A Highly Accurate Framework for Self-Labeled Semisupervised Classification in Industrial Applications

Di Wu, Xin Luo[®], Senior Member, IEEE, Guoyin Wang, Senior Member, IEEE, Mingsheng Shang, Ye Yuan, and Huyong Yan

Abstract-Self-labeled technique, a paradigm of semisupervised classification (SSC), is highly effective in alleviating the shortage of labeled data in classification tasks via an iterative self-labeling process. Although existing selflabeled SSC models show great prospect in industrial applications, they suffer from performance degeneration caused by false-positive label-predictions of unlabeled data during the iterative self-labeling process. For addressing this issue, this paper proposes a novel SSC framework, which is highly compatible with most existing self-labeled SSC models. The main idea of this framework is to incorporate a differential-evolution-based positioning optimization algorithm for classification into the iterative self-labeling process, aiming at optimizing the positioning of newly labeled data. Specifically, five representative self-labeled SSC models with different characteristics are modified based on the proposed framework to check their performances. Experimental results on 45 benchmark datasets demonstrate that the proposed framework is highly compatible with tested self-labeled SSC models, and significantly effective in improving their performances.

Index Terms—Differential evolution (DE), general framework, industrial application, positioning optimization, selflabeled, semi-supervised classification (SSC).

Manuscript received June 15, 2017; accepted July 25, 2017. Date of publication August 9, 2017; date of current version March 1, 2018. This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFC0804002, in part by the National Natural Science Foundation of China under Grant 61702475, Grant 61772493, Grant 91646114, Grant 61602434, and Grant 51609229, and in part by the Pioneer Hundred Talents Program of Chinese Academy of Sciences. Paper no. TII-17-1273. (Di Wu and Xin Luo contributed equally to this work.) (Corresponding authors: Guoyin Wang and Mingsheng Shang.)

D. Wu and Y. Yuan are with the Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China, and also with the University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wudi@cigit.ac.cn; yuanye@cigit.ac.cn).

X. Luo, G. Wang, and M. Shang are with the Chongqing Key Laboratory of Big Data and Intelligent Computing, Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing 400714, China (e-mail: luoxin21@cigit.ac.cn; wanggy@ieee.org; msshang@cigit.ac.cn).

H. Yan is with the School of Computer Science and Information Engineering, Chongqing Technology and Business University, Chongqing 400067, China (e-mail: yanhuyong@cigit.ac.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TII.2017.2737827

I. INTRODUCTION

► LASSIFICATION is an active research problem in data mining and machine learning communities [1]. So far, classification has been widely and frequently used in industrial applications, including fault diagnosis, mineral resources exploration, face recognition, power system security, etc. [2]–[5]. Classification tasks rely heavily on the training data with class labels to achieve an efficient classifier, which can accurately assign labels to previously unseen data. However, it is difficult to obtain sufficient labeled data to build an efficient classifier because of the technical support from experts and the long time consumption of manual data labeling. In addition, data without labels are often abundant in real applications. Under such circumstances, traditional classification methods often fail to learn an appropriate classifier relying on labeled data only [6]. As a result, there is a rapidly growing demand for semi-supervised classification (SSC) in industrial applications because it is a learning paradigm aiming at improving the performances of traditional classifiers by using the abundant unlabeled data [7], [8]. Hence, SSC is a suitable data processing method for industrial applications.

SSC has been deeply investigated by prior researchers owing to its effectiveness in alleviating the shortage of labeled data. According to the development of SSC approaches, they can generally fall into four categories:

- Generative models [9], [10]: These models are able to incorporate unlabeled data to extend supervised generative models and use techniques such as the expectation maximization algorithm to estimate model parameters and labels.
- Graph-based models [11]–[13]: These models are able to map both the labeled and unlabeled data into a unique graph, where the unlabeled data are assigned class labels based on its topology characteristics.
- 3) Semisupervised support vector machines (SVMs) [14], [15]: They are able to maximize the margin between labeled and unlabeled data by adapting to the hyper plane of SVM and the labels of unlabeled data.
- Self-labeled models: These models are able to take advantage of a supervised classifier to label data automatically without making any specific hypothesis about input data
 Self-labeled models can be further divided into two

1551-3203 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

groups, i.e., self-training methods (STMs) [16]–[19] and cotraining methods (CTMs) [20]–[26].

It is well known that SSC models exploit unlabeled data to enhance the final performance of a classifier, particularly when the number of labeled instances is low [7]. However, they are restricted by the number of labeled instances and their distribution as well as the outliers. As a result, cases that unlabeled data degrade the performance of classifiers with SSC are frequently discussed in prior research works [27]–[31].

For self-labeled models, if unlabeled data are predicted with false-positive labels during the iterative self-labeling process, they would be impaired instead of being improved. Hence, it is vital to control the negative effects caused by unlabeled data. In order to address this issue, two fundamental methods, i.e., the data editing based methods [16], [32]-[35] and disagreement mechanism maintain based methods [21], [25], [36] are proposed. However, these methods may become ineffective under some circumstances. For instance, for the data editing-based methods, they employ specifically designed strategy such as nearest neighbor rule-based data editing method [32] to control the performance degeneration caused by false-positive labelpredictions of unlabeled data, which cannot be easily extended to other self-labeled SSC models. Additionally, for the synthetic data-based method [36], the problem of false-positive label-predictions of unlabeled data may still exist in the iterative self-labeling process whilst the additional synthetic data lead to the increase of computational burden.

Differential evolution (DE), which is an evolutionary algorithm, is simple to implement, reliable, and fast. Meanwhile, it is significantly efficient and robust in solving global numerical optimization problems [37]. Therefore, it has been widely used in industrial compressor supply system [38], medical image registration [38], data reduction [39], etc. Recently, it has been successfully used for optimizing the positioning of data [40]. Here, optimizing the positioning of data means optimally adjusting the attributes values of data. Subsequently, we find that the controlling of negative effects caused by unlabeled data for self-labeled SSC models can also be viewed as a positioning optimization problem of data. Thus, we expect that DE is a suitable method to achieve the controlling for self-labeled SSC models.

This work aims to construct a general framework, which is highly compatible with any self-labeled SSC model, to alleviate the weaknesses caused by unlabeled data during the iterative self-labeling process. To do so, we innovatively propose to incorporate a DE-based positioning optimization algorithm for classification (DE-POAC) into the iterative self-labeling process, aiming at optimizing the positioning of newly labeled data before they are added into the labeled set. Here, the newly labeled data means unlabeled data labeled by classifiers during the self-labeling process and optimizing the positioning means optimally adjusting the attributes values. We name the proposed framework as DE-SSC. Five representative self-labeled SSC models with significantly different characteristics are modified to check the performances of this novel DE-SSC framework. Detailed experiments on 45 benchmark datasets extracted from the University of California Irvine (UCI) [41], knowledge extraction evolutionary learning (KEEL) [42], and book [43] repositories are carried out to compare the performances between the original models and their modified versions based on DE-SSC. The compared results clearly validate the efficiency of the DE-SSC in controlling unlabeled data for self-labeled SSC models to achieve strong generalization ability. The main contributions of this work include the following.

- DE-SSC framework, which is able to alleviate the negative effects brought by unlabeled data during the iterative self-labeling process and compatible with most of the existing self-labeled SSC models.
- 2) Detailed algorithm design and analysis for DE-SSC.
- Detailed empirical studies conducted on 45 benchmark classification datasets, along with analyses regarding the experimental results.

To the authors' best knowledge, such efforts have never been seen in any prior work.

The rest of the paper is organized as follows: Section II introduces the SSC problem. Section III presents the DE-SSC framework. Section IV provides and discusses the experimental results. Finally, Section V concludes this paper.

II. SELF-LABELED SEMISUPERVISED CLASSIFICATION

A. Semisupervised Classification

In SSC, a sample of data is described with a *D*dimensional vector of attributes plus on class label as $X_i = (X_i^1, X_i^2, \ldots, X_i^d, \ldots, X_i^D, \omega)$, where X_i is the *i*th instance of all the data, $d \in \{1, 2, \ldots, D\}$ is the *d*th feature of X_i , and ω indicates that X_i belongs to a class ω . *L* is the labeled set with ω known and consists of N_L instances. *U* is the unlabeled set with ω unknown and consists of N_U instances. Particularly, for a typical SSC problem, $L \cup U$ forms the training set T_R under the condition of $N_U >> N_L$. In addition, there are some unseen data which have the same characteristics as X_i with ω unknown to form the testing set T_S . The purpose of SSC is to learn a better classifier *C* by using T_R instead of only using *L* to predict the class labels of unlabeled data (transductive, *U*) or unseen data (inductive, T_S).

B. Self-Labeled SSC Models: Related Work

Self-labeled models are effective in solving SSC problems [7]. They can be divided into two groups, i.e., STMs [16]–[19] and CTMs [20]–[26]. In STMs, a classifier keeps on labeling unlabeled data and retraining itself on an enlarged L iteratively. STMs have been successfully applied to many industrial applications such as word sense disambiguation and subjective nouns [7], [17]. However, STMs may label some unlabeled data with false-positive label-predictions in some cases, which will degrade the performance of the ultimate classifier. In order to deal with this issue, Li and Zhou utilized a specific data editing method based on the local-cut-edge-weight statistic to identify the unlabeled data with false-positive label-predictions [16].

On the other hand, in CTMs, the attributes of data are split into two conditionally independent views to train a single classifier, respectively. The two classifiers cooperate to predict the labels of unlabeled data [20]. Advanced approaches, which do not require explicit attributes splits, also were proposed by some researchers [22], [24], [26]. Unfortunately, the problem of false-positive label-predictions may still exist in the iterative self-labeling process of CTMs in some cases [32]–[35]. Deng and Guo [32] first incorporated a nearest neighbor rule based data editing method into tri-training algorithm [23] to discard or correct the noise or unlabeled data with false-positive label-predictions. After that, they developed their proposed algorithm by simultaneously exploiting an effective data editing technique to identify possible unlabeled data with false-positive label-predictions and an adaptive strategy to control the validity of editing operation [33]. In addition, they used the same technology to achieve a new coforest algorithm [24] with adaptive data editing to improve the performance of original proposal [34]. In fact, the key strategy of CTMs is that a large disagreement between base classifiers must be maintained during the iterative self-labeling process [25]. Zhou et al. empirically analyzed the mechanism [21] and proposed a new CTMs style algorithm based on data editing. In details, data editing techniques are incorporated into the iterative self-labeling process to improve the quality of the T_R by identifying and eliminating unlabeled data with false-positive label-predictions [35].

