Multi-Step Fuzzy Bridged Refinement Domain Adaptation Algorithm and Its Application to Bank Failure Prediction

Vahid Behbood, *Member, IEEE,* Jie Lu, *Senior Memebr, IEEE*, Guangquan Zhang, *Member, IEEE* and Witold Pedrycz, *Fellow, IEEE*

***Abstract*— Machine learning plays an important role in data classification and data-based prediction. In some real world applications, however, the training data (coming from the source domain) and test data (from the target domain) come from different domains or time periods, and this may result in the different distributions of some features. Moreover, the values of the features and/or labels of the data sets might be non-numeric and involve vague values. Traditional learning-based prediction and classification methods cannot handle these two issues. In this study, we propose a multi-step fuzzy bridged refinement domain adaptation algorithm, which offers an effective way to deal with both issues. It utilizes a concept of similarity to modify the labels of the target instances that were initially predicted by a shift-unaware model. It then refines the labels using instances that are most similar to a given target instance. These instances are extracted from mixture domains composed of source and target domains. The proposed algorithm is built on a basis of some data and refines the labels, thus performing completely independently of the shift-unaware prediction model. The algorithm uses a fuzzy set-based approach to deal with the vague values of the features and labels. Four different data sets are used in the experiments to validate the proposed algorithm. The results, which are compared with those generated by the existing domain adaptation methods, demonstrate a significant improvement in prediction accuracy in both the above-mentioned data sets.**

***Index Terms*— Classification, Domain Adaptation, Fuzzy set-based Approach, Fuzzy similarity, Transfer Learning**

I. Introduction

ALTHOUGH machine learning has attracted the attention of researchers in fields such as classification and prediction, most learning models such as neural networks and support vector machines work under a common assumption that the training data and test data are positioned in the same feature space and adhere to the same distribution [1]. When the distribution or feature space of the test data changes, the learning models need to be rebuilt and trained from scratch using new training data. For example, labeled financial data quickly go out of date and may not follow the same distribution over time; thus, previous labeled data cannot be used to reliably predict the financial status of an organization. In many real world applications such as banking systems, collecting new training data and retraining the learning model is very expensive or practically not feasible. It would be useful if the data and knowledge gained in different domains or time periods could be utilized to assist in the formation of the current learning model.

A new framework of machine learning called Transfer Learning has emerged under a variety of names, such as Learning to Learn, Life-long Learning, Meta Learning, and Multi-task Learning [1]. Transfer learning, which is different from traditional machine learning and semi-supervised algorithms [2-5], can handle situations in which the domains of the training data sets and test data sets are different [6]. The study of transfer learning has been inspired by the human ability to utilize previously-acquired knowledge to solve new, similar (but not identical) problems more quickly and efficiently than if this form of knowledge were not available.

Nevertheless, current transfer learning methods still have a number of drawbacks: 1) there is a direct reliance on statistical models in current transfer learning methods with probabilistic assumptions (e.g., about specific probability distribution functions) that may be difficult to satisfy. Consequently, it could be difficult to achieve highly accurate prediction in some real-world applications; 2) Existing transfer learning methods only consider features and labels whose values are numeric or assume a single value from a discrete set of attribute values, and this assumption could be viewed as a serious impediment in the presence of uncertainty; 3) one development in existing transfer learning methods attempts to solve the domain adaptation problem by adjusting the decision boundaries and models using global learning, but this makes the methods highly dependent on the shift-unaware prediction or classification models; 4) The existing application of transfer learning methods is limited to the field of text classification and reinforcement learning, and other potentially promising application areas have not so far been investigated [2].

Manuscript received November 2012; revised May 2014; accepted September 2014. This work was supported by Australian Research Council (ARC) under Discovery Projects DP 140101366 and DP110103733.

V. Behbood, J. Lu and G. Zhang are with School of Software, Center for Quantum Computation and Intelligent System (QCIS), Faculty of Engineering and Information Technology, University of Technology Sydney (UTS), Sydney, NSW, 2007, Australia. (vahid.behbood@uts.edu.au; Jie.lu@uts.edu.au; Guangquan.Zhang@uts.edu.au).

W.Pedrycz is with Department of Electrical and Computer Engineering, Faculty of Engineering, University of Alberta, Edmonton, AB, Canada and Department of Electrical and Computer Engineering, Faculty of Engineering, King Abdulaziz University, Jeddah, 21589, Saudi Arabia and Systems Research institute, Polish Academy of Sciences, Warsaw, Poland. (wpedrycz@ualberta.ca).

To address and overcome these limitations, we propose a Multi-step Fuzzy Bridge Refinement Domain Adaptation (MFBRDA) algorithm in this study and investigate its applicability to the problem of bank failure prediction. The MFBRDA algorithm can handle situations of data uncertainty in which the features assume vague values, and the outputs must provide flexible and explanatory results to solve the problem. The key facet of originality of the study lies in the fact that the domain adaptation problem is solved by refining the fuzzy initial labels present in the target domain by similarity-based local learning. The efficiency of the fuzzy set-based approach and local learning (using fuzzy similarity) for the problem of domain adaptation has been quantified as well.

The main idea behind the proposed MFBRDA algorithm is to explore the most similar instances in a set of mixture domains of the training and test data and treat them as a "bridge" for transferring the feature distribution from the source domain to the target domain. The label values of these instances are utilized to refine the labels of the initial target instances which are reported by a given prediction model, referred to as a shift-unaware model. Using label refinement instead of model adjustment makes the MFBRDA algorithm completely independent of the shift-unaware model. US bank failure data set and Newsgroup data set are used for benchmarking the MFBRDA algorithm against three traditional machine learning models such as Support Vector Machine [7], Multi-Layer Perceptron Neural Network [8], and Fuzzy Neural Network [9], along with the existing domain adaptation methods [7, 10, 11]. The results demonstrate the superior performance of the proposed algorithm and show the significant role of a fuzzy set-based approach and local learning in accuracy enhancement. The empirical results show the better performance of the proposed algorithm. We also quantify the significant impact of the fuzzy set-based approach and local learning on accuracy enhancement.

Real-world bank failure data are used here as a case study of the proposed MFBRDA algorithm to realize a long-term bank failure prediction. As the data distribution changes over a longer period and there is an uncertainty factor, the existing machine learning methods cannot handle these two issues and form a prediction model that has an acceptable level of prediction accuracy. We show that the proposed MFBRDA algorithm significantly improves the quality of the results and offers a new idea of transfer learning to solve this challenging real-world problem.

To the best of our knowledge, the proposed MFBRDA algorithm is the first to use the technology of fuzzy sets and local learning for domain adaptation, and the first study to apply a transfer learning method for financial failure prediction.

