Two Novel Approaches for Automatic Labelling in Semi-Supervised Methods

Cephas A. da S. Barreto
Dept. of Informatics and Applicated Mathematics (DIMAp)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
cephasax@gmail.com

Anne Magály de P. Canuto
Dept. of Informatics and Applicated Mathematics (DIMAp)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
anne@dimap.ufrn.br

João C. Xavier-Júnior
Digital Metropolis Institute (IMD)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
jcxavier@imd.ufrn.br

Arthur Costa Gorgônio
Dept. of Informatics and Applicated Mathematics (DIMAp)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
arthurgorgonio@ppgsc.ufrn.br

Douglas F. A. Lima
Digital Metropolis Institute (IMD)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
douglas.felipelima18@gmail.com

Ranna R. F. da Costa
Digital Metropolis Institute (IMD)
Federal University of Rio Grande do Norte (UFRN)
Natal, Brazil
ranna.raabe@gmail.com

***Abstract*—In real world classification problems, the amount of labelled data is usually limited (very hard or expensive to manually label the instances). However, a natural limitation of a classification algorithm is that it needs to have a set of labelled instances with a reasonable size in order to achieve a reasonable performance. Therefore, one solution to smooth out this problem is the use of semi-supervised learning. Several semi-supervised approaches (e.g. self training) have been proposed in the literature, aiming to use only a few labelled instances, to train a classifier, and to apply a labelling process in which a high number of unlabelled instances is labelled and included in the labelled set. However, this approach can include unreliable instances to the labelled set, impairing the performance of a semi-supervised method. In other words, the selection criterion to include newly labelled instances in the labelled set as well as the labelling step have an important effect in the performance of a semi-supervised method. In this paper, we propose two new approaches for automatic labelling in semi-supervised methods based on the prediction agreement of a pool of classifier as selection criterion. In addition, we compare them to the standard self-training method, and one variation of it called Flexible Confidence Classifier as baselines. In general, both methods obtained significantly better predictive results than the other two methods over 40 classification datasets.**

I. Introduction

Classification is one of the most traditional tasks on the Machine Learning (ML) area. In this task, an instance of a problem is analysed accordingly to a model (function), aiming to define a label for this instance. The building of an ML model, normally, consists of three main phases, data preprocessing, training and testing, and analysis of the obtained results.

It is well known that a natural limitation of a classification algorithm is that it needs to have a set of labelled instances with a reasonable size in order to achieve a reasonable performance. However, in some real world problems, the amount of available labelled data is limited because it is either very hard or expensive to manually label the instances.

In order to deal with this common lack of labelled data, semi-supervised learning (SSL) methods have been proposed in the literature [1], [2], [3], [4]. In particular, the Self-training method was proposed by [5] and explained by [6]. It is a method that uses only a few labelled data, and iteratively applies training and labelling processes, then it selects some newly labelled instances to be included in the labelled set. In this sense, the Self-training method can build a good predictive model to a given problem.

As a semi-supervised technique, although Self-training has proved its efficiency, this method can include unreliable instances in the labelled set, deteriorating the performance of this method. In this sense, the selection criterion and the labelling step have an important impact in the performance of this method and they have to be accurately selected in order to explore the full potential of the Self-training method.

In this context, the main contribution of this work is to propose two new approaches for automatic labelling in semi-supervised methods based on the prediction agreement of a pool of classifiers as selection criterion. The main aim is to apply this prediction agreement as the selection criterion and in the labelling step. These approaches will be assessed in the Self-training method, although it can be applied to any semi-

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

supervised method. In order to assess the feasibility of the proposed approaches, an empirical analysis will be conducted and they will be compared to the standard Self-training method [5] and to one variation called Flexible Confidence Classifier [7] as baselines. In this analysis, we use 40 classification datasets for evaluating the performance of the four aforementioned methods. In order to perform a fair analysis of the experimental results, we apply statistical tests which will be discussed in section V.

The remainder of this paper is organised as follows. Section 2 discusses the background on Self-training methods and Classifier Ensembles. Section 3 discusses related work on different applications for Self-training. Section 4 presents the proposed versions and describes how they differs from the two baseline methods. Section 5 describes the experimental methodology and Section 6 presents the computational results. Finally, Section 7 presents our conclusions and a direction for future work.

II. Background

A. *Self-training*

As mentioned, while supervised learning uses fully labelled datasets for building an ML model, semi-supervised learning (e.g., Self-training) uses only a few labelled instances. Broadly speaking, a Self-training method is designed to deal with domains where it is difficult, expensive or even impossible to have many labelled instances. The general idea (shown in steps) of a basic Self-training method is the following:

1) train classifier C over labelled set (L);
2) label unlabelled set (U);
3) select best labelled instances of U set and join them to L;
4) if U is not empty, go to step 1.

The main idea of the Self-training is simple but effective. Firstly, the Self-training classifier is initially trained with a reduced set of labelled instances. Then it uses its own knowledge to label the unlabelled instances. After that, it selects the best labelled instances based on its confidence prediction (step 3) to be joined to the initial labelled set of instances.

