IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 14, 2022, accepted April 15, 2022, date of publication April 22, 2022, date of current version April 28, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3169498

Applying Efficient Selection Techniques of Unlabeled Instances for Wrapper-Based Semi-Supervised Methods

CEPHAS A. S. BARRETO[1], ARTHUR COSTA GORGÔNIO[1], JOÃO C. XAVIER-JÚNIOR[2], AND ANNE MAGÁLY DE PAULA CANUTO[1], (Member, IEEE)

[1]Department of Informatics and Applicated Mathematics (DIMAp), Federal University of Rio Grande do Norte (UFRN), Natal 59078-970, Brazil
[2]Digital Metropolis Institute (IMD), Federal University of Rio Grande do Norte (UFRN), Natal 59078-970, Brazil

Corresponding author: Cephas A. S. Barreto (cephasax@gmail.com)

This work was supported in part by the Federal University of Rio Grande do Norte; and in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil, under Finance Code 001.

ABSTRACT Semi-supervised learning (SSL) is a machine learning approach that integrates supervised and unsupervised learning mechanisms. This integration may be done in different ways and one possibility is to use a wrapper-based strategy. The main aim of a wrapper-based strategy is to use a small number of labelled instances to create a learning model. Then, this created model is used in a labelling process, where some unlabelled instances are labelled, and consequently, these instances are incorporated into the labelled set. One important aspect of a wrapper-based SSL method is the selection of unlabelled instances to be labelled in the labelling process. In other words, an efficient selection process plays an important role in the design of a wrapper-based SSL method since it can lead to an efficient labelling process, and in turn, the creation of efficient learning models. In this paper, we propose the use of three selection methods that can be applied to wrapper-based SSL methods. The main idea is to use two different selection criteria, prediction confidence or classification agreement with a distance metric, to perform an efficient selection of the unlabelled instances. In order to assess the feasibility of the proposed approach, the selection methods are applied in two well-known wrapper-based SSL methods, which are: Self-training and Co-training. Additionally, an empirical analysis will be conducted in which we compare the standard Self-training and Co-training methods against the proposed versions of these two SSL methods over 35 classification datasets.

INDEX TERMS Artificial intelligence, machine learning, semi-supervised learning, self-training semi-supervised method, co-training semi-supervised method.

I. INTRODUCTION

In the last decades, Machine Learning (ML) techniques have gained considerable relevance in many real-world problems since they offer a fantastically powerful framework for solving complex systems in an efficient way [1]. An ML technique is capable of creating a hypothesis (learning model) or function capable of solving the problem to be addressed based on past experiences. In these techniques, a learning model is created in a training phase and assessed in a testing phase [2].

In relation to the degree of supervision used during the training phase, ML techniques can be divided into three categories: supervised, unsupervised and semi-supervised [3], [4]. In these three types, ML algorithms learn from past experience and from the implicit knowledge present in existing data. Nevertheless, the main difference among these three categories is the fact that the data which these techniques use have information that may or may not be labelled. The supervised learning (classification system), for instance, uses only labelled data. In this category, an instance of a given problem is analysed by a learning model, aiming to define a label for this instance (class label) [5]. The efficiency of a classification system is usually related to the knowledge distribution which is spread among data points. In addition, it may need a large number of labelled instances to create an efficient learning model. Nonetheless, the amount of labelled data is usually limited in several classification problems.

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

In fact, for some problems, it is expensive or difficult to manually label instances [4], [6].

In order to smooth out the problems raised by the limited amount of labelled data, the semi-Supervised Learning (SSL) category has been proposed [6]–[9]. With the development of different SSL methods, these methods were grouped according to their characteristics [4], [6], [10], [11]. In [11], for instance, the SSL methods were divided into inductive and transductive methods. Among the inductive methods, a wrapper-based SSL method trains a classification algorithm (model or classifier) with a small amount of labelled data and a large amount of unlabelled data. The main aim of these methods is to iteratively create a learning model from a labelled set, select and label new instances (unlabelled) to be included into the labelled set. Several methods have been proposed in the literature and, among them, we can mention Self-training [12] and Co-training [13]. In this paper, the proposed selection procedure is applied to wrapper-based SSL methods.

In an SSL method, it may occur that wrongly classified instances are added to the labelled set, deteriorating the system performance. In other words, the automatic labelling process (selection and label) of unlabelled instances plays an important role in the design of SSL methods. In this sense, the use of an efficient selection process can produce efficient SSL methods.

In order to increase the robustness of SSL methods, this paper proposes a selection approach for choosing unlabelled instances. In this proposed approach, a selection criterion (prediction confidence or classification agreement) is combined with a distance metric, in an approach called Distance-weighted Selection (DwS) criterion. The main aim of this proposed selection approach is to perform an efficient selection of the unlabelled instances and, in this sense, to lead to a robust labelling process. Additionally, in order to evaluate the feasibility of the proposed approach, an empirical analysis will be performed. This analysis will compare the performance of six versions of each SSL method (Co-training and Self-training). Moreover, 35 classification datasets will be used for evaluating the performance of all versions of the aforementioned methods.

In [14], a Distance-Weighted Selection was proposed that combines prediction confidence and distance metric. The obtained results were promising, showing that the using of a selection criterion (prediction confidence) combined with distance metric led to a more efficient selection procedure. In this sense, a more exploratory analysis needs to be done and it will be performed in this paper. Therefore, this paper is an extension of the work proposed in [14], and its main contributions are:

- The proposal of a new Distance-Weighted Selection, which combines classification agreement and a distance metric;
- The improvement of the empirical analysis, including more selection criteria and increasing the number of datasets from 20 to 35;
- The inclusion of a comparative analysis, comparing the performance of the proposed method with existing SSL methods.

The Distance-Weighted Selection (DwS) was initially proposed in [14], in which prediction confidence and distance metric were used as parameters in a DwS method. Additionally, in [15], a selection criterion using only classification agreement was proposed. This paper extends the DwS investigation, using classification agreement or prediction confidence combined with distance, in a DwS approach. In this sense, we will have two DwS versions in this paper, DwS-C that uses prediction confidence (originally proposed in [14]) and DwS-A that uses classification agreement (originally proposed in [15]), proposed in this paper.

This paper is divided into seven sections and organised as follows. Section 2 describes the main concepts related to semi-supervised learning methods while Section 3 presents some important studies in the SSL literature, focusing on Co-training and Self-training methods. Section 4 presents the proposed approach, describing the main differences to the corresponding standard methods. The experimental methodology is presented in Section 5 while the computational results are illustrated in Section 6. Finally, Section 7 presents the main conclusions and some directions for future work.

II. BACKGROUND

This Section presents the SSL methods that will be used as basis in this paper, Co-training and Self-training (Section II-A). In addition, two extended versions of these methods using an ensemble-based selection criterion are also described (Section II-B).

A. STANDARD SSL METHODS

The next subsections will present two semi-supervised methods (Self-Training and Co-Training). In these SSL methods, the following acronyms and names are used: $\boldsymbol{L}$: labelled set; $\boldsymbol{i}$: instance; $\boldsymbol{U}$: unlabelled set; and $\boldsymbol{C}$: classifier.

1) SELF-TRAINING

The Self-training SSL technique, which was originally proposed in [12], is a simple and an efficient way to label unlabelled instances. Algorithm 1 presents the main steps of this method.

Algorithm 1 Self-Training

```
1 while U is not empty do
2     train C with L;
3     label U using C;
4     select the best labelled-instances of U and join them
        to L;
5 end
```

In this method, the first step is to train a classifier (model learning) using the available set of labelled instances (line 2).

Once the classifier is trained, it labels the whole set of unlabelled instances (line 3). Then, the best labelled instances are selected to be added to the labelling set (line 4). This selection is based on the confidence prediction delivered by the unlabelled instances. The loop containing these three steps (train, label and select) stops when the unlabelled set (U) is empty.

2) CO-TRAINING

Co-training [13] is a semi-supervised method that iteratively applies two different views of an instance in the labelling process. In order to do this, it trains two classification algorithms, each one with a different attribute subset. Then, it uses the predictions of one classification algorithm to choose the unlabelled instances that will be added to the labelled set of the other algorithm. The main steps of the Co-training method are described in Algorithm 2. The Co-training functioning is similar to the Self-training, but it uses two classification algorithms instead of only one.

Algorithm 2 Co-Training Algorithm

1 create L_1 and L_2 with a vertical split in L;
2 create U_1 and U_2 with a vertical split in U;
3 **while** U_1 *and* U_2 *are not empty* **do**
4 train C_1 with L_1 and C_2 with L_2;
5 apply C_1 in U_1 and C_2 in U_2;
6 select the best instances from U_1 and include them to L_2;
7 select the best instances from U_2 and include them to L_1;
8 **end**

In the first step of the Co-training algorithm, the input attributes are divided into two sets (one for each classification algorithm). In this sense, both instance sets, labelled (L) and unlabelled (U), will have two different views of the instances, L_1 and L_2 (line 1) as well as U_1 and U_2 (line 2). The next step is to train two classification algorithms, C_1 and C_2, using L_1 and L_2, respectively (line 4). Once the classification algorithms are trained, the unlabelled instances, U_1 and U_2, are labelled by their corresponding classification algorithms, C_1 and C_2 (line 5). In the last step, the most confident instances from U_1 and U_2 are incorporated to L_2 and L_1 (lines 6 and 7), respectively. This is an iterative process and continues until both unlabelled sets are empty.

The main objective of the Co-training method is to promote cooperation between both classification methods by crossing the acquired knowledge [13]. In this method, the C_2 algorithm selects the most confident instances to be incorporated in L_1, and vice-versa. In doing this, it aims at cooperating with the quality of the other classification algorithm (C_1). With this crossing distribution, it is expected that these added instances are not biased and it will improve the classifier performance.

In summary, it is possible to observe that Self-training (Algorithm 1) and Co-training (Algorithm 2) have similar labelling processes. However, they differ in two different ways. The first one is the attribute distribution performed by the Co-training method. It is important to emphasise that the attribute selection needs to assure that an attribute is put in only one subset (null intersection) and all attributes have to be placed in the subsets (total union). These two conditions will lead to two different subsets and it is expected to improve the classification efficiency of the Co-training method. The second difference is related to the addition of the unlabelled instances. In Self-training, the unlabelled instances are selected and added to the unlabelled set of one classifier. On the other hand, in the Co-training method, one classification algorithm labels an instance and it is added to the unlabelled set of the other classifier.

B. ENSEMBLE-BASED SELECTION CRITERION

Classifier ensembles are classification structures composed of a set of base classifiers. These systems have a two-layer structure in which all base classifiers receive input data and predict a class for a new instance in the first layer. These predictions are sent to a combination module in the second layer, which combines all received predictions into a single predicted class for each instance (e.g. via majority voting). The output combination performed by an ensemble usually surpasses the performance of individual classification algorithms [16], [17]. The ensemble structure makes this system an interesting alternative for the selection of unlabelled instances in semi-supervised methods. By using ensemble as part of a selection technique, it allows the reduction of errors provided by the selection based on only one classifier or metric.

The Ensemble-based Automatic Labelling (EbAL) is an approach to select instances from unlabelled set more efficiently, and also it uses an ensemble-based selection criterion in the labelling process [15]. This approach uses classification agreement from a pool of classifiers (ensemble) for selecting and labelling instances in SSL methods. In fact, the EbAL approach can have two main versions. In the first version, named EbAL-v1, the pool of classifiers is used for selecting instances with higher classification agreement from the unlabelled set. In the second version (EbAL-v2), besides the use for selecting instances, it also uses the ensemble output for labelling instances. Although both versions of EbAL approach were originally applied to the Self-training method, it can be applied to any wrapper-based SSL method.

1) SELF-TRAINING WITH EbAL

In this extended work, both EbAL versions for Self-training were originally presented in [15] and they will be named as Self-training with EbAL - version 1 (St-EbAL(v1)) and Self-training with EbAL - version 2 (St-EbAL(v2)). Algorithms (3 and 4) will describe both versions, defining the main steps of them. For simplicity reason, both algorithms use the following names and acronyms: $\boldsymbol{i}$: instance; $\boldsymbol{L}$: labelled set; $\boldsymbol{U}$: unlabelled set; $\boldsymbol{C}$: main classifier; $\boldsymbol{PC}$: pool of classifiers; $\boldsymbol{n}$: pool size; $\boldsymbol{A}$: classification agreement; and $\boldsymbol{t}$: threshold for classification agreement (in percentage).