The paper is organized as follows. In Section 2, preliminary concepts including the definition of domain adaptation and related works are introduced. The *MFBRDA algorithm* and its detailed implementation are presented in Section 3. Section 4 describes the experimental illustration and results for the MFBRDA algorithm. A bank failure case study is presented in Section 5. Section 6 concludes the paper and outlines future research.

II. Preliminaries and Related Studies

The definition of transfer learning, particularly domain adaptation, is introduced in this section, following which the categories of transfer learning and related works are described. Definition

A. Definition

Definition 1 (Domain) [1]: A domain is denoted by $D = \{F, P(X)\}$ where F is a feature space and $P(X)$, $X = \{x_1, \dots, x_n\}$ is the marginal probability distribution of instances.

Example: If the learning task is to realize bank failure prediction**,** F is a set of financial features such as *Average Capital Ratio* and *Average Loan Loss*, X is the set of all instances (banks) and $P(X)$ is the marginal distribution of these instances on the above features; for example, the distribution of instances on *Average Capital* Ratio is described as normal distribution with mean value μ and variance σ^2 : $P_{Average\ Capital\ Ratio}(X) \sim N(\mu, \sigma)$. In general, if two domains are different, they may have different feature spaces or/and different marginal probability distributions.

Definition 2 (Task) [1]: A task is denoted by $T = \{Y, f(\cdot)\}$ where $Y = \{y_1, \dots, y_m\}$ is a label space and $f(\cdot)$ is an objective predictive function which is not observed and has to be learned by pairs (x_i, y_i). The function $f(\cdot)$ can be used to predict the corresponding label, $f(x_i)$, of a new instance x_i. From a probabilistic viewpoint, $f(x_i)$ can be expressed as $P(y_i|x_i)$ where P is the probability of label y_i for given instance x_i.

Example: In the bank failure prediction example, which is a binary (two-class) prediction task, y_i denotes failed or survived bank. More specifically, the source domain can be denoted as $D_s = \{(x_{s_1}, y_{s_1}), \dots, (x_{s_n}, y_{s_n})\}$ where $x_{s_i} \in X_s$ is the source instance or bank in the bank failure prediction example and $y_{s_i} \in Y_s$ is the corresponding class label, (failed, survived) for bank failure prediction. Similarly, the target domain is denoted as $D_t = \{(x_{t_1}, y_{t_1}), \dots, (x_{t_n}, y_{t_n})\}$ where $x_t \in \chi_t$ is the target instance and $y_{t_i} \in Y_t$ is the corresponding class label. In the most scenarios we have $t_n \ll s_n$.

Definition 3 (Transfer Learning) [1]: Given a source domain D_s and learning task T_s a target domain D_t, and learning task T_t,, transfer learning aims to improve the learning of the target predictive function $f_t(\cdot)$ in D_t using the knowledge in D_s and T_s, where $D_s \neq D_t$ or $T_s \neq T_t$. In addition, there are explicit or implicit relationships among the feature spaces of two domains, such that we imply that the source domain and the target domain are related. It should be mentioned that when the target and source domains are the same ($D_s = D_t$) and their learning tasks are also the same ($T_s = T_t$), the learning problem becomes a traditional machine learning problem.

Example: In the above definition, the condition $T_s \neq T_t$ implies that either $Y_s \neq Y_t$ or $f_s(\cdot) \neq f_t(\cdot)$ or both (*Multi- task learning*). For instance, it corresponds to a situation in which

the source banking system comes with binary class labels (failed and survived), whereas the target banking system has more than two class labels, or the source prediction model is so different from the target prediction model that may result in predicting two different class labels for the same instance.

Example: Similarly, the condition $D_s \neq D_t$ implies that either $F_s \neq F_t$ or $P_s(X) \neq P_t(X)$ (*Transductive transfer learning*). For example, in the bank failure prediction example, this means that between a source banking system and a target banking system, either the financial features of the two domains are different between, or the marginal distributions of banks are different.

Definition 4 (Domain Adaptation) [1]: Domain Adaptation is a category of transductive transfer learning in which $F_t = F_s$ but $P_t(X) \neq P_s(X)$.

Example: As an illustration for domain adaptation in long-term bank failure prediction, we would like to predict the bank's status on date t_2 (D_t) using the bank's data on date $t_1(D_s)$. It is assumed that the same features are applied for prediction in both domains, but the distribution of those data might be different. For example, the marginal distribution of banks on *Capital Ratio* (a financial feature for bank failure prediction) in date t_1 is: $P_s(X) \sim N(\mu_s, \sigma_s)$, but, on date t_2 is: $P_t(X) \sim N(\mu_t, \sigma_t)$ and $\mu_t \neq \mu_s$ and/or $\sigma_t \neq \sigma_s$. This means that the mean and/or variance values of *Average Capital Ratio* of banks have changed through time t_1 to t_2.

B. Related Studies

The studies that aim to solve the domain adaptation problem can be categorized into two groups [1]: (i) Transferring the knowledge of instances: this approach is motivated by the importance of samples. Here an attempt is made to find an optimum weight for each instance to learn a more accurate model for the target domain. Several domain adaptation methods, such as covariate shift [12] or sampling selection bias [10], were proposed to cope with the inconsistency of data distributions. These methods re-weighted the training samples coming from the auxiliary domain by using unlabeled data from the target domain such that the statistics of samples from both domains are matched. We select the sample selection bias [10] as a popular domain adaptation method in this group for benchmarking. The reader may refer to the recently published book by Quionero-Candela et al. [13]; (ii) Transferring the knowledge of feature representation: this approach focuses on the feature space and attempts to extract and/or convert relevant features which reduce the difference between the domains. Blitzer et al. [14-16] proposed an SCL algorithm to define pivot features on the target domain from both domains and then used unlabeled instances from the target to create the classification model. Dai et al. [17] proposed a co-clustering based algorithm to propagate the label information across domains. Xue et al. [18] presented a cross-domain text classification known as TPLSA to integrate target instances and source instances into a unified probabilistic model. Very recently, Bruzzone and Marconcini [11] proposed Domain Adaptation Support Vector Machine (DASVM), which extended Transductive SVM (TSVM) to label unlabeled target patterns progressively and simultaneously remove some auxiliary labeled patterns. We use this advanced domain adaptation method for benchmarking.

