



Article

A Novel Maximum Mean Discrepancy-Based Semi-Supervised Learning Algorithm

Qihang Huang 1, Yulin He 1,2,* and Zhexue Huang 1,2

- Big Data Institute, College of Computer Science & Software Engineering, Shenzhen University, Shenzhen 518060, China; 1900271056@email.szu.edu.cn (Q.H.); zxhuang@szu.edu.cn (J.H.)
- National Engineering Laboratory for Big Data System Computing Technology, Shenzhen University, Shenzhen 518060, China
- * Correspondence: yulinhe@szu.edu.cn; Tel.: +86-185-3131-5747

Abstract: To provide more external knowledge for training self-supervised learning (SSL) algorithms, this paper proposes a maximum mean discrepancy-based SSL (MMD-SSL) algorithm, which trains a well-performing classifier by iteratively refining the classifier using highly confident unlabeled samples. The MMD-SSL algorithm performs three main steps. First, a multilayer perceptron (MLP) is trained based on the labeled samples and is then used to assign labels to unlabeled samples. Second, the unlabeled samples are divided into multiple groups with the *k*-means clustering algorithm. Third, the maximum mean discrepancy (MMD) criterion is used to measure the distribution consistency between *k*-means-clustered samples and MLP-classified samples. The samples having a consistent distribution are labeled as highly confident samples and used to retrain the MLP. The MMD-SSL algorithm performs an iterative training until all unlabeled samples are consistently labeled. We conducted extensive experiments on 29 benchmark data sets to validate the rationality and effectiveness of the MMD-SSL algorithm. Experimental results show that the generalization capability of the MLP algorithm can gradually improve with the increase of labeled samples and the statistical analysis demonstrates that the MMD-SSL algorithm can provide better testing accuracy and kappa values than 10 other self-training and co-training SSL algorithms.

Keywords: semi-supervised learning; maximum mean discrepancy; distribution consistency; self-training; co-training



Citation: Huang, Q.; He, Y.; Huang, Z. A Novel Maximum Mean
Discrepancy-Based Semi-Supervised
Learning Algorithm. *Mathematics*2022, 10, 39. https://doi.org/
10.3390/math10010039

Academic Editor: Victor Leiva

Received: 8 November 2021 Accepted: 17 December 2021 Published: 23 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Semi-supervised learning (SSL) is an important branch of data mining and machine learning [1], which uses a large number of unlabeled samples to improve the generalization capability of classifiers trained on a small number of labeled samples. Different from active learning [2], SSL focuses on the selection of easily classified samples rather than the selection of easily misclassified samples. The goal of active learning is to minimize the number of samples labeled by domain experts, while the goal of SSL is to maximize the usage of information from unlabeled samples without the intervention of domain experts. The lower labor and time costs achieved using SSL make it more suitable than active learning for a wide range of real-world applications such as automatic query classification [3], image recognition [4], fraudulent cash-out detection [5], and biological sequence analysis [6].

Up to now, researchers have proposed many useful methods to deal with SSL problems [7]. These methods can be categorized as self-training models, co-training models, generative models, semi-supervised SVM, and graph models. Self-training [8] and co-training [9] methods have attracted much attention as they are simple-to-implement and easy-to-interpret SSL paradigms. The former is a single-view SSL paradigm, which iteratively updates a classifier based on the combination of labeled samples and pseudo-labeled samples, while the latter is a multiple-view SSL paradigm that iteratively updates multiple classifiers based on the combination of labeled samples and pseudo-labeled samples. The

Mathematics 2022, 10, 39 2 of 19

objective of self-training and co-training SSL is to create one or more classifiers that are as good as possible by efficiently using a large number of unlabeled samples. Some key studies about each paradigm are summarized as follows.

- Self-training methods. Li and Zhou [10] devised a self-training algorithm named SETRED (self-training with editing), which introduced a data editing technique into the self-training process to filter out the noise in self-labeled examples. Wang et al. [11] proposed a self-training nearest neighbor rule using cut edges (SNNRCE) method, which is based on a nearest neighbor rule for classification and cuts edges in the relative neighborhood graph. Halder et al. [12] presented an advanced aggregation pheromone density based semi-supervised classification (APSSC) algorithm which makes no assumption on the data distribution and has no user-defined parameters. Wu et al. [13] designed a self-training semi-supervised classification (self-training SSC) framework based on density peaks of data, where the structure of the data space is integrated into the self-training process of SSC to help train a better classifier.
- Co-training methods. Zhou and Goldman [14] proposed a democratic co-learning (DemoCoL) method, which employs a set of different learning algorithms to train a set of classifiers separately on the labeled data and then combines the outputs using weighted voting to predict the labels of unlabeled examples. Zhou and Li [15] designed an extended co-training semi-supervised learning algorithm named Tri-Training, which generates three classifiers from the original labeled samples and then refines them using the unlabeled samples in the tri-training process. Wang et al. [16] proposed a random subspace co-training (RASCO) method which trains many classifiers based on feature subspaces of the original feature space. Yaslan and Cataltepe [17] improved the classical RASCO algorithm and gave a relevant RASCO named Rel-RASCO, which produces relevant random subspaces by considering the mutual information between features and class labels. Huang et al. [18] presented a classification algorithm based on local cluster centers (CLCC) for SSL, which was able to reduce the interference of mislabeled data.

Although the aforementioned SSL methods have shown good performance in experiments, they still have some important drawbacks that can be further improved for the self-training and co-training SSL paradigms. In particular, the selection of the most confident pseudo-labeled samples for self-training SSL mainly depends on internal judgment rather than external judgment, i.e., a classifier teaches itself using its own cognition until it is satisfied with its own learning. Moreover, for co-training SSL, the assumption that multiple views are conditionally independent always results in a high computational complexity.

To address these issues, this paper presents a novel SSL algorithm, named maximum mean discrepancy-based semi-supervised learning (MMD-SSL) that performs three main steps. First, unlabeled samples are divided into different groups using the k-means clustering algorithm. Then, the k-means-labeled samples are used as external knowledge to train a multilayer perceptron (MLP), which is then used to assign labels to unlabeled samples. The MLP-classified samples are used as internal information for the classifier training. Third, the maximum mean discrepancy (MMD) criterion measures the distribution consistency between the k-means-clustered samples and MLP-classified samples. Then, samples having a consistent distribution are labeled and used to retrain the MLP. We conduced extensive experiments on 29 benchmark data sets to validate the rationality and effectiveness of the MMD-SSL algorithm. Results show that the generalization capability of the MLP algorithm can gradually improve with the increased number of labeled samples. Moreover, a statistical analysis demonstrates that the MMD-SSL algorithm provides better testing accuracy and kappa values than 10 other self-training and co-training SSL algorithms, i.e., SETRED, SNNRCE, APSSC, Self-Training-NN, DemoCoL, Tri-Training, RASCO, Rel-RASCO, CLCC, and Co-Training-NN, where Self-Training-NN and Co-Training-NN are the classical self-training [8] and co-training [9] paradigms using neural networks as classifiers.

The remainder of this paper is organized as follows. In Section 2, we introduce the preliminaries of SSL. In Section 3, we propose the MMD-SSL method. In Section 4, we

Mathematics 2022, 10, 39 3 of 19

describe the experimental evaluation method and analyze the results. Finally, in Section 5, we conclude this paper and discuss future works.

2. Preliminaries

Assume there is a labeled data set containing ${\cal N}$ samples, described using ${\cal D}$ condition attributes and one class attribute as

$$\bar{D} = \{ (\bar{x}_n, \bar{y}_n) | \bar{x}_n = (\bar{x}_{n1}, \bar{x}_{n2}, \cdots, \bar{x}_{nD}), \bar{y}_n \in \{c_1, c_2, \cdots, c_K\}, n = 1, 2, \cdots, \mathcal{N} \}$$

and an unlabeled data set having $\mathcal M$ samples with $\mathcal D$ condition attributes as

$$D = \{(x_m, y_m) | x_m = (x_{m1}, x_{m2}, \dots, x_{mD}), y_m = \text{null}, m = 1, 2, \dots, \mathcal{M} \},$$

where c_1, c_2, \dots, c_K are K discrete labels of the data set $\bar{\mathbb{D}}$. The initial classifier $L^{(0)}$ is trained with the small number of samples from the data set $\bar{\mathbb{D}}$. The generalization capability of $L^{(0)}$ is restricted due to the insufficient sample size. The data set D is easier to obtain than $\bar{\mathbb{D}}$ because the class labels of samples are ignored in D. It is very expensive to label unlabeled samples with the help of experts. How to use the unlabeled samples to improve the generalization capability of a classifier $L^{(0)}$ trained with labeled samples is the primary focus of semi-supervised learning (SSL). Self-training and co-learning are two classical SSL paradigms. A brief description of these paradigms is given next.

2.1. SSL with Self-Training Paradigm

The origin of the self-training SSL paradigm can be traced back to Scudder [8]. After that, several extended self-training SSL methods have been developed [10–13,19]. The main algorithmic steps of the self-training SSL paradigm are listed below.

Step 1: Train a classifier L on the labeled data set \bar{D} ;

Step 2: Label the unlabeled samples in D with L;

Step 3: Evaluate the confidence scores of these newly labeled samples and obtain the data set \underline{D} including the samples with high confidence scores;

Step 4: Update the labeled data as $\bar{D} \leftarrow \bar{D} \cup \underline{D}$;

Step 5: Update the unlabeled data as $D \leftarrow D - D$.

Step 6: Repeat Step 1-5 until the stopping criteria are met.

To design an effective self-training SSL method, the key aspect is how to calculate confidence scores for the labels given to unlabeled samples. Here, we only introduce the simplest way of selecting samples with high confidence scores for reference. Assume that the probability output of an unlabeled sample x_m is $\left(p_1^{(m)}, p_2^{(m)}, \cdots, p_K^{(m)}\right)$, where

$$\sum\limits_{k=1}^{\mathcal{K}}p_k^{(m)}=1$$
. The confidence score of \mathbf{x}_m is calculated as

$$cs_m = \begin{cases} 1 & \text{, if there exists } k \in \{1, 2, \dots, K\} \text{ such that } p_k^{(m)} > \gamma \text{ holds} \\ 0 & \text{, otherwise} \end{cases}$$
, (1)

where $\gamma \in (0,1)$ is a threshold used to produce a hard label for x_m . The samples having a confidence score of 1 are selected to update the classifier. This method usually leads to many incorrectly labeled samples and can yield relatively poor training performance.