Algorithm 3 Self-Training With EbAL - Version 1 (St-EbAL(v1))

```
while U is not empty or no instances were included to L do
    train PC with L
    train C with L
    for i in U do
        Using PC assign n pseudo-labels to i
        Compute A of i using the pseudo-labels
        if A of i ≥ t then
            remove i of U
            assign class label to i using C
            add i to L
        end
    end
end
```

The main difference between the standard Self-training method and St-EbAL(v1) is related to the selection criterion. While Self-training uses the confidence prediction as selection criterion, St-EbAL(v1) applies the classification agreement of a PC (agreement defined by the set of classifiers of an ensemble) to select the most prominent unlabelled instances.

Algorithm 4 Self-Training With EbAL - Version 2 (St-EbAL(v2))

```
while U is not empty or No instances were added to L do
    train PC with L
    for i in U do
        Using PC assign n pseudo-labels to i
        Compute A of i using the pseudo-labels
        if A of i ≥ t then
            remove i of U
            assign the class with the highest A to i
            add i to L
        end
    end
end
train C with L
```

As previously mentioned, the main difference between St-EbAL(v2) and St-EbAL(v1) is that the St-EbAL(v2) labelling step is also performed by a classifier ensemble combined by a majority voting method. In other words, in St-EbAL(v2), an ensemble is used to select and label unlabelled instances.

2) CO-TRAINING WITH EbAL

The Co-training versions using EbAL approach follow the same flow presented in the standard Co-training (Ct-std), but using two Pools of Classifiers (PCs) and classification agreement in both PCs. In this sense, Co-training has also two EbAL versions to be introduced. In the first version, named Co-training with Ensemble-based Automatic Labelling - version 1 (Ct-EbAL(v1)), each pool of classifiers (one at each side) is responsible for selecting the best instances according to classification agreement; In the second one, named Co-training with Ensemble-based Automatic Labelling - version 2 (Ct-EbAL(v2)), the pools of classifiers are responsible for the selection and also the labelling of instances. It is important to emphasise that the use of EbAL in the Self-training method was originally presented in [15]. However, the combination of EbAL for the Co-training method is new, and also it represents one of the selection methods proposed in this paper.

C. SELECTION CRITERIA OF SELF-TRAINING AND CO-TRAINING METHODS

As it could be observed from the previous sections, prediction confidence is used as a selection criterion in the standard versions of both SSL methods (one classifier for St-std or two classifiers for Ct-std). This criterion describes the confidence level in which a classification algorithm assigns an instance to a class. Two main aspects can strongly affect this confidence: 1) the class distribution of the training set; and 2) the characteristics of the used classification algorithm. An alternative approach that can be used in the selection of unlabelled instances is by calculating the similarity between instances that can be performed by a distance metric. Hence, the selection the unlabelled instances is based on the similarity information of labelled and unlabelled instances.

As presented in Algorithms 3 and 4, both approaches use the classification agreement from a pool of classifiers as selection criterion and/or labelling instances. This criterion brings to wrapper-based SSL methods the benefits of using classifier ensemble in the selection process. As an example of one possible benefit, the diversity promoted by different classifiers within the pool can lead to more accurate precision. For both EbAL versions, classification agreement has similar role as confidence in the standard versions, as the instances with classification agreement equals or greater than a threshold (t) are selected to be included in the labelled set.

One drawback of a semi-supervised method is the effect of including a wrongly classified instance in the labelled set. This error is usually carried out throughout the following iterations (snowball case), resulting in weak models (low accuracy). Because of this, efficient approaches for the selection process must be used, aiming to smooth out the selection errors that may occur in the processing of a wrapper-based semi-supervised method.

III. RELATED WORK

Semi-supervised learning (SSL) is, as previously mentioned, a combination of supervised and unsupervised learning categories. Since SSL has emerged as a robust attempt to handle problems with a small number of labelled instances, several real-world problems can be efficiently solved using SSL. Therefore, many application domains can benefit from the use of SSL methods, such as: audio and acoustics [18]; bio-informatics [19]; image processing and classification

[20]–[22]; text classification and natural language processing [23], [24]; industry [25], transport [26], among others.

The functioning of SSL methods are grounded on some assumptions (e.g. smoothness assumption, low-density assumption, manifold assumption and cluster assumption) [11]. These assumptions are conditions that make possible the use of unlabelled data along with labelled data to improve the accuracy of supervised methods [4]. As already mentioned, the literature has divided the SSL methods into groups and these groups are divided by the way an SSL method uses labelled and unlabelled data. In [11], for instance, a taxonomy is presented that divides the SSL methods into **Transductive methods** (e.g. graph-based methods, as Mincut [27], Gaussian Random Fields (GRF) [28], and Learning with Local and Global Consistency (LLGC) [29]); and **Inductive methods - wrapper** (e.g. Self-training [12], Co-training [13], Boosting [30], Tri-training [9]).

Still according to the taxonomy presented in [11], the inductive methods are further divided into three sub-classes, which are: wrapper methods, unsupervised pre-processing methods and intrinsically semi-supervised methods. This work presents a selection approach of unlabelled instances to be used in wrapper-based SSL methods due to the that fact that the predictions of a base classifier is used to generate additional labelled data. Additionally, Self-training and Co-training are two well-known semi-supervised methods that belong to the wrapper-based SSL sub-class. Therefore, the selection approaches proposed in this paper will be assessed in these two wrapper-based SSL methods. Therefore, for simplicity reason, hereinafter, the terms SSL methods and wrapper-based SSL methods will be used interchangeably.

The majority of the studies related to Self-training and Co-training present different ways to select and label the unlabelled instances [21], [31]–[35]. In this sense, we divide the SSL methods based on the used selection criterion and they are classified into two groups, which are: confidence-based methods and distance-based methods. To the best of the authors knowledge, there is no study using an agreement-based selection (only the previous study of the authors).

A. CONFIDENCE-BASED METHODS

As an example of the use of confidence-based selection, the studies presented in [21], [26], [31]–[33], [36] used prediction confidence as the unique criterion to select unlabelled instances. In [31], for instance, the authors combined a Decision Tree algorithm with a threshold value as a way to define the number of labelled instances to be selected in a Self-training method.

An interesting selection strategy was presented in [36], in which the authors proposed a new semi-supervised learning algorithm that dynamically selects the most promising learners for a classification problem from a pool of classifiers based on a Self-training. They assume a strategy based on Darwinism, in which a classifier that labels few instances is not useful for the pool. In other words, they generate a pool of classifiers and repeat the classification process, dropping the worst classifier until only the best classifier for the problem remains.

A domain-based study was presented in [26], in which a second-order inference methodology was proposed to take advantage of the Self-training method to predict the missing destinations for urban transportation systems. The proposed method uses two phases: a base learner to predict the missing destinations based on the statistics of a selected similarity-based ''training set'', and the selection of data with high prediction confidence to update the training set.

In [37] the authors propose the CPSSDS, which uses the Self-training in the data stream context. In that approach, the authors combine inductive conformal prediction of base classifiers with the Self-training algorithm to determine the most reliable unlabelled samples. These selected instances were used to update the model during the evaluation phase. The results point out that their approach was proposed for improving the classification performance of the semi-supervised Self-training approach in non-stationary environments.

In another study, [32], the authors proposed a Self-training extension based on density peaks of data, being an approach in which the concept of differential evolution was employed. It is used as a way to both discover the data structure for a better classifier training and to improve the placement of labelled data. In addition to this, the authors carried out an empirical analysis, comparing the accuracy results obtained to the standard Self-training version.

Still in the context of confidence-based selection, there are some attempts to incorporate a confidence-based selection criterion in the functioning of a Co-training method. For instance, in [33], the authors proposed a Co-training method with confidence-based selection for sentiment classification of Massive Open On-line Course (MOOC) posts. Additionally, this method used a mixed loss function computed over labelled and unlabelled data. According to the authors, the results were better than those obtained by methods trained with massive labelled data.

Another study with Co-training was presented in [24]. In the cited paper, a multi-Co-training method was proposed and its objective is to improve semi-supervised document classification performance. This work used three document representation techniques to increase the diversity of the features. The selection criterion of this method is based on the confidence of a instance. Only the instances with the highest confidence are added at each iteration.

In [21], several deep neural networks were combined with the general concept of co-training as a way to create a deep multi-view of image datasets. A confidence-based selection criterion was also used to select the unlabelled instances. According to the authors, the proposed method yielded good results, when compared to state-of-the-art SSL methods.

B. DISTANCE-BASED METHODS

Semi-supervised methods with distance-based selection are the ones that apply a distance metric (or a similarity metric) as

basis to select the instances of the unlabelled set. The studies presented in [34], [35], [38]–[40] are examples of distance-based selection approaches for Self-training and Co-training methods. In [38], for instance, a distance-based Self-training is proposed. The main aim was to select the images of the unlabelled set that were more similar to the labelled instances in order to improve the video classification performance.

Another distance-based Self-training study was proposed in [34], in which the *k*-NN classification algorithm k-NN was used as a noise filter, allowing the aggregation only of the nearest instances at the selection step. In [39], the authors proposed a new SSL method based on self-training algorithm and named as RDE_self-training. The main difference between the original and the proposed approaches is the adjustment of mislabelled instances (labelling step). This adjustment is performed over mislabelled samples according to a nearest neighbour voting rule. They use an SVM classifier to perform the labelling process, achieving the best accuracy in 15 out of 18 cases.

A different example of distance-based selection for semi-supervised methods was proposed in [41]. In this paper, the authors use the neighbourhood close to the decision boundary and assign the label of the selected instances using an agreement between classifier and neighbourhood. Based on the Apollonius circle approach, that proposal samples the instances from unlabelled data and determines the label of each instance in an iterative process. The results compare their approach with supervised SVM, standard Self-training, and other state-of-art methods and state that their proposal achieves better results and superiority when few labelled instances.

For Co-training, in [35], the authors proposed a distance-based selection criterion for Co-training, using the K-means clustering technique. This technique was applied to select the closest instances that are capable of representing a cluster. This concept was then employed to two Co-training steps, data splitting and subsets crossing over. This cited work also compared their approach with some state-of-the-art SSL methods, and, according to the authors, it outperformed them in all analysed metrics, such as unsupervised accuracy, normalised mutual information and purity.

The authors in [40] proposed a variation of the Self-Organizing Map (SOM) using the Co-training algorithm and the Mahalanobis distance metric. In this approach, the Co-training algorithm is used to create two views of the same dataset. Then, a clustering process is performed until convergence. The authors compared their proposal with three other SOM-based methods, obtaining their proposal the best results in both multi and single views.

Besides the previously mentioned studies, there are other SSL studies that seek to strengthen the robustness of SSL approaches. The improvements can be reached either by switching the standard selection process [42] or applying extra apparatus as a way to improve the selection step [32], [43], [44]. In [42], for instance, an adjustable confidence parameter was defined as a threshold to select unlabelled instances. This parameter varies throughout the iterative process and, according to the authors, it improved the performance of a Self-training method.

An improved selection approach was also proposed in [45]. The authors proposed a pseudo-label aware robust sample selection, a hybrid selection that combines the best from all three most used selection strategies for large-scale datasets (sample selection approaches, noise-robust loss functions, and label correction methods) in a framework to achieve robustness to noisy labels. The Self-training iterative process performs the pseudo-labelling step to filter the ambiguous instances. The results show that their proposal obtains significant gains over state-of-the-art methods; in some cases, the gains are 27% in the test dataset.

An example of a more robust Co-training was presented in [46], which proposed an approach based on entropy and multi-criteria. That proposal uses two views with the same amount of information by entropy, then a clustering criterion and confidence criterion are adopted to select unlabelled data from both views. The results show that the multi-criteria approach achieves good classification effectiveness when compared with state-of-art.

In summary, it is evident that the use of confidence-based or distance-based selection can be considered robust selection approaches for wrapper-based SSL methods. Nevertheless, there is still possible ways to improve the selection process and the combination of these approaches can potentially improve the performance of this class of SSL method. Thus, this extended work proposes selection approaches for wrapper-based SSL methods, distinctively Self-training and Co-training methods. In this proposal, the confidence-based selection criterion (or an agreement-based selection) is combined with a distance metric (distance-based selection) as a way to define a general parameter for the selection of unlabelled instances, which has been named as Distance-Weighted Selection (DwS).

IV. THE PROPOSED APPROACH

As mentioned previously, this paper proposes a different selection approach for wrapper-based SSL methods. Our approach uses confidence-based or agreement-based criterion in the selection step. This criterion is then combined with a distance metric in order to establish the overall parameter to select unlabelled instances, in an approach called Distance-weighted Selection (DwS). In [15], a DwS was proposed that combined a confidence-based criterion with a distance metric. In this paper, we proposed a new DwS that combines an agreement-based selection criterion with a distance metric.