The most traditional implementation of Self-training uses confidence prediction. At each iteration, after step 2, the instances for which the Self-training classifier has assigned the highest degree of confidence prediction are selected and joined to the labelled set (L) (step 3). The confidence prediction is particular to the type of the used classifier. Moreover, this confidence depends on the characteristics of each classifier when building a model. On top of that, misclassifying instances at the beginning of the process may cause a weak model at the end.

Another approach for this methodology (Self-training) uses the similarity between instances, generally measured with any distance metric. In this sense, instead of using confidence prediction at the end of step 2, this approach uses the most similar (nearest) to be selected and joined to the labelled set (L) (step 3). Usually, this is the case when the Self-training classifier is implemented based on any distance (e.g., k-Nearest Neighbour).

As mentioned, an important aspect about this method is that if some misclassified instances are joined to the labelled set, at step 3, the error will be carried out through each next iterations (snowballing), resulting in weak ML models (i.e., less effective in terms of accuracy). For this reason, the variations of Self-training methodology often aim to reduce errors in labelling and selecting at steps 2 and 3, respectively.

B. *Classifier Ensembles*

As well pointed out by [8], there is no ML method better than all others since the predictive performance of an algorithm is strongly dependent on characteristics of the input dataset. In this context, combining the outputs of different base classifiers improves predictive accuracy by comparison with a single base classifier [9].

Classifier ensembles are methods in which a set of base classifiers receive input data and their predicted classes are sent to a combination module, which combines all received predictions into a single predicted class for each instance [10]. Combining the results of different classifiers often outperforms base classifiers [9], [10], since ensembles predictions are usually more robust than the predictions performed by a single classifier. In this sense, diversity plays an important role in choosing different base classifiers for building a robust classifier ensemble.

After choosing the members of the ensemble, a combination method (i.e., this method is used to combine results of the output of the classifiers) needs to be chosen. Several methods have been proposed [10], such as: simple majority voting, weighted voting, sum, among others. In simple majority voting, for instance, the prediction result will be the class which receives the majority of the votes considering all the base classifiers. The number of similar votes by each base classifiers within an ensemble will be called as agreement hereafter.

III. Related Work

Recently, a reasonable number of researches on Self-training methods have been proposed in a wide range of ML areas. This is due to the fact that labelled data is difficult to find or to produce in particular domains. In this sense, these methods have been applied aiming to tackle this lack of labelled data in real world problems.

Moreover, Self-training methods have been applied to deal with problems in different domains, such as: text classification and natural language processing [11], [12], [13]; brain computer interface [14]; bioinformatics [15], [16], [17]; image processing and classification [5], [18], [19], among others.

In addition, the vast majority of the proposed studies aims to improve the original Self-training method in different aspects (e.g., accuracy, robustness, and processing time), as in [20], [21], [22], [23], [24], [25], [26]. These improvements are mainly related to the selection criterion to label the unlabelled instance. In this context, they can be broadly divided into two main groups: confidence-based and distance-based selection methods.

In [20], for instance, the authors proposed a Decision Tree based Self-training method which uses a threshold for determining the initial amount of labelled instances that will be selected. In another work [21], the authors proposed a deep neural network, aiming to lighten possible problems of this methodology by combining the following strategies: pre-training, dropout and error forgetting. Both studies are confidence-based methods and their accuracy results were compared to the original Self-training method.

Differently from the two former works, the work of [22] proposed an ensemble approach which uses a pool of classifiers and a threshold to determine the number of instances that will be included to the labelled set. At each interaction, the best classifier is selected from the classifier pool according to its confidence prediction to label the unlabelled instance and the newly labelled instances are included to the labelled set (L).

On the other hand, in [25] the authors proposed a method based on density peaks of data and differential evolution. The proposed algorithm consists of two main parts. The first part is to use the underlying structure of data space, which is discovered based on density peaks of data, to help training a better classifier. The second part uses the differential evolution to optimise the positioning of the newly labelled data during the self-training process.

Finally, aiming to improve accuracy and to prevent selection of unreliable instances, [26] proposed FlexCon-C1(s), which is a method for dynamically tune the confidence threshold based on the performance of the main classifier at the previous iteration. The confidence threshold is tuned according to Equation III, where $conf(t_{i+1})$ is the confidence threshold for this iteration, mp is the minimum accepted accuracy, cr is a constant that defines how much the confidence threshold changes, and ε is an acceptable change in accuracy. This confidence threshold is used to select the unlabelled instances to be labelled at each iteration.

$$conf(t_{i+1}) = \begin{cases} conf(t_i) - cr, & \text{if } acc \geq mp + \varepsilon \\ conf(t_i), & \text{if } mp - \varepsilon < acc < mp + \varepsilon \\ conf(t_i) + cr, & \text{if } acc \leq mp - \varepsilon \end{cases}$$

In terms of distance-base methods, in [23] and [24], the authors applied this distance metrics for selecting the more similar unlabelled instances. The first one uses this approach for video classification. In the same context, the second one uses the k-NN algorithm as noise reduction classifier by aggregating only the nearest labelled instances.

