RMkNN and KNORA-IU: Combining Imbalanced Dynamic Selection Techniques for Credit Scoring

Leopoldo Melo Junior, Jose Fernandes Macêdo
Computer Science Department - Universidade Federal do Ceará
Universidade Federal do Ceará
Fortaleza, Brazil
{leopoldosmj, jose.macedo}@insightlab.ufc.br

Franco Maria Nardini, Chiara Renso
ISTI-CNR
Pisa, Italy
{francomaria.nardini, chiara.renso}@isti.cnr.it

***Abstract*—Credit scoring has become a critical tool to discriminate "bad" applicants from "good" ones for financial institutions. One common characteristic of the credit datasets is the imbalance between good and bad applicants, with low defaults (no paid loans). Ensemble classification methodology is widely used in this field. However, dynamic ensemble selection approaches to imbalanced datasets have drawn little consideration. This study aims to measure the performance of the combination of two recent dynamic selection techniques for imbalanced credit scoring datasets, Reduced Minority k-NN (RMkNN) and KNORA-Imbalanced Union (KNORA-IU). We conduct a comprehensive evaluation of the proposed combination against state-of-the-art competitors on six real-world public datasets and one private one. Experiments show that this combination improves the classification performance on the evaluated datasets in terms of AUC, balanced accuracy, H-measure, G-mean, F-measure, and Recall.**

***Index Terms*—dynamic selection classification, imbalanced, credit scoring**

I. Introduction

Credit offer is a crucial activity for banks that aim at improving their profitability and competitiveness. Minor improvements in the default prediction imply significant profits to financial institutions [1]. However, the decision to grant a loan to a customer is complex and risky because it requires an accurate default prediction to protect banks from financial losses, especially during financial crises. Thomas et al. [2] 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 of credit over time. Besides these aspects, data availability, accuracy, and reliability make the default prediction much harder than other domain-specific classification problems. Therefore, new methods and techniques, called credit scoring models, are required to cope with these problems while guaranteeing a low percentage of defaults.

Available historical loan data creates an excellent opportunity to take advantage of trending machine learning methods for detecting defaulters, people that do not pay back the loan. However, real credit scoring datasets are usually high imbalanced. They are called Low Default Portfolios (LDP) since they are highly skewed and with a low default rate.

Recent credit scoring papers [3]–[7] evaluate improvements in defaulter's prediction by using ensembles, a classification approach that combines the predictions of a set of base classifiers instead of only one. These papers usually use a set of available credit scoring data to evaluate their approaches. We observe that most of the datasets used in these papers are low imbalanced, when the IR, the ratio between the number of samples of the classes, is under 3. However, in the real world, credit scoring datasets are moderate or high imbalanced, $IR \geq 3$; and skewed data is a challenge for machine learning methods since classifiers tend to predict only the majority class. This paper aims at improving the use of Dynamic Selection Classification (DSC) in imbalanced credit scoring problems. To this end, we investigate the combination of two recent imbalanced approaches to attenuate the impact of skewed data on DSC techniques.

The first technique is called Reduced Minority k-NN (RMkNN) recently introduced in [8]. This technique changes the local region (the neighborhood of a query sample) definition to reduce the imbalanced level in overlapping regions, keeping non-overlapping areas unchanged. The second technique is called KNORA-Imbalanced Union (KNIU) and it is an extension of KNORA-U [9] and it uses a performance measure called FA^2 that combines *F-measure* and *accuracy* to compute the local competence of base classifiers. RMkNN and KNIU use different strategies to attenuate the imbalanced problem of dynamic ensemble selection techniques and therefore we believe that the combination of these two promising and complementary techniques could bring improvements in the dynamic classification of imbalanced credit scoring problems.

The remainder of this paper is presented as follows. Section II outlines a brief review of the literature concerning credit scoring, imbalanced learning, and dynamic selection classification. Section III shows the main contribution of this paper that is the combination of these two techniques. Section IV shows the experimental setup adopted while Section V present the study results and Section VI presents conclusions of this study.

II. Background and related works

In this section, we report the background knowledge involved with the credit scoring field and the tools combined in this paper.

A. Credit Scoring

As defined by Thomas et al. [2], credit scoring is a set of decision models that aid lenders in granting consumer credit. Financial institutions use these techniques to decide who will get credit, how much they should get, what price they should get it at, and what operational strategies will enhance the profitability of the borrowers to the lenders.

A credit scoring dataset contains two types of information — the first one corresponds to the regular customer information, such as age or level of scholarship. The second type corresponds to the credit behavior of this customer in previous loans. Sometimes, only the first type is available to evaluate the customer.

The vital point in a credit scoring system is that there is a large sample of previous customers with their application details and subsequent credit history available [2]. All the credit scoring techniques use samples to identify the connection between the characteristics of the consumers and how "good" or "bad" theirs subsequent story is, where bad usually means defaulting, not paid ones, in a given period, and good means not defaulting. Next, we discuss imbalanced learning approaches.

B. Imbalanced learning approaches

As mentioned in the Introduction, the prediction task in credit scoring datasets suffers from the lack of sufficient samples of the minority class, the defaulters. Haixiang et al. [10] defined four categories of techniques for handling class imbalance: (1) modify the data distribution, *preprocessing solutions*; (2) apply different costs to misclassification of positive and negative samples, the *cost-sensitive solutions*. (1) and (2) are "basic strategies" for imbalanced learning. (3) and (4) are "classification algorithms": (3) adapts a classifier to deal with the class imbalance, the *algorithm level solutions*; and (4) *ensemble-based solutions*, combines the previous solutions using an ensemble. We describe the two most common imbalanced approaches briefly in the following paragraphs, preprocessing and ensemble-based.

Preprocessing comes before the learning phase. Resampling, the most common preprocessing technique, balances 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 over-sampling, Undersampling, and hybrid. The first one is over-sampling, which consists of creating new minority class samples synthetically. The widely used method is SMOTE [11]. The second one is under-sampling, which consists of removing samples from the majority class. The most used method is RUS [12]. Finally, The hybrid methods combine oversampling and undersampling methods.