Recently, Triguero *et al.* proposed a framework for selflabeled SSC models to improve the classification performance by incorporating the synthetic labeled data into the iterative selflabeling process [36]. In fact, the purpose of using synthetic labeled data is to introduce diversity into multiple classifiers and fulfill the distribution of labeled data.

III. DE-SSC FRAMEWORK

A. DE-SSC

Although each self-labeled SSC model works in a different way, they are very similar or at least share some operations. Thus, the proposed DE-SSC framework is designed to be as flexible as possible for any self-labeled SSC model. Fig. 1 depicts the flowchart of the DE-SSC framework.

The first step is initialization. There are some objects, including initial L, iteration stopping criteria, and optimized parameters, must be initialized at the beginning. The second step is the iterative self-labeling process, which is vital for a selflabeled SSC model. Generally, STMs employ one classifier and CTMs employ two or more than two classifiers to predict the class labels of unlabeled data. Thus, they can be generalized as using a set of n classifiers C_k , $k \in \{1, n\}$, to achieve the predictions. Each C_k is trained on the initial L. Meanwhile, different self-labeled methods adopt different strategies to select the high-confidence unlabeled data from U. Then, the learned n classifiers of C_k are exploited to label the selected unlabeled data. After that, a specific and different strategy is used by different self-labeled methods to determine high-confidence unlabeled data as newly labeled data to form a newly labeled set L'. However, L' may contain some unlabeled data with falsepositive label-predictions because of the small initial number of labeled instances and their unrepresentative distributions as well as the existence of outliers. If L' is directly added into L, it will bring negative influences to the iterative self-labeling



Fig. 1. Flowchart of the DE-SSC framework for any self-labeled SSC model.

process. Thus, the function DE-POAC (L', L) derived from the DE-POAC algorithm is employed to optimally adjust the attributes values for each instance of L' under the supervision of initial L. After that, we obtain an optimized L' and then add it into L to enlarge the labeled data. The above operations of the second step are repeated until satisfying the stopping criteria which depend on the specific self-labeled SSC method. Finally, the third step will output the final n classifiers of C_k which are learned from the enlarged L. The final n classifiers of C_k then can be applied to classify the unlabeled data (transductive, U) or unseen data (inductive, T_S).

B. DE-POAC Algorithm

Next, we present the function DE-POAC (L', L) in detail first and then conclude the pseudocode of the DE-POAC algorithm.

The function DE-POAC (L', L) starts with an initial dataset L' with N instances, where each one is called a "target vector" with a D-dimensional attributes plus on class label as

$$X_{i,g} = \left(\left[X_{i,g}^1, \ X_{i,g}^2, \ \dots, \ X_{i,g}^d, \ \dots, \ X_{i,g}^D \right], \ \omega \right)$$
(1)

where $i \in \{1, 2, ..., N\}$ stands for the *i*th instance, $d \in \{1, 2, ..., D\}$ stands for the *d*th attribute, ω indicates that $X_{i,g}$ belongs to a specified class ω , and $g \in \{1, 2, ..., G\}$ stands for the *g*th generation with *G* denoting the generation upper bound, respectively. Given the dataset *L*, where each instance has the same size as $X_{i,g}$, the function DE-POAC (L', L) aims

at obtaining an improved dataset L' by adjusting each $X'_{i,g}$ s attributes values under the supervision of L. The function DE-POAC (L', L) mainly relies on three operators, i.e., mutation, crossover, and selection. It is noted that the appropriate strategies and its associated parameters are important to the three operators. Triguero *et al.* [40] have systematically studied the DE to optimally adjust the attributes values or positioning of data. Thus, we choose the recommended strategies and its associated parameters as discussed in [40] for the function DE-POAC (L', L).

Mutation: It is the prime operator of function DE-POAC (L', L). It generates a mutant vector $V_{i,g}$ for each target vector $X_{i,g}$ at the current generation g. Among the frequently mentioned mutation strategies, we choose the DE/ Rand/ 1 owing to its popularity and robustness [39]. DE/ Rand/ 1 generates the mutant vector $V_{i,g}$ as follows:

$$V_{i,g} = X_{r_1,L} + F_i \cdot (X_{r_2,L} - X_{r_3,L})$$
(2)

where F_i is the scaling factor which positively controls the scaling of different vectors; $X_{r_1,L}$, $X_{r_2,L}$, and $X_{r_3,L}$ are randomly selected from L and belong to the same class as $X_{i,g}$. When there are not enough instances having the same class label as $X_{i,g}$ in L, we artificially generate the necessary number of new instances $X_{r_m,L}$, $m \in \{1, 2, 3\}$, with the same class label as $X_{i,g}$ as follows:

$$X_{r_m,L} = ([X_{i,g}^1 + \operatorname{rand}[-0.1, 0.1], X_{i,g}^2 + \operatorname{rand}[-0.1, 0.1], \dots, X_{i,g}^d + \operatorname{rand}[-0.1, 0.1]], \dots, X_{i,g}^D + \operatorname{rand}[-0.1, 0.1]], \omega)$$
(3)

where rand [-0.1, 0.1] denotes picking a random real number from the [-0.1, 0.1] interval.

Crossover: The target vector $X_{i,g}$ and its corresponding mutant vector $V_{i,g}$ are subject to the crossover operation to generate a new trial vector $U_{i,g}$. There are three kinds of crossover schemes, i.e., binominal, exponential, and arithmetic, and we focus on the arithmetic crossover. Concretely, the DE/ CurrentToRand/ 1 strategy [44] is used to generate the trial vector $U_{i,g}$, which linearly combines the target vector $X_{i,g}$ and the corresponding mutant vector $V_{i,g}$ as follows:

$$U_{i,g} = X_{i,g} + K \cdot (V_{i,g} - X_{i,g})$$
(4)

where $K \in [0, 1]$ is a random number. Next, let us bring formula (2) into formula (4) and then simplify it

$$U_{i,g} = X_{i,g} + K \cdot (X_{r_1,L} - X_{i,g}) + F_i \cdot (X_{r_2,L} - X_{r_3,L}).$$
(5)

It is noted that formula (5) only produces modifications in the attributes of data. Thus, each trial vector $U_{i,g}$ has the same class label as each corresponding target vector $X_{i,g}$ throughout the mutation and crossover operators. However, we should check whether there have been values of $U_{i,g}^d$ that are out of range $[\min(X_{i,g}^d), \max(X_{i,g}^d)]$. If a computed value of $U_{i,g}^d$ is greater than $\max(X_{i,g}^d)$, then we set it as $\max(X_{i,g}^d)$, else if it is lower than $\min(X_{i,g}^d)$, then we set it as $\min(X_{i,g}^d)$.

Selection: After mutation and crossover operators are applied to all the target vectors $X_{i,g}$ of L', a dataset that consists of N number of trial vectors $U_{i,g}$ is obtained. We represent it as L'_{DE} . The selection operator decides which dataset of L' or L'_{DE} should survive in the next generation g + 1:

$$L' = \begin{cases} L'_{\text{DE}} & \text{if accuracy } (L'_{\text{DE}}) >= \text{accuracy } (L'); \\ L' & \text{otherwise.} \end{cases}$$
(6)

In formula (6), any classification algorithm, likes *K*-nearest neighbor (KNN) [45] or SVM [46], can be used to evaluate the classification accuracy. As a result, the performance of the next generation g + 1 is always better than, or at least same as, the current generation g. Finally, an improved dataset L' is obtained after applying the function DE-POAC (L', L).

From formula (5), we note that the setting of scaling factor F_i decides the convergence speed and the optimal performance of function DE-POAC (L', L). Neri and Tirronen proposed a scale factor local search in DE (SFLSDE) algorithm to self-adaptively set the value of scaling factor F_i for DE [47]. Triguero *et al.* [39], [40] have exploited SFLSDE to obtain a valuable result. In order to guarantee a high-quality DE-POAC, we also employ the SFLSDE to determine the value of scaling factor F_i as follows:

$$F_{i} = \begin{cases} \text{SFGSS} & \text{if } \text{rand}_{3} < \tau_{2}; \\ \text{SFHC} & \text{if } \tau_{2} \leq \text{rand}_{3} < \tau_{3}; \\ F_{l} + F_{u} \cdot \text{rand}_{1} & \text{if } \text{rand}_{2} < \tau_{1} \text{ and } \text{rand}_{3} > \tau_{3} \\ F_{l} + F_{u} \cdot \text{rand}_{1} & \text{if } \text{rand}_{2} \geq \tau_{1} \text{ and } \text{rand}_{3} > \tau_{3} \end{cases}$$
(7)

where rand₁, rand₂, and rand₃ are uniform pseudorandom numbers between 0 and 1; τ_1 , τ_2 , and τ_3 are constant threshold values. According to [47], all these parameters in formula (7) are recommended to set as SFGSS = 8, SFHC = 20, F_l = 0.1, $F_u = 0.9$, $\tau_1 = 0.1$, $\tau_2 = 0.03$, $\tau_3 = 0.07$, and F_i is initialized to a random value between 0 and 1.

Obviously, the function DE-POAC (L', L) means that the original dataset L' is optimized by the DE-POAC algorithm under the supervision of L. Thus, the pseudocode of DE-POAC algorithm can be concluded in Table I, where function Learn (C, L') means learning a classifier C based on dataset L' and function Evaluate(C, L) means evaluating the classification accuracy of classifier C on the dataset L. For analyzing the computational complexity of the DE-POAC algorithm, let $\Theta(C)$ denotes the computational cost of classifier C. As analyzed in Table I, the computational complexity of the DE-POAC algorithm is

$$T = \Theta(C) + G \times N \times \Theta(C) = \Theta(C) \times (1 + G \times N)$$
$$= \Theta(C) \times G \times N = \Theta(C \times G \times N).$$
(8)

From formula (8), we obtain that the classifier C and the maximum subsequent generations G play key roles in deciding the computational complexity of the DE-POAC algorithm.

