On combining dynamic selection, sampling, and pool generators for credit scoring ⋆ ⋆⋆

Leopoldo Melo Junior[1], Franco Maria Nardini[2], Chiara Renso[2], and Jose Antonio Macedo[1]

[1] Federal University of Ceará, Fortaleza, Brazil
[2] ISTI-CNR, Pisa, Italy

Abstract. The profitability of the banks highly depends on the models used to decide on the customer's loans. State of the art credit scoring models are based on machine learning methods. These methods need to cope with the problem of imbalanced classes since credit scoring datasets usually contain many paid loans and few not paid ones (defaults). Recently, dynamic selection approaches combined with pre-processing techniques have been evaluated for imbalanced datasets. However, previous works only evaluate oversampling techniques combined with bagging pool generator ensembles. For this reason, we propose to combine different dynamic selection, preprocessing and pool generation techniques. We assess the prediction performance by using four public real-world credit scoring datasets with different levels of imbalanced ratio and four evaluation measures. Experimental results show that KNORA-Union dynamic selection technique combined with Balanced Random Forest improves the classification performance concerning the static ensemble for all levels of imbalance ratio.

Keywords: Credit scoring · Imbalanced datasets · Dynamic classification · ensemble pool generators.

1 Introduction

Credit offer is a key activity for banks that aim at improving their profitability and competitiveness. Small improvements in the default prediction imply high profits to the financial institutions [13]. However, the decision of allowing a loan to a customer is complex and risky because it requires an accurate default prediction that should protect banks from financial losses, especially during the financial crises. Thomas et al. [23] pointed out several aspects affecting the default rate over time, such as the cost of the money (interest rate), the supply and demand for credit, the state of the economy, and the cyclical variations in credit over time. These aspects, and the data availability and accuracy make the default prediction much harder than other domain-specific classification problems. New methods and techniques are required to cope with these

⋆ Research supported by FUNCAP Brazilian funding agency under project number FUNCAP SPU 8789771/2017.

⋆⋆ This work has been supported by the MASTER project that has received funding from the European Unions Horizon 2020 research and innovation programme under the Marie Slodowska-Curie grant agreement N. 777695.

problems, supporting the implementation of accurate loan models, also called credit scoring models, that can guarantee a low percentage of defaults.

Available historical loan data creates an excellent opportunity to take advantage of trending machine learning methods for building accurate credit scoring models. However, real credit scoring datasets are usually imbalanced. They are called low default portfolios (LDP) [28], since they are highly skewed and with a low default rate.

In the past few decades, researchers have attempted to optimize the predictive performance in imbalance ratio data. According to Haixiang et al. [11], the two most used approaches are *Resampling*, a kind of preprocessing technique that changes the class distribution of the training set, and *Ensemble methods*, which could combine several base classifiers, sampling, and cost-sensitive approaches. This paper evaluates the combination of Resampling and Ensemble methods in imbalanced credit scoring datasets.

Ensemble methods are justified by several theoretical [6] and empirical [18] studies which demonstrate the advantages of Multiple Classifier System (MCS) over individual classifier models. They are widely used to solve many real-world problems, including credit scoring [17, 26] and class imbalance [8]. An MCS is typically composed of three phases: Pool Generation, Selection, and Integration [4]. This paper evaluates several techniques for the two first phases of an MCS.

Although several papers evaluate the prediction performance of classification approaches to credit scoring datasets [9, 7, 14, 22, 24, 1, 25, 3, 26, 2, 17], to the best of our knowledge an investigation of the combination of preprocessing approaches, dynamic selection techniques, and pool generators ensembles is missing. In this paper, we perform an empirical evaluation of the combination of these techniques to find improvements in credit scoring classification. Specifically, we seek to answer the following research questions related to the credit scoring problem: **RQ1)** Which preprocessing technique generates the best dataset to calibrate the dynamic selection approach?, **RQ2)** Is the combination of dynamic selection and preprocessing techniques better than the single application of a preprocessing approach?, **RQ3)** Does the imbalance level influence the prediction performance of the combinations?.

This work is partially motivated by the outstanding results achieved recently by the dynamic selection techniques [20, 4]. Most of these approaches are based on k-NN classifier to evaluate the local competence of each base classifier. However, the k-NN is biased to consider more the majority class. That means that the simple application of a dynamic selection technique in an imbalanced dataset produces poor results.

We measure the performance of the approaches using four common measures used in credit scoring predictions: Area Under the ROC Curve (AUC), F1-score, G-mean, and H-measure. The average rank of these metrics is used to compare the techniques. Our results reveal the improvement of the dynamic selection over the static ensemble, considering different degrees of class-imbalance. In particular, the combination of Balanced Random Forest with KNORA-Union [16] improves all the metrics we have evaluated.

The organization of this paper is as follows. Section 2 reviews the literature about credit scoring, imbalance learning approaches, and dynamic selection techniques. Section 3 presents the methodology used. Section 4.2 comments the results obtained from the experiments. Finally, the last section is dedicated to the conclusion and future work.

2 Background and related work

This study involves four main elements: credit scoring, imbalanced learning, pool generators, and dynamic selection classification. We briefly review each of these subjects in the following paragraphs.

Credit scoring. Several works have been published in last years focusing on default loan prediction. However, none of the previous papers combined dynamic selection, preprocessing techniques and different pool generators. Table 1 shows some aspects of recent papers about loan default prediction. Besides the year of publication, and the sampling approaches used, the table also presents the kind of ensemble used, homogeneous or heterogeneous, the selection approach, dynamic (DS) or static (SS), and the pool generation techniques evaluated. We can see that no paper combined dynamic selection, pool generation, and sampling techniques.