In the last few years, some researches aimed to use fuzzy sets and systems into the transferring process of knowledge among domains. Deng et al [19] proposed a fuzzy system modelling approach with knowledge-leverage capability from source domain. The Mamdani–Larsen-type fuzzy system (ML-FS) is chosen to incorporate a knowledge-leverage mechanism. It makes use of some objective function (criterion) to integrate the model knowledge of the source domain and the data of the target domain and, thus, learn the induced fuzzy rules of the model accordingly. The proposed method is distinctive in preserving the data privacy as only the knowledge (such as the corresponding density distribution) rather than the data of the source domain is adopted. Synthetic and real-world data sets have been used in the empirical study. The experimental results demonstrate the outperformance of the proposed method when compared with several existing related methods. To tackle transfer learning tasks, Seera and Lim [20] proposed a Fuzzy Min–Max (FMM) neural network that is equipped with an incremental or online learning capability. Based on the authors' previous work in deploying the FMM neural network with off-line learning [21], an online learning strategy for the FMM network is further devised to tackle transfer learning problems. Behbood et al [22] developed a fuzzy-based transductive transfer learning for long term bank failure prediction in which the distribution of data in source domain different from that in target domain. They applied three classical predictors including Neural Network, Support vector machine and Fuzzy Neural Network to predict the initial labels for samples in target domain then try to refine the labels using fuzzy similarity measures. Afterwards, the authors improved the performance of the fuzzy refinement domain adaptation method [23] by developing a novel fuzzy measure to involve both the similarity and dissimilarity in the refinement process. The proposed method has been applied to text categorization and bank failure prediction. The experimental and results demonstrated the outperformance of proposed method comparing with popular classical transductive transfer learning methods. Using fuzzy techniques in similarity measurement and label production, the authors revealed the advantage of fuzzy logic in knowledge transfer where target domain lacks of critical information and involves uncertainty and vagueness. Shell and Coupland [24] proposed a framework of fuzzy transfer learning to form and prediction model in intelligent environments. To address the issues of modelling environments in the presence of uncertainty and noise, they introduced a fuzzy logic based transfer learning enabling the absorption of inherent uncertainty and dynamic nature of transfer knowledge in intelligent environments. They created a transferable fuzzy inference system using labelled data in the source domain, then, adapted and applied the resulted rule base to predict the labels for samples in the target domain. The source rules were adjusted and adapted to the

target domain using Euclidean distance measure. The proposed method is examined in two simulated intelligent environment. The experimental results demonstrated superior performance of the fuzzy transfer learning comparing with classical prediction models; however the method has not been compared with any transfer learning method. Deng et al. [25] proposed the generalized hidden-mapping ridge regression (GHRR) method in order to train various types of Computational Intelligence models, including neural networks, fuzzy logical systems and kernel methods. Furthermore, the knowledge-leverage based transfer learning mechanism is integrated with GHRR to realize the inductive transfer learning method called transfer GHRR (TGHRR). The proposed GHRR and TGHRR algorithms have been evaluated experimentally by carrying out regression and classification on synthetic and real world datasets. The results demonstrated that the performance of TGHRR is competitive with or even superior to existing state-of-the-art inductive transfer learning algorithms.

III. Main Concepts and Techniques of The Proposed MFBRDA Algorithm

To describe the MFBRDA algorithm, we first look at the three main intuitively appealing concepts: similarity-based refinement, multi-step refinement, and fuzzy-based refinement.

A. Similarity-based Refinement

In contrast to "standard" machine learning methods, it is assumed in the definition of domain adaptation that joint distribution in the source domain (training data) is different from the target domain (test data), that is: $\mathrm{p_s}(\mathrm{X},\mathrm{Y}) \neq \mathrm{p_t}(\mathrm{X},\mathrm{Y})$. In some cases the joint distributions only differ in the marginal distributions of covariate, namely $\mathrm{p_s}(\mathrm{X}) \neq \mathrm{p_t}(\mathrm{X})$, while $\mathrm{p_s}(\mathrm{Y}|\mathrm{X}) = \mathrm{p_t}(\mathrm{Y}|\mathrm{X})$ which is termed **covariate shift** or **sample selection bias** [13]. There is a number of research studies have been published about this category of domain adaptation. One of these methods is the Bridge Refinement algorithm which is motivated by the PageRank algorithm [26]. In the algorithm, it is assumed that the conditional probability of a specified label C given an instance d, does not vary among different distributions including source, target and union of source and target distributions: $P_s(Y = C|X = d) \cong P_{sUt}(Y = C|X = d) \cong P_t(Y = C|X = d)$ although the marginal probability of instance $d, P(d)$, varies. This is based on the fact that if identical instances appear in the target domain, the source domain or the mixture domain of both domains, their predicted label should be the same. Furthermore, the greater the number of similar instances in the target domain, the greater is the probability that they will have the same label. This situation implies a mutual reinforcement relationship between instances in the target domain and the source domain and which can be used to correct the predicted labels. For instance, in a two-class problem, significantly similar instances are located in the same class. In other words, if the similarity measure considers the cluster structure (distributional) difference between two domains in the computation then the similarity between two instances from two different domains provides some evidence in support of the same class. The similarity function plays an important role and needs to be defined well enough to map the instances discriminate them accurately. Fuzzy similarity should be defined such that there will be a high probability that similar instances will have the same label.

The proposed bridge refinement approach comprises two sequential phases: preprocessing and refinement. In the first phase, the labels of the instances in the target domain are initially determined by a given prediction model (shift-unaware prediction model) which has been trained by labeled instances from the source domain. In phase two, the initial label of each target instance is refined using the labels of the k-nearest instances to the target instance. These similar instances are selected from mixture domains which are composed of portions of instances in both the source domain and the target domain. We then apply the label values of the most similar instances for refinement purposes. In this study, the fundamental assumption is that there are no labeled data in the target domain; the MFBRDA algorithm is developed based on this assumption. However, in rare real-world cases, a small number of labeled data might be present in the target domain. Having a few labeled data in the target domain is helpful for the transfer learning process and help increase the prediction accuracy of the MFBRDA algorithm. For example, in the experimental section, only a single setting (viz. Setting 9) considers 5% of data in target domain as labeled samples, while the other eight settings assume no labeled data in the target domain. The empirical results show an improvement in accuracy when there are a few labeled data in the target domain. As an illustration of a binary prediction problem, Fig. 1 visualizes the instances present in the source domain for classes 1 and 2. Fig. 2 indicates that there might be a few labeled samples in the target domain and presents the instances in the target domain, including those that belong to classes 1 and 2 and those that do not have labels. We would like to determine a label for one of the unlabeled instances located in the target domain, indicated by a red triangle. We select the most similar labeled instances from both domains to find the label of a given target instance. Fig. 3 shows the nine most similar instances in both domains – all of them belong to class 1. The conclusion is that the given instance most likely belongs to class 1. In this case, these labeled samples are also used to determine the labels of samples in the target domain. The target domain labeled samples are used in the same way as the labeled source domain samples are used in the Refinement Phase of the MFBRDA algorithm.