IV. EXPERIMENTS AND RESULTS

A. Compared Algorithms With Parameter Settings

As shown in Fig. 1, DE-SSC is compatible with any selflabeled SSC model. In order to test its performance, five representative self-labeled SSC algorithms, i.e., self-training,

TABLE I ALGORITHM DE-POAC

Input: L' , L , Learn, G	
Code:	Cost
$\overline{1. \ g = 0}$	$\Theta(1)$
2. $Learn(C, L')$	$\Theta(C)$
3. $Accuracy_{L'} = Evaluate(C, L)$	
4. while $Accuracy_{L'} < 1$ and $g < G$ do	$\times G_{\max}$
5. for $i = 1$ to N do (N instances of L')	$\times N$
6. compute F_i by formula (7)	$\Theta(1)$
7. generate $U_{i,g}$ by formula (5)	$\Theta(1)$
8. $L'_{\rm DE}(i) = U_{i,g}$	$\Theta(1)$
9. end for	_
10. Learn (C', L'_{DE})	$\Theta(C)$
11. $Accuracy_{L'_{DD}}^{DD} = Evaluate(C', L)$	
12. if $Accuracy_{L'_{DE}}^{DE} > Accuracy_{L'}$	$\Theta(1)$
13. $Accuracy_{L'} = Accuracy_{L'_{\text{pr}}}$	$\Theta(1)$
14. $L' = L'_{\text{DE}}$	$\Theta(1)$
15. end if	_
16. $g = g + 1$	$\Theta(1)$
17. end while	_
18. return <i>L</i> ′	_
Output: Optimized dataset L'	

tri-training [23], coforest [24], SSFCM-SVM [19], and SEG-SSC [36] are chosen to integrate into it. The five algorithms have different characteristics, including initialization schemes, learning strategies, and selection mechanisms of determining high-confidence unlabeled data. A brief description about the five self-labeled SSC algorithms and their associated base classifiers are given below.

- KNN: It is an instance-based classification learning algorithm and belongs to the lazy learning family. The function is only locally approximate and all computation is deferred until classification [45]. As a result, it commonly relies on the Euclidean distance between a test sample and the specified training samples without building a model during the learning process. For self-labeled SSC models, they can be incorporated into the iterative self-labeling process to play the role of base classifiers.
- 2) SVM: It is a powerful machine learning technique with excellent generalization performance based on the principle of structural risk minimization. It can solve linearly nonseparable problems by using kernel tricks [46]. For self-labeled SSC models, it can also be incorporated into the iterative self-labeling process to play the role of base classifiers. We will choose the library for support vector machines [48] to achieve the SVM classifiers in the experiments.
- 3) *Self-training:* We design it according to the principle of STMs. In self-training process, the unlabeled data, which are nearest to the labeled data, are selected to be labeled by a classifier. After that, the classifier is retrained on an enlarged labeled set. Each kind of classifier can be used as the base classifier in this algorithm.
- 4) *Tri-training [23]:* It is a new CTMs style SSC algorithm, which generates three classifiers from the initial labeled set. These classifiers are refined by using unlabeled data

 TABLE II

 DESCRIPTIONS OF ALL THE PARAMETERS USED IN THE EXPERIMENTS

Mark	Algorithm	Parameters
/	KNN	Number of neighbors $= 3$; Euclidean distance.
/	SVM	LIBSVM: all the parameters are set as default values.
M1	Self-	Confidence rate = 0.2 , which is computed by: (the
	training	number of selected unlabeled instances that are nearest
		to the labeled instances) / (the number of unlabeled
		instances); Euclidean distance; Base classifiers: SVM
		or KNN.
M2	Tri-	Base classifiers: SVM or KNN.
	training	
M3	Coforest	Number of Fandom Forest classifiers = 6; Threshold =
		0.75.
M4	SSFCM-	Threshold $\varepsilon_1 = 1/$ (the number of classes).
	SVM	
M5	SEG-SSC	Oversampling factor = 0.01; Base SSC model: Coforest
		with the same parameters as M3.
/	DE-SSC	Iteration $= 100;$
		$SFGSS = 8; SFHC = 20; F_l = 0.1; F_u =$
		0.9; $\tau_1 = 0.1$; $\tau_2 = 0.03$; $\tau_3 = 0.07$; $F_i = a$
		random value between 0 and 1.

in the tri-training process. Tri-training requires neither sufficient and redundant views nor different supervised learning algorithms. The interchange of the base classifier is allowed in tri-training.

- 5) Coforest [24]: It extends CTMs by using a well-known ensemble method named random forest [49], which enables coforest to estimate the labeling confidence of unlabeled data and easily produce the final hypothesis. It requires neither the data described by sufficient and redundant attribute subsets nor special learning algorithms that frequently employ time-consuming cross validation in learning. The random tree is used as the base classifier in this algorithm.
- 6) SSFCM-SVM [19]: It is an STMs style SSC algorithm that combines clustering and classification. In particular, a semi-supervised fuzzy c-means algorithm is integrated into the self-training process to train a better classifier. The SVM is used as the base classifier in SSFCM-SVM.
- 7) SEG-SSC [36]: This is a framework which is designed to improve the performance of a self-labeled SSC model based on synthetic examples generation. The purpose of using synthetic examples is to diminish the drawbacks occasioned by the absence of labeled data. In order to read easier, the names of these self-labeled SSC models are simplified as shown in Table II. In experiments, we modify these self-labeled SSC models by incorporating the DE-POAC algorithm into their iterative self-labeling process to achieve the DE-SSC framework and name them as DE-SSC+M1, DE-SSC+M2, DE-SSC+M3, DE-SSC+M4, and DE-SSC+M5, respectively. We will choose the recommended values for all the parameters according to the original papers. To show how the DE-SSC framework can improve the performances of self-labeled SSC models, all the experimental conditions and parameters are same in both the original models and their modified versions. Table II summarizes all the pa-

TABLE IIIPROPERTIES OF ALL THE DATASETS*

ID	Name	N.	D.	C.	ID	Name	N.	D.	C.
D1	appendicitis	106	7	2	D24	hepatitis	80	19	2
D2	australian	690	14	2	D25	housevotes	232	16	2
D3	automobile	158	25	6	D26	ionosphere	351	33	2
D4	banana	5300	2	2	D27	iris	150	4	3
D5	banknote authentic	1372	4	2	D28	led7digit	500	7	10
D6	bci	400	117	2	D29	mammographic	961	5	2
D7	breast	286	9	2	D30	monks	432	6	2
D8	bupa	345	6	2	D31	pima	768	8	2
D9	cleveland	297	13	5	D32	saheart	462	9	2
D10	coil	1500	241	6	D33	segment	2310	19	7
D11	coil2	1500	241	2	D34	sonar	208	60	2
D12	contraceptive	1473	9	3	D35	spectheart	267	44	2
D13	crx	653	15	2	D36	tae	151	5	3
D14	dermatology	358	33	6	D37	text	1500	11960	2
D15	diabetic	1151	19	2	D38	titanic	2201	3	2
	retinopathy debrecen								
D16	digit1	1500	241	2	D39	usps	1500	241	2
D17	ecoli	336	7	8	D40	vehicle	846	18	4
D18	flaresolar	1066	9	2	D41	vowel	990	13	11
D19	glass	214	9	7	D42	waveform	5000	21	3
D20	g241c	1500	241	2	D43	wine	178	13	3
D21	g241n	1500	241	2	D44	wisconsin	683	9	2
D22	haberman	306	3	2	D45	yeast	1484	8	10
D23	heart	270	13	2		-			

^{*}Note that the column label N. stands for instance count, D. stands for dimension count, and C. stands for class count, respectively.

rameters used in these models. In addition, the parameters involved in the DE-SSC framework, such as the number of optimization iterations and the parameters needed for setting scaling factor F_i , are also presented in Table II.

B. Datasets

Forty-five benchmark classification datasets are selected to validate the effectiveness of the DE-SSC framework. These datasets are extracted from the UCI [41] and KEEL [42] repositories and can be easily downloaded together from http://sci2s.ugr.es/keel/datasets.php. In addition, for highdimensional problems, eight high-dimensional datasets extracted from the book written by Chapelle et al. [43] are also contained in the experiments, and they can be downloaded from http://olivier.chapelle.cc/ssl-book/benchmarks. html. These datasets contain binary and multiclass classification problems. The number of instances ranges from 80 to 5300, while the number of attributes varies between 2 and 11 960 among these datasets. Most of them are real data and collected from the mining industry, auto industry, financial industry, healthcare industry, etc. Table III summarizes the properties of all the datasets.

C. Comparisons of Classification Accuracy

In the experimental phase, we use the ten-fold cross-validation strategy to obtain the final experimental results. First, each dataset is split into ten folds, each of which contains 10% of the instances. Then, nine folds are selected to use as the T_R

and the remaining one forms the T_S . After that, T_R is divided into labeled part L and unlabeled part U by using a random stratified selection. That means the selected number of instances for each class is proportional to the number of them in the T_R . In addition, we will ensure that at least one representative instance of each class is selected in L. Thus, each dataset is divided into three parts: L, U, and T_S (L and U form T_R). Subsequently, the five original self-labeled SSC models and their modified versions are trained on T_R , and then are tested on U (transductive) and T_S (inductive). Since these models have some random operations during the training process, they will be carried out three times. The above steps will be executed ten times to ensure that each fold can serve as the T_S once.

Specifically, an initial labeled ratio of 10% is adopted over all datasets in the experiments. For M1 and M2, two kinds of base classifiers, KNN and SVM, are used respectively. In order to eliminate the influence of attributes with different range domains, we normalize all attributes of all the datasets to the range [0, 1]. The obtained results of accuracy (mean \pm standard deviation) on transductive setting and inductive setting are shown in Tables IV and V respectively, where modified models perform better than or at least same as their original versions are highlighted in bold.

According to Tables IV and V, we observe that all of five modified models based on the DE-SSC framework generally achieve better results than their original versions, respectively. That means the applications of the DE-SSC framework have improved the generalization capacity of the models. In order to examine the significant differences between the original methods and their modified versions, we perform nonparametric pairwise comparison procedures to conduct the statistical analysis. Concretely, the Wilcoxon signed-ranks test [50] is applied to achieve it. Tables VI and VII record the statistical results of accuracy on transductive and inductive settings, respectively. In Tables VI and VII, there are two columns, which show the achieved rankings R+ and R- values and its associate p-value for the mean and the standard deviation of accuracy, respectively. We have checked whether the modified versions are able to outperform the original versions under the condition of the significance level $\alpha = 0.1$ and the accepted hypotheses are highlighted.