2.2. SSL with Co-Training Paradigm

The co-training SSL paradigm [9,20,21] requires two different views of a data set, i.e., two different feature subsets to label the unlabeled samples. Ideally, these two feature subsets are conditionally independent given the class and the class of samples can be correctly predicted using each view. The main algorithmic steps of the co-training SSL paradigm are provided as follows.

Mathematics 2022, 10, 39 4 of 19

Step 1: Partitioning the labeled data set \bar{D} into two labeled data sets \bar{D}_1 and \bar{D}_2 according to two different views $A^{(1)}$ and $A^{(2)}$;

Step 2: Train two classifiers L_1 and L_2 on the labeled data sets \bar{D}_1 and \bar{D}_2 , respectively;

Step 3: Label the unlabeled samples in D with L_1 ;

Step 4: Evaluate the confidence scores of these newly labeled samples with L₁ and obtain the $A^{(2)}$ -view data set $\underline{\mathbb{D}}_2$ including the samples having high confidence scores;

Step 5: Label the unlabeled samples in D with L₂;

Step 6: Evaluate the confidence scores of these newly labeled samples with L₂ and obtain the $A^{(1)}$ -view data set \underline{D}_1 including the samples having high confidence scores;

Step 7: Update the labeled data as $D_1 \leftarrow D_1 \cup \underline{D}_2$;

Step 8: Update the labeled data as $D_2 \leftarrow D_2 \cup \underline{D}_1$;

Step 9: Update the unlabeled data as $\bar{D} \leftarrow \bar{D} - \underline{D}$, where \underline{D} is composed of the samples in \underline{D}_1 and \underline{D}_2 with full views.

Step 10: Repeat Step 1-9 until the stopping criteria are met.

Developing an effective co-training SSL method requires selecting two conditionally independent and sufficient views. Prior studies [22] have shown that the generalization capability of a classifier can be improved when the dependence between the two views is weak.

3. The Proposed MMD-SSL Algorithm

This section presents the novel maximum mean discrepancy-based semi-supervised learning (MMD-SSL) algorithm. Its main steps are listed in Algorithm 1. MMD-SSL belongs to the self-training SSL paradigm and perform three main operations, i.e., training a multilayer perceptron (MLP) classifier on the labeled data set, clustering the unlabeled samples using the *k*-means algorithm, measuring the distribution consistency between the classification, and clustering results using the maximum mean discrepancy (MMD) criterion [23,24].

To train the MLP classifier and apply the *k*-means algorithm, we use the standard *sklearn* packages, i.e., *MLPClassifier* (https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html, accessed on 16 December 2021), and *KMeans* (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html (accessed on 16 December 2021)). For the data sets

$$Q_{i} = \left\{ \begin{pmatrix} x_{m}^{(i)}, y_{m}^{(i)} \end{pmatrix} \middle| \begin{array}{c} x_{m}^{(i)} = \begin{pmatrix} x_{m1}^{(i)}, x_{m2}^{(i)}, \cdots, x_{mD}^{(i)} \end{pmatrix}, \\ y_{m}^{(i)} \in \{w_{1}, w_{2}, \cdots, w_{\bar{K}}\}, m = 1, 2, \cdots, \mathcal{M}_{i} \end{array} \right\}$$

and

$$D_{k} = \left\{ \left(\hat{\mathbf{x}}_{n}^{(k)}, \hat{y}_{n}^{(k)} \right) \middle| \begin{array}{c} \hat{\mathbf{x}}_{n}^{(k)} = \left(\hat{\mathbf{x}}_{n1}^{(k)}, \hat{\mathbf{x}}_{n2}^{(k)}, \cdots, \hat{\mathbf{x}}_{n\mathcal{D}}^{(k)} \right), \\ \hat{y}_{n}^{(k)} \in \{v_{1}, v_{2}, \cdots, v_{\mathcal{K}}\}, n = 1, 2, \cdots, \mathcal{N}_{k} \end{array} \right\},$$

the MMD value between them is calculated as

$$MMD(Q_{i}, D_{k}) = \begin{bmatrix} \frac{1}{\mathcal{M}_{i}} \sum_{m=1}^{\mathcal{M}_{i}} \sum_{n=1}^{\mathcal{M}_{i}} \kappa\left(\mathbf{x}_{m}^{(i)}, \mathbf{x}_{m}^{(i)}\right) + \frac{1}{\mathcal{N}_{k}} \sum_{m=1}^{\mathcal{N}_{k}} \sum_{n=1}^{\mathcal{N}_{k}} \kappa\left(\hat{\mathbf{x}}_{m}^{(k)}, \hat{\mathbf{x}}_{n}^{(k)}\right) \\ -\frac{2}{\mathcal{M}_{i} \mathcal{N}_{k}} \sum_{m=1}^{\mathcal{N}_{k}} \sum_{n=1}^{\mathcal{N}_{k}} \kappa\left(\mathbf{x}_{m}^{(i)}, \hat{\mathbf{x}}_{n}^{(k)}\right) \end{bmatrix}^{\frac{1}{2}}, \quad (2)$$

where $\mathbf{x}_m^{(i)}$, $\hat{\mathbf{x}}_n^{(k)} \in \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{\mathcal{M}}\}$, $\{v_1, v_2, \cdots, v_{\bar{\mathcal{K}}}\}$, $\{w_1, w_2, \cdots, w_{\bar{\mathcal{K}}}\} \subset \{c_1, c_2, \cdots, c_{\mathcal{K}}\}$, \mathcal{M}_i and \mathcal{N}_k are the numbers of samples in \mathbf{Q}_i and \mathbf{D}_k ,

$$\kappa(\mathbf{a}, \mathbf{b}) = \exp\left[-\frac{\sum_{d=1}^{D} (a_d - b_d)^2}{2\sigma^2}\right]$$
(3)

Mathematics 2022, 10, 39 5 of 19

is the Gaussian kernel function to measure the distance between two vectors $\mathbf{a}=(a_1,a_2,\cdots,a_{\mathcal{D}})$ and $\mathbf{b}=(b_1,b_2,\cdots,b_{\mathcal{D}})$ in reproducing the kernel Hilbert space, and σ^2 is the kernel radius.

Algorithm 1 MMD-SSL Algorithm.

Input: A labeled data set \bar{D} and an unlabeled data set D.

Output: The predicted labels y_1, y_2, \dots, y_M and a multilayer perceptron (MLP) *Learner*.

- 1: repeat
- 2: Train a MLP *Learner* with two hidden layers on the data set \bar{D} ;
- 3: Predict the labels of samples from the unlabeled data set D with *Learner* and partition D into $\bar{\mathcal{K}}(\bar{\mathcal{K}} \leq \mathcal{K})$ disjoint data subsets $D_1, D_2, \cdots, D_{\bar{\mathcal{K}}}$ according to the predicted labels;
- 4: Apply the k-means clustering algorithm to partition D into $\bar{\mathcal{K}}$ disjointed data subsets $Q_1, Q_2, \dots, Q_{\bar{\mathcal{K}}}$;
- 5: **for** i = 1; $i \leq \mathcal{K}$; i + + **do**
- 6: **for** $k = 1; k \le \bar{\mathcal{K}}; k + + \mathbf{do}$
- Calculate the maximum mean discrepancy (MMD) between Q_i and D_k as MMD(Q_i , D_k);
- 8: end for
- 9: end for
- 10: $D = \emptyset$;
- 11: $M = N = \emptyset;$
- 12: repeat
- 13: Determine the data pair (Q_m, D_n) with the consistent probability distribution, where

$$(m,n) = \underset{(i,k) \in \{(x,y) \mid x \in \{1,2,\cdots,\mathcal{K}\} - M, y \in \{1,2,\cdots,\mathcal{K}\} - N\}}{\operatorname{arg\,min}} \{ \operatorname{MMD}(Q_i, D_k) \};$$

- 14: Label the samples in Q_m with the label of D_n ;
- 15: $M \leftarrow M \cup \{m\};$
- 16: $N \leftarrow N \cup \{n\};$
- 17: $\underline{\mathbf{D}} \leftarrow \underline{\mathbf{D}} \cup (\mathbf{Q}_m \cap \mathbf{D}_n);$
- 18: **until** $M = \{1, 2, \dots, \bar{\mathcal{K}}\}$ or $N = \{1, 2, \dots, \bar{\mathcal{K}}\}$.
- 19: Update the labeled data as $\bar{D} \leftarrow \bar{D} \cup \underline{D}$;
- 20: Update the unlabeled data as $D \leftarrow D \underline{D}$.
- 21: **until** The number of labels predicted with *Learner* for unlabeled data set D is 1 or the number of samples in the unlabeled data set D is less than the given threshold $\zeta > 0$.

In the MMD-SSL algorithm, it is feasible to match the k-means-clustered data set Q_m with the MLP-classified data set D_n . Assume that Q_m has a consistent probability distribution with D_n , which indicates that

$$MMD(Q_m, D_n) < MMD(Q_m, D_k)$$

holds for any $k \in \{1, 2, \dots, \bar{K}\}$ and $k \neq n$. This observation can be demonstrated by the following illustration in Figure 1. We can see that the unlabeled data sets are, respectively, partitioned into 5 parts by the k-means algorithm $(Q_1, Q_2, Q_3, Q_4, Q_5)$ and MLP classifier $(D_1, D_2, D_3, D_4, D_5)$. MMD values corresponding to different data pairs are calculated as

$MMD(Q_i, D_k)$	D_1	D_2	D_3	D_4	D_5
Q_1	0.08	0.52	0.48	0.58	0.49
Q_2	0.45	0.30	0.43	0.26	0.44
Q_3	0.45	0.51	0.46	0.57	0.33 .
Q_4	0.44	0.48	0.00	0.54	0.44
O ₅	0.54	0.57	0.53	0.63	0.25

Mathematics 2022, 10, 39 6 of 19

Then, we can get the distribution consistency measure results as

$$\begin{cases} & \text{MMD}(Q_4, D_3) = \min_{\substack{i=1,2,3,4,5 \\ k=1,2,3,4,5}} \{\text{MMD}(Q_i, D_k)\} = 0.00 \\ & \text{MMD}(Q_1, D_1) = \min_{\substack{i=1,2,3,5 \\ k=1,2,4,5}} \{\text{MMD}(Q_i, D_k)\} = 0.08 \\ & \text{MMD}(Q_5, D_5) = \min_{\substack{i=2,3,5 \\ k=2,4,5}} \{\text{MMD}(Q_i, D_k)\} = 0.25 \\ & \text{MMD}(Q_2, D_4) = \min_{\substack{i=2,3,5 \\ k=2,4}} \{\text{MMD}(Q_i, D_k)\} = 0.26 \\ & \text{MMD}(Q_3, D_2) = 0.51 \end{cases}$$

for the data sets shown in Figure 1. It indicates that the k-means-clustered data sets Q_1, Q_2, Q_3, Q_4, Q_5 have the same class labels as the MLP-classified data sets D_3, D_1, D_5, D_4, D_2 , respectively. Taking the data pair (Q_4, D_3) as example, the samples in both Q_4 and D_3 are labeled as class 3 and further added into the labeled data set to update the training of the MLP classifier in the next iteration.