Although the source and target domains have the same dimension, the cluster structure (membership functions for each feature) in the source domain is different from that in the target domain. Hence, simply using k-nearest neighbor to find the most similar samples in the source domain and specify the label for a sample in the target domain may not lead to a correct result. However, contrary to the existing domain adaptation methods, a local learning-based method similar to k-nearest neighbor can be used in a set of mixture domains of

both source and target domains to specify the labels in the target domain if the similarity measure, used in the local learning-based method, takes the cluster structure difference between the source and target domains into account.

B. Multi-step based Refinement

The main objective of domain adaptation is to build the mixture domain (D_w) by combining data from the target domain (D_t) and the source domain (D_s), from which the most similar instances are selected. The refinement process can be performed in a single iteration where the mixture domain composed of only target domain instances (D_t) or only source domain instances (D_s), or a combination of source and target domain instances ($D_w \subset D_s \cup D_t$). Likewise, the refinement can also be executed in two iterations in which the second mixture domain is composed of target domain instances (D_t) or a combination of both domains ($D_w \subset D_s \cup D_t$).

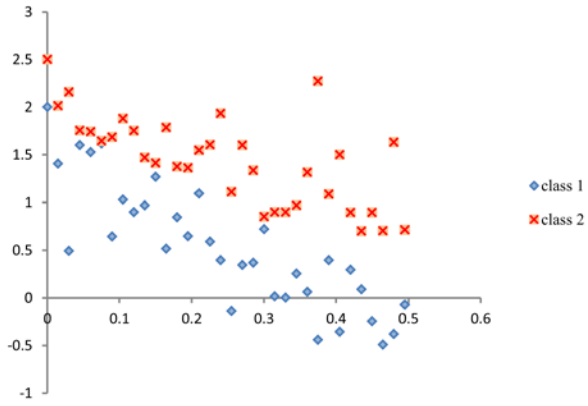


Fig. 1. Labeled data present in the source domain

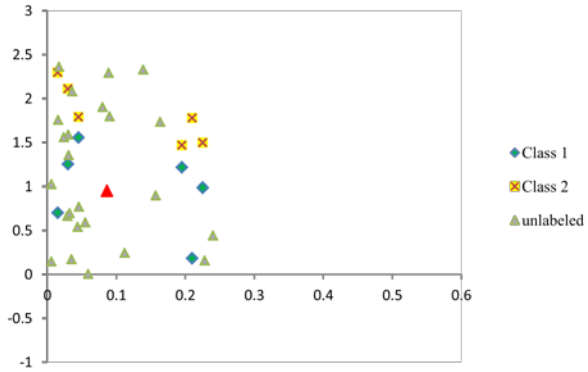


Fig. 2. Labeled and unlabeled data present in the target domain

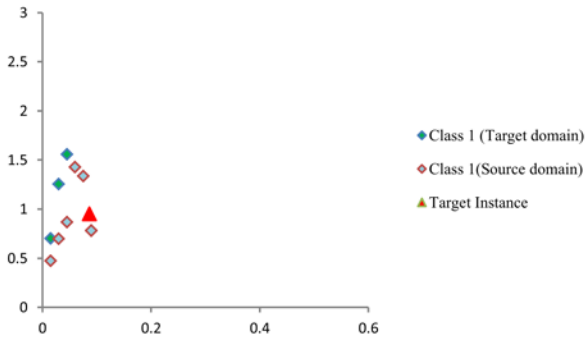


Fig. 3. Nine most similar instances coming from a mixture domain

In the two-step setting, the labels are first refined using similar instances coming from the first mixture domain and are then refined in the second iteration using similar instances coming from the second mixture domain. The three-step setting of refinement can also be defined in the same manner. In this scenario, the first, second and third mixture domains will be composed of the source domain (D_s), the group of both domains ($D_s \cup D_t$) and the target domain (D_t) instances respectively. All the possible settings in different iterations are reported in Table I, which shows the above-mentioned settings of refinement when different mixture domains are used in each iteration.

The results of the two-step and three-step refinement settings may be significantly improved compared to the initial labels computed by the shift-unaware prediction model. However, the accuracy of each data set follows the performance of the shift-unaware model and consequently demonstrates poor performance in some cases. To solve these problems and improve predictive accuracy, we propose to use multiple steps to refine the initial labels. The refinement process moves from the source domain (D_s) toward the target domain (D_t) through $\boldsymbol{n}$ steps using the trade-off parameter μ_w, which specifies the percentage of instances of the source domain and target domain in the mixture domain in each step of refinement. As $\boldsymbol{n}$ increases, μ_w becomes greater and the contribution of source domain data in the mixture domain becomes lower; conversely, the portion of target domain data increases. At each step, the samples coming from the source and target domains are selected by random sampling mechanism to create the mixture domain, but the samples from target domain at each step are reserved for the next step. Hence, the mixture domain at each step always includes the target instances of the previous step. Accordingly, this method of sampling allows the opportunity for all samples to participate in the refinement process; additionally, the consequent neighboring mixture domains are similar to each other and smoothly transfer the label structure from the source domain to the target domain. Through the multi-step process, a set of adjacent similar mixture domains create a bridge to transfer the label structure between the source and target domains more accurately. As previously mentioned, the proposed refinement algorithm is motivated by the PageRank modification algorithm [26]. The PageRank algorithm conveys a mutual reinforcement principle stating that good pages may also link to other good pages. According to this PageRank algorithm we have, 1) multi-step process is required, and 2) modifications converge to a unique point. The two issues have been shown in our experiments: the algorithm converged in a few iterations (5 to 10) in each step (mixture domain). Fig. 4 demonstrates the multiple steps approach, used in the proposed algorithm.