The best way to combine a distance metric with any additional information (selection criterion) is through a weighted sum. In Equation 1 formally defines how to compute the Distance-weighted Selection for an instance i (DwS_i).

$$\mathrm{DwS}_i = \max(\forall_{j \in J}\ \mathrm{DwS}_{ij}) \tag{1}$$

where:

$$\mathrm{DwS}_{ij} = \mathrm{W}_{ij} \times \frac{1}{d_{ij}} \tag{2}$$

where:

- DwS_{ij} is the distance-weighted selection of instance i to class j;
- W_{ij} defines the additional information (selection criterion) for an instance i to the j-th class. In this paper, we will use prediction confidence and classification agreement as the selection criterion, W, in this equation;
- J represents the set of classes of a problem;
- d_{ij} represents the distance between the i-th instance and the centroid of the j-th class.

As it can be observed in Equation 1, the Distance-weighted Selection of an instance i is defined by the maximum value among values of all classes. For each class, the Distance-weighted Selection is defined by multiplying the distance of this instance to the centroid of a class with its corresponding weight. In this paper, we will use the Euclidean distance as d_{ij} and it calculates the distance between the i-th instance and the centroid of the j-th class. Additionally, we will use two selection criteria as weight (W_{ij}), prediction confidence and classification agreement, leading to two DwS versions, DwS-C and DwS-A. The next subsections will describe the use of these selection approaches in both Self-training and Co-training SSL methods.

A. SELF-TRAINING WITH DwS-C

Algorithm 5 shows the functioning of Self-training with DwS-C (St-dws-C). Regarding the differences between St-std and St-dws-C, the former uses a confidence-based selection while the later uses a distance-weighted selection, combining the confidence-based selection with a distance metric.

Algorithm 5 Self-Training With Distance-Weighted (St-dws-C)

```
1 while U is not empty do
2     train C with L;
3     compute d_ij for all classes j using L;
4     label U using C;
5     compute DwS-C_i for each i within U;
6     select the best labelled-instances of U according
        DwS_i and join them to L;
7 end
```

Considering the following illustrative example: suppose that we have two unlabelled instances u_1 and u_2 from U. Let d_{1a} and d_{2a} be the distance from these instances to class a, respectively. The steps described in Algorithm 5 are used in this example and it selects only one instance at each iteration. Then, suppose that a classification algorithm assigns class label "a" for both instances u_1 and u_2 (line 4), with the prediction confidences ($conf_{ij}$) for these two instances (u_1 and u_2) equal to 0.80 and 0.90, respectively. Based on these confidence values, we can state that the classification algorithm is more confident in assigning u_2 to class a than u_1.

Note that if the standard Self-training (St-std) is used, then u_2 would be selected. Nevertheless, for St-dws-C, suppose that the Euclidean distance is used to calculate the distance between an instance and the centroid of a class and we obtained $d_{1a} = 1.58$ and $d_{2a} = 2.73$, for u_1 and u_2, respectively. This shows that u_1 is much similar to all instances of class a than u_2. Based on this, we can then calculate DwS_{ij} for u_1 and class a, as follows

$$\mathrm{DwS}_{u_1 a} = \mathrm{conf\_pred}_i \times \frac{1}{d},$$

$$\mathrm{DwS}_{u_1 a} = 0.80 \times \frac{1}{1.58} = \mathbf{0.5063}$$

For u_2 and class a, the same equation (Eq. (2)) was applied and $DwS_{ij} = 0.3297$. Suppose that the results delivered by $DwS_{u_{1a}}$ and $DwS_{u_{2a}}$ represent, respectively, the maximum value achieved by DwS_{ij}, for both instances. In this sense, the DwS values for u_1 and u_2 would be, respectively, 0.5063 and 0.3297. Finally, based on the St-dws-C flow, the instance selected in the selection step would be u_1.

The DwS strategy focus on benefiting the unlabelled instances which are closer to instances of the same class in the labelled set. Consequently, we aim at decreasing the selection of poorly labelled instances, which often happens with semi-supervised methods, mainly in the beginning of the selection process. In the empirical analysis of this paper, as in the above example, the Euclidean Distance between an instance and the centroid of a class will be used as distance metric, for all methods that use a DwS approach.

B. SELF-TRAINING WITH DwS-A

In this approach, the Self-training method uses a Distance-weighted Selection in which an agreement-based selection criterion is combined with a distance metric. For the agreement-based criterion, we use the ensemble-based agreement criterion described in Section II-B (EbAL).

As shown in Algorithms 3 and 4, EbAL has two versions, $v1$ and $v2$. For each version, we implemented the use of Selt-training with DwS, leading to St-dws-A(v1) and St-dws-A(v2) SSL methods. It is important to emphasise that the difference between St-dws-A(v1) and St-dws-A(v2) lies only in the labelling phase (the selection phase is the same), in which St-dws-A(v1) performs labelling using a single classifier (C), and St-dws-A(v2) performs labelling using the output of a classifier ensemble.

C. CO-TRAINING WITH DwS-C

Co-training standard version that uses the DwS-C approach, named Co-training with Distance-weighted Selection (Ct-dws-C), follows the same concept used in Self-training. In addition, DwS-C values are calculated using Eqs. (1) and (2) as well as the confidence prediction as the W_{ij} weight, for both views (classifiers) of a Co-training method. Then, the selection is performed based on the calculated DwS-C values.

After the selection procedure, the same flow of a standard Co-training method is used by Ct-dws-C, in which unlabelled instances chosen by one classification algorithm are added to the labelled set of the other one.

D. CO-TRAINING WITH DwS-A

Co-training with DwS-A versions follow the same idea of St-dws-A(v1) and St-dws-A(v2), including the EbAL as the agreement-based selection criterion and a distance metric. Two versions have been implemented, one for each EbAL version, Ct-dws-A(v1) and Ct-dws-A(v2). In these two versions of Co-training, after computing the classification agreement for all instances from the unlabelled set, the DwS process is applied for each view (classifier). Then, as in the standard Co-training, the unlabelled instances selected by one classifier is included in the labelled set of the other classifier.

V. EXPERIMENTAL METHODOLOGY

This section will present a detailed description of the experimental framework performed in this paper. These details include a description of the datasets employed in the empirical analysis; the baseline methods used in the empirical analysis; the predictive accuracy measures used to assess the quality of the analysed methods; and other materials and methods of this empirical analysis. The methodology and the experimental framework used in this paper are based on the experimental methodology presented in [15].

A. THE EXPERIMENTAL FRAMEWORK

The experimental methodology applied in this paper uses the general functioning of a n-fold cross validation method, with $n = 10$. The main steps are described as follows.

1) shuffle the original dataset;
2) divide the dataset into *10* stratified folds;
3) separate one fold for validation (Validation set - *V*) and 9 folds for training (Training set - *T*);
4) divide the Training set into labelled and unlabelled sets. Usually, we use a small set of labelled instances. In this paper, the proportion of initially labelled instances (*L*) is 10%. Then, the proportion of initially unlabelled instances (*U*) is 90%;
5) apply the semi-supervised method to create an ML model, using *L*;
6) validate the created ML model with *V*;
7) repeat steps 3-6 (using a different fold for validation) until all folds have been used as validation.

The above process is repeated 10 times and a different data distribution for the 10 folds is applied each time. At the end of this process, we will obtain 100 values (10×10-fold cross-validation) and the overall result is defined by averaging these values. Another important aspect of this methodology is that the proportion between labelled and unlabelled instances is set to 10% and 90%, respectively. These values were selected based on previous experiments performed in our former studies, as in [14], [15]. In these experiments, methods using the proportion 10-90 obtained the best results. Therefore, this proportion has been selected as part of the configuration for the experimental methodology of this paper.

TABLE 1. Description of the datasets.

No	Dataset	Inst	Att	Class	Type
d1	Abalone	4177	9	28	C, N
d2	Adult	32561	15	2	C, N
d3	Arrhythmia	452	261	13	N
d4	Automobile	205	26	7	C, N
d5	Blood Transfusion Service	748	5	2	N
d6	Car	1728	6	4	N
d7	Cnae-9	1080	857	9	N
d8	Dermatology	366	35	6	N
d9	Ecoli	336	8	8	C, N
d10	Haberman	306	4	2	N
d11	Hill Valley	606	101	2	N
d12	Indian Liver Patient Dataset (ILPD)	582	10	2	N
d13	King-Rook vs King Pawn	3196	36	2	C
d14	Leukemia Haslinger	100	50	2	N
d15	Madelon	2600	501	2	N
d16	Multiple Features Karhunen	2000	64	10	N
d17	Mushroom	8124	22	2	C
d18	Musk	6598	168	2	N
d19	Nursery	12960	9	5	C
d20	Ozone Level Detection	2536	73	2	N
d21	Pen-based digits	10992	16	10	N
d22	Phishing Website	2456	30	3	N
d23	Planning Relax	182	13	2	N
d24	Seeds	210	7	3	N
d25	Semeion	1593	256	10	N
d26	Solar Flare	1389	13	6	C, N
d27	Solar Flare 1	323	11	8	C, N
d28	Sonar	208	61	2	C, N
d29	Spectf Heart	267	14	2	N
d30	Tic Tac Toe Endgame	958	9	2	C
d31	Twonorm	7400	21	2	N
d32	Waveform	5000	40	3	N
d33	Wilt	4839	6	2	N
d34	Wine	4898	12	11	N
d35	Yeast	1484	9	10	N

B. DATASETS

In this empirical analysis, we aim at assessing the feasibility of the proposed methods. In order to do this, a wide range of classification problems is selected and it is represented in 35 datasets, which were selected from a well-known machine learning repository.[1] Table 1 illustrates a short description of the used datasets, including their reference number (No), name (Dataset), number of instances (Inst), attributes (Att) and classes, as well as the attribute data types (categorial - C or Numeric - N). For this extended version, we included more datasets, seeking to conduct a more exploratory analysis.

C. COMPARATIVE ANALYSIS

As stated earlier, this extended work aims to assess the effectiveness of our proposed DwS approach. In order to do so, a comparative analysis will be performed and discussed in two parts. In the first part, 14 SSL methods are

[1] UC Irvine Machine Learning Repository. Available on https://archive.ics.uci.edu/ml/datasets.php

assessed, our 6 proposed methods, St-dws-C, Ct-dws-C, St-dws-A(v1), St-dws-A(v2), Ct-dws-A(v1) and Ct-dws-A(v2) will be compared to their standard SSL versions, St-std and Ct-std, as well as to the corresponding versions with random selection (St-rand and Ct-rand), and their corresponding SSL versions using only the agreement-based selection criterion, St-EbAL(v1), St-EbAL(v2), Ct-EbAL(v1) and Ct-EbAL(v2). In the second part, the proposed methods that achieved the best results in the first part (best Self-training and best Co-training) will be compared to some existing SSL methods, GRF, LLGC, YATSI and one supervised method with two training strategies, J48-10% and J48-90%.

In [15], a comparative analysis included a different SSL method with a confidence-based selection criterion (Self-training Flex-Con-C1(s) proposed in [42]). Therefore, for simplicity reasons, we decided not to include this SSL method in the comparative analysis of this paper.

D. TIME EFFICIENCY ASPECTS

The set of experiments did not take into account time-related aspects since they were performed in different machines with different hardware resources and with different programming languages. Nonetheless, it was observed that the Ensemble Based (EbAL) approaches take 20 to 40 times longer than the non-EbAL versions, which is justified by the use of a pool of classifiers in EbAL-based Self-training versions and two pools of classifiers in the EbAL-based Co-training versions. Finally, it is important to note that the time spent in an EbAL-based method is strongly related to the type of classifiers used in the pool of classifiers. Thus, we recommend the use of weak classifiers in the composition of the pool. Apart from this observation, the remaining SSL methods have a similar processing time.

E. PREDICTIVE ACCURACY MEASURES

In this experimental analysis, all semi-supervised methods are assessed using two distinct predictive measures, which are: classification accuracy and F-measure. The classification accuracy (or simply accuracy) is defined by dividing the number of instances that are correctly predicted by the total number of testing instances, whereas the F-measure (or F-score) is defined by the harmonic mean between precision and recall [47] and it can be expressed in Eq. 3. Moreover, considering the characteristics of the datasets (i.e., multi-class), we use macro-averaging F-measure.

$$\text{F-measure} = \frac{(2 * \text{P} * \text{R})}{(\text{P} + \text{R})} \tag{3}$$

where:

- *P* is precision, or the positive predictive value. It can be defined as the division of true positive instances by the number of all instances labelled as positive (true and false); and
- *R* is recall and it is defined by the division of true positive instances by the sum of true positive and false negative instances.