The other common imbalanced approach is ensemble-based methods. Ensemble methods are learning algorithms that construct a set of classifiers and then classify new data points by taking a vote, weighted or not, of their predictions [13]. Ensemble approaches to imbalanced learning consist of combining preprocessing, cost-sensitive, and classifier algorithm modifications. They combine the power of an ensemble with the ability of other imbalanced techniques to overcome the imbalance issue.

Besides these methods, there are also "cost-sensitive solutions" and "classification algorithms". Cost-sensitive solutions consist of assuming higher costs for the misclassification of minority class samples. On the other hand, the classification algorithms consist of changing the kernel or the activation function to improve the classification performance for imbalanced data.

On the other hand, the classification algorithms approach consists of changing the kernel or the activation function to improve the classification performance for imbalanced data.

C. Ensembles

A typical ensemble has the following phases: pool generation, selection, and integration. The following subsections present the preliminary concepts of each phase.

1) Pool generators: The main challenge of the pool generation phase is to generate a pool of accurate and diverse classifiers [14]. Homogeneous or heterogeneous base classifiers can achieve this diversification. 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).

2) Selection: The second phase of an ensemble is the base classifiers' selection to the prediction procedure.The main concepts of this phase are related to the type of selection and the notion of classifier competence (ability to predict a new sample correctly). The type of selection may be static [13], where the decision about the competence of the base learners occurs at the fitting time, or dynamic [15] when the decision occurs at prediction time.

3) Integration: The integration is the last step of an ensemble, and it consists of merging the predictions of the selected classifiers to compute the prediction of the query sample.

D. Dynamic Selection Classification

A dynamic selection Classification (DSC) allows, given a test instance, the choice of one or more base learners from a pool instead of using all the classifiers [9], [16]. The intuition behind the preference for dynamic over static selection is to select the most locally accurate classifiers to predict each unknown sample.

A dynamic selection approach uses a set of known samples in the neighborhood of the query sample (usually defined with kNN algorithm), a competence measure, and a procedure to select the best local estimators. These known samples are called Dynamic Selection Dataset (DSEL), and are used to compute the local competence of the base classifiers in the local region of the query sample. To avoid over-fitting of the selection procedure, the DSEL must be different from the training data. Finally, according to the selection strategy, only the most competent base classifiers are used to predict the unknown sample [16].

The dynamic selection approaches are classified by the selection methodology [9]. According to this classification, there are two kinds of strategies: DCS and DES. The difference between them is the number of classifiers selected to predict each sample. DCS selects only the most competent base classifier, and DES selects a set of competent local classifiers.

Roy et al. [17] is one of the most recent works we found that evaluated dynamic selection techniques to solve imbalance classification problems. As previous papers [18] that evaluated DSC in the context of imbalanced learning, they test DSC strategies based on different notions of competence measure. For example, LCA considers the local class accuracy separately. The RNK ranks the classifiers. These two techniques are DCS. They also test two versions of KNORA, which are DES techniques. Next, we briefly describe the four DSC strategies obtained from [19] and adopted in this paper.

- The Local Class Accuracy (LCA) [16], [20] 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 LCA chooses the classifier with the higher class accuracy to predict the test sample.
- The Modified Classifier Rank (RNK) [16], [21] method ranks the accuracy of the base classifiers in the neighborhood of each test instance. The classifier with the highest accuracy is used to predict the test instance.
- The K-Nearest Oracles (KNORA) [9] techniques are inspired by the Oracle [22] concept. The most promising are KNE and 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 local region. In this case, every classifier that correctly classified at least one instance can vote for the final prediction.

A challenge of DSC is to reserve the DSEL without compromise the training data. In an imbalanced dataset, this challenge is even bigger once there are few samples of the minority class to learn the class pattern. Roy et al. [17] solved it using oversampling approaches to generate the DSEL. It guarantees the difference between the DSEL and the training data by including synthetically created samples. Next, we briefly describe the techniques we combine to improve the credit scoring classification.

E. Combined techniques

1) RMkNN - Reduced Minority kNN: Melo Jr et al. [8] proposed a novel kNN algorithm that redefines the local region for the dynamic selection classification of imbalanced credit scoring datasets. To redefine the local region, the authors develop a new kNN that uses the label of the neighbors and the imbalance ratio of the dataset to choose the list of neighbors of a query sample.

This modified kNN computes the nearest neighbors considering a reduced distance of the minority class samples. The intuition of this technique is to increase the number of minority class samples on the local region definitions. Thus, we increase the chance of the base classifiers' local competence also considers minority class samples. Eq 1 indicates the function used to reduce the distance. In this function, D_m indicates the distance of the minority class sample and IR indicates the imbalanced ratio of the dataset.

$$f(D_m, IR) = \frac{D_m}{(1 + \frac{log(IR)}{10})} \tag{1}$$

2) KNORA - Imbalanced Union (KNIU): Most DSC techniques use accuracy to define the local competency and to determine the contribution weight of each base classifier in the final prediction. As mentioned before, accuracy in an imbalanced dataset reports good results even for a naive learner that predicts only the majority class.

One possible approach to overcome this weakness is the use of oversampling approaches. However, these techniques usually include noise to data. It is not desirable, mainly because the DSEL is used to select which base classifiers are used to predict the query sample.

To overcome this issue, KNIU [23] combines f-measure with *accuracy*. We base this decision on the fact that the accuracy is enough to assess the classifiers in a neighborhood with only one class sample. However, even in this scenario, we discover empirically that the accuracy measure does not penalize the base learners appropriately with few prediction errors. We observe that the influence in the final prediction of classifiers with few errors is still strong. To attenuate this "good performance", we decided to consider the square of accuracy. Thus, our proposed performance measure to compute the competence of each base learner can be written as Eq. (2).

In this context, [23] proposed a performance measure to compute the local competence of base classifiers in a dynamic selection technique. This performance measure uses a specific approach in overlapping and non-overlapping areas. In overlapping areas, the measure is F1-score, and in non-overlapping areas is accuracy. Eq 2

$$FA^2(yt, yp) = \begin{cases} f1\text{-}score(yt, yp), & \text{if } yt \supset \{-1, 1\} \\ (accuracy(yt, yp))^2, & \text{otherwise} \end{cases} \tag{2}$$

III. How do RMkNN and KNIU work together?

RMkNN and KNIU use different strategies to attenuate the imbalanced problem of dynamic ensemble selection techniques. In the following paragraphs, we describe how to combine the approaches adopted by each one.

