C$ and $|S| = |P| = |Y|$. By considering that each sample has a set of features $F =$

$\{f_1, f_2, \ldots, f_N\}$ and that each sample belongs to only one class in the set C, we can formalize our objective as shown in Equation 1 as follows:

$$\max_{0 \leq \alpha \leq |S|} \alpha = \sum_{z=1}^{|S|} \beta_{(p_z == y_z)}, \tag{1}$$

where β_b is a logical function that converts any proposition b into 1 if the proposition is true, and 0 otherwise. In other words, our goal is to maximize the total number of correct predictions, or $\beta_{(p_z == y_z)} = 1$. To increase α of this objective function, several approaches can be chosen, as discussed previously in the related work in Section 2. These can be: (i) select and/or transform features [13, 17, 18]; (ii) select the best classifier [30, 32, 33]; or (iii) select the best ensemble of classifiers [39, 40, 42].

In this work, we choose a solution that includes the first and second approaches simultaneously, proposing a twofold transforming technique that boosts features $f \in F$ and applying them to the best classifier. This boosting is done in such a way to better distribute the features to the classes of interest in the N dimensional space. With such a procedure, we expect to maximize α when applying such boosted features to the best classifier for this task.

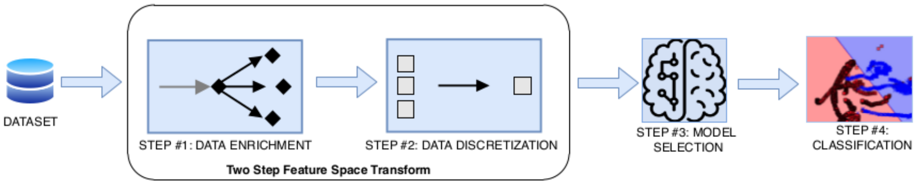

Fig. 2. Full pipeline of credit scoring systems including our proposed Two-Step Feature Space Transforming approach.

As can be seen in proposed model pipeline in Figure 2, our approach is composed of four main steps, described as follows:

1. **Data Enrichment:** a series of additional features $\hat{F}$ are added to the original ones in F, in order to include useful information for better credit scoring.
2. **Data Discretization:** once enriched, the features are now discretized to lie in a given range, which is defined in the context of experiments done in the *in-sample* part of the dataset.
3. **Model Selection:** chooses the model to use in the context of the credit score machine learning applications.
4. **Classification:** implements the classification algorithm to classify new instances $\hat{S}$ into reliable or unreliable.

We discuss each of the above-mentioned steps of our proposed method in the following subsections.

3.1 Data Enrichment

As discussed previously, several works in the literature have pointed out that transforming features can improve the data domain, thus benefiting any machine learning technique able to discriminate them into disjoint classes. One specific kind of features transformation is adding meta-features [60]. Such transformation is commonly used in a machine learning research branch called *meta-learning* [61]. Such additional features are composed of summarizing or reusing the existing ones, by calculating values such as the minimum, maximum, mean value, among others. Such values can be calculated at each vector domain, or considering all vectors in a matrix of features.

In our proposed method, we use these meta-features in order to balance the loss of information caused by the data heterogeneity issue present in credit scoring datasets, adding further data created to boost the characterization of features F into the *reliable* or *unreliable* classes of interest. Formally, given the set of features $F = \{f_1, f_2, \dots, f_N\}$, we add $MF = \{mf_{N+1}, mf_{N+2}, \dots, mf_{N+Z}\}$ new meta-features, obtaining the new set of features shown in Equation 2.

$$\hat{F} = \{f_1, f_2, \dots, f_N, mf_{N+1}, mf_{N+2}, \dots, mf_{N+Z}\}. \tag{2}$$

Therefore, we chose for our proposed method $Z = |MF| = 4$ or, in other words, we add to the original features four additional meta-features. These meta-features have been calculated feature vector-wise and are the following: *Minimum value* (*min*), *Maximum value* (*max*), *Mean* (*mean*), and *Standard Deviation* (*std*), then we have $MF = \{min, max, mean, std\}$. By adding more insightful data to the original feature set, this new process minimizes the pattern reduction effects that are normally present in the heterogeneous nature of credit scoring data. Such additional features are better formalized in a parameter u in Equation 3

$$\mu = \begin{cases} min = min(f_1, f_2, \dots, f_N) \\ max = max(f_1, f_2, \dots, f_N) \\ mean = \frac{1}{N} \sum_{n=1}^{N} (f_n) \\ std = \sqrt{\frac{1}{N-1} \sum_{n=1}^{N} (f_n - \bar{f})^2} \end{cases} \tag{3}$$

3.2 Data Discretization

The data discretization process is commonly used in machine learning algorithms as a way of data transform [62]. It focuses on transforming the features by dividing each of them into a discrete number that falls in independent intervals. It means that numerical features, being discrete or continuous, will be mapped to lie in one of these intervals, standardizing the whole set of original features. Such a procedure was proven to boost the performance of many machine learning models [63, 64].

Although the fact that, in one hand, the process of discretization comes with the drawback of filtering some sort of additional information gathered from the meta-features in the previous step of our method, it comes with the advantage of *understandability*, which comes from the conversion of the continuous space to a more limited (discrete) space [62], which guides a faster and precise learning [63]. Figure 3 shows one example of discretizing six feature values in the continuous range $\{0, \dots, 150\}$ into discrete values in the discrete range $\{0, 1, \dots, 15\}$.

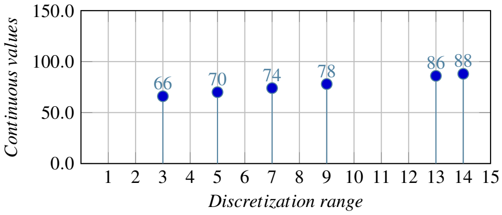

Fig. 3. Discretization process example of continuous features.