In addition, the results delivered by both measures will be analysed from a statistical perspective. In order to do this, the Friedman test will be applied, with a significance level equal to 0.05 (or 5%). The null hypothesis of this test states that there is no significant difference between the average values achieved by the evaluated methods. In cases where the null hypothesis is not accepted, we perform a post-hoc test, which will be the Nemenyi test [48].

F. METHODS AND MATERIALS

The semi-supervised methods that employ the DwS approach will apply the Euclidean Distance between an instance and the centroid of a class as the distance metric (d_{ij} in Eq. 2). This distance is well-known and it is widely applied in several ML tasks. Because of this, we decided to use the Euclidean Distance as the distance metric.

All twelve methods as well as the experimental framework were developed based on the Weka API [49]. For all semi-supervised methods analysed in this paper, a Decision Tree is selected to be the base classifier. In [50], we evaluated the use of several ML models (k-NN, SVM, MLP and DT) as base classifiers in SSL methods with automatic selection procedure. As a result of this evaluation, they all have similar performance (based on statistical analysis). Therefore, this algorithm (DT) was selected due to its simplicity and efficiency. Nevertheless, any other well-known classification algorithm can be used as a base classifier. We used the J48 version which is an implementation of Decision Tree in Weka. In addition, all hyper-parameters were set to their default values. The only exception was confidence factor (depth of tree) that was set to 0.05, defined after an initial investigation.

The proportion of initially labelled instances (L) is set to 10% of the original training set. Regarding this proportion, we tested different other values, such as: 5% and 15%, but none of them produced better results. Therefore, we decided to present the results of this value. Additionally, in the DwS methods, the number of instances to be selected at each iteration is set to 10%. Therefore, these versions select the 10% best-ranked instances according to its DwS values. This value was also based on initial analysis.

Finally, all experiments were performed on a PC (desktop) with the following configuration: Ubuntu 16.04 64 bit; Intel(R) Xeon(R) CPU E5-4610 v4 - 1.80GHz, 6 core; HD with 1 TB; and RAM with 24Gb. In fact, as the Transductive SSL methods (e.g. GRF and LLGC) demand a reasonable quantity of RAM, we added 16Gb of RAM to the initial configuration (8Gb) and ran all the experiments, including the ones previously presented.

1) EXISTING SSL METHODS

For a better understanding of the results obtained by the proposed methods, a comparative analysis will be conducted, comparing the obtained result to some existing SSL methods. In this comparative analysis, We also include a Supervised

method. Thus, the methods used in the comparative analysis are the following ones.

- **Learning With Local and Global Consistency (LLGC)**: this method is a graph-based SSL method that was originally proposed in [29]. The experiments with LLGC were performed using a Weka-based version presented in the Collective Classification Project.[2] For this method, all hyper-parameters are set to the default values;
- **Yet Another Two Stage Idea (YATSI)**: this is a wrapper-based SSL method and was proposed in [51]. The used YATSI version is also available on Weka Collective Project. For this method, the number of nearest neighbours was set to 10 (*setKNN(10)*), without weights (*setNoWeights(true)*) and all other hyper-parameters with default values;
- **Gaussian Random Fields (GRF)**: this is also a graph-based SSL method and GRF was originally proposed in [28]. The experiments using GRF were performed using an R implementation with all hyper-parameters set to default values;
- **Decision Tree**: it is a well-known supervised method available in Weka through its implementation called J48. Regarding the hyper-parameters, confidence factor (depth of tree) was set to 0.05, and all others were set to default values. Moreover, we used two additional training set configurations for J48 which are: a) it uses 10% of the original training set as its actual training set (named as J48-10%); b) it uses 90% of the original training as its actual training set (named as J48-90%). These two training set configurations were used as benchmarks for comparison purposes.

The experimental framework presented in Subsection V-A is also used for all SSL methods described in this section, including their parameter settings. The only exception is the DT method which is a Supervised method and it does not use the unlabelled set of instances.

VI. EXPERIMENTAL RESULTS

This section presents the experimental results, comparing the predictive performance of the proposed approach (DwS-based versions) to eight other baseline methods: (a) St-rand, (b) St-std, (c) St-EbAL(v1), (d) St-EbAL(v2), (e) Ct-rand, (f) Ct-std, (g) Ct-EbAL(v1) and (h) Ct-EbAL(v2). In this paper, a pairwise analysis will be performed, comparing a DwS method and the corresponding non DwS method, in a two-by-two basis. The obtained results are presented in Tables 2 – 5. These tables present average accuracy and F-measure results for all analysed methods (columns from 2 to 8) over all 35 datasets (lines). In these tables, the bold numbers represent the highest value among all analysed methods for each dataset. In addition, the last three lines of each table represent: (a) the overall average accuracy over all datasets (Avg); (b) the overall number of wins for each method (General wins); and (c) the average ranking over all datasets for each method (Avg rank).

For simplicity reasons, this analysis will be divided into two main parts, Self-training and Co-training and the obtained results are evaluated in the next two subsections.

A. SELF-TRAINING METHOD

1) PREDICTIVE ACCURACY

Table 2 presents the experimental results for Self-training method using accuracy as metric of performance. According to Table 2, in general, we can state that St-EbAL(v2) (Self-training using agreement-based selection criterion) obtained the best average accuracy (**73.89%**) among all other methods, having the highest number of wins (12 out of 35 datasets), and also the lowest average ranking (3.03).

When analysing the impact of DwS-C, in comparison with both standard and random selection self-training versions, columns (2-4) in Table 2, we can observe that the use of DwS as a selection criterion had a positive effect in the selection phase of this SSL method. It is important to note that, in this comparison, St-dws-C was more accurate than St-std and St-rand. In addition, St-dws-C achieved a better ranking position and a higher number of wins than St-std and St-rand. In other words, St-dws-C outperformed St-std and St-rand in all three evaluation criteria.

In relation to the use of classification agreement as selection criterion in the first version (named v1), we compare its use on its own (column 5) with its use in the DwS approach (column 6), we can state that its use in the DwS approach had a positive effect on the Self-training performance, allowing St-dws-A(v1) to be slightly better than St-EbAL(v1) in terms of average accuracy, presenting also a better ranking position. However, St-EbAL(v1) had a higher number of wins than St-dws-A(v1) (7 against 3 wins). In fact, we can observe that St-EbAL(v1) is more unstable than St-dws-A(v1) with a higher accuracy variation throughout the analysed datasets.

Although the use of DwS-A had a positive effect on the first DwS-A version, a similar pattern has not been noticed in the second DwS-A version. When comparing the results presented in columns 7 (St-EbAL(v2)) and 8 (St-dws-A(v2)), we can observe St-EbAL(v2) outperformed St-dws-A(v2) in all three evaluation criteria. We believe that this behaviour pattern is due to the fact that the performance of St-EbAL(v2) was already very high, not allowing the proposed combination any opportunity for improvement.

In summary, based on Table 2, we can state that DwS-based methods had better performance in 2 analysed scenarios (out of 3), which may indicate that our proposal has enhanced the accuracy performance of the Self-training SSL method.

Figure 1 presents the CD diagram of the Self-training results (average accuracy) presented in Table 2. In this diagram, the method with the lowest average ranking is St-EbAL(v2) (the leftmost method), followed by St-dws-C and St-dws-A(v2). Of the top four methods, three of them

[2]available on https://github.com/fracpete/collective-classification-weka-package

TABLE 2. Average accuracy for self-training methods.

Dataset	St-rand	St-std	St-dws-C	St-EbAL(v1)	St-dws-A(v1)	St-EbAL(v2)	St-dws-A(v2)
d1	21.31%	20.04%	21.81%	22.96%	22.60%	22.89%	**23.41%**
d2	84.24%	**84.58%**	84.29%	84.18%	84.43%	84.18%	83.30%
d3	54.67%	54.89%	**59.30%**	56.88%	57.54%	54.62%	55.75%
d4	41.60%	41.64%	42.05%	**43.98%**	40.52%	37.43%	38.95%
d5	76.07%	**76.73%**	76.47%	75.53%	76.20%	75.94%	75.26%
d6	76.39%	74.76%	76.22%	**78.18%**	76.45%	78.12%	77.37%
d7	67.69%	68.98%	69.17%	70.19%	68.98%	71.67%	**76.67%**
d8	72.97%	73.49%	**78.94%**	74.32%	73.52%	77.28%	76.77%
d9	73.78%	74.08%	74.98%	74.10%	72.91%	77.05%	**78.88%**
d10	72.87%	74.19%	74.26%	**74.74%**	73.84%	72.53%	74.20%
d11	49.75%	46.46%	**50.90%**	47.52%	50.24%	50.58%	49.26%
d12	66.55%	67.77%	64.50%	64.14%	66.22%	**71.70%**	53.03%
d13	95.28%	94.74%	95.78%	**96.37%**	95.87%	94.24%	95.99%
d14	**73.00%**	60.00%	**73.00%**	69.00%	**73.00%**	69.00%	62.00%
d15	54.15%	53.23%	55.85%	53.19%	**58.12%**	55.23%	52.58%
d16	66.05%	63.95%	69.95%	65.95%	70.30%	80.40%	**81.35%**
d17	98.77%	98.84%	98.87%	98.77%	99.37%	99.02%	**99.43%**
d18	86.12%	88.03%	88.65%	85.68%	85.80%	**93.36%**	90.53%
d19	89.79%	89.81%	89.84%	89.79%	89.75%	**89.92%**	88.34%
d20	96.96%	97.04%	96.96%	96.65%	96.96%	**97.12%**	96.14%
d21	89.05%	89.05%	88.32%	89.26%	89.48%	**91.09%**	88.56%
d22	92.16%	91.92%	92.09%	91.96%	92.02%	**92.68%**	92.57%
d23	57.78%	57.69%	55.50%	60.26%	58.86%	**70.47%**	69.24%
d24	80.95%	79.52%	87.62%	**88.10%**	84.29%	87.14%	85.71%
d25	53.61%	52.35%	56.88%	51.53%	57.81%	69.37%	**70.25%**
d26	71.20%	70.70%	71.49%	70.05%	71.06%	**72.64%**	71.99%
d27	**88.90%**	**88.90%**	**88.90%**	**88.90%**	**88.90%**	88.25%	87.94%
d28	66.69%	65.26%	64.79%	64.31%	63.40%	60.64%	**67.50%**
d29	72.80%	67.62%	69.62%	**74.46%**	68.50%	72.79%	62.45%
d30	68.05%	66.07%	68.05%	66.60%	68.05%	**70.36%**	68.37%
d31	80.61%	79.82%	82.55%	80.68%	81.58%	**85.05%**	84.97%
d32	71.42%	69.84%	74.72%	70.22%	74.74%	**76.52%**	74.28%
d33	**96.73%**	96.67%	95.89%	96.61%	96.42%	94.61%	94.79%
d34	47.35%	43.75%	44.42%	49.88%	49.04%	**51.04%**	47.69%
d35	49.33%	49.73%	52.29%	49.67%	48.85%	51.08%	**52.76%**
Avg	71.56%	70.63%	72.43%	71.85%	72.16%	**73.89%**	72.81%
Wins	3	3	5	7	3	**12**	8
Rank	4.37	4.91	3.31	4.09	3.80	**3.03**	3.77

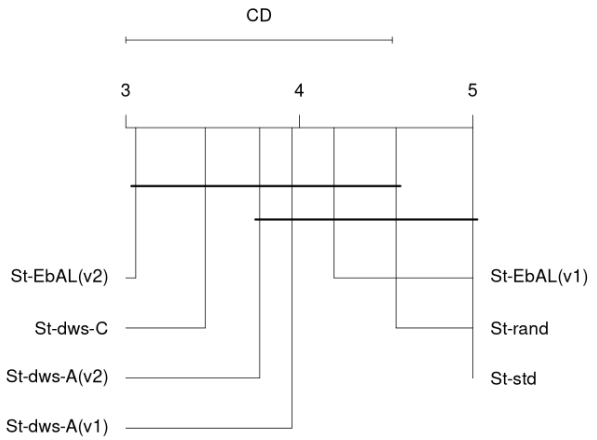

FIGURE 1. Critical difference diagram presenting average accuracy for self-training methods.

contain the proposed selection approach, which is a promising result. Additionally, when comparing St-dws-C and St-std, the statistical test detected difference in performance, with St-dws-C providing more accurate results.

2) F-MEASURE

Table 3 presents the experimental results for Self-training method using F-measure as metric of performance. In general, we can state that St-dws-C (one of the proposed methods) obtained the best F-measure values (**0.5862**) among all other methods, and the lowest average ranking (2.74). However, regarding the number of wins, St-dws-A(v2) provided a higher number of wins, 10 out of 35.