As indicated in the previous section, RMkNN changes the local region definition in the procedure to include more minority samples in the overlapping areas. This modification does not produce any influence on the performance measure used to measure the local competence of the base classifiers. On

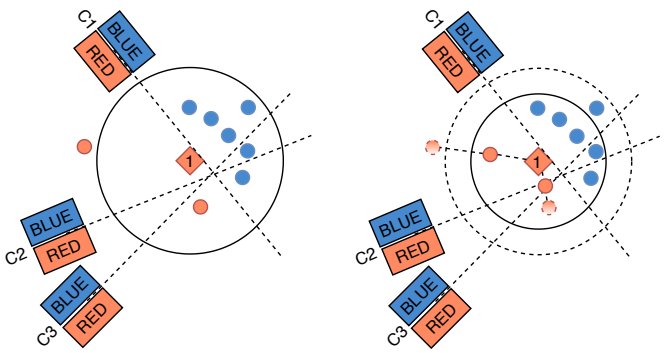

Fig. 1: Example of the combination of RMkNN and KNIU. The left side of the figure shows the local region definition without RMkNN. The right side shows the local region definition with RMkNN, reducing the distance between the query sample and the minority class samples.

the other hand, KNIU uses a different performance measure to compute the local competence of the base classifiers. Again, a modification produced by RMkNN does not create any issue with this new performance measure.

To illustrate how this combination works, we define a small ensemble example with three linear base classifiers in a bi-dimensional data space. We illustrate a local region definition and the prediction fusion of these two techniques alone and their combination.

Figure 1 shows the neighborhood of a query sample represented by the red diamond 1. The left side of Figure 1 shows the local region definition using regular kNN. The seven nearest neighbors of the query sample have six samples of the majority class, blue circles, and one sample of the minority class, red circle. We draw a circle to facilitate the visualization of the seven nearest neighbors. The right side of Figure 1 shows the local region definition using the RMkNN. The new seven nearest neighbors of the query sample have five samples of the majority class and two samples of the minority class. In this part of this figure, we highlight the distance reduction between the query sample and the minority class samples. This distance reduction of RMkNN replaced one sample of the majority class and included a sample of the minority class in the nearest neighbors list. We also highlight the new set of nearest neighbors with a smaller circle.

Figure 1 also shows three linear binary classifiers $C1$, $C2$, e $C3$. We use these base classifiers to compute and compare the KNIU, KNU+RMkNN, and KNIU+RMkNN predictions. To compute this experiment, we consider that the positive class, the red circles, is represented by 1 and the negative class, the blue circles, is represented by -1.

Table I shows the computation of the predictions of the query sample 1 using the three compared approaches. The columns "Learner" and "Pred" show, respectively, the base classifiers and their predictions for the query sample 1. The next three columns contain the computation of the fusion procedure of the three dynamic selection classification approaches. The column "KNIU" contains the FA^2 measure for each base classifier and the corresponding weight of the base classifier in the final prediction. Next, the column "KNU+RMkNN" contains the accuracy of each base classifier in the local region of the query sample and the contribution of each base classifier. Finally, the column KNIU+RMkNN contains the FA^2 measure for each base classifier. The three columns called "weight" indicate each base classifier's contribution in the final prediction of the dynamic selection approaches.

TABLE I: Classification example results of KNIU, KNU+RMkNN, and KNIU+RMkNN.

S[a]	Learner	Pred	KNIU		KNU+RMkNN		KNIU+RMkNN	
			FA^2	weight	Acc[b]	weight	FA^2	weight
1	C1	1	1	1	1	1	1	1
	C2	-1	0.67	-0.67	0.71	-0.71	0.50	-0.50
	C3	-1	0.40	-0.40	0.57	-0.57	0.40	-0.40
	DSC prediction		-0.02 (-1)		-0.10 (-1)		0.03 (1)	

[a] **S** means the query sample evaluated.
[b] **Acc** means the accuracy of the learner in the neighborhood of the query sample.

Analyzing Table I, we see that both KNIU and KNU+RMkNN can not predict correctly sample 1. However, KNIU+RMkNN combined predicts correctly the sample 1. The reasons why the combination of these techniques can predict correctly are: (i) FA^2 increases the weight of the most competent learners in the predictions' fusion step; and (ii) RMkNN reduces the imbalance of the local area around the sample 1. Next, we repeat the experiments performed in [8] including the KNIU+RMkNN approach to measuring its performance in real credit scoring datasets.

Now, we investigate the combination of these two techniques. We believe that these two modifications can cooperate and improve even more the prediction performance of dynamic selection classification techniques in imbalanced credit scoring datasets.

IV. Experimental Setup

In this section, we provide a complete description of our experiments. Next, we present the datasets used, the experimental setting, the evaluation measures, and the statistical test.

A. Real credit data

We perform our experiments by exploiting seven real-world credit scoring datasets. *German* and *Default* are provided by UCI machine learning repository[1]. *PPDai* comes from a Chinese internet finance enterprise named PaiPaiDai[2]. *Iran* comes from [24]. The private one cames from a Brazilian financial instituion. *GiveMe*[3] comes from Kaagle competition. The last one, *LC2015*, contains loan data of 2015 from Lending Club[4]. The details of the datasets are shown in Table II.

[1] https://archive.ics.uci.edu
[2] https://www.ppdai.com
[3] https://www.kaggle.com/c/GiveMeSomeCredit
[4] https://www.lendingclub.com

TABLE II: Datasets description

Dataset	#Samples	#Features	Imbalance Ratio (IR)
German	1,000	20	2.33
Default	30,000	24	3.52
PPDai	55,596	29	6.74
Private	4,976	56	9.05
GiveMe	150,000	10	13.96
Iran	1000	27	19.77
LC2015	95,633	72	77.45

TABLE III: Techniques evaluated.

