Int. J. Fuzzy Syst.
https://doi.org/10.1007/s40815-019-00666-2

Impact of Fuzziness Measures on the Performance of Semi-supervised Learning

Muhammed J. A. Patwary[1] · Xi-Zhao Wang[1] · Dasen Yan[1]

Received: 18 January 2019 / Revised: 30 April 2019 / Accepted: 20 May 2019

Abstract Usage of fuzziness in the study of semi-supervised learning is relatively new. In this study, the divide-and-conquer strategy is used to investigate the performance of semi-supervised learning. To this end, testing dataset is divided into three categories, namely low, medium and high-fuzzy samples based on the magnitude of fuzziness of each sample. It is experimentally confirmed that if the low-fuzzy samples are added from the testing dataset to the original training dataset and the model is retrained, then the accuracy can be improved. To measure the amount of fuzziness of each sample, four different fuzziness measuring models are used in this study. Experimental results support that improvement of accuracy is dependent on which fuzziness measuring model is used to measure the fuzziness of each sample. Wilcoxon signed-rank test shows that choosing a specific fuzziness measuring model is significant or not. Finally, from the Wilcoxon signed-rank test, the best model is chosen, which can be used along with semi-supervised learning to improve its performance.

Keywords Fuzziness · Semi-supervised learning · Divide-and-conquer strategy · Measures of fuzziness · Fuzzy classifier · Wilcoxon signed-rank test

✉ Muhammed J. A. Patwary
jamshed@szu.edu.cn

1 Big Data Institute, College of Computer Science and Software Engineering, Key Guangdong of Intelligent Laboratory Information Processing Shenzhen University, Shenzhen 518060, Guangdong, China

1 Introduction

Semi-supervised learning (SSL) is a machine learning paradigm. After the development of supervised and unsupervised learning algorithms, machine learning researchers realized that both types of algorithms have their own advantages and disadvantages. Exploiting the advantages of both supervised and unsupervised learning algorithms and trying to remove their limitations, machine learning researchers proposed a new type of learning algorithm which is known as semi-supervised learning. Labeled data are required to train a supervised learning algorithm. On the other hand, unsupervised learning does not require labeled data. In contrast to both supervised and unsupervised algorithms, semi-supervised learning algorithm uses both labeled and unlabeled data [1–3]. In most of the cases, the type of data we encounter in our daily life activities is of unlabeled data. Generally, unlabeled data are not expensive because they do not require human expert to process this type of data. On the other hand, labeled data have some limitations: (1) It takes long time to process such data, (2) labeled data are expensive, and (3) domain information is needed to handle this type of data. So, semi-supervised learning is very useful learning technique when there are huge volume of unlabeled data and small amount of labeled data [4].

Fuzziness is a common phenomenon in our daily life activities because many events are not crisp in nature rather they are fuzzy. The term fuzziness was first proposed by Lotfi A. Zadeh in 1965 in association with his great invention fuzzy set theory [5].

With the advent of fuzzy set theory, its usages have become common in many fields such as machine learning, data mining, pattern recognition. As the concept of fuzzy set theory has been widely applied to many application areas, consequently researchers in this field felt the need for measuring the amount of fuzziness in a fuzzy sets or events. In this regard, many researchers proposed different methods to measure the fuzziness of a fuzzy set and tried to justify their methods by investigating some mathematical properties.

During the last few decades, SSL has become very popular as a learning method due to its potential advantages and numerous diversified applications [6–8]. From the literature, we find different techniques that were proposed for semi-supervised learning, such as self-training, co-training, generative model, graph-based model, transductive SVM. The existing semi-supervised learning techniques usually do not use the concept of fuzziness in the learning process [9, 10]. Therefore, the use and the investigation of fuzziness in the learning process of semi-supervised technique is very significant. The practice of the concept of fuzziness in SSL is relatively new, and it is gaining popularity among the machine learning researchers because it helps to model more generalized semi-supervised learning algorithm than its counterparts. Ashfaq et al. [11] showed how the fuzziness-based semi-supervised learning can be used to detect intruder in a system. In their study, they only used one fuzziness measuring model. As an extension to this work, we use different fuzziness measuring models and investigate how the selection of specific fuzziness measuring model impacts on the learning performance of fuzzy semi-supervised learning.

The purpose of this present work is to build a model to perform a comprehensive analysis of the effect of using various measures of fuzziness on the learning performance of fuzzy semi-supervised learning.

Our main contributions in this paper are as follows:

(1) It is experimentally shown that in a SSL, we can categorize the testing samples into three groups, namely low-, medium- and high-fuzziness samples.
(2) It is also experimentally shown that if we add the low-fuzzy instances from the test dataset to the original training dataset, then its training and testing accuracy can be improved.
(3) Improvement of training and testing accuracy is dependent on which fuzziness measuring model is used to measure the fuzziness of samples.
(4) Wilcoxon signed-rank test shows that whether it is significant to choose a specific fuzziness measuring model over others or not, and also helps to find the best model which can be used with semi-supervised learning to improve its performance.

This paper is arranged as follows. Section 1 is an introduction. Section 2 gives an overview of various measures of fuzziness. Section 3 introduces our proposed method to study the effects of different fuzziness measures in a fuzziness-based semi-supervised learning. Section 4 experimentally analyzes and discusses our results. Finally, Sect. 5 concludes the paper.

2 Background

In this section, we first discuss the basic idea of fuzziness and its mathematical properties. Then, we present several measures of fuzziness. Later, we give a brief overview of semi-supervised learning technique.

2.1 Fuzziness and Its Properties

In general, the term fuzziness means the quality of being indistinct and without sharp outlines. It was first introduced by Zadeh [5]. It describes the unclearness existing in an ill-defined event. It is a type of cognitive uncertainty because of the absence of exact boundaries of concepts. The concept of fuzziness is discussed with fuzzy set theory. Luca and Termini [12] described fuzziness as an uncertainty [13, 14] connected with fuzzy sets and used non-probabilistic entropy to measure the fuzziness associated with an event. D. Sanzhez and E. Trillas used different measures of fuzziness under different uses of fuzzy sets [15]. Ralescu and Adams [16] used the fuzzy integral of a positive and measurable function with respect to a fuzzy measure. Wang [17] proved that a measure of fuzziness satisfying some conditions can be represented as a fuzzy integral with respect to some fuzzy measures. Farhadinia, Bahram and Xu, Zeshui used different entropy measuring techniques for hesitant fuzzy sets [18]. Here, we list out some potential properties that satisfy while measuring the fuzziness of an event [19, 20]. Different authors considered different properties to measure the fuzziness.

Property 1: $d(f) = 0$ if and only if $f(x) \in \{0, 1\}$ for all $x \in X$. This property is called sharpness property, because $f(x)$ takes only crisp values instead of fractional values.

Property 2: $d(f)$ is maximum if and only if $f(X) = \{0.5\}$. This property is called maximality property, because it describes the maximum value of a fuzzy set.

Property 3: $d(f^*) \leq d(f)$, if f^* is any sharpened version of f, that is $f(x^*) \leq f(x)$ if $f(x) \leq 0.5$ and is $f(x^*) \geq f(x)$ if $f(x) \geq 0.5$. This property is known as resolution property of fuzzy set.

Property 4: $d((f^-) = d(f)$. The symmetrical property tells the degree of fuzziness is symmetrical about 0.5 and takes values across 0 to 1. For example, 0.4 and 0.6 have the same degree of fuzziness.