According to Tables IV–VII, there are several observations can be concluded as follows.

1) In Tables IV–V, there are a few cases that the modified models do not improve the performances of original models. Two major reasons can be supposed from this phenomenon. First, the random selection process in the realization of algorithm or the initial labeled set is sufficient for the original models to learn. Second, there are some unlabeled data with false-positive label-predictions during the iterative self-labeling process in those cases. However, regarding to most datasets, each modified model achieves a better accuracy than its original model on transductive and inductive settings. The above observations validate the fact that the DE-SSC framework has improved the performance of the original models.

TABLE IV ACCURACY (MEAN \pm STANDARD DEVIATION, %) OF COMPARISON MODELS IN TRANSDUCTIVE PHASE

							<u>n</u>							
Data.	1.		Or	riginal mod	els				. N	10dified mo	odels based	on DE-SS	С	
sets	M1	M1	M2	M2	M3	M4	M5	DE-SSC+M1	DE-SSC+M1	DE-SSC+M2	DE-SSC+M2	DE-SSC+M3	DE-SSC+M4	DE-SSC+M5
3013	(KNN)	(SVM)	(KNN)	(SVM)		ļ'		(KNN)	(SVM)	(KNN)	(SVM)			[]
D1	80.5±0.98	80.19±0.32	81.38±2.18	80.19±0.32	80.64±3.79	80.93±2.29	81.24±3.28	80.71±1.63	80.19±0.32	81.41±2.11	80.19±0.32	80.99±4.31	81.09±4.39	81.98±8.39
D2	83.64±1.61	85.22±0.93	83.01±2.71	85.66±0.44	82.17±2.86	84.99±1.27	74.57±7.78	82.81±1.47	85.44±0.46	84.02±1.07	85.66±0.42	79.9±4.18	85.63±3.55	77.18±6.7
D3	40.14±6.31	32.37±0.61	40.83±3.37	30.03±2.25	35.08±4.63	23.61±6.4	39.08±10.62	41.18±5.67	32.37±0.61	40.96±2.52	32.51±4.56	35.95±4.38	21.86±10.59	40.7±5.78
D4	76.46±6.83	55.17±0.18	77.98±6.43	55.45±0.83	76.3±3.77	57.84±2.52	69.86±5.12	73.67±6.75	55.17±0.18	77.96±6.44	55.38±0.75	76.62±3.53	57.85 ± 2.61	70.41±4.81
D5	98.4±0.99	98.81±0.19	91.24±2.24	96.65±1.03	93.37±2.38	98.46±0.54	93.12±3.73	$97.33{\pm}0.99$	98.5±0.42	98.78±0.63	97.21±1.53	97.39±0.88	$97.64{\pm}1.47$	96.39±1.73
D6	51±2.8	50±0	55.17±1.74	49.98±0.06	52.45±1.49	48.78±1.99	52.55±1.47	55.56±2.4	50±0	55.67±1.08	50.13±0.4	53.46±2.24	50.17 ± 4.48	51.94±1.35
D7	69.63±3.05	70.76±0.13	72.91±2.3	70.93±0.54	71.68±2.44	71.36±1.42	71.13±1.67	71.11±2.29	70.76±0.13	72.8±2.2	71.11±1.08	72.03±1.44	70.3±3.2	72.75±0.97
D8	52.78±6.03	57.97±0.1	55.34±3.65	57.38±1.69	54.76±4.56	56.39±3.78	55.24±4.72	54.95±6.06	57.97±0.1	55.17±3.93	57.43±1.67	52.97±6.97	57.89±6.81	55.4±5.94
D9	56.84±3.04	53.87±0.36	55.85±2.64	54.02±0.49	56.49±1.42	55.26±1.97	56.6±1.34	57.7±2.67	53.87±0.36	56.36±1.36	54.1±0.78	57.22±1.94	53.6±3.99	57.8±1.02
D10	74.41±4.17	30.39±5.44	75.55±3.93	27.91±1.73	69.73±1.63	27.88±4.89	70.09±6.64	79.06±5.14	27.64±3.05	75.49±3.81	26.42±2.87	67.57±2.81	37.67±9.49	65.95±4.03
D11	83.52±2.74	53.31±3.93	84.67±2	58.28±4.43	74.63±3.22	58.84±3.87	79.22±3.74	87.3±2.59	57.31±8.87	84.51±1.95	57.95±3.7	75.82±3.67	72.93±6.43	75.28±3.67
D12	44.3±1.81	43.02±0.32	46.29±1.52	43.51±1.17	49.65±2.4	44.53±1.4	49.86±2.59	45.39±1.88	43.02±0.32	46.34±1.45	43.24±1.05	49.23±2.06	43.81±1.95	49.48±1.77
D13	83.15±2.08	85.06±4.39	81.75±2	86.14±1.47	78.13±6.9	85.72±1.7	76.83±10.15	80.62±3.42	86.16±1.31	82.13±2.04	86.06±1.39	78.77±6.38	83.58±4.71	79.01±8.62
D14	89.26±3.75	56.11±12.79	87.71±3.3	61.19 ± 3.96	89.24±1.9	67.2±12.66	90.46±1.46	90.36±3.56	60.12±12.03	88.83±2.68	60.73±4.46	90.19±2.49	68.43±11.35	89.14±2.23
D15	51.62 ± 5.63	52.47±1.95	53.9±4.53	54.14±1.57	57.89±3.45	54.95±2.75	56.76±1.94	54.52±3.89	52.47±1.95	54.03±4.72	54.15±1.83	57.86±3.49	53.69±1.91	57.45±1.5
D16	92.24 ± 2.12	51.96 ± 2.82	92.1±2.32	75.12±5.57	88.06±1.63	90.15±2.35	82.8±7.59	94.59±1.02	56.04±12.8	96.71±0.75	76.95±9.16	89.53±2.33	93.13±1.31	84.95±7.61
D17	63.06±6.97	42.69 ± 0.72	66±5.53	46.32±3.61	69.72 ± 4.78	60.15 ± 3.21	66.49±5.4	63.53±5.82	42.69±0.72	66.19±5.82	45.45±2.98	68.79±5.19	40.07±28.39	66.06±6.83
D18	63 38+3 34	33 5+1 97	61 23+3 51	49 06+2 78	71 48+1 91	62 11+4 83	71 65+2 52	62.32±3.27	33.26±1.93	61.24±3.5	46.54±3.33	71.66±3.51	43.59 ± 21.42	71.71±3.03
D19	31 76+14 25	33 44+3 03	40.82+6.9	37 15+4 62	40 19+5 66	41 38+5 17	43 19+9 06	33.69±12.77	33.03±1.73	42.34±5.91	40.18±3.4	39.7±8.83	42.2±9.67	41.66±6.75
D20	58 19+2 6	50+0	55 66+1 95	50+0	58 38+1	40.36+13.66	56 46+1 57	65.52±1.13	50±0	60.33 ± 2.12	50.75±2.38	59.54±0.89	68.69±13.31	56.37±1.68
D20	58 62+2 46	50 13+0 02	58 82+3 13	50.03+0.00	57 54+1 22	40.30±13.00	55 55+1 77	65.11±1.67	50.13+0.02	60.25+4.17	50±0.09	57.33 ± 1.12	49.87±0.17	54 85±1 45
D21	40.00±4.65	7257 ± 0.02	40 85±2 11	72 57±0 72	72 50±0 71	49.07±0.02 72.57±0.72	73531011	71 19+2 47	72 57+0 73	70 15+3 38	72 57+0 73	72 45+0 94	73 58+6 23	73 53+0 11
D22	76 42+7 1	74 77+8 48	76 7217 58	77 53±4 40	70.27±7.04	47 28+22 31	70.1±3.04	74 8+4 11	76 31+8 49	77 83+3 01	75 72+6 37	72 13+6 45	75 56+10 98	71 4+5 85
D25 D24	/0.42±/.1	92 75±0 53	0.23±2.30	02 75±0 52	0.2/±/.24	07.20±22.31	0.1±3.54	93 75+0 53	93 75+0 53	83 04+0 73	83 75+0 53	93 76+0 77	84 05+4 4	94 45+2 63
D24	83.75±0.55	83.75±0.55	83.73±0.33	83.75±0.55	00.18±2.14	00.22±1.91	84.04±1.00	00 1+1 02	80 06+1 50	97 03+1 87	01 2+2 10	01 10+2 4	89.97+5.15	99.85+7.36
D25	89.05±1.09	88.94±1.02	87.31±2.00	90.77 ± 1.20	90.18±2.14	90.32±1.01	88.43±3.72	70 07+7 14	65.70±1.37 44 21±0 63	75 49+7 63	91.2±2.17 64.7±0.70	91.17-4.7	80.25+6.3	87 81+1 17
D20	71.04±0.5	04.31±0.03	/4.94±/.+1	04.4±0.05	83.92±3.40	81.84±0.10	87.98±0.04	0.71+1.17	(1.02)11.07	13.47-1.03	77.710.14	00.00-0.07	00.23±0.3	00 44+3 85
D27	/0.81±9.23	33.33±17.21	8/.14±3.15	72.33±3.10	80.0/±3.0/	64.39±20.12	88.0/±0.10	80.79±8.91	01.83±11.07	88.57±2.00	77.7±0.14	89.03±7.94	79.33±10.03	63 0+4 0
D28	00.04±3.04	34.38±0.39	03./0±0.39	55.59±7.62	04.41±3.70	40.18±9.0	64.03±3.30	50.50±1.11	74.04±0.12	03.72±0.77	57.39±0.00	04.20±2.5	4/.55±10.4	03.7±4.7
D29	/5.5±/./	76.01±9.30	71.72.2.01	79.33±0.04	68.35±15.02	80.12±4.10	6/.//±/.43	09.9/±0.27	/0.18=7.45	/3.40±3.13	/9.59±0.70	00.4/=14.42	70.04+7.22	00.4/±0.51
D30	67.61±4.25	76.5±8.97	71.78±3.81	78.7±4.47	92.18±9.65	78.59±3.80	94.61±2.87	71.05±2.02	80.02=2.50	(9.25+2.22	(1,51) 0.42	91.85±5.10	79.94±7.34	88.03±4.01
D31	65.97±2.91	64.35±0.21	68.12±1.58	64.51±0.45	69.07±3.77	66.95±3.88	66.32±1.08	00.84±1.04	64.35±0.21	68.25±2.22	64.51±0.42	07.03±3.29	00.21±3.79	66./5±2.05
D32	65.85±2.83	65.34±0.09	67.59±3.39	66.12±1.5/	68.68±1.54	67.8±3.03	67.68±1.40	69.11±2.01	65.34±0.09	6/.33±3.24	66.21±1.72	68.62±1.47	68.08±3.93	67.99±1.80
D33	87.68±1.55	66.7±5.8	88.09±1.44	66.63±3.66	84.87±5.29	69.53±3.8	85.23±3.47	8/.0/±1.34	60.9±5.57	88.08±1.37	66.36±3.3	85.11±4.05	73.12±0.52	83.5±2.53
D34	58.59±7.16	53.37±1.57	63.21±3.07	52.46±2.36	62.82±5.23	46.21±5.61	64.75±6.11	63.68±8.27	53.37±1.57	65.66±5.28	51.05±2.94	61.56±3.72	54.97±15.55	64.48±6./
D35	64.05±20.41	79.4±1.07	79.01±1.32	79.4±1.07	78.43±2.23	79.4±1.07	78.83±0.56	67.87±18.42	79.4±1.07	78.8±1.41	79.4±1.07	78.96±1.07	79.35±9.78	79.21±0.82
D36	41.6±5.61	34.06±2.25	37.74±4.14	37.6±2.99	39.97±6.49	27.18±5.47	44.83±4.52	43.66±4.4	34.48±2.13	41.47±5.09	37.21±3.2	43.32±3.96	32.94±16.76	45.06±4.47
D37	75.55±4.51	55.17±0.16	76.78±4.93	55.37±0.67	76.64±4.62	56.61±2.15	76.91±4.39	73.25±4.68	55.17±0.16	76.82±4.88	55.42±0.89	76.66±4.55	56.87±2.59	76.75±4.45
D38	76.47±2.55	77.55±0.21	76.47±2.55	77.58±0.23	77.95±1.06	77.6±0.25	74.88±3.22	76.89±2.37	77.55±0.21	76.47±2.55	77.58±0.23	78.18±0.74	77.33±1.67	74.58±2.63
D39	82.92±1.22	80±0.46	90.07±1.46	80±0.46	81.71±1.95	83.09±5.48	81.59±0.55	87.71±2.38	80±0.46	91.35±0.88	80±0.46	82.48±1.52	86.49±2.85	81.52±0.69
D40	40.58±10.11	26.82±3.72	47.24±9.4	37.17±3.55	49.79±11.33	34.96±5.64	52.49±13.72	46.25±10.88	27.83±4.43	47.15±9.04	35.98±2.04	50.39±12.06	39.47±10.46	51.63±10.38
D41	36±3.18	9.51±0.37	35.75±2.02	13.39±1.95	40.66±5.98	6.67±2.43	47.94±2.64	38.97±3.31	9.51±0.37	35.05±2.16	12.1±1.33	39.98±5.74	16.16±6.29	47.53±3.21
D42	74.8±2.1	80.9±2.06	78.04 ± 2.52	82.43±2.69	77.14±1.37	84.44±1.97	73.96±2.6	75.76±1.29	81.77±1.84	77.92±2.48	82.47±2.7	77.14±0.96	82.26±2.82	75.73±1.27
D43	77.79±14.5	48.27±16.06	65.93±10.88	48.26±7.54	76.89±11.17	52.85±13.98	78.97±13.38	85.25±7.8	50.33±13.07	81.59±8.43	50.98±8.5	80.92±8.24	79.87±19.56	81.83±13.28
D44	94.96±1.33	95.73±0.48	94.64±1.35	96.19±0.63	94.82±1.97	96.16±0.6	94.46±2.39	95.89±0.9	96.08±0.78	96.34±0.91	96.15±0.74	94.78±1.43	96.27±1.85	94.37±3.07
D45	33.39±7.89	31.01±0.08	42.88±7.03	31.07±1.03	37.54±9.53	31.47±13.12	37.92±3.72	33.38±9.28	31.01±0.08	43.02±6.06	31.35±0.93	37.7±10.08	24.07±19.67	39.6±4.35
Aver-	(7.04+4.00	50.001.2.90	(0.25) 2.(2	(1.92) 2.07	(0.7) 4.19	(2.20) 4.01	(0.75+4.19	(0.24) 4.4	50 7412 (0	50 4512 24	(2.02) 2.41	70 11 4 05	(12(17))	(0.0) 4.1
age	67.84±4.88	59.09±2.86	69.25±3.63	61.83±2.07	69./±4.18	62.38±4.81	69.75±4.18	69.34±4.4	59.74±2.69	70.47±3.34	62.03±2.41	70.11±4.05	64.36±7.5	69.9±4.1