As the literature points out, the use of confidence-based and distance-based methods for selecting the best unlabelled instances in Self-training methodologies has been investigated by different studies, as in [22], [23], [24]. In addition, the use of dynamic approaches to define the confidence threshold for selecting the best unlabelled instances has been proposed in [25], [26].

Unlike the aforementioned works, our work proposes two new approaches for automatic labelling in semi-supervised methods based on classifier ensemble agreement in a confidence-based method. In these two approaches, a pool of classifiers is trained with limited labelled instances, and through majority voting the best unlabelled instances are selected and included in the labelled set in each iteration.

IV. Proposal

As already discussed in section II (Background), the standard implementation of Self-training classification method uses one base classifier. In this method, a base classifier is responsible for: the classification of unlabelled set (U) at the step 2; and the definition of confidence prediction that can be used as a selection criterion of the unlabelled instances to be included in the labelled set. As it is expected, confidence prediction can be entirely affected by the quality (representation) of the instances used for training. Therefore, the selection a good set of labelled instances is a key factor for promoting robust confidence prediction in a Self-training method. In order to have a set of good quality labelled instances, only reliable newly labelled instances should be include in this set.

In addition, considering the fact that a single classifier can perform poorly in scenarios in which few labelled instances are available, the use of classifier ensembles can improve the predictive accuracy by combining the outputs of many base classifiers (e.g. by majority voting) [9]. In this sense, using a pool of base classifiers for selecting and labelling unlabelled instances, in general, allows the building up of more robust semi-supervised systems.

Based on this fact, this work proposes two new approaches for automatic labelling in semi-supervised methods based on classifier ensemble agreement as selection criterion. As the Self-training has two main steps (e.g., selection and labelling), the version 1 has been designed for using the pool of classifiers only for the selection. This approach (version 1) enhances the processing of selecting the best instances. On the other hand, the second version uses the classifier ensemble for those two steps, allowing the evaluation of improvements (processes) such as: selection (by agreement) and labelling (by voting).

The next two subsections will describe both proposed methods, defining the main steps of them. For simplicity reason, both methods use the following names and their acronyms, described as follows.

- ***i***: instance;
- ***L***: labelled set;
- ***U***: unlabelled set;
- ***C***: main classifier;
- ***PC***: pool of classifiers;
- ***n***: pool size;
- ***A***: agreement;
- ***t***: threshold for agreement (in percentage).

A. Ensemble-Based Automatic Labelling - version 1 (EbAL-v1)

Ensemble based automatic labelling - version 1 (EbAL-v1) is composed of a main classifier (C) and a pool of classifiers. The main difference between EbAL-v1 and the original Self-training method is that the prior uses the agreement of the pool of classifiers (PC) as the selection criterion. The working flow of EbAL-v1 is shown in Algorithm 1.

Algorithm 1: Ensemble-based automatic labelling - version 1

```
1  while $U$ is not empty or no instances were included to $L$ do
2      train $PC$ with $L$
3      train $C$ with $L$
4      for $i$ in $U$ do
5          Using $PC$ assign n pseudo-labels to $i$
6          Compute $A$ of $i$ using the pseudo-labels
7          if $A$ of $i \geq t$ then
8              remove i of U
9              assign class label to i using C
10             add i to L
11         end
12     end
13 end
```

The top-down flow of Algorithm 1 starts with the training of all classifiers of the pool of classifiers (PC), as well as the main classifier (C), using the labelled set (lines 2 and 3). Then, all instances of the unlabelled set (U) are pseudo-labelled by each classifier of the pool, and the prediction agreement among the classifiers (A) is computed for each instance (lines 4, 5 and 6). In case of an unlabelled instance reaches an agreement equal or higher than t% (agreement threshold), then this instance is selected to be included in the labelled set. For these selected instances, the main classifier (C) assigns a definitive label to it, and finally, this current instance is included to the labelled set (L) (lines 7, 8, 9 and 10), and the algorithm starts a new iteration. This process repeats until the unlabelled set (U) is empty or no one instance has been selected according to the agreement threshold at the current iteration.

As mentioned previously, this proposed algorithm differs from the original Self-training method mainly in the used selection criterion. The main idea behind EbAL-v1 is to select an unlabelled instance only when the majority of classifiers of a PC can agree about its label. In this case, by using an agreement from different classifiers we believe that only reliable instances will be selected. As a consequence, this proposed method allows the building up of more accurate semi-supervised models.

After selecting the unlabelled instances (in EbAL-v1), the main classifier assigns the definitive label for each instance that has been selected, and then, this current instance is included to the labelled set. This labelling step is similar to the standard Self-training method.

B. Ensemble-based Automatic Labelling - version 2 (EbAL-v2)

Ensemble-based Automatic labelling - version 2 (EbAL-v2) is also composed of a pool of classifiers (PC), as EbAL-v1. However, the main difference is that the labelling step is performed by a classifier ensemble combined by a majority voting method. The working flow of EbAL-v2 is presented in Algorithm 2.