Table 1. Approaches tracking credit scoring in literature

Ref.	Year	Sampling	Ensemble		
			Kind	Selection	Pool generators
[9]	2019	-	Homog.	-	(a)
[7]	2018	-	Heterog.	DS	Bagging and different parameters
[14]	2018	Based on RUS	Heterog.	SS	Based on bagging
[22]	2018	SMOTE	Homog	SS	Bagging
[24]	2018	-	Heterog.	-	Based on bagging
[1]	2017	-	Homog.	-	(b)
[25]	2017	-	Homog.	-	Based on boosting
[3]	2016	-	Heterog.	DS	Feature selection based
[26]	2016	-	Heterog.	DS	Bagging based on clustering
[2]	2016	-	Heterog.	DS	Bagging
[17]	2015	-	Both	SS, DS	(c)

(a) Bagging, Boosting, Random subspace, Random Forest, Rotation Forest, DECORATE
(b) Bagging, Boosting, Random Subspace, DECORATE, Rotation Forest
(c) Bagging, Boosting, Random Forest, Rotation Forest

Sampling techniques for imbalanced learning. As mentioned in Section 1, the prediction task in credit scoring datasets suffers from the lack of sufficient samples of the minority class. Haixiang et al. [11] defined four categories of techniques for dealing with class imbalance. The first one modifies the data distribution, called *preprocessing solutions*. The next category applies different costs to misclassification of positive and negative samples, the *cost-sensitive solutions*. These classes are called "basic strategies" for addressing imbalanced learning. The third one adapts a classifier to deal with the class imbalance, the *algorithm level solutions*. The last one, *ensemble-based solutions*, combines the previous solutions using an ensemble. These last two categories are called "classification algorithms" for imbalanced learning. We describe the preprocessing and the ensemble-based approaches briefly in the following paragraphs.

Preprocessing is applied before the learning phase. Resampling techniques are used to rebalance the sample space for an imbalanced dataset to reduce the skewed class

distribution in the learning process. There are three possible methods to do it. The first one is over-sampling, which consists of creating new minority class samples. We test the widely used method, Synthetic Minority Over-sampling Technique (SMOTE), and Ranked minority oversampling (RAMO) . The second one is under-sampling, that consists of removing samples from the majority class. We test the most used method, Random Under-sampling (RUS). The hybrid method combines the two previous ones.

Pool generators. This paper focuses on six modified ensembles to handle imbalanced classification. A typical ensemble, also known as multiple classifier systems (MCS), has the following phases: the pool generation, the selection, and the integration. The MCS phases are presented in Figure 1. The main challenge of the pool generation phase is to generate a pool of accurate and diverse classifiers. The diversification can be achieved by homogeneous or heterogeneous base classifiers. Regarding the homogeneous pools, the diversity comes from different subsets of training data (Bagging, Boosting, or Hybrid), or using different features subspaces (Random Subspace Selection), or based on feature extraction (Rotation Forest). We test each of these homogeneous approaches combined with preprocessing techniques that we call "imbalanced ensembles".

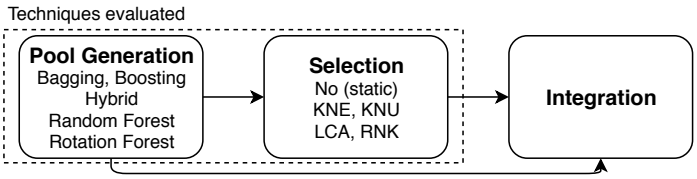

Fig. 1. The three MCS phases and the techniques evaluated in this work.

There is a significant number of pool generators and an even higher number of preprocessing approaches. We select the most used ones. The first selected imbalanced ensemble is Balanced Bagging (BBAG), it is the Bagging pool generator with an additional step to balance the training set at the fit time with Random Undersampling (RUS). The Balanced Random Forest (BRDF) [5] is the Random Forest ensemble with the application of RUS in each bootstrap sample. The same approach was made to Balanced Rotation Forest (BRTF), the balanced version of Rotation Forest [19]. The next two imbalanced ensembles, RUSBoost and SMOTEBoost, are the AdaBoost ensemble modified to give less or more samples in each boosting step using RUS or SMOTE, respectively. The last one, Easy Ensemble (EASY) is a bag of balanced AdaBoost ensembles. That is the reason for the hybrid terminology, with the behavior of bagging and boosting. The EASY ensemble uses RUS to balance the data. In all imbalance ensembles that use RUS, each base classifier receives a subset of the dataset with the same number of samples in each class. In SMOTEBoost, we double the number of samples of the minority class in each boost iteration.

Overview of dynamic selection. The second phase of an MCS is the selection. The main concepts are related to the type of selection and the notion of classifier competence (ability to predict correctly). The type of selection may be static, where the decision

about the competence of the base learners is made at the fitting time, or dynamic, when the decision is made at prediction time. The intuition behind the preference for dynamic over static selection is to select the most locally-accurate classifiers for each unknown sample. A dynamic selection approach defines competence measures, mostly related to the classifier accuracy in some part of the feature space, and a procedure to select the best estimators.

The dynamic selection approaches are classified by the selection methodology. According to this classification, there are two kinds of strategies: dynamic classifier selection (DCS) and dynamic ensemble selection (DES). The difference between them is the number of classifiers selected to predict each sample. The DCS selects the most competent base classifier, and the DES selects an ensemble of competent classifiers.