2) In Tables VI and VII, we have analyzed the statistical results regarding two aspects: mean of accuracy and standard deviation of accuracy. In the aspect of mean of accuracy, the statistical results accept the hypothesis that each modified model outperforms its original version, respectively, with a significance level $\alpha = 0.1$, except for two cases of M2 with SVM in transductive phase and M5 in inductive phase. Although the two cases have not been significantly improved, we can illustrate that they also achieve higher R+ rankings than their original models, which reflects the fact that they perform slightly better. In another aspect of standard deviation of accuracy, we observe that the cases of M2 with SVM in transductive and inductive phases and M4 in transductive phase have not been improved by the DE-SSC framework. However, it can be found that the modified models have significantly or slightly better performances than their original models

in most cases on both transductive and inductive settings. Thus, we can say that the modified models have better overall robustness than original models. These conclusions obtained from Tables VI and VII further support the previous conclusions derived from Tables IV and V.

3) From Tables IV and V, we also observe that each modified model, respectively, achieves a relative stable improvement of accuracy longitudinally on the different 45 benchmark classification datasets in both transductive and inductive phases. In addition, compared with all the models horizontally, we find that M4 is most beneficial by applying the DE-SSC framework. Especially, for example, on the dataset D17 and in inductive phase, the accuracy of M4 is improved from 40.08 ± 28.46 to 60.92 ± 4.93 , while the accuracy of M3 is improved from 66.08 ± 10.18 to 66.38 ± 8.7 . Thus, we can say that the DE-SSC framework can provide more help for a self-labeled SSC

