A Region-based Training Data Segmentation Strategy to Credit Scoring

Roberto Saia, Salvatore Carta, Gianni Fenu and Livio Pompianu
Department of Mathematics and Computer Science,
University of Cagliari, Via Ospedale 72 - 09124 Cagliari, Italy

Keywords: Business Intelligence, Decision Support System, Risk Assessment, Credit Scoring, Machine Learning.

Abstract: The rating of users requesting financial services is a growing task, especially in this historical period of the COVID-19 pandemic characterized by a dramatic increase in online activities, mainly related to e-commerce. This kind of assessment is a task manually performed in the past that today needs to be carried out by automatic credit scoring systems, due to the enormous number of requests to process. It follows that such systems play a crucial role for financial operators, as their effectiveness is directly related to gains and losses of money. Despite the huge investments in terms of financial and human resources devoted to the development of such systems, the state-of-the-art solutions are transversally affected by some well-known problems that make the development of credit scoring systems a challenging task, mainly related to the unbalance and heterogeneity of the involved data, problems to which it adds the scarcity of public datasets. The Region-based Training Data Segmentation (RTDS) strategy proposed in this work revolves around a divide-and-conquer approach, where the user classification depends on the results of several sub-classifications. In more detail, the training data is divided into regions that bound different users and features, which are used to train several classification models that will lead toward the final classification through a majority voting rule. Such a strategy relies on the consideration that the independent analysis of different users and features can lead to a more accurate classification than that offered by a single evaluation model trained on the entire dataset. The validation process carried out using three public real-world datasets with a different number of features, samples, and degree of data imbalance demonstrates the effectiveness of the proposed strategy, which outperforms the canonical training one in the context of all the datasets.

1 INTRODUCTION

The exponential increase in e-commerce activities that characterizes today's modern societies has undergone further growth as a result of the restrictions on movement imposed by the COVID-19 pandemic, which has prompted people to increase their online purchases and, more generally, to increase the use of the services offered by the Internet. This scenario has at the same time increased requests for consumer credit and therefore, generalizing, the need for financial operators to assess the solvency of their potential customers. Unlike in the past, where the low number of requests allowed human management, today this activity is carried out through automatic systems that, based on the outcome of past customers, evaluate the new users, performing an operation that in the literature is named as *Credit Scoring*. The credit scoring task is usually performed by classifying the new users according to a binary (classification as *reliable* or *unreliable*) or continuous (assignment of *credit ratings*) criterion. Such a classification relies on a large number of approaches, where the evaluation model is trained by using the users' information (which from now on we define *instances*), e.g., age, current job, total income, other loans in progress, etc.

It should be noted that the type of data involved in the credit scoring processes has drastically reduced the number of public datasets available to researchers, compared to other domains. This is due to a whole series of reasons mainly related to the privacy of financial operators and their customers, as also when such data are anonymized it is possible to extract privacy-sensitive information. In addition to this problem of data scarcity, the data available for the training of the evaluation models are commonly affected by a high degree of unbalance of the data classes, which are typically two: *reliable* and *unreliable* users. This configuration, to define evaluation models not influenced by the samples that belong to the majority class, must be appropriately managed through re-sampling techniques (Leevy et al., 2018), which add synthetic in-

Saia, R., Carta, S., Fenu, G. and Pompianu, L.
A Region-based Training Data Segmentation Strategy to Credit Scoring.
DOI: 10.5220/0011137400003283
In *Proceedings of the 19th International Conference on Security and Cryptography (SECRYPT 2022)*, pages 275-282
ISBN: 978-989-758-590-6; ISSN: 2184-7711

stances to the minority class or remove them from the majority one. However, such a process generates data with similar series of features that characterize *reliable* and *unreliable* users.

Initially formalized by us for a different domain, the proposed strategy revolves around the idea that we can reduce the heterogeneity problem by adopting a divide-and-conquer criterion, which relies on the consideration that the instances that compose the datasets used to train the evaluation models refer to different users and features. On the basis of this consideration, we can split the classification task into several subclassifications, each of them performed by an evaluation model trained by using a different region of the dataset, in terms of instances and features. In this regard, then we propose a *Region-based Training Data Segmentation* (RTDS) strategy, according to which the training dataset is divided into several regions by following an experimentally-defined number of rows and columns that, respectively, bound a region composed by certain instances (dataset rows) and features (dataset columns). The defined regions are then used to train independent evaluation models and the final classification of the new instances is reached according to an ensemble criterion regulated by a majority voting rule.

Unlike most credit scoring literature approaches, all the experiments related to this work were carried out by ensuring a real separation between the data used to select the best credit scoring algorithm in the context of a canonical training model process (to use as competitor algorithm and strategy) and to define the optimal number of regions for the proposed RTDS strategy, and the data used to validate it (performance comparison). In more detail, each dataset is divided into two parts (50 percent each one) named *in-sample* and *out-sample*, which are respectively used for the aforementioned activities.

