IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 19 May 2023, accepted 5 June 2023, date of publication 8 June 2023, date of current version 15 June 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3284137

RESEARCH ARTICLE

Improving the Results in Credit Scoring by Increasing Diversity in Ensembles of Classifiers

SERAFÍN MORAL-GARCÍA AND JOAQUÍN ABELLÁN
Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

Corresponding author: Serafín Moral-García (seramoral@decsai.ugr.es)

This work was supported in part by UGR-Fondo Europeo de Desarrollo Regional (FEDER) Funds under Project A-TIC-344-UGR20; and in part by the "FEDER/Junta de Andalucía-Consejería de Transformación Económica, Industria, Conocimiento y Universidades" under Project P20_00159.

ABSTRACT The ensembles of classifiers are techniques that have obtained excellent results in the credit scoring domain. It is known that Decision Trees (DTs) are suitable for ensembles because they encourage diversity, the key point for the success of an ensemble scheme. Ensembles of DTs have obtained good performance in a wide range of areas, including credit scoring. Some works have highlighted that DTs that employ imprecise probability models, called Credal Decision Trees (CDTs), improve the results of ensembles in credit scoring. The performance of CDT is strongly influenced by a hyperparameter. In fact, it was shown that different values of the hyperparameter might yield different models. Hence, the diversity in ensemble schemes can be increased by randomly selecting the value of the hyperparameter in each CDT, instead of fixing one. In this work, it is shown that increasing the diversity of the ensembles that use CDT by varying the value of the hyperparameter in each base classifier improves the results in credit scoring. Thereby, the use of CDT randomly selecting the value of the hyperparameter would suppose notable economic benefits for banks and financial institutions. Few gains in accuracy might imply huge gains in economic benefits.

INDEX TERMS Credit scoring, ensemble schemes, diversity, base classifiers, random credal decision tree.

I. INTRODUCTION

Credit scoring is one of the main tools to analyze credit risk. It consists of a set of methods that allow classifying credit applicants into two classes: good or bad. These techniques help to make decisions about granting credit to an applicant. In consequence, credit scoring systems are very useful for banks and financial institutions. Actually, any slight improvement would produce great benefits in terms of faster decisions, risk reduction, and less cost in credit analysis [1].

So far, many approaches have been applied to credit scoring. They can be divided into *statistical methods* and *Artificial Intelligent* techniques [2]. The methods belonging to the first category assume previous knowledge about the data. These prior assumptions are not always realistic. In contrast, Artificial Intelligence techniques do not need prior knowledge about the data to extract information directly from them. Artificial Intelligence approaches, especially from the Data Mining field, yield better results than statistical methods in credit scoring [3], [4], [5].

Within Data Mining, many individual algorithms have been applied to credit scoring, such as Support Vector Machines (SVM) [6], [7], [8], Decision Trees (DT) [9], [10], [11], K-nearest neighbors (KNN) [12], Neural Networks (NN) [13], [14], [15], [16] and Bayesian Networks (BN) [17], [18], [19]. Nonetheless, the data mining techniques that have achieved the best results in credit scoring are the *ensembles* of classifiers [20], [21], [22], [23], [24]. They consider multiple individual classifiers and, then, the pieces of information obtained by them are combined to make a final prediction.

Experimental studies have been carried out to determine which base classifiers are the most suitable for ensembles in credit scoring [21], [25]. NNs have obtained satisfactory results as individual classifiers for credit scoring [14], but they perform worse than other individual learners in ensemble schemes for credit scoring datasets [25], [26]. The reason is that ensembles do not have to use very accurate and complex individual learners. As pointed out by Breiman [27], the key

The associate editor coordinating the review of this manuscript and approving it for publication was Jad Nasreddine.

issue for the success of an ensemble scheme is that the base classifiers are not only accurate but also *diverse* or *unstable*.

Several experimental studies have been carried out to compare the performance of several individual classifiers in the main ensemble schemes for credit scoring [21], [28]. These studies have shown that the base classifiers that obtain the best results are the unstable classifiers, not necessarily the most accurate as individual learners.

Decision trees (DTs) are individual classifiers that are built fast and are easy to interpret. DTs are well-known to be very unstable since, with these classifiers, little variations in the training sets might give rise to considerable variations in the learned models. Thereby, DTs are very suitable individual learners to employ in ensembles because they enhance diversity for the combination of classifiers [27].

DTs based on imprecise probabilities, called Credal Decision Trees (CDTs), were proposed in [29]. They use uncertainty measures on credal sets (closed and convex sets of probability distributions) in the tree-building process. CDTs have obtained good results, especially with class noise in the data [30], [31], [32]. Furthermore, in [33], it was highlighted that CDTs also have very good performance when they are used in Bagging schemes. In addition, CDTs have shown to be the individual learners that provide the best results in ensembles of classifiers for credit scoring [28], [34].

CDTs have an important hyperparameter that strongly influences their performance [31], [35]. As shown in [36], different values of the hyperparameter may lead to different trees when they are built with the same training set. Therefore, when CDT is the base classifier of an ensemble scheme, diversity can be increased by varying the value of the hyperparameter in each tree.

Considering the previous points, in this work, we propose the use of a base classifier in ensemble schemes for credit scoring datasets called Random Credal Decision Tree (RCDT). It consists of a version of the CDT algorithm that randomly chooses the value of the hyperparameter before building the tree within a range of possible values. Hence, the value might be different for each base classifier of the ensemble in such a way that the diversity of the ensembles is increased.

An experimental analysis is carried out in this work with several credit scoring datasets and the main ensemble schemes that have been employed in credit scoring in the literature. As the individual classifiers, we use RCDT, as well as the ones that achieved the best results in ensembles in an experimental study for credit scoring carried out in [28]. This analysis shows that, as expected, RCDT is the individual classifier that obtains the best results in ensembles for credit scoring datasets.

This paper is arranged as follows: In Section II, the main ensemble schemes used for credit scoring are detailed. Section III describes Credal Decision Trees. Section IV introduces the Random Credal Decision Tree algorithm. Our experimental study is detailed in Section V. Section VI concludes this paper.

II. ENSEMBLES OF CLASSIFIERS

In many areas of science, it is very common to combine multiple opinions to make a decision. In this way, that decision is probably better than the one made from a single opinion. This idea has also been applied in Machine Learning via *ensemble schemes*. They consider multiple classifiers and, for classifying an instance, the predictions made by the individual learners are normally combined via a scheme vote.

In the literature, the combination of information obtained from different classifiers has supposed improvement in classification over the use of single classifiers. Indeed, "with ensembles of classifiers, the predictions tend to be more accurate, and the robustness is increased" [37].