In our approach, each of the features $f \in F$ in the enriched $\hat{S}$ from the previous step are transformed through discretization. This is done to move the original continuous range to a defined discrete range $[0, 1, \dots, r] \in \mathbb{Z}$, where r is found experimentally as will be described later in this work. By defining the discretization procedure $f \xrightarrow{r} d$, we operate in order transform each $f \in F$ into one of the values in the discrete range of integers $d = [1, 2, \dots, r]$. Such a process reduces significantly the number of possible different patterns in each $f \in F$, as shown in Equation 4.

$$\{f_1, f_2, \dots, f_N\} \xrightarrow{r} \{d_1, d_2, \dots, d_N\}, \ \forall \ \hat{s} \in \hat{S} \tag{4}$$

3.3 Model Selection

The following step of the credit scoring pipeline chooses the model to be applied in preprocessed features by our TSFST approach. According to the u and r parameters from the enrichment and and discretization phases of our approach respectively, a new set of features $TSFST(S)$ is formalized as shown in Equation 5.

$$TSFST(S) = \begin{pmatrix} d_{1,1} & d_{1,2} & \dots & d_{1,N} & mf_{u(1,1)} & mf_{u(1,2)} & mf_{u(1,3)} & mf_{u(1,4)} \\ d_{2,1} & d_{2,2} & \dots & d_{2,N} & mf_{u(2,1)} & mf_{u(2,2)} & mf_{u(2,3)} & mf_{u(2,4)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ d_{X,1} & d_{X,2} & \dots & d_{X,N} & mf_{u(X,1)} & mf_{u(X,2)} & mf_{u(X,3)} & mf_{u(X,4)} \end{pmatrix} \tag{5}$$

Such features are used both in the training and evaluation steps of the model. The model chosen in our TSFST model is the Gradient Boosting (GB) algorithm [65]. We chose this algorithm mainly because it follows the idea of *boosting*. In other words, the classifier is composed initially of weak learning models that keep only observations these models successfully classified. Then, a new learner is created and trained on the set of data that was poorly classified before. Decision trees are usually used in GB.

Performance experiments done in the *in-sample* part of each dataset assess the choice of such a model, as will be discussed with further details later in this work.

3.4 Data Classification

The last step of our approach applies the new $TSFST$-based classifier in the evaluation (unknown) data, in order to maximize the α metric discussed in Equation 1. For that, we consider again S as the classified samples, which will be the training (or known) samples, but now we also define $\bar{S}$, which is a new set of unclassified (or unknown) samples to be evaluated in the $TSFST$ classifier. Such a procedure is done as shown in Algorithm 1.

Algorithm 1 TSFST classification pipeline

```
Input: cla=classifier, S=classified (training) instances, $\bar{S}$=unclassified instances, u=meta-features to calculate, r=upper
    bound of the discretization process.
Output: out=Classification of instances in $\bar{S}$
 1: procedure InstanceClassification(cla, S, $\bar{S}$, u, r)
 2:     $MF \leftarrow getMetaFeatures(S,u)$                    ▷ Step #1 (enrichment) of TSFST model in the training data
 3:     $S \leftarrow concat(S,MF)$                             ▷ Concat original data with meta features found
 4:     $\hat{S} = getDiscretizedFeatures(S,r)$                 ▷ Step #2 (discretization) of TSFP model in the training data
 5:     $model \leftarrow ClassifierTraining(alg,\hat{S})$      ▷ Classifier training using the TSFST transformed training data
 6:     $MF' \leftarrow getMetaFeatures(\bar{S},u)$             ▷ Repeat TSFST procedure in the testing samples
 7:     $\bar{S} \leftarrow concat(\bar{S},MF')$
 8:     $\hat{\bar{S}} = getDiscretizedFeatures(\bar{S},r)$
 9:     for each $\hat{\bar{s}} \in \hat{\bar{S}}$ do           ▷ Classifier evaluation in each TSFST transformed testing sample
10:         $c \leftarrow classify(model,\hat{\bar{s}})$
11:         $out.add(c)$
12:     end for
13:     return out
14: end procedure
```

In the *step 1* of this algorithm, the following parameters are used as input: (i) the classification algorithm cla to be trained and tested using the $TSFST$ feature set; (ii) the training classified data S in its original format; (iii) the new instances to be classified $\bar{S}$ also in their original format; and (iv) $TSFST$ parameters, such as the meta-features u to be used in the enhancement step and the upper bound r to be used in the discretization step. The data transformation related to our $TSFST$ approach is performed for sets of training data S and testing data $\bar{S}$ at *steps 2-4* and *6-8* respectively, and the transformed data $\hat{S}$ of the training set trains the model cla at *step 5*. The classification process is performed at *steps 9-12* for each instance $\hat{\bar{s}}$ in the transformed testing samples set $\hat{\bar{S}}$, with final classifications stored in the *out* vector. At the end of the process, classification labels generated by our proposed boosted classifier are returned by the algorithm at *step 13*.

In order to evaluate the impact in terms of *response-time* of the proposed approach in a *real-time scoring system*, we evaluated the asymptotic time complexity of the proposed Algorithm 1 in terms of *big-O notation*. According to the formal notation provided in Section 3, we can do the following observations:

(i) the complexity of the steps *2-4* and *6-8* is $O(N)$, since our $TSFST$ data transformation performs a discretization of the original feature values F at $N \cdot |F|$ times, after several meta-features are added to them;

(ii) the complexity of the step *5* depends on the adopted algorithms, which in our case is the *Gradient Boosting*, an algorithm characterized by a training complexity of $O(N \cdot |F| \cdot \pi)$, where π denotes the number of used trees;

(iii) the complexity of the cycle in the *steps 9-12* is $O(N^2)$, since it involves the prediction complexity of Gradient Boosting (*i.e.*, $O(N \cdot \pi)$) for each instance in the set $\bar{\hat{S}}$.