Algorithm 2: Ensemble-based automatic labelling - version 2

```
1  while $U$ is not empty or No instances were added to $L$ do
2      train $PC$ with $L$
3      for $i$ in $U$ do
4          Using $PC$ assign $n$ pseudo-labels to $i$
5          Compute $A$ of $i$ using the pseudo-labels
6          if $A$ of $i \geq t$ then
7              remove i of U
8              assign the class with the highest A to i
9              add i to L
10         end
11     end
12 end
13 train $C$ with $L$
```

The top-down flow of Algorithm 2 starts with the training of all classifiers within the pool of classifiers (PC) using the labelled set (L) (line 2). Then, for each instance (i) of the unlabelled set (U) a pseudo-label is assigned by each classifier of PC, and the prediction agreement (A) is computed for each instance (lines 3, 4 and 5). In case of an instance reaches an prediction agreement equals or higher than t% (agreement threshold), then this instance is selected to be labelled and it is removed from the unlabelled set (U). Then the predictions of all classifiers are then combined using a majority voting method (lines 6, 7 and 8). Thereafter, these instances are included to the labelled set (L) (line 9). As in EbAL-v1, this algorithm repeats until the unlabelled set (U) is empty or no instance was selected, according to the prediction agreement at the current iteration.

After the labelling process, as in the standard Self-training, the main classifier is trained with the final labelled set (L) to be further tested.

C. General Remarks

As previously discussed, EbAL-v1 version uses a pool of classifiers to pseudo-label instances at each iteration. Based on this context, as in the standard Self-training method, the main classifier performs the labelling step for the selected unlabelled instances. On the other hand, in EbAL-v2, the pool of classifiers itself performs the labelling step by using a classifier ensemble combined by majority voting. In the later version, the idea of having an ensemble performing selection and, more importantly, the labelling step. It tends to avoid that a high number of labelling errors occurs in future iterations.

Finally, the results of these two methods proposed in this paper may indicate that the agreement promoted by a pool of classifiers is a promising way to build up ML models based on the Self-training methodology.

V. Experimental Methodology

This section discusses important details of the used experimental framework, the datasets used in experiments, the baseline methods and their configuration settings, and, finally, the predictive measures used for comparison purposes.

A. The Experimental Framework

The general methodology of this empirical analysis is based on an n-fold cross validation method, and it can be explained in the following steps.

1) shuffle the dataset;
2) split the dataset into *10* stratified folds;
3) separate fold 1 for validation (Validation set - V);
4) use the remaining folds (2 to 10) being 10% for the labelled set (L) and 90% for the unlabelled set (U);
5) build an ML model by performing the Self-training implementation using U and L;
6) validate the built model using V and save the obtained results;
7) repeat steps from 3 (changing the fold used for validation) until all folds have been used as validation.

At the end of this process, it is expected that 10 values have been saved. This process is repeated 10 times, with different data distribution in the folds. The obtained result is given by the average result of a 10x10-fold cross validation.

B. Datasets

In order to evaluate the proposed methods, 40 classification datasets were used, available for downloading from well known machine learning repositories. Table I presents the description of all datasets, including the reference number of the dataset (No), name (Dataset), number of instances (Inst), attributes (Att) and classes, and also the data type of the attributes (categorical - C or Numeric N).

TABLE I
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	Cnae-9	1080	857	9	N
d7	Dermatology	366	35	6	N
d8	Ecoli	336	8	8	C, N
d9	German Credit	1000	21	2	C, N
d10	Glass	214	10	6	N
d11	Haberman	306	4	2	N
d12	Hill Valley	606	101	2	N
d13	Image Segmentation	2310	19	7	N
d14	Indian Liver Patient	582	10	2	N
d15	King-Rook vs King Pawn	3196	36	2	C
d16	Leukemia Haslinger	100	50	2	N
d17	Madelon	2600	501	2	N
d18	Mammographic Mass	961	6	2	N
d19	Multiple Features Karhunen	2000	64	10	N
d20	Mushroom	8124	22	2	C
d21	Nursery	12960	9	5	C
d22	Ozone Level Detection	2536	73	2	N
d23	Pen-based digits	10992	16	10	N
d24	Phishing Website	2456	30	3	N
d25	Pima	768	9	2	N
d26	Planning Relax	182	13	2	N
d27	Secon	1567	591	2	N
d28	Seeds	210	7	3	N
d29	Semeion	1593	256	10	N
d30	Solar Flare 1	323	11	8	C, N
d31	Solar Flare	1389	13	6	C, N
d32	Sonar	208	61	2	C, N
d33	Spectf Heart	267	14	2	N
d34	Tic Tac Toe Endgame	958	9	2	C
d35	Twonorm	7400	21	2	N
d36	Vehicle	946	18	4	N
d37	Waveform	5000	40	3	N
d38	Wilt	4839	6	2	N
d39	Wine	4898	12	11	N
d40	Yeast	1484	9	10	N

C. Methods for the Comparative Analysis

Automatic Labelling Ensemble Based version 1 (EbAL-v1) and version 2 (EbAL-v2) are compared to a strong baseline semi-supervised method, called FlexCon-C1(s), proposed in [26]. We also compared both approaches to the standard Self-training method proposed in [5].