Property 5: $d(f \cup g) + d(f \cap g) = d(f) + d(g)$.

Property 6: There exist mappings $s, t : [0, 1] \to [0, \infty)$ such that: $d(fxg) = d(f) \cdot t(P(g)) + d(g) \cdot s(P(f))$ For all

$f \in [0,1]^X$ and $g \in [0,1]^Y$, X and Y are any finite sets.

2.2 Various Measures of Fuzziness

Measures of fuzziness mean the degree of fuzziness of a fuzzy set. In the literature, various approaches have been proposed to measure the fuzziness of a fuzzy set. Most of the approaches are influenced by the famous Shannon entropy formula to measure the amount of information. In this subsection, we give an overview of various measures of fuzziness along with their properties.

In 1972, DeLuca and Termini proposed the following model to measure the magnitude of fuzziness of a fuzzy event [12].

$$d(f) = -K \cdot \sum_{i=1}^{N} f(x_i) \cdot \log(f(x_i)) + (1 - f(x_i)) \cdot \log(1 - f(x_i)) \tag{1}$$

where N denotes the number of elements of $f(x)$ and K is a positive integer. Equation (1) satisfies the properties from 1 to 5, previously listed in Sect. 2.1:

In 1975, Kaufmann proposed the following models to measure the magnitude of fuzziness of a fuzzy event, where he used the generalized relative Hamming distance and Euclidean distance, respectively, in Eqs. (2) and (3) [21]. Equations (2) and (3) satisfy the properties from 1 to 5, previously listed in Sect. 2.1:

$$d(f) = \frac{2}{N} \cdot \sum_{i=1}^{N} |f(x_i) - f_{\frac{1}{2}}(x_i)| \tag{2}$$

$$d(f) = \frac{2}{N^{\frac{1}{2}}} \left\{ \sum_{i=1}^{N} (f(x_i) - f_{\frac{1}{2}}(x_i))^2 \right\}^{\frac{1}{2}} \tag{3}$$

Later in 1983, Ebanks proposed the following model to measure the magnitude of fuzziness of a fuzzy set [22]. Equation (4) satisfies the properties from 1 to 6, previously listed in Sect. 2.1:

$$d(f) = \sum_{i=1}^{N} f(x_i) \cdot (1 - f(x_i)) \tag{4}$$

where $f \in [0,1]^X$.

2.3 A Brief Overview of Semi-supervised Learning

Semi-supervised learning is a machine learning strategy. It is considered as halfway between supervised and unsupervised learning techniques which takes advantages of both learning techniques. In 1965, H. J. Scuder first introduced the concept of SSL [23]. We know that unlabeled data are common and less expensive compared to labeled data. As unlabeled data are available and labeled data are rare, to deal with the situation machine learning researchers tried to find some techniques. They found semi-supervised learning can properly handle this problem. In order to solve this problem, they train a classifier with small amount of labeled data and using that classifier, they classify huge amount of unlabeled data. Semi-supervised learning has been effectively used to solve many real-world problems from the beginning of its development [24–28].

Suppose, we have n number of samples; out of them there are l number of labeled data and the rest are unlabeled data. So, the labeled data are $(x_l, y_l) = \{x_{1:l}, y_{1:l}\}$ and unlabeled data are $x_u = \{x_{l+1:n}\}$. Then, a classifier function is defined as $f : x \rightarrow y$ to classify unlabeled data. Over the last couple of decades, many SSL techniques were proposed by many researchers, such as self-training, co-training, multi-view learning, graph-based method, fuzziness-based semi-supervised learning.

2.3.1 *Self-training*

Self-training is perhaps the oldest type of semi-supervised learning technique. In self-training, a classifier is built from a small amount of labeled data. Then, this classifier is used to classify huge volume of unlabeled data. Later, this newly labeled data are added to the old trained data to form a new training dataset and retrain the model as long as certain criteria are fulfilled. The steps of self-training are as follows: (Algorithm 1)

Algorithm 1 Self-training algorithm

1: Use the labeled samples, (x_l, y_l) to train a classifier f.
2: Use the classifier, f to classify unlabeled samples, $x \in x_u$.
3: Add newly added labeled samples, $(x, f(x))$ to old training samples.
4: Repeat as long as the prediction accuracy is improved.

Self-training algorithms is used to solve different types of machine learning problems [29]. Wu et al. proposed a self-training-based semi-supervised learning to classify data based on density peaks [30]. Yarosky [31] used self-training in some natural language processing (NLP) problems. Riloff et al. [32] proposed self-training algorithm to find the subjective nouns from a document.

2.3.2 Co-training

Co-training [33, 34] is another popular semi-supervised learning technique. In some real situations, data have different views. For example, both an image and an HTML text can describe the same item. When there are two views of the same item, co-training can be used by two classifiers to classify data. Steps of co-training algorithm are summarized as follows: (Algorithm 2)

Algorithm 2 Co-training algorithm

1: Construct two classifiers f_1 and f_2 respectively from $(x_l^{(1)}, y_l)$ (1st view of labeled data) and $(x_l^{(2)}, y_l)$ (2nd view of labeled data).
2: Classify unlabeled samples, x_u by f_1 and f_2 independently.
3: Add f_1's most confirmed samples of $(x, f_1(x))$ to $f_{(2)}$'s labeled data and $f_{(2)}$'s most confirmed samples of $(x, f_{(2)}(x))$ to $f_{(1)}$'s labeled data.
4: Repeat as long as the prediction accuracy is improved.

Maeireizo et al. [35] used co-training algorithm to predict emotions with spoken language data. Chawla et al. [3] used co-training algorithm where they used both labeled and unlabeled data.

2.3.3 Multi-view Learning

Multi-view learning [36, 37] is the extension of co-training where more than two classifiers are trained with small amount of labeled data and these classifiers are used to provide labels to unlabeled data. For example, there are three different views of the same item. For example, an image, an HTML text and an audio all three can describe the same item. So, multi-view learning can be used to model this type of phenomenon. Three classifiers are trained based on the three views of labeled data, and unlabeled data are classified by using those three classifiers. Zhou et al. [38] proposed tri-training to exploit unlabeled data using three classifiers. D Kim et al. proposed a multi-co-training for document classification using various document representations [39].

2.3.4 Graph-Based Methods

Graph-based method [40, 41] is one of the important semi-supervised learning methods. In this technique, a graph is used to represent the problem where the set of vertices V represents the training samples and the set of edges E represents the connection between two samples and the weight $W_{i,j}$ represent the closeness of two sample i and j. Zhou et al. [42] used directed graph to learn from labeled and unlabeled data. Blum and Chawla [43] used graph mincuts to learn from both labeled and unlabeled data.

2.3.5 Fuzziness-Based Semi-supervised Learning

Suppose there is a dataset D, and most of the samples of D have no labels and a few of them have labels. A classifier can be built from labeled samples, and this classifier will be used to classify unlabeled samples. Wang and He [44] proposed the fuzziness-based semi-supervised learning algorithm which has the following steps: (Algorithm 3)

Algorithm 3 Fuzziness based semi-supervised leaning

1: Randomly split the dataset into a training and a testing set.
2: Build a classifier based on the training set.
3: For each sample from both training and testing set, get the fuzzy vector output provided by the base classifier.
4: Calculate the fuzziness for each sample.
5: Sort the samples and categorize them into three groups namely low, medium and high fuzziness based on the magnitude of fuzziness.
6: Add the low fuzziness sample from testing samples to the original training samples.
7: Retrain the model.