The following ensemble approaches, very known in the data mining area, have been employed:

- **Bagging** [27]: "This method creates M samples $S_1, \ldots, S_m$ randomly drawn from the original training set with replacement. The size of the samples coincides with the size of the original set. Hence, in each sample S_i, $1 \leq i \leq M$, some instances might appear more than once, whereas others may not appear. A classifier C_i is trained from each sample S_i, $\forall i = 1, \ldots, M$. Since each classifier is built with a different training set, the learners are often different from each other. To classify an instance, the predictions made by the classifiers are combined via a majority vote".
- **Boosting** [38]: "It also creates M samples randomly choosen with replacement. However, unlike Bagging, the resampling is directed to obtain the most informative data for each consecutive learner. When an instance is required to be classified, the predictions of the individual classifiers, weighted by their accuracy, are combined". The most known Boosting method is Adaboost [39]. In this algorithm, "the successive samples are obtained by re-weighting the training instances. Initially, the same weight is assigned to all instances. In each iteration, these weights are adjusted depending on the classification errors made by the resulting base learners in such a way that instances erroneously classified are more probable to appear in the next sample" [39].
- **Random Subspace** [40]: This approach considers M classifiers built with F attributes randomly selected from the original attribute set, that is, each classifier is learned with the same instances as the original training set but using only F features. Experiments have revealed that "the standard value of F that yields good results is equal to half of the total number of attributes" [41]. In order to classify an instance, the predictions are combined via the majority vote.
- **DECORATE** [42] (Diverse Ensemble Creation by Oppositional Relabelling of Artificial Training Examples): "It generates an ensemble by learning a new classifier in each iteration. The first base classifier is built with the original training set. The remaining learners are built with an artificial training set resulting from the union of the original set and artificial instances,

which are generated from the data distribution and obtained by probabilistically estimating the value of each attribute [42]. The labels of the artificial instances are selected so that they maximally differ from the current predictions. Hence, diversity is increased. For maintaining training accuracy, the classifier is added to the ensemble scheme if, and only if, its incorporation does not decrease the performance of the ensemble''.

- **Rotation Forest** [43]: ``This ensemble approach separates the predictive variables into F non-overlapping subsets equally sized. As in bagging, bootstrap sampling is carried out. Then, a principal component analysis (PCA) is run separately in each subset, and a new set of variables is extracted by integrating all principal components. Thus, K axis rotations happen to obtain the new attributes used to build the base classifier. Each base learner of the ensemble is built from the bootstrap sample in a rotate feature space''.

A. DIVERSITY IN ENSEMBLES

In ensemble schemes, there is no much gain combining very similar classifiers. Consequently, ``to the ensemble schemes are successful, the base classifiers must be not only accurate, but few variations in training data have to give rise to considerable changes in the model, which is known as *instability* or *diversity*'' [44].

For the previous reason, Decision Trees (DTs) are frequently used in ensemble schemes; they are very simple and unstable models, which enhance the diversity of the ensembles of classifiers.

Experimental analyses carried out in [28] and [34] have shown that the base classifiers than perform better in ensemble schemes for credit scoring are not the ones that achieve the best results as individual learners, but DTs. It is because DTs are accurate and diverse classifiers and, thus, very suitable for ensembles.

Other ways for increasing diversity in the set of base classifiers for an ensemble method are:

- Randomly selecting samples from the original training set with replacement to build each base classifier. Examples of ensemble methods that use this procedure are Bagging and Rotation Forest. Some areas of the search space of the problem may not be studied by the base classifiers because these areas are hidden by the more frequent samples. With the random selection of instances for each base classifier, some base classifiers can study these zones with less frequent samples and, in this way, highlight some interesting characteristics for improving the accuracy and robustness of the ensemble method.
- Randomly selecting attributes from the original set of variables to build each base classifier. Examples of ensemble methods with this property are Random Subspace and Random Forest. Some attributes might not be used by the base classifiers because they are hidden by other more important attributes according to the ranking of the variables established by the base classifier. However, these hidden attributes can provide interesting information for the ensemble classifier. Hence, the random selection of attributes can give an opportunity to the hidden attributes for providing their knowledge to the ensemble method. Thereby, this method can improve its accuracy and generalization.

III. CREDAL DECISION TREES

Decision Trees (DTs) have been widely used in the classification task since the publication of the ID3 algorithm [45]. A few years later, Quilan also proposed the C4.5 classifier [46]. Since then, C4.5 has been a standard classification algorithm. Furthermore, DTs have been commonly applied to many areas, such as biology, medicine, and astronomy.

In a DT structure, each node corresponds to an attribute or feature. Each branch between a node and its child is associated with a possible value of that feature, and each leaf or terminal node is labeled with a class value.

Once the tree has been built, to classify a new instance, ``a path from the root node to a leaf is followed using its attribute values. The predicted class value is the one labeled to the terminal node'' [41].

The tree-building process is determined by the following points:

1) The criterion utilized to choose the feature to insert in a node, i.e, the *split criterion*.
2) The criterion to stop ramifying.
3) The criterion to assign a class label to the terminal nodes.
4) Optionally, a DT can use a final post-pruning process to simplify the tree structure and reduce the over-fitting.

A terminal node is labeled with the most frequent class value in it. Regarding point 2, the branching is stopped when there is no information gain for any attribute.

Hence, the main difference among the different DTs is the split criterion.

Let C be the class variable and $\{c_1, \ldots, c_k\}$ its possible values. Let $\mathcal{D}$ denote the dataset associated with a node. Let X be an attribute that takes values in $\{x_1, \ldots, x_t\}$.

Classical DTs use precise probabilities in the split criterion. Specifically, the basis of the split criterion in them is the Shannon Entropy [47] of the class variable, defined as:

$$H^{\mathcal{D}}(C) = \sum_{i=1}^{k} -p(c_i) \log p(c_i). \tag{1}$$

The average entropy that derives from the feature X is determined by:

$$H^{\mathcal{D}}(C \mid X) = \sum_{j=1}^{t} P^{\mathcal{D}}(X = x_j) H^{\mathcal{D}_j}(C \mid X = x_j), \tag{2}$$

$P^{\mathcal{D}}(X = x_j)$ being the probability that $X = x_j$ in $\mathcal{D}$ and $\mathcal{D}_j$ the set of instances of $\mathcal{D}$ such that $X = x_j, \quad \forall j = 1, 2, \ldots, t$.

The split criterion used in many classical DTs, called *Info-Gain*, is defined as follows:

$$IG(C,X)^{\mathcal{D}} = H^{\mathcal{D}}(C) - H^{\mathcal{D}}(C \mid X). \tag{3}$$

In these trees, the attribute that provides the maximum value of IG is selected for branching.

Credal Decision Trees (CDTs) were proposed by Abellán and Moral [29]. In the split criterion, they utilize uncertainty measures on credal sets, i.e, closed and convex sets of probability distributions.

Specifically, CDTs employ the Imprecise Dirichlet Model (IDM) [48], a formal imprecise probability model that uses probability intervals. The IDM estimates that, $\forall j = 1, 2, \ldots, k$, the probability that $C = c_j$, denoted by $p(c_j)$, is within the following interval:

$$p(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s}\right], \tag{4}$$

where N is the number of instances in the dataset, n_{c_j} is the number of instances in the dataset that verify that $C = c_j, \quad \forall j = 1, \ldots, k$, and s is a given hyperparameter. It is easy to check that IDM intervals are wider as the s value is higher. In [48], a definitive recommendation for the s value was not given, although two values were suggested: $s = 1$ and $s = 2$, and it was recommended $s = 1$.

These probability intervals lead to the following credal set on C [49]:

$$K^{\mathcal{D}}(C) = \left\{ p \mid p(c_j) \in \left[\frac{n_{c_j}}{N+s}, \frac{n_{c_j}+s}{N+s}\right], \right. \\ \left. \forall j = 1, \ldots, k, \sum_{j=1}^{k} p(c_j) = 1 \right\}. \tag{5}$$

Uncertainty measures can be applied on this credal set. The maximum entropy is the one used in the split criterion of CDTs. It is determined by:

$$H^*(K^{\mathcal{D}}(C)) = \max \left\{ H^{\mathcal{D}}(p) \mid p \in K^{\mathcal{D}}(C) \right\} \tag{6}$$

The maximum entropy is a well-established uncertainty measure because it verifies the required properties. It is consistent with the *principle of maximum uncertainty* [50], which states that "the probability distribution that attains the maximum entropy, compatible with the available restrictions, should be chosen" [51].