On the basis of the aforementioned observations, we can express the asymptotic time complexity of the algorithm as $O(N^2)$, an asymptotic time complexity that can be reduced by distributing the process over different machines, by employing large scale distributed computing models (*e.g. MapReduce* [66, 67]).

4 Experimental Setup

In this section, we discuss all the experimental environment we considered to perform credit scoring experiments. In the following subsections we discuss: (i) the datasets considered; (ii) the metrics used to assess performances; (iii) methodology used to evaluate the methods; and (iv) models considered and implementation aspects of the proposed TSFST method.

4.1 Datasets

We consider three datasets to evaluate our approach: (i) the *Australian Credit Approval* (AC); (ii) the *German Credit* (GC); and (iii) the *Default of Credit Card Clients* (DC). These datasets represent three real-world data, characterized by a different number of instances and features, and also a different level of data unbalance. Such datasets are publicly available[2], and previous works the literature have used them to benchmark their approaches. Such data distribution is described in Table 1.

Table 1. Datasets information

Dataset name	Total instances $\lvert S\rvert$	Reliable instances $\lvert S^+\rvert$	Unreliable instances $\lvert S^-\rvert$	Number of features $\lvert F\rvert$	Number of classes $\lvert C\rvert$	Reliable/unreliable instances (%)
AC	690	307	383	14	2	44.50 / 55.50
GC	1,000	700	300	21	2	70.00 / 30.00
DC	30,000	23,364	6,636	23	2	77.88 / 22.12

The *AC* dataset is composed of *690* instances, of which *307* classified as *reliable* (44.50%) and *387* classified as *unreliable* (55.50%), and each instance is composed of *14* features, as detailed in Table 2. For data confidentiality reasons, feature names and values have been changed to meaningless symbols.

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

Table 2. Features of AC Dataset

Field	Type	Field	Type
01	*Categorical field*	*08*	*Categorical field*
02	*Continuous field*	*09*	*Categorical field*
03	*Continuous field*	*10*	*Continuous field*
04	*Categorical field*	*11*	*Categorical field*
05	*Categorical field*	*12*	*Categorical field*
06	*Categorical field*	*13*	*Continuous field*
07	*Continuous field*	*14*	*Continuous field*

The *GC* dataset is composed of *1,000* instances, of which *700* classified as *reliable* (70.00%) and *300* classified as *unreliable* (30.00%), and each instance is composed of *20* features, as detailed in Table 3 below.

Table 3. Features of GC Dataset [19]

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

Finally, the *DC* dataset is composed of *30,000* instances, of which *23,364* classified as *reliable* (77.88%) and *6,636* classified as *unreliable* (22.12%), and each instance is composed of *23* features, as detailed in Table 4.

4.2 Metrics

The literature in machine learning has been investigating several different metrics through the last decades, in order to find criteria suitable for a correct performance evaluation of credit scoring models [68]. In [69], several metrics based on confusion matrix were considered, such as *Accuracy*, *True Positive Rate (TPR)*, *Specificity*, or the *Matthews Correlation Coefficient* (MCC). Authors in [70] choose metrics based on the error analysis, such as the *Mean Square Error* (MSE), the *Root Mean Square Error* (RMSE) or the *Mean Absolute Error* (MAE). Finally there are also some works like in [71] that evaluate metrics based on the *Receiver Operating Characteristic* (ROC) curve, such as the *Area Under the ROC Curve* (AUC). Considering that some of these metrics do not

Table 4. Features of DC Dataset [19]

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

work well with unbalanced datasets, like, for example, the metrics based on on the confusion matrix, many works in literature have been addressing the problem of unbalanced datasets by adopting more than one metric to correctly evaluate their results [72].

In our work, we choose to follow that direction, adopting a hybrid strategy to measure the performance of the tested approaches. Our metrics chosen are based on confusion matrix results and ROC curve calculation and are described in the following:

True Positive Rate Given *TP* the number of instances correctly classified as *unreliable*, and *FN* the number of *unreliable* instances wrongly classified as *reliable*, the True Positive Rate (TPR) measures the rate of correct classification of *unreliable* users in a credit scoring model *m* in any test set *S*, as can be shown in Equation 6:

$$TPR_m(S) = \frac{TP}{(TP+FN)}. \tag{6}$$

Such a metric, also known as *Sensitivity*, indicates the proportion of instances from the positive class that are correctly classified by an evaluation model, according to the different classes of a given problem [36].

Matthews Correlation Coefficient The *Matthews Correlation Coefficient* (MCC) is suitable for unbalanced problems [73, 74] as it does a balanced evaluation of performance. Its formalization, shown in Equation 7, results in a value in the range $[-1,+1]$, with $+1$ when all the classifications are correct and -1 otherwise, whereas 0 indicates the performance related to a random predictor. The MCC of a model *m* that classifies any new set *S* is calculated as:

$$MCC_m(S) = \frac{(TP \cdot TN) - (FP \cdot FN)}{\sqrt{(TP+FP) \cdot (TP+FN) \cdot (TN+FP) \cdot (TN+FN)}}. \tag{7}$$

It should be observed that MCC can be seen as a discretization of the *Pearson correlation* [75] for binary variables.