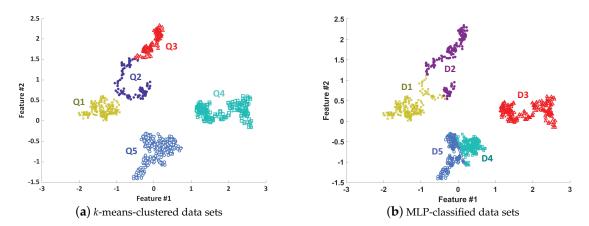


Figure 1. Distribution consistency measures between *k*-means-clustered and MLP-classified data sets ($\sigma^2 = 0.01$).

The stopping criteria of the MMD-SSL algorithm is that there is only one class predicted by the MLP classifier in the unlabeled data set or that the number of unlabeled samples in the unlabeled data set is less than the threshold $\zeta > 0$. The rationale of the first stopping criterion is that applying the k-means clustering algorithm is unnecessary for a data set having one cluster, that is, where the MLP classifier predicts the same label for all unlabeled samples. In this situation, the training of the MMD-SSL algorithm stops and all samples from the unlabeled data set obtain the predicted label. For the second stopping criterion, we adopt an adaptive threshold determination strategy, i.e, let $\zeta = \bar{\mathcal{K}}$. It indicates that the training of MMD-SSL is stopped when the number of remaining unlabeled samples is equal to the number of labels predicted by the MLP classifier.

4. Experimental Results and Analysis

We conducted two experiments to validate the rationality and effectiveness of the proposed MMD-SSL algorithm. The MMD-SSL algorithm was implemented using the Python programming language and other SSL algorithms were downloaded from sci2s (https://sci2s.ugr.es/SelfLabeled (accessed on 16 December 2021)), which is a soft computing and intelligent information system developed by the University of Granada research group. All the experiments were carried out on a personal computer equipped with an Intel(R) Quad-core 3.00 GHz i5-7400 CPU and 16 GB of main memory.

Mathematics 2022, 10, 39 7 of 19

4.1. Rationality Validation

The first experiment was done to evaluate the suitability of using the MLP classification algorithm and the k-means clustering algorithm in the MMD-SSL algorithm. We checked the testing performance of 28 different combinations of classification and clustering algorithms on a synthetic data set #1, which can be downloaded in any country from our BaiduPan online storage (https://pan.baidu.com/s/1aDm8n7AA2ETtSumM5LXVBQ (accessed on 16 December 2021)) with extraction code nc19. The classification algorithms included MLP, Bernoulli naive Bayes (BNB) BernoulliNB (https://scikit-learn.org/stable/mo dules/generated/sklearn.naive_bayes.BernoulliNB.html (accessed on 16 December 2021)), Gaussian naive Bayes (GNB) Gaussian NB (https://scikit-learn.org/stable/modules/genera ted/sklearn.naive_bayes.GaussianNB.html (accessed on 16 December 2021)), support vector machines (SVM) SVC (https://scikit-learn.org/stable/modules/generated/sklearn.sv m.SVC.html (accessed on 16 December 2021)), k-nearest neighbors (k-NN) KNeighborsClassifier (https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighbors Classifier.html (accessed on 16 December 2021)), decision tree DecisionTreeClassifier (https:// scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html (accessed on 16 December 2021)), and random forest RandomForestClassifier (https://scikit-l earn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html (accessed on 16 December 2021)). The clustering algorithms included k-means, agglomerative clustering AgglomerativeClustering (https://scikit-learn.org/stable/modules/ge nerated/sklearn.cluster.AgglomerativeClustering.html (accessed on 16 December 2021)), spectral clustering SpectralClustering (https://scikit-learn.org/stable/modules/generate d/sklearn.cluster.SpectralClustering.html (accessed on 16 December 2021)), and BIRCH Birch (https://scikit-learn.org/stable/modules/generated/sklearn.cluster.Birch.html (accessed on 16 December 2021)). There were five classes in the synthetic data set #1, which was divided into three parts, i.e., labeled training data set (356 samples), unlabeled training data set (692 samples), and labeled testing data set (298 samples). The labeled training data set was randomly generated by using the standard sklearn package make_blobs (https://scikit-learn.org/dev/modules/generated/sklearn.datasets.make_blobs.html (accessed on 16 December 2021)) with $n_{features} = 2$, centers = 5, and cluster_std = 0.3. The unlabeled training and labeled testing samples were generated by randomly adding small values in the interval [-0.08, 0.08] to the feature values of labeled samples. The MLP classifier with a learning rate of 0.1 and a maximum iteration number of 300 used in this experiment had two hidden layers and each hidden layer included 400 ReLU activation nodes.

The comparative results corresponding to 10 times the independent training of the MMD-SSL algorithms are summarized in Table 1. We can see that the clustering algorithms combined with the MLP classifier obtain better average testing accuracies than the clustering algorithms combined with other classifiers. The average testing accuracies of the MMD-SSL algorithm with MLP classification and *k*-means, agglomerative, spectral, and the BIRCH clustering algorithm are 0.975, 0.958, 0.996, and 0.954, respectively. Due to its better generalization capability, the MLP classifier was selected to carry out the classification in the MMD-SSL algorithm. For MMD-SSL with an MLP classifier, we used *k*-means to conduct the clustering task for the unlabeled data set due to its simple model structure and acceptable semi-supervised learning performance.

Secondly, the convergence of the proposed MMD-SSL algorithm on another synthetic data set #2 as shown in Figure 2 was validated. The synthetic data set #2 had the same data distribution as the synthetic data set #1. Figure 3 shows the convergence process of the MMD-SSL algorithm. We can see that the MMD-SSL algorithm reaches convergence with only six iterations, i.e., all unlabeled samples are labeled with the gradually updated MLP classifier. The testing accuracies of the MLP classifier corresponding to these six iterations in Figure 2d are 0.711, 0.819, 0.836, 0.859, 0.869, and 0.869. The increase rate of the testing accuracy reaches more than 15%. It indicates that the designed MMD-SSL algorithm is

Mathematics 2022, 10, 39 8 of 19

reasonable and able to improve the generalization capability of a classifier by properly utilizing the unlabeled training samples.

Table 1. Testing accuracies of different combinations of classification and clustering algorithms in MMD-SSL algorithm.

Classification Algorithm	Clustering Algorithm	Maximum	Minimum	Mean
	k-means	0.673	0.673	0.673
DNID	Agglomerative	0.673	0.673	0.673
BNB	Spectral	0.673	0.673	0.673
	BIRCH	0.673	0.673	0.673
	k-means	0.957	0.947	0.952
CNIP	Agglomerative	0.907	0.907	0.907
GNB	Spectral	0.947	0.930	0.937
	BIRCH	0.937	0.937	0.937
	k-means	0.953	0.953	0.953
CV II A	Agglomerative	0.920	0.920	0.920
SVM	Spectral	0.997	0.997	0.997
	BIRCH	0.920	0.920	0.920
	k-means	0.963	0.963	0.963
1 3737	Agglomerative	0.930	0.930	0.930
k-NN	Spectral	1.000	0.993	0.997
	BIRCH	0.923	0.923	0.923
	k-means	0.987	0.867	0.888
D	Agglomerative	1.000	0.873	0.919
Decision tree	Spectral	0.920	0.830	0.863
	BIRCH	0.947	0.830	0.878
	k-means	0.993	0.890	0.930
D 1 (Agglomerative	0.923	0.867	0.902
Random forest	Spectral	0.970	0.807	0.892
	BIRCH	0.930	0.813	0.867
	k-means	0.990	0.967	0.975
) (T.P.	Agglomerative	0.987	0.937	0.958
MLP	Spectral	0.997	0.990	0.996
	BIRCH	0.967	0.930	0.954

Mathematics 2022, 10, 39 9 of 19

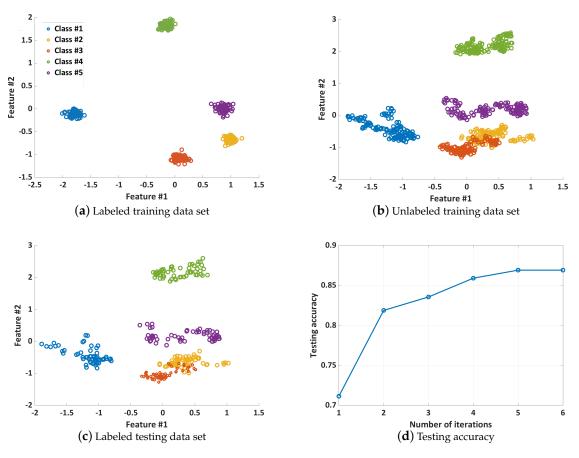


Figure 2. The synthetic data set generated with the *make_blobs* package.

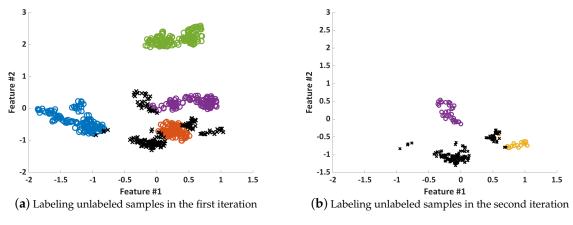


Figure 3. Cont.

Mathematics 2022, 10, 39 10 of 19

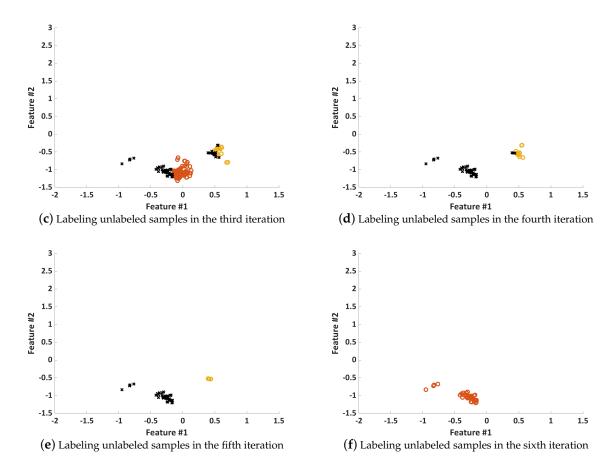


Figure 3. The convergence of the MMD-SSL algorithm.