The procedure to obtain H^* was exposed in [49]. It reaches its lowest computational cost when $s \leq 1$.

The split criterion utilized in CDTs, called the *Imprecise Info-Gain* (IIG) [29], is similar to IG. Nevertheless, the former is based on the maximum entropy in $K^{\mathcal{D}}(C)$, whereas IG uses the Shannon entropy. Formally, IIG is defined as:

$$IIG^{\mathcal{D}}(C,X) = H^*(K^{\mathcal{D}}(C)) - H^*(K^{\mathcal{D}}(C \mid X)) \tag{7}$$

where the value of $H^*(K^{\mathcal{D}}(C \mid X)$ is obtained similarly to $H^{\mathcal{D}}(C \mid X)$ in IG.

Algorithm 1 summarizes the building procedure of a CDT.

Algorithm 1 Building Procedure of a CDT

```
Procedure Build_CDT(Node $\mathcal{N}$, set of features $\mathcal{F}$)
if $\mathcal{F} = \emptyset$ then
    Exit
end
1. Let $\mathcal{D}$ be the dataset corresponding to $\mathcal{N}$
if $|\mathcal{D}| <$ minimum number of instances then
    Exit
end
2. $max\_entropy = \max_{X \in \mathcal{F}} IIG^{\mathcal{D}}(C,X)$
if $max\_entropy \leq 0$ then
    Exit
end
else
    3. Let $X' \in \mathcal{F}$ be the attribute that reaches
      $max\_entropy$
    4. Assign $X'$ to $\mathcal{N}$
    5. $\mathcal{F}' = \mathcal{F} \setminus \{x'\}$
    for $x_i$ possible value of $X'$ do
        6. Add a node $\mathcal{N}_i$ child of $\mathcal{N}$
        7. Build_CDT($\mathcal{N}_i$, $\mathcal{F}'$)
    end
end
```

A. CLASSICAL DTs VS CREDAL DECISION TREES

When $s = 0$, the credal set determined by Equation (5) only contains the probability distribution corresponding to relative frequencies, and, consequently, CDTs and classical DTs are identical. If $s > 0$, then IDM intervals are wider as the size of the set (N) is larger. Thus, at the upper levels of the tree, where N is often very large, IIG and IG tend to obtain similar values and, therefore, classical DTs and CDTs may behave similarly. Nonetheless, at the lower levels of the tree, where N is usually small, the IDM credal set probably contains many probability distributions far different from the one used to compute the Shannon entropy. Hence, at the lower levels of the tree, CDT and classical DTs might have notably different behavior since IIG and IG may provide very different values.

Unlike IG, the value of IIG can be negative for an attribute [26]. In consequence, CDTs avoid choosing attributes that worsen the information about the class variable. Thereby, CDTs might stop branching the tree before classical DTs, which means that CDTs usually overfit less the data than classical DTs.

As pointed out in [35], CDTs are less sensitive to noise than DTs based on precise probabilities. Moreover, in [28], [34], it was highlighted that CDTs perform much better than classical DTs when they are employed in ensembles for credit scoring datasets.

IV. RANDOM CREDAL DECISION TREE

The Random Credal Decision Tree algorithm (RCDT) randomly chooses a value for the s hyperparameter before building the tree between a given range of possible values. Recall that this value is used in the calculation of the split criterion of the algorithm. Then, the tree is built in the same way as CDT. Thus, the building process of an RCDT can be summarized as follows:

1) Select s' a random value of the s hyperparameter.
2) Build the tree using s' in the split criterion.

For classifying a new instance, the RCDT algorithm carries out the same procedure as CDT.

Concerning the range of values for the s hyperparameter, in this work, we use $\{1, 1.5, 2, 2.5, 3, 3.5\}$, the one established as appropriated in [36].[1]

A. JUSTIFICATION OF RCDT

In [36] and [49], it was demonstrated that, if $s_1 > s_2$, then the credal set determined by Equation (5) with s_1 as the value of the s hyperparameter contains the credal set corresponding to s_2. In consequence, the maximum entropy value for the class variable does not decrease as the s value is higher.

Also, for an attribute X, if $s_1 > s_2$, then the maximum entropy of the class variable conditioned on X is greater for s_1 than for s_2. Given two attributes X_1 and X_2, it is possible that $H^*_{s_1}(C \mid X_1) - H^*_{s_2}(C \mid X_1) > H^*_{s_1}(C \mid X_2) - H^*_{s_2}(C \mid X_2)$, where $H^*_{s_i}$ is the maximum entropy on the credal set corresponding to $s = s_i$, for $i = 1, 2$. Hence, different values of the s hyperparameter might yield different split attributes via the IIG criterion. This is highlighted in the following example.

Example 1: Suppose that we have two class values, namely c_1 and c_2. Let us assume that, in a certain node, we have 5 instances for which $C = c_1$ and 10 instances for which $C = c_2$. Suppose that, in this node, there are three attributes X_1, X_2, and X_3, and that each one of them takes values in $\{x^i_1, x^i_2\}$, for $i = 1, 2, 3$. Let us assume the following class frequencies for each attribute value:

$$\begin{aligned}
X_1 &= x^1_1 \rightarrow (n_{c_1} = 4, n_{c_2} = 10) \\
X_1 &= x^1_2 \rightarrow (n_{c_1} = 1, n_{c_2} = 0) \\
X_2 &= x^2_1 \rightarrow (n_{c_1} = 4, n_{c_2} = 4) \\
X_2 &= x^2_2 \rightarrow (n_{c_1} = 1, n_{c_2} = 6) \\
X_3 &= x^3_1 \rightarrow (n_{c_1} = 3, n_{c_2} = 9) \\
X_3 &= x^3_2 \rightarrow (n_{c_1} = 2, n_{c_2} = 1)
\end{aligned}$$

We have the following maximum entropy values for $s = 0$, $s = 1$ and $s = 2$:

$$\begin{aligned}
H^*(K^{\mathcal{D}}(C))_{s=0} &= 0.9183. \\
H^*(K^{\mathcal{D}}(C))_{s=1} &= 0.9544. \\
H^*(K^{\mathcal{D}}(C))_{s=2} &= 0.9774.
\end{aligned}$$

The change of the maximum entropy value when s passes from 0 to 1 is notably higher than when s passes from 1 to 2.

Regarding the conditioned entropies, we have the following values:

$$\begin{aligned}
H^*(K^{\mathcal{D}}(C \mid X_1))_{s=0} &= 0.8056. \\
H^*(K^{\mathcal{D}}(C \mid X_1))_{s=1} &= 0.9237. \\
H^*(K^{\mathcal{D}}(C \mid X_1))_{s=2} &= 0.9575. \\
H^*(K^{\mathcal{D}}(C \mid X_2))_{s=0} &= 0.8094. \\
H^*(K^{\mathcal{D}}(C \mid X_2))_{s=1} &= 0.9119. \\
H^*(K^{\mathcal{D}}(C \mid X_2))_{s=2} &= 0.9619. \\
H^*(K^{\mathcal{D}}(C \mid X_3))_{s=0} &= 0.8327. \\
H^*(K^{\mathcal{D}}(C \mid X_3))_{s=1} &= 0.9124. \\
H^*(K^{\mathcal{D}}(C \mid X_3))_{s=2} &= 0.9522.
\end{aligned}$$

Again, the most notable changes are produced when s passes from 0 to 1.