Label	Type	Acronym	Method
(I)	Reduced Minority kNN	RMkNN	Modified kNN that reduce the distance of the minority class samples
(II)	Imbalance Preprocessing	SMTE	Synthetic Minority Over-sampling Technique
		RUS	Random under-sampling
(III)	Imbalanced Ensembles (Pool generator + sampling)	BBAG	Balanced Bagging (Bagging + RUS)
		BGSM	Bagging SMOTE (Bagging + SMOTE)
		BRND	Balanced Random Forest (Random Forest + RUS)
		RFSM	Random Forest SMOTE (Random Forest + SMOTE)
		BROT	Balanced Rotation Forest (Rotation Forest + RUS)
		RUSB	RUS Boost (AdaBoost + RUS)
		SMTB	SMOTE Boost (AdaBoost + SMOTE)
		EASY	Easy ensemble (Bagging of AdaBoost + RUS)
(IV)	Dynamic Selection	KNE	k-Nearest Oracles-Eliminate
		KNU	k-Nearest Oracles-Union
		KNIU	k-Nearest Oracles-Imbalanced Union
		LCA	Local Class Accuracy
		RNK	Modified Classifier Rank
(V)	Credit Scoring Benchmarks	LOGR	Logistic Regression
		XGB	eXtreme Gradient Boosting
		ANN	Airtificial Neural Networks
		LSVM	Linear Support Vector Machine
		SVM	Support Vector Machine
		RNDF	Random Forest

B. Experimental setting

To compare the effectiveness of RMkNN combined with KNIU, we repeat the experiment performed in [8], including KNIU, as a dynamic selection technique. We compare its results against six credit scoring benchmarks. They are logistic regression, eXtreme Gradient Boosting, Artificial Neural Networks, linear and non-linear support vector machine, and a static random forest ensemble.

Table III shows the list of evaluated combinations: **(I)** contains our proposal of modification of kNN to select balanced samples of the DSEL; **(II)** contains preprocessing techniques (SMOTE, and RUS) to balance the DSEL; **(III)** contains the imbalanced ensembles strategies; **(IV)** lists the dynamic selection techniques evaluated, including KNIU; and **(V)** lists the credit scoring benchmarks evaluated.

Figure 2 shows the experimental framework of this work. We perform 5-fold cross-validation to get each method's mean and standard deviation to evaluate each classification approach. For each training fold of the 5-fold, we perform 3-fold grid search cross-validation to find the best hyper-parameters of each static classifier (steps III and V of boxes C and D of Figure 2). For boxes C, and D, we use the best static ensemble to predict the test part of the 5-fold cross-validation. For boxes A and B of Figure 2, we use the 3-fold training data as DSEL, box A, or to generate the DSEL using a preprocessing approach, box B. Then, we use the DSELs, and the dynamic selection approaches on the imbalanced ensembles to find the best dynamic selection model, boxes A and B of Figure 2.

C. Evaluation measures and statistical test

We evaluate six metrics to measure the predictive accuracy of the classifiers: Area under the ROC curve (AUC), H-measure, balanced accuracy (BAcc), G-mean, F-measure, and True Positive Rate (TPR). As in other work about imbalanced classification, we consider the minority class, namely the bad credit, as the positive class to avoid bias results in F-measure.

As recommended by [25] and followed by other credit scoring papers [6], [26], we employed nonparametric tests instead of parametric ones because the assumptions of parametric tests tend to be violated when comparing classification models. We employ the Friedman test [27], which is a rank-based nonparametric test, to compare different models. Eq. 3 formalizes the statistic of the Friedman test.

$$X_F^2 = \frac{12D}{K(K+1)}\left[\sum_{j=1}^{K} AR_j^2 - \frac{K(K+1)^2}{4}\right], \quad where\ AR_j = \frac{1}{D}\sum_{i=1}^{D} r_i^j. \quad (3)$$

In Eq (3), D denotes the number of datasets used in the study, K is the total number of classifiers and r_i^j is the rank of classifier *j* on dataset i. X_F^2 is distributed according to the Chi-square $(\tilde{\chi}^2)$ distribution with $K-1$ degrees of freedom. If the value of X_F^2 is large enough, then the null hypothesis that there is no difference between the techniques can be rejected. The Friedman statistic is well suited for this data analysis as it is less susceptible to outliers.

The post hoc Nemenyi test [28] is applied to report any significant differences between individual classifiers. The Nemenyi post hoc test states that the performances of two or more classifiers are significantly different if their average ranks differ by at least the critical difference (CD), given by

$$CD = q_{\alpha,\infty,K}\sqrt{\frac{K(K+1)}{12D}}. \quad (4)$$

In this formula, the value $q_{\alpha,\infty,K}$ is based on the Studentized range statistic [28]. Finally, the results from Friedman's statistic and the Nemenyi post hoc tests are displayed using a modified version of significance diagrams [25], [29]. These diagrams display the ranked performances of the classification techniques and the critical difference to clearly show any techniques that are significantly different from the best-performing classifiers. Next, we discuss the results achieved in these tests.

V. Results and analysis

We perform two experiments. First, we compute the overall average ranking of 134 classification approaches. After, we compare the best estimator of the previous test with the credit scoring benchmarks. The following subsections describe them.

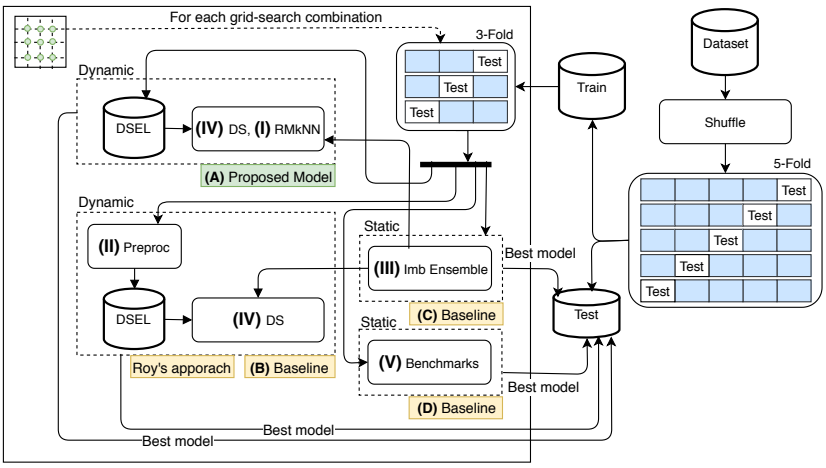

Fig. 2: The proposed approach and the baselines (adapted from [17]).