TABLE I
POSSIBLE SETTINGS OF REFINEMENT

Category	Iteration 1	Step 2	Step 3
On-step Setting	$D_m \subset D_s$	----------	----------
	$D_m \subset D_t$	----------	----------
	$D_m \subset D_s \cup D_t$	----------	----------
Two-step Setting	$D_m \subset D_s$	$D_m \subset D_t$	----------
	$D_m \subset D_s$	$D_m \subset D_s \cup D_t$	----------
	$D_m \subset D_s \cup D_t$	$D_m \subset D_t$	----------
Three-step Setting	$D_m \subset D_s$	$D_m \subset D_s \cup D_t$	$D_m \subset D_t$

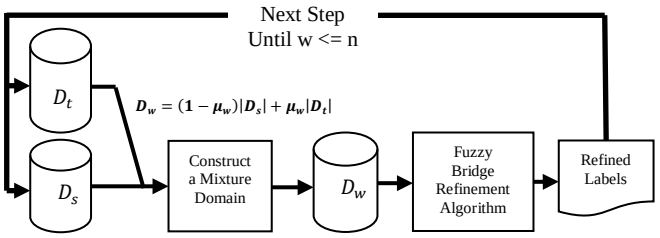


Fig. 4. The multiple steps in Multi-step Fuzzy Bridge Refinement Domain Adaptation

C. *Fuzzy-based Refinement*

The fuzzy set-based approach is utilized in the proposed MFBRDA algorithm to tackle the imprecise nature of financial features and related vague values. In particular, the class labels are defined by fuzzy sets and each instance comes with a certain membership value in each class. These fuzzy labels allow for some partial membership in each class, enabling the algorithm to classify a target instance more accurately than the Boolean-like (0,1) labels. As a simple illustration, Fig. 5a and 5b show a target instance (indicated in red) and the nine most similar instances with 0-1-value labels and fuzzy labels, respectively present in a binary classification problem. In Fig. 5a, the target instance will be classified as belonging to class 1 because five out of nine (majority) most similar instances belong to class 1. The target instance in Fig. 5b cannot be classified as simply as it is done in Fig. 5a. In Fig. 5b, it is assumed that the instances have fuzzy labels (membership values in each class) and the class of the target instance should be specified using the average of membership values of the nine most similar instances in each class. According to the following calculation, the target instance will be classified as class 2 because the average value demonstrates that the membership value of the target instance in class 2 is more than that in class 1, despite the fact that the number of most similar instances in class 1 is larger than in class 2.

Average membership value in class 1: $\frac{0.2+0.3+0.25+\cdots+0.6}{9} = 0.44$

Average membership value in class 2: $\frac{0.8+0.7+0.75+\cdots+0.4}{9} = 0.56$

More importantly, by engaging fuzzy sets, the output of the proposed MFBRDA algorithm forms the membership value of the target organization in each class. These labels clearly reflect the level of financial health of the organization instead of a binary (yes-no) description of the organization. This

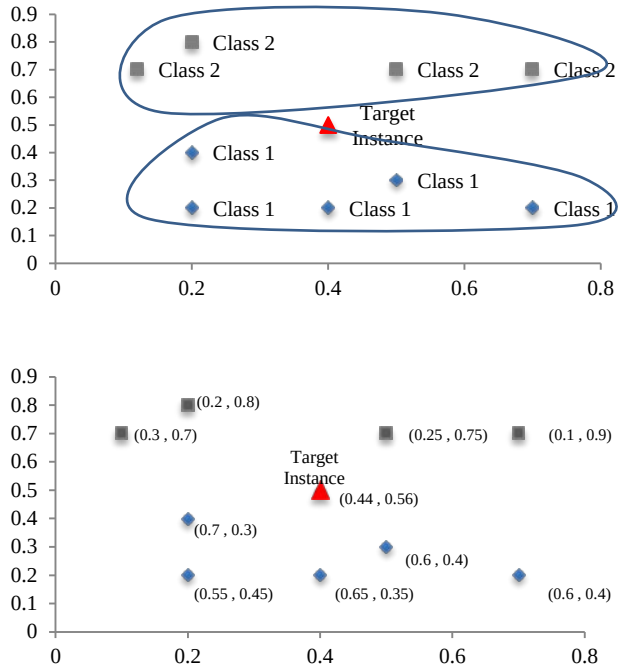


Fig. 5. (a) Nine most similar instances with {0,1}-value labels. (b) Nine most similar instances with fuzzy labels (membership value in each class)

offers flexibility in dealing with uncertainty of data in information systems such as bank failure warning systems.

Our study provides a cluster-based representation of the decision region by fuzzy clusters rather than distribution-based representation of data. Since the source and target domains may not have the same cluster structure, we use a fuzzy similarity measure to find the most similar instances to the target instance to improve predictive accuracy in the refinement process. Using fuzzy similarity in the cluster-based structure of a domain, it is possible to incorporate the domain information in computing the similarity, consequently achieve more accurate results in similarity measurement. In fact, if the fuzzy sets (denoted by a set of linguistic terms) of fuzzy features are computed using the numerical data of instances of a domain, each fuzzy set represents a fuzzy cluster in a decision region to which a considerable number of instances belong. Hence, any pair of instances may belong to two different fuzzy clusters and the distance between these fuzzy clusters should be incorporated when measuring the similarity between two instances. Using a distance-based fuzzy similarity measurement, the distance of two instances is calculated simultaneously based on the distance between two instances and the distance between the fuzzy clusters to which these instances belong. Hence a distance-based fuzzy similarity measurement not only takes the distance between two instances into account but also reflects the distance between the two fuzzy clusters to which the corresponding instances belong. To clarify the above-mentioned arguments, the following example is presented.

Example 1: Given the three instances $A = (0.5,0.3,0.7)$, $B = (0.8,0.6,0.9)$ and $C = (0.2,0.1,0.4)$ in a 3-dimensional feature space, we first calculate the similarity between instances using Euclidean distance-based similarity:

$s(A,B) = e^{-d(A,B)}$

where $d(A,B) = \sqrt[2]{\frac{(x1_A - x1_B)^2 + (x2_A - x2_B)^2 + (x3_A - x3_B)^2}{3}}$

We define trapezoidal fuzzy sets (membership functions) for each variable shown in Fig. 6a-6c and then compute the similarity between instances using fuzzy Euclidean distance-based similarity:

$fs(A,B) = e^{-\sqrt{d(A,B) \times fd(A,B)}}$

Where $fd(A,B) = \frac{d\mu_L + d\mu_M + d\mu_H}{3}$ and

$d\mu_L = \sqrt[2]{\frac{(\mu_L(x1_A) - \mu_L(x1_B))^2 + (\mu_L(x2_A) - \mu_L(x2_B))^2 + (\mu_L(x3_A) - \mu_L(x3_B))^2}{3}}$.