4.2. Effectiveness Validation

In this experiment, we compared the testing accuracy and kappa coefficient [25] of the MMD-SSL algorithm with 10 other self-training and co-training SSL algorithms, namely SETRED [10], SNNRCE [11], APSSC [12], Self-Training-NN [8], DemoCoL [14], Tri-Training [15], RASCO [16], Rel-RASCO [17], CLCC [18], and Co-Training-NN [9]. The parameters of these SSL algorithms were set as follows.

- SETRED: the maximum number of iterations was 40 and the size of the initial unlabeled sample pool was 75;
- SNNRCE: the rejection threshold to test the critical region was 0.5;
- APSSC: the spread of Gaussian was 0.3, evaporation coefficient was 0.7, and MT was 0.7:
- Self-Training-NN: the maximum number of iterations was 40 and the number of nearest neighbors was 3;
- DemoCoL: the number of nearest neighbors was 3 and the confidence of pruned tree was 0.25;
- RASCO: the maximum number of iterations was 40 and the number of views was 30;
- Rel-RASCO: the maximum number of iterations was 40 and the number of views was 30;
- CLCC: the number of random forests was 6, the manipulative beta parameter was 0.4, the number of initial clusters was 4, the running frequency was 10, and the number of best center sets was 6;
- Co-Training-NN: the maximum number of iterations was 40, the number of nearest neighbors was 3, and the size of the initial unlabeled sample pool was 75.

We selected 29 data sets from the KEEL data repository (https://sci2s.ugr.es/kee l/category.php?cat=clas&order=name#sub2 (accessed on 16 December 2021)) to test the

Mathematics 2022, 10, 39 11 of 19

performance of these SSL algorithms. The detailed descriptions of these data sets are summarized in Table 2. Each data set was randomly partitioned into three parts: labeled training data, unlabeled training data, and labeled testing data. The ratios of labeled training data were set as 10% and 30%. For each labeled data ratio, the percentages of unlabeled training data and labeled testing data were 70% and 30% of the rest of the data set. Each algorithm was trained using the labeled and unlabeled training data and tested based on the labeled testing data.

Table 2. Descriptions of the 29 benchmark data sets.

Data Sets	Samples	Features	Classes	Class Distribution
appendicitis	106	7	2	85/21
australian	690	14	2	383/307
banana	5300	4	3	2924/2376
chess	3196	36	2	1527/1669
coil2000	9822	85	2	9236/586
magic	19,020	10	2	12332/6688
mammographic	830	5	2	427/403
monk-2	432	6	2	204/228
nursery	12,960	8	5	4320/4266/2/4044/328
page-blocks	5472	10	5	4913/329/28/87/115
penbased	10,992	16	10	1143/1143/1144/1055/1144/1055/1056/1142/1055/1055
phoneme	5404	5	2	3818/1586
pima	768	8	2	500/268
ring	7400	20	2	3664/3736
saheart	462	9	2	302/160
satimage	6435	36	7	1533/703/1358/626/707/1508
segment	2310	19	7	330/330/330/330/330/330
sonar	208	60	2	111/97
spambase	4597	57	2	2785/1812
spectfheart	267	44	2	55/212
texture	5500	40	11	500/500/500/500/500/500/500/500/500/500
thyroid	7200	21	3	166/368/6666
tic-tac-toe	958	9	2	332/626
titanic	2201	3	2	1490/711
twonorm	7400	20	2	3703/3697
vowel	990	13	11	90/90/90/90/90/90/90/90/90
wine	178	13	3	59/71/48
wisconsin	683	9	2	444/239
ZOO	101	16	7	41/20/5/13/4/8/10

The testing accuracy and kappa value were calculated as the average value of 10 different testing accuracies and kappa values corresponding to 10 different data partitions. Tables 3–6 present the detailed comparative results of these 11 SSL algorithms for the two labeled data ratios. We can see that (1) the MMD-SSL algorithm obtains higher testing accuracy and kappa values than the other 10 SSL algorithms for each labeled data ratio; (2) the testing accuracy and kappa value of the SSL algorithm increase gradually as the label training data increase. We also use the critical difference diagram [26] as shown in Figures 4 and 5 to present the statistical analysis results for this comparison of 11 SSL algorithms. There are 11 algorithms which are compared based on 29 data sets. For the given significance level of 0.01, the critical difference (CD) value is calculated as $q_{0.05}\sqrt{\frac{11\times(11+1)}{6\times29}}=2.516$, where $q_{0.05}$ is the CD value of Tukey's distribution corresponding to a significance level of 0.01. The statistical analysis results indicate that the MMD-SSL algorithm obtains (1) significantly better testing accuracy and kappa value than APSSC, Self-Training-NN, Tri-Training, RASCO, Rel-RASCO, CLCC, Co-Training-NN for SSL based

Mathematics 2022, 10, 39 12 of 19

on 10% labeled training data and (2) significantly better testing accuracy and kappa value than APSSC, SNNRCE, Tri-Training, RASCO, Rel-RASCO, CLCC, Co-Training-NN for SSL based on 30% labeled training data. Above all, the MMD-SSL algorithm obtains the highest average testing accuracy and kappa value among the compared SSL algorithms and thus demonstrates its effectiveness when dealing with the SSL problems. In addition, the comparative results indicate that MMD-SSL is more suitable for dealing with SSL for imbalanced data sets, because the kappa coefficient is one of the most popular indices to measure the ability to handle imbalanced classification of a learning algorithm and the average testing kappa values of MMD-SSL are higher than those of other SSL algorithms.

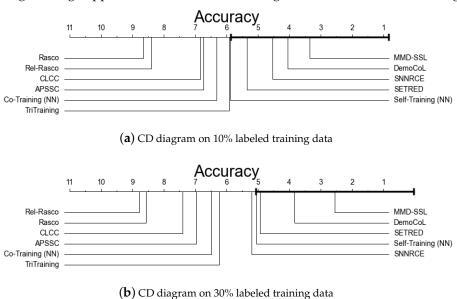


Figure 4. Critical difference diagrams corresponding to accuracy comparisons in Tables 3 and 4.

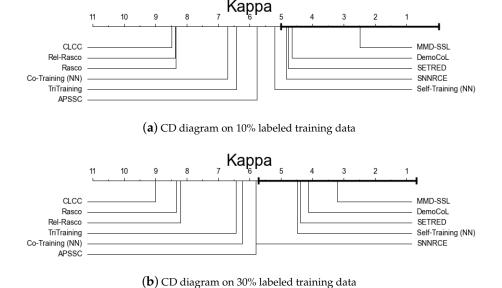


Figure 5. Critical difference diagrams corresponding to kappa comparisons in Tables 5 and 6.

Table 3. Accuracy comparison of MMD-SSL and 10 other SSL algorithms on 10% labeled data.