The standard Self-training method is composed of a main classifier and, as described in section II, it selects the best labelled instances at each iteration based on the confidence prediction of its classifier. On the other hand, FlexCon-C1(s) uses a threshold confidence value to decides whether an instance will be selected or not in the current iteration [26]. This method changes the threshold dynamically throughout the iterations, and it uses a base classifier to define whether the threshold confidence value will be changed or maintained (e.g., increase, decrease or constant).

D. Predictive Accuracy Measures

All methods are evaluated based on two predictive accuracy measures, which are classification accuracy rate and F-measure. The accuracy rate simply measures the confidence (number of correct predictions) of the analysed model. On the other hand, F-measure (also called F-score) is the harmonic average between precision and recall [27] and it is defined as:

$$F - measure = \frac{(2 * precision * recall)}{(precision + recall)} \qquad (1)$$

Precision, also known as positive predictive value, is defined as the proportion of positive results that truly are positive (regardless of they belonging to the positive or negative class). Recall, also refer to as sensitivity, is defined as the ability of a test to correctly identify positive results to get the true positive rate (regardless of they being correctly or wrongly classified).

E. Methods and Materials

The implementation of all four methods and development of the experimental framework are based on the Weka API

[28]. In these methods, a Decision Tree (J48 - i.e., confidence factor = 0.05) was used as the main classifier. The pool of classifiers (i.e., used by methods EbAL-v1 and EbAL-v2) is composed by 20 classifiers with moderate diversity. The names of the classifiers, their acronyms, and the number of each one are highlighted as follows: Support Vector Machine (SMO) - 5; k-Nearest Neighbour (IBK) - 5; Decision Tree (J48) - 4; Naive Bayes (NB) - 3; and Decition Table (DT) - 3. Table II shows the parameter settings used for each classifier within the pool for both EbAL-v1 and EbAL-v2.

TABLE II
CLASSIFIERS USED FOR SELF-TRAINING VERSIONS

Ord	Type	Parameter Settings
1	SMO	-C 1.0; PolyKernel -E 1.0 -C 250007
2	SMO	-C 0.8; PolyKernel -E 1.0 -C 250007
3	SMO	-C 1.0; NormalizedPolyKernel -E 2.0 -C 250007
4	SMO	-C 1.0; RBFKernel -C 250007
5	SMO	-C 1.0; Puk -O 1.0 -S 1.0 -C 250007
6	IBK	-K 1; LinearNNSearch; EuclideanDistance
7	IBK	-K 3; LinearNNSearch; EuclideanDistance
8	IBK	-K 3; LinearNNSearch; ManhattanDistance
9	IBK	-K 5; LinearNNSearch; EuclideanDistance
10	IBK	-K 5; LinearNNSearch; ManhattanDistance
11	J48	-C 0.25 -M 2
12	J48	-C 0.20 -M 2
13	J48	-C 0.10 -M 2
14	J48	-C 0.05 -M 2
15	NB	–
16	NB	-K
17	NB	-D
18	DT	-X 1; BestFirst -D 1 -N 5
19	DT	-X 1; BestFirst -D 1 -N 3
20	DT	-X 1; BestFirst -D 1 -N 7

Regarding the number of instances to be selected at each iteration (step 3), this value differs according to the characteristics of each method. Both EbAL-v1 and EbAL-v2 are designed to select a number of instances in agreement equal or greater than a prediction agreement (A). In this empirical analysis A was set to 75%. On the other hand, the standard Self-training was implemented to select 10% of the unlabelled set (U) at each iteration. And, finally, FlexCon-C1(s) was implemented to select all instances with a confidence prediction equals or higher than 95% at the first iteration. After that, the confidence threshold changes dynamically according to Eq. III.

Finally, we run the experiments on a desktop PC with Ubuntu 16.04 64 bit operating system driven by an Intel(R) Xeon(R) CPU E5-4610 v4 - 1.80GHz, 6 core, and RAM with 6 Gb.

VI. EXPERIMENTAL RESULTS

This section presents the experimental results, comparing the predictive performance of the proposed two approaches (EbAL-v1 and EbAL-v2) to two other baseline methods: (a) the FlexCon-C1(s), and (b) the standard Self-training method.

A. Results for the Average Accuracy

Table III presents the accuracy rates for all four methods. Note that the best result for each dataset is presented in boldface. In addition, for each method, its number of wins (i.e., the number of datasets where it obtained the highest accuracy rate) and its average rank are shown at the bottom of this table. The lower the average rank of a method is, the better its predictive performance.

As shown at the bottom of Table III, EbAL-v2 obtained the best (lowest) average rank (1.83), with EbAL-v1 in the second place (rank 2.23). In addition, EbAL-v2 and EbAL-v1 achieved the highest accuracy rate among all methods in 21 and 9 of the 40 datasets, respectively. On the other hand, FlexCon-C1(s) achieved 4 wins and the standard Self-training achieved 6 wins, reaching a total of 10 wins.