TABLE V ACCURACY (MEAN \pm STANDARD DEVIATION, %) OF COMPARISON MODELS IN INDUCTIVE PHASE

	Original models						Modified models based on DE-SSC							
Data-	M1	M1	M2	M2	M3	M4	M5	DE-SSC+M1	DE-SSC+M1	DE-SSC+M2	DE-SSC+M2	DE-SSC+M3	DE-SSC+M4	DE-SSC+M5
sets	(KNN)	(SVM)	(KNN)	(SVM)				(KNN)	(SVM)	(KNN)	(SVM)			
D1	81.09±4.39	80.18±2.77	80.42±5.03	80.18±2.77	79.64±9	81.09±4.39	83±7.53	81.09±4.39	80.18±2.77	81.09±4.66	80.18±2.77	80.94±8.11	80.93±2.29	81.42±6.16
D2	82.35±4.73	85.09±3.68	82.88±4.42	85.52±3.13	81.08±4.78	84.51±3.81	77.84±5.34	82.05±3.04	85.76±3.24	83.6±3.87	85.67±3.04	79.65±5.79	85.17±1.76	78.3±5.73
D3	41.35±8.79	31.92±3.74	31.95±9.67	28.73±3.31	31.43±8.14	21.44±10.07	35.49±14.55	39.06±6.59	31.92±3.74	32.78±7.09	32.59±4.38	33.2±8.62	24.73±5.69	40.79±10.43
D4	80±3.46	55.17±1.65	80.11±3.24	55.89±3.33	78.98±6.85	56.92±3.43	70.97±4.71	76.09±3.15	55.17±1.65	80.12±3.24	55.78±3.47	79.22±6.39	57.9±2.51	71.55±5.11
D5	98.18±1.73	98.25±1.15	90.68±4.79	93.85±2.5	90.71±3.41	96.72±1.8	92.09±4.57	94.7±3.61	97.35±2.68	96.59±1.75	95.11±3.38	94.99±1.54	98.79±0.32	95.27±2.46
D6	47.25±3.22	50±0	51.92±4.5	50±0	50.5±5.65	52.75±11.87	49.42±4.99	49.25±4.49	50±0	49.75±5.8	49.42±2.12	50.17±5.58	52.42±2.85	52.58±4.28
D7	71.14±3.62	70.75±1.13	74.87±4.41	70.63±1.12	70.03±3.75	70.05±2.73	71.99±3.96	70.64±1.97	70.75±1.13	74.74±4.49	71±1.27	71.49±4.63	71.44±1.44	71.38±4.71
D8	49.66±9.48	57.98±0.89	57±10.22	56.84±2.55	58.51±8.31	51.7±13.18	57.21±6.72	55.41±9.92	57.98±0.89	58.55±10.96	57.51±1.99	56.08±10.36	58.16±0.65	59.67±8.03
D9	52.93±4.46	54.04±3.16	54.15±5.03	53.92±3.05	53.57±4.69	53.62±5.33	55.98±6.15	52.82 ± 3.32	54.04±3.16	55.08±5.68	54.46±2.69	54.76±5.79	55.71±2.57	55.3±5.33
D10	72.93±7.17	30.27±6.65	74.67±5.92	27.11±2.13	65.64±3.53	27.93±4.89	67.07±5.08	77.42±6.03	27.6±3.63	74.6±5.93	$26.44{\pm}2.61$	64.42 ± 3.84	37.63±8.58	63.33±6.81
D11	84.73±4.27	52.33±3.99	84.69±3.54	57.96±5.88	75.42±3.16	58.67±7.37	77.62±3.9	86.69 ± 4.24	56.58 ± 8.82	84.44±3.6	58.4±5.4	76.18±3.65	72.02±6.6	75.2±3.41
D12	41.07±3.19	42.58±0.64	42.49±3.97	43.23±1.52	47.53±4.26	43.59±2.07	46.32±2.68	43.25±3.24	42.58±0.64	42.95±3.92	43.03±2.27	45.74±2.71	44.69±1.7	46.13±2.39
D13	83.12±3.8	84.62±5.98	82.36±4.48	85.29±5.8	78.8±5.41	84.18±5.25	79.11±7.54	$80.88{\pm}4.39$	$84.94{\pm}5.02$	82.87±4.11	84.79±5.36	80.58±3.66	85.16 ± 2.42	79.22±10.01
D14	88.02±7.15	51.45±16.04	85.8±8.43	58.33±7.26	86.88±5.46	63.33±11.84	88.85±3.03	88.13 ± 8.03	55.62±13.9	87.92 ± 5.05	59.34±7.47	89.32 ± 4.79	73.72±11.68	88.83±4.31
D15	50.05±4.56	53.08±0.13	51.44±6.74	53.2±0.56	56.5±5.08	52.65±1.39	55.57±3.78	51.12±5.9	53.08 ± 0.13	51.32 ± 6.72	53.11±0.24	58.27±3.56	55.82 ± 3.75	58.32±2.77
D16	91.27±1.97	52.27±3.89	93.78±1.34	77.67±5.56	84.51±2.22	90.53±1.2	81.34±4.39	94.36±1.45	56.73±14.11	96.82±0.7	79.15±9.13	87.56±2.59	92.99±2.2	84.37±3.03
D17	54.18±12.33	35.8±12.95	61.42±16.04	35.67±22.25	$65.47{\pm}5.87$	40.08±28.46	66.26±9.14	57.66±13.81	35.8±12.95	61.52±16.12	35.3±22.98	65.47±5.74	60.92 ± 4.93	65.92±11.46
D18	36.29±24.02	32.37±3.17	35.62±23.16	30.36±19.88	71.48±3.92	35.38±29.34	70.24±1.93	36.04±23.63	32.37±3.17	35.65±23.19	$28.51{\pm}20.2$	71.32 ± 5.23	$63.35 {\pm} 6.07$	69.78±2.85
D19	43.93±11.53	35.63±1.05	49.27±9.28	42.21±4.94	44.17 ± 9.55	42.2±9.67	44.47±15.38	41.61±8.21	34.37±3.29	50.96 ± 8.52	43.19±5.25	44.21±8.72	41.38 ± 5.17	50.69±12.39
D20	54.67±5.15	50±0	57.27±2.23	50±0	56.22±2.05	38.07±15.58	53.8±3.26	64.02±3.1	50±0	60.84±3.54	50.49±1.55	56.98±1.99	69.21±13.43	54.93±1.62
D21	$55.34{\pm}4.02$	$50.13 {\pm} 0.17$	60.69±4.39	50±0.17	$55.18{\pm}1.52$	49.87±0.17	53.6±1.95	62.98±2.59	50.13 ± 0.17	61.82±2.96	49.96±0.12	53.71 ± 1.88	$49.87{\pm}0.02$	53.66±2.48
D22	67.7±6.34	73.58±6.23	68.24±7.12	73.58±6.23	73.01±5.8	73.58±6.23	73.42±0.97	69.88±7.83	73.58±6.23	$68.03{\pm}7.18$	73.58±6.23	73.34±6.7	72.57±0.73	73.53±1
D23	71.11±12.44	72.59±14.52	76.05±14.39	74.69±10.75	65.19±11.01	64.07±23.3	69.38±7.98	73.33±14.86	72.59±14.1	76.3±14.3	72.59±11.09	66.42 ± 8.88	78.93±4.76	70±9.58
D24	84.05 ± 4.4	84.05±4.4	84.05±4.4	84.05±4.4	82.95 ± 5.75	84.05±4.4	82.53±7.93	84.05±4.4	84.05±4.4	84.05±4.4	84.05±4.4	83.2±5.1	83.75 ± 0.53	83.58±9.69
D25	$87.53{\pm}5.08$	89.27±4	85.21±7.84	90.14±3.92	$89.68{\pm}4.63$	88±5.79	90.55±6.5	90.39±2.79	89.41±4.79	87.19±4.33	90.86±3.57	88.1 ± 5.88	91.19±2.05	89.11±6.4
D26	70.95 ± 9.06	64.1±5.63	73.62±8.65	64.1±5.63	81.27 ± 8.13	80.61±7.13	85.62±4.62	70.1±6.75	64.1±5.63	74.28±8.16	64.38±6.09	86.51±6.87	80.88 ± 5.68	81.29±6.82
D27	84±11.42	53.33±17.21	90.22±4.94	74±4.06	86.67±7.03	61.33±20.8	88.67±7.73	87.78 ± 6.81	62±10.91	91.11±5.13	78.89±6.97	86±6.63	83.26±12.41	90.67±7.83
D28	66±8.69	31.2±7.32	63.2±9.05	52.6±8.99	60±5.45	39.6±12.54	61.96 ± 4.45	55.53±7.24	30.93±7.13	62.8±8.74	54±9.14	61.47±4.99	52.49±8.31	$60.01 {\pm} 7.21$
D29	$74.25{\pm}8.73$	75.87±11.23	74.05 ± 9.57	80.56±3.86	67.82±13.76	80.12±4.16	66.55±11.01	71.36±9.3	76.27±11.44	74.93±8.22	80.72 ± 3.86	68.33±14.76	78.93±0.83	67.23±10.89
D30	$70.09{\pm}4.11$	$75.38{\pm}14.7$	68.27 ± 6.85	78.48±5.76	$92.18{\pm}9.63$	76.88±7.79	93.7±4.7	71.82±3.33	79.44±5.57	69.67±8.29	79.27±5.27	91.85±5.18	82.31±2.03	89.1±7.56
D31	65.54±5.31	64.39±1.71	66.94±4.4	64.7±2	$68.94{\pm}5.46$	66.21±3.91	67.87±2.55	65.53±6.71	64.39±1.71	66.76±4.55	64.65±2.05	68.59±2.86	66.85±3.29	66.22±2.21
D32	$63.18{\pm}6.01$	$65.58{\pm}0.78$	65±6.32	66.23±1.85	67.8±5.18	67.5±4.05	66.67±5.61	66.02±3.75	65.58±0.78	64.71±6.68	66.3±2.27	66.08±3.69	68.81±3.49	66.03±4.98
D33	87.58±2.54	66.84±5.45	87.69±2.28	67.63±4.92	85.53±5.1	69.48±3.33	83.15±4.35	87.73±2.21	67.01±5.25	87.81±2.15	67.98±3.44	86.52±3.97	72.59±5.54	82.22±4.59
D34	55.74±13.66	53.56±14.32	59.64±10.3	50.24±11.5	58.99±9.23	44.06±14.84	60.02±5.73	61.96±12.38	53.56±14.32	62.98±6.69	46.28±11.2	59.83±8.93	54.86±7.2	61.02±6.15
D35	64.32±19.46	79.35±9.78	77.22±9.3	79.35±9.78	79.49±9.04	79.35±9.78	78.53±1.84	68.93±17.85	79.35±9.78	75.35±8.68	79.35±9.78	78.88±8.33	79.4±1.07	78.16±1.74
D36	35.71±9.73	35.83±14.6	31.71±14.38	36.25±8.09	34.25±8.76	26.46±12.46	39.29±9.51	39.93±11.35	35.17±14.54	38.6±15.16	40.18±12.42	40.64±8.13	39.52±8.05	41.82±11.57
D37	80.15±6.16	55.17±1.47	80.56±7.31	56.21±2.07	78.11±7.38	56.87±2.59	78.32±7.32	76.93±6.86	55.17±1.47	80.61±7.3	56.16±2.26	78.43±7.28	56.61±2.15	78.32±7.24
D38	75.56±2.93	77.51±1.73	75.56±2.93	77.51±1.73	77.5±1.68	77.55±1.69	74.89±3.44	76.01±2.68	77.51±1.73	75.56±2.93	77.51±1.73	77.95±1.47	77.61±0.39	74.8±2.86
D39	82.73±3.94	80±4.17	90.76±2.57	80±4.17	80.91±3.96	82.2±7.46	81.87±1.01	88.07±3.83	80±4.17	91.58±2.88	80±4.17	82.2±3.43	87.47±1.53	81.4±0.78
D40	42.46±11.94	25.88±4.37	47.36±10.44	38.99±4.83	48.55±8.6	35.2±6.86	47.73±10.61	47.36±9.43	27.84±4.47	47.88±9.19	37.62±4.07	47.22±9.91	38.47±8.21	49.14±8.8
D41	28.79±6.92	5.35±3.37	29.33±4.3	12.36±2.02	35.99±3.71	8.59±4.45	39.6±4.05	33.57±6.38	29.33±4.3	28.72±3.36	10.94±2.85	35.89±6.06	15.08±4.27	41.82±2.2
D42	79.47±1.11	82.92±2.22	78.04±2.52	82.43±2.69	75.17±2.27	83.88±1.71	70.54±4.03	79.41±1.17	82.94±2.22	77.92±2.48	82.47±2.7	75.23±3.11	82.75±2.5	71.22±3.79
D43	80.98±23.05	47.91±19.09	75.52±13.03	52.42±7.46	76.81±9.78	49.87±16.74	79.72±13.04	88.09±12.32	51.98±18.04	86.58±9.86	57.23±10	83.29±9.72	79.57±20.35	79.44±12.42
D44	94.96±3.34	95.99±2.13	93.75±3.58	96.46±2.16	94.42±2.6	96.72±2.62	92.74±6.06	95.81±2.15	96.43±1.91	96.39±1.92	96.46±2.16	94.2±2.41	96.24±0.82	94.33±2.57
D45	35.41±5.47	25.34±12.31	40.73±7.08	25.16±12.73	40.32±5.61	23.93±19.54	37.32±6.37	35.57±4.49	25.34±12.31	40.01±6.77	25.37±12.68	39.74±5.77	31.64±13.2	37.85±4.5
Aver- age	66.84±7.22	58.55±5.68	68.14±6.99	60.99±5.1	68.53±5.92	60.25±8.4	68.64±5.82	68.23±6.49	59.71±5.61	69.2±6.47	61.34±5.51	69.2±5.67	66.31±4.58	69±5.76

model when it has poor performance in some cases. However, overall, DE-SSC+M2 with KNN has the highest accuracy in transductive phase, and both DE-SSC+M2 with KNN and DE-SSC+M3 have the highest accuracy in inductive phase.