The scientific contributions of our work are summarized as follows:

- formalization of the *Region-based Training Data Segmentation* (RTDS) strategy, where the training dataset is divided into several regions that bound a certain number of instances (rows) and features (columns), and the instance classification depends on a series of independent evaluation models, each of them trained on a different region, according to an ensemble approach regulated by a majority voting rule;
- formalization of criteria that allow the adoption of the RTDS strategy even when the data configuration does not permit the division of the training dataset into equal-size regions (*padding criterion*) or/and it does not permit the application of the majority voting rule during the ensemble classification (*classification criterion*) because the regions are even in number;
- definition of a classification algorithm that implements the RTDS strategy, classifying each new instance as *reliable* or *unreliable* on the based of a given training dataset;
- validation of the proposed RTDS strategy performed by comparing its performance (using the *out-of-sample* part of the dataset) to that of a canonical training approach based on the same best classification algorithm (previously detected using the *in-sample* part of the dataset).

2 BACKGROUND AND RELATED WORK

Premising that an instance classification as *unreliable* indicates a *default* status, i.e., the failure for the user to grant the legal obligations related to the requested financial service (e.g., a loan), the literature reports three different risk models associated to the *default* concept: *Probability of Default* (PD), when we need to evaluate the likelihood of a *default* over a certain period; *Exposure At Default* (EAD), when we need to evaluate the total value of exposition of a financial operator in case of *default*; *Loss Given Default* (LGD), when we need to evaluate the amount of monetary losses of a financial operator in case of *default*. In the above context, the work we proposed is aimed to perform binary classification of the instances into the *reliable* or *unreliable* classes, then it takes into account the PD model.

The literature shows how the credit scoring task is faced by a large number of approaches, from those focused on statistical algorithms to those that exploit transformed data domains, machine and deep learning algorithms, and a large number of hybrid approaches that combine different algorithms and strategies. Regarding statistical approaches, literature offers many works, such as the one that improve the *Logistic Regression* algorithm with non-linear decision-tree effects (Dumitrescu et al., 2022), or that where the *Linear Discriminant Analysis* has been used for the credit scoring task (Khemais et al., 2016). Regarding transformed data domains approaches, in a work (Saia and Carta, 2017b) the authors face the credit scoring task by exploiting the *Fourier Transform*, similarly in another work (Saia et al., 2018), which instead exploits the *Wavelet Transform*, or in (Carta et al., 2021; Carta et al., 2019), where the authors use a transformed feature space. Regarding machine learning approaches, the *Decision Tree* and *Support Vector Machine* al-

gorithms were combined in a work (Roy and Urolagin, 2019) in order to define a credit scoring system, whereas in the work (Liu et al., 2022) the authors designed a credit scoring system based on tree-enhanced gradient boosting decision trees. Regarding deep learning approaches, an *Artificial Neural Network* is exploited in a work (Liu et al., 2019) in order to perform the credit scoring task, whereas for the same goal an *Imbalanced Generative Adversarial Fusion Network* based both on a *feed-forward neural network* and on a *Bidirectional Long Short-Term Memory network* is proposed in another work (Lei et al., 2019). Regarding other approaches, an entropy criterion is exploited by the authors in several works (Saia and Carta, 2016a; Carta et al., 2020), whereas a linear-dependence criterion is used in (Saia and Carta, 2016c; Saia and Carta, 2016b), and the combination of different algorithms and strategies to perform the credit scoring task is faced in another work (Zhang et al., 2019). Another interesting work (Saia. et al., 2021) investigates the feasibility to define a credit scoring model based on the bank transactions instead of the canonical users' information.

Open Problems: Although over time state-of-the-art credit scoring solutions have gradually improved their performance, there are well-known problems that reduce the effectiveness of all the approaches, as they depend on the nature of the involved data. The main problems are the scarcity of public datasets to be used for the definition of new credit scoring approaches/strategies, and the complication that those available typically present a high degree of data imbalance because the examples of *unreliable* instances are fewer in number than those relating to *reliable* ones. In more detail, the scarcity of real-world datasets is mainly related to the privacy policies adopted by many public and private entities (Sloan and Warner, 2018) such as, for instance, the banks and other credit institutions. Concerning the data imbalance, it leads toward the underestimation of the unreliable cases during the training of the credit scoring evaluation models, since that process results biased by the majority class (i.e., *reliable* cases). In this case, the only solution is the adoption of balancing techniques that work by removing some majority class samples (*undersampling*), or by adding some synthetic instances to the minority class samples based on the existing ones (*oversampling*), and, in some cases, these two approaches can be combined. It should be observed how the adoption of *undersampling* techniques that remove samples from the majority class reduces the available information about this class, making the trained valuation model less accurate (Park and Park, 2021), whereas the adoption of *oversampling* techniques (the most used in the literature) could lead to an *overfitting* problem because the introduction of synthetic samples in a class might overestimate it in terms of probability (Weiss, 2004). In the light of the current literature (Shen et al., 2021), which demonstrates that the adoption of a data balancing technique based on the *oversampling* of the minority class can improve the performance of a credit scoring system, we will adopt it for preprocessing the used datasets during the experiments. A side effect related to the scarcity of real-world datasets and the reduced number of *unreliable* cases available for the training of the evaluation model is the *cold start*. It means that until we have an adequate number of *unreliable* samples, we can not train an evaluation model, not even by recurring to an *oversampling* technique.