Ashfaq et al. [11] proposed a fuzziness-based semi-supervised learning technique to detect intruder in a system. In their work, they demonstrated that when the low-fuzziness samples from the testing dataset are added to the training dataset, the classification rate for intruder detection system (IDSs) is high and when the medium-fuzziness samples are added, there is a higher risk of misclassification. Patwary and Wang [45] analyzed the sensitivity of initial classifier accuracy in a fuzziness-based semi-supervised learning.

3 Effects of Different Measures of Fuzziness on the Fuzziness-Based SSL

In this section, we first describe the fuzziness-based divide-and-conquer strategy. Then, we briefly discuss about two fuzzy classifiers and four different fuzziness measuring methods that can be used along with our proposed fuzziness-based SSL to study the effect of different measures of fuzziness on the learning performance of fuzzy SSL. Finally, we describe the Wilcoxon signed-rank test to test whether selecting a specific fuzziness measuring model is statistically significant or not. Wilcoxon signed-rank test helps to select the best model from all the models.

3.1 Fuzziness-Based Divide-and-Conquer Strategy

Wang et al. [46] proposed a divide-and-conquer strategy to deal with fuzziness in a classification type of problem where they divided the dataset into the training set and the testing set. Then, based on the magnitude of fuzziness the testing data are grouped into three categories, namely low-, medium- and high-fuzziness groups. In their work, they used one model to measure the fuzziness of fuzzy set. As an extension of the work of Wang et al., we use four different fuzziness measuring models and proposed our new algorithm to study the significance of using different fuzziness measuring models in a fuzzy SSL (Algorithm 4).

There are many classifiers that give fuzzy vector output. We randomly choose fuzzy KNN and non-iterative single hidden layer feed-forward neural network, i.e., fuzzy ELM to exhibit the effectiveness of our proposed algorithm.

3.1.1 Fuzzy KNN

Fuzzy KNN is the more generalized version of its counterparts, that is, crisp KNN. Unlike crisp KNN algorithm, it gives the output as a membership vector. Instead of giving 0 or 1 type of output, it gives output as fraction between 0 and 1 which says to what extent the instance belongs to a specific class. The following formula provides the class memberships to a given sample [47].

$$u_i(x) = \frac{\sum_{j=1}^{K} u_{ij}(1/||x - x_j||^{2/(m-1)})}{\sum_{j=1}^{K}(1/||x - x_j||^{2/(m-1)})} \tag{5}$$

3.1.2 Non-iterative Single Hidden Layer Feed-Forward Neural Network

This subsection briefly describes the non-iterative single hidden layer feed-forward neural network, that is, extreme learning machine (ELM) [48–51]. ELM consists of three layers, namely input, output and hidden layers. Weights between input and output layers are given randomly, and weights between hidden and output layers are obtained

Algorithm 4 New fuzziness based semi-supervised learning algorithm

Input: Dataset.

Output: Training and testing accuracy respectively before and after adding low fuzziness samples from testing dataset to the original training dataset.

Randomly partition the dataset into a training dataset $\mathbf{X}_{tr}$ and a testing dataset $\mathbf{X}_{te}$.
$k \leftarrow 1$
while $k <= 4$ **do**
  Train the classifier $\mathbf{C}$ according to a training algorithm(FKNN/FELM).
  Get the training accuracy tr_{accB} and testing accuracy te_{accB}
  Get the fuzzy vector $f = \{f(x_1), f(x_2), \cdots, f(x_n)\}$ for each sample in testing set by classifier $\mathbf{C}$.
  Calculate the fuzziness $d(f)$ of each sample in testing set by one of the following 4 models
  if $k = 1$ **then**
    $d(f) = -K \cdot \sum_{i=1}^{N} f(x_i) \cdot log(f(x_i)) + (1 - f(x_i)) \cdot log(1 - f(x_i))$ //Model-1
  else if $k = 2$ **then**
    $d(f) = \frac{2}{N} \cdot \sum_{i=1}^{N} |f(x_i) - f_{\frac{1}{2}}(x_i)|$ //Model-2
  else if $k = 3$ **then**
    $d(f) = \frac{2}{N^{\frac{1}{2}}}\{\sum_{i=1}^{N}(f(x_i) - f_{\frac{1}{2}}(x_i))^2\}^{\frac{1}{2}}$ //Model-3
  else
    $d(f) = \sum_{i=1}^{N} f(x_i) \cdot (1 - f(x_i))$ //Model-4
  end if
  Sort the samples by the fuzziness $d(f)$ and group the testing set $\mathbf{X}_{te}$ into three fractions: $\mathbf{X}_{te}low$, $\mathbf{X}_{te}medium$ and $\mathbf{X}_{te}high$.
  Get new training set $\mathbf{X}_{tr}new$ by adding the low-fuzziness samples $\mathbf{X}_{te}low$ to the original training set $\mathbf{X}_{tr}$.
  Retrain a new classifier $\mathbf{C}_{new}$ according to the given training algorithm with $\mathbf{X}_{tr}new$.
  Again record the training accuracy tr_{accA} and testing accuracy te_{accA} by classifier $\mathbf{C}_{new}$ with $\mathbf{X}_{tr}new$
  $k = k + 1$
end while

analytically. For a training set $\aleph = \{(x_i, t_i)|x_i \in R^n, t_i \in R^m, i = 1, 2, \ldots, N\}$ Activation function $h(x)$ and number of hidden layer node M, ELM algorithm summarizes as follows:

Step 1 Arbitrarily allocate the input weight w_i and the bias b_i, where, $i = 1, 2, \ldots, M$

Step 2 Compute the hidden layer output matrix H.

Step 3 Compute the output weight β where, $\beta = (H^T H)^{-1} H^T T$.

Using Algorithm 1, we can get the training and testing accuracies, respectively, before and after adding low-fuzziness samples from testing dataset to the original training dataset. In this algorithm, we use 4 different fuzziness measuring models. We want to compare the performance of these 4 models. To this end, we use Wilcoxon signed rank rest.

3.2 Wilcoxon Signed-Rank Test to Compare Fuzziness Measuring Models

Wilcoxon signed-rank test is a nonparametric test which does not follow any distribution. This test is applicable when the sample size is very small. Suppose the sample size is S. So, there are $2S$ data points. Data points are denoted by $x_{1,i}$ and $x_{2,i}$ where $i = 1, 2, \cdots, S$. In Wilcoxon signed-rank test [52, 53], we have the following hypotheses.

H_0: New model does not lower the scores. H_1: New model lowers the scores.

random weight non-iterative single layer feed-forward neural network also called (FELM), respectively, in the first and second experiments. In each of the experiment, we use 4 different fuzziness measuring models to see the effect of using each of the fuzziness measuring model on the performance of fuzzy SSL. Tables 2 and 3 represent the results of our proposed algorithm when FKNN and FELM are, respectively, used as base classifiers.

The algorithms are implemented using Python 3 software, and the experiments are conducted on a PC with Windows 10 operating system, an Intel core i5-4590 CPU, 3.30 GHz and a 12 GB RAM.