We test DS strategies based on different notions of competence measure as previous papers that evaluated dynamic selection in the context of imbalanced learning [27, 20]. For example, Local Class Accuracy (LCA) considers the local class accuracy separately. The Modified Classifier Rank (RNK) ranks the classifiers. These two techniques are DCS. They always select only the most competent classifier to the predict task. We also test two versions of K-Nearest Oracles (KNORA), that are DES techniques. Next, we briefly describe the four DS strategies adopted in this paper.

- The Local Class Accuracy (LCA) gets the prediction of the test sample of each base classifier and, according to the predicted class, compute the class accuracy regarding only the predicted class. The classifier with the higher class accuracy is used to predict the test sample.
- The Modified Classifier Rank (RNK) method ranks the accuracy of the base classifiers in the neighborhood for each test instance. The classifier with the highest accuracy is used to predict the test instance.
- The K-Nearest Oracles (KNORA) techniques are inspired by the Oracle [16] concept. Among them, the most promising are KNORA-Eliminate (KNE) and KNORA-Union (KNU). The KNE selects only the base classifiers with the perfect accuracy in the neighborhood of the test instance. On the other hand, in the KNU technique, the level of competence of a base classifier is measured by the number of correctly classified instances in the defined region of competence. In this case, every classifier that correctly classified at least one instance can vote.

The dynamic selection approaches require a dynamic selection dataset (DSEL) to determine the competence regions of the base classifiers. This data is used to measure the competence of the base classifiers on each part of the feature space. The main challenge in the DSEL generation is to use a reasonable part of the training data to allow a good performance of the DS approach and keep the other part to train the ensemble. The separation between the training data is important to avoid overfitting. In an imbalanced dataset, this task is even more difficult due to the lack of samples in the minority class. We present in Section 3 the approach of Roy et al. [20] to work around this problem.

The integration is the last step of an MCS, and it consists of applying the selected classifiers to recognize a given testing pattern. In cases where all classifiers are used (without selection) or when a subset is selected, a fusion strategy is necessary. This paper uses the fusion strategy of the ensembles and the dynamic selection strategies.

3 Methodology

In this section, we provide the methodology used in our experiments. We intend to assess in which scenarios dynamic selection combined with a preprocessing method improves the performance of an ensemble for imbalance credit scoring learning. To achieve it, we evaluate the techniques described in Table 2. The upper part **(I)** of Table 2 contains the preprocessing techniques that we use to generate the dynamic selection datasets. The middle part **(II)** of this table contains the imbalanced ensembles strategies, and the bottom part **(III)** lists the dynamic selection techniques evaluated.

Table 2. Techniques evaluated.

Label	Type	Acronym	Method
(I)	Imbalance Pre-processing	SMOTE	Synthetic Minority Over-sampling Technique
		RUS	Random under-sampling
		RAMO	Ranked minority oversampling
(II)	Imbalanced Ensembles (Pool generator + sampling)	BBAG	Balanced Bagging (Bagging + RUS)
		BRDF	Balanced Random Forest (Random Forest + RUS)
		BRTF	Balanced Rotation Forest (Rotation Forest + RUS)
		RUSB	RUS Boost (AdaBoost + RUS)
		SMTB	SMOTE Boost (AdaBoost + SMOTE)
		EASY	Easy ensemble (AdaBoost + RUS)
(III)	Dynamic Selection	KNE	k-Nearest Oracles-Eliminate
		KNU	k-Nearest Oracles-Union
		LCA	Local Class Accuracy
		RNK	Modified Classifier Rank

We use a grid-search to find the best hyper-parameters for each ensemble. The criteria used to choose the best model was the maximum F-measure. We test three pool sizes for all ensembles: [60, 100, 200]. For Balanced Random Forest (BRDF), we test three values for the maximum number of features, [$\sqrt{\#features/2}$, $\sqrt{\#features}$, $\sqrt{2\times\#features}$]. Form Balanced Rotation Forest (BRTF), we test two possibilities for the size of the feature group. [3, 9]. These are the most common values adopted on the credit scoring papers of Table 1.

The preprocessing techniques, RAMO and SMOTE, also have parameters. For RAMO, we use $k_1 = 10$, $k_2 = 5$ and $\alpha = 0.3$. For SMOTE, we use the number of nearest neighbors equal 5. Finally, for all the dynamic selection methods, we use seven nearest neighbors to define the region of competence. These parameters were adopted from [20].

We use three-fold cross-validation to measure the performance of each combination of techniques on 80% of each dataset. We then evaluate the best model achieved from each grid-search procedure on the remaining 20%, i.e., the test set. To train each combination of imbalance ensemble, preprocessing and dynamic selection techniques, we apply the three steps: i) generate the Dynamic selection dataset (DSEL) with the preprocessing technique, ii) train the imbalanced ensemble, and iii) compute the competence of each base classifier on each point of the feature space. This procedure is executed on the three-fold cross-validation for each grid-search hyper-parameter combination.

The left side of Figure 2 presents the schema of this approach. As mentioned earlier, we call here "imbalanced ensemble" an ensemble that uses a preprocessing approach to balance the data in the pool generation step. We also included in the center of Figure 2 the scheme of the baseline method. It consists of applying the grid-search using the three-fold cross-validation directly to the imbalanced ensemble without dynamic selection.

After obtaining the classification results, we compute different average ranks of the performance measures to assess each technique evaluated. The next subsections describe the details of the experiment and the results, as in previous works [17, 20].