Performance Evaluation: In order to evaluate the performance of a credit scoring system, aimed at the binary classification of users as reliable or unreliable, different metrics are used in the literature, many of which based on the *confusion-matrix*, a matrix of size $2x2$ that contains the numbers of *True Negatives* (TN), *False Negatives* (FN), *True Positives* (TP), and *False Positives* (FP). Some of these metrics largely used are the $Accuracy = \frac{TP+TN}{TP+TN+FP+FN}$, the $Sensitivity = \frac{TP}{TP+FN}$ (also defined *True Positive Rate*), the $Specificity = \frac{TN}{TN+FP}$ (also defined *True Negative Rate*), the $Fallout = \frac{FP}{FP+TN}$ (also defined *False Positive Rate*), and the $Matthews\ Correlation\ Coefficient = \frac{TP \cdot TN}{\sqrt{(TP+FP)\cdot(TP+FN)\cdot(TN+FP)\cdot(TN+FN)}}$ (MCC).

The literature also shows how, in order to provide a more reliable assessment of the credit scoring performance with any data configuration, then regardless of the level of balance of the classes, these aforementioned metrics are usually combined with other metrics such as, for instance, those based on the *Receiver Operating Characteristic* (ROC) curve (Green and Swets, 1966) the most used of which is the *Area Under the ROC Curve* (AUC). The ROC plots the *Sensitivity* on the y-axis, against the *Fallout* on the x-axis, evaluating the separability, i.e., the ability to discriminate the two data classes (i.e., *reliable* and *unreliable*) correctly.

3 RTDS STRATEGY

Before describing the proposed strategy, we report the adopted formal notation. Denoting as $|S|$ the cardinality of a generic set S, we further denote a series of instances $I = \{i_1, i_2, \ldots, i_N\}$ composed by: a subset $I^+ = \{i_1^+, i_2^+, \ldots, i_X^+\}$ of *reliable* instances, then

$I^+ \subseteq I$; a subset $I^- = \{i_1^-, i_2^-, \ldots, i_Y^-\}$ of *unreliable* instances, then $I^- \subseteq I$; a subset $\hat{I} = \{\hat{i}_1, \hat{i}_2, \ldots, \hat{i}_M\}$ of unclassified instances, then $\hat{I} \subseteq I$.
So we have that $I = (I^+ \cup I^- \cup \hat{I})$, and each instance $i \in I$ is characterized by the features in the set $F = \{f_1, f_2, \ldots, f_W\}$, and it can belong to one of the classes in the set $C = \{reliable, unreliable\}$, then we also formalize: the *training set* $T = \{i_1, i_2, \ldots, i_K\}$ given by $I^+ \cup I^-$; the possibility to divide T into $R = \{r_1, r_2, \ldots, r_Z\}$ regions, according to the T instances (set rows) and features (set columns); the regions definition operation as $R_{(IR,FC)}$, where IR is the number of *Instance Rows*, and FC is the number of *Feature Columns*, then $|R| = Z = (IR \times FC)$.

As a result of the above: concerning the set I, each region is composed by $\frac{N}{IR}$ instances and $\frac{W}{FC}$ features, since $|I| = N$ and $|F| = W$; the bounds of IR and FC are, respectively, $1 \leq IR \leq |T|$ and $1 \leq FC \leq |F|$; it should be observed that $IR = FC = 1$ indicates the canonical data configuration and that the IR value must define a region with samples of both classes in the set C, differently, the training process of an evaluation model is not possible.

Problem Definition: Considering that we face the credit scoring problem in terms of a binary classification related to the two classes defined in the previously formalized set C, it is possible to define such a problem as shown in the Equation 1, where α denotes a generic classification algorithm, and the evaluation function of an instance $\hat{i}$ (that returns 1 when performs a correct classification and 0 otherwise) is denoted as $Evaluate(\hat{i}, \alpha)$. This means that the problem is formulated as the maximization of the Θ value, since it reports the sum of the instances correctly classified (its upper bound is then $|\hat{I}|$).

$$\max_{0 \leq \Theta \leq |\hat{I}|} \Theta = \sum_{m=1}^{|\hat{I}|} Evaluate(\hat{i}_m, \alpha) \tag{1}$$

Strategy Overview: The proposed strategy relies on a *fusion fashion* independent evaluation model, this means that several evaluation models are trained by using a different region of the $IR \times FC$ regions, where each of these regions bounds specific *user's instances* (rows) and *user's features* (columns). Based on this division into regions, the classification process is performed by several sub-processes, each of them based on the training data bounded by the respective region, according to our idea that such a strategy can reduce the problem related to the data heterogeneity because each new instance classification depends on a different group of instances and features.