AUC The *Area Under the Receiver Operating Characteristic* curve (AUC) represents a reliable metric for the evaluation of the performance related to a *credit scoring* model [76, 69]. To calculate such a metric, the Receiver Operating Characteristic (ROC) curve is firstly built by plotting the Sensitivity and the False Positive Rate at different classification thresholds and, finally, the area under that curve is calculated.

The AUC metric returns a value in the range $[0,1]$, where 1 denotes the best performance. AUC is a metric able to assess the predictive capability of an evaluation model, even in the presence of unbalanced data [76].

Performance This metric is used in the context of our work in order to compare the several classifiers performance. It is calculated by summarizing all metrics presented before in all datasets in just one final metric. Considering ζ the number of datasets in the experiments, the final performance P of a method m in any set S is calculated as:

$$P_m(S) = \frac{\sum_{z=1}^{\zeta} \frac{TPR(S)_z + AUC(S)_z + MCC(S)_z}{3}}{\zeta} \tag{8}$$

We also calculate the performance for a model m in a single dataset z as follows:

$$P_{m,z}(S) = \frac{TPR(S)_z + AUC(S)_z + MCC(S)_z}{3}. \tag{9}$$

Such a metric also returns a value in the range $[0,1]$, as it comes from three metrics in the same values range.

4.3 Experimental Methodology

We choose an evaluation criterion that divides each dataset into two pieces: (i) the (*in-sample*), used to identify the best model to use to compare with and without our $TSFST$ method and the best parameters of our method; and (ii) the (*out-of-sample*), which we use for final evaluation. Such a strategy allows the correct evaluation of the results by preventing the algorithm from yielding results biased by over-fitting [20]. Such an evaluation procedure has also been followed by other works in the literature [77].

For this reason, each of the adopted datasets has been divided into an *in-sample* part, containing 80% of the dataset, and an *out-of-sample* part, containing the remaining 20%. We opt for such a data split to follow some works in literature [78–80]. In addition, with the aim to further reduce the impact of the data dependency, we have adopted a *k-fold cross-validation* criterion (k=*5*) inside each *in-sample* subset. Information about these subsets are reported in Table 5.

4.4 Considered Models and Implementation Details

In order to evaluate the qualities of the proposed transforming approach, we consider several models, represented by machine learning classifiers, in order to select the best one to be used in our approach and to compare its performance before and after our $TSFST$ approach is considered in that model. For this task, we have taken into account

Table 5. In-sample and out-of-sample datasets information

Dataset	In-sample				Out-of-sample			
name	Reliable	%	Unreliable	%	Reliable	%	Unreliable	%
AC	124	45.0	152	55.0	125	45.5	150	54.5
GC	292	73.0	108	27.0	268	67.2	131	32.8
DC	9307	77.5	2693	22.5	9404	78.4	2595	21.6

the following machine learning algorithms widely used in the credit scoring literature: (i) *Gradient Boosting* (GB) [81]; (ii) *Adaptive Boosting* (AD) [82]; (iii) *Random Forests* (RF) [83]; (iv) *Multilayer Perceptron* (MLP) [84]; and (v) *Decision Tree* (DT) [85].

The code related to the experiments was created with *Python* using the *scikit-learn*[3] library. 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 $bins = \{0, 1, \ldots, r-2, r-1\}$, where r is calculated experimentally (we show how we find r later in this section). In order to keep the experiments reproducible, we have fixed the seed of the *pseudo-random number generator* to *1*. In our proposed method, we fixed $|u| = 4$, calculating the four meta-features described in section 3.

5 Experimental Results

To validate our proposed approach, we performed an extensive series of experiments. We classified the experiments as follows:

1. Experiments performed in the *in-sample* part of each dataset: used to assess the benefits of our approach according to several configurations of parameters in credit scoring. For that, we average results of a five-fold cross validation.
2. Experiments performed in the *out-of-sample* part of each dataset: used to compare our approach with some baselines in real credit scoring. For this experiment, we used the *in-sample* part to train and unknown *out-of-sample* data to test.

5.1 In-sample Experiments

In this set of experiments which uses cross validation in the *in-sample* part of each dataset, we choose the following evaluation scenarios:

1. We evaluate the advantages, in terms of performance, of the adoption of some canonical data preprocessing techniques as input to our data transform approach;
2. We report results in order to find the best parameter r of the discretization step of our proposed approach.

We discuss the results of these experiments as follows.

[3] http://scikit-learn.org

Preprocessing Benchmarking The literature has been strongly suggesting the use of several preprocessing techniques [86, 87] to organize better the data distribution as to training better and boosting machine learning algorithms performance. One straightforward way of doing this is to put feature values in the same range of values, therefore, we decided to verify the performance improvement related to the adoption of two largely used preprocessing methods: *normalization* and *standardization*. In the normalization process, each feature $f \in F$ is scaled into the range $[0, 1]$, whereas the standardization (also known as *Z-score normalization*) re-scales the feature values in such a way that they assume the properties of a *Gaussian distribution*, with mean equals to zero and standard deviation equals to one.

Performance results are shown in Table 6, which reports the mean performance of the five fold cross validation (*i.e.*, related to the *Accuracy*, *MCC*, and *AUC* metrics) measured in all datasets and all algorithms after the application of the aforementioned methods of data preprocessing, along to that measured without any data preprocessing. Premising that the best performances are highlighted in bold, and all the experiments involve only the *in-sample* part of each dataset, on the basis of the obtained results, we can do the following observations:

- the data *normalization* and *standardization* processes do not lead toward significant improvements, since *7* times out of *15* (against *4* out of *15* and *4* out of *15*) we obtain a better performance without using any canonical data preprocessing.
- in the context of the experiments performed without a data preprocessing, *Gradient Boosting* (GB) shows to be the best algorithm between those taken into account, since it gets the better mean performance on all datasets (*i.e.*, *0.6574* against *0.6431* of *ADA*, *0.6388* of *RFA*, *0.5317* of *MLP*, and *0.6147* of *DTC*);
- for the aforementioned reasons we decided to not apply any method of data preprocessing, using *Gradient Boosting* as reference algorithm to evaluate our approach performance.