Table 2 describes the first experimental results of our proposed algorithm when FKNN is used as the base classifier. Column 2 and column 3 of Table 2 report the initial training and testing accuracies when first (model-1) fuzziness measuring model is used to measure the fuzziness of each sample. Then, the test samples are categorized into three groups based on the magnitude of fuzziness and low-fuzziness samples are added from the testing dataset to the original training dataset. We retrain the model with new training dataset. Then, both training and testing accuracies are improved for all 12 datasets. And the improved training and testing accuracies are represented in the column 4 and column 5 of Table 2.

Column 6 and column 7 of Table 2 show the initial training and testing accuracies when second (model-2) fuzziness measuring model is used. After adding low-fuzziness samples from the testing dataset to the training dataset, the model is retrained. Then, new training and testing accuracies are represented in the column 8 and column 9 of Table 2. It is clear that training accuracy is improved for all 12 datasets, but testing accuracy is improved only for Blood Transfusion Service Center (BTSCD) dataset and Pima-Indians-diabetes (PID) dataset and for all other datasets testing accuracy is not improved.

The initial training and testing accuracies are represented in the column 10 and column 11 of Table 2 when

Algorithm 5 Wilcoxon signed rank test to compare fuzziness measuring models

Input: Sample size S, $2S$ number of data points

Output: Null hypothesis is either accepted or rejected.

$i \leftarrow 1$

while $i <= S$ **do**

  Compute $d_i = |x_{2,i} - x_{1,i}|$ and $sign(x_{2,i} - x_{1,i})$

  Remove the pairs when $d_i = 0$. Suppose S_r be the new sample size.

  Sort out the new samples, S_r from the smallest value of d_i to the largest value.

  Ranking of pairs, start with 1 when d_i is the smallest. Ties accept a rank equal to the average of the ranks they span. Let T_i indicate the rank.

  Let A indicates the sum of T_i if $sign(x_{2,i} - x_{1,i}) \geq 0$ and B indicates the sum of T_i if $sign(x_{2,i} - x_{1,i}) < 0$. Compute the test statistic W, $W = min(abs(A), abs(B))$

  $i = i + 1$

end while

W is compared to a critical value from a reference table. Null hypothesis H_0 is rejected if $|W| < W_{critical,S_r}$.

4 Experimental Analysis and Discussion

To conduct the experiments, 12 standard datasets were taken from UCI machine learning repository [54]. Datasets were preprocessed before actual use. Description of the datasets is given in Table 1. Two different experimental setups were built to demonstrate the effectiveness of our proposed algorithm. We used fuzzy KNN (FKNN) and

Table 1 Dataset description

No.	Dataset	# Instances	# Features	# Classes
1	Blood Transfusion Service Center dataset (BTSCD)	749	4	3
2	Indian Liver Patient dataset (ILPD)	582	10	2
3	Phishing dataset	1354	9	3
4	Pima-Indians-diabetes (PID)	769	8	2
5	Iris	150	4	3
6	Wine-quality-white (WQW)	4898	11	7
7	Yeast	1484	8	10
8	Vehicle	846	18	4
9	Ecoli	336	7	8
10	Sonar	208	60	2
11	Parkinson	195	22	2
12	Glass	214	9	6

Table 2 FKNN is used as base classifier with 4 fuzziness measuring models

Dataset	Model-1				Model-2				Model-3				Model-4			
	Initial		Improved		Initial		Improved		Initial		Improved		Initial		Improved	
	Tr_a	Te_a	Tr_a	Te	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a
BTSCD	0.933	0.713	0.934	0.760	0.921	0.713	0.934	0.730	0.933	0.680	0.935	0.640	0.923	0.647	0.929	0.680
ILPD	0.861	0.632	0.865	0.667	0.852	0.650	0.853	0.615	0.865	0.675	0.873	0.615	0.854	0.667	0.855	0.731
Phishing	0.930	0.856	0.936	0.922	0.933	0.867	0.938	0.817	0.924	0.867	0.934	0.811	0.937	0.882	0.939	0.950
PID	1.000	0.675	1.000	0.735	1.000	0.721	1.000	0.725	1.000	0.721	1.000	0.637	1.000	0.695	1.000	0.765
Iris	0.975	0.933	0.977	1.000	0.950	0.967	0.962	0.950	0.967	0.967	0.962	0.950	0.958	1.000	0.962	1.000
WQW	0.709	0.577	0.713	0.614	0.701	0.588	0.707	0.551	0.703	0.580	0.709	0.559	0.707	0.583	0.710	0.651
Yeast	0.741	0.569	0.741	0.657	0.734	0.576	0.738	0.530	0.738	0.582	0.745	0.500	0.749	0.559	0.750	0.576
Vehicle	0.818	0.659	0.827	0.761	0.808	0.653	0.822	0.549	0.821	0.688	0.831	0.593	0.818	0.747	0.829	0.832
Ecoli	0.892	0.912	0.897	0.978	0.888	0.809	0.890	0.711	0.881	0.868	0.886	0.822	0.896	0.824	0.900	0.889
Sonar	0.904	0.833	0.917	0.964	0.928	0.738	0.933	0.714	0.928	0.786	0.928	0.714	0.946	0.833	0.950	0.929
Parkinson	0.968	0.846	0.970	0.962	0.962	0.923	0.964	0.885	0.962	0.949	0.964	0.923	0.929	0.923	0.935	0.962
Glass	0.743	0.605	0.746	0.643	0.725	0.512	0.757	0.393	0.749	0.651	0.757	0.571	0.731	0.674	0.751	0.821
Average	0.873	0.734	0.877	0.805	0.867	0.726	0.875	0.681	0.872	0.751	0.877	0.695	0.871	0.753	0.876	0.815

Here, Tr_a and Te_a, respectively, indicate the training accuracy and the testing accuracy

third (model-3) fuzziness measuring model is used. The model is retrained after adding low-fuzziness samples from the testing dataset to the training dataset. Then, new training and testing accuracies are given in the column 12 and column 13 of Table 2. It is obvious that training accuracy is improved for all 12 datasets, but testing accuracy is improved for none of the datasets.

Column 14 and column 15 of Table 2 report the initial training and testing accuracies when fourth (model-4) fuzziness measuring model is used to measure the fuzziness of each sample. Then, the test samples are categorized into three groups based on the magnitude of fuzziness and low-fuzziness samples are added from the testing dataset to the original training dataset. We retrain the model with new training dataset. Then, both training and testing accuracies are improved for all 12 datasets. And the improved training and testing accuracies are represented in the column 16 and column 17 of Table 2.

Table 3 shows the second experimental results of our proposed algorithm where FELM is used as the base classifier. Four fuzziness measuring models are used in this experiment. It is noticeable that when we add the low-fuzziness samples from the testing dataset to the original training dataset and retrain the algorithm, the training accuracy is improved for all 12 datasets, but testing accuracy is only improved for some of the datasets. For example, if we consider Indian Liver Patient Dataset (ILPD) and model-2 is used to measure the fuzziness of samples, then although the training accuracy is improved, testing accuracy is not improved.