Strategy Formalization: Based on the proposed RTDS strategy, the problem defined in Equation 1 needs to be revised by dividing the evaluation process into Z sub-processes, i.e., $|R| = Z$. Therefore, the generic credit scoring algorithm α runs Z times, and the final classification depends on all the results, as shown in Equation 2, which assumed $K = 4$, $W = 4$, $IR = 2$, and $FC = 2$, giving rise to a subdivision of the training set T into $|R| = Z = (2 \times 2) = 4$ regions, where each region is composed by $\frac{K}{IR} = \frac{4}{2} = 2$ instances and $\frac{W}{FC} = \frac{4}{2} = 2$ features, generating four m_1, m_2, m_3, m_4 evaluation models. In other words, the training process of an evaluation model m related to a classification algorithm α uses the instances and features bounded by the r_1, r_2, r_3, r_4 regions, individually, obtaining four m_1, m_2, m_3, m_4 evaluation models.

$$R_{(2,2)} = \left[\begin{array}{c|c} r_1 & r_2 \\ \hline r_3 & r_4 \end{array}\right] = \left[\begin{array}{cc|cc} f_{1,1} & f_{2,1} & f_{3,1} & f_{4,1} \\ f_{1,2} & f_{2,2} & f_{3,2} & f_{4,2} \\ \hline f_{1,3} & f_{2,3} & f_{3,3} & f_{4,3} \\ f_{1,4} & f_{2,4} & f_{3,4} & f_{4,4} \end{array}\right] \Rightarrow \left[\begin{array}{c|c} m_1 & m_2 \\ \hline m_3 & m_4 \end{array}\right] \tag{2}$$

Whereby, the process of classification of a new instance $\hat{i} \in \hat{I}$ will involve its f_1, f_2, f_3, f_4 features, which are compared to all the evaluation models m_1, m_2, m_3, m_4, producing the four classification c_1, c_2, c_3, c_4, as shown in Equation 3, where the comparison operation is denoted with $\Leftrightarrow$.

$$\begin{array}{l} c_1 = \left[m_1\right] \Leftrightarrow \left[f_1\ f_2\right] \quad c_2 = \left[m_2\right] \Leftrightarrow \left[f_3\ f_4\right] \\ c_3 = \left[m_3\right] \Leftrightarrow \left[f_1\ f_2\right] \quad c_4 = \left[m_4\right] \Leftrightarrow \left[f_3\ f_4\right] \end{array} \tag{3}$$

Padding Criterion: The *padding criterion* is used when the number of regions given by the IC and FC values does not generate equal-size regions, as shown inEquation 4.

$$\begin{array}{l} (|T| \bmod IR) \neq 0 \\ (|F| \bmod FC) \neq 0 \end{array} \tag{4}$$

According both to the notations $\mu_2 = (|T| \bmod IR)$ and $\mu_1 = (|F| \bmod FC)$ and $F = \{f_1, f_2, \ldots, f_W\}$ and $T = \{i_1, i_2, \ldots, i_K\}$, Equation 5 formalizes the *padding criterion*.

$$\begin{array}{l} \texttt{pad}(T) = \{i_1, i_2, \ldots, i_K, i_{K+1}, i_{K+2}, \ldots, i_{K+\mu_2}\} \\ \textit{with } i_{K+1} = i_K, i_{K+2} = i_{K-1}, \ldots = i_{K+\mu_2} = i_{K-\mu_2} \\ \texttt{pad}(F) = \{f_1, f_2, \ldots, f_W, f_{W+1}, f_{W+2}, \ldots, f_{W+\mu_1}\} \\ \textit{with } f_{W+1} = f_{W+2} = \ldots = f_{W+\mu_1} = f_W \end{array} \tag{5}$$

The adopted criterion is aimed at not altering the information significantly since it follows two different strategies: concerning T, it duplicates the last rows (instances) μ_2 times, facing the risk that the added instances belong to the same class in C; concerning F, it duplicates the last column of data (features) μ_1 times. This approach does not bias the machine learning process because it involves both the training and the test data. It should be noted that, in order to simplify the exposition of the proposed strategy, we assume that

this criterion is used as preprocessing step, automatically, during the definition of the regions.

Classification Criterion: The *classification criterion* is aimed to face the case when is not possible to apply a *majority criterion* during the ensemble classification. In this regard, taking into account that the application of the RTDS strategy (except in the case $IR = FC = 1$) generates $c_1, c_2, \ldots, c_Z$ classifications, this can lead to the two cases reported in Equation 6, where, differently from the `Case 2` that allows us the use of the *majority criterion* to perform a classification of the instance, in the `Case 1` it is not possible. For this reason, we need to introduce a discriminating element, which is an additional classification c_{Z+1} performed through a canonical training approach for the algorithm, then by using for this purpose the whole set E, obtaining as result the $c_1, c_2, \ldots, c_Z, c_{Z+1}$ classifications, which lead us to the `Case 2`.

$$\begin{array}{l} \texttt{Case 1}: Z = 2n, \ n \in \mathbb{N} \\ \texttt{Case 2}: Z = 2n-1, \ n \in \mathbb{N} \end{array} \tag{6}$$