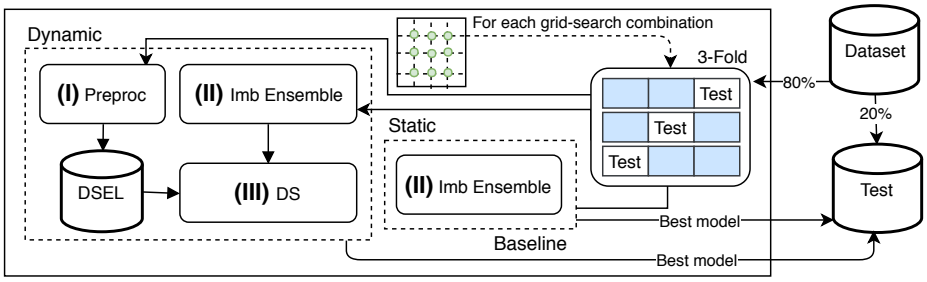

Fig. 2. The approach evaluated and the corresponding baseline (adapted from [20]).

4 Experiments

4.1 Experimental setup

Credit datasets. We perform the comparison by exploiting four real-world credit scoring datasets. Two data sets: German and Default are provided by UCI machine learning repository[3]. PPDai comes from a Chinese internet finance enterprise named PaiPaiDai[4]. The last one, Iranian, comes from [21]. The details of the datasets are shown in Table 3. We used the Imbalance Ratio (IR) measure, the cardinality of the majority class divided by the cardinality of the minority class, to separate these datasets into two groups. The first one, composed by German and Default, is the class of low imbalanced, with IR lower than 4. The second class is composed by Iranian and PPDai, and is the class of high imbalanced datasets, with IR over 4. We perform one experiment with the two groups of datasets separately. All other experiments use the full set of datasets.

Evaluation measures. A correct selection of evaluation measures is critical to avoid biased results. For instance, the percentage of correctly classified measure is widely used in classification but is not appropriate to an imbalanced dataset, since a naive classifier always predicting the majority class achieves a high score.

[3] https://archive.ics.uci.edu

[4] https://www.ppdai.com

Table 3. Datasets description

Dataset	# Samples	# Features	Imbalance Ratio (IR)
German	1,000	24	2.33
Default	30,000	24	3.52
PPdai	55,596	29	6.74
Iranian	1,000	27	19.0

We consider four metrics to measure the predictive accuracy of the classifiers: Area under the ROC curve (AUC), H-measure, F-measure, and G-mean. As in other imbalanced classification works, we consider the minority class, the bad credit, as the positive class in order to avoid bias results in F-measure. In the next paragraphs, we present the measures adopted.

Based on the elements of the confusion matrix, true positive (TP), false negative (FN), true negative (TN), and false positive (FP), we can define the precision, $Precision = \frac{TP}{TP+FP}$, the recall, or sensitivity or true positive rate (TPR), $Recall = \frac{TP}{TP+FN}$, the specificity or true negative rate (TNR), $Specificity = \frac{TN}{TN+FP}$, and the false positive rate (FPR), $FPR = 1 - TNR = \frac{FP}{TN+FP}$.

We now describe the performance metrics used in this paper. AUC is an extensively used evaluation measure obtained from the area under the ROC curve. The x-axis of the ROC curve represents the FPR, and the y-axis represents TPR (sensitivity). The F-measure is the weighted harmonic mean between precision and recall, as shown in Eq. (1). The β in F-measure formula is an hyper-parameter for weighting differently the precision and recall. The G-mean is the geometric mean of sensitivity and specificity and is also shown in Eq. (1).

$$\text{F-measure} = \frac{(1+\beta^2) \times Precision \times Recall}{\beta^2 \times Precision + Recall} \quad , \quad \text{G-mean} = \sqrt{Sensitivity \times Specificity} \tag{1}$$

H-measure is a threshold-varying evaluation metric proposed by Hand et al. [12]. It overcomes the AUC deficiency of use of different misclassification costs distributions for different classifiers. H-measure gives a normalized classifier assessment based on expected minimum misclassification loss, ranging from zero to one for a random and perfect classifier, respectively.

As pointed out by Garcia et al. [10], AUC and G-mean minimize the negative influence of skewed distributions, but they do not show up the contribution of each class to the overall performance. This means that different combinations of true positive rate and true negative rate produce the same result for these metrics. As in the credit scoring problem the cost of a false negative is much higher than a false positive, the F-measure is a more appropriate metric than the other two.

Data pre-processing. Data pre-processing is a crucial step to prepare data for training any prediction model. We used one-hot encoding to transform each categorical feature with N values in N binary features, for all values that have at least 1% of the cardinality of the dataset. For instance, consider a dataset with 1000 samples and one categorical

feature in this dataset with four possible values. Consider that the number of samples with each value is 500, 300, 195 and 5. We only consider three binary classes, for the three most numerous values. We also filled the missing values with the mean/mode for numeric/nominal features. Numeric features were standardized by removing the mean and scale the data to unit variance. We also removed the outliers limiting the maximum and minimum value up to $3 \times \sigma$ of each feature.

4.2 Experimental results

We now present the results by answering each research question. First, we analyze which technique combination produces the best results. Then, we compare the dynamic ensemble approaches with the static ones. Finally, we study the influence of the imbalance ratio in the performance of the combinations.