Table 3 FELM is used as base classifier with 4 fuzziness measuring models

Dataset	Model-1				Model-2				Model-3				Model-4			
	Initial		Improved		Initial		Improved		Initial		Improved		Initial		Improved	
	Tr_a	Te_a	Tr_a	Te	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a	Tr_a	Te_a
BTSCD	0.773	0.767	0.779	0.800	0.771	0.763	0.778	0.820	0.774	0.760	0.759	0.690	0.778	0.747	0.787	0.830
ILPD	0.749	0.735	0.766	0.833	0.760	0.684	0.768	0.628	0.758	0.726	0.758	0.577	0.747	0.752	0.770	0.846
Phishing	0.898	0.856	0.902	0.933	0.898	0.852	0.898	0.817	0.886	0.845	0.888	0.794	0.897	0.882	0.905	0.956
PID	0.788	0.734	0.798	0.814	0.781	0.734	0.785	0.676	0.787	0.734	0.782	0.657	0.787	0.792	0.798	0.843
Iris	0.942	0.933	0.954	1.000	0.942	0.917	0.954	0.900	0.942	1.000	0.946	1.000	0.958	0.933	0.962	0.950
WQW	0.566	0.555	0.574	0.613	0.574	0.549	0.575	0.554	0.572	0.540	0.573	0.519	0.577	0.536	0.578	0.570
Yeast	0.624	0.572	0.632	0.616	0.633	0.579	0.641	0.601	0.628	0.609	0.630	0.540	0.621	0.606	0.631	0.652
Vehicle	0.879	0.759	0.880	0.858	0.882	0.771	0.888	0.805	0.882	0.800	0.884	0.796	0.879	0.800	0.881	0.841
Ecoli	0.914	0.912	0.921	0.933	0.914	0.824	0.921	0.800	0.903	0.838	0.910	0.778	0.929	0.779	0.930	0.800
Sonar	0.831	0.571	0.833	0.571	0.837	0.667	0.839	0.821	0.831	0.714	0.856	0.536	0.843	0.643	0.844	0.679
Parkinson	0.981	0.872	0.988	0.923	0.962	0.949	0.964	0.923	0.961	0.897	0.976	0.846	0.968	0.897	0.970	1.000
Glass	0.760	0.674	0.778	0.679	0.758	0.535	0.762	0.429	0.725	0.767	0.730	0.679	0.749	0.698	0.762	0.699
Average	0.809	0.745	0.817	0.798	0.809	0.735	0.814	0.731	0.804	0.769	0.808	0.701	0.811	0.755	0.818	0.805

Here, Tr_a and Te_a, respectively, indicate the Training accuracy and the Testing accuracy

Both in the Figs. 1 and 2, the $x-axis$ represents the datasets and the $y-axis$ represents the improvement of accuracy. We find the improvement of accuracy by subtracting the initial accuracy from the improved accuracy. In our first and second experiments, FKNN and FELM are, respectively, used as base classifiers and 4 fuzziness measuring models are used along with them. When the low-fuzziness samples are added from the testing dataset to the original training dataset and retrain the model, both the training and the testing accuracies may be improved. In Figs. 1 and 2, for all 12 datasets, if we use, respectively, model-1 and model-4 to measure the fuzziness of samples then the training and the testing accuracies are improved. But when we use, respectively, model-2 and model-3, then only training accuracy is improved, but testing accuracy is improved only for some datasets.

As in the case of model-1 and model-4, both the training and the testing accuracies are improved and in the case of

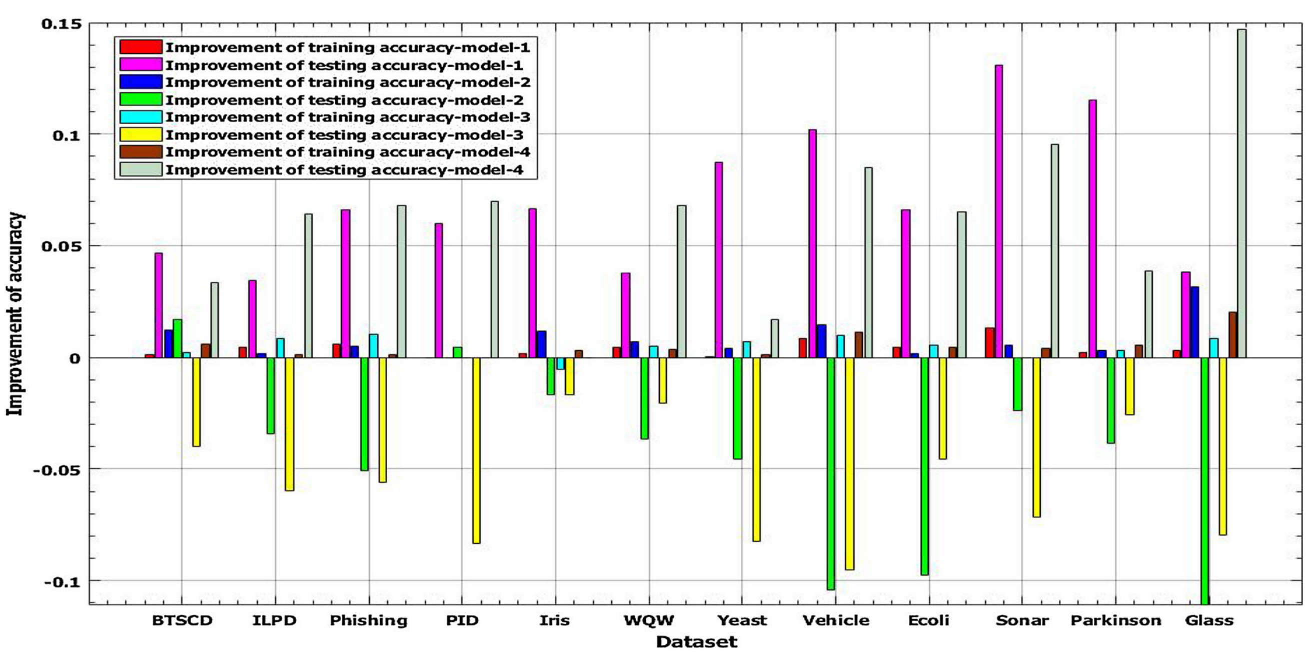

Fig. 1 Comparison of 4 fuzziness measuring models when used with FKNN as base classifier in fuzzy SSL