When analysing the results of columns (2-4) in Table 3 (St-rand, St-std and DwS-C), the use of DwS as a selection criterion had a positive effect on the selection procedure of Self-training, when comparing St-std and St-rand against St-dws-C. In this comparison, note that St-dws-C was better than St-std and St-rand in all three evaluation criteria, average F-measure values, ranking position and number of wins (8 out of 35).

Regarding the use of classification agreement in the first DwS-A version, columns 5 (St-EbAL(v1)) and 6 (St-dws-A(v1)), we can state that the use of DwS-A as a selection criterion made St-dws-A(v1) deliver a slightly better than St-EbAL(v1), in all three evaluation criteria, average F-measure values, ranking position and number of wins (6 out of 35).

Finally, when comparing the use of classification agreement in the second DwS-A version, columns 7 (St-EbAL(v2)) and 8 (St-dws-A(v2)), we can state that the use of DwS-A as a selection criterion improved the performance of the selection procedure, leading to an increase in performance, ranking and number of wins.

In summary, based on Table 3, we can state that DwS-based methods had better F-measure performance in all three analysed scenarios, which may indicate that our proposal has enhanced the F-measure performance of the Self-training SSL method.

Figure 2 presents the CD diagram for the Self-training results (average F-measure) presented in Table 3. In this figure, the leftmost method, St-dws-C obtained the lowest average ranking of all analysed methods. From this figure, we can also observe that all three proposed approaches are located in the left part of this diagram, showing that the proposed approaches can help the self-training method to become more effective in imbalanced datasets. Additionally, when comparing St-dws-C against St-std and St-rand, the statistical test detected difference in performance, with St-dws-C providing higher F-measure results.

B. CO-TRAINING METHOD

1) PREDICTIVE ACCURACY

Table 4 presents the experimental results for the Co-training method using accuracy as metric of performance. From this table, we can state that Ct-EbAL(v2) obtained the best results for all three evaluation criteria, the highest average accuracy (**66.62%**), the highest number of wins (14 out of 35 datasets), and the lowest average ranking (2.69).

When analysing DwS-C, in comparison with the standard Co-training and the one with random selection (columns 2-4), we can see that the use of DwS as a selection criterion had

TABLE 3. Average F-measure for self-training methods.

Dataset	St-rand	St-std	St-dws-C	St-EbAL(v1)	St-dws-A(v1)	St-EbAL(v2)	St-dws-A(v2)
d1	0.0920	0.0840	0.0920	0.0890	**0.0970**	0.0760	0.0840
d2	0.7713	**0.7770**	0.7712	0.7700	0.7720	0.7650	0.7500
d3	0.1761	0.1540	0.1890	0.1810	**0.1940**	0.0780	0.0730
d4	0.1876	0.1880	**0.2220**	0.2010	0.1950	0.2020	0.1950
d5	0.4419	0.5490	0.4550	0.4580	0.4320	0.4310	**0.5830**
d6	0.3671	0.3500	0.3735	0.3820	0.3730	0.3600	**0.4810**
d7	0.7461	0.7530	0.7745	0.7570	0.7480	0.7760	**0.7970**
d8	0.6069	0.6460	**0.7165**	0.6080	0.6380	0.6420	0.6280
d9	0.4096	0.4000	**0.4152**	0.4090	0.4070	0.3470	0.3830
d10	**0.5744**	0.5100	0.5723	0.4720	0.5450	0.4790	0.4740
d11	0.3320	0.3400	**0.5180**	0.4140	0.4960	0.5090	0.4750
d12	0.4838	0.5500	0.5690	0.4880	0.5090	0.4390	**0.6320**
d13	0.9530	0.9480	0.9580	**0.9640**	0.9490	0.9430	0.9400
d14	0.7106	0.6330	**0.7110**	0.6580	**0.7110**	0.6840	0.5990
d15	0.5419	0.5320	0.5580	0.5330	**0.5880**	0.5650	0.4980
d16	0.6614	0.6510	0.7030	0.6660	0.7050	0.8070	**0.8110**
d17	0.9878	0.9890	0.9890	0.9880	**0.9940**	0.9900	**0.9940**
d18	0.5876	0.6730	0.7070	0.5480	0.6490	**0.8630**	0.7990
d19	0.5456	0.5700	**0.5730**	0.5460	0.5510	0.5530	0.5540
d20	0.4923	0.5180	0.4920	**0.5290**	0.4920	0.4930	0.5060
d21	0.8909	0.8910	0.8960	0.8930	0.8950	**0.9130**	0.8910
d22	0.9204	0.9180	0.9200	0.9190	0.9190	**0.9260**	0.9250
d23	0.4361	0.4590	0.4200	0.4520	**0.4700**	0.4100	0.4230
d24	0.8099	0.8060	**0.8770**	**0.8770**	0.8460	0.8710	0.8590
d25	0.5479	0.5310	0.5770	0.5260	0.5830	0.7020	**0.7120**
d26	0.5551	0.5720	**0.5950**	0.5400	0.5550	0.5670	0.5440
d27	0.1175	0.1180	0.1180	0.1180	0.1180	0.1250	**0.1310**
d28	0.6652	0.6490	0.6550	0.6410	0.6450	0.6350	**0.6810**
d29	0.6451	0.6020	0.6430	0.6810	0.6180	0.5960	**0.7100**
d30	0.5887	0.4410	0.5880	0.5090	0.5880	**0.6620**	0.5310
d31	0.8064	0.8000	0.8260	0.8070	0.8160	**0.8510**	0.8500
d32	0.7149	0.6980	0.7500	0.7030	0.7500	**0.7720**	0.7590
d33	**0.8165**	0.8080	0.7870	0.8110	0.7950	0.4860	0.5370
d34	**0.1549**	0.1420	0.1460	0.1410	0.1490	0.1080	0.1300
d35	0.3284	0.3440	**0.3600**	0.3300	0.3330	0.3160	0.3500
Avg	0.5619	0.5598	**0.5862**	0.5603	0.5750	0.5698	0.5797
Wins	3	1	9	3	6	6	**10**
Rank	4.54	4.57	**2.74**	4.37	3.51	4.09	3.66

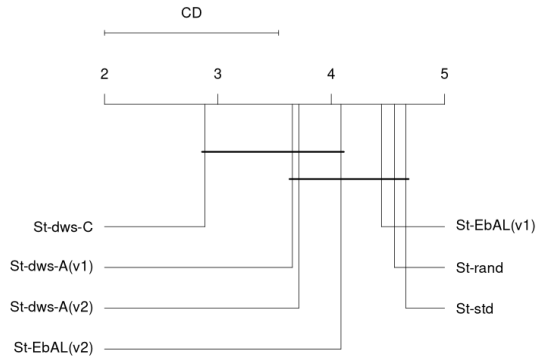

FIGURE 2. Critical difference diagram presenting average F-measure for self-training methods.

a positive effect in the Co-training selection procedure, and this can be observed when comparing Ct-std against Ct-dws-C and Ct-rand. In this comparison, we can observe that Ct-dws-C is more accurate than Ct-std and Ct-rand, and also had a lower ranking position. However, it delivered a slight lower number of wins (4 against 5 and 6, respectively).

Regarding both DwS-A versions, columns (5-8), comparing Ct-EbAL(v1) against Ct-dws-A(v1) and Ct-EbAL(v2) against Ct-dws-A(v2), unlike DwS-C, the use of DwS as a selection criterion deteriorated the performance of the selection procedure of the Co-training method. In this comparison, for both versions, the agreement-based versions

TABLE 4. Average accuracy for co-training methods.

Dataset	Ct-rand	Ct-std	Ct-dws-C	Ct-EbAL(v1)	Ct-dws-A(v1)	Ct-EbAL(v2)	Ct-dws-A(v2)
d1	15.72%	20.17%	21.14%	20.92%	**22.88%**	22.33%	20.38%
d2	75.92%	82.00%	82.10%	81.74%	76.90%	**82.24%**	77.44%
d3	56.82%	56.48%	57.50%	56.36%	57.16%	**58.98%**	57.61%
d4	26.50%	**38.00%**	37.50%	33.25%	32.25%	**38.00%**	26.25%
d5	**76.00%**	73.13%	74.87%	73.60%	73.47%	**76.00%**	75.07%
d6	69.94%	75.09%	74.51%	**75.12%**	69.94%	**75.12%**	74.31%
d7	11.11%	54.40%	**57.08%**	55.42%	44.63%	49.58%	10.79%
d8	28.34%	75.56%	76.25%	75.00%	72.36%	**77.64%**	75.28%
d9	43.75%	61.56%	63.44%	63.44%	52.66%	61.10%	**64.22%**
d10	73.33%	**74.16%**	72.00%	71.83%	73.33%	72.33%	73.16%
d11	**50.00%**	49.96%	49.10%	49.96%	48.61%	49.51%	49.38%
d12	**71.19%**	67.71%	68.05%	66.27%	**71.19%**	70.76%	68.48%
d13	52.19%	74.36%	74.27%	**74.59%**	57.11%	74.20%	72.27%
d14	53.00%	65.50%	69.00%	65.50%	61.00%	62.00%	**74.00%**
d15	49.35%	51.98%	52.80%	52.37%	50.77%	**56.50%**	52.71%
d16	15.30%	44.20%	49.40%	45.42%	30.15%	51.78%	**55.18%**
d17	51.78%	89.68%	**90.56%**	89.56%	79.53%	89.55%	89.06%
d18	84.56%	90.09%	91.18%	90.54%	89.21%	**92.02%**	89.05%
d19	32.42%	62.01%	61.58%	**62.83%**	33.54%	62.41%	61.41%
d20	**97.23%**	97.03%	97.03%	96.87%	**97.23%**	**97.23%**	95.61%
d21	22.80%	78.87%	78.73%	78.80%	**82.11%**	81.33%	18.02%
d22	55.70%	81.95%	82.28%	**82.91%**	82.42%	81.34%	81.77%
d23	67.78%	66.94%	60.56%	64.72%	64.44%	**69.44%**	69.16%
d24	34.28%	75.00%	76.43%	76.43%	74.05%	74.76%	**85.00%**
d25	15.60%	41.76%	43.18%	44.56%	34.31%	**50.60%**	50.34%
d26	23.93%	53.46%	53.00%	52.64%	36.75%	53.43%	**53.86%**
d27	**87.88%**	87.73%	87.42%	**87.88%**	**87.88%**	**87.88%**	86.21%
d28	50.95%	**63.81%**	55.00%	58.09%	57.38%	54.76%	57.86%
d29	**73.53%**	66.47%	67.65%	70.00%	71.91%	72.94%	66.47%
d30	65.62%	**68.02%**	67.86%	67.34%	64.16%	67.29%	65.52%
d31	54.60%	79.05%	81.58%	80.37%	81.93%	**84.80%**	84.60%
d32	33.25%	51.45%	**54.32%**	52.44%	33.47%	54.10%	53.81%
d33	94.63%	95.70%	**96.18%**	95.78%	94.63%	94.63%	94.69%
d34	44.90%	43.96%	45.03%	45.78%	45.76%	**46.97%**	46.86%
d35	30.69%	**43.77%**	41.85%	41.16%	31.10%	38.22%	41.47%
Avg	51.16%	65.74%	66.01%	65.70%	61.03%	**66.62%**	63.35%
Wins	6	5	4	5	5	**14**	5
Rank	5.40	3.71	3.37	3.51	4.66	**2.69**	3.97

(Ct-EbAL(v1) and Ct-EbAL(v2)) outperformed the DwS-A versions (Ct-dws-A(v1) and Ct-dws-A(v2)) in all three evaluation criteria, average accuracy, ranking position and number of wins.

In summary, based on Table 4, we can state that DwS-based methods had better performance in only 1 analysed scenario (out of 3). Unlike Self-training, our proposal did not improve the performance of the selection procedure for the Co-training SSL method.

Figure 3 presents a CD diagram for the Co-training results (average accuracy) showed in Table 4. In this diagram, the leftmost method, Ct-EbAL(v2), obtained the lowest average ranking of all analysed methods. From this diagram, we can also observe that Ct-dws-C is the second method, from the left to right, showing that this method has the second lowest ranking. Nevertheless, the statistical test detected statistically significant difference, only in relation to Ct-rand.

2) F-MEASURE

Table 5 presents the experimental results for Co-training method using F-measure as metric of performance. From a general perspective, we can state that Ct-dws-C obtained the best average F-measure (**0.5203**) among all other analysed methods, having also the highest number of wins (11 out of 35) and providing the lowest average ranking (2.49).