As in previous works [17, 20], we use the average rank of the four performance measures, namely, AUC, H-measure, F-measure, and G-mean in all experiments. For the F-measure, we adopted $\beta = 1$, that means to give the same weight for precision and recall in the Equation 1. Henceforth, we refer to F-measure as F1-score, the name used to refer to it when $\beta = 1$.

Best preprocessing technique to generate the DSEL. To assess **RQ1) "Which preprocessing technique generates the best dataset to calibrate the dynamic selection approach?"**, we applied the three preprocessing techniques to every combination between the four dynamic selection techniques and the six pool generators. Then, we compute the average and the variance of their 24 performance ranks (the lower the better). The results in Table 4 show that the Ranked Minority Oversampling (RAMO) got the lowest average rank. However, the difference for the second better, RUS, is small, only 0.005, and the variance is high. This ranking analysis shows that the use of different preprocessing approaches to generate the DSEL does not produce a considerable influence in the combination performance.

Table 4. Performance comparison of the diverse preprocessing techniques applied to DSEL

Preprocessing technique	Average rank	Variance
Ranked minority oversampling (RAMO)	1.898	0.1993
Random undersampling (RUS)	1.903	0.3930
Synthetic Minority Over-sampling Technique (SMOTE)	2.010	0.1025

Differently from the premise assumed by Roy et al. [20] that oversampling always produce better results than undersampling, the DSEL generated by undersampling and oversampling got similar results in our experiments. Roy et al. did not test undersampling techniques because of diversity issues of this technique. They argue that since undersampling maintains the minority class intact, the ensemble may not exhibit enough diversity. However, RUS got similar results of the oversampling techniques evaluated.

This result is significant because the undersampling approaches are usually more efficient than oversampling, once that the first generates a smaller dataset to be processed by the classifiers. Also, the reduced difference between the average ranks let us

conclude that the use of the different preprocessing approaches to generate the dynamic selection dataset does not impact strongly on the results of the dynamic selection.

Static selection vs. Dynamic selection. To answer **RQ2) "Is the combination of dynamic selection and preprocessing techniques better than the single application of a preprocessing approach?"**, we compare the combinations of pool generators, preprocessing approaches, and dynamic selection techniques with the direct application of the imbalanced ensemble. We evaluated the average rank of the static approach against the dynamic ones. Table 5 presents the average ranks of each combination evaluated. The lowest average rank of each row is highlighted in light gray. The best combination of each pool generator, hence considering all fifteen results, is highlighted in dark gray. As we can see, the dynamic selection approaches outperform the static approach in almost all pool generation techniques tested.

Table 5. Average rank of the static and dynamic selection approaches. The column "Pool" shows the imbalanced pool generators. The column "Preproc" the preprocessing techniques, while columns ST, KNE, KNU, LCA, and RNK report the average ranks of static ensemble, KNORA-Eliminate, KNORA-Union, Local Classifier Accuracy, and Modified Rank, respectively.

Pool	Preproc	ST	KNE	KNU	LCA	RNK	Pool	Preproc	ST	KNE	KNU	LCA	RNK
	RUS	**1.25**	2.75	1.75	4.75	4.25		RUS	2.44	2.75	**2.06**	4.81	2.94
BBAG	SMTE	**1.94**	2.62	2.12	3.38	4.94	EASY	SMTE	2.94	2.56	**2.44**	3.12	3.94
	RAMO	1.75	2.81	**1.5**	3.94	5.0		RAMO	3.38	**2.25**	**2.56**	3.06	3.5
	RUS	2.75	2.44	**1.69**	4.62	3.5		RUS	4.19	3.06	**1.44**	2.5	3.81
BRDF	SMTE	2.81	3.0	**1.56**	3.56	3.81	RUSB	SMTE	4.88	2.88	**1.38**	2.38	3.5
	RAMO	2.62	2.56	**1.19**	4.12	4.5		RAMO	4.38	3.5	**1.44**	2.19	3.5
	RUS	2.25	2.75	**2.12**	3.81	4.0		RUS	**1.69**	3.19	2.31	4.12	3.69
BRTF	SMTE	2.19	2.88	**1.94**	3.25	4.75	SMTB	SMTE	**1.44**	4.12	1.94	4.0	3.5
	RAMO	2.0	2.75	**1.69**	3.88	4.69		RAMO	**1.19**	4.12	2.56	3.69	3.44

By analyzing Table 5, we can see that KNORA-Union (KNU) achieved the lowest rank in almost all combinations tested. It means that the KNU is an excellent technique to combine with pool generators. In Table 5 (right), the SMOTEBoost (SMTB), i.e., ADABoost with SMOTE preprocessing, is the only ensemble where the static approach overcomes the dynamic one in all combinations. As the result of the static ensemble is only one, all the three cells of SMTB are highlighted in dark gray in Table 5. In addition to SMTB, the KNU using RUS and SMOTE preprocessing techniques to generate the dynamic selection dataset (DSEL) also lost against the static version of Balanced Bagging (BBAG), Table 5 (left).

The results presented in Table 5 reinforce our previous conclusion of **RQ1)** about the application of different preprocessing techniques to generate the dynamic selection dataset. In the six different pool generators tested, the modification of the way to produce the DSEL changed the winner only in two cases, Balanced Bagging (BBAG), and SMOTEBoost (SMTB). For this reason, we consider only the results of RAMO in the next results. We select RAMO because it is present in four best combinations (BBAG, BRDF, BRTF, and EASY) of the six pool generators, the dark gray cells in Table 5.