A. Overall average ranking

In this experiment, we perform the comparison of the combinations of pool generators, preprocessing approaches, and dynamic selection techniques of Table III. We evaluate the average rank of all 134 combinations (8 imbalanced ensembles $\times$ 5 selection approaches $\times$ 3 strategies to handle the DSEL + 8 static imbalanced ensembles + 6 credit scoring benchmarks) to evaluate the best approaches to imbalanced credit scoring.

As in [8], we start with the average rank of all 134 classification combinations. We compute the average rank of seven performance measures evaluated, AUC, H-measure, balanced accuracy, geometric mean, F1-score, F5-score, and recall (TPR). After, we compute the average of these averages to find a unique global rank. Table IV shows the first 15 best combinations of this global rank. In this table, the gray calls indicate the lowest average rank of each performance measure. In green, we also highlight the combinations that use RMkNN and KNIU, in blue the combinations that use only RMkNN, and in yellow the combinations that use only KNIU.

As we can see, nine of the fifteen best combinations use at least one of the two proposed techniques, KNIU or RMkNN. Additionally, three of them use both techniques. Evaluating the pool generators of Table IV, we observe only three ensembles, BRND, BROT, and EASY. The best combination of the three pool generators is KMkNN with KNIU.

Analyzing the four first lines of Table IV, we see RNDF and ROTF combined with RMkNN, KNIU, and KNU. Comparing KNIU and KNU, we observe that KNIU combinations achieve better rankings in measures that emphasize positive class misclassification, such as F5 and TPR. It means that RMkNN and KNIU reduce the number of default loans. On the other hand, KNU+RMkNN combinations achieve the lowest average rankings in the remaining performance measures, AUC, H-measure, BAcc, G-mean, and F1. It means that KNU+RMkNN grants more good loans than KNIU+RMkNN. Despite that, the global ranking of the KNIU combination still achieves a lower rank, indicating that the default loan reduction of KNIU+RMkNN is more significant than the good loan improvement of KNU+RMkNN.

B. Comparison of the best average ranking with the credit scoring benchmarks

After this preliminary evaluation, we compare the actual prediction results of Balanced Random Forest (BRND), the lowest rank of Table IV with the best credit scoring benchmarks observed in [8], XGboost (XGB), Logistic Regression (LOGR), Random Forest (RNDF). We aim to identify the differences between these approaches.

For each dataset evaluated, Table V shows the average and the standard deviation of 5-fold execution explained in Figure 2. Here, we highlight the best result of each dataset and each performance measure in bold and dark gray. The second-best result is also highlighted in light gray. For each approach and each dataset, Table V shows seven performance measures, AUC, H-measure, BAcc, G-mean, F1-score, F5-score, and True Positive Rate (TPR).

We begin the analysis comparing the BRND+KNIU+RMkNN and BRND+KNU+RMkNN. We observe that BRND+KNIU+RMkNN achieves the best result 24 times in the 49 possible (seven dataset and seven performance measures). On the other hand, BRND+KNU+RMkNN achieves the best score only once. These results lead us to conclude that KNIU improves the performance of BRND combined with RMkNN regarding the use of regular KNU.

Next, we observe the superiority of BRND+KNIU+RMkNN in the performance measures that give more importance to the positive class misclassification, F5-score, and Recall (TPR). Regarding these two measures, BRND+KNIU+RMkNN achieves the best result or the second-best result in 78.5% (11 times in 14 results). It means that the proposed combination grants fewer default loans.

Additionally, the superiority of BRND+KNIU+RMkNN also occurs among the measures that gives the same weight to misclassification of both classes, AUC, H-measure, BAcc,

TABLE IV: Average ranking of all 134 techniques

Appr.	Selection	Performance Measures [average ranking (standard deviation)]							Avg
		AUC	H	BAcc	G-mean	F1	F5	TPR	
BRND	KNIU+RMkNN	21.7 (17.4)	22.5 (19.5)	14.5 (17.2)	15.4 (19.1)	26.1 (22.1)	**19.9 (18.7)**	25.5 (17.3)	21.2
BRND	KNU+RMkNN	17.1 (17.1)	20.4 (17.8)	**14.1 (16.7)**	15.5 (18.6)	23.9 (20.4)	23.9 (20.3)	29.1 (20.2)	21.5
BROT	KNIU+RMkNN	22.2 (14.5)	23.1 (23.0)	16.9 (13.7)	15.7 (13.1)	25.3 (15.6)	21.2 (14.6)	26.0 (13.1)	21.9
BROT	KNU+RMkNN	**16.5 (11.4)**	**19.8 (22.4)**	16.0 (16.5)	**15.3 (15.6)**	**21.9 (16.5)**	25.9 (19.0)	31.4 (14.9)	22.1
BROT	KNU+SMTE	19.2 (10.5)	25.9 (17.9)	19.6 (12.2)	18.9 (13.1)	28.1 (14.8)	25.4 (15.6)	29.6 (15.2)	24.4
BRND	KNU+SMTE	19.0 (11.2)	27.7 (17.3)	18.4 (15.8)	18.8 (17.6)	31.1 (19.6)	24.7 (18.3)	28.8 (18.2)	24.6
BRND	STATIC	20.1 (22.9)	32.7 (24.0)	18.8 (19.1)	19.4 (19.7)	38.3 (29.8)	23.1 (18.3)	23.9 (18.9)	25.1
BRND	KNIU+SMTE	24.9 (13.4)	31.7 (20.0)	19.4 (16.2)	19.9 (18.0)	33.4 (22.4)	22.5 (19.0)	26.3 (18.7)	25.5
BROT	STATIC	19.3 (17.7)	31.5 (20.3)	20.3 (15.5)	20.0 (16.4)	37.5 (24.8)	24.7 (16.1)	25.4 (17.0)	25.6
BROT	KNU+RUS	20.8 (16.4)	30.9 (20.5)	19.9 (14.5)	19.3 (14.9)	35.9 (23.6)	25.5 (16.5)	27.8 (16.6)	25.9
BROT	KNIU+SMTE	27.0 (13.1)	32.6 (21.0)	23.4 (12.2)	22.1 (13.4)	34.0 (16.0)	21.9 (13.5)	25.3 (13.9)	26.4
BRND	KNU+RUS	20.3 (20.5)	33.3 (22.9)	20.9 (20.1)	21.3 (20.4)	38.6 (29.2)	25.4 (18.6)	27.9 (19.2)	26.8
BROT	KNIU+RUS	28.6 (18.0)	36.6 (23.0)	22.7 (15.3)	21.7 (15.7)	41.2 (26.1)	23.1 (15.5)	23.9 (16.3)	27.7
BRND	KNIU+RUS	31.7 (21.9)	37.2 (24.3)	22.4 (21.2)	22.1 (20.9)	42.5 (31.7)	22.2 (18.6)	**21.9 (18.6)**	27.8
EASY	KNIU+RMkNN	29.2 (24.3)	30.7 (26.4)	30.2 (34.0)	29.3 (33.8)	40.0 (27.9)	27.1 (25.4)	29.1 (24.5)	30.6