When analysing DwS-C, comparing it to the standard Co-training and the one with random selection (columns 2-4),

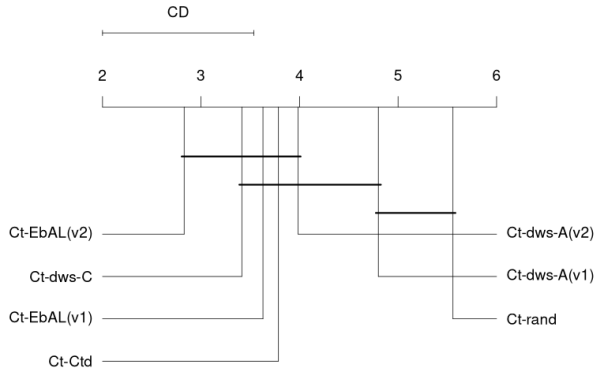

FIGURE 3. Critical difference diagram presenting average accuracy for co-training methods.

we can observe an improvement in performance caused by the use of DwS as a selection criterion, when comparing Ct-std and Ct-rand against Ct-dws-C. Additionally, it is possible to observe that Ct-dws-C outperformed Ct-std and Ct-rand in all three evaluation criteria, F-measure values, ranking position and number of wins (11 against 4 and 0, respectively).

Regarding both DwS-A versions, columns 5 and 6 (Ct-EbAL(v1) against Ct-dws-A(v1)) and 7 and 8 (Ct-EbAL(v2) against Ct-dws-A(v2)), we can state that the use of DwS as a selection criterion did not have a positive effect on the F-measure performance. In this table, it can be seen that Ct-EbAL(v1) overcomes Ct-dws-A(v1) in average F-measure, ranking position and number of wins. Moreover, Ct-dws-A(v2) overcomes Ct-EbAL(v2) in the number of wins and average ranking. However, Ct-EbAL(v2) delivers a much higher F-measure value than Ct-dws-A(v2).

In summary, based on Table 5, we can state that DwS-based methods had better performance in 1 out of 3 analysed scenarios. However, it is important to emphasise that the improvement in performance has always been happening when using the DwS-C selection criterion and, in some cases, in the first DwS-A version.

Figure 4 presents a CD diagram for the Co-training results (average F-measure) showed in Table 5. The leftmost method of this diagram, Ct-dws-C, obtained the lowest average ranking of all. This method uses one of the proposed approaches and it is statistically better than Ct-dws-A(v1) and Ct-rand. Although Ct-dws-C was not statistically better than Ct-dws-A(v2) as it was against Ct-dws-A(v1), we can state that Ct-dws-C obtained the best results among all three DwS-based methods. Therefore, the best combination for the Co-training method, assessing with F-measure, is a confidence-based selection criterion combined with a distance metric.

C. COMPARATIVE ANALYSIS

Based on the extensive analysis carried out over the experimental results presented in Sections VI-A and VI-B, we were able to select the best two proposed methods (one for Self-training and one for Co-training), according to their accuracy and F-measure performances. The best proposed methods are St-dws-A(v2) and Ct-dws-C. These two methods will be compared to the existing methods discussed in Section V-F1, which are: LLGC, YATSI, GRF, J48-10% and J48-90%.

TABLE 5. Average F-measure for co-training methods.

Dataset	Ct-rand	Ct-std	Ct-dws-C	Ct-EbAL(v1)	Ct-dws-A(v1)	Ct-EbAL(v2)	Ct-dws-A(v2)
d1	0.0097	0.0790	**0.0834**	0.0812	0.0722	0.0669	0.0707
d2	0.4316	0.7334	0.7360	0.7292	0.5499	**0.7387**	0.5991
d3	0.0557	0.1613	0.1595	**0.1699**	0.1621	0.0944	0.0692
d4	0.0776	**0.2340**	0.2213	0.2102	0.1989	0.1726	0.1512
d5	0.4318	0.4624	**0.4842**	0.4756	0.4423	0.4318	0.4816
d6	0.2058	0.3012	0.2948	**0.3046**	0.2058	0.2982	0.2989
d7	0.0224	0.6138	**0.6381**	0.6207	0.4889	0.5540	0.0258
d8	0.0868	0.6789	0.6704	0.6564	0.5924	**0.6900**	0.6363
d9	0.0761	0.2458	**0.2839**	0.2638	0.1500	0.2212	0.2577
d10	0.4231	0.4869	0.4425	**0.4912**	0.4231	0.4252	0.4532
d11	0.3333	0.3488	0.4586	0.3488	0.4510	0.3750	**0.4660**
d12	0.4158	0.4436	0.4670	**0.5107**	0.4348	0.4490	0.4795
d13	0.3429	0.7531	0.7531	**0.7556**	0.5139	0.7527	0.7361
d14	0.4610	0.6567	**0.6995**	0.6567	0.6141	0.5738	0.6827
d15	0.4934	0.5199	0.5286	0.5238	0.5078	**0.5515**	0.4600
d16	0.1532	0.4545	0.5035	0.4635	0.3076	0.5318	**0.5708**
d17	0.3412	0.9047	**0.9118**	0.9016	0.8152	0.9034	0.8997
d18	0.4776	0.8029	0.8270	0.8064	0.7689	**0.8336**	0.7643
d19	0.0979	0.3648	0.3621	**0.3730**	0.1368	0.3664	0.3684
d20	0.4930	0.4925	0.4953	0.4921	0.4930	0.4930	**0.4994**
d21	0.2271	0.7909	0.7893	0.7904	**0.8256**	0.8181	0.1967
d22	0.3577	0.8178	0.8209	**0.8285**	0.8238	0.8119	0.8179
d23	0.3992	0.4637	0.4383	**0.4728**	0.4214	0.4154	0.4406
d24	0.1942	0.7573	0.7669	0.7841	0.7398	0.7527	**0.8516**
d25	0.1460	0.4221	0.4330	0.4488	0.3396	0.4967	**0.4977**
d26	0.0642	**0.4122**	0.4093	0.3870	0.2564	0.3946	0.4066
d27	0.3118	0.3115	0.3155	0.3118	0.3118	0.3118	**0.3172**
d28	0.3980	**0.6445**	0.5552	0.5864	0.5679	0.5175	0.5691
d29	0.4237	0.5807	**0.6328**	0.6228	0.6019	0.5746	0.6324
d30	0.3962	0.5289	**0.5672**	0.5651	0.4902	0.5634	0.5490
d31	0.4362	0.7914	0.8160	0.8041	0.8200	**0.8483**	0.8462
d32	0.1976	0.5137	**0.5440**	0.5230	0.1672	0.4921	0.4661
d33	0.4862	0.6507	**0.6835**	0.6610	0.4862	0.4862	0.5165
d34	0.0885	0.1994	**0.2120**	0.1966	0.1412	0.1421	0.1620
d35	0.0469	**0.2310**	0.2072	0.2100	0.0474	0.1703	0.2071
Avg	0.2744	0.5101	**0.5203**	0.5151	0.4391	0.4948	0.4699
Wins	0	4	**11**	8	1	5	6
Rank	6.54	3.43	**2.49**	2.71	4.89	3.89	3.54

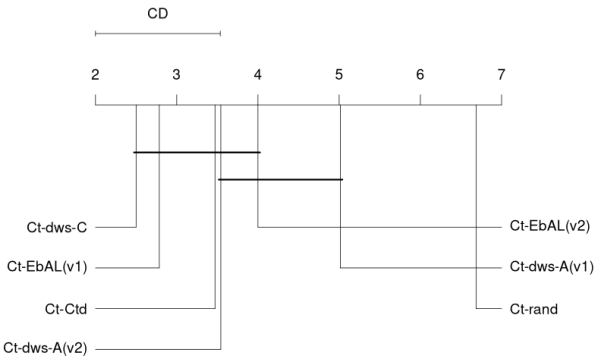

FIGURE 4. Critical difference diagram presenting average F-measure for co-training methods.

1) PREDICTIVE ACCURACY

Table 6 presents the average accuracy results for all seven methods (i.e., three existing SSL methods, two supervised methods and the two proposed methods). In general, we can state that J48-90% obtained the best average accuracy (79.70%) among all analysed methods, having the highest

number of wins (24 out of 35 datasets), and also the lowest average ranking (1.86). This result was expected mainly because J48-90% is a supervised method using a training set of 90% of the original training set, which is nine times bigger than the initial training set of the SSL methods.

When analysing our two proposal methods, we can state that St-dws-A(v2) obtained the best results among them, including average accuracy, number of wins and average ranking. In this sense, we will compare St-dws-A(v2) (i.e., best proposed method) against the existing SSL methods (columns 2-4 and 7). We can observe that St-dws-A(v2) obtained the best average accuracy (72.81%) and the lowest average ranking (3.40). However, it did not obtained the highest number of wins, being only the third method according to this metric (3 out of 35 datasets).

Regarding the overall performance of the existing SSL methods (columns 2-4), we can state that YATSI obtained the best average accuracy (72.67%) and the lowest average ranking (3.46). But it was only second in number of wins (4 out of 35 datasets). Moreover, GRF obtained the highest number of wins (5 out of 35 datasets). However, its performance in average accuracy and average ranking was poor (56.32% and 5.06, respectively). Finally, LLGC was the overall worst baseline method in all three metrics (50.30%), (5.51) and 3 wins.

Figure 5 presents the CD diagram for the average accuracy results presented in Table 6. As we can see, the leftmost method, J48-90% obtained the lowest average ranking of all analysed methods. From this figure, we can also observe that St-dws-A(v2) is the second best method located in that diagram. Additionally, when comparing St-dws-A(v2) against the existing methods, we can observe that the statistical test produced two statistically significant results: St-dws-A(v2) was significantly better than both LLGC ((p-value = 0.0007) and GRF (0.0226). Finally, according to the statistical test, there is no statistical difference between J48-90% (i.e., the overall best method) and St-dws-A(v2) (i.e., best proposed method). This is a promising result since it shows that our proposed method delivered similar performance to a supervised method that has a training set nine times bigger than its initial training set, from a statistical point of view.

TABLE 6. Average accuracy for the best proposed method and baselines.

Dataset	GRF	LLGC	YATSI	J48-10%	J48-90%	St-dws-A(v2)	Ct-dws-C
d1	3.21%	16.49%	20.76%	21.38%	21.88%	**23.41%**	21.14%
d2	55.38%	76.41%	82.06%	84.26%	**85.68%**	83.30%	82.10%
d3	57.05%	54.24%	54.90%	57.32%	**65.53%**	55.75%	57.50%
d4	3.81%	31.17%	35.64%	43.07%	**79.52%**	38.95%	37.50%
d5	73.60%	76.20%	76.07%	76.07%	**76.47%**	75.26%	74.87%
d6	69.94%	70.08%	77.43%	76.45%	**87.38%**	77.37%	74.51%
d7	72.68%	8.04%	66.20%	67.41%	**87.50%**	76.67%	57.08%
d8	82.43%	30.89%	83.88%	74.05%	**93.99%**	76.77%	76.25%
d9	67.50%	42.52%	**80.34%**	73.50%	80.02%	78.88%	63.44%
d10	71.61%	73.22%	73.86%	72.23%	71.28%	**74.20%**	72.00%
d11	**52.79%**	49.51%	49.75%	49.75%	47.36%	49.26%	49.10%
d12	66.95%	**71.19%**	56.73%	66.39%	67.26%	53.03%	68.05%
d13	22.78%	52.25%	94.43%	95.40%	**99.16%**	95.99%	74.27%
d14	**97.00%**	45.00%	61.00%	73.00%	77.00%	62.00%	69.00%
d15	48.27%	48.62%	52.77%	52.58%	**69.27%**	52.58%	52.80%
d16	10.75%	7.40%	**84.25%**	67.55%	81.30%	81.35%	49.40%
d17	**100.00%**	50.90%	99.21%	98.88%	**100.00%**	99.43%	90.56%
d18	55.36%	84.59%	90.01%	85.60%	**99.95%**	90.53%	91.18%
d19	40.31%	33.41%	90.37%	89.79%	**95.63%**	88.34%	61.58%
d20	**98.35%**	97.12%	96.92%	96.96%	97.00%	96.14%	97.03%
d21	11.11%	10.29%	85.08%	89.15%	**96.03%**	88.56%	78.73%
d22	56.09%	56.09%	92.47%	92.14%	**94.94%**	92.57%	82.28%
d23	69.44%	**71.49%**	63.77%	56.08%	**71.49%**	69.24%	60.56%
d24	**90.96%**	34.29%	86.67%	80.48%	90.95%	85.71%	76.43%
d25	17.11%	10.42%	70.81%	53.42%	**74.76%**	70.25%	43.18%
d26	55.21%	29.23%	71.49%	71.20%	**88.90%**	71.99%	53.00%
d27	88.49%	**88.90%**	**88.90%**	**88.90%**	75.31%	87.94%	87.42%
d28	42.38%	48.60%	63.93%	64.31%	**72.19%**	67.50%	55.00%
d29	58.82%	73.09%	70.50%	72.23%	**83.97%**	62.45%	67.65%
d30	68.96%	65.34%	70.98%	68.05%	**83.82%**	68.37%	67.86%
d31	76.41%	50.34%	84.76%	80.82%	**84.97%**	**84.97%**	81.58%
d32	70.02%	32.32%	**77.90%**	71.44%	75.48%	74.28%	54.32%
d33	94.32%	94.61%	94.71%	96.84%	**98.18%**	94.79%	96.18%
d34	0.31%	44.86%	47.16%	47.12%	**56.96%**	47.69%	45.03%
d35	21.81%	31.47%	47.65%	49.06%	**58.22%**	52.76%	41.85%
Avg	56.32%	50.30%	72.67%	71.51%	**79.70%**	72.81%	66.01%
win	5	3	4	1	**24**	3	0
Avg rank	5.06	5.51	3.46	3.71	**1.86**	3.40	4.71