The values of IIG for each attribute and for each attribute and value of the s hyperparameter are the following ones:

$$\begin{aligned}
IIG_{s=0}(C, X_1) &= H^*(K^{\mathcal{D}}(C))_{s=0} - H^*(K^{\mathcal{D}}(C \mid X_1))_{s=0} \\
&= 0.1127. \\
IIG_{s=0}(C, X_2) &= H^*(K^{\mathcal{D}}(C))_{s=0} - H^*(K^{\mathcal{D}}(C \mid X_2))_{s=0} \\
&= 0.1089. \\
IIG_{s=0}(C, X_3) &= H^*(K^{\mathcal{D}}(C))_{s=0} - H^*(K^{\mathcal{D}}(C \mid X_3))_{s=0} \\
&= 0.0856. \\
IIG_{s=1}(C, X_1) &= H^*(K^{\mathcal{D}}(C))_{s=1} - H^*(K^{\mathcal{D}}(C \mid X_1))_{s=1} \\
&= 0.0307. \\
IIG_{s=1}(C, X_2) &= H^*(K^{\mathcal{D}}(C))_{s=1} - H^*(K^{\mathcal{D}}(C \mid X_2))_{s=1} \\
&= 0.0425. \\
IIG_{s=1}(C, X_3) &= H^*(K^{\mathcal{D}}(C))_{s=1} - H^*(K^{\mathcal{D}}(C \mid X_3))_{s=1} \\
&= 0.0420. \\
IIG_{s=2}(C, X_1) &= H^*(K^{\mathcal{D}}(C))_{s=2} - H^*(K^{\mathcal{D}}(C \mid X_1))_{s=2} \\
&= 0.0199. \\
IIG_{s=2}(C, X_2) &= H^*(K^{\mathcal{D}}(C))_{s=2} - H^*(K^{\mathcal{D}}(C \mid X_2))_{s=2} \\
&= 0.0155. \\
IIG_{s=2}(C, X_3) &= H^*(K^{\mathcal{D}}(C))_{s=2} - H^*(K^{\mathcal{D}}(C \mid X_3))_{s=2} \\
&= 0.0252.
\end{aligned}$$

In this way, when $s = 0$, the selected feature is X_1. However, the X_2 attribute is chosen when $s = 1$, whereas, for $s = 2$, the feature X_3 is selected.

Consequently, in this case, the choice of the s value decisively influences the split attribute.

The above example shows that the value of the s hyperparameter might be crucial for the split attribute in a node. Hence, different s values may lead to different tree structures. In this way, in an ensemble, the diversity is increased by randomly choosing the value of the s hyperparameter in each CDT. For this reason, in ensemble schemes, it is more suitable to employ RCDT as the base classifier than CDT with the same s value for all trees. This is validated in Section V with exhaustive experimentation.

[1] We do not use s=0 because, in such a case, RCDT would be equivalent to classical DTs. We do also not employ very larger values of s since it would suppose a quite high pruning in the trees [26].

TABLE 1. Datasets description.

Dataset	N	Features	%Good	% Bad
Australian	690	14	44.5	55.5
German	1000	24	70	30
Iranian	1000	27	95	5
Japanese	653	15	45.3	54.7
Polish	240	30	53.3	46.6
UCSD	2435	38	75.4	24.6

V. EXPERIMENTS

A. EXPERIMENTAL SETUP

We have based on the experimental study about the performance of individual classifiers in ensembles for credit scoring datasets carried out in [28], considering the same experimental setting.

In this experimental analysis, six credit scoring datasets have been employed. Table 1 allows us to observe the main characteristics of each dataset: number of instances, number of attributes, and percentages of instances labeled as good/positive and bad/negative.

The datasets *Australian*, *German*, and *Japanese* can be found in the *UCI Machine Learning repository* [52]. The *Iranian* dataset [53] "comes from the corporate client of the work of a small private bank in Iran". The *Polish* dataset derives from the research carried out in [54] about companies of bankruptcy forecast. The *UCSD* dataset "is a reduced version of a very large dataset employed in the 2007 Data Mining Contest of the University of California, San Diego" [41].

In our experimental analysis, five ensemble schemes have been used: Adaboost, Bagging, Random Subspace, DECORATE, and Rotation Forest. They were described in Section II. Regarding the base classifiers, we have used RCDT and the ones that achieved the best results in [28]: LOG-R, C4.5, and CDT. We aim to analyze the performance of the base classifiers in each ensemble for the credit scoring datasets considered in this research.

For our experimentation, we have utilized the *Weka* software [55]. The implementations provided in this software for the ensemble schemes have been employed, as well as the implementations for LOG-R, C4.5, and CDT. We have added the structures and methods required for using the RCDT algorithm. Consistently with the experimental analysis carried out in [28], for CDT, we have fixed the s value to 1, and the post-pruning process has not been considered. In both CDT and RCDT, the missing values and continuous attributes have been treated as in the C4.5 algorithm [26]. Remark that, for RCDT, the range of values $\{1, 1.5, 2, 2.5, 3, 3.5\}$ has been used, the one established as the most suitable in [36]. The rest of the parameters for the algorithms have been the ones given by default in Weka.

As in the experimental analyses carried out in [21] and [28], for each pair ensemble/classifier and dataset, a 5-fold cross-validation procedure has been repeated 50 times.

Regarding the evaluation of the performance, remark that Accuracy is not the most suitable evaluation measure for credit scoring since, in this field, a false positive may have worse consequences than a false negative. For this reason, as in [28], we have not only used Accuracy, but we have also considered the area under the ROC curve (AUC). It is a well-established evaluation measure in the literature for binary classification problems where the error costs are different.

Following the recommendations given in [56], for statistical comparisons in Accuracy and AUC, we have used the following statistical tests to compare the results obtained by more than two classifiers, with a level of significance of $\alpha = 0.1$:

- **Friedman test** [57]: "It is a non-parametric test that separately ranks the algorithms for each dataset (the best-performing algorithm is assigned to rank 1, the second-best to rank 2, and so on). The null hypothesis of the Friedman test is that all algorithms perform equivalently".
- If the null hypothesis of the Friedman test is rejected, then all algorithms are compared to each other via the **Holm test** [58].

B. RESULTS AND DISCUSSION

In order to analyze the results, we principally consider Friedman's ranks for Accuracy and AUC. In the literature, it has been considered an important reference when it is required to compare the performance of several algorithms [56]. Apart from the order in the performance, we also want to know the cases in which the differences are statistically significant. In some cases, the differences among the base classifiers are hardly notable. Nevertheless, as said previously, small gains in performance might imply important gains in economical benefits, which is the main goal.

1) ACCURACY RESULTS

Table 2 shows the average accuracy results obtained by each ensemble with each base classifier for each dataset. It also allows us to see Friedman's ranks of the base classifiers in each ensemble scheme. For each ensemble, the best result is marked with bold fonts and the second-best with italic fonts. Table 3 presents a summary of Friedman's ranks obtained by each base classifier in Accuracy. Specifically, it shows the Friedman ranks of the base classifiers in each one of the ensemble schemes, as well as the average Friedman rank for each base classifier.

Taking into account Tables 2 and 3, we express the following comments about the Accuracy results for each base classifier:

1) **LogR**:
   - It is the worst base classifier concerning Friedman's ranks in all ensembles, except for DECORATE, where it achieves the second-best results.
   - This base classifier obtains the worst average Friedman rank.
   - It obtains 9 of the best performances out of 30 by ensemble/dataset. For only 1 pair

TABLE 2. Average accuracy results obtained by each base classifier in each dataset grouped by ensemble.