TABLE V: Balanced Random Forest combined with KNORA-U, KNIU and RMkNN compared with state-of-the-art classifiers in credit scoring problem

Dataset	Classif.	Selection	Performance Measures						
			AUC	H	BAcc	G-mean	F1	F5	TPR
German	XGB	STATIC	0.79 (0.02)	0.23 (0.04)	0.72 (0.02)	0.72 (0.02)	0.61 (0.02)	0.67 (0.04)	0.67 (0.04)
	LOGR	STATIC	0.80 (0.03)	0.26 (0.05)	0.74 (0.03)	0.74 (0.03)	0.63 (0.03)	0.72 (0.08)	0.73 (0.08)
	RNDF	STATIC	0.79 (0.03)	0.23 (0.04)	0.71 (0.02)	0.71 (0.02)	0.60 (0.03)	0.66 (0.07)	0.66 (0.07)
	BRND	STATIC	0.80 (0.03)	0.24 (0.07)	0.73 (0.03)	0.73 (0.03)	0.62 (0.04)	0.75 (0.04)	0.76 (0.04)
	BRND	KNU+RMk	**0.80 (0.03)**	0.26 (0.06)	0.74 (0.03)	0.74 (0.03)	0.63 (0.04)	0.76 (0.04)	0.77 (0.04)
	BRND	KNIU+RMk	0.80 (0.03)	**0.27 (0.07)**	**0.74 (0.03)**	**0.74 (0.03)**	**0.63 (0.04)**	**0.78 (0.03)**	**0.79 (0.03)**
Default	XGB	STATIC	0.78 (0.02)	0.23 (0.04)	0.71 (0.02)	**0.71 (0.02)**	0.54 (0.03)	0.62 (0.03)	0.62 (0.03)
	LOGR	STATIC	0.72 (0.02)	0.14 (0.03)	0.67 (0.02)	0.67 (0.02)	0.48 (0.02)	0.62 (0.03)	0.64 (0.03)
	RNDF	STATIC	**0.78 (0.02)**	**0.24 (0.04)**	**0.71 (0.02)**	0.70 (0.02)	**0.55 (0.03)**	0.59 (0.03)	0.60 (0.03)
	BRND	STATIC	0.78 (0.02)	0.21 (0.04)	0.71 (0.02)	0.71 (0.02)	0.53 (0.02)	0.63 (0.03)	0.64 (0.03)
	BRND	KNU+RMk	0.78 (0.02)	0.22 (0.04)	0.71 (0.02)	0.71 (0.02)	0.53 (0.03)	0.63 (0.03)	0.64 (0.03)
	BRND	KNIU+RMk	0.78 (0.02)	0.21 (0.04)	0.71 (0.02)	0.71 (0.02)	0.53 (0.02)	**0.63 (0.03)**	**0.64 (0.03)**
PPDai	XGB	STATIC	**0.63 (0.05)**	**0.02 (0.02)**	**0.56 (0.04)**	**0.46 (0.26)**	**0.21 (0.12)**	0.36 (0.21)	0.38 (0.22)
	LOGR	STATIC	0.63 (0.03)	0.02 (0.04)	0.52 (0.04)	0.15 (0.20)	0.07 (0.13)	0.06 (0.12)	0.06 (0.12)
	RNDF	STATIC	0.63 (0.04)	0.02 (0.02)	0.56 (0.04)	0.44 (0.25)	0.20 (0.12)	0.41 (0.27)	0.45 (0.31)
	BRND	STATIC	0.61 (0.05)	0.02 (0.01)	0.55 (0.03)	0.43 (0.23)	0.20 (0.11)	**0.46 (0.30)**	**0.52 (0.35)**
	BRND	KNU+RMk	0.61 (0.04)	0.02 (0.01)	0.55 (0.03)	0.43 (0.23)	0.20 (0.11)	0.45 (0.30)	0.51 (0.35)
	BRND	KNIU+RMk	0.60 (0.05)	0.02 (0.01)	0.55 (0.03)	0.43 (0.23)	0.20 (0.11)	0.46 (0.30)	0.52 (0.35)
Private	XGB	STATIC	0.68 (0.04)	0.07 (0.05)	0.60 (0.06)	0.54 (0.13)	0.24 (0.07)	0.37 (0.17)	0.39 (0.19)
	LOGR	STATIC	0.67 (0.05)	0.06 (0.03)	0.62 (0.03)	0.62 (0.03)	0.24 (0.02)	0.55 (0.06)	0.61 (0.08)
	RNDF	STATIC	0.72 (0.02)	0.11 (0.06)	0.62 (0.05)	0.54 (0.12)	0.28 (0.07)	0.34 (0.14)	0.35 (0.15)
	BRND	STATIC	0.72 (0.03)	0.10 (0.02)	0.66 (0.02)	0.66 (0.02)	0.28 (0.01)	0.60 (0.05)	0.67 (0.06)
	BRND	KNU+RMk	0.72 (0.03)	0.11 (0.03)	0.67 (0.03)	0.67 (0.02)	0.28 (0.01)	0.62 (0.06)	**0.69 (0.07)**
	BRND	KNIU+RMk	**0.72 (0.03)**	**0.11 (0.03)**	**0.67 (0.02)**	**0.67 (0.02)**	**0.28 (0.01)**	**0.62 (0.06)**	**0.69 (0.07)**
GiveMe	XGB	STATIC	**0.86 (0.00)**	0.34 (0.01)	**0.79 (0.00)**	**0.79 (0.00)**	0.34 (0.00)	0.70 (0.01)	0.77 (0.01)
	LOGR	STATIC	0.81 (0.01)	0.25 (0.01)	0.73 (0.00)	0.73 (0.00)	0.31 (0.01)	0.59 (0.01)	0.64 (0.01)
	RNDF	STATIC	0.86 (0.00)	**0.35 (0.02)**	0.78 (0.00)	0.78 (0.01)	**0.36 (0.02)**	0.67 (0.02)	0.73 (0.03)
	BRND	STATIC	0.86 (0.00)	0.34 (0.01)	0.79 (0.00)	0.79 (0.00)	0.33 (0.00)	**0.71 (0.00)**	**0.78 (0.00)**
	BRND	KNU+RMk	0.86 (0.00)	0.34 (0.01)	0.79 (0.00)	0.79 (0.00)	0.34 (0.00)	0.70 (0.00)	0.77 (0.01)
	BRND	KNIU+RMk	0.86 (0.00)	0.34 (0.01)	0.78 (0.00)	0.78 (0.00)	0.34 (0.00)	0.70 (0.00)	0.77 (0.00)
Iran	XGB	STATIC	0.76 (0.06)	0.15 (0.06)	0.61 (0.03)	0.49 (0.06)	**0.27 (0.06)**	0.25 (0.06)	0.25 (0.06)
	LOGR	STATIC	0.78 (0.06)	0.00 (0.00)	0.50 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)	0.00 (0.00)
	RNDF	STATIC	0.79 (0.04)	0.11 (0.04)	0.57 (0.02)	0.37 (0.06)	0.23 (0.06)	0.15 (0.05)	0.14 (0.05)
	BRND	STATIC	0.77 (0.05)	0.18 (0.08)	0.71 (0.05)	0.71 (0.05)	0.19 (0.04)	0.58 (0.07)	**0.71 (0.08)**
	BRND	KNU+RMk	0.81 (0.07)	0.28 (0.12)	0.73 (0.07)	0.72 (0.08)	0.27 (0.06)	0.57 (0.12)	0.63 (0.15)
	BRND	KNIU+RMk	**0.82 (0.06)**	**0.29 (0.14)**	**0.74 (0.08)**	**0.73 (0.09)**	0.26 (0.07)	**0.59 (0.14)**	0.67 (0.16)
LC2015	XGB	STATIC	**0.71 (0.04)**	0.08 (0.02)	0.64 (0.02)	0.62 (0.04)	**0.05 (0.00)**	0.29 (0.03)	0.52 (0.08)
	LOGR	STATIC	0.69 (0.02)	0.03 (0.04)	0.56 (0.08)	0.25 (0.35)	0.02 (0.02)	0.12 (0.16)	0.23 (0.32)
	RNDF	STATIC	0.71 (0.03)	0.03 (0.05)	0.54 (0.06)	0.20 (0.29)	0.02 (0.03)	0.09 (0.13)	0.12 (0.19)
	BRND	STATIC	0.70 (0.03)	0.08 (0.01)	0.65 (0.01)	0.65 (0.01)	0.04 (0.00)	0.32 (0.01)	**0.68 (0.03)**
	BRND	KNU+RMk	0.70 (0.03)	0.09 (0.03)	0.66 (0.02)	0.66 (0.02)	0.05 (0.00)	0.32 (0.02)	0.63 (0.04)
	BRND	KNIU+RMk	0.69 (0.03)	**0.10 (0.03)**	**0.66 (0.03)**	**0.66 (0.03)**	0.05 (0.00)	**0.32 (0.03)**	0.62 (0.05)