In order to conduct a statistical analysis of the results, the Friedman test and Nemenyi post-hoc test are used (as recommended in [29]) to determine whether or not there is a statistically significant difference between the predictive accuracies of the analysed methods across the 40 datasets. Both tests are applied at the conventional significance level of 5%. Table IV presents the results of the Friedman test (first line) and of the Nemenyi post-hoc test (lines 2 to 4).

As it has been shown in Table IV, the Friedman test produced the p-value = 0.000005. Therefore, the difference between the average accuracy of all four methods is statistically significant. The pairwise comparisons using the Nemenyi post-hoc test produced three statistically significant results, which are: EbAL-v1 against FlexCon-C1(s) (p-value = 0.001570), EbAL-v2 against FlexCon-C1(s) (p-value = 0.000003) and against standard Self-training (p-value = 0.046268). That is, there is no significant difference between the average accuracy of remaining pairs of methods.

B. Results for the Average F-measure

Table V presents the average values of F-measure for all four methods. As shown at the bottom of this table, EbAL-v2 obtained the best (lowest) average rank (2.33), with EbAL-v1 in the second place (rank 2.35). In addition, EbAL-v2 and EbAL-v1 achieved the highest accuracy among all methods in 13 and 8, reaching a total of 21 out of 40 datasets, respectively, whilst FlexCon-C1(s) achieved 9 wins and the standard Self-training achieved 10 wins, reaching a total of 19.

We also applied the aforementioned Friedman and post-hoc Nemenyi tests to the method results for F-measure. The Friedman test produced the p-values of 0.1558, therefore the difference between the average F-measure of the four methods is no statistically significant.

Although there was no statistical difference between the F-measure results for all four compared methods (EbAL-v1, EbAL-v2, FlexCon-C1(s) and the standard), it is important to note that EbAL-v2 and EbAL-v1 obtained the best results in the average rank. This fact shows that the proposed approaches (EbAL-v1 and EbAL-v2) have outperformed the baseline methods (FlexCon-C1(s) and the standard Self-training) in 3 out of 4 cases - i.e, in terms of accuracy (wins and rank), and in terms of F-measure (wins and rank).

Additionally, although EbAL-v2 outperformed all methods, there was no statistically significant difference between the

TABLE III
AVERAGE ACCURACY

Dataset	EbAL-v1	EbAL-v2	Standard	FlexCon-C1(s)
d1	**22.96%**	22.89%	20.04%	8.76%
d2	84.18%	84.18%	**84.58%**	84.12%
d3	56.88%	54.62%	54.89%	**57.73%**
d4	**43.98%**	41.43%	41.64%	36.00%
d5	75.53%	75.94%	**76.73%**	75.07%
d6	70.19%	**71.67%**	68.98%	56.67%
d7	74.32%	**77.28%**	73.49%	68.61%
d8	74.10%	**77.05%**	74.08%	74.38%
d9	**70.90%**	69.70%	70.60%	68.50%
d10	46.65%	**50.30%**	49.05%	43.18%
d11	**74.74%**	72.53%	74.19%	74.00%
d12	47.52%	**50.58%**	46.46%	49.92%
d13	64.14%	**71.70%**	67.77%	68.14%
d14	88.79%	**90.17%**	89.87%	89.70%
d15	**96.37%**	94.24%	94.74%	95.16%
d16	**69.00%**	69.00%	60.00%	68.00%
d17	53.19%	**55.23%**	53.23%	52.08%
d18	79.08%	78.87%	**79.71%**	76.39%
d19	65.95%	**80.40%**	63.95%	48.75%
d20	98.77%	**99.02%**	98.84%	85.77%
d21	89.79%	**89.92%**	89.81%	89.58%
d22	96.65%	**97.12%**	97.04%	95.69%
d23	89.26%	**91.09%**	89.05%	83.87%
d24	91.96%	**92.68%**	91.92%	87.26%
d25	68.86%	71.63%	**72.65%**	64.03%
d26	60.26%	**70.47%**	57.69%	61.67%
d27	91.07%	93.37%	91.58%	**93.59%**
d28	**88.10%**	87.14%	79.52%	74.76%
d29	51.53%	**69.37%**	50.35%	39.43%
d30	**88.91%**	88.25%	88.90%	62.50%
d31	70.05%	72.64%	69.70%	**87.88%**
d32	64.31%	60.64%	**65.26%**	55.24%
d33	**74.46%**	72.79%	67.62%	67.94%
d34	66.60%	**70.36%**	66.07%	63.44%
d35	80.68%	**85.05%**	79.82%	79.85%
d36	57.83%	56.62%	**60.75%**	57.76%
d37	70.22%	**76.52%**	69.84%	69.42%
d38	96.61%	94.61%	96.67%	**97.27%**
d39	49.88%	**51.04%**	43.75%	26.02%
d40	49.67%	**51.08%**	49.53%	47.33%
Wins	9	**21**	6	4
Avg Rank	2.23	**1.83**	2.60	3.30