Ensemble	Base	Australian	German	Japanese	Iranian	Polish	UCSD	Rank
AdaBoost	LogR	84.93	**75.70**	**87.29**	94.20	72.92	83.86	2.8333
	C4.5	82.90	72.30	*85.91*	**95.10**	*75.42*	85.83	*2.6667*
	CDT	*85.26*	72.48	85.87	94.42	74.72	*85.93*	*2.6667*
	RCDT	**85.94**	*74.72*	85.79	*94.63*	**77.91**	**88.26**	**1.8333**
Bagging	LogR	84.93	**76.20**	**87.75**	94.30	73.33	84.07	3
	C4.5	85.94	73.70	*86.83*	94.70	**77.08**	86.28	2.6667
	CDT	*86.30*	*74.78*	86.46	*94.75*	75.45	*86.32*	*2.6667*
	RCDT	**87.06**	74.81	86.60	**95.04**	*76.98*	**86.97**	**1.6667**
Random Subspace	LogR	85.94	**75.50**	85.45	94.80	72.92	83.78	3.4167
	C4.5	85.65	73.90	85.60	**95.10**	*76.25*	85.26	2.8333
	CDT	*85.99*	74.14	*86.24*	95.02	75.70	*86.52*	*2.5*
	RCDT	**86.35**	**75.50**	**86.66**	*95.06*	**76.60**	**87.05**	**1.25**
DECORATE	LogR	84.64	**77.40**	**87.13**	**94.70**	74.17	83.78	*2.5*
	C4.5	*85.94*	73.20	84.23	94.20	74.58	*84.89*	2.9167
	CDT	85.20	*73.28*	84.86	94.20	*76.31*	84.31	2.75
	RCDT	**85.97**	73.00	*85.32*	*94.52*	**76.97**	**85.82**	**1.8333**
Rotation Forest	LogR	84.93	*76.10*	**87.29**	94.20	72.92	84.11	3.1667
	C4.5	85.94	74.50	*87.13*	**95.20**	*76.67*	86.16	*2.5*
	CDT	*86.10*	75.20	86.40	94.76	76.50	*86.66*	2.8333
	RCDT	**86.59**	**76.22**	86.63	*94.97*	**78.35**	**87.19**	**1.5**

TABLE 3. Results of the Friedman's ranks about accuracy of each base classifier.

Base	Adaboost	Bagging	Random Subspace	DECORATE	Rotation Forest	Average
LogR	2.8333	3	3.4167	*2.5*	3.1667	2.9833
C4.5	*2.6667*	*2.6667*	2.8333	2.9167	*2.5*	2.7167
CDT	*2.6667*	*2.6667*	*2.5*	2.75	2.8333	*2.6834*
RCDT	**1.8333**	**1.6667**	**1.25**	**1.8333**	**1.5**	**1.7333**

ensemble/dataset, it achieves the second-best performance.

- *German* and *Japanese* are the datasets in which Log-R performs better. For these datasets, it obtains the best results for four ensembles.

2) **C4.5**:

- It obtains the second-worst average Friedman rank, after LogR.
- C4.5 occupies the second position regarding Friedman's ranks, tied with CDT, in Adaboost and Bagging. It obtains the worst Friedman rank in DECORATE, and it achieves the second-best performance in Rotation Forest.
- This base classifier has 4 wins by ensemble/dataset. For 8 combinations ensemble/dataset, C4.5 occupies the second position.
- *Iranian* is the dataset for which C4.5 performs better. Indeed, all its wins occur in this dataset.

3) **CDT**:

- It obtains the second-best average Friedman rank.
- In 3 of the 5 ensemble schemes considered, CDT achieves the second-best Friedman rank (tied with C4.5 in Adaboost and Bagging). It is not the worst base classifier in any ensemble.
- This base classifier does not obtain the best result for any combination ensemble/dataset. Nevertheless, for 13 pairs ensemble/dataset, it obtains the second-best performance.
- CDT achieves the best results in the *UCSD* dataset.

4) **RandomCDT**:

- It achieves the best Accuracy results by far; it obtains the best Friedman rank for all ensembles. Furthermore, it obtains a much lower average Friedman rank than other base classifiers.
- For 18 pairs out of 30 ensemble-dataset, RandomCDT achieves the best performance, and, for other 7 pairs, it obtains the second-best results.
- The most notable differences among RandomCDT and the rest of the base classifiers can be found in Rotation Forest and Random Subspace. In these two ensemble schemes, RandomCDT obtains the best results.
- Even though in DECORATE and Adaboost RandomCDT performs better than the remaining base classifiers, in these two ensemble schemes, RandomCDT achieves the worst results.
- It is convenient to remark that, for the *Polish* and *UCSD* datasets, RandomCDT always achieves the best performance. For *Australian*, it also obtains the best results in all ensembles, except for DECORATE, in which it obtains the second-best performance.

TABLE 4. Summary of the results of the Holm tests corresponding to Accuracy. In each column, the base classifier significantly outperforms the ones in the row in the ensembles indicated in the cell.

	LogR	C4.5	CDT	RCDT
LogR	-			Random Subspace
C4.5		-		
CDT			-	
RCDT				-

Table 4 summarizes the results of the Holm tests corresponding to Accuracy for the cases in which there are statistically significant differences via the Friedman test.

As can be observed, there are only statistically significant differences in Random Subspace, where RandomCDT outperforms LogR. C4.5 and CDT obtain statistically equivalent results to all base classifiers for all the ensemble schemes considered here.

2) AUC RESULTS

Table 5 presents the average AUC results obtained by each base classifier in each ensemble scheme for each dataset. It also shows the Friedman rank of each base classifier in each ensemble scheme. Similar to Table 2, the best results are noted in bold and the second-best results in italic. Table 6 summarizes the Friedman's ranks results for AUC. Specifically, it shows the Friedman rank obtained by each base classifier in each ensemble scheme, as well as the average Friedman rank for each base classifier.

From Tables 5 and 6, the following issues can be observed about each one of the base classifiers for the AUC measure:

1) **LogR**:

- This base classifier occupies the third position in the average Friedman rank.
- LogR has the worst performance with Adaboost, whereas it achieves the best results with

TABLE 5. Average AUC results obtained by each base classifier in each dataset grouped by ensemble.