In other words, by taking into consideration the scenario related to the `Case 1`, assuming $IR = FC = 2$, it leads toward m_1, m_2, m_3, m_4 classification models and c_1, c_2, c_3, c_4 classifications of an instance $\hat{i}$, then we add the the classification c_5 by training an additional evaluation model on the whole set T. This makes it possible to apply the X criterion The majority criterion can be apply by following the *classification criterion* ρ that is formalized in Equation 7, where c_1 and c_2 are, respectively, the elements *reliable* and *unreliable* of the set C.

$$\rho(\hat{e}) = \begin{cases} c_1, \ if \sum_{i=1}^{Z} \phi(c_i, c_1) > \sum_{i=1}^{Z} \phi(c_i, c_2) \\ c_2, \ if \sum_{i=1}^{Z} \phi(c_i, c_1) < \sum_{i=1}^{Z} \phi(c_i, c_2) \\ c_1, \ if \sum_{i=1}^{Z} \phi(c_i, c_1) = \sum_{i=1}^{Z} \phi(c_i, c_2) \wedge c_{Z+1} = c_1 \\ c_2, \ if \sum_{i=1}^{Z} \phi(c_i, c_1) = \sum_{i=1}^{Z} \phi(c_i, c_2) \wedge c_{Z+1} = c_2 \end{cases} \tag{7}$$

with

$$\phi(a,b) = \begin{cases} 0, \ if \ a \neq b \\ 1, \ if \ a = b \end{cases}$$

Classification Algorithm: The Algorithm 1 exploits the proposed RTDS Strategy in order to classify the new instances in the set $\hat{I}$: it takes as input the classification algorithm α, the training set T, the set of unclassified instances $\hat{I}$, and the values (IR) and (FC) for the division of the training set into regions, returning as output the classification of all the instances in the set $\hat{E}$.

Algorithm 1: RTDS strategy classifier algorithm.

Input: α=Classification algorithm, T=Training set, $\hat{I}$=Unevaluated instances, IR=Instances rows, FC=Feature columns
Output: κ=Classification of the $\hat{I}$ instances

procedure CLASSIFIER(α, T, $\hat{I}$, IR, FC)
  if Z is *even* **then** ▷ Verifies if the number of regions is even
    $m'' \leftarrow getTraining(\alpha, T)$ ▷ Trains model using the whole set T
  end if
  $R \leftarrow getRegions(T, IR, FC)$ ▷ Divides training set into regions
  for each $r \in R$ **do** ▷ Trains an evaluation model for each region
    $m \leftarrow getTraining(\alpha, r)$ ▷ Trains evaluation model
    $M.add(m)$ ▷ Stores evaluation model
  end for
  for each $\hat{i} \in \hat{I}$ **do** ▷ Processes instances in $\hat{I}$
    $R'' \leftarrow getRegions(\hat{e}, IR, FC)$ ▷ Divides instance into regions
    for each $m \in M$ **do** ▷ Gets all instances classifications
      $c \leftarrow getInstanceClass(m, R'')$ ▷ Classifies instance according to regions
      $C.add(c)$ ▷ Stores classification
    end for
    if Z is *even* **then** ▷ Verifies if the number of regions is even
      $c'' \leftarrow getInstanceClass(m'', \hat{e})$ ▷ Classifies instance according to the whole set T
      $C.add(c'')$ ▷ Adds classification to the set C
    end if
    $\kappa.add(getFinalClassification(\hat{i}, C))$ ▷ Gets and store final instance classification
  end for
  return κ ▷ Returns classification of $\hat{I}$ instances
end procedure

4 EXPERIMENTS

All the code related to this work was developed in the *Python* language with the *scikit-learn* (http://scikit-learn.org) library. We set the seed of the *pseudo-random number generator* to *1* to grant the experiments reproducibility. We also performed the independent-samples two-tailed Student's t-test, which showed no statistical difference between the results ($p > 0.05$).

The validation process was performed by using three real-world datasets widely used in the literature and publicly available(https://archive.ics.uci.edu/ml/machine-learning-databases/statlog/): the *Australian Credit Approval* (ACD), the *Default of Credit Card Clients* (DCD), and the *German Credit* (GCD) datasets, whose characteristics are summarized in Table 1.

Table 1: Datasets Characteristics.

Dataset name	Total instances	Reliable instances	Unreliable instances	Feature number	Unreliable (%)
ACD	690	307	383	15	55.50
DCD	30,000	23,364	6,636	24	22.12
GCD	1,000	700	300	24	30.00

The first two metrics used to evaluate the performance of the proposed RTDS strategy are the *Sensitivity* and the *Specificity*. These two metrics (formalized in Equation 8) assess, respectively, the *true positive rate* and the *true negative rate*, evaluating the capability of a credit scoring approach to classify the *reliable* and *unreliable* instances correctly.

$$Sensitivity(\hat{I}) = \frac{TP}{(TP+FN)}, \; Specificity(\hat{I}) = \frac{TN}{(TN+FP)} \tag{8}$$