The ranks presented above are the combination of four performance measures: AUC, F1-score, H-measure, and G-mean. However, as we mentioned in Section 4.1, the F1-score is better than AUC and G-mean to measure the performance of imbalanced datasets when the minority class has a higher misclassification cost than the majority one. The advantage of F1-score in this scenario derives from the different treatment of the misclassification of the minority class. With this in mind, we compared the average ranks considering each of these three performance measures separately. The intuition is to check if we can find a particular combination that outperforms the others regarding F1-score and did not get the same result in other performance measures. Table 6 presents the average ranks considering only the AUC, F1-score, and G-mean measures. Again, the lowest rank of each row as marked in gray.

Table 6. Average rank of static and dynamic selection approaches according to G-mean, F1, and AUC. The column "Pool" shows the imbalanced pool generators, while the columns ST, KNE, KNU, LCA, and RNK the average ranks of static ensemble, KNORA-Eliminate, KNORA-Union, Local Classifier Accuracy, and Modified Rank, respectively.

Pool	**AUC**					**F1-score**					**G-mean**				
	ST	**KNE**	**KNU**	**LCA**	**RNK**	**ST**	**KNE**	**KNU**	**LCA**	**Rnk**	**ST**	**KNE**	**KNU**	**LCA**	**RNK**
BBAG	1.75	3.0	**1.25**	4.0	5.0	**1.75**	2.5	**1.75**	4.0	5.0	1.75	3.25	**1.25**	3.75	5.0
BRDF	2.5	2.5	**1.25**	4.25	4.5	3.0	2.5	**1.0**	4.0	4.5	2.0	2.75	**1.5**	4.25	4.5
BRTF	**1.75**	2.75	**1.75**	4.0	4.75	2.5	2.5	**1.75**	3.75	4.5	**1.5**	3.25	1.75	3.75	4.75
EASY	3.5	**2.25**	2.5	3.25	3.25	3.25	**2.25**	2.5	3.0	3.75	3.25	**2.25**	2.75	3.25	3.25
RUSB	4.5	3.5	**1.5**	2.0	3.5	4.5	3.5	**1.25**	2.25	3.5	4.5	3.0	**1.25**	3.25	3.0
SMTB	**1.0**	4.25	2.5	3.75	3.5	**1.0**	4.0	2.75	3.75	3.5	**1.75**	4.25	2.0	4.25	2.75

The Balanced Bagging (BBAG) presents a tie in the average rank considering only F1-score. It means that the static and the KNORA-Union (KNU) combinations of BBAG have similar performance regarding the four credit scoring data sets. This result helps to clarify the not uniform result of BBAG in Table 5 (left). The previous paper that combined dynamic selection and preprocessing techniques to imbalanced datasets [20] also found divergent results concerning AUC, F1, and G-mean for the bagging ensemble. However, their experiments with 84 multiple domain datasets shown that the F1-score of dynamic approaches combined with bagging outperforms the static bagging. One possible explanation for this divergence is given by Britto et al. in [4], where they concluded that the performance of dynamic selection approaches against the static ones is related to the complexity of the datasets [15]. We believe that more work is needed to better understand this issue. Except for SMOTEBoost (SMTB) and BBAG, all the other pool generators we evaluated presented a lower F1-score rank, demonstrating the superiority of dynamic selection for imbalanced credit scoring datasets.

We also evaluate the performance of the pool generators. Table 7 shows the winners. In this comparison, Balanced Bagging (BBAG) get the lowest average rank among the static ensembles, i.e., 1.69. Balanced Rotation Forest (BRTF), the second lowest rank among the static ensembles, get a rank of 3.19, with a difference of 1.5 from BBAG.

Table 7. Average rank of the pool generators strategies. The columns KNE, KNU, LCA, and RNK show the average ranks of KNORA-Eliminate, KNORA-Union, Local Classifier Accuracy, and Modified Rank, respectively.

Imbalanced Pool Generator	**Static**	**Dynamic Selection**			
	Ensemble	**KNE**	**KNU**	**LCA**	**RNK**
Balanced Bagging (BBAG)	**1.69**	1.81	2.19	2.69	3.06
Balanced Random Forest (BRDF)	3.44	3.12	**1.25**	4.19	3.31
Balalnced Rotation Forest (BRTF)	3.19	3.75	2.75	4.25	5.12
EASY Ensemble (EASY)	3.5	**1.69**	4.0	**1.19**	**1.19**
RUSBoost (RUSB)	5.56	5.38	5.0	4.25	3.94
SMOTEBoost (SMTB)	3.62	5.0	5.56	4.44	4.12

Among the dynamic selection approaches, we achieve two clear winners. Balanced Random Forest (BRDF) got the lowest average rank when combined with KNORA-Union (KNU), and Easy Ensemble (EASY) got the lowest average rank when combined with KNORA-Eliminate (KNE), and with the dynamic classifier selection (DCS) techniques, Local Classifier Accuracy (LCA) and Modified Classifier Rank (RNK). The excellent performance of EASY with DCS can be explained by the fact that the base classifier of EASY is an AdaBoost ensemble, and an ensemble usually outperforms a base classifier. The same reason can be used to explain the lowest rank of EASY when combined with KNE. As KNE considers only the base classifiers with perfect classification in the neighborhood of the sample predicted, the size of the ensemble selected dynamically tends to be small. Again, as in the case of LCA and RNK, the fact that EASY uses an AdaBoost ensemble as base classifier gives it an advantage in a small ensemble regarding the other ensembles that use base classifiers.