Ensemble	Base	Australian	German	Japanese	Iranian	Polish	UCSD	Rank
AdaBoost	LogR	0.8951	0.7106	0.9048	0.6503	0.7841	0.8565	3.8333
	C4.5	**0.9123**	**0.7384**	*0.9148*	**0.7534**	0.8164	*0.8987*	*1.8333*
	CDT	*0.9160*	*0.7355*	**0.9193**	*0.7396*	*0.8174*	**0.8996**	**1.5**
	RCDT	0.9066	0.7248	0.9046	0.7084	**0.8299**	0.8687	2.8333
Bagging	LogR	**0.9321**	**0.7916**	**0.9348**	0.7330	0.8198	0.8841	*2.5*
	C4.5	0.9273	*0.7695*	0.9285	*0.7501*	0.8264	0.9020	3
	CDT	0.9288	0.7648	0.9290	0.7452	*0.8288*	*0.9056*	2.8333
	RCDT	*0.9319*	0.7675	*0.9322*	**0.7870**	**0.8457**	**0.9160**	**1.6667**
Random Subspace	LogR	*0.9263*	**0.7880**	*0.9285*	*0.7478*	0.8311	0.8799	*2.3333*
	C4.5	0.9226	0.7569	0.9219	0.7013	0.8212	0.9019	3.6667
	CDT	0.9246	0.7506	0.9225	0.7195	*0.8346*	*0.9090*	2.8333
	RCDT	**0.9339**	*0.7841*	**0.9344**	**0.7539**	**0.8432**	**0.9185**	**1.1667**
DECORATE	LogR	**0.9263**	**0.7937**	**0.9279**	**0.7451**	0.8100	*0.8771*	**1.6667**
	C4.5	0.9050	0.7345	0.9104	0.6820	0.8216	0.8684	3.8333
	CDT	0.9164	*0.7523*	0.9131	0.6989	*0.8324*	0.8730	2.6667
	RCDT	*0.9199*	0.7457	*0.9190*	*0.7217*	**0.8442**	**0.8972**	*1.8333*
Rotation Forest	LogR	**0.9316**	**0.7925**	**0.9349**	0.7327	0.8049	0.8841	*2.5*
	C4.5	0.9257	0.7709	0.9263	0.7331	0.8373	0.9070	3.3333
	CDT	0.9251	0.7715	0.9264	*0.7762*	*0.8428*	*0.9088*	2.6667
	RCDT	*0.9277*	*0.7883*	*0.9277*	**0.7875**	**0.8586**	**0.9211**	**1.5**

TABLE 6. Results of the Friedman's ranks in AUC for each base classifier.

Base	Adaboost	Bagging	Random Subspace	DECORATE	Rotation Forest	Average
LogR	3.8333	*2.5*	*2.3333*	**1.6667**	*2.5*	2.5666
C4.5	*1.8333*	3	3.6667	3.8333	3.3333	3.1333
CDT	**1.5**	2.8333	2.8333	2.6667	2.6667	*2.5*
RCDT	2.8333	**1.6667**	**1.1667**	*1.8333*	**1.5**	**1.8**

DECORATE. LogR is the second-best in Bagging, Random Subspace, and Rotation Forest.

- It achieves 11 of the best results out of 30 by ensemble-dataset. For 4 combinations ensemble/dataset, it obtains the second-best result.
- LogR obtains very good results for the *Australian*, *German*, and *Japanese* datasets, except in Adaboost.

2) **C4.5**:

- This base classifier obtains the worst results, except in AdaBoost, where it occupies the second position.
- For 3 pairs out of 30 ensemble-dataset, it obtains the best results, and, for 4 pairs, it achieves the second-best performance.
- There is no dataset for which C4.5 obtains good results.

3) **CDT**:

- It achieves the second-lowest average Friedman rank.
- CDT occupies the third position in all ensembles regarding Friedman's ranks, except in Adaboost, where it achieves the best results.
- It is the best method for 2 combinations ensemble/dataset, and, for 13 combinations, it obtains the second-best performance.
- *UCSD* is the dataset for which CDT achieves the best results.

4) **RandomCDT**:

- It is the clear winner base classifier for the AUC measure; it obtains the lowest average Friedman rank.
- RandomCDT achieves the best Friedman rank in 3 of 5 ensembles, the second position for DECORATE, and the third one for Adaboost. However, it is easy to check that, in general, Adaboost is the ensemble that obtains the worst AUC results.
- For 14 pairs ensemble/dataset out of 30, it obtains the best performance, and, for 9 pairs, it achieves the second-best results.
- This base classifier is always the best for the *Polish* dataset. It also obtains very good results for *Iranian* and *UCSD*, except with Adaboost.

TABLE 7. Summary of the results of the Holm tests corresponding to AUC. In each column, the base classifier significantly outperforms the ones in the row in the ensembles indicated in the cell.

	LogR	C4.5	CDT	RCDT
LogR	-	Adaboost	Adaboost	
C4.5	DECORATE	-		DECORATE, Random Subspace
CDT			-	
RCDT				-

Table 7 allows us to see a summary of the results of the Holm tests associated with AUC when there are statistically significant differences according to the Friedman test.

Taking into account Table 7, we can describe the following cases of statistical differences for each base classifier:

- **LogR**: It performs significantly worse than C4.5 and CDT in Adaboost, whereas it obtains significantly better results than C4.5 in DECORATE.
- **C4.5**: The results obtained by this base classifier are significantly better than the ones obtained by LogR in Adaboost. Nonetheless, in DECORATE, it performs significantly worse than LogR and RandomCDT. It is also significantly outperformed by RandomCDT in Random Subspace.
- **CDT**: It does not perform significantly worse than any base classifier for any ensemble, and it obtains significantly better results than LogR in Adaboost.
- **RandomCDT**: It performs significantly better than C4.5 in DECORATE and Random Subspace. As CDT, RandomCDT does not perform worse than any base classifier for any ensemble scheme.

A financial expert is interested in optimizing the predictive model for a credit scoring dataset. For this reason, we analyze, for each credit scoring dataset considered here, for which combination ensemble/base classifier the best AUC results are attained.

- **Australian**: The best result in this dataset is obtained by Random Subspace with RCDT as the base classifier, and the second-best one by Bagging with LogR.
- **German**: Rotation Forest with LogR as the base classifier gets the best result in this dataset. The second-best result in German is also achieved by Rotation Forest using RCDT.

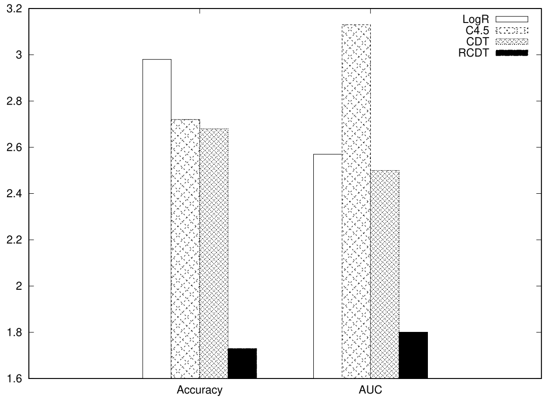

FIGURE 1. Average Friedman rank for each base classifier in Accuracy and AUC.

- **Japanese**: In this dataset, the best result is obtained by Rotation Forest using LogR and the second-best one by Bagging also using LogR.
- **Iranian**, **Polish** and **UCSD**: The best results in these datasets are got by Rotation Forest with RCDT as the base classifier, and the second-best ones by Bagging also employing RCDT.

3) SUMMARY OF THE RESULTS

The following issues summarize the results obtained in this experimental study:

- Figure 1 shows the average Friedman ranks obtained by the base classifiers. It is easy to observe that RandomCDT is the base classifier that achieves, in general, the best results in both Accuracy and AUC.
- RandomCDT obtains the best Friedman ranks for almost all ensembles, and it is the base classifier that achieves the best results by ensemble/dataset. The only ensemble where it does not obtain good results in AUC is Adaboost. Nonetheless, Adaboost is the ensemble scheme that performs worst in AUC.
- Therefore, we can state that RandomCDT is the winner base classifier among the ones considered in this work. The reason is that, as we argued in Section IV-A, the RandomCDT algorithm is quite suitable for ensembles because it encourages diversity, which is the most important issue in ensembles.
- The ensemble schemes where RandomCDT achieves the best results are Random Subspace and Rotation Forest.
- LogR obtains the worst results in Accuracy, whereas it achieves the second-best results in AUC, although, with Adaboost, it obtains the worst performance for this measure.
- The C4.5 algorithm obtains the third position in Accuracy and the worst results in AUC. We can state that it is the worst base classifier among the ones considered in this research in terms of Friedman ranks and wins ensemble-dataset. In this way, DTs based on classical Probability theory are not the most appropriate to employ in ensembles for credit scoring.
- CDT achieves the second-best results in Accuracy and AUC concerning Friedman ranks and wins ensemble-dataset. It achieves the best results for very few combinations ensemble-dataset, but CDT obtains the second-best performance for a considerable number of pairs ensemble-dataset.
- Concerning the combinations of ensemble/dataset that lead to the best results in each dataset, it must be remarked that Rotation Forest and Bagging are the ensemble schemes that achieve the best performance in almost all datasets. The base classifier that gets the best results in these ensembles is RCDT, followed by LogR.
- From a financial point of view, the important result here might be the performance of the final procedure (ensemble + base classifier) on each dataset. We can conclude that, for each dataset, our proposal, RCDT join with a type of ensemble is always the best one or the second best in results on the AUC measure. It is the first one in that measure for 4 of those 6 datasets, and the second one on the other 2 but with results very close to the best ones. For example, for the Japanese dataset, the best procedure is Rotation Forest + LogR with 0.9349 value, and the second one is Random Subspace+RCDT with 0.9344 value.

VI. CONCLUSION AND FUTURE WORK

Improving credit scoring methods is an essential issue for banks and financial institutions. Slight improvements may produce great benefits. Ensembles of classifiers are the techniques that have achieved the best results in credit scoring. In this work, we have completed previous experimental studies about the use of base classifiers in ensemble schemes for credit scoring datasets. Specifically, we have considered a CDT, an algorithm that has obtained the best results in ensembles for credit scoring and strongly depends on a hyperparameter. This algorithm takes into account the lack of precision of the information obtained from data. The proposed version, called Random Credal Decision Tree (RCDT), randomly chooses the value of the hyperparameter, among a range established as suitable in a previous study, before building the tree. As pointed out previously, different values of the hyperparameter might give rise to different trees when they are built with the same training set. In this way, RCDT encourages more diversity than CDT when both methods are used in ensembles. Remark that "the key point for the success of an ensemble scheme is that the base classifiers are not only accurate but also diverse and unstable" [44]. Thus, the RCDT method is suitable to be used as the base classifier in ensemble schemes.

An exhaustive experimental analysis has been carried out in this research with several credit scoring datasets. In such an analysis, the main ensemble schemes considered in the literature for credit scoring have been utilized. As the base classifiers, we have considered RCDT and the ones that achieved the best results in ensembles for credit scoring in previous experimental studies: LogR, C4.5, and CDT. It has been shown that the RCDT algorithm is, by far, the base classifier that performs best in ensemble schemes for credit scoring datasets.

Therefore, RCDT is an appropriate algorithm to use as the base classifier in ensembles for credit scoring since it would lead to considerable benefits for banks and financial institutions.

As future research, other ensemble schemes could be applied to credit scoring datasets, such as the one utilized in [59]. Furthermore, it would be interesting to employ dynamic selection procedures [60] for the best combinations ensemble/base classifier to improve the results in credit scoring.

REFERENCES

[1] D. J. Hand and W. E. Henley, "Statistical classification methods in consumer credit scoring: A review," *J. Roy. Stat. Soc. A, Statist. Soc.*, vol. 160, no. 3, pp. 523–541, Sep. 1997.

[2] X. Dastile, T. Celik, and M. Potsane, "Statistical and machine learning models in credit scoring: A systematic literature survey," *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106263.

[3] H. Ince and B. Aktan, "A comparison of data mining techniques for credit scoring in banking: A managerial perspective," *J. Bus. Econ. Manage.*, vol. 10, no. 3, pp. 233–240, Sep. 2009.

[4] D. Martens, T. Van Gestel, M. De Backer, R. Haesen, J. Vanthienen, and B. Baesens, "Credit rating prediction using ant colony optimization," *J. Oper. Res. Soc.*, vol. 61, no. 4, pp. 561–573, Apr. 2010.

[5] B.-W. Chi and C.-C. Hsu, "A hybrid approach to integrate genetic algorithm into dual scoring model in enhancing the performance of credit scoring model," *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2650–2661, Feb. 2012.

[6] T. Harris, "Credit scoring using the clustered support vector machine," *Expert Syst. Appl.*, vol. 42, no. 2, pp. 741–750, Feb. 2015.

[7] A. B. Hens and M. K. Tiwari, "Computational time reduction for credit scoring: An integrated approach based on support vector machine and stratified sampling method," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6774–6781, Jun. 2012.

[8] J. M. Tomczak and M. Zięba, "Classification restricted Boltzmann machine for comprehensible credit scoring model," *Expert Syst. Appl.*, vol. 42, no. 4, pp. 1789–1796, Mar. 2015.

[9] K. Bijak and C. Lyn Thomas, "Does segmentation always improve model performance in credit scoring? *Expert Syst. Appl.*, vol. 39, no. 3, pp. 2433–2442, 2012.

[10] P. Makowski, "Credit scoring branches out," *Credit World*, vol. 74, no. 2, pp. 30–37, 1985.

[11] B. W. Yap, S. H. Ong, and N. H. M. Husain, "Using data mining to improve assessment of credit worthiness via credit scoring models," *Expert Syst. Appl.*, vol. 38, no. 10, pp. 13274–13283, Sep. 2011.

[12] D. J. W. E. Hand Henley, "A *k*-nearest-neighbour classifier for assessing consumer credit risk," *J. Roy. Stat. Soc. D, Statistician*, vol. 45, no. 1, pp. 77–95, 1996.

[13] Z. Zhang, K. Niu, and Y. Liu, "A deep learning based online credit scoring model for P2P lending," *IEEE Access*, vol. 8, pp. 177307–177317, 2020.

[14] X. Dastile and T. Celik, "Making deep learning-based predictions for credit scoring explainable," *IEEE Access*, vol. 9, pp. 50426–50440, 2021.

[15] Z. Zhao, S. Xu, B. H. Kang, M. M. J. Kabir, Y. Liu, and R. Wasinger, "Investigation and improvement of multi-layer perceptron neural networks for credit scoring," *Expert Syst. Appl.*, vol. 42, no. 7, pp. 3508–3516, May 2015.

[16] C. Wang, D. Han, Q. Liu, and S. Luo, "A deep learning approach for credit scoring of peer-to-peer lending using attention mechanism LSTM," *IEEE Access*, vol. 7, pp. 2161–2168, 2019.

[17] W.-W. Wu, "Improving classification accuracy and causal knowledge for better credit decisions," *Int. J. Neural Syst.*, vol. 21, no. 4, pp. 297–309, Aug. 2011.

[18] K. Masmoudi, L. Abid, and A. Masmoudi, "Credit risk modeling using Bayesian network with a latent variable," *Expert Syst. Appl.*, vol. 127, pp. 157–166, Aug. 2019.

[19] B. Anderson, "Using Bayesian networks to perform reject inference," *Expert Syst. Appl.*, vol. 137, pp. 349–356, Dec. 2019.

[20] S. Guo, H. He, and X. Huang, "A multi-stage self-adaptive classifier ensemble model with application in credit scoring," *IEEE Access*, vol. 7, pp. 78549–78559, 2019.

[21] A. I. Marqués, V. García, and J. S. Sánchez, "Exploring the behaviour of base classifiers in credit scoring ensembles," *Expert Syst. Appl.*, vol. 39, no. 11, pp. 10244–10250, Sep. 2012.