4) It is well known that the base classifiers play an important role in self-labeled SSC models. For instance, the generalization ability of self-labeled SSC models depends chiefly on the performances of their base classifiers. According to Tables IV–VII, both M1 and M2 obtain the different accuracies with different base classifiers as expected, including original models and modified models. In addition, the base classifier KNN achieves the better performances than the base classifier SVM. Furthermore, M4 outperforms than M2 and M2 outperforms than M1 under the condition of the same SVM base classifier. On the other hand, M2 also outperforms than M1 under the condition of the same KNN base classifier. However, we can find that all original models have been improved by applying the DE-SSC framework, despite what kind of base classifiers they use.

Therefore, we conclude that the DE-SSC framework is effective for any self-labeled SSC model to alleviate the weaknesses caused by unlabeled data during the iterative self-labeling process and then achieve a stronger generalized ability.

D. Impacts of Labeled Ratio

This part will analyze the behaviors of the DE-SSC framework with the different labeled ratio. We choose D16 and D25 to test the impacts of the labeled ratio because they have covered the common problems of low dimension and high dimension. The ten-fold cross-validation strategy is also used to determine the final performances. In order to save space, we have put the

TABLE VIResults of Wilcoxon Signed-Ranks Test in Transductive PhaseWITH A SIGNIFICANCE LEVEL $\alpha = 0.1$

Comparison	М	ean of ac	curacy	Standard deviation of accuracy				
	R+ $R-$		<i>p</i> -value	R+	R-	<i>p</i> -value		
DE- SSC+M1(KNN) versus M1(KNN)	778	212	0.0005	711	279	0.0060		
DE- SSC+M1(SVM) versus M1(SVM)	192	39	0.0041	157	74	0.0771		
DE- SSC+M2(KNN) versus M2(KNN)	846	144	0.0000	658	332	0.0290		
DE- SSC+M2(SVM) versus M2 (SVM)	415	326	0.2617	258	522	0.9668		
DE-SSC+M3 versus M3	701	334	0.0194	558	477	0.3258		
DE-SSC+M4 versus M4	665	370	0.0485	176	859	0.9999		
DE-SSC+M5 versus M5	608	382	0.0946	512	478	0.4237		

_		 N / I I
	D	 \//II
		V I I

Results of Wilcoxon Signed-Ranks Test in Inductive Phase With a Significance Level $\alpha = 0.1$

Comparison	Mo	ean of acc	curacy	Standard deviation of accuracy			
	R+	R-	<i>p</i> -value	R+	R-	<i>p</i> -value	
DE- SSC+M1(KNN) versus M1(KNN)	685	261	0.0053	652	294	0.0156	
DE- SSC+M1(SVM) versus M1(SVM)	164	46	0.0145	131	100	0.3011	
DE- SSC+M2(KNN) versus M2(KNN)	746	200	0.0005	652	294	0.0156	
DE- SSC+M2(SVM) versus M2 (SVM)	475.5	265.5	0.0648	225	516	0.9829	
DE-SSC+M3 versus M3	677	358	0.0363	616	419	0.1343	
DE-SSC+M4 versus M4	940	95	0.0000	905	130	0.0000	
DE-SSC+M5 versus M5	617	418	0.1295	544	491	0.3803	

complete results in the Supplementary File¹. In the following discussions, we will cite the supplementary figures from that file.

Figs. S1–S4 in the Supplementary File (see footnote 1) graphically show the results when we increase the initial labeled ratio from 10% to 90%. Note that some of the results are picked as an example to present in Fig. 2. Since more labeled data are used to train the classifier with the increase of the initial labeled ratio, the accuracies of both original and modified models overall in-



Fig. 2. Results with different initial labeled ratio in inductive phase: (a) D16, (b) D25.

crease as expected. For dataset D16, we find that the accuracies unexpectedly drop with the increase of the labeled ratio when it is larger than 0.4. This may be explained by the reason that the redundant labeled data actually increase the probability of overfitting when the labeled ratio is larger than 0.4. As presented in Section III-B, we know that the DE-POAC algorithm is supervised by the labeled data. When the redundant labeled data are used to train in the DE-SSC framework, the probability of over fitting will be further increased after the applications of the DE-POAC algorithm.

In addition, there are several cases that the DE-SSC has not improved the performances of original models. Similarly, the reason may be that the random selection process in the realization of algorithm or that the initial labeled data can already represent the whole data space. Thanks to the increase of initial labeled ratio, the appearances of unlabeled data with falsepositive label-predictions will occur less frequently. Therefore, the improvements of accuracy achieved by DE-SSC decrease with the increase of the labeled ratio.

In summary, we conclude that the DE-SSC framework has a better performance in the situation of low labeled ratio, which is a common phenomenon in SSC problems.

E. Impacts of Iteration Number and Computational Efficiency

First, we will evaluate the impacts of different maximum subsequent generations *G* of the DE-POAC algorithm when the initial labeled ratio is 10%. D16 and D25, and the tenfold cross-validation strategy are also adopted to carry out the experiments. Similarly, the complete results are presented in Figs. S5–S8 in the Supplementary File (see footnote 1) and some of the results are picked as an example to present in Fig. 3. From these figures, we find that the DE-SSC framework shows unstable performances during the initial steps, whilst it predictably shows a robust performance in general in the next steps. In summary, the DE-SSC framework performs well with a relative small iteration number, which means that it can achieve a good performance with the low-cost computing. Similarly, we attribute the exceptions for the reason that random selection process or the labeled data is sufficient for training.

Furthermore, in order to illustrate the computational complexity of the DE-SSC framework, we have compared the central processing unit (CPU) running time between the original

¹See https://pan.baidu.com/s/1hrK0BQk.



Fig. 3. Results with different iteration number in inductive phase: (a) D16, (b) D25.



Fig. 4. Results of the running time between the original models and the modified models on two datasets: (a) D16, (b) D25.

models and their modified versions on datasets D16 and D25. Each model is only executed once under the conditions of maximum subsequent generations G is 100 and initial labeled ratio is 10%. The corresponding results are depicted in Fig. 4. Unsurprisingly, each modified model takes more CPU running time than its corresponding original model in most cases because the modified models have extra computing burden of DE-POAC algorithm. However, it is noted that the modified model surprisingly reduces the CPU running time in the case of M5 on dataset D16, which means that the DE-SSC framework can accelerate convergence for M5 in some cases. In addition, we still find that the increase of CPU running time caused by the DE-POAC algorithm is relatively small except for the case of M1. This exception may be explained by the principle of M1. Original M1 simply selects the unlabeled data which are nearest to labeled data to label, while modified M1 adds the extra DE-POAC algorithm into each iterative self-labeling process, resulting in the much time cost. In summary, Fig. 4 shows that modified models have comparable CPU running time to the original models.

In fact, the core of the DE-SSC framework is the DE-POAC algorithm. Regarding to formula (8), the classifier C and the maximum subsequent generations G are the critical factors in deciding the computational efficiency of the DE-POAC algorithm. Consequently, we suppose that the DE-SSC framework will also perform well in computational efficiency and detection ability of final solutions by selecting the appropriate classifier C and maximum subsequent generations G. Obviously, the results shown in Figs. S5–S8 in the Supplementary File (see footnote 1) and Fig. 4 confirm this supposition. Therefore, we can say that the DE-SSC framework can find a right balance between computational efficiency and detection ability of final solutions.

V. CONCLUSION

Summary: In this paper, DE-SSC is proposed as a framework for any self-labeled SSC model to achieve a stronger generalized ability. The DE-SSC framework employs DE-POAC algorithm to alleviate the weaknesses caused by unlabeled data during the iterative self-labeling process of self-labeled SSC models. Specifically, five representative self-labeled SSC models with different characteristics are modified to evaluate the performance of the proposed DE-SSC framework. A series of experiments on 45 benchmark datasets are conducted to check the efficiency of the DE-SSC framework in improving the classification accuracy of self-labeled SSC models. Experimental results clearly reflect that the DE-SSC framework is a suitable tool for a self-labeled SSC model to control the unlabeled data to achieve higher classification accuracy. Moreover, some other issues of DE-SSC framework, such as impacts of the labeled ratio, impacts of iteration number, and computational efficiency, are also analyzed on two representative datasets. According to the experimental results, we conclude that the DE-SSC framework can perform well on the conditions of low labeled ratio as well as relative small iteration number, which is favorable to practical application in industry.

Possible extensions: Although the proposed DE-SSC framework has shown the promising prospect, there are several open issues to be considered. In the future, we plan to research a technology to automatically determine the value of iteration number for DE-POAC, rather than a fixed value. Subsequently, the improved DE-SSC framework will be developed and applied in recommender systems [51]–[53].

REFERENCES

- M. Kafai and B. Bhanu, "Dynamic bayesian networks for vehicle classification in video," *IEEE Trans. Ind. Informat.*, vol. 8, no. 1, pp. 100–109, Feb. 2012.
- [2] Y. Xiong and R. Zuo, "Effects of misclassification costs on mapping mineral prospectivity," Ore Geol. Rev., vol. 82, pp. 1–9, Apr. 2017.
- [3] Y. Su, S. Shan, X. Chen, and W. Gao, "Hierarchical ensemble of global and local classifiers for face recognition," *IEEE Trans. Image Process.*, vol. 18, no. 8, pp. 1885–1896, Aug. 2009.
- [4] F. J. Luo *et al.*, "Advanced pattern discovery-based fuzzy classification method for power system dynamic security assessment," *IEEE Trans. Ind. Informat.*, vol. 11, no. 2, pp. 416–426, Apr. 2015.
- [5] M. Cococcioni, B. Lazzerini, and S. L. Volpi, "Robust diagnosis of rolling element bearings based on classification techniques," *IEEE Trans. Ind. Informat.*, vol. 9, no. 4, pp. 2256–2263, Nov. 2013.
- [6] F. Schwenker and E. Trentin, "Pattern classification and clustering: A review of partially supervised learning approaches," *Pattern Recognit. Lett.*, vol. 37, no. 1, pp. 4–14, 2014.
- [7] I. Triguero, S. García, and F. Herrera, "Self-labeled techniques for semisupervised learning: Taxonomy, software and empirical study," *Knowl. Inf. Syst.*, vol. 42, no. 2, pp. 245–284, 2015.
- [8] X. Zhu, "Semi-supervised learning literature survey," Comput. Sci., vol. 37, no. 1, pp. 63–77, 2008.
- [9] D. J. Miller and H. S. Uyar, "A mixture of experts classifier with learning based on both labelled and unlabelled data," *Adv. Neural Inf. Process. Syst.*, vol. 9, pp. 571–577, 1996.
- [10] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, nos. 2/3, pp. 103–134, 2000.
- [11] J. Wang, T. Jebara, and S. F. Chang, "Semi-supervised learning using greedy max-cut," J. Mach. Learn. Res., vol. 14, no. 1, pp. 771–800, 2013.
- [12] X. Liu, S. Pan, Z. Hao, and Z. Lin, "Graph-based semi-supervised learning by mixed label propagation with a soft constraint," *Inf. Sci.*, vol. 277, no. 2, pp. 327–337, 2014.