In addition, we used the *AUC* since it allows us to evaluate the performance regardless of the level of data balancing. Considering the *reliable* (I_+) and *unreliable* (I_-) subsets of instances in I, it is formalized in the Equation 9, where α denotes all possible comparisons between the scores of each instance i, and the result in the range $[0,1]$ (where 1 indicates the best performance) is the average of them.

$$\alpha(i_+, i_-) = \begin{cases} 1, & if\ i_+ > i_- \\ 0.5, & if\ i_+ = i_- \\ 0, & if\ i_+ < i_- \end{cases} \tag{9}$$

$$AUC = \frac{1}{I_+ \cdot I_-} \sum_{1}^{|I_+|} \sum_{1}^{|I_-|} \alpha(i_+, i_-)$$

The experiments were performed by dividing each dataset into two parts: *in-sample* and *out-of-sample* parts, respectively 50% and 50%. The *in-sample* part is used to detect the best credit scoring algorithm to use as a competitor and in the ensemble approach related to the proposed RTDS strategy, in addition to the optimal values of *IR* and *FC*, and the *out-of-sample* part is instead used in order to perform the validation process. It should be added that a canonical *k-fold cross-validation* criterion with $k = 10$ is still used in all the experiments. In addition to the *k-fold cross-validation* criterion, this dataset division, largely used in the literature with regard to some crucial data domains (e.g., financial market forecasting), allows us to avoid any *over-fitting* (Hawkins, 2004) problem, considering that it operates a real separation between the data used to define and tune the evaluation model and the ones used for the performance evaluation. In order to avoid that after the oversampling process (performed by us as preprocessing step on all the datasets) the two classes of instances are contiguous, creating issues during the *k-fold cross-validation* and the division into the region operations (i.e. due to the absence of one of the two data classes in a data fold/region), the oversampled datasets were also shuffled. The algorithms used in the experiments are those reported in Table 2 together with their configuration.

Results and Discussion: As a first step we evaluate the state-of-the-art algorithms in the context of a canonical approach, i.e., in order to train the related evaluation models we use the whole *in-sample* subset of data, applying the *k-fold cross-validation* criterion. The results are shown in Table 3, where the *Average* column reports the mean value of the three used metrics and where the best performance for each metric is highlighted in bold, indicating as the most performing algorithm to use RFA (i.e., *Random Forests*).

Table 2: Algorithms Configuration.

Algorithm	Parameter	Value
AdaBoost (*ABA*)	*n_estimators*	50
	learning_rate	0.1
	algorithm	SAMME.R
Decision Tree (*DTA*)	*min_samples_split*	2
	max_depth	none
	min_samples_leaf	1
Gradient Boosting (*GBA*)	*n_estimators*	100
	learning_rate	0.1
	max_depth	3
Multilayer Perceptron (*MPA*)	*alpha*	0.0001
	max_iter	200
	solver	adam
Random Forests (*RFA*)	*n_estimators*	10
	max_depth	none
	min_samples_split	2

Table 3: Algorithms Canonical Performance.

Algorithm	Dataset	Sensitivity	Specificity	AUC	Average
ABA	ACD	0.8278	0.8372	0.8322	0.8324
DTA	ACD	0.7994	0.7763	0.7850	0.7869
GBA	ACD	0.8577	0.8454	0.8487	0.8506
MPA	ACD	0.7986	0.7937	0.7824	0.7916
RFA	ACD	**0.8811**	**0.8675**	**0.8716**	**0.8734**
ABA	DCD	0.7597	0.7398	0.7492	0.7496
DTA	DCD	0.7243	0.7398	0.7319	0.7320
GBA	DCD	0.7999	0.7654	0.7813	0.7822
MPA	DCD	0.5972	0.6649	0.5771	0.6131
RFA	DCD	**0.8377**	**0.8088**	**0.8224**	**0.8230**
ABA	GCD	0.7476	0.7547	0.7512	0.7512
DTA	GCD	0.6580	0.6865	0.6731	0.6725
GBA	GCD	0.7846	0.7745	0.7796	0.7796
MPA	GCD	0.7918	0.7566	0.7718	0.7734
RFA	GCD	**0.7987**	**0.7841**	**0.7940**	**0.7923**

As a second step we identify the optimal number of regions (i.e., *IR* and *FC* values) to partition the training set. Also in this case we use the whole *in-sample* subset of data and the *k-fold cross-validation* criterion. In this context we tested all the *IR* and *FC* values in the range $\{2,3,\ldots,6\}$ (i.e., the most significant range of values, and the pair of values *IR=1* and *FC=1* is not considered, as it refers to a canonical configuration without regions). We perform the evaluation using the average value of all the used metrics on the y-axis, since this offers a global vision of the strategy performance, considering that it takes into account both the capability to detect the *reliable* (*Sensitivity*) and *unreliable* (*Specificity*) cases, and the capability to discriminate them effectively (*AUC*). The results indicate *(IR=1, FC=2)* as optimal values in the context of all the datasets with the previously selected RFA algorithm.