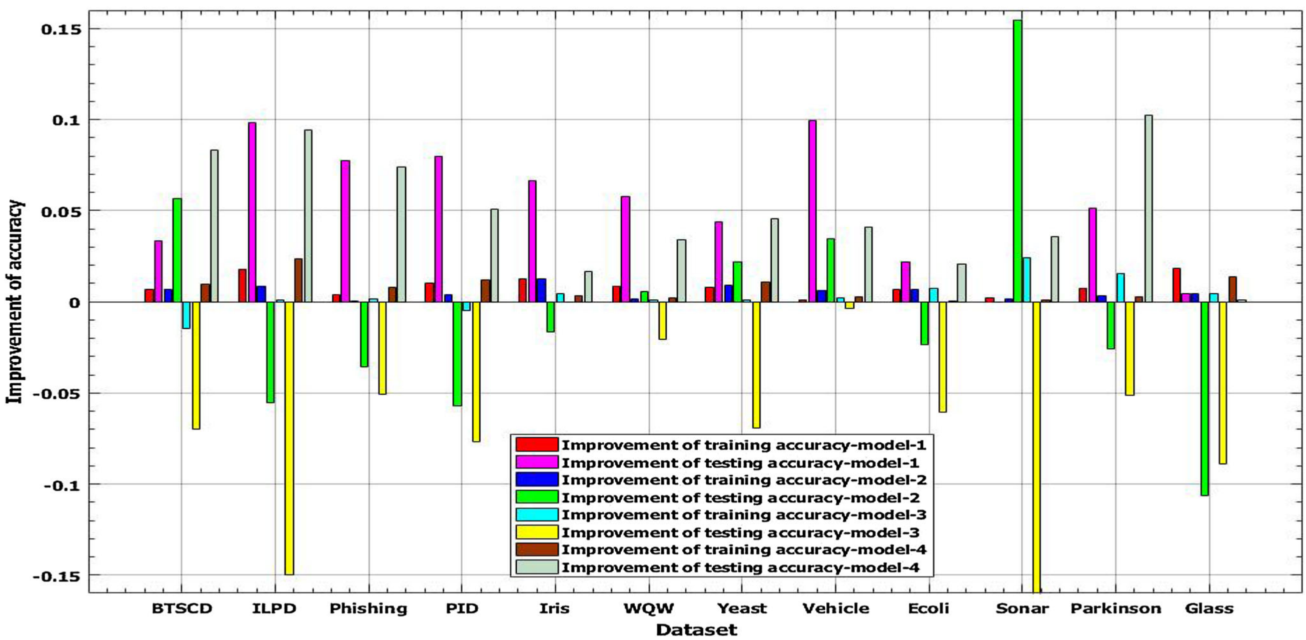

Fig. 2 Comparison of 4 fuzziness measuring models when used with FELM as base classifier in fuzzy SSL

model-2 and model-3, only the training accuracy is improved, but the testing accuracy is not improved. So, we can say that model-1 and model-4 are better than model-2 and model-3. Now, we need to check whether the difference between model-1 and model-4 is statistically significant or not. To this end, we use Wilcoxon signed-rank test.

4.1 Wilcoxon Signed-Rank Test to Compare Fuzziness Measuring Models

To compare among model-1, model-2, model-3 and model-4 using Wilcoxon signed-rank test, we have the following hypotheses:

H_0 New model to measure the fuzziness does not lower the scores.

H_1 New model to measure the fuzziness lowers the scores.

For example, if we want to compare between FKNN-model-1 and FKNN-model-2 taking initial training accuracy into account (i.e., second and sixth columns of Table 2), then the Wilcoxon signed-rank test results are given in Table 4. In Table 4, the absolute value of the positive sum of signed rank is 64 and the absolute value of the negative sum of signed rank is 13. So the test statistic, $W = 13$. From the table of critical values for the Wilcoxon test, the critical value is 11 (when level of significance is 0.05 and $N = 11$). Since the test statistic is higher than the critical value, that is, $13 > 11$, the null hypothesis is accepted and we can say that there is enough evidence to support the claim that the new model (FKNN-model-2) does not lower the score, i.e., when we use FKNN-model-2 to measure the fuzziness of samples, then the training accuracy is not lower than that of FKNN-model-1. In the same way, one can easily verify the results given in Table 5 to compare among model-1, model-2, model-3 and model-4 used with FKNN and FELM.

From Table 5, it is clear that if we compare between model-1 and model-4 taking both training and testing accuracies into account, for every case the null hypotheses are accepted. Therefore, these two models are not statistically different. We can choose any one of the models to measure the fuzziness of samples and use it with base classifier to improve the performance of our proposed semi-supervised learning algorithm.

4.2 Wilcoxon Signed-Rank Test to Compare Initial Classifiers

To compare between FKNN and FELM, used with the same four fuzziness measuring models, we use Wilcoxon signed-rank test.

H_0 New model to measure the fuzziness does not lower the scores.

H_1 New model to measure the fuzziness lowers the scores.

For example, if we want to compare between FKNN-model-1 and FELM-model-1 taking initial training accuracy into account (i.e., second column of Table 2 and second column of Table 3), then the Wilcoxon signed-rank test results are given in Table 6. In Table 6, the absolute

Table 4 Wilcoxon signed-rank test to compare between FKNN-model-1 and FKNN-model-2

Dataset	FKNN-model-1	FKNN-model-2	Sign	Absolute value	Signed rank
BTSCD	0.933110368	0.921404682	1	0.011705686	9
ILPD	0.860515021	0.85193133	1	0.008583691	7
Phishing	0.929759704	0.933456562	− 1	0.003696858	− 2
PID	1	1	0	0	0
Iris	0.975	0.95	1	0.025	12
WQW	0.708524758	0.700612557	1	0.0079122	6
Yeast	0.740522325	0.733782645	1	0.00673968	5
Vehicle	0.818047337	0.807692308	1	0.01035503	8
Ecoli	0.891791045	0.888059701	1	0.003731343	3
Sonar	0.903614458	0.927710843	− 1	0.024096386	− 11
Parkinson	0.967948718	0.961538462	1	0.006410256	4
Glass	0.742690058	0.725146199	1	0.01754386	10

Table 5 Wilcoxon signed-rank test for comparing among models

Old model	New model	|W|	Critical value	Decision
FKNN-model-1 (Tr_acc)	FKNN-model-2 (Tr_acc)	13	11	Null hypothesis is accepted
FKNN-model-1 (Tr_acc)	FKNN-model-3 (Tr_acc)	29	8	Null hypothesis is accepted
FKNN-model-1 (Tr_acc)	**FKNN-model-4 (Tr_acc)**	**29**	**8**	**Null hypothesis is accepted**
FKNN-model-1 (Te_acc)	FKNN-model-2 (Te_acc)	0	14	Null hypothesis is rejected
FKNN-model-1 (Te_acc)	FKNN-model-3 (Te_acc)	0	14	Null hypothesis is rejected
FKNN-model-1 (Te_acc)	**FKNN-model-4 (Te_acc)**	**40**	**8**	**Null hypothesis is accepted**
FELM-model-1 (Tr_acc)	FELM-model-2 (Tr_acc)	29	6	Null hypothesis is accepted
FELM-model-1 (Tr_acc)	FELM-model-3 (Tr_acc)	32	11	Null hypothesis is accepted
FELM-model-1 (Tr_acc)	**FELM-model-4 (Tr_acc)**	**32**	**11**	**Null hypothesis is accepted**
FELM-model-1 (Te_acc)	FELM-model-2 (Te_acc)	14	11	Null hypothesis is accepted
FELM-model-1 (Te_acc)	FELM-model-3 (Te_acc)	0	8	Null hypothesis is rejected
FELM-model-1 (Te_acc)	**FELM-model-4 (Te_acc)**	**31**	**14**	**Null hypothesis is accepted**

Table 6 Wilcoxon signed-rank test to compare between FKNN-model-1 and FELM-model-1

Dataset	FKNN-model-1	FELM-model-1	Sign	Absolute value	Signed rank
BTSCD	0.933110368	0.772575	1	0.160535117	11
ILPD	0.860515021	0.748927	1	0.111587983	8
Phishing	0.929759704	0.898336	1	0.03142329	4
PID	1	0.788274	1	0.211726384	12
Iris	0.975	0.941667	1	0.033333333	5
WQW	0.708524758	0.565595	1	0.142930066	10
Yeast	0.740522325	0.624263	1	0.116259478	9
Vehicle	0.818047337	0.878698	− 1	0.060650888	− 6
Ecoli	0.891791045	0.914179	− 1	0.02238806	− 3
Sonar	0.903614458	0.831325	1	0.072289157	7
Parkinson	0.967948718	0.980769	− 1	0.012820513	− 1
Glass	0.742690058	0.760234	− 1	0.01754386	− 2

value of the positive sum of signed rank is 66 and the absolute value of the negative sum of signed rank is 12. So the test statistic, $W = 12$. From the table of critical values for the Wilcoxon test, the critical value is 14 (when level of significance is 0.05 and N = 12). Since the test statistic is lower than the critical value, that is, $12 < 14$, the null hypothesis is rejected and we can say that there is enough evidence to support the claim that new model (FELM-model-1) lowers the score, i.e., when we use FELM-model-1 to measure the fuzziness of samples, the training accuracy is lower than that of FKNN-model-1. In the same way, one can easily compare between FKNN and FELM with different fuzziness measuring models.