	MMD-SSL				:	Self-Train	ing Meth	ods					Co-Training Methods									
Data Sets	MML)-SSL	SETRE	D (2005)	SNNRC	CE (2010)	APSSC	(2013)	Self-Tra	ining (NN)	DemoC	oL (2004)	Tri-Train	ning (2005)	Rasco	(2008)	Rel-Ras	sco (2010)	CLCC	(2010)	Co-Train	ning (NN)
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
appendicitis	0.7500	0.0395	0.7373	0.1304	0.7927	0.0900	0.6773	0.2126	0.7573	0.1229	0.8218	0.0471	0.7382	0.0728	0.7936	0.0682	0.7555	0.0837	0.8500	0.0939	0.7827	0.0721
australian	0.7981	0.0323	0.8043	0.0362	0.8087	0.0329	0.8377	0.0404	0.8043	0.0362	0.8449	0.0268	0.8029	0.0421	0.7087	0.0699	0.7435	0.0389	0.8536	0.0293	0.8058	0.0319
banana	0.8725	0.0076	0.8638	0.0119	0.8662	0.0125	0.8240	0.0208	0.8638	0.0119	0.8417	0.0226	0.8681	0.0101	0.8513	0.0129	0.8472	0.0121	0.5804	0.0419	0.8460	0.0166
chess	0.9341	0.0126	0.8104	0.0283	0.8220	0.0184	0.8326	0.0218	0.8098	0.0292	0.9199	0.0175	0.8309	0.0294	0.7991	0.0278	0.8010	0.0267	0.6640	0.0374	0.7994	0.0272
coil2000	0.9147	0.0047	0.8926	0.0052	0.9128	0.0076	0.6838	0.0364	0.8904	0.0067	0.9322	0.0078	0.8795	0.0109	0.8932	0.0071	0.8946	0.0045	0.9403	0.0005	0.9009	0.0058
magic	0.8480	0.0035	0.7840	0.0074	0.7918	0.0060	0.7379	0.0114	0.7840	0.0074	0.7842	0.0164	0.7678	0.0072	0.7800	0.0070	0.7812	0.0078	0.7594	0.0187	0.7812	0.0061
mammographic	0.6528	0.0883	0.7580	0.0594	0.7773	0.0534	0.8022	0.0417	0.7591	0.0606	0.7963	0.0551	0.7699	0.0629	0.7277	0.0595	0.7151	0.0585	0.7985	0.0405	0.7128	0.0629
monk-2	0.7492	0.0618	0.6459	0.0409	0.6923	0.0831	0.6563	0.0784	0.6459	0.0452	0.9075	0.0452	0.6460	0.0751	0.6530	0.0565	0.6852	0.0894	0.7082	0.0766	0.6373	0.0823
nursery	0.9654	0.0075	0.8101	0.0081	0.7499	0.0111	0.6683	0.0158	0.7143	0.0145	0.8951	0.0041	0.8698	0.0061	0.4587	0.0162	0.4573	0.0117	0.3603	0.0159	0.7698	0.0094
page-blocks	0.9436	0.0125	0.9359	0.0082	0.9373	0.0087	0.8012	0.1202	0.9256	0.0105	0.9077	0.0956	0.9364	0.0080	0.8467	0.0129	0.8538	0.0111	0.8993	0.0033	0.9329	0.0104
penbased	0.9779	0.0044	0.9778	0.0054	0.9730	0.0047	0.8552	0.0079	0.9778	0.0054	0.9474	0.0117	0.9801	0.0049	0.9054	0.0101	0.9104	0.0112	0.7281	0.0483	0.9752	0.0062
phoneme	0.8106	0.0068	0.8046	0.0204	0.8059	0.0112	0.6832	0.0213	0.8044	0.0206	0.7874	0.0177	0.8046	0.0192	0.7998	0.0282	0.7951	0.0241	0.7585	0.0271	0.8059	0.0212
pima	0.7411	0.0130	0.6565	0.0542	0.6383	0.0660	0.7332	0.0339	0.6565	0.0542	0.6967	0.0586	0.6265	0.0562	0.6407	0.0478	0.6341	0.0560	0.6954	0.0350	0.6486	0.0497
ring	0.9024	0.0108	0.6691	0.0203	0.5573	0.0081	0.5049	0.0007	0.6691	0.0203	0.8741	0.0120	0.6041	0.0098	0.6668	0.0218	0.6692	0.0220	0.6388	0.0490	0.6776	0.0183
saheart	0.6446	0.0641	0.6300	0.0835	0.6215	0.0740	0.6559	0.0772	0.6408	0.0784	0.6819	0.0472	0.6277	0.0690	0.6039	0.0627	0.6080	0.0835	0.6538	0.0273	0.6364	0.0557
satimage	0.8542	0.0061	0.8570	0.0131	0.8547	0.0160	0.8011	0.0153	0.8466	0.0151	0.8462	0.0142	0.8521	0.0130	0.7818	0.0207	0.7855	0.0223	0.7944	0.0216	0.8491	0.0180
segment	0.8954	0.0165	0.9065	0.0172	0.9022	0.0200	0.8519	0.0248	0.9061	0.0170	0.9026	0.0169	0.9074	0.0126	0.7052	0.0474	0.7260	0.0539	0.7359	0.0333	0.8827	0.0339
sonar	0.6635	0.0597	0.6633	0.0964	0.6490	0.0961	0.7017	0.1273	0.6633	0.0964	0.6005	0.1112	0.6345	0.1276	0.6205	0.0797	0.6250	0.1276	0.5633	0.0940	0.6867	0.1160
spambase	0.9049	0.0103	0.8281	0.0189	0.8327	0.0176	0.6324	0.1340	0.8281	0.0189	0.8777	0.0188	0.8110	0.0204	0.8164	0.0175	0.8138	0.0213	0.7966	0.0234	0.8140	0.0206
spectfheart	0.7852	0.0060	0.7201	0.1189	0.7426	0.0729	0.3742	0.0744	0.6865	0.1170	0.7379	0.0828	0.6905	0.0911	0.7264	0.0860	0.7009	0.0621	0.7942	0.0166	0.5110	0.1227
texture	0.9708	0.0041	0.9513	0.0080	0.9515	0.0078	0.8733	0.0135	0.9513	0.0076	0.8944	0.0156	0.9524	0.0056	0.8124	0.0216	0.8211	0.0243	0.7182	0.0338	0.9480	0.0070
thyroid	0.9597	0.0031	0.9090	0.0093	0.9204	0.0053	0.6554	0.1572	0.8963	0.0111	0.9393	0.0175	0.9067	0.0107	0.8956	0.0096	0.8962	0.0081	0.9258	0.0025	0.9072	0.0077
tic-tac-toe	0.6722	0.0248	0.7255	0.0406	0.7360	0.0298	0.6701	0.0494	0.7150	0.0457	0.6900	0.0309	0.7067	0.0268	0.6754	0.0536	0.7005	0.0532	0.6461	0.0392	0.7193	0.0443
titanic	0.7628	0.0098	0.6402	0.1419	0.6416	0.1423	0.7756	0.0293	0.6402	0.1419	0.7756	0.0282	0.7415	0.0353	0.6402	0.1415	0.6402	0.1419	0.6979	0.0193	0.6402	0.1419
twonorm	0.9740	0.0017	0.9358	0.0074	0.9459	0.0071	0.9759	0.0075	0.9358	0.0074	0.9645	0.0082	0.9109	0.0065	0.9242	0.0102	0.9265	0.0057	0.9589	0.0104	0.9364	0.0081
vowel	0.4148	0.0226	0.4808	0.0533	0.4838	0.0591	0.4343	0.0517	0.4879	0.0521	0.4162	0.0639	0.4980	0.0491	0.3152	0.0778	0.3172	0.0842	0.2273	0.0362	0.4859	0.0680
wine	0.9815	0.0117	0.9438	0.0249	0.9271	0.0664	0.9605	0.0359	0.9438	0.0249	0.9493	0.0390	0.9265	0.0446	0.6686	0.1034	0.6180	0.1399	0.9490	0.0400	0.8716	0.0963
wisconsin	0.9600	0.0037	0.9478	0.0428	0.9622	0.0294	0.9593	0.0219	0.9478	0.0428	0.9650	0.0257	0.9462	0.0366	0.8642	0.0601	0.8629	0.0595	0.9522	0.0351	0.9434	0.0481
zoo	0.8800	0.0340	0.9347	0.0548	0.9228	0.0937	0.9347	0.0548	0.9236	0.0683	0.9314	0.0650	0.9347	0.0548	0.6019	0.1902	0.6619	0.0879	0.8364	0.1369	0.8367	0.1171
Average	0.8339	0.0201	0.8008	0.1263	0.8007	0.1315	0.7432	0.1474	0.7957	0.1256	0.8320	0.1247	0.7990	0.1275	0.7302	0.1370	0.7327	0.1358	0.7409	0.1675	0.7843	0.1295

Testing accuracies of the 10 other SSL algorithms: https://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/SelfLabeled/SelfLabeled10.ods (accessed on 16 December 2021).

Table 4. Accuracy comparison of MMD-SSL and 10 other SSL algorithms on 30% labeled data.

	MME	N CCI				Self-Train	ing Meth	nods				Co-Training Methods										
Data Sets	MME)-55L	SETRE	D (2005)	SNNRC	CE (2010)	APSSC	(2013)	Self-Tra	ining (NN)	DemoC	oL (2004)	Tri-Trair	ning (2005)	Rasco	(2008)	Rel-Ras	sco (2010)	CLCC	(2010)	Co-Train	ning (NN)
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
appendicitis	0.8563	0.0424	0.8109	0.1389	0.8209	0.1305	0.8291	0.0947	0.8300	0.1240	0.8691	0.0734	0.8027	0.1039	0.7218	0.1410	0.7873	0.2023	0.8482	0.0983	0.8391	0.1194
australian	0.8510	0.0199	0.8101	0.0229	0.8130	0.0307	0.8594	0.0275	0.8101	0.0229	0.8536	0.0300	0.8000	0.0323	0.7710	0.0414	0.7797	0.0377	0.8522	0.0462	0.8101	0.0188
banana	0.8914	0.0035	0.8700	0.0124	0.8725	0.0110	0.8326	0.0220	0.8700	0.0124	0.8728	0.0133	0.8687	0.0114	0.8645	0.0118	0.8651	0.0127	0.5813	0.0304	0.8609	0.0131
chess	0.9564	0.0062	0.8648	0.0239	0.8673	0.0120	0.9083	0.0188	0.8651	0.0240	0.9596	0.0174	0.7991	0.0247	0.8604	0.0237	0.8608	0.0213	0.6755	0.0377	0.8542	0.0257
coil2000	0.9209	0.0078	0.8992	0.0075	0.9158	0.0073	0.7317	0.0358	0.8977	0.0081	0.9320	0.0036	0.8820	0.0112	0.9001	0.0079	0.8965	0.0101	0.9403	0.0005	0.9018	0.0083
magic	0.8526	0.0057	0.7950	0.0098	0.7952	0.0060	0.7450	0.0096	0.7950	0.0098	0.8016	0.0081	0.7776	0.0125	0.7940	0.0098	0.7940	0.0103	0.7531	0.0166	0.7932	0.0092
mammographic	0.7600	0.0188	0.7620	0.0752	0.7876	0.0782	0.8033	0.0405	0.7620	0.0752	0.8300	0.0556	0.7557	0.0784	0.7401	0.0714	0.7353	0.0736	0.7902	0.0462	0.7329	0.0662
monk-2	0.9677	0.0185	0.7513	0.0588	0.7421	0.0396	0.7807	0.0707	0.7581	0.0579	0.9452	0.0434	0.6795	0.0663	0.7350	0.0640	0.7467	0.0513	0.7393	0.0975	0.7017	0.0688
nursery	0.9968	0.0016	0.8357	0.0115	0.7673	0.0171	0.7143	0.0094	0.7687	0.0114	0.9212	0.0107	0.7415	0.0116	0.6737	0.0115	0.6833	0.0110	0.3633	0.0127	0.8355	0.0118
page-blocks	0.9496	0.0014	0.9461	0.0060	0.9450	0.0058	0.8520	0.0177	0.9428	0.0062	0.9289	0.0523	0.9468	0.0075	0.8818	0.0143	0.8765	0.0133	0.8978	0.0005	0.9448	0.0062
penbased	0.9857	0.0011	0.9901	0.0028	0.9740	0.0047	0.8877	0.0069	0.9901	0.0028	0.9729	0.0047	0.9888	0.0051	0.9664	0.0035	0.9640	0.0044	0.7325	0.0326	0.9902	0.0028
phoneme	0.8560	0.0019	0.8470	0.0192	0.8436	0.0155	0.7150	0.0260	0.8470	0.0192	0.8029	0.0224	0.8464	0.0147	0.8420	0.0206	0.8434	0.0190	0.7716	0.0153	0.8464	0.0188
pima	0.7117	0.0089	0.6694	0.0494	0.6903	0.0537	0.7253	0.0285	0.6733	0.0475	0.7305	0.0474	0.6590	0.0610	0.6511	0.0434	0.6563	0.0278	0.7267	0.0366	0.6550	0.0582
ring	0.9566	0.0026	0.7104	0.0131	0.6007	0.0197	0.5049	0.0007	0.7104	0.0131	0.9089	0.0093	0.6453	0.0099	0.7097	0.0126	0.7099	0.0134	0.6170	0.0215	0.7132	0.0130
saheart	0.6935	0.0264	0.6644	0.0441	0.6733	0.0481	0.6472	0.0969	0.6644	0.0408	0.7080	0.0520	0.6687	0.0451	0.6753	0.0264	0.6275	0.0565	0.6753	0.0565	0.6513	0.0590
satimage	0.8694	0.0038	0.8862	0.0118	0.8723	0.0200	0.7972	0.0172	0.8822	0.0105	0.8693	0.0104	0.8738	0.0110	0.8553	0.0172	0.8544	0.0131	0.7863	0.0258	0.8850	0.0109
segment	0.9432	0.0060	0.9411	0.0173	0.9368	0.0140	0.8887	0.0193	0.9411	0.0173	0.9416	0.0144	0.9459	0.0156	0.8797	0.0244	0.8775	0.0232	0.7589	0.0389	0.9377	0.0149
sonar	0.8444	0.0119	0.7645	0.0840	0.7450	0.1232	0.7829	0.1025	0.7645	0.0840	0.7310	0.0739	0.7690	0.0807	0.6864	0.1072	0.7119	0.1064	0.6200	0.0758	0.7688	0.0849
spambase	0.9264	0.0042	0.8695	0.0091	0.8658	0.0139	0.8095	0.0252	0.8695	0.0091	0.9052	0.0168	0.8488	0.0159	0.8662	0.0104	0.8653	0.0096	0.7881	0.0207	0.8669	0.0087
spectfheart	0.7704	0.0346	0.7127	0.0988	0.7534	0.0874	0.4309	0.0591	0.7010	0.0925	0.7121	0.0799	0.7346	0.0757	0.7044	0.1260	0.6744	0.0908	0.7942	0.0166	0.5769	0.1289
texture	0.9919	0.0011	0.9805	0.0049	0.9605	0.0047	0.8867	0.0123	0.9800	0.0055	0.9331	0.0108	0.9733	0.0055	0.9256	0.0099	0.9225	0.0141	0.7240	0.0481	0.9780	0.0055
thyroid	0.9631	0.0072	0.9183	0.0080	0.9285	0.0049	0.4918	0.0903	0.9100	0.0068	0.9521	0.0059	0.9153	0.0046	0.9064	0.0063	0.9089	0.0052	0.9258	0.0025	0.9144	0.0049
tic-tac-toe	0.7917	0.0496	0.7923	0.0348	0.7955	0.0385	0.7264	0.0461	0.7975	0.0274	0.7630	0.0482	0.7192	0.0197	0.7798	0.0359	0.7882	0.0267	0.6608	0.0168	0.7861	0.0307
titanic	0.8000	0.0049	0.6407	0.1414	0.6552	0.1464	0.7774	0.0338	0.6407	0.1414	0.7792	0.0272	0.7083	0.0309	0.6407	0.1414	0.6402	0.1419	0.7197	0.0190	0.6407	0.1414
twonorm	0.9726	0.0008	0.9439	0.0093	0.9493	0.0095	0.9758	0.0067	0.9439	0.0093	0.9701	0.0072	0.9131	0.0134	0.9405	0.0094	0.9411	0.0098	0.9573	0.0105	0.9435	0.0098
vowel	0.6765	0.0145	0.7737	0.0240	0.7566	0.0232	0.6990	0.0393	0.7889	0.0273	0.5960	0.0509	0.7889	0.0245	0.6101	0.0507	0.6162	0.0344	0.2263	0.0268	0.7606	0.0239
wine	0.9704	0.0189	0.9275	0.0557	0.8869	0.0627	0.9549	0.0336	0.9382	0.0461	0.9660	0.0373	0.9386	0.0631	0.7121	0.1182	0.6856	0.1316	0.9327	0.0597	0.8526	0.0757
wisconsin	0.9737	0.0024	0.9535	0.0435	0.9637	0.0214	0.9578	0.0234	0.9535	0.0435	0.9666	0.0280	0.9635	0.0225	0.9170	0.0540	0.9083	0.0563	0.9593	0.0236	0.9372	0.0480
zoo	0.9067	0.0133	0.9331	0.0714	0.9264	0.0627	0.9397	0.0679	0.9331	0.0714	0.9133	0.0848	0.9347	0.0548	0.8036	0.0782	0.8222	0.0928	0.8806	0.0881	0.9164	0.0830
Average	0.8830	0.0117	0.8367	0.1001	0.8312	0.1022	0.7812	0.1361	0.8355	0.1004	0.8633	0.0996	0.8238	0.1045	0.7936	0.1027	0.7946	0.1034	0.7482	0.1649	0.8240	0.1096