Table 6. Average performance with preprocessing

Algorithm	Dataset	Non-preprocessed	Normalized	Standardized
GBC	AC	**0.8018**	0.8005	0.8000
ADA	AC	**0.7495**	0.6735	0.7179
RFA	AC	0.8011	0.7505	**0.8120**
MLP	AC	0.5225	**0.8079**	0.8073
DTC	AC	0.7662	0.7093	**0.7690**
GBC	GC	0.5614	0.5942	**0.6007**
ADA	GC	0.5766	**0.6246**	0.5861
RFA	GC	0.5540	**0.5614**	0.5579
MLP	GC	**0.6114**	0.5649	0.5589
DTC	GC	**0.5796**	0.5456	0.5521
GBC	DC	**0.6087**	0.5442	0.6076
ADA	DC	**0.6031**	0.5361	0.5980
RFA	DC	**0.5613**	0.4909	0.5586
MLP	DC	0.4613	**0.6177**	0.5985
DTC	DC	0.4982	0.4572	**0.5185**
Best cases		7	4	4

Discretization Range Experiments The goal related to this set of experiments is the definition of the optimal range of discretization r to use in the context of the selected classification algorithm. Figure 4 reports the obtained results in terms of the performance metric for each dataset. Such results indicate *106*, *25*, and *187* as optimal r values for the *AC*, *GC*, and *DC* datasets respectively.

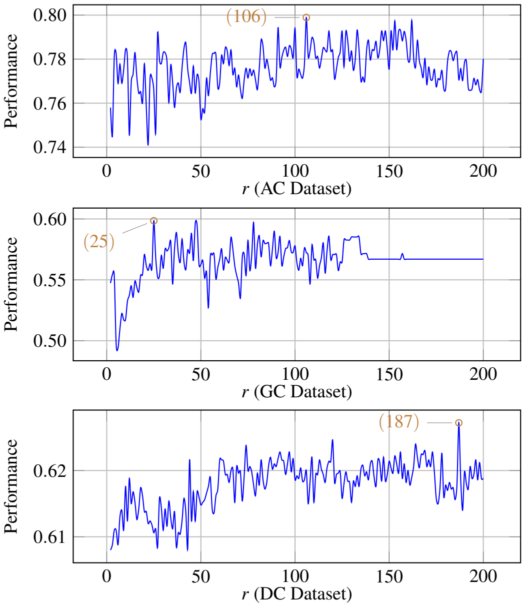

Fig. 4. In-sample r Value Definition

5.2 Out-of-sample Experiments

Now that we found the discretization parameter of our proposed approach, we focus our attention on discussing the more realistic scenario of credit scoring. For that we perform testing on unseen *out-of-sample* part of the dataset, comparing the effectiveness of such approach with other competitors. We apply the algorithm and the r value detected through the previous experiments in order to evaluate the capability of the proposed $TSFST$ model with regard to a canonical data model (*GB*), based on the original feature space. The analysis of the experimental results shown in Figure 5 leads us toward the following considerations:

1. as shown in Figure 5, the proposed *TSFST* model outperforms its competitor in terms of *TPR*, *MCC*, and *AUC*, in all the datasets, except for a single case (*i.e.*, *TPR* in the *DC* dataset);

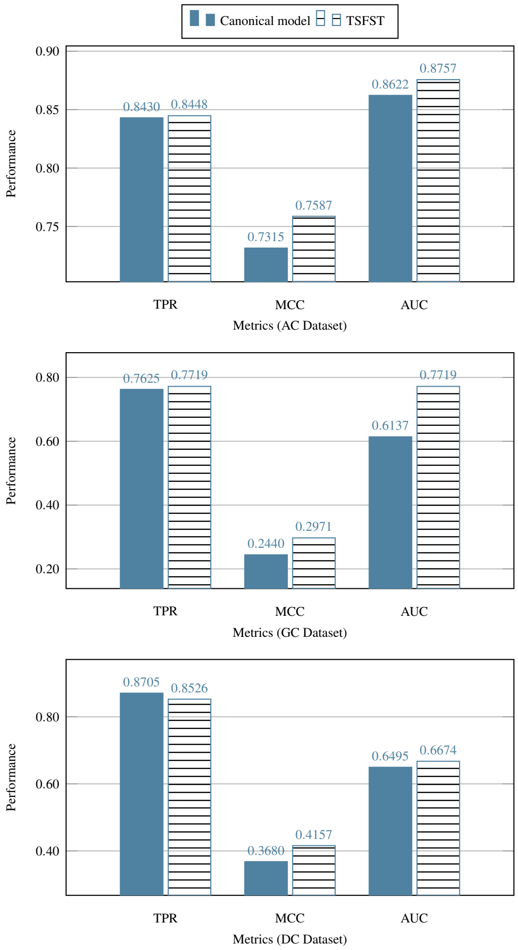

Fig. 5. Out-of-sample classification results, comparing *GB* with and without our proposed $TSFST$

2. although it does not outperform its competitor in terms of *TPR* in the *DC* dataset, its better performance in terms of *MCC*, and *AUC* indicates that the best competitor value has produced a greater number of false positives and/or negatives;
3. for the same reason above, the better performance of our approach in terms of *TPR* can not be considered a side effect related to the increase in terms of *false positive rate* and/or *false negative rate*, since we also outperform the competitor in terms of *MCC* and *AUC*;
4. considering that the *AC*, *GC*, and *DC* datasets are different in terms of data size, level of balancing, and number of features, the obtained results prove the effectiveness of the proposed approach in heterogeneous credit scoring contexts;
5. the adopted validation method, based on the *in-sample/out-of-sample* strategy, combined with a *k-fold cross-validation* criterion, proves the real effectiveness of the proposed approach, since the performance has been evaluated on data never used before, avoiding over-fitting;

In summary, the experimental results have proved that the proposed approach improves the performance of a machine learning algorithm in the credit scoring context, allowing us its exploitation in several state-of-the-art approaches.