We can infer from these results that the combination of dynamic selection and preprocessing techniques improves the prediction performance of ensembles. Several pool generators tested improved their performance with dynamic selection techniques.

Influence of the imbalance level on the prediction performance. The third research question, **RQ3) "Does the imbalance level influence the prediction performance of the combinations?"**, aims at discovering whether the imbalance level has some influence on the prediction performance of the approaches evaluated. To answer this question, we divided the four datasets into two groups. The first one contains only the low-imbalanced datasets, with imbalance ratio (IR) under 4: German and Default. The second group contains the high imbalanced datasets, datasets with IR above 4: PPDai, and Iranian. This division is shown in Table 3. For each group of datasets, we compared the performance of the static imbalanced ensembles and the imbalanced ensembles combined with dynamic approaches.

Table 8 shows small variations in the results regarding the imbalance rate. Balanced Random Forest (BRDF), and SMOTEBoost (SMTB) pool generators do not change the ranking between low IR and high IR. The static SMTB and the BRDF combined with KNU get the lowest average rank on both imbalanced levels. Balanced Bagging (BBAG) and Balanced Rotation Forest (BRTF) also keep the same winner, but the static ensemble and KNU combination tie on the low imbalanced datasets.

Table 8. Average rank of static and dynamic approaches across different imbalance levels.

Imbalance level	**Low (IR $<$ 4)**					**High (IR $>$ 4)**				
Imb. Pool Gen.	**Static**	**KNE**	**KNU**	**LCA**	**RNK**	**Static**	**KNE**	**KNU**	**LCA**	**Rank**
BBAG	**1.5**	3.0	**1.5**	4.0	5.0	2.0	2.62	**1.5**	3.88	5.0
BRDF	2.88	2.5	**1.12**	4.5	4.0	2.38	2.62	**1.25**	3.75	5.0
BRTF	**1.5**	3.12	**1.5**	4.38	4.5	2.5	2.38	**1.88**	3.38	4.88
EASY	**2.38**	2.5	2.5	3.12	4.0	4.38	**2.0**	2.62	3.0	3.0
RUSB	4.75	3.12	**1.12**	2.88	3.12	4.0	3.88	1.75	**1.5**	3.88
SMTB	**1.0**	4.88	2.38	2.75	4.0	**1.38**	3.38	2.75	4.62	2.88

On the other hand, Easy Ensemble (EASY) and RUSBoost (RUSB) changed the selected winner. The static EASY got the lowest average rank with the less imbalanced datasets, but not in the most imbalanced. The same behavior happened with RUSBoost. However, the lowest rank changed between dynamic selection strategies.

In this subsection, we found that the performance of the combination between some imbalanced ensembles and selection approaches does not depend on the imbalance level of the datasets. The static version of SMOTEBoost (SMTB) performs better than the other combinations. On the other hand, the Balanced Random Forest (BRDF) and Balanced Rotation Forest (BRTF) perform better when combined with KNU.

Discussion. We now investigate the best combination strategy among all evaluated. To achieve it, we compute a new average rank of the best results of each combination between pool generator and selection techniques presented in Table 5, the dark gray cells. Figure 3 presents the average ranks of these best combinations. As we can see, the three lowest ranks approaches, balanced random forest (BRDF), Balanced Bagging (BBAG), and Balanced Rotation Forest (BRTF) use KNORA-Union (KNU) and Ranked Minority Oversampling (RAMO). The only static ensemble in this rank is the SMOTEBoost. This result demonstrates that the combination of dynamic selection using oversampling techniques to generate the DSEL with balanced ensembles (ensembles that use random undersampling to balance the data in each step) improves the prediction performance of imbalanced credit scoring datasets.

With these experiments, we empirically verify that the random undersampling is a possible option to generate the dynamic selection training dataset (DSEL). We also observed that the application of KNORA-Union (KNU) improves the prediction of Balanced Random Forest (BRDF), and Balanced Rotation Forest (BRTF) even if we consider only F1-score, an appropriate metric for low and moderate imbalanced credit scoring data sets. After, we observed that BRDF is the best pool generator to combine with KNU and EASY is the best option to combine with KNORA-Eliminate (KNE), Local Class Accuracy (LCA) and Modified Rank (RNK). Finally, we observe that the BRDF combined with KNU using Ranked Minority Oversampling (RAMO) to generate the dynamic selection dataset was the best combination tested.

5 Conclusions and future work

We presented a comprehensive study of the credit scoring task. We assessed the combination of Dynamic Selection (DS) methods, data preprocessing and pool generation

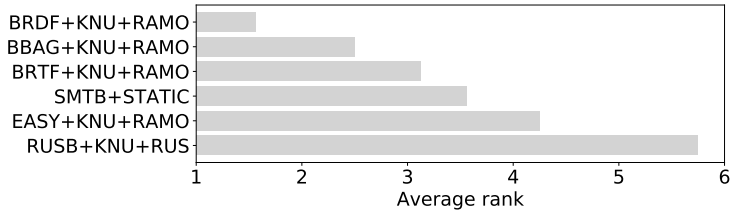

Fig. 3. The average rank of the best combinations.

ensembles to deal with the imbalanced nature of the credit scoring data sets. In the literature, data preprocessing has been adopted in imbalanced datasets for over 15 years. Moreover, several different pool generation ensembles have been used for decades. More recently, dynamic selection methods have shown excellent performance for some classification scenarios. However, to the best of our knowledge, an investigation of the combination of preprocessing approaches, dynamic selection techniques, and pool generators ensembles is missing. We combined these three techniques to observe the effects in credit scoring problem. Experiments conducted on four datasets shown that combining preprocessing with DS enhances the prediction performance according to 4 measures. We empirically concluded that the KNORA-Union (KNU) is the best DS technique to use in these combinations. We also noticed that Ranked Minority Oversampling got better results than SMOTE and RUS in generating the dynamic selection dataset (DESL), but the difference is not significant. Additionally, we empirically concluded that dynamic selection techniques improve the prediction performance of low and high imbalanced credit scoring datasets. Finally, we found that the balanced version of random forest (BRDF) overcome the other pool generation approaches.