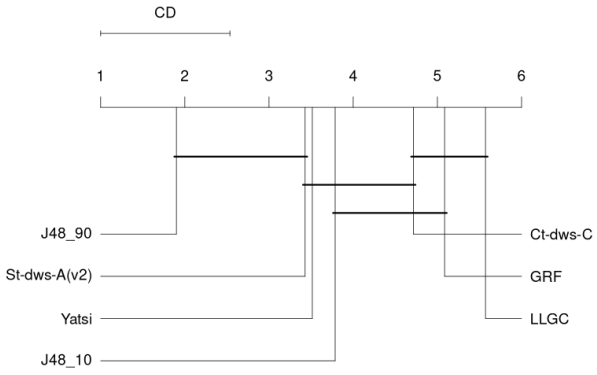

FIGURE 5. Critical difference diagram presenting average accuracy for the best proposed method and baselines.

2) F-MEASURE PERFORMANCE OF THE BEST METHODS

Table 7 presents the average F-measure results for the aforementioned methods. Once again, J48-90% delivered the best results, achieving 0.6538 in average F-measure, 2.11 in average ranking, and number of wins equals to 18, out of 35 datasets.

When analysing our two proposed methods, St-dws-A(v2) also obtained the best results, including average F-measure, number of wins and average ranking. Regarding St-dws-A(v2) compared to the existing SSL methods, we can state that St-dws-A(v2) achieved the best average F-measure (0.5797) and the second lowest average ranking (3.49). However, it did not obtained the highest number of wins, being only the third method according to this metric (2 out of 35 datasets).

In relation to the existing SSL methods, we can state that YATSI obtained the best average F-measure (0.5793) and the lowest average ranking (3.20). But it was only second in number of wins (4 out of 35 datasets). Moreover, once again, GRF obtained the highest number of wins (11 out of 35 datasets). However, it was the fourth in average F-measure (0.5369) and the third in average ranking (3.80). Finally, LLGC was the overall worst baseline method in all three metrics (0.2476), (6.71) and 0 wins.

Figure 6 presents the CD diagram for the average F-measure results presented in Table 7. As we can see that

J48-90% obtained the lowest average ranking of all analysed methods (the leftmost method). From this figure, we can also observe that St-dws-A(v2) is the third best method located in that diagram. Additionally, when comparing St-dws-A(v2) against the existing methods, we can observe that the statistical test produced one statistically significant result: St-dws-A(v2) was significantly better than LLGC (p-value = 0.0001). Finally, according to the statistical test, there is no statistical difference between J48-90% (i.e., the overall best method), YATSI (i.e., the second best method) and St-dws-A(v2) (i.e., best proposed method).

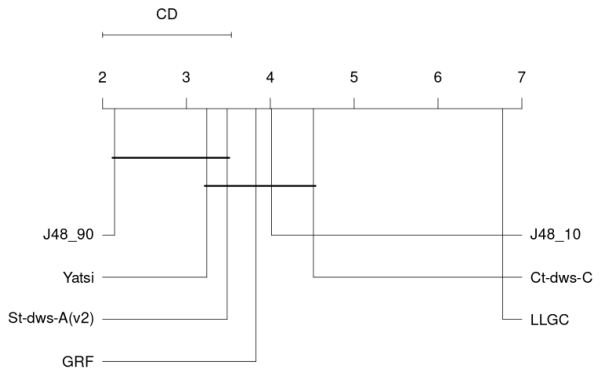

FIGURE 6. Critical difference diagram presenting average F-measure for the best proposed method and baselines.

TABLE 7. Average F-measure for the best proposed method and baselines.

Dataset	GRF	LLGC	YATSI	J48-10%	J48-90%	St-dws-A(v2)	Ct-dws-C
d1	0.0492	0.0101	0.0825	0.0924	**0.0986**	0.0840	0.0834
d2	0.5013	0.4331	0.7362	0.7715	**0.7922**	0.7500	0.7360
d3	**0.7220**	0.0537	0.1018	0.1829	0.2986	0.0730	0.1595
d4	0.0401	0.0735	0.2090	0.1997	**0.5488**	0.1950	0.2213
d5	**0.5949**	0.4321	0.4870	0.4419	0.5730	0.5830	0.4842
d6	**0.8231**	0.2060	0.4291	0.3735	0.6663	0.4810	0.2948
d7	0.7781	0.0280	0.7229	0.7423	**0.8829**	0.7970	0.6381
d8	0.8069	0.0777	0.7195	0.6325	**0.9097**	0.6280	0.6704
d9	0.4408	0.0739	0.4463	0.4071	**0.4584**	0.3830	0.2839
d10	**0.6470**	0.4197	0.4879	0.5656	0.4138	0.4740	0.4425
d11	**0.5285**	0.3309	0.3320	0.3320	0.3350	0.4750	0.4586
d12	0.5618	0.4154	0.4979	0.4627	0.5036	**0.6320**	0.4670
d13	0.2172	0.3430	0.9443	0.9544	**0.9916**	0.9400	0.7531
d14	**0.9725**	0.3052	0.6082	0.7106	0.7724	0.5990	0.6995
d15	0.4736	0.3271	0.5310	0.5267	**0.6929**	0.4980	0.5286
d16	0.0946	0.0414	**0.8455**	0.6780	0.8133	0.8110	0.5035
d17	**1.0000**	0.3349	0.9922	0.9889	**1.0000**	0.9940	0.9118
d18	0.5348	0.4582	**0.8318**	0.5560	0.6990	0.7990	0.8270
d19	0.3812	0.1002	0.5651	0.5456	**0.7063**	0.5540	0.3621
d20	**0.9423**	0.4927	0.5054	0.4923	0.5162	0.5060	0.4953
d21	0.1027	0.0307	0.8414	0.8921	**0.9604**	0.8910	0.7893
d22	0.5287	0.5287	0.9241	0.9203	**0.9488**	0.9250	0.8209
d23	**0.6406**	0.4160	0.4855	0.4346	0.4160	0.4230	0.4383
d24	**0.9106**	0.1691	0.8710	0.8088	0.9097	0.8590	0.7669
d25	0.1720	0.0461	0.7277	0.5485	**0.7523**	0.7120	0.4330
d26	0.5042	0.1243	**0.5614**	0.5557	0.1175	0.5440	0.4093
d27	**0.8026**	0.1175	0.1175	0.1175	0.6091	0.1310	0.3155
d28	0.4010	0.3221	0.6543	0.6412	**0.7259**	0.6810	0.5552
d29	0.5688	0.4210	0.6730	0.6384	**0.7944**	0.7100	0.6328
d30	0.6221	0.3948	0.6698	0.5887	**0.8211**	0.5310	0.5672
d31	0.7641	0.3671	0.8478	0.8083	0.8199	**0.8500**	0.8160
d32	0.6956	0.1627	**0.7810**	0.7157	0.7558	0.7590	0.5440
d33	0.8253	0.4861	0.5737	0.8225	**0.9085**	0.5370	0.6835
d34	0.0187	0.0563	0.1350	0.1519	0.2047	0.1300	**0.2120**
d35	0.1254	0.0676	0.3380	0.3229	**0.4661**	0.3500	0.2072
Avg	0.5369	0.2476	0.5793	0.5607	**0.6538**	0.5797	0.5203
win	11	0	4	0	**18**	2	1
Avg rank	3.80	6.71	3.20	3.97	**2.11**	3.49	4.51

+

D. DISCUSSION OF THE OBTAINED RESULTS

In the previous section, we conducted a robust analysis to investigate the benefits of using a combination of a distance metric with a selection criterion. In order to investigate this combination, we implemented different versions of Self-training and Co-training methods. This section presents an analysis of the obtained results.

When analysing the experimental results presented in Tables 2 and 3, we can conclude that the DwS-based Self-training versions obtained better results than the corresponding versions without this selection criterion (5 out of 6 analysed scenarios for both accuracy and F-measure). It is important to emphasise that St-dws-C uses the proposed approach, showing the importance of combining a selection criterion with a distance metric in the performance of the Self-training method.

In relation to the Co-training methods, the experimental results presented in Tables 4 and 5 show that the DwS-based Co-training versions had better results in only 3 cases (out 6), when compared to their corresponding versions. It is important to highlight that the Co-training with the best F-measure values, Ct-dws-C, uses the proposed approach and it shows the importance of combining a selection criterion with a distance metric in the performance of the Co-training method. However, when compared to the Self-training methods, the use of the DwS approach did not cause a stronger impact in the performance of the Co-training methods. As previously mentioned, Co-training uses two subsets (views) which are presented to two different classifiers. The proposed Co-training methods use the selection values of one classifier (view) to be included in the other one. It means that the selected values are calculated based on one subset, but its corresponding instance is added to the other subset, which may not be as important as it was in the original subset. This is a step that is used in the original Co-training and we decided to maintain it in the proposed methods. However, we believe that this may cause a deterioration in the performance of the proposed approaches.

When comparing the best proposed method against the existing methods (SSL and supervised ones), we can state that J48-90% obtained the best average accuracy and average F-measure among all other methods, having the highest number of wins, and also the lowest average ranking. As mentioned previously, it is an expected result mainly because J48-90% is a supervised method, using a training set composed of 90% of the original training set. However, despite of J48-90%'s performance, our best proposed method (i.e., St-dws-A(v2)) obtained the second best average accuracy and average ranking among all SSL methods. Moreover, St-dws-A(v2) obtained the second best average F-measure and the third best average ranking for this metric. This is a promising result since it shows that St-dws-A(v2) outperformed three well-known SSL methods, mainly in terms of predictive accuracy. On top of that, St-dws-A(v2) was

statistical similar to J48-90% (i.e., the best baseline method), in both predictive accuracy and F-measure.

VII. FINAL REMARKS

This paper presented a new methodology for selecting unlabelled instances for wrapper-based Semi-supervised Learning (SSL) methods. This work is an extension of a previous work [14], including the proposal of one selection version and performing much deeper empirical analysis than [14]. In the proposed approach, a selection criterion (prediction confidence or classification agreement) was combined with a distance metric, in a method called Distance-weighted Selection (DwS-C or DwS-A). This selection method can be adjusted to be used in any wrapper-based SSL method, but, in this paper, we apply this method for Self-training and Co-training SSL methods.

Aiming to investigate the effects of using DwS, different versions of Self-training and Co-training have been implemented, including confidence-based (DwS-C) and agreement-based (DwS-A) methods. Moreover, random versions, standard versions and agreement-based based versions of these methods were also implemented. In the empirical analysis, the main aim was to investigate deeper its effect over 35 well known and diverse datasets using two performance metrics (accuracy and F-measure).

In general, we can state that the adoption of a combined selection criterion had a positive effect in the performance of both analysed SSL methods, being Self-training the one that obtained the highest improvements. It is also important to emphasise that the proposed approach achieved better performance than the original SSL versions, for the majority of analysed scenarios. On top of that, the best proposed method obtained competitive results against five existing methods, being statistically better than two SSL methods, and similar to the supervised methods and one SSL method (YATSI). We can then conclude that the obtained results are promising and they show us that a more effective selection criterion can improve the performance of a wrapper-based SSL method.

In order to further improve the investigation presented in this paper, it would be relevant to apply the selection approaches in other wrapper-based SSL methods, investigating the feasibility of employing our proposal to SSL methods that use distinct sampling and labelling procedures. Finally, distinct approaches for formulating DwS (e.g. linear combination of confidence prediction and distance metric) could also be assessed.