Although these instances, particularly the pair A and B and the pair A and C, are close to each other (similar), it can be seen that the instances A, B and C belong to three completely different fuzzy sets (clusters) as Medium, High and Low respectively in all three variables in the decision region. It would be desirable for the similarity measure to take this fact into consideration to produce a more accurate result. It is expected that the more accurate similarity value of these instances would not be as high as that calculated by the Euclidean distance-based similarity ($(A,)$). Because $(A,)$ only considers the distance between the instances ($(A,)$) and not the distance between the fuzzy clusters, it produces a high value for similarity. While the fuzzy Euclidean distance-based similarity ($fs(A,B)$) simultaneously computes the distance between instances ($d(A,B)$) and the distance between clusters ($fd(A,B)$) (incorporated in the similarity formula), which results in a lower value of similarity. Because these instances are located in three completely different fuzzy clusters in the decision region, the lower similarity is desirable if we want to employ the domain information in the similarity measurement. We expect that the more accurate similarity value among these instances will not be as high as the Euclidean distance-based similarity, because the similarity measure explained in the example takes the cluster structure of the problem region mapping on each feature (fuzzy membership function) into account when computing similarity, while the Euclidian distance does not. For instance, if samples A and B are located in the centre of two different clusters, then the appropriate similarity value should signify this difference and will be smaller than when two samples are located in the centre of the same cluster. The Euclidean distance does not incorporate this issue into the calculation, unlike our simple example, which does. The similarity values between instances using both similarity measurements are shown in Table II. As can be seen, the fuzzy measure provides lower similarity values between these instances than the non-fuzzy measure, thus giving a more accurate similarity measure. For the sake of conciseness we describe a simple example to show that any distance measure that takes the cluster structure difference into account will be more appropriate in a situation where the cluster structure is different between domains (Domain Adaptation) from those data comes. In the Experiment section (Section 5), the similarity between data samples is calculated using the similarity measure developed in [27]. This measure calculates similarity based on the similarity between data samples (numeric or fuzzy sets) and the clusters in which the samples are located.

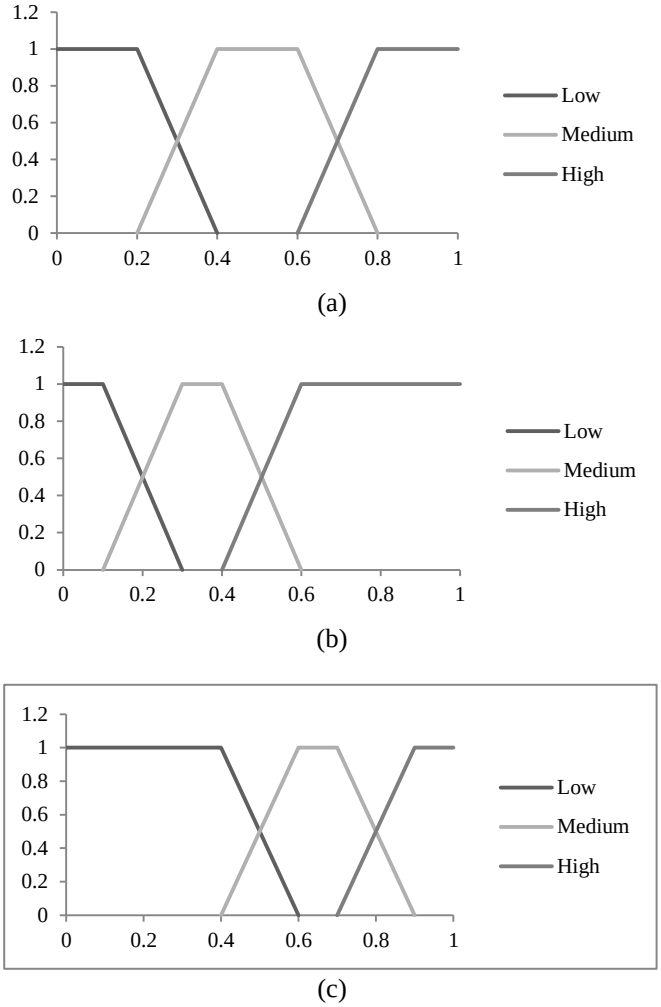


Fig. 6. (a) Fuzzy sets formed over x_1. (b) Fuzzy feature sets formed over x_2. (c) Fuzzy sets formed over x_3

TABLE II
SIMILARITY VALUES BETWEEN INSTANCES USING SIMILARITY AND FUZZY SIMILARITY

	(A,B)	(A,C)	(B,C)
s	0.7628	0.7628	0.5854
fs	0.6538	0.6538	0.5502

The proposed algorithm is capable of handling the vague values of input features by using fuzzy clusters to represent the decision region and capturing the covariate values in the form of fuzzy membership functions (fuzzy sets). It distributes the decision region into fuzzy clusters that have been created by mapping the defined fuzzy sets in each input covariate on the decision region. The antecedents of the input fuzzy values (say ABOUT 10) are matched against the fuzzy sets in each covariate and their position is calculated (in the form of a

fuzzy set) in the fuzzy clusters in the decision region. The similarities between the computed fuzzy sets in the partitioned region are then calculated to find the k-most similar instances for a given instance in the region. The proposed algorithm handles vague values by matching input fuzzy values (in the form of fuzzy membership function) with fuzzy sets in all covariates, consequently regulating the input sample position in the fuzzy cluster-based decision region. The example below is provided to compare two circumstances of numerical inputs and fuzzy inputs for clarification.

Example 2: As shown in Fig. 7, we assume that there is a 2-D decision region which is partitioned into nine fuzzy clusters using three fuzzy sets in each dimension. Given samples *A* and *B* are a fuzzy numbers in a 2-D decision region defined as A (about 4, about 8) and B (about 13, about 5) where fuzzy values are defined as a triangular-shaped membership function in each dimension. As shown in Fig. 7, the fuzzy number A is matched with Low and Medium fuzzy sets on dimension X1 and with High fuzzy set on dimension X2; the result is mapped in the overlap region of cluster 3 and cluster 6. Also, the fuzzy number B is matched with fuzzy set High on dimension X1 and with Medium and High fuzzy sets on dimension X2. Finally, the similarity between the mapped fuzzy numbers A and B in the decision region are calculated using the method developed by [27].

IV. Description of The MFBRDA Algorithm

Given D_s is the source domain with fuzzy feature sets $F^s = \{f_1^s, \dots, f_m^s\}$ and D_t is the target domain with fuzzy features $F^t = \{f_1^t, \dots, f_m^t\}$, where f_i is a set of fuzzy trapezoidal-shaped membership functions for each feature. Discrete Incremental Clustering (DIC) [28] is applied to build the fuzzy sets of features. The steps of the MFBRDA algorithm are organized into two phases, as described below:

Phase 1 is a pre-processing phase completed to represent (encode) numeric input in terms of the fuzzy sets (reference fuzzy sets) defined in the given input variable, compute the initial label values using a shift-unaware prediction or the given classification model $g(\cdot)$ and calculate the similarity matrix.

Phase 2 is the refinement phase, in which we apply the proposed refinement equation denoted by (1) in Step 2-3.