As a last step we compare the canonical approach (denoted as *BASE*) based on the whole training set to

the proposed *RTDS* strategy configured according to the optimal number of regions defined in the previous step. The comparison process was performed in the context of the *out-of-sample* subset of data, applying the *k-fold cross-validation* criterion, and the results for each metric and in terms of the average of all metrics are reported, respectively, in Table 4 and Figure 1.

Table 4: Performance Comparison.

Approach	Algorithm	Dataset	Sensitivity	Specificity	AUC
BASE	RFA	ACD	**1.0000**	0.8000	0.9130
BASE	RFA	DCD	0.8507	0.8117	0.8302
BASE	RFA	GCD	0.7429	0.8286	0.7878
RTDS	RFA	ACD	0.9376	**0.9224**	**0.9266**
RTDS	RFA	DCD	**0.9111**	**0.8911**	**0.9008**
RTDS	RFA	GCD	**0.9259**	**0.9085**	**0.9166**

Based on the experimental results, we can make the following considerations:

- the experiments aimed to detect the optimal *IR* and *FC* values for each dataset show regions bounded only along with the features since these optimal parameters (i.e., *IR=1* and *FC=2* for all the datasets) do not split in terms of instances, and it depends on the nature of the training data, since in these datasets each row refers to a different user, differently from other domains where there is a relation between the dataset rows (time-series);
- additional experiments we conducted showed two aspects: the average value we used for the tuning of the *IR* and *FC* parameters leads to the same results of the *AUC* metric, proving the effectiveness of this combined metric as a criterion of optimization; it is possible an optimization based on a single metric, which leads toward different *IR* values (i.e., by *Sensitivity* in the ACD, DCD, and GCD datasets we get, respectively, 1, 2, and 6, and by *Specificity* in the ACD, DCD, and GCD datasets we get, respectively, 5, 3, and 3), but only in one case the *IR* value is different from 1 (i.e., 2 by *Sensitivity* in the ACD dataset), supporting the initial hypothesis we made;
- the comparison of the canonical approach of training to the proposed RTDS strategy, performed in the *out-of-sample* part of the datasets, shows that it outperforms the canonical approach, except for the *Sensitivity* in the context of the ACD dataset, but it is directly related to the increase of *unreliable* instances erroneously classified as *reliable*, as evidenced by the *Specificity* and *AUC* values;
- in more detail, in spite of a lower performance regarding *Sensitivity* in the ACD dataset (-6.24%) we get better performance in terms of *Specificity* ($+15.30\%$) and *AUC* ($+1.49\%$), analogously to the DCD dataset, where in terms of *Sensitivity*, *Specificity* and *AUC* we get, respectively, $+7.10\%$, $+9.78\%$, and $+8.50\%$, and to the GCD dataset, where we get in terms of *Sensitivity*, *Specificity* and *AUC*, respectively, $+24.63\%$, $+9.64\%$, and $+16.35\%$;
- the best performance of the proposed RTDS strategy is further highlighted by the average performance reported in Figure 1, where it outperforms the canonical approach in all the datasets;
- in the light of the above considerations, strengthened by the fact that the adoption of the *in-sample/out-of-sample* and *k-fold cross-validation* criteria ensure experimental results not biased by over-fitting (since they grant that the algorithm selection and the RTDS parameter tuning operations do not affect the results), the experiment demonstrated how the proposed RTDS strategy can improve the performance of a credit scoring system.

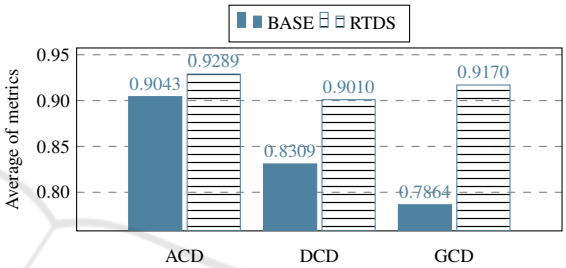

Figure 1: Average Performance.

5 CONCLUSIONS AND FUTURE DIRECTIONS

The *Region-based Training Data Segmentation* strategy proposed in this work relies on the idea that it is possible to improve the performance of a credit scoring system dividing the training set of a classification model into several regions, along with instances and features, using each of them to define an independent model, and obtaining the instance classification based on all classification models, according to an ensemble criterion regulated by a majority voting rule. The experimental results were performed on three real-world datasets by following an *in-sample/out-of-sample* criterion aimed at creating an effective separation between the data used for the operations of choosing the classification algorithm and those related to the tuning of the parameters of the proposed strategy, together with the *k-fold cross-validation* criterion, demonstrate the advantages of the proposed strategy. This is because its adoption leads toward an improvement of the credit scoring system in the context of all the datasets.

As future work, we would like to experiment with this strategy in different data domains, such as, for instance, those related to the *Intrusion Detection* (Saia et al., 2019) and *Fraud Detection* (Saia and Carta, 2017a) areas, in order to evaluate its effectiveness on

different nature of data, such as the time-series.