REFERENCES

[1] M. Gopal, *Applied Machine Learning*. New York, NY, USA: McGraw-Hill, 2019.

[2] S. Gollapudi, *Practical Machine Learning*. Birmingham, U.K.: Packt Publishing, 2016.

[3] I. H. Witten, E. Frank, and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. San Mateo, CA, USA: Morgan Kaufmann, 2017.

[4] X. Zhu, A. B. Goldberg, R. Brachman, and T. Dietterich, *Introduction to Semi-Supervised Learning*. San Rafael, CA, USA: Morgan and Claypool, 2009.

[5] G. Bonaccorso, *Machine Learning Algorithms*. Birmingham, U.K.: Packt Publishing, 2017.

[6] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. AISTATS*. Princeton, NJ, USA: Citeseer, 2005, pp. 57–64.

[7] W. Wang and Z. H. Zhou, "Analyzing co-training style algorithms," in *Proc. Eur. Conf. Mach. Learn.* Berlin, Germany: Springer, Sep. 2007, pp. 454–465.

[8] T. Miyato, S.-I. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: A regularization method for supervised and semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 8, pp. 1979–1993, Aug. 2019.

[9] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 11, pp. 1529–1541, Nov. 2005.

[10] X. J. Zhu, "Semisupervised learning literature survey," *World*, vol. 10, p. 10, Sep. 2005.

[11] J. E. Van Engelen and H. H. Hoos, "A survey on semi-supervised learning," *Mach. Learn.*, vol. 109, no. 2, pp. 373–440, 2020.

[12] D. Yarowsky, "Unsupervised word sense disambiguation rivaling supervised methods," in *Proc. 33rd Annu. Meeting Assoc. Comput. Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995, pp. 189–196, doi: 10.3115/981658.981684.

[13] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. 11th Annu. Conf. Comput. Learn. Theory (COLT)*, New York, NY, USA, 1998, pp. 92–100.

[14] C. A. Barreto, A. C. Gorgônio, A. M. Canuto, and J. C. Xavier-Júnior, "A distance-weighted selection of unlabelled instances for self-training and co-training semi-supervised methods," in *Proc. Brazilian Conf. Intell. Syst.* Rio Grande, Brazil: Springer, 2020, pp. 352–366.

[15] C. A. D. S. Barreto, A. M. D. P. Canuto, J. C. Xavier, A. C. Gorgônio, D. F. A. Lima, and R. R. F. da Costa, "Two novel approaches for automatic labelling in semi-supervised methods," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2020, pp. 1–8.

[16] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Hoboken, NJ, USA: Wiley, 2004.

[17] Z.-H. Zhou, *Ensemble Methods: Foundations and Algorithms*. Boca Raton, FL, USA: CRC Press, 2012.

[18] B. Elizalde, A. Shah, S. Dalmia, M. H. Lee, R. Badlani, A. Kumar, B. Raj, and I. Lane, "An approach for self-training audio event detectors using web data," in *Proc. 25th Eur. Signal Process. Conf. (EUSIPCO)*, Kos, Greece, Aug. 2017, pp. 1863–1867. [Online]. Available: http://ieeexplore.ieee.org/document/8081532/

[19] Z. Ju and H. Gu, "Predicting pupylation sites in prokaryotic proteins using semi-supervised self-training support vector machine algorithm," *Anal. Biochem.*, vol. 507, pp. 1–6, Aug. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0003269716300707

[20] J. Jiang, H. Gan, L. Jiang, C. Gao, and N. Sang, "Semi-supervised discriminant analysis and sparse representation-based self-training for face recognition," *Optik*, vol. 125, no. 9, pp. 2170–2174, May 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0030402613013892

[21] S. Qiao, W. Shen, Z. Zhang, B. Wang, and A. L. Yuille, "Deep co-training for semi-supervised image recognition," *CoRR*, vol. abs/1803.05984, pp. 1–17, Mar. 2018.

[22] Y. Xia, F. Liu, D. Yang, J. Cai, L. Yu, Z. Zhu, D. Xu, A. Yuille, and H. Roth, "3D semi-supervised learning with uncertainty-aware multi-view co-training," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Mar. 2020, pp. 3646–3655.

[23] M. S. Hajmohammadi, R. Ibrahim, A. Selamat, and H. Fujita, "Combination of active learning and self-training for cross-lingual sentiment classification with density analysis of unlabelled samples," *Inf. Sci.*, vol. 317, pp. 67–77, Oct. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025515002650

[24] D. Kim, D. Seo, S. Cho, and P. Kang, "Multi-co-training for document classification using various document representations: TF–IDF, LDA, and Doc2Vec," *Inf. Sci.*, vol. 477, pp. 15–29, Mar. 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0020025518308028

[25] P. Kang, D. Kim, and S. Cho, "Semi-supervised support vector regression based on self-training with label uncertainty: An application to virtual metrology in semiconductor manufacturing," *Expert Syst. Appl.*, vol. 51, pp. 85–106, Jun. 2016. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0957417415008295

[26] R. Shi, P. Steenkiste, and M. Veloso, "Second-order destination inference using semi-supervised self-training for entry-only passenger data," in *Proc. 4th IEEE/ACM Int. Conf. Big Data Comput., Appl. Technol.*, Austin, TX, USA, Dec. 2017, pp. 255–264. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3148055.3148069

[27] A. Blum and S. Chawla, "Learning from labeled and unlabeled data using graph mincuts," in *Proc. 18th Int. Conf. Mach. Learn.*, 2001, pp. 19–26.

[28] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," in *Proc. 20th Int. Conf. Mach. Learn. (ICML)*, 2003, pp. 912–919.

[29] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, 2004, pp. 321–328.

[30] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu, "SemiBoost: Boosting for semi-supervised learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 11, pp. 2000–2014, Nov. 2009.

[31] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *Int. J. Mach. Learn. Cybern.*, vol. 8, no. 1, pp. 355–370, Feb. 2017. [Online]. Available: http://link.springer.com/10.1007/s13042-015-0328-7

[32] D. Wu, M. Shang, X. Luo, J. Xu, H. Yan, W. Deng, and G. Wang, "Self-training semi-supervised classification based on density peaks of data," *Neurocomputing*, vol. 275, pp. 180–191, Jan. 2018. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231217309608

[33] J. Chen, J. Feng, X. Sun, and Y. Liu, "Co-training semi-supervised deep learning for sentiment classification of MOOC forum posts," *Symmetry*, vol. 12, no. 1, p. 8, Dec. 2019.

[34] I. Triguero, J. A. Sáez, J. Luengo, S. García, and F. Herrera, "On the characterization of noise filters for self-training semi-supervised in nearest neighbor classification," *Neurocomputing*, vol. 132, pp. 30–41, May 2014. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0925231213011016

[35] S. Bettoumi, C. Jlassi, and N. Arous, "Collaborative multi-view K-means clustering," *Soft Comput.*, pp. 937–945, Sep. 2017, doi: 10.1007/s00500-017-2801-6.

[36] I. Livieris, A. Kanavos, V. Tampakas, and P. Pintelas, "An auto-adjustable semi-supervised self-training algorithm," *Algorithms*, vol. 11, no. 9, p. 139, Sep. 2018. [Online]. Available: http://www.mdpi.com/1999-4893/11/9/139

[37] J. Tanha, N. Samadi, Y. Abdi, and N. Razzaghi-Asl, "CPSSDS: Conformal prediction for semi-supervised classification on data streams," *Inf. Sci.*, vol. 584, pp. 212–234, Jan. 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0020025521010926

[38] T. Suzuki, J. Kato, Y. Wang, and K. Mase, "Domain adaptive action recognition with integrated self-training and feature selection," in *Proc. 2nd IAPR Asian Conf. Pattern Recognit.*, Naha, Japan, Nov. 2013, pp. 105–109. [Online]. Available: http://ieeexplore.ieee.org/document/6778291/

[39] J. Wang, N. Jiang, G. Zhang, B. Hu, and Y. Li, "Automatic framework for semi-supervised hyperspectral image classification using self-training with data editing," in *Proc. 7th Workshop Hyperspectral Image Signal Process., Evol. Remote Sens. (WHISPERS)*, Tokyo, Japan, Jun. 2015, pp. 1–4. [Online]. Available: http://ieeexplore.ieee.org/document/8075485/

[40] K. Yoneda and T. Furukawa, "Distance metric learning for the self-organizing map using a co-training approach," *Int. J. Innov. Comput. Inf. Control*, vol. 14, no. 6, pp. 2343–2351, 2018.

[41] M. Emadi, J. Tanha, M. E. Shiri, and M. H. Aghdam, "A selection metric for semi-supervised learning based on neighborhood construction," *Inf. Process. Manage.*, vol. 58, no. 2, Mar. 2021, Art. no. 102444. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0306457320309365

[42] K. M. O. Vale, A. M. D. P. Canuto, F. L. Gorgônio, A. J. F. Lucena, C. T. Alves, A. C. Gorgônio, and A. M. Santos, "A data stratification process for instances selection in semi-supervised learning," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2019, pp. 1–8.

[43] F. Ma, D. Meng, X. Dong, and Y. Yang, "Self-paced multi-view co-training," *J. Mach. Learn. Res.*, vol. 21, no. 57, pp. 1–38, 2020. [Online]. Available: http://jmlr.org/papers/v21/18-794.html

[44] S. Karlos, G. Kostopoulos, and S. Kotsiantis, "A soft-voting ensemble based co-training scheme using static selection for binary classification problems," *Algorithms*, vol. 13, no. 1, p. 26, Jan. 2020, doi: 10.3390/a13010026.

[45] A. Goel, Y. Jiao, and J. Massiah, "PARS: Pseudo-label aware robust sample selection for learning with noisy labels," *CoRR*, vol. abs/2201.10836, pp. 1–16, Jan. 2022.

[46] J. Lu and Y. Gong, "A co-training method based on entropy and multi-criteria," *Appl. Intell.*, vol. 51, pp. 3212–3225, Jun. 2021.

[47] D. M. Powers, "Evaluation: From precision, recall and f-measure to ROC, informedness, markedness and correlation," *J. Mach. Learn. Technol.*, vol. 2, no. 1, pp. 37–63, 2011. [Online]. Available: https://arxiv.org/ftp/arxiv/papers/2010/2010.16061.pdf

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

[49] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA data mining software: An update," *ACM SIGKDD Explor. Newslett.*, vol. 11, no. 1, pp. 10–18, 2009.

[50] C. A. D. S. Barreto, A. M. D. P. Canuto, J. C. Xavier-Júnior, A. C. Gorgônio, D. F. Lima, and R. R. da Costa, "Evaluating machine learning methods as base classifiers in a SSL method with Na automatic selection procedure," Dept. Inform. Appl. Math., UFRN, Federal Univ. Rio Grande do Norte, Natal, Brazil, Tech. Rep. 2020-1, 2020.

[51] K. Driessens, P. Reutemann, B. Pfahringer, and C. Leschi, "Using weighted nearest neighbor to benefit from unlabeled data," in *Proc. Pacific–Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, Apr. 2006, pp. 60–69.

CEPHAS A. S. BARRETO received the master's degree in software engineering from the Federal University of Rio Grande do Norte, Natal, Brazil, in 2018, where he is currently pursuing the Ph.D. degree in systems and computation. His research interests include machine learning, automotive data, intelligent applications, and semi-supervised methods.

ARTHUR COSTA GORGÔNIO received the B.S. degree in information systems and the M.Sc. degree in systems and computation from the Federal University of Rio Grande do Norte, Natal, Caicó, Brazil, in 2018 and 2021, respectively, where he is currently pursuing the Ph.D. degree. His research interests include machine learning systems, semi-supervised learning, and data-stream classification.

JOÃO C. XAVIER-JÚNIOR received the master's degree from the University of Kent, U.K., in 2001, and the Ph.D. degree from the Federal University of Rio Grande do Norte, Brazil, in 2012. Currently, he is an Associate Professor with the Digital Metropolis Institute, Federal University of Rio Grande do Norte. He has published several articles in scientific journals and conferences. His research interests include semi-supervised learning methods, classifier ensemble methods, clustering and distance metrics, automated machine learning, and evolutionary algorithms optimization.

ANNE MAGÁLY DE PAULA CANUTO (Member, IEEE) received the Ph.D. degree from the University of Kent, in 2001. Currently, she is an Associate Professor with the Informatics and Applied Mathematics Department, Federal University of Rio Grande do Norte, Brazil. She has published several articles in scientific journals and conferences. Her research interests include pattern recognition, clustering algorithms, biometrics, classifier combination, semi-supervised learning, and multiagent systems.