- [13] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. S. Olkopf, "Learning with local and global consistency," *Adv. Neural Inf. Process. Syst.*, vol. 16, no. 4, pp. 321–328, 2004.
- [14] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn.*, 1999, pp. 200–209.
- [15] O. Chapelle and A. Zien, "Semi-supervised classification by low density separation," in *Proc. 10th Int. Workshop Artif. Intell. Stat.*, 2005, pp. 57–64.
- [16] M. Li and Z.-H. Zhou, "SETRED: Self-training with editing," in Proc. Pac. Asia Conf. Knowl. Discovery Data Mining, 2005, pp. 611–621.
- [17] E. Riloff, J. Wiebe, and T. Wilson, "Learning subjective nouns using extraction pattern bootstrapping," in *Proc. 7th Conf. Natural Lang. Learn. HLT-NAACL*, 2003, pp. 25–32.
- [18] M. M. Adankon and M. Cheriet, "Help-training for semi-supervised support vector machines," *Pattern Recognit.*, vol. 44, no. 9, pp. 2220–2230, 2011.
- [19] H. Gan, N. Sang, R. Huang, X. Tong, and Z. Dan, "Using clustering analysis to improve semi-supervised classification," *Neurocomputing*, vol. 101, no. 3, pp. 290–298, 2013.
- [20] A. Blum, "Combining labeled and unlabeled data with co-training," in Proc. 11th Annu. Conf. Comput. Learn. Theory, 1998, pp. 92–100.
- [21] J. Du, C. X. Ling, and Z.-H. Zhou, "When does cotraining work in real data?" *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 5, pp. 788–799, May 2011.
- [22] Z. Jiang, S. Zhang, and J. Zeng, "A hybrid generative/discriminative method for semi-supervised classification," *Knowl.-Based Syst.*, vol. 37, pp. 137–145, 2013.
- [23] 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.
- [24] M. Li and Z.-H. Zhou, "Improve computer-aided diagnosis with machine learning techniques using undiagnosed samples," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 6, pp. 1088–1098, Nov. 2007.
- [25] Z.-H. Zhou and M. Li, "Semi-supervised learning by disagreement," *Knowl. Inf. Syst.*, vol. 24, no. 3, pp. 415–439, 2010.
- [26] S. Sun, "A survey of multi-view machine learning," *Neural Comput. Appl.*, vol. 23, nos. 7/8, pp. 2031–2038, 2013.
- [27] A. Singh, R. Nowak, and X. Zhu, "Unlabeled data: Now it helps, now it doesn't," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2008, pp. 1513–1520.
- [28] K. Chen and S. Wang, "Semi-supervised learning via regularized boosting working on multiple semi-supervised assumptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 129–143, Jan. 2011.
- [29] Y.-F. Li and Z.-H. Zhou, "Towards making unlabeled data never hurt," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 175–188, Jan. 2015.
- [30] Y. Wang and S. Chen, "Safety-aware semi-supervised classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 24, no. 11, pp. 1763–1772, Nov. 2013.
- [31] A. Dong, F.-I. Chung, and S. Wang, "Semi-supervised classification method through oversampling and common hidden space," *Inf. Sci.*, vol. 349, pp. 216–228, 2016.
- [32] C. Deng and M. Zu Guo, "Tri-training and data editing based semisupervised clustering algorithm," in *Proc. Mexican Int. Conf. Artif. Intell.*, 2006, pp. 641–651.
- [33] Y. Li and M. Guo, "A new relational Tri-training system with adaptive data editing for inductive logic programming," *Knowl.-Based Syst.*, vol. 35, pp. 173–185, 2012.
- [34] C. Deng and M. Zu Guo, "A new co-training-style random forest for computer aided diagnosis," J. Intell. Inf. Syst., vol. 36, no. 3, pp. 253–281, 2011.
- [35] M.-L. Zhang and Z.-H. Zhou, "CoTrade: Confident co-training with data editing," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 41, no. 6, pp. 1612–1626, Dec. 2011.
- [36] I. Triguero, S. García, and F. Herrera, "SEG-SSC: A framework based on synthetic examples generation for self-labeled semi-supervised classification," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 622–634, Apr. 2015.
- [37] R. M. Storn and K. Price, "Differential evolution: A simple and efficient adaptive scheme for global optimization over continuous spaces," J. *Global Optim.*, vol. 23, no. 1, pp. 341–359, 1995.
- [38] K. Price, R. M. Storn, and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*. New York, NY, USA: Springer, 2005.

- [39] I. Triguero, S. García, and F. Herrera, "IPADE: Iterative prototype adjustment for nearest neighbor classification," *IEEE Trans. Neural Netw.*, vol. 21, no. 12, pp. 1984–1990, Dec. 2010.
- [40] I. Triguero, S. García, and F. Herrera, "Differential evolution for optimizing the positioning of prototypes in nearest neighbor classification," *Pattern Recognit.*, vol. 44, no. 4, pp. 901–916, 2011.
- [41] C. Blake, E. Keogh, and C. J. Merz, "UCI Repository of Machine Learning Databases," Dept. Informat. Comput. Sci., University of California, Irvine, CA, 1998. Available: http://archive.ics.uci.edu/ml
- [42] J. Alcalá Fernández *et al.*, "KEEL data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework," *J. Multiple-Valued Logic Soft Comput.*, vol. 17, no. 3, pp. 255–287, 2011.
- [43] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006)[Book reviews]," *IEEE Trans. Neural Netw.*, vol. 20, no. 3, pp. 542–542, Mar. 2009.
- [44] K. V. Price, An Introduction to Differential Evolution, Maidenhead, U.K.: McGraw-Hill, 1999, pp. 79–108.
- [45] D. W. Aha, D. Kibler, and M. K. Albert, "Instance-based learning algorithms," *Mach. Learn.*, vol. 6, no. 1, pp. 37–66, 1991.
- [46] J. Shawe-Taylor and S. Sun, "A review of optimization methodologies in support vector machines," *Neurocomputing*, vol. 74, no. 17, pp. 3609–3618, 2011.
- [47] F. Neri and V. Tirronen, "Scale factor local search in differential evolution," *Memetic Comput.*, vol. 1, no. 2, pp. 153–171, 2009.
- [48] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," ACM Trans. Intell. Syst. Technol., vol. 2, no. 3, 2011, Art. no. 27.
- [49] L. Breiman, "Random forests," Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.
- [50] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80–83, 1945.
- [51] X. Luo *et al.*, "An efficient second-order approach to factorize sparse matrices in recommender systems," *IEEE Trans. Ind. Informat.*, vol. 11, no. 4, pp. 946–956, Aug. 2015.
- [52] X. Luo, M. Zhou, Y. Xia, and Q. Zhu, "An efficient non-negative matrixfactorization-based approach to collaborative-filtering for recommender systems," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1273–1284, May 2014.
- [53] X. Luo, M. Zhou, S. Li, Z. You, Y. Xia, and Q. Zhu, "A non-negative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 3, pp. 524–537, Mar. 2016.



Di Wu received the B.S. degree in applied physics from Nanjing University of Science and Technology, Jiangsu, China, and the M.S. degree in optical engineering from the Chongqing University, Chongqing, China, in 2009 and 2012, respectively. He is currently working toward the Ph.D. degree in computer science from Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing.

His research interests include data mining, machine learning, and their applications.



Xin Luo (M'14–SM'17) received the B.S. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the Ph.D. degree in computer science from Beihang University, Beijing, China, in 2011.

In 2016, he joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China, where he is currently a Professor of computer science and engineering. His research interests include

big data analysis and intelligent control. He has published more than 70 papers (including 20 IEEE Transactions papers) in his related areas.



Guoyin Wang (M'98–SM'03) received the B.E. degree in computer software, the M.S. degree in computer software, and the Ph.D. degree in computer organization and architecture from Xi'an Jiaotong University, Xi'an, China, in 1992, 1994, and 1996, respectively.

He worked at the University of North Texas, USA, and the University of Regina, Canada, as a Visiting Scholar during 1998–1999. Since 1996, he has been working at the Chongqing University of Postsand Telecommunications,

Chongqing, China, where he is currently a Professor and a Ph.D. supervisor, the Director of the Chongqing Key Laboratory of Computational Intelligence, and the Dean of the College of Computer Science and Technology. His research interests include data mining, machine learning, rough set, granular computing, cognitive computing, etc.

Dr. Wang is the President of International Rough Set Society (IRSS), a Vice-President of the Chinese Association for Artificial Intelligence (CAAI), and a council member of the China Computer Federation (CCF).



Mingsheng Shang received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China.

He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China, in 2015, and is currently a Professor of computer science and engineering. His research interests include data mining, complex networks, cloud computing, and their applications.



Ye Yuan received the B.S. degree in electronic information engineering and the M.S. degree in signal processing from the University of Electronic Science and technology, Chengdu, China, in 2010 and 2013, respectively. He is currently working toward the Ph.D. degree in computer science from Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China.

His research interests include data mining, recommender system, intelligent computing, and their applications.



Huyong Yan received the B.E. degree in computer science and technology from the University of Electronic Science and Technology of China, Chengdu, China, the M.S. degree in petroleum engineering computing technology from Northeast Petroleum University, Daqing, China, in 2007 and 2013, respectively, and the Ph.D. degree in computer science from Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China, in 2017.

His research interests include data mining, rough set, and water eutrophication assessment.