Testing accuracies of the 10 other SSL algorithms: https://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/SelfLabeled/SelfLabeled30.ods (accessed on 16 December 2021).

 $\textbf{Table 5.} \ \ \textbf{Kappa comparison of MMD-SSL algorithm and 10 other SSL algorithms on 10\% labeled data}.$

					9	Self-Train	ing Meth	nods				Co-Training Methods											
Data Sets	MMI)-SSL	SETRE	D (2005)	SNNRC	CE (2010)	APSSC	(2013)	Self-Tra	ining (NN)	Democra	tic-Co (2004)	Tri-Train	ning (2005)	Rasco	(2008)	Rel-Rasc	o (2010)	CLCC	(2010)	Co-Train	ning (NN)	
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	
appendicitis	0.3858	0.2156	0.0982	0.4071	0.1343	0.3257	0.3360	0.2999	0.1607	0.3967	0.1231	0.2462	0.0502	0.2347	0.1929	0.2756	-0.0089	0.2613	0.3487	0.4145	0.0787	0.2266	
australian	0.6122	0.0581	0.5994	0.0762	0.6071	0.0707	0.6727	0.0804	0.5994	0.0762	0.6826	0.0567	0.5956	0.0911	0.4073	0.1459	0.4756	0.0827	0.7025	0.0629	0.6006	0.0695	
banana	0.7714	0.0138	0.7238	0.0243	0.7287	0.0253	0.6451	0.0422	0.7238	0.0243	0.6789	0.0460	0.7328	0.0203	0.6986	0.0267	0.6905	0.0241	0.0755	0.1045	0.6862	0.0343	
chess	0.8504	0.0276	0.6196	0.0568	0.6431	0.0366	0.6638	0.0438	0.6183	0.0588	0.8392	0.0353	0.5142	0.0612	0.5968	0.0559	0.6008	0.0538	0.3114	0.0794	0.5967	0.0551	
coil2000	0.0809	0.0242	0.0529	0.0350	0.0385	0.0363	0.0777	0.0331	0.0515	0.0329	0.0404	0.0444	0.0491	0.0328	0.0412	0.0419	0.0450	0.0511	0.0000	0.0000	0.0490	0.0362	
magic	0.6577	0.0107	0.5163	0.0157	0.5252	0.0132	0.4042	0.0202	0.5163	0.0157	0.4722	0.0472	0.4795	0.0160	0.5074	0.0141	0.5100	0.0168	0.4040	0.0587	0.5066	0.0127	
mammographic	0.5607	0.0774	0.5160	0.1201	0.5543	0.1077	0.6064	0.0826	0.5183	0.1224	0.5937	0.1088	0.5393	0.1270	0.4548	0.1208	0.4303	0.1190	0.5981	0.0806	0.4223	0.1273	
monk-2	0.5329	0.0688	0.2912	0.0790	0.3751	0.1710	0.3098	0.1565	0.2914	0.0877	0.8150	0.0885	0.2856	0.1557	0.2962	0.1094	0.3641	0.1786	0.4094	0.1452	0.2811	0.1653	
nursery	0.9380	0.0211	0.7201	0.0119	0.6333	0.0162	0.5751	0.0229	0.5828	0.0213	0.8448	0.0058	0.5619	0.0096	0.2059	0.0240	0.2039	0.0162	0.0556	0.0253	0.6627	0.0141	
page-blocks	0.7381	0.0384	0.6409	0.0397	0.6419	0.0528	0.3701	0.1119	0.6073	0.0525	0.6415	0.2004	0.6445	0.0369	0.1928	0.0499	0.2339	0.0590	0.0640	0.1288	0.6282	0.0513	
penbased	0.9761	0.0047	0.9753	0.0060	0.9700	0.0052	0.8391	0.0088	0.9753	0.0060	0.9416	0.0130	0.9779	0.0054	0.8949	0.0112	0.9004	0.0124	0.6978	0.0536	0.9724	0.0069	
phoneme	0.5840	0.0148	0.5192	0.0548	0.5220	0.0287	0.3994	0.0306	0.5186	0.0556	0.5024	0.0418	0.5193	0.0506	0.5074	0.0732	0.4975	0.0622	0.3605	0.1412	0.5177	0.0595	
pima	0.4098	0.0260	0.2652	0.1026	0.1994	0.1139	0.4025	0.0809	0.2652	0.1026	0.3122	0.1445	0.1963	0.0805	0.2376	0.0944	0.2135	0.1166	0.2649	0.1305	0.1913	0.1035	
ring	0.8170	0.0274	0.3338	0.0412	0.1068	0.0164	0.0000	0.0000	0.3338	0.0412	0.7475	0.0241	0.2019	0.0202	0.3293	0.0442	0.3341	0.0447	0.2725	0.0991	0.3511	0.0370	
saheart	0.2037	0.1129	0.1896	0.1880	0.1325	0.1633	0.2835	0.1496	0.2162	0.1796	0.2508	0.0979	0.1078	0.1798	0.0968	0.1586	0.1217	0.1846	0.0694	0.1052	0.1286	0.1405	
satimage	0.8376	0.0068	0.8235	0.0159	0.8206	0.0196	0.7561	0.0190	0.8108	0.0183	0.8102	0.0174	0.8177	0.0159	0.7304	0.0256	0.7350	0.0274	0.7424	0.0278	0.8135	0.0223	
segment	0.8874	0.0255	0.8909	0.0201	0.8859	0.0234	0.8273	0.0289	0.8904	0.0198	0.8864	0.0197	0.8919	0.0147	0.6561	0.0553	0.6803	0.0628	0.6919	0.0388	0.8631	0.0396	
sonar	0.2193	0.1694	0.3072	0.2097	0.2813	0.2017	0.3927	0.2639	0.3078	0.2100	0.1688	0.2429	0.2520	0.2665	0.2133	0.1834	0.2229	0.2788	0.0940	0.2049	0.3742	0.2352	
spambase	0.7920	0.0205	0.6393	0.0389	0.6465	0.0350	0.3420	0.2042	0.6393	0.0389	0.7439	0.0382	0.6034	0.0396	0.6136	0.0345	0.6093	0.0434	0.5646	0.0505	0.6138	0.0406	
spectfheart	0.0352	0.0431	0.2229	0.3057	0.1883	0.2330	0.1020	0.0540	0.1466	0.2706	0.3629	0.1527	0.1686	0.2238	0.2078	0.2188	0.0924	0.1830	0.0000	0.0000	0.1152	0.1792	
texture	0.9722	0.0056	0.9464	0.0088	0.9466	0.0085	0.8606	0.0149	0.9464	0.0084	0.8838	0.0172	0.9476	0.0062	0.7936	0.0237	0.8032	0.0268	0.6900	0.0371	0.9428	0.0077	
thyroid	0.6701	0.0164	0.2564	0.0685	0.2164	0.0508	0.0764	0.0258	0.2238	0.0621	0.2620	0.2377	0.2153	0.0591	0.1646	0.0310	0.1652	0.0619	0.0000	0.0000	0.2465	0.0571	
tic-tac-toe	0.3618	0.0213	0.3692	0.0916	0.4150	0.0563	0.3077	0.1102	0.3648	0.1022	0.2599	0.1340	0.2087	0.0785	0.2805	0.1135	0.3224	0.1165	0.0721	0.1256	0.3317	0.1058	
titanic	0.4533	0.0415	0.2700	0.1939	0.2717	0.1946	0.4376	0.0786	0.2700	0.1939	0.4378	0.0800	0.2584	0.1352	0.2701	0.1924	0.2700	0.1939	0.1595	0.1113	0.2700	0.1939	
twonorm	0.9454	0.0033	0.8716	0.0147	0.8919	0.0141	0.9519	0.0150	0.8716	0.0147	0.9289	0.0163	0.8219	0.0130	0.8484	0.0205	0.8530	0.0115	0.9178	0.0208	0.8727	0.0162	
vowel	0.4663	0.0069	0.4289	0.0586	0.4322	0.0651	0.3778	0.0569	0.4367	0.0573	0.3578	0.0702	0.4478	0.0540	0.2467	0.0856	0.2489	0.0926	0.1500	0.0398	0.4344	0.0748	
wine	0.9327	0.0225	0.9154	0.0370	0.8908	0.0987	0.9404	0.0540	0.9154	0.0370	0.9235	0.0587	0.8887	0.0678	0.5022	0.1535	0.4174	0.2122	0.9230	0.0604	0.8103	0.1384	
wisconsin	0.9280	0.0182	0.8824	0.0992	0.9158	0.0659	0.9088	0.0490	0.8824	0.0992	0.9238	0.0564	0.8787	0.0829	0.6870	0.1442	0.6792	0.1433	0.8914	0.0803	0.8711	0.1109	
Z00	0.6286	0.0771	0.9080	0.0790	0.8963	0.1251	0.9079	0.0790	0.8917	0.0970	0.9031	0.0925	0.9083	0.0788	0.4530	0.2920	0.5034	0.1747	0.7838	0.1694	0.7538	0.1957	
Average	0.6293	0.0421	0.5446	0.2831	0.5349	0.2959	0.5095	0.2838	0.5371	0.2799	0.5979	0.2855	0.5091	0.3007	0.4251	0.2406	0.4212	0.2525	0.3902	0.3137	0.5168	0.2798	