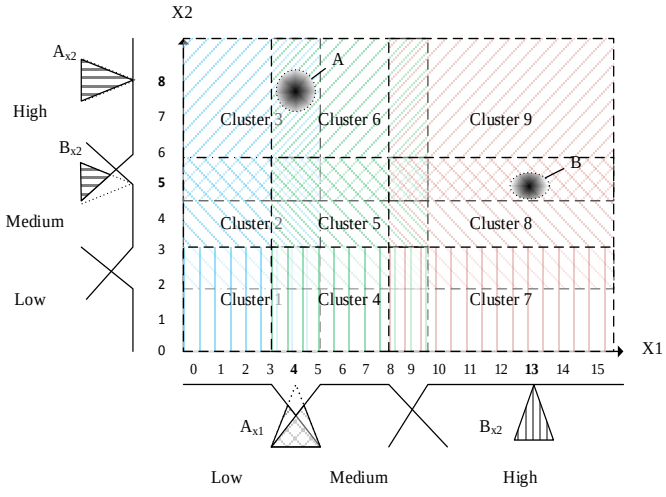


Fig. 7. Fuzzy cluster structure-based 2-Dimensional decision region with two input fuzzy numbers A and B

Algorithm: Multi-Step Fuzzy Bridged Refinement Domain Adaptation (*MFBRDA*)

Phase 1: Preprocessing

Input:

Source domain: D_s

Target domain: D_t

Fuzzy label space: Y

Shift-unaware model: $g(\cdot)$

Fuzzy similarity function: $S(\cdot)$

Output: Unrefined label matrix for instances in target domain: G

Step 1-1: Use singleton fuzzifier to encode numeric input of instances from both domains. Singleton fuzzifier simplifies the computation involved in the fuzzy inference engine and the antecedent matching for any type of membership functions.

$$\mu_{\tilde{x}_i}(\tilde{x}_i) = \begin{cases} 1, & if\ \tilde{x}_i = x_i \\ 0, & Otherwise \end{cases}$$

Step 1-2: Perform antecedent matching of fuzzified (encoded) inputs $x_i \in D_s$ and D_t against fuzzy features F^s and F^t respectively. The input membership value of each fuzzy set is computed as follows:

$$\mu_{\widetilde{f_k}}(\tilde{x}_i) = \begin{cases} 0, & if \quad x_i \le l_{\widetilde{f_k}} \\ \frac{x_i - l_{\widetilde{f_k}}}{u_{\widetilde{f_k}} - l_{\widetilde{f_k}}}, & if \quad l_{\widetilde{f_k}} \le x_i \le u_{\widetilde{f_k}} \\ 1, & if \quad u_{\widetilde{f_k}} \le x_i \le v_{\widetilde{f_k}} \\ \frac{r_{\widetilde{f_k}} - x_i}{r_{\widetilde{f_k}} - u_{\widetilde{f_k}}} & if \quad v_{\widetilde{f_k}} \le x_i \le r_{\widetilde{f_k}} \\ 0, & if \quad x_i \ge r_{\widetilde{f_k}} \end{cases}$$

Step 1-3: Compute the similarity matrix using the fuzzy similarity function $S(\cdot)$:

For $i = 1$ to $|D_s| + |D_t|$

  For $j = 1$ to $|D_s| + |D_t|$

    $m_{ij} = S(x_i, x_j)$

  Next j

Next i

Step 1-4: Train shift-unaware learning-based model $g(\cdot)|D_s$ by labeled data of source domain.

Step 1-5: Calculate the unrefined label matrix for target domain instances (G) using $g(\cdot)|D_s$ as follows:

For $i = 1$ to $|D_t|$

  For $j = 1$ to L

  $g_{ij} = [g(x_i)|D_s]_j = \left(\mu_{Y_j}(x_i)\right)$

  Next j

Next i

Phase 2: Refinement

Input:

Source domain: D_s

Target domain: D_t

Fuzzy label space: Y

Fuzzy similarity function: $S(\cdot)$

Unrefined Label Matrix: G

Impact tradeoff parameter: α

Number of most similar instances: K

Number of steps of refinement: $\boldsymbol{n}$

Output: Refined Label matrix for instances in target domain MR^n.

For $w = 1$ to n

Step 2-1: Create the mixture domain D_w combination of source and target domain as follows:

$D_w = (1 - {}^{w}/_{n})|D_s| + {}^{w}/_{n}\,|D_t|$

Step 2-2: Find K most similar instances ($N_i^{D_w}$) for each target instance. These instances are extracted from mixture domain D_w.

For $i = 1$ to $|D_t|$

$N_i^{D_w} = \{n_i = argmax_j m_{ij},\ n_i \in D_w\}$

Next i

Step 2-3: Refine the fuzzy label for each target instance.

Repeat t

For $i = 1$ to $|D_t|$

For $j = 1$ to L

$$mr_{ij}^{(w)}(t) = \alpha\left(\frac{\sum_{x_0 \in N_i^{D_w}} S(x_i, x_0)\left(mr_{ij}^{(w-1)}(t-1) - mr_{x_0 j}^{(w-1)}(t-1)\right)}{|N_i^{D_w}|}\right) + (1-\alpha) g_{ij} \quad \textbf{(1)}$$

Next j

Next i

Until MR^w converges

Next w

As mentioned previously, D_s is with fuzzy feature sets $F^s = \{f_1^s, \dots, f_m^s\}$ and D_t is with fuzzy features $F^t = \{f_1^t, \dots, f_m^t\}$, where f_i is a fuzzy trapezoidal membership function for each feature. It is assumed that the number of these features is the same for both D_s and D_t the source and target domains but the membership function of these fuzzy sets may be different. The difference between the source and target domains is the result of the difference between the data distribution for each feature in the source domain and the same feature in the target domain. This difference may cause the feature to exhibit a different number of concepts (fuzzy partitions) and/or for different membership functions to be encountered in the source and target domains. The fuzzy partitions (describing membership functions) are designed by running DIC fuzzy clustering [28]. This method forms the membership functions for each feature based on the distribution of data in a domain. This assumption implies a

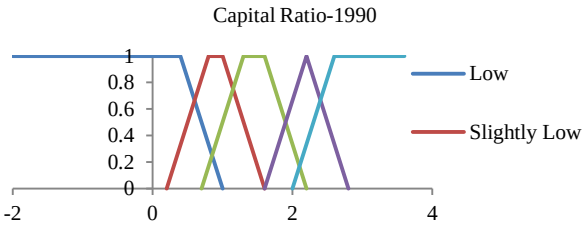


Fig. 8. Capital Ratio fuzzy feature as an example in fuzzy feature space in vear 1990

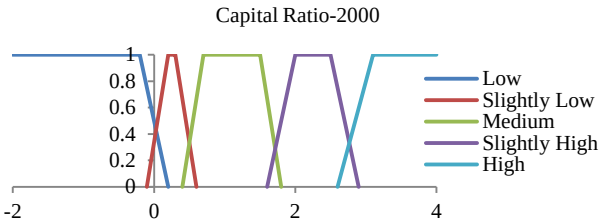


Fig. 9. Capital Ratio fuzzy feature as an example in fuzzy feature space in year 2000

transductive transfer learning problem in which the feature space is the same but the distribution of data is different. Fig. 8 and 9 show the *Capital Ratio* as an example of fuzzy feature (f_i) in long-term bank failure prediction in the years 1990 and 2000, respectively. There are five membership functions (linguistic terms) distributed over the range of (-2, 4) in both 1990 and 2000, but the membership functions are different.