ACKNOWLEDGEMENTS

This research is partially funded and supported by: project “Studio per l’adeguamento di aree portale per tematismo - BRIC INAIL 2019 - FENU” CUP F24G20000100001”; “PON R&I 2014-2020 Action IV.6 - CUP F25F21002270003”.

REFERENCES

Carta, S., Fenu, G., Ferreira, A., Recupero, D. R., and Saia, R. (2019). A two-step feature space transforming method to improve credit scoring performance. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 134–157. Springer.

Carta, S., Ferreira, A., Recupero, D. R., Saia, M., and Saia, R. (2020). A combined entropy-based approach for a proactive credit scoring. *Engineering Applications of Artificial Intelligence*, 87:103292.

Carta, S., Ferreira, A., Recupero, D. R., and Saia, R. (2021). Credit scoring by leveraging an ensemble stochastic criterion in a transformed feature space. *Progress in Artificial Intelligence*, pages 1–16.

Dumitrescu, E., Hue, S., Hurlin, C., and Tokpavi, S. (2022). Machine learning for credit scoring: Improving logistic regression with non-linear decision-tree effects. *European Journal of Operational Research*, 297(3):1178–1192.

Green, D. M. and Swets, J. A. (1966). *Signal Detection Theory and Psychophysics*. Wiley, New York.

Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

Khemais, Z., Nesrine, D., Mohamed, M., et al. (2016). Credit scoring and default risk prediction: A comparative study between discriminant analysis & logistic regression. *International Journal of Economics and Finance*, 8(4):39.

Leevy, J. L., Khoshgoftaar, T. M., Bauder, R. A., and Seliya, N. (2018). A survey on addressing high-class imbalance in big data. *Journal of Big Data*, 5(1):42.

Lei, K., Xie, Y., Zhong, S., Dai, J., Yang, M., and Shen, Y. (2019). Generative adversarial fusion network for class imbalance credit scoring. *Neural Computing and Applications*, pages 1–12.

Liu, C., Huang, H., and Lu, S. (2019). Research on personal credit scoring model based on artificial intelligence. In *International Conference on Application of Intelligent Systems in Multi-modal Information Analytics*, pages 466–473. Springer.

Liu, W., Fan, H., and Xia, M. (2022). Credit scoring based on tree-enhanced gradient boosting decision trees. *Expert Systems with Applications*, 189:116034.

Park, S. and Park, H. (2021). Combined oversampling and undersampling method based on slow-start algorithm for imbalanced network traffic. *Computing*, 103(3):401–424.

Roy, A. G. and Urolagin, S. (2019). Credit risk assessment using decision tree and support vector machine based data analytics. In *Creative Business and Social Innovations for a Sustainable Future*, pages 79–84. Springer.

Saia, R. and Carta, S. (2016a). An entropy based algorithm for credit scoring. In *International Conference on Research and Practical Issues of Enterprise Information Systems*, pages 263–276. Springer.

Saia, R. and Carta, S. (2016b). Introducing a vector space model to perform a proactive credit scoring. In *International Joint Conference on Knowledge Discovery, Knowledge Engineering, and Knowledge Management*, pages 125–148. Springer.

Saia, R. and Carta, S. (2016c). A linear-dependence-based approach to design proactive credit scoring models. In *KDIR*, pages 111–120.

Saia, R. and Carta, S. (2017a). Evaluating credit card transactions in the frequency domain for a proactive fraud detection approach. In *SECRYPT*, pages 335–342.

Saia, R. and Carta, S. (2017b). A fourier spectral pattern analysis to design credit scoring models. In *Proceedings of the 1st International Conference on Internet of Things and Machine Learning*, page 18. ACM.

Saia, R., Carta, S., and Fenu, G. (2018). A wavelet-based data analysis to credit scoring. In *Proceedings of the 2nd International Conference on Digital Signal Processing*, pages 176–180. ACM.

Saia, R., Carta, S., Recupero, D. R., Fenu, G., and Stanciu, M. (2019). A discretized extended feature space (defs) model to improve the anomaly detection performance in network intrusion detection systems. In *KDIR*, pages 322–329.

Saia., R., Giuliani., A., Pompianu., L., and Carta., S. (2021). From payment services directive 2 (psd2) to credit scoring: A case study on an italian banking institution. In *Proceedings of the 13th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management - KDIR,*, pages 164–171. INSTICC, SciTePress.

Shen, F., Zhao, X., Kou, G., and Alsaadi, F. E. (2021). A new deep learning ensemble credit risk evaluation model with an improved synthetic minority oversampling technique. *Applied Soft Computing*, 98:106852.

Sloan, R. H. and Warner, R. (2018). When is an algorithm transparent? predictive analytics, privacy, and public policy. *IEEE Security & Privacy*, 16(3):18–25.

Weiss, G. M. (2004). Mining with rarity: A unifying framework. *SIGKDD Explor. Newsl.*, 6(1):7–19.

Zhang, W., He, H., and Zhang, S. (2019). A novel multi-stage hybrid model with enhanced multi-population niche genetic algorithm: An application in credit scoring. *Expert Systems with Applications*, 121:221–232.