5 Conclusions and Future Works

In this study, the divide-and-conquer strategy was used to investigate the performance of semi-supervised learning. In order to improve the performance of semi-supervised learning, testing dataset was categorized into three groups, namely low-, medium- and high-fuzzy samples according to the amount of fuzziness of each sample. It was experimentally observed that adding the low-fuzzy samples from the testing dataset to the original training dataset could improve the accuracy of SSL. Four different fuzziness measuring models were used in this study. Experimental results confirmed that the improvement of accuracy of SSL was largely dependent on which fuzziness measuring model was used to measure the fuzziness of each sample. Wilcoxon signed-rank test was used to compare among the four different fuzziness measuring models, and it was found that model 1 and model 4 were better than model 2 and model 3 and it was also found that model 1 and model 4 were not statistically different, so either of the two models could be used with SSL to improve its performance. In our proposed SSL technique, FKNN and FELM were used as the base classifiers, and in our future work, we will try to use different base classifiers to investigate whether it is significant to select one base classifier on the learning performance of SSL or not. The classifiers, used in this study, have some parameters. For example, in FKNN number of K is a parameter which can be changed and in FELM the number of hidden layer nodes and biases are two parameters which also can be changed. We think if we change these parameters, then it will certainly have some impacts on the learning performance of semi-supervised learning. It is still needed to do some experiments to investigate the impact which we will try to solve in our future work. Although it is experimentally observed that adding low-fuzziness samples from the testing dataset to the original training dataset could improve the performance of SSL, in our future work we will try to establish a strong mathematical model to explain this phenomenon.

Acknowledgements This work was supported in part by the National Natural Science Foundation of China (Grants 61772344, 61732011, 71371063 and 61811530324) and in part by Basic Research Project of Knowledge Innovation Program in ShenZhen (JCYJ2018030512585 0156).

References

1. Zhu, X., Goldberg, A.: Introduction to semi-supervised learning. Synth. Lect. Artif. Intell. Mach. Learn. **3**(1), 1–130 (2009)
2. Seeger, M.: Learning with Labeled and Unlabeled Data (Tech. Rep.). Edinburgh, UK: Institute for Adaptive and Neural Computation, University of Edinburgh (2000)
3. Chawla, N.V., Karakoulas, G.: Learing from labeled and unlabeled data: an empirical study across techniques and domain. J. Artif. Intell. Res. **23**, 331–366 (2005)
4. Zhou, Z.-H., Zhan, D.-C., Yang, Q.: Semi-supervised learning with very few labeled training examples. In: AAAI, pp. 675–680 (2007)
5. Zadeh, L.A.: Probability measures of fuzzy events. J. Math. Anal. Appl. **23**(2), 421–427 (1968)
6. Miyato, T., Maeda, S., Ishii, S., Koyama, M.: Virtual adversarial training: a regularization method for supervised and semi-supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. (2018). https://doi.org/10.1109/tpami.2018.2858821
7. Li, C., Zhu, J., Zhang, B.: Max-margin deep generative models for (semi-) supervised learning. IEEE Trans. Pattern Anal. Mach. Intell. **40**(11), 2762–2775 (2018)
8. Pratama, M., Lughofer, E., Lim, C.P., Rahayu, W., Dillon, T., Budiyono, A.: pClass+: a novel evolving semi-supervised classifier. Int. J. Fuzzy Syst. **19**(3), 863–880 (2017)
9. Zhou, W., Qiao, S., Yi, Y., Han, N., Chen, Y., Lei, G.: Automatic optic disc detection using low-rank representation based semi-supervised extreme learning machine. Int. J. Mach. Learn. Cybern. (2019). https://doi.org/10.1007/s13042-019-00939-0
10. Sang, N., Gan, H., Fan, Y., Wu, W., Yang, Z.: Adaptive safety degree-based safe semi-supervised learning. Int. J. Mach. Learn. Cybern. **10**(5), 1101–1108 (2019)
11. Ashfaq, R.A.R., Wang, X.-Z., Huang, J.Z., Abbas, H., He, Y.-L.: Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci. **378**, 484–497 (2017)
12. Luca, A., Termini, S.: A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. Inf. Control **20**(4), 301–312 (1972)
13. Patwary, M.J.A., Akter, S., Mahmud, T.: An expert system to detect uterine cancer under uncertainty, IOSR J. Comput. Eng. (IOSR-JCE) **16**(5), 36–47 (2014)
14. Patwary, M.J.A., Rahman, M.O., Hossain, M.S.: Uncertainty handling in ship assessment: a case study of Bangladesh. J. Invest. Manag. **4**(5), 152–161 (2015)
15. Sanchez, D., Trillas, E.: Measures of fuzziness under different uses of fuzzy sets. In: International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems, pp. 25–34. Springer (2012)
16. Ralescu, D., Adams, G.: The fuzzy integral. J. Math. Anal. Appl. **75**(2), 562–570 (1980)
17. Wang, Z.-X.: Fuzzy measures and measures of fuzziness. J. Math. Anal. Appl. **104**(2), 589–601 (1984)
18. Farhadinia, B., Xu, Z.: Entropy measures for hesitant fuzzy sets and their extensions. In: Information Measures for Hesitant Fuzzy Sets and Their Extensions. Uncertainty and Operations Research, pp. 69-102. Springer, Singapore (2019)
19. Jozsef, D., Lorant, P.: Measures of fuzziness. Ann. Universitasis Scientiarium Budapestinensis Sect. Computatorica **12**, 69–78 (1991)
20. Li, W., Pedrycz, W., Xue, X., Xu, W., Fan, B.: Fuzziness and incremental information of disjoint regions in double-quantitative decision-theoretic rough set model. Int. J. Mach. Learn. Cybern. (2018). https://doi.org/10.1007/s13042-018-0893-7
21. Kaufmann, A.: Introduction to the Theory of Fuzzy Subsets, vol. 1. Academic Press, Cambridge (1975)