TABLE V
AVERAGE F-MEASURE

Dataset	EbAL-v1	EbAL-v2	Standard	FlexCon-C1(s)
d1	**0.089**	0.076	0.084	0.060
d2	0.770	0.775	**0.777**	0.770
d3	**0.181**	0.078	0.154	0.180
d4	0.201	0.202	0.188	**0.210**
d5	0.458	0.431	**0.549**	0.520
d6	0.757	**0.776**	0.753	0.600
d7	0.608	0.642	**0.646**	0.610
d8	**0.409**	0.347	0.400	0.350
d9	0.495	0.492	0.491	**0.560**
d10	0.296	**0.314**	0.300	0.250
d11	0.472	0.479	**0.510**	0.470
d12	0.414	**0.509**	0.340	0.370
d13	0.488	0.489	**0.550**	0.520
d14	0.889	**0.904**	0.901	0.900
d15	**0.964**	0.943	0.948	0.950
d16	0.658	0.684	0.633	**0.690**
d17	0.533	**0.565**	0.532	0.520
d18	**0.800**	0.790	0.797	0.760
d19	0.666	**0.807**	0.651	0.500
d20	0.988	**0.990**	0.989	0.870
d21	0.546	0.553	0.570	**0.580**
d22	0.529	0.523	0.518	**0.530**
d23	0.893	**0.913**	0.891	0.840
d24	0.919	**0.926**	0.918	0.880
d25	0.681	0.633	**0.712**	0.610
d26	0.452	0.410	0.459	**0.510**
d27	**0.521**	0.483	0.513	0.480
d28	**0.877**	0.871	0.806	0.750
d29	0.526	**0.702**	0.531	0.410
d30	0.118	0.125	0.118	**0.480**
d31	0.540	0.567	**0.572**	0.310
d32	0.635	0.641	**0.649**	0.550
d33	**0.681**	0.596	0.602	0.590
d34	0.509	**0.662**	0.441	0.600
d35	0.807	**0.851**	0.800	0.800
d36	0.589	0.556	**0.613**	0.580
d37	0.703	**0.772**	0.698	0.700
d38	0.811	0.486	0.808	**0.830**
d39	0.141	0.108	0.142	**0.180**
d40	0.330	0.316	**0.344**	0.310
Wins	8	**13**	10	9
Avg Rank	2.35	**2.33**	2.40	2.90

TABLE IV
FRIEDMAN AND NEMENYI - ACCURACY

p-value		*0.000005*	
	EbAL-v1	**EbAL-v2**	**Standard**
EbAL-v2	0.508353	-	-
Standard	0.618912	***0.046268***	-
FlexCon-C1(s)	***0.001570***	***0.000003***	0.072451

results of EbAL-v2 and EbAL-v1, for all measures. This fact indicates that both proposed methods have dealt well with the challenge of automatic labelling of unlabelled data.

VII. CONCLUSION AND FUTURE WORK

This work proposed two new approaches for automatic labelling in semi-supervised methods, which were based on the prediction agreement of a pool of classifiers as selection criterion, called EbAL-v1 and EbAL-v2. The main idea behind of the proposed methods is to select an unlabelled instance only when the majority of classifiers of a pool can agree about its label. In this case, by using an agreement from different classifiers we believe that only reliable instances will be selected. As a consequence, the proposed methods produce more accurate semi-supervised models.

In order to assess the feasibility of the proposed methods, an empirical analysis was conducted. In this analysis, these two new approaches were compared against two strong baseline methods: the FlexCon-C1(s) method proposed in [26], and the standard Self-training method proposed in [5]. In addition, 40 classification datasets were used and, in general, the proposed methods outperformed the baselines ones in all predictive accuracy measures used in our analysis, namely accuracy and F-measure. As a result, we can conclude that EbAL-v2 significantly outperformed FlexCon-C1(s) and the standard self-training in terms of accuracy. In addition, the EbAL-v1 significantly outperformed FlexCon-C1(s) and it has also outperformed the standard Self-training, however there was no statistically significant difference between these two methods.

We also analysed the prediction results of both proposed methods, and, overall, EbAL-v2 outperformed EbAL-v1 with no statistical significance in terms of accuracy and F-measure. In fact, EbAL-v2 was the best method among all four over 40 classification datasets, being EbAL-v1 the second best.

However, when analysing only strongly unbalanced and very small datasets we detected that both EbAL-v1 and EbAL-v2 decreased their performances over those datasets. This fact may be explained based on the required 75% agreement for selection witch is quite demanding in terms of the total number of ensemble members.

As future work, it would be interesting to extend the experiments in terms of agreement within the classifier ensemble (e.g., distance metrics and different agreement percentages) aiming to reduce the errors propagated in each iteration. Moreover, we intend to investigate the use of our proposed Automatic Labelling in fairly new SSL methods found in literature.

Acknowledgements

This work has been partially supported by CAPES (Brazilian Research Council).

References

[1] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation." in *AISTATS*, vol. 2005. Citeseer, 2005, pp. 57–64.

[2] W. Wang and Z.-H. Zhou, "Analyzing co-training style algorithms," in *European conference on machine learning*. Springer, 2007, pp. 454–465.

[3] T. Miyato, S.-i. Maeda, M. Koyama, and S. Ishii, "Virtual adversarial training: a regularization method for supervised and semi-supervised learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 8, pp. 1979–1993, 2018.