G-mean, and F1-score. The proposed classification approach achieves the best result in 45.7% (16 of 35 results).

As in [8], 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 ensemble combination and the credit scoring benchmarks. Applying the Friedman test on the average ranking of these fourteen classifiers, we get a Friedman test statistic $= 90.51$, and a $p - value < 0.005$. As the Friedman test result is significant ($p < 0.005$), we can apply the post hoc Nemenyi test to the distribution.

Figure 3 shows the average ranks of these best combinations and the Critical Distance of the Nemenyi test. This figure shows that balanced random forest (BRND) combined with KNORA Imbalanced Union (KNIU) and using RMkNN to generate the DSEL is the best approach, the lowest average rank. This approach is statistically better than Artificial Neural Networks and Support Vector Machine, as indicated by the critical distance bar.

We also observe that RMkNN is present on four best combinations of eight ensembles. They are highlighted in green on Figure 3, and they are Balanced Random Forest (BRND), Balanced Rotation Forest (BROT), Easy Ensemble (EASY), and Balanced Bagging (BBAG). The following three best ranking positions are combinations that use Random Undersampling (RUS) to generate the dynamic selection dataset (DSEL). They are: SMOTEBoost (SMTB), RUSBoost (RUSB), and Random Forest SMOTE (RFSM). Only the last position, Bagging SMOTE (BGSM), uses SMOTE to generate the DSEL. Figure 3 highlights these last four combinations in yellow.

With these experiments, we observe that KNIU combined with RMkNN improves the use of RMkNN combined with KNU. We also observe that KNORA-Imbalanced Union (KNIU) is an excellent dynamic selection technique to combine with imbalanced ensembles. After, we observe that BRND is the best pool generator to combine with KNIU.

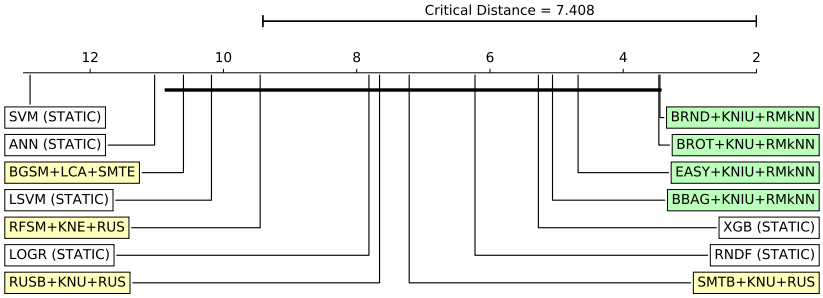

Fig. 3: The average rank of the best combinations including RMkNN and KNORA-IU.