Testing kappas of 10 other SSL algorithms: https://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/SelfLabeled/SelfLabeled10.ods (accessed on 16 December 2021).

 $\textbf{Table 6.} \ \ \text{Kappa comparison of MMD-SSL algorithm and 10 other SSL algorithms on 30\% labeled data}.$

) () (F	N C C Y			9	Self-Train	ing Meth	ods			Co-Training Methods											
Data Sets	MME)-55L	SETRE	D (2005)	SNNRC	CE (2010)	APSSC	(2013)	Self-Trai	ning (NN)	Democra	atic-Co (2004)	Tri-Traiı	ning (2005)	Rasco	(2008)	Rel-Ras	sco (2010)	CLCC	(2010)	Co-Trai	ning (NN)
	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std	Mean	Std
appendicitis	0.2580	0.1411	0.4948	0.3553	0.4203	0.3990	0.5251	0.2277	0.4850	0.3679	0.5746	0.1911	0.4284	0.2312	0.1479	0.2958	0.3976	0.4958	0.4593	0.3375	0.4029	0.4151
australian	0.6318	0.0837	0.6139	0.0487	0.6194	0.0661	0.7178	0.0553	0.6139	0.0487	0.7024	0.0606	0.5919	0.0688	0.5358	0.0854	0.5532	0.0776	0.7007	0.0966	0.6135	0.0398
banana	0.7872	0.0119	0.7368	0.0251	0.7417	0.0225	0.6619	0.0441	0.7368	0.0251	0.7415	0.0273	0.7343	0.0231	0.7258	0.0238	0.7269	0.0255	0.0802	0.0792	0.7178	0.0263
chess	0.9620	0.0058	0.7290	0.0479	0.7339	0.0241	0.8160	0.0377	0.7296	0.0482	0.9190	0.0350	0.5921	0.0507	0.7201	0.0477	0.7208	0.0428	0.3368	0.0807	0.7072	0.0518
coil2000	0.0617	0.0189	0.0696	0.0542	0.0430	0.0451	0.0963	0.0308	0.0672	0.0540	0.0296	0.0325	0.0629	0.0576	0.0680	0.0541	0.0722	0.0506	0.0000	0.0000	0.0693	0.0588
magic	0.6747	0.0123	0.5407	0.0200	0.5321	0.0124	0.4270	0.0221	0.5407	0.0200	0.5144	0.0225	0.4989	0.0292	0.5385	0.0201	0.5384	0.0212	0.3945	0.0561	0.5353	0.0187
mammographic	0.5692	0.0736	0.5236	0.1505	0.5753	0.1562	0.6083	0.0807	0.5236	0.1505	0.6598	0.1111	0.5104	0.1568	0.4797	0.1427	0.4704	0.1473	0.5812	0.0918	0.4644	0.1319
monk-2	0.9012	0.0932	0.5029	0.1140	0.4734	0.0869	0.5595	0.1415	0.5152	0.1139	0.8903	0.0861	0.3553	0.1373	0.4699	0.1234	0.4923	0.0988	0.4773	0.1967	0.4090	0.1243
nursery	0.9944	0.0015	0.7590	0.0168	0.6583	0.0248	0.6361	0.0197	0.6621	0.0170	0.8836	0.0160	0.6233	0.0173	0.5219	0.0167	0.5360	0.0162	0.0639	0.0188	0.7590	0.0173
page-blocks	0.8214	0.0339	0.6910	0.0406	0.6793	0.0443	0.4276	0.0402	0.6817	0.0417	0.6625	0.1495	0.7065	0.0411	0.3615	0.0668	0.3528	0.0566	0.0000	0.0000	0.6923	0.0375
penbased	0.9887	0.0009	0.9890	0.0031	0.9711	0.0052	0.8752	0.0077	0.9890	0.0031	0.9699	0.0052	0.9876	0.0057	0.9627	0.0039	0.9600	0.0049	0.7026	0.0362	0.9891	0.0031
phoneme	0.6355	0.0284	0.6245	0.0503	0.6142	0.0410	0.4376	0.0414	0.6245	0.0503	0.5443	0.0435	0.6221	0.0400	0.6122	0.0519	0.6165	0.0492	0.4380	0.0448	0.6227	0.0489
pima	0.3189	0.0650	0.2757	0.1129	0.3010	0.1288	0.3865	0.0702	0.2867	0.1059	0.3952	0.1076	0.2427	0.1384	0.2385	0.0742	0.2430	0.0532	0.3356	0.1042	0.2154	0.1278
ring	0.9209	0.0096	0.4175	0.0262	0.1950	0.0401	0.0000	0.0000	0.4175	0.0262	0.8175	0.0186	0.2856	0.0201	0.4162	0.0253	0.4165	0.0268	0.2282	0.0441	0.4233	0.0261
saheart	0.2776	0.0351	0.2705	0.1113	0.2464	0.0947	0.2797	0.1599	0.2754	0.1003	0.3288	0.1133	0.2361	0.1437	0.2701	0.0663	0.1732	0.1392	0.1290	0.1913	0.1957	0.1408
satimage	0.8649	0.0082	0.8597	0.0146	0.8421	0.0247	0.7516	0.0211	0.8548	0.0130	0.8386	0.0129	0.8445	0.0134	0.8214	0.0215	0.8205	0.0163	0.7311	0.0333	0.8581	0.0136
segment	0.9459	0.0096	0.9313	0.0202	0.9263	0.0163	0.8702	0.0225	0.9313	0.0202	0.9318	0.0168	0.9369	0.0182	0.8596	0.0285	0.8571	0.0271	0.7187	0.0453	0.9273	0.0174
sonar	0.5223	0.1687	0.5266	0.1703	0.4818	0.2601	0.5594	0.2128	0.5266	0.1703	0.4582	0.1482	0.5294	0.1716	0.3671	0.2198	0.4167	0.2136	0.2351	0.1463	0.5354	0.1780
spambase	0.8373	0.0039	0.7262	0.0185	0.7170	0.0290	0.6136	0.0454	0.7262	0.0185	0.8016	0.0359	0.6828	0.0331	0.7195	0.0208	0.7175	0.0200	0.5468	0.0427	0.7212	0.0178
spectfheart	0.1314	0.1199	0.2046	0.2665	0.2864	0.2439	0.1405	0.0439	0.1895	0.2308	0.3549	0.1456	0.2702	0.2121	0.1679	0.3166	0.1066	0.2395	0.0000	0.0000	0.1484	0.1924
texture	0.9943	0.0016	0.9786	0.0054	0.9566	0.0052	0.8754	0.0135	0.9780	0.0061	0.9264	0.0119	0.9706	0.0060	0.9182	0.0109	0.9148	0.0155	0.6964	0.0529	0.9758	0.0060
thyroid	0.8235	0.0307	0.3216	0.0749	0.2808	0.0759	0.0764	0.0232	0.2996	0.0619	0.5177	0.0785	0.2948	0.0634	0.2549	0.0674	0.2653	0.0783	0.0000	0.0000	0.3080	0.0599
tic-tac-toe	0.4976	0.0696	0.5274	0.0733	0.5329	0.0888	0.4313	0.0897	0.5447	0.0560	0.4450	0.1232	0.2362	0.0560	0.4999	0.0784	0.5189	0.0568	0.0659	0.0699	0.5095	0.0696
titanic	0.4294	0.0443	0.2713	0.1921	0.2921	0.1991	0.4498	0.0861	0.2713	0.1921	0.4414	0.0793	0.1262	0.1190	0.2713	0.1921	0.2700	0.1939	0.2442	0.0604	0.2713	0.1921
twonorm	0.9474	0.0044	0.8878	0.0187	0.8986	0.0190	0.9516	0.0135	0.8878	0.0187	0.9403	0.0145	0.8262	0.0268	0.8811	0.0188	0.8822	0.0196	0.9146	0.0210	0.8870	0.0197
vowel	0.7630	0.0335	0.7511	0.0264	0.7322	0.0256	0.6689	0.0433	0.7678	0.0300	0.5556	0.0560	0.7678	0.0270	0.5711	0.0558	0.5778	0.0378	0.1489	0.0295	0.7367	0.0263
wine	0.9440	0.0251	0.8915	0.0826	0.8307	0.0931	0.9320	0.0504	0.9074	0.0682	0.9487	0.0560	0.9075	0.0948	0.5607	0.1858	0.5281	0.1968	0.8980	0.0905	0.7785	0.1145
wisconsin	0.9103	0.0371	0.8962	0.0967	0.9198	0.0469	0.9060	0.0516	0.8962	0.0967	0.9274	0.0612	0.9192	0.0495	0.8107	0.1226	0.7923	0.1300	0.9095	0.0521	0.8576	0.1092
zoo	0.8912	0.0539	0.9065	0.0987	0.8965	0.0877	0.9137	0.0972	0.9061	0.0988	0.8812	0.1110	0.9080	0.0789	0.7442	0.1357	0.7635	0.1346	0.8370	0.1094	0.8745	0.1311
Average	0.7002	0.0423	0.6179	0.2557	0.5999	0.2603	0.5722	0.2748	0.6150	0.2562	0.6759	0.2427	0.5758	0.2784	0.5350	0.2507	0.5414	0.2467	0.4094	0.3105	0.5933	0.2636

Testing kappas of 10 other SSL algorithms: https://sci2s.ugr.es/sites/default/files/files/ComplementaryMaterial/SelfLabeled/SelfLabeled30.ods (accessed on 16 December 2021).