[4] Z.-H. Zhou and M. Li, "Tri-training: Exploiting unlabeled data using three classifiers," *IEEE Transactions on Knowledge & Data Engineering*, no. 11, pp. 1529–1541, 2005.

[5] C. Rosenberg, M. Hebert, and H. Schneiderman, "Semi-Supervised Self-Training of Object Detection Models," in *2005 Seventh IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05) - Volume 1*. Breckenridge, CO: IEEE, Jan. 2005, pp. 29–36. [Online]. Available: http://ieeexplore.ieee.org/document/4129456/

[6] X. Zhu and A. B. Goldberg, "Introduction to semi-supervised learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[7] K. M. Ovidio Vale, A. M. d. P. Canuto, A. d. M. Santos, F. d. L. e. Gorgonio, A. d. M. Tavares, A. C. Gorgnio, and C. T. Alves, "Automatic Adjustment of Confidence Values in Self-training Semi-supervised Method," in *2018 International Joint Conference on Neural Networks (IJCNN)*. Rio de Janeiro: IEEE, Jul. 2018, pp. 1–8. [Online]. Available: https://ieeexplore.ieee.org/document/8489128/

[8] Y.-C. Ho and D. L. Pepyne, "Simple explanation of the no-free-lunch theorem and its implications," *Journal of optimization theory and applications*, vol. 115, no. 3, pp. 549–570, 2002.

[9] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. Chapman and Hall/CRC, 2012.

[10] L. I. Kuncheva, "Combining pattern classifiers: Methods and algorithms," 2004.

[11] 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," *Information Sciences*, vol. 317, pp. 67–77, Oct. 2015. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0020025515002650

[12] M. Pavlinek and V. Podgorelec, "Text classification method based on self-training and lda topic models," *Expert Systems with Applications*, vol. 80, pp. 83–93, 2017.

[13] P. Zhang and Z. He, "A weakly supervised approach to Chinese sentiment classification using partitioned self-training," *Journal of Information Science*, vol. 39, no. 6, pp. 815–831, Dec. 2013. [Online]. Available: http://journals.sagepub.com/doi/10.1177/0165551513480330

[14] Y. Li, C. Guan, H. Li, and Z. Chin, "A self-training semi-supervised svm algorithm and its application in an eeg-based brain computer interface speller system," *Pattern Recognition Letters*, vol. 29, no. 9, pp. 1285–1294, 2008.

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

[16] Y. El-Manzalawy, E. E. Munoz, S. E. Lindner, and V. Honavar, "PlasmoSEP: Predicting surface-exposed proteins on the malaria parasite using semisupervised self-training and expert-annotated data," *PROTEOMICS*, vol. 16, no. 23, pp. 2967–2976, Dec. 2016. [Online]. Available: http://doi.wiley.com/10.1002/pmic.201600249

[17] Y. Li, Y. Yin, L. Liu, S. Pang, and Q. Yu, "Semi-supervised Gait Recognition Based on Self-Training," in *2012 IEEE Ninth International Conference on Advanced Video and Signal-Based Surveillance*. Beijing, China: IEEE, Sep. 2012, pp. 288–293. [Online]. Available: http://ieeexplore.ieee.org/document/6328031/

[18] X. Zhao, N. Evans, and J.-L. Dugelay, "Semi-supervised face recognition with LDA self-training," in *2011 18th IEEE International Conference on Image Processing*. Brussels, Belgium: IEEE, Sep. 2011, pp. 3041–3044. [Online]. Available: http://ieeexplore.ieee.org/document/6116305/

[19] 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

[20] J. Tanha, M. van Someren, and H. Afsarmanesh, "Semi-supervised self-training for decision tree classifiers," *International Journal of Machine Learning and Cybernetics*, vol. 8, no. 1, pp. 355–370, Feb. 2017. [Online]. Available: http://link.springer.com/10.1007/s13042-015-0328-7

[21] H.-W. Lee, N.-r. Kim, and J.-H. Lee, "Deep Neural Network Self-training Based on Unsupervised Learning and Dropout," *The International Journal of Fuzzy Logic and Intelligent Systems*, vol. 17, no. 1, pp. 1–9, Mar. 2017. [Online]. Available: http://www.ijfis.org/journal/view.html?doi=10.5391/IJFIS.2017.17.1.1

[22] 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

[23] T. Suzuki, J. Kato, Y. Wang, and K. Mase, "Domain Adaptive Action Recognition with Integrated Self-Training and Feature Selection," in *2013 2nd IAPR Asian Conference on Pattern Recognition*. Naha, Japan: IEEE, Nov. 2013, pp. 105–109. [Online]. Available: http://ieeexplore.ieee.org/document/6778291/

[24] I. Triguero, J. A. Sez, J. Luengo, S. Garca, 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

[25] 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

[26] K. M. Vale, A. M. d. P. Canuto, F. L. Gorgônio, A. J. Lucena, C. T. Alves, A. C. Gorgônio, and A. M. Santos, "A data stratification process for instances selection in semi-supervised learning," in *2019 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2019, pp. 1–8.

[27] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, 2011.

[28] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

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