VI. Conclusion

In this paper, we evaluated the combination of the two techniques presented in [8] and [23], Reduced Minority kNN, and KNORA-Imbalanced Union, respectively. First, we offer a hypothesis about the use of these two techniques together. After, we demonstrate by an example of how RMkNN and KNIU work together. Next, we compute the average ranking of the combinations of techniques of Table III.

We conclude that the combination of RMkNN and KNIU improves the prediction performance of three imbalanced ensembles regarding the use of RMkNN alone. We also observe that RMkNN and KNIU improve the performance regarding measures that give more weight to positive class misclassification, such as F5 and TPR [2].

References

[1] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: a review," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 160, no. 3, 1997.

[2] L. Thomas, J. Crook, and D. Edelman, *Credit scoring and its applications*, vol. 2. Siam, 2017.

[3] V. García, A. I. Marqués, and J. S. Sánchez, "Exploring the synergetic effects of sample types on the performance of ensembles for credit risk and corporate bankruptcy prediction," *Information Fusion*, vol. 47, pp. 88–101, 2019.

[4] H. He, W. Zhang, and S. Zhang, "A novel ensemble method for credit scoring: Adaption of different imbalance ratios," *Expert Systems with Applications*, vol. 98, pp. 105–117, 2018.

[5] J. Sun, J. Lang, H. Fujita, and H. Li, "Imbalanced enterprise credit evaluation with dte-sbd: Decision tree ensemble based on smote and bagging with differentiated sampling rates," *Information Sciences*, vol. 425, pp. 76–91, 2018.

[6] Y. Xia, C. Liu, B. Da, and F. Xie, "A novel heterogeneous ensemble credit scoring model based on bstacking approach," *Expert Systems with Applications*, vol. 93, pp. 182–199, 2018.

[7] J. AbellÁn and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Systems with Applications*, vol. 73, pp. 1–10, 2017.

[8] L. Melo Junior, F. M. Nardini, C. Renso, R. Trani, and J. A. Macedo, "A novel approach to define the local region of dynamic selection techniques in imbalanced credit scoring problems," *Expert Systems with Applications*, vol. 152, p. 113351, 2020.

[9] A. H. Ko, R. Sabourin, and A. S. Britto, Jr, "From dynamic classifier selection to dynamic ensemble selection," *Pattern Recognition*, vol. 41, no. 5, pp. 1718–1731, 2008.

[10] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Systems with Applications*, vol. 73, pp. 220–239, 2017.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[12] R. Barandela, R. M. Valdovinos, and J. S. Sánchez, "New applications of ensembles of classifiers," *Pattern Analysis & Applications*, vol. 6, no. 3, pp. 245–256, 2003.

[13] T. G. Dietterich, "Ensemble methods in machine learning," in *International workshop on multiple classifier systems*, pp. 1–15, Springer, 2000.

[14] L. K. Hansen and P. Salamon, "Neural network ensembles," *IEEE transactions on pattern analysis and machine intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[15] G. Giacinto and F. Roli, "Methods for dynamic classifier selection," in *Proceedings 10th International Conference on Image Analysis and Processing*, pp. 659–664, IEEE, 1999.

[16] A. S. Britto, Jr, R. Sabourin, and L. E. Oliveira, "Dynamic selection of classifiers—a comprehensive review," *Pattern Recognition*, vol. 47, no. 11, pp. 3665–3680, 2014.

[17] A. Roy, R. M. Cruz, R. Sabourin, and G. D. Cavalcanti, "A study on combining dynamic selection and data preprocessing for imbalance learning," *Neurocomputing*, vol. 286, pp. 179–192, 2018.

[18] J. Xiao, L. Xie, C. He, and X. Jiang, "Dynamic classifier ensemble model for customer classification with imbalanced class distribution," *Expert Systems with Applications*, vol. 39, no. 3, 2012.

[19] R. M. O. Cruz, L. G. Hafemann, R. Sabourin, and G. D. C. Cavalcanti, "DESlib: A Dynamic ensemble selection library in Python," *arXiv preprint arXiv:1802.04967*, 2018.

[20] K. Woods, W. P. Kegelmeyer, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE transactions on pattern analysis and machine intelligence*, vol. 19, no. 4, pp. 405–410, 1997.

[21] M. Sabourin, A. Mitiche, D. Thomas, and G. Nagy, "Classifier combination for hand-printed digit recognition," in *Document Analysis and Recognition, 1993., Proceedings of the Second International Conference on*, pp. 163–166, IEEE, 1993.

[22] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[23] L. Melo, Jr, F. M. Nardini, C. Renso, and J. A. Macedo, "Knora-iu: Improving the dynamic selection prediction in imbalanced credit scoring problems," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 424–431, IEEE, 2019.

[24] H. Sabzevari, M. Soleymani, and E. Noorbakhsh, "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*, pp. 1–5, Citeseer, 2007.

[25] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.

[26] S. Lessmann, B. Baesens, H.-V. Seow, and L. C. Thomas, "Benchmarking state-of-the-art classification algorithms for credit scoring: An update of research," *European Journal of Operational Research*, vol. 247, no. 1, pp. 124–136, 2015.

[27] M. Friedman, "A comparison of alternative tests of significance for the problem of m rankings," *The Annals of Mathematical Statistics*, vol. 11, no. 1, pp. 86–92, 1940.

[28] P. Nemenyi, "Distribution-free multiple comparisons," in *Biometrics*, vol. 18, p. 263, International Biometric Soc 1441 I ST, NW, Suite 700, Washington, DC 20005-2210, 1962.

[29] S. Lessmann, B. Baesens, C. Mues, and S. Pietsch, "Benchmarking classification models for software defect prediction: A proposed framework and novel findings," *IEEE Transactions on Software Engineering*, vol. 34, no. 4, pp. 485–496, 2008.