22. Ebanks, B.R.: On measures of fuzziness and their representations. J. Math. Anal. Appl. **94**(1), 24–37 (1983)
23. Scudder, H.: Probability of error of some adaptive pattern-recognition machines. IEEE Trans. Inf. Theory **11**(3), 363–371 (1965)
24. Ma, M., Deng, T., Wang, N., Chen, Y.: Semi-supervised rough fuzzy Laplacian Eigenmaps for dimensionality reduction. Int. J. Mach. Learn. Cybernet. **10**(2), 397–411 (2019)
25. Wei, S., Li, Z., Zhang, C.: Combined constraint-based with metric-based in semi-supervised clustering ensemble. Int. J. Mach. Learn. Cybernet. **9**(7), 1085–1100 (2018)
26. Huang, R., Zhang, G., Chen, J.: Semi-supervised discriminant Isomap with application to visualization, image retrieval and classification. Int. J. Mach. Learn. Cybern. **10**(6), 1269–1278 (2019)
27. Uddin, M.T., Patwary, M.J.A., Ahsan, T., Alam, M.S.: Predicting the popularity of online news from content metadata. In: International Conference on Innovations in Science, Engineering and Technology (ICISET), pp. 1–5 (2016)
28. Patwary, M.J.A., Hossain, S.: Risk analysis of buildings using an expert system: a case study in Bangladesh. In: International Conference on Advances in Electrical Engineering (ICAEE), pp. 141–144 (2015)
29. Yu, Y., Ji, Z., Li, X., Guo, J., Zhang, Z., Ling, H., Wu, F.: Transductive zero-shot learning with a self-training dictionary approach. IEEE Trans. Cybernet. **99**, 1–12 (2018)
30. 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 **275**, 180–191 (2018)
31. Yarowsky, D.: Unsupervised word sense disambiguation rivaling supervised methods. In: Proceedings of the 33rd annual meeting on Association for Computational Linguistics, Association for Computational Linguistics, pp. 189–196 (1995)
32. Riloff, E., Wiebe, J., Wilson, T.: Learning subjective nouns using extraction pattern bootstrapping. In: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003-Volume 4, Association for Computational Linguistics, pp. 25–32 (2003)
33. Abdelgayed, T.S., Morsi, W.G., Sidhu, T.S.: Fault detection and classification based on co-training of semisupervised machine learning. IEEE Trans. Ind. Electron. **65**(2), 1595–1605 (2018)
34. Hu, T., Huang, X., Li, J., Zhang, L.: A novel co-training approach for urban land cover mapping with unclear Landsat time series imagery. Remote Sens. Environ. **217**, 144–157 (2018)
35. Maeireizo, B., Litman, D., Hwa, R.: Co-training for predicting emotions with spoken dialogue data. In: Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions, Association for Computational Linguistics, p. 28 (2004)
36. Nie, F., Cai, G., Li, J., Li, X.: Auto-weighted multi-view learning for image clustering and semi-supervised classification. IEEE Trans. Image Process. **27**(3), 1501–1511 (2018)
37. Jing, P., Su, Y., Nie, L., Bai, X., Liu, J., Wang, M.: Low-rank multi-view embedding learning for micro-video popularity prediction. IEEE Trans. Knowl. Data Eng. **30**(8), 1519–1532 (2018)
38. Zhou, Z.-H., Li, M.: Tri-training: exploiting unlabeled data using three classifiers. IEEE Trans. Knowl. Data Eng. **17**(11), 1529–1541 (2005)
39. Kim, D., Seo, D., Cho, S., Kang, P.: Multi-co-training for document classification using various document representations: TF-IDF, LDA, and Doc2Vec. Inf. Sci. **477**, 15–29 (2019)
40. Li, D., Dick, S.: Residential household non-intrusive load monitoring via graph-based multi-label semi-supervised learning. IEEE Trans. Smart Grid (2018). https://doi.org/10.1109/TSG.2018.2865702
41. Gan, H., Li, Z., Wu, W., Luo, Z., Huang, R.: Safety-aware graph-based semi-supervised learning. Expert Syst. Appl. **107**, 243–254 (2018)
42. Zhou, D., Huang, J., Scholkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 1036–1043. ACM (2005)
43. Blum, A., Chawla, S.: Learning from labeled and unlabeled data using graph mincuts [C]. In: Proceedings of the Eighteenth International Conference on Machine Learning (ICML 2001), Morgan Kaufmann Publishers Inc., Williams College, Williamstown, MA, USA, June 28–July 1 (2001)
44. Wang, X., He, Y.: Learning from uncertainty for big data: future analytical challenges and strategies. IEEE Syst. Man Cybernet. Mag. **2**(2), 26–31 (2016)
45. Patwary, M.J.A., Wang, X.-Z.: Sensitivity analysis on initial classifier accuracy in fuzziness based semi-supervised learning. Inf. Sci. **490**, 93–112 (2019)
46. Wang, X.-Z., Ashfaq, R.A.R., Fu, A.-M.: Fuzziness based sample categorization for classifier performance improvement. J. Intell. Fuzzy Syst. **29**(3), 1185–1196 (2015)
47. Keller, J.M., Gray, M.R., Givens, J.A.: A fuzzy k-nearest neighbor algorithm. IEEE Trans. Syst. Man Cybernet. **4**, 580–585 (1985)
48. Huang, G.-B., Chen, L., Siew, C.K., et al.: Universal approximation using incremental constructive feedforward networks with random hidden nodes. IEEE Trans. Neural Netw. **17**(4), 879–892 (2006)
49. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1–3), 489–501 (2006)
50. Huang, G.-B., Wang, D.H., Lan, Y.: Extreme learning machines: a survey. Int. J. Mach. Learn. Cybernet. **2**(2), 107–122 (2011)
51. Liu, J., Patwary, M.J., Sun, X., Tao, K.: An experimental study on symbolic extreme learning machine. Int. J. Mach. Learn. Cybernet. **10**(4), 787–797 (2019)
52. Wilcoxon, F.: Individual comparisons by ranking methods. Biom. Bull. **1**(6), 80–83 (1945)
53. Demsar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
54. Frank, A.: Uci machine learning repository. University of California, School of Information and Computer Science (2010). http://archive.ics.uci.edu/ml. Accessed 10 Dec 2018

Muhammed J. A. Patwary received the Bachelor's degree and Master's degree in Computer Science and Engineering from University of Chittagong, Bangladesh, in 2006 and 2008, respectively. Currently, he is doing Ph.D. in Parallel information processing under Big Data Institute of College of Computer Science and Software Engineering of Shenzhen University, Guangdong, China. From August 2010 to March 2015 and From March 2015 to August 2017, he was, respectively, a Lecturer and an Assistant Professor in the department of Computer Science and Engineering at International Islamic University Chittagong (IIUC). His major research interests include semi-supervised learning, fuzzy reasoning, neuro-fuzzy systems and applications of machine learning.

Xi-Zhao Wang received the Ph.D. degree in computer science from the Harbin Institute of Technology, Harbin, China, in 1998. He is currently a Professor with the Big Data Institute, Shenzhen University, Shenzhen, China. His current research interests include uncertainty modeling and machine learning for big data. He has edited more than ten special issues and published three monographs, two textbooks, and more than 200 peer-reviewed research papers. By the Google scholar, the total number of citations is over 7000. He is on the list of Elsevier 2015/2016 most cited Chinese authors. He is the Chair of the IEEE SMC Technical Committee on Computational Intelligence, the Editor-in-Chief of Machine Learning and Cybernetics Journal, and Associate Editor for a couple of journals in the related areas. He was a recipient of the IEEE SMCS Outstanding Contribution Award in 2004 and a recipient of the IEEE SMCS Best Associate Editor Award in 2006.

Dasen Yan received the B.D. degree in Electrical Engineering and Automation from Wuyi University, China, in 2017. He is currently pursuing His M.D. degree in Computer Science and Technology, Shenzhen, China. His main fields of research are Data Preprocessing, Fuzzy System Modeling and Model Tree.