6 Conclusion

The growth of credit in economy nowadays has required scoring tools in order to allow reliable loans in complex scenarios. Such an opportunity has led an increasing number of research focusing on proposing new methods and strategies. Notwithstanding, similarly to other applications such as fraud detection or intrusion detection, a natural imbalanced distribution of data among classes of interest is commonly found in credit scoring datasets. Such a limitation raises issues in models that could be biased in always classifying samples as the class they have more access in their training. In a such scenario, a slight performance improvement of a classification model produces enormous advantages, which in our case are related to the reduction of financial losses.

In this work, we report a new research inspired by our previous findings [19]. We propose a method composed of a twofold transforming process in credit scoring data, which acts by transforming the features through adding meta-features and also discretizing the resulting new feature space. From our experiments, we could raise the following conclusions; (i) our approach boosts classifiers that use original features; (ii) it is able to improve the performance of the machine learning algorithms; (iii) our approach fits better in boosted-based classifiers such as gradient boosting. Such findings open new perspectives for the definition of more effective credit scoring solutions, considering that many state-of-the-art approaches are based on machine learning algorithms.

As future work, we envision to validate the performance of the proposed data model in the context of credit scoring solutions that implement more than a single machine learning algorithm, such as, for example, homogeneous and heterogeneous ensemble approaches. By achieving good results in this new modelling scenario, we believe we can achieve a more real world solution for credit scoring.

References

1. Economics, T.: Euro area consumer credit. https://tradingeconomics.com/euro-area/consumer-credit?continent=europe (2019)
2. Economics, T.: Euro area consumer spending. https://tradingeconomics.com/euro-area/consumer-spending?continent=europe (2019)
3. Siddiqi, N.: Intelligent credit scoring: Building and implementing better credit risk scorecards. John Wiley & Sons (2017)
4. Mester, L.J., et al.: Whats the point of credit scoring? Business review **3** (1997) 3–16
5. Hassan, M.K., Brodmann, J., Rayfield, B., Huda, M.: Modeling credit risk in credit unions using survival analysis. International Journal of Bank Marketing **36**(3) (2018) 482–495
6. Dal Pozzolo, A., Caelen, O., Le Borgne, Y.A., Waterschoot, S., Bontempi, G.: Learned lessons in credit card fraud detection from a practitioner perspective. Expert systems with applications **41**(10) (2014) 4915–4928
7. Saia, R., Carta, S., et al.: A frequency-domain-based pattern mining for credit card fraud detection. In: IoTBDS. (2017) 386–391
8. Saia, R.: A discrete wavelet transform approach to fraud detection. In: International Conference on Network and System Security, Springer (2017) 464–474
9. Rodda, S., Erothi, U.S.R.: Class imbalance problem in the network intrusion detection systems. In: 2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT), IEEE (2016) 2685–2688
10. Saia, R., Carta, S., Recupero, D.R.: A probabilistic-driven ensemble approach to perform event classification in intrusion detection system. In: KDIR, SciTePress (2018) 139–146
11. Khemakhem, S., Ben Said, F., Boujelbene, Y.: Credit risk assessment for unbalanced datasets based on data mining, artificial neural network and support vector machines. Journal of Modelling in Management **13**(4) (2018) 932–951
12. Haixiang, G., Yijing, L., Shang, J., Mingyun, G., Yuanyue, H., Bing, G.: Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications **73** (2017) 220–239
13. Saia, R., Carta, S.: A linear-dependence-based approach to design proactive credit scoring models. In: KDIR. (2016) 111–120
14. Saia, R., Carta, S.: Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach. In: SECRYPT, SciTePress (2017) 335–342
15. Saia, R., Carta, S.: Introducing a vector space model to perform a proactive credit scoring. In: International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management, Springer (2016) 125–148
16. Saia, R., Carta, S.: An entropy based algorithm for credit scoring. In: International Conference on Research and Practical Issues of Enterprise Information Systems, Springer (2016) 263–276
17. Saia, R., Carta, S.: A fourier spectral pattern analysis to design credit scoring models. In: Proceedings of the 1st International Conference on Internet of Things and Machine Learning, ACM (2017) 18
18. 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, ACM (2018) 176–180
19. Saia, R., Carta, S., Recupero, D.R., Fenu, G., Saia, M.: A discretized enriched technique to enhance machine learning performance in credit scoring. In: KDIR, ScitePress (2019) 202–213
20. Hawkins, D.M.: The problem of overfitting. Journal of chemical information and computer sciences **44**(1) (2004) 1–12