[22] W. Yotsawat, P. Wattuya, and A. Srivihok, "A novel method for credit scoring based on cost-sensitive neural network ensemble," *IEEE Access*, vol. 9, pp. 78521–78537, 2021.

[23] S. Wei, D. Yang, W. Zhang, and S. Zhang, "A novel noise-adapted two-layer ensemble model for credit scoring based on backflow learning," *IEEE Access*, vol. 7, pp. 99217–99230, 2019.

[24] X. Chen, S. Li, X. Xu, F. Meng, and W. Cao, "A novel GSCI-based ensemble approach for credit scoring," *IEEE Access*, vol. 8, pp. 222449–222465, 2020.

[25] J. Abellán and J. G. Castellano, "Improving the naive Bayes classifier via a quick variable selection method using maximum of entropy," *Entropy*, vol. 19, no. 6, p. 247, May 2017.

[26] C. J. Mantas and J. Abellán, "Analysis and extension of decision trees based on imprecise probabilities: Application on noisy data," *Expert Syst. Appl.*, vol. 41, no. 5, pp. 2514–2525, Apr. 2014.

[27] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[28] J. Abellán and J. G. Castellano, "A comparative study on base classifiers in ensemble methods for credit scoring," *Expert Syst. Appl.*, vol. 73, pp. 1–10, May 2017.

[29] J. Abellán and S. Moral, "Building classification trees using the total uncertainty criterion," *Int. J. Intell. Syst.*, vol. 18, no. 12, pp. 1215–1225, Dec. 2003.

[30] J. Abellán and A. R. Masegosa, "An experimental study about simple decision trees for bagging ensemble on datasets with classification noise," in *Symbolic and Quantitative Approaches to Reasoning With Uncertainty* (Lecture Notes in Computer Science), vol. 5590. Berlin, Germany: Springer, 2009, pp. 446–456.

[31] C. J. Mantas and J. Abellán, "Credal-C4.5: Decision tree based on imprecise probabilities to classify noisy data," *Expert Syst. Appl.*, vol. 41, no. 10, pp. 4625–4637, Aug. 2014.

[32] J. Abellán, "Ensembles of decision trees based on imprecise probabilities and uncertainty measures," *Inf. Fusion*, vol. 14, no. 4, pp. 423–430, Oct. 2013.

[33] J. Abellán and A. R. Masegosa, "Bagging schemes on the presence of class noise in classification," *Expert Syst. Appl.*, vol. 39, no. 8, pp. 6827–6837, Jun. 2012.

[34] J. Abellán and C. J. Mantas, "Improving experimental studies about ensembles of classifiers for bankruptcy prediction and credit scoring," *Expert Syst. Appl.*, vol. 41, no. 8, pp. 3825–3830, Jun. 2014.

[35] C. J. Mantas, J. Abellán, and J. G. Castellano, "Analysis of credal-C4.5 for classification in noisy domains," *Expert Syst. Appl.*, vol. 61, pp. 314–326, Nov. 2016.

[36] J. Abellán, C. J. Mantas, J. G. Castellano, and S. Moral-García, "Increasing diversity in random forest learning algorithm via imprecise probabilities," *Expert Syst. Appl.*, vol. 97, pp. 228–243, May 2018.

[37] G. T. Dietterich, "Ensemble methods in machine learning," in *Proc. 1st Int. Workshop Multiple Classifier Syst. (MCS)*, London, U.K. Berlin, Germany: Springer-Verlag, 2000, pp. 1–15.

[38] M. Kearns, "Thoughts on hypothesis boosting," Mach. Learn. Class Project, Tech. Rep., 1988.

[39] Y. Freund and E. R. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn. (ICML)*, L. Saitta, Ed. San Mateo, CA, USA: Morgan Kaufmann, 1996, pp. 148–156.

[40] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[41] J. Abellán, G. López, L. Garach, and J. G. Castellano, "Extraction of decision rules via imprecise probabilities," *Int. J. Gen. Syst.*, vol. 46, no. 4, pp. 313–331, May 2017.

[42] P. Melville and R. J. Mooney, "Creating diversity in ensembles using artificial data," *Inf. Fusion*, vol. 6, no. 1, pp. 99–111, Mar. 2005.

[43] J. J. Rodriguez, L. I. Kuncheva, and C. J. Alonso, "Rotation forest: A new classifier ensemble method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1619–1630, Oct. 2006.

[44] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[45] J. R. Quinlan, "Induction of decision trees," *Mach. Learn.*, vol. 1, no. 1, pp. 81–106, Mar. 1986.

[46] J. R. Quinlan, *C4.5: Programs for Machine Learning*. San Francisco, CA, USA: Morgan Kaufmann, 1993.

[47] C. E. Shannon, "A mathematical theory of communication," *Bell Syst. Tech. J.*, vol. 27, no. 3, pp. 379–423, Jul. 1948.

[48] P. Walley, "Inferences from multinomial data: Learning about a bag of marbles," *J. Roy. Stat. Soc., B, Methodol.*, vol. 58, no. 1, pp. 3–57, 1996.

[49] J. Abellán, "Uncertainty measures on probability intervals from the imprecise Dirichlet model," *Int. J. Gen. Syst.*, vol. 35, no. 5, pp. 509–528, Oct. 2006.

[50] E. T. Jaynes, "On the rationale of maximum-entropy methods," *Proc. IEEE*, vol. 70, no. 9, pp. 939–952, Sep. 1982.

[51] J. G. Klir, *Uncertainty and Information: Foundations of Generalized Information Theory*. Hoboken, NJ, USA: Wiley, 2005.

[52] M. Lichman, "UCI machine learning repository," School Inf. Comput. Sci., Univ. California, Irvine, Irvine, CA, USA, Tech. Rep., 2013.

[53] 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 *Proc. 3rd CRC Credit Scoring Conf.*, Edinburgh, U.K., 2007, p. 25.

[54] W. Pietruszkiewicz, "Dynamical systems and nonlinear Kalman filtering applied in classification," in *Proc. 7th IEEE Int. Conf. Cybernetic Intell. Syst. (CIS)*, Sep. 2008, pp. 1–6.

[55] H. I. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques* (Morgan Kaufmann Series in Data Management Systems), 2nd ed. San Francisco, CA, USA: Morgan Kaufmann, 2005.

[56] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Dec. 2006.

[57] M. Friedman, "A comparison of alternative tests of significance for the problem of *m* rankings," *Ann. Math. Statist.*, vol. 11, no. 1, pp. 86–92, Mar. 1940.

[58] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandin. J. Statist.*, vol. 6, no. 2, pp. 65–70, 1979.

[59] M. Zięba, S. K. Tomczak, and J. M. Tomczak, "Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction," *Expert Syst. Appl.*, vol. 58, pp. 93–101, Oct. 2016.

[60] L. M. 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 Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113351.

SERAFÍN MORAL-GARCÍA received the dual master's degree in informatics and mathematics and the Ph.D. degree in information and communication technologies from the University of Granada, Spain, in 2016, 2018, and December 2022, respectively.

He is currently with the University of Granada. He has published several articles in different journals with an impact index. His research interests include imprecise probabilities and applications in the data mining area, especially on classic/imprecise/multilabel classification methods.

JOAQUÍN ABELLÁN received the Ph.D. degree in mathematics science from the University of Granada, Spain, in 2003.

Currently, he is a Full Professor with the University of Granada. His current research interests include representation of the information, principally via imprecise probabilities and its quantification, and also general applications in the data mining area related to the information theory. He has been the principal research of some national and regional research projects.