Mathematics 2022, 10, 39 17 of 19

5. Conclusions and Future Work

In this paper, we proposed a maximum mean discrepancy-based semi-supervised learning (MMD-SSL) algorithm which is a data distribution-oriented SSL algorithm. The unlabeled samples were gradually labeled by considering the consistencies between the clustering and classification results. The MMD-SSL algorithm belongs to the category of self-training SSL and performs three main steps: the training of a multilayer perceptron (MLP) classifier, the clustering of unlabeled samples, and the consistent labeling of unlabeled samples. The experimental results demonstrated the rationality and effectiveness of the designed MMD-SSL algorithm by comparing it with 10 other SSL algorithms on 29 benchmark data sets. Here, we briefly summarized the technical advantages of the MMD-SSL algorithm.

- Highly confident pseudo labeling. Because the MMD criterion is used to measure the distribution consistency between the *k*-means-clustered samples and MLP-classified samples, the pseudo labeling considers both the inherent features (*k*-means clustering results) and extrinsic characteristics (MLP classification results) of unlabeled samples. This kind of pseudo labeling provides more confidence than the pseudo labeling done using only the internal or external information.
- Good generalization capability of the classifier. The MLP classifier is trained based on
 the samples with highly confident pseudo labels and thus its testing performance is
 gradually improved with the increase of training samples. The experimental results
 have demonstrated this conclusion. The highly confident pseudo labeling leads to the
 good generalization capability of the MLP classifier.
- Easy implementation. The MMD-SSL algorithm is easy to understand and implement in any programming language. Moreover, training the MMD-SSL algorithm converges with the decrease of unlabeled samples.

Future work will focus on the following three directions. First, the MMD-SSL algorithm will be implemented on a distributed computation environment and used to deal with large-scale SSL problems. Second, an ensemble version of the MMD-SSL algorithm will be developed to further enhance labeling confidence of unlabeled samples. Third, the MMD-SSL algorithm will be applied in real-world applications, e.g., the identification of harassing phone calls and the detection of abnormal power consumption behavior.

Author Contributions: Methodology, Writing-Original Draft Preparation, Formal Analysis, Q.H.; Writing-Original Draft Preparation, Writing-Review and Editing, Y.H.; Supervision, Z.H. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was supported by the Scientific Research Foundation of National Natural Science Foundation of China (61972261), Basic Research Foundation of Shenzhen (JCYJ20210324093609026), and Shenzhen University for Newly introduced Teachers (2018060).

Data Availability Statement: The data presented in this study are available in BaiduPan https://pan.baidu.com/s/1aDm8n7AA2ETtSumM5LXVBQ (accessed on 16 December 2021) with extraction code vn6j.

Acknowledgments: The authors would like to thank the editor and two anonymous reviewers who carefully read the paper and provided valuable suggestions that considerably improved the paper. They thank Philippe Fournier-Viger for helping them improve the linguistic quality of the manuscript so that it can be read more smoothly.

Conflicts of Interest: The authors declare no conflict of interest.

Acronyms

MMD Maximum mean discrepancy. SSL Semi-supervised learning.

MMD-SSL Maximum mean discrepancy-based semi-supervised learning.

MLP Multilayer perceptron. SVM Support vector machine. Mathematics 2022, 10, 39 18 of 19

SETRED Self-training with editing.

SNNRCE Self-training nearest neighbor rule using cut edges.

SSC Semi-supervised classification.

APSSC Aggregation pheromone density based semi-supervised classification.

DemoCoL Democratic co-learning.

RASCO Random subspace co-training.

Rel-RASCO Relevant RASCO.

CLCC Classification algorithm based on local cluster centers.

BNB Bernoulli naive Bayes.
GNB Gaussian naive Bayes.
SVM Support vector machines.
k-NN k-nearest neighbors.

sci2s Soft Computing and Intelligent Information Systems.
KEEL Knowledge extraction based on evolutionary learning.

CD Critical difference.

References

1. Zhu, X.J.; Goldberg, A.B. Introduction to semi-supervised learning. *Synth. Lect. Artif. Intelli. Mach. Learn.* **2009**, *3*, 1–130. [CrossRef]

- 2. Cohn, D.A.; Ghahramani, Z.; Jordan, M.I. Active learning with statistical models. J. Artif. Intell. Res. 1996, 4, 129–145. [CrossRef]
- 3. Beitzel, S.M.; Jensen, E.C.; Frieder, O.; Lewis, D.D.; Chowdhury, A.; Kolcz, A. Improving automatic query classification via semi-supervised learning. In Proceedings of the Fifth IEEE International Conference on Data Mining, Houston, TX, USA, 7–30 November 2005; pp. 8–15.
- Guillaumin, M.; Verbeek, J.; Schmid, C. Multimodal semi-supervised learning for image classification. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 902–909.
- Li, Y.; Sun, Y.; Contractor; N. Graph mining assisted semi-supervised learning for fraudulent cash-out detection. In Proceedings of the 2017 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Sydney, Australia, 31 July–3 August 2017; pp. 546–553.
- 6. Tamposis, I.A.; Tsirigos, K.D.; Theodoropoulou, M.C.; Kontou, P.I.; Bagos, P.G. Semi-supervised learning of Hidden Markov Models for biological sequence analysis. *Bioinformatics* **2019**, *35*, 2208–2215. [CrossRef] [PubMed]
- 7. Van Engelen, J.E.; Hoos, H.H. A survey on semi-supervised learning. Mach. Learn. 2020, 109, 373-440. [CrossRef]
- 8. Scudder, H. Probability of error of some adaptive pattern-recognition machines. *IEEE Trans. Inf. Theory* **1965**, *11*, 363–371. [CrossRef]
- 9. Blum, A.; Mitchell, T. Combining labeled and unlabeled data with co-training. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory, Madison, WI, USA, 24–26 July 1998; pp. 92–100.
- 10. Li, M.; Zhou, Z.H. SETRED: Self-training with editing. In Proceedings of the 2005 Pacific-Asia Conference on Knowledge Discovery and Data Mining, Hanoi, Vietnam, 18–20 May 2005; pp. 611–621.
- 11. Wang, Y.; Xu, X.; Zhao, H.; Hua, Z. Semi-supervised learning based on nearest neighbor rule and cut edges. *Knowl.-Based Syst.* **2010**, 23, 547–554. [CrossRef]
- 12. Halder, A.; Ghosh, S.; Ghosh, A. Aggregation pheromone metaphor for semi-supervised classification. *Pattern Recognit.* **2013**, *46*, 2239–2248. [CrossRef]
- 13. Wu, D.; Shang, M.; Luo, X.; Xu, J.; Yan, H.; Deng, W.; Wang, G. Self-training semi-supervised classification based on density peaks of data. *Neurocomputing* **2018**, 275, 180–191. [CrossRef]
- 14. Zhou, Y.; Goldman, S. Democratic co-learning. In Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence, Boca Raton, FL, USA, 15–17 November 2004; pp. 594–602.
- 15. Zhou, Z.H.; Li, M. Tri-training: Exploiting unlabeled data using three classifiers. *IEEE Trans. Knowl. Data Eng.* **2005**, *17*, 1529–1541. [CrossRef]
- 16. Wang, J.; Luo, S.W.; Zeng, X.H. A random subspace method for co-training. In Proceedings of the 2008 IEEE International Joint Conference on Neural Networks, Hong Kong, China, 1–8 June 2008; pp. 195–200.
- 17. Yaslan, Y.; Cataltepe, Z. Co-training with relevant random subspaces. Neurocomputing 2010, 73, 1652–1661. [CrossRef]
- 18. Huang, T.; Yu, Y.; Guo, G.; Li, K. A classification algorithm based on local cluster centers with a few labeled training examples. *Knowl.-Based Syst.* **2010**, 23, 563–571. [CrossRef]
- 19. Piroonsup, N.; Sinthupinyo, S. Analysis of training data using clustering to improve semi-supervised self-training. *Knowl.-Based Syst.* **2018**, *143*, 65–80. [CrossRef]
- 20. Wang, W.; Zhou, Z.H. A new analysis of co-training. In Proceedings of the 27 th International Conference on Machine Learning, Haifa, Israel, 21–24 June 2010; pp. 1–8.
- 21. Zhan, W.; Zhang, M.L. Inductive semi-supervised multi-label learning with co-training. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; pp. 1305–1314.

- 22. Zhou, Z.H. Disagreement-based Semi-supervised Learning. Acta Autom. Sin. 2013, 39, 1871–1878. [CrossRef]
- 23. Gretton, A.; Borgwardt, K.M.; Rasch, M.J.; Schölkopf, B.; Smola, A. A kernel two-sample test. J. Mach. Learn. Res. 2012, 13, 723–773.
- 24. He, Y.L.; Huang, D.F.; Dai, D.X.; Huang, J.Z. General bounds for maximum mean discrepancy statistics. *Math. Appl.* **2021**, 2, 284–288.
- 25. Vieira, S.M.; Kaymak, U.; Sousa, J.M. Cohen's kappa coefficient as a performance measure for feature selection. In Proceedings of the 2010 International Conference on Fuzzy Systems, Barcelona, Spain, 18–23 July 2010; pp. 1–8.
- 26. Demšar, J. Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. 2006, 7, 1–30.