21. Henrique, B.M., Sobreiro, V.A., Kimura, H.: Literature review: Machine learning techniques applied to financial market prediction. Expert Systems with Applications (2019)
22. Crook, J.N., Edelman, D.B., Thomas, L.C.: Recent developments in consumer credit risk assessment. European Journal of Operational Research **183**(3) (2007) 1447–1465
23. Thomas, L.C.: A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. International Journal of Forecasting **16**(2) (2000) 149 – 172
24. Chen, B., Zeng, W., Lin, Y.: Applications of artificial intelligence technologies in credit scoring: A survey of literature. In: International Conference on Natural Computation (ICNC). (Aug 2014) 658–664
25. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research. European Journal of Operational Research **247**(1) (2015) 124–136
26. Fang, F., Chen, Y.: A new approach for credit scoring by directly maximizing the kolmogorovsmirnov statistic. Computational Statistics & Data Analysis **133** (2019) 180 – 194
27. Sohn, S.Y., Kim, D.H., Yoon, J.H.: Technology credit scoring model with fuzzy logistic regression. Applied Soft Computing **43** (2016) 150–158
28. Khemais, Z., Nesrine, D., Mohamed, M., et al.: Credit scoring and default risk prediction: A comparative study between discriminant analysis & logistic regression. International Journal of Economics and Finance **8**(4) (2016) 39
29. Laha, A.: Developing credit scoring models with som and fuzzy rule based k-nn classifiers. In: IEEE International Conference on Fuzzy Systems. (July 2006) 692–698
30. Zhang, X., Yang, Y., Zhou, Z.: A novel credit scoring model based on optimized random forest. In: IEEE Annual Computing and Communication Workshop and Conference (CCWC). (Jan 2018) 60–65
31. Maldonado, S., Peters, G., Weber, R.: Credit scoring using three-way decisions with probabilistic rough sets. Information Sciences (2018)
32. Zhu, B., Yang, W., Wang, H., Yuan, Y.: A hybrid deep learning model for consumer credit scoring. In: International Conference on Artificial Intelligence and Big Data (ICAIBD). (May 2018) 205–208
33. Tian, Y., Yong, Z., Luo, J.: A new approach for reject inference in credit scoring using kernel-free fuzzy quadratic surface support vector machines. Applied Soft Computing **73** (2018) 96 – 105
34. de Castro Vieira, J.R., Barboza, F., Sobreiro, V.A., Kimura, H.: Machine learning models for credit analysis improvements: Predicting low-income families default. Applied Soft Computing **83** (2019) 105640
35. Liu, C., Huang, H., Lu, S.: Research on personal credit scoring model based on artificial intelligence. In: International Conference on Application of Intelligent Systems in Multimodal Information Analytics, Springer (2019) 466–473
36. Bequé, A., Lessmann, S.: Extreme learning machines for credit scoring: An empirical evaluation. Expert Systems with Applications **86** (2017) 42–53
37. Pasila, F.: Credit scoring modeling of indonesian micro, small and medium enterprises using neuro-fuzzy algorithm. In: IEEE International Conference on Fuzzy Systems. (June 2019) 1–6
38. Neagoe, V., Ciotec, A., Cucu, G.: Deep convolutional neural networks versus multilayer perceptron for financial prediction. In: International Conference on Communications (COMM). (June 2018) 201–206
39. Ala'raj, M., Abbod, M.F.: A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Systems with Applications **64** (2016) 36–55
40. Tripathi, D., Edla, D.R., Cheruku, R.: Hybrid credit scoring model using neighborhood rough set and multi-layer ensemble classification. Journal of Intelligent & Fuzzy Systems **34**(3) (2018) 1543–1549

41. Lpez, J., Maldonado, S.: Profit-based credit scoring based on robust optimization and feature selection. Information Sciences **500** (2019) 190 – 202
42. Guo, S., He, H., Huang, X.: A multi-stage self-adaptive classifier ensemble model with application in credit scoring. IEEE Access **7** (2019) 78549–78559
43. Zhang, H., He, H., Zhang, W.: Classifier selection and clustering with fuzzy assignment in ensemble model for credit scoring. Neurocomputing **316** (2018) 210 – 221
44. Feng, X., Xiao, Z., Zhong, B., Qiu, J., Dong, Y.: Dynamic ensemble classification for credit scoring using soft probability. Applied Soft Computing **65** (2018) 139 – 151
45. Tripathi, D., Edla, D.R., Kuppili, V., Bablani, A., Dharavath, R.: Credit scoring model based on weighted voting and cluster based feature selection. Procedia Computer Science **132** (2018) 22 – 31 International Conference on Computational Intelligence and Data Science.
46. Vedala, R., Kumar, B.R.: An application of naive bayes classification for credit scoring in e-lending platform. In: International Conference on Data Science Engineering (ICDSE). (July 2012) 81–84
47. Sewwandi, D., Perera, K., Sandaruwan, S., Lakchani, O., Nugaliyadde, A., Thelijjagoda, S.: Linguistic features based personality recognition using social media data. In: 2017 6th National Conference on Technology and Management (NCTM). (Jan 2017) 63–68
48. Sun, X., Liu, B., Cao, J., Luo, J., Shen, X.: Who am i? personality detection based on deep learning for texts. In: IEEE International Conference on Communications (ICC). (May 2018) 1–6
49. Boratto, L., Carta, S., Fenu, G., Saia, R.: Using neural word embeddings to model user behavior and detect user segments. Knowledge-based systems **108** (2016) 5–14
50. Zhao, Y., Shen, Y., Huang, Y.: Dmdp: A dynamic multi-source default probability prediction framework. Data Science and Engineering **4**(1) (2019) 3–13
51. López, R.F., Ramon-Jeronimo, J.M.: Modelling credit risk with scarce default data: on the suitability of cooperative bootstrapped strategies for small low-default portfolios. JORS **65**(3) (2014) 416–434
52. Lika, B., Kolomvatsos, K., Hadjiefthymiades, S.: Facing the cold start problem in recommender systems. Expert Syst. Appl. **41**(4) (2014) 2065–2073
53. Son, L.H.: Dealing with the new user cold-start problem in recommender systems: A comparative review. Inf. Syst. **58** (2016) 87–104
54. Fernández-Tobías, I., Tomeo, P., Cantador, I., Noia, T.D., Sciascio, E.D.: Accuracy and diversity in cross-domain recommendations for cold-start users with positive-only feedback. In Sen, S., Geyer, W., Freyne, J., Castells, P., eds.: Proceedings of the 10th ACM Conference on Recommender Systems, Boston, MA, USA, September 15-19, 2016, ACM (2016) 119–122
55. Attenberg, J., Provost, F.J.: Inactive learning?: difficulties employing active learning in practice. SIGKDD Explorations **12**(2) (2010) 36–41
56. Thanuja, V., Venkateswarlu, B., Anjaneyulu, G.: Applications of data mining in customer relationship management. Journal of Computer and Mathematical Sciences Vol **2**(3) (2011) 399–580
57. He, H., Garcia, E.A.: Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. **21**(9) (2009) 1263–1284
58. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. Intelligent Data Analysis **6**(5) (2002) 429–449
59. Chatterjee, A., Segev, A.: Data manipulation in heterogeneous databases. ACM SIGMOD Record **20**(4) (1991) 64–68
60. Giraud-Carrier, C., Vilalta, R., Brazdil, P.: Introduction to the special issue on meta-learning. Machine learning **54**(3) (2004) 187–193
61. Vilalta, R., Drissi, Y.: A perspective view and survey of meta-learning. Artificial intelligence review **18**(2) (2002) 77–95

62. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An enabling technique. Data mining and knowledge discovery **6**(4) (2002) 393–423
63. García, S., Ramírez-Gallego, S., Luengo, J., Benítez, J.M., Herrera, F.: Big data preprocessing: methods and prospects. Big Data Analytics **1**(1) (2016) 9
64. Wu, X., Kumar, V.: The top ten algorithms in data mining. CRC press (2009)
65. Breiman, L.: Random forests. Machine Learning **45**(1) (2001) 5–32
66. Hashem, I.A.T., Anuar, N.B., Gani, A., Yaqoob, I., Xia, F., Khan, S.U.: Mapreduce: Review and open challenges. Scientometrics **109**(1) (2016) 389–422
67. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1) (2008) 107–113
68. Chen, N., Ribeiro, B., Chen, A.: Financial credit risk assessment: a recent review. Artificial Intelligence Review **45**(1) (2016) 1–23
69. Powers, D.: Evaluation: From precision, recall and f-factor to roc, informedness, markedness & correlation. Mach. Learn. Technol. **2** (01 2008)
70. Chai, T., Draxler, R.R.: Root mean square error (rmse) or mean absolute error (mae)?–arguments against avoiding rmse in the literature. Geoscientific model development **7**(3) (2014) 1247–1250
71. Huang, J., Ling, C.X.: Using auc and accuracy in evaluating learning algorithms. IEEE Transactions on knowledge and Data Engineering **17**(3) (2005) 299–310
72. Jeni, L.A., Cohn, J.F., De La Torre, F.: Facing imbalanced data–recommendations for the use of performance metrics. In: 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction, IEEE (2013) 245–251
73. Luque, A., Carrasco, A., Martín, A., de las Heras, A.: The impact of class imbalance in classification performance metrics based on the binary confusion matrix. Pattern Recognition **91** (2019) 216–231
74. Boughorbel, S., Jarray, F., El-Anbari, M.: Optimal classifier for imbalanced data using matthews correlation coefficient metric. PloS one **12**(6) (2017) e0177678
75. Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. In: Noise reduction in speech processing. Springer (2009) 1–4
76. Abellán, J., Castellano, J.G.: A comparative study on base classifiers in ensemble methods for credit scoring. Expert Systems with Applications **73** (2017) 1–10
77. Rapach, D.E., Wohar, M.E.: In-sample vs. out-of-sample tests of stock return predictability in the context of data mining. Journal of Empirical Finance **13**(2) (2006) 231–247
78. Cleary, S., Hebb, G.: An efficient and functional model for predicting bank distress: In and out of sample evidence. Journal of Banking & Finance **64** (2016) 101–111
79. Adhikari, R.: A neural network based linear ensemble framework for time series forecasting. Neurocomputing **157** (2015) 231–242
80. Tamadonejad, A., Abdul-Majid, M., Abdul-Rahman, A., Jusoh, M., Tabandeh, R.: Early warning systems for banking crises? political and economic stability. Jurnal Ekonomi Malaysia **50**(2) (2016) 31–38
81. Chopra, A., Bhilare, P.: Application of ensemble models in credit scoring models. Business Perspectives and Research **6**(2) (2018) 129–141
82. Xia, Y., Liu, C., Li, Y., Liu, N.: A boosted decision tree approach using bayesian hyper-parameter optimization for credit scoring. Expert Systems with Applications **78** (2017) 225–241
83. Malekipirbazari, M., Aksakalli, V.: Risk assessment in social lending via random forests. Expert Systems with Applications **42**(10) (2015) 4621–4631
84. Luo, C., Wu, D., Wu, D.: A deep learning approach for credit scoring using credit default swaps. Engineering Applications of Artificial Intelligence **65** (2017) 465–470

85. Damrongsakmethee, T., Neagoe, V.E.: Principal component analysis and relieff cascaded with decision tree for credit scoring. In: Computer Science On-line Conference, Springer (2019) 85–95
86. Ghodselahi, A.: A hybrid support vector machine ensemble model for credit scoring. International Journal of Computer Applications **17**(5) (2011) 1–5
87. Wang, C.M., Huang, Y.F.: Evolutionary-based feature selection approaches with new criteria for data mining: A case study of credit approval data. Expert Systems with Applications **36**(3) (2009) 5900–5908