Interesting future work regards the evaluation of different combinations of preprocessing and DS. There is also a possibility to apply the preprocessing technique before and use this modified dataset to train the ensemble and the DS approach. We also consider as a future work a more in-depth study of the performance of dynamic selection applied to credit scoring datasets regarding AUC, F1-score, and G-mean.

References

1. Abellán, J., Castellano, J.G.: A comparative study on base classifiers in ensemble methods for credit scoring. Expert Systems with Applications **73**, 1–10 (2017)
2. Ala'raj, M., Abbod, M.F.: Classifiers consensus system approach for credit scoring. Knowledge-Based Systems **104**, 89–105 (2016)
3. Ala'raj, M., Abbod, M.F.: A new hybrid ensemble credit scoring model based on classifiers consensus system approach. Expert Systems with Applications **64**, 36–55 (2016)
4. Britto Jr, A.S., Sabourin, R., Oliveira, L.E.: Dynamic selection of classifiersa comprehensive review. Pattern Recognition **47**(11), 3665–3680 (2014)
5. Chen, C., Liaw, A., Breiman, L.: Using random forest to learn imbalanced data. University of California, Berkeley **110**, 1–12 (2004)
6. Dietterich, T.G.: Ensemble methods in machine learning. In: International workshop on multiple classifier systems. pp. 1–15. Springer (2000)

7. Feng, X., Xiao, Z., Zhong, B., Qiu, J., Dong, Y.: Dynamic ensemble classification for credit scoring using soft probability. Applied Soft Computing **65**(C), 139–151 (2018)
8. Galar, M., Fernandez, A., Barrenechea, E., Bustince, H., Herrera, F.: A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **42**(4), 463–484 (2012)
9. García, V., Marqués, A.I., Sánchez, J.S.: Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction. Information Fusion **47**, 88–101 (2019)
10. Garcia, V., Mollineda, R.A., Sanchez, J.S.: Theoretical analysis of a performance measure for imbalanced data. In: Pattern Recognition (ICPR), 2010 20th International Conference on. pp. 617–620. IEEE (2010)
11. 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**, 220–239 (2017)
12. Hand, D.J.: Measuring classifier performance: a coherent alternative to the area under the roc curve. Machine learning **77**(1), 103–123 (2009)
13. Hand, D.J., Henley, W.E.: Statistical classification methods in consumer credit scoring: a review. Journal of the Royal Statistical Society: Series A (Statistics in Society) **160**(3) (1997)
14. He, H., Zhang, W., Zhang, S.: A novel ensemble method for credit scoring: Adaption of different imbalance ratios. Expert Systems with Applications **98**, 105–117 (2018)
15. Ho, T.K., Basu, M.: Complexity measures of supervised classification problems. IEEE Transactions on Pattern Analysis & Machine Intelligence (3), 289–300 (2002)
16. Kuncheva, L.I.: A theoretical study on six classifier fusion strategies. IEEE Transactions on Pattern Analysis & Machine Intelligence (2), 281–286 (2002)
17. 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), 124–136 (2015)
18. Opitz, D., Maclin, R.: Popular ensemble methods: An empirical study. Journal of artificial intelligence research **11**, 169–198 (1999)
19. Rodriguez, J.J., Kuncheva, L.I., Alonso, C.J.: Rotation forest: A new classifier ensemble method. IEEE transactions on pattern analysis and machine intelligence **28**(10) (2006)
20. Roy, A., Cruz, R.M., Sabourin, R., Cavalcanti, G.D.: A study on combining dynamic selection and data preprocessing for imbalance learning. Neurocomputing **286**, 179–192 (2018)
21. Sabzevari, H., Soleymani, M., Noorbakhsh, E.: A comparison between statistical and data mining methods for credit scoring in case of limited available data. In: Proceedings of the 3rd CRC Credit Scoring Conference, Edinburgh, UK. Citeseer (2007)
22. Sun, J., Lang, J., Fujita, H., Li, H.: Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates. Information Sciences **425**, 76–91 (2018)
23. Thomas, L., Crook, J., Edelman, D.: Credit scoring and its applications. Siam (2017)
24. Xia, Y., Liu, C., Da, B., Xie, F.: A novel heterogeneous ensemble credit scoring model based on bstacking approach. Expert Systems with Applications **93**, 182–199 (2018)
25. 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)
26. Xiao, H., Xiao, Z., Wang, Y.: Ensemble classification based on supervised clustering for credit scoring. Applied Soft Computing **43**, 73–86 (2016)
27. Xiao, J., Xie, L., He, C., Jiang, X.: Dynamic classifier ensemble model for customer classification with imbalanced class distribution. Expert Systems with Applications **39**(3) (2012)
28. Zhou, L., Wang, H.: Loan default prediction on large imbalanced data using random forests. Indonesian Journal of Electrical Engineering and Computer Science **10**(6) (2012)