Let $Y = \{Y_1, \dots, Y_L\}$ is the predictive fuzzy label set which is the same for both domains D_s and D_t . For instance, in the case of bank failure prediction we have $L = 2$, $Y_1 = (survived)$ and $Y_2 = (Failed)$. $g(\cdot)$ is a learning-based model such as a neural network. If the learning-based model $g(\cdot)$ has been trained using the labeled data coming from the source domain, we will call it a shift-unaware model, denoted by$g(\cdot)|D_s$. Thus, $g(x)|D_s = \left(\mu_{Y_1}(x), \dots, \mu_{Y_L}(x)\right)$ is the vector of membership values of x in each class computed by the shift-unaware learning-based model. Let matrix G denote the unrefined label matrix where g_{ij} is the membership value of a given instance x_i in class j which has been computed by shift-unaware learning-based model $g(\cdot)|D_s$. S is the fuzzy similarity function. Let M denote the similarity matrix where m_{ij} measures the similarity between the given instances x_i and x_j using $S(x_i, x_j)$. Likewise MR^1 stands for the refined label matrix in first iteration where mr_{ij}^1 is the refined membership value of given instance x_i in class j after the first iteration of the refinement. The following expression describes the first iteration of this refinement process:

$$mr_{ij}^1 = \alpha\left(\frac{\sum_{x_0 \in N_i^D} S(x_i, x_0)\left(\mu_{Y_j}(x_i) - \mu_{Y_j}(x_0)\right)}{|N_i^D|}\right) + (1-\alpha) g_{ij}, \quad \textbf{(2)}$$

where $N_i^D$is a set of instances most similar to instance x_i that can be extracted from a given domain D using the similarity matrix M. The sample x_0 is one of the k-most similar

instances to x_i and it comes from a mixture domain composed of source and target domains. If x_0 belongs to source domain or 5% labelled samples in target domain then $\mu_{Y_j}(x_0)$ will be the actual label. If it belongs to an originally unlabeled sample from the target domain, then $\mu_{Y_j}(x_0)$ will be the unrefined or refined (in previous steps) predicted label that is produced by such shift-unaware models as SVM, NN or FNN. According to the refined Equation 1, we calculate the difference between the label values of most similar instances and that of the given instance $\left(\mu_{Y_j}(x_i) - \mu_{Y_j}(x_0)\right)$. This is multiplied by the similarity value between most similar instances and the given instance $S(x_i, x_0)$ to amplify the influence of more similar instances on refinement. Finally, the average value is used to refine the unrefined label values g_{ij} by an impact factor of α. Given α is the parameter, which is used to specify the impact of refinement, the final refined label at each step is an α-weighted convex sum of: 1) the refined label computed by most similar samples' labels, and 2) the initial label computed by a given prediction model. Hence, α is a parameter in (0, 1) that specifies the contribution of refinement in the final refined label of the target sample, i.e. the higher the value of the alpha, the more evident the impact of the refinement becomes. To calculate the refined value in the subsequent iterations, the label values computed in the prior iteration are applied as follows:

$$mr_{ij}^{(t+1)} = \alpha\left(\frac{\sum_{x_0 \in N_i^D} S(x_i, x_0)\left(mr_{ij}^{(t)} - mr_{x_0 j}^{(t)}\right)}{|N_i^D|}\right) + (1-\alpha) g_{ij} \quad \textbf{(3)}$$

According to (3), the refined labels of instances in the previous iterations are used to further adjust the label values in the current iteration. The use of this approach leads to a mutual reinforcement relationship between instances in domains that can help to transfer the label pattern from the source domain to the target domain and consequently improve accuracy.

Example 3: As a simple illustration of the refinement process just described, we look at a binary classification problem in which x_0 is the given instance such that $g(x_0)|D_s = (0.5, 0.5)$ and $N_0^D = \{x_1, \dots, x_{10}\}$ is the set of instances that are the most similar to x_0, such that $g(x_i)|D_s = (\frac{i}{20}, \frac{20-i}{20})$ and $S(x_i, x_0) = \frac{10-0.5i}{10}$. The instances are depicted in Fig. 10.

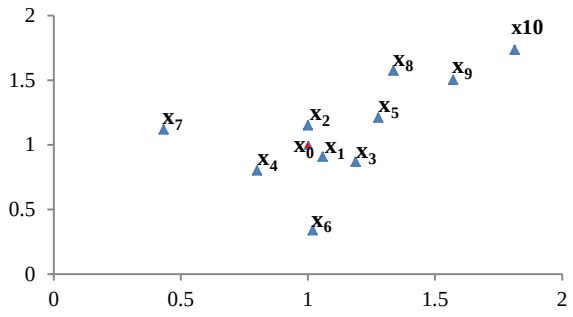


Fig. 10. The instances of Example 1

TABLE III
LABEL VALUE OF THE GIVEN INSTANCE IN TEN CONSECUTIVE ITERATIONS

Iteration	1	2	3	…	8	9	10
$g_1(x_0)$	0.5	0.1977	0.3186	…	0.2842	**0.2840**	**0.2841**
$g_2(x_0)$	0.5	0.4022	0.4413	…	0.4293	**0.4304**	**0.4300**

Eq. (3) is applied to refine the label of instance x_0 through a series of iterations until the refinement function converges by 10^{-3}. The results are reported in Table III. As can be seen in the last two columns of the table, the refinement function converges after ten iterations: $|0.2841 - 2840| < 10^{-3}$. Since most similar instances belong to class 2, the membership value of instance x_0 to class 1 decreases, while its membership value to the second class increases. We conclude that the given instance is classified as belonging to class 2.

The refinement expression (3) is applied in the proposed MFBRDA algorithm. This refinement is based on the fact that the labels of the most similar instances to the target instance are used to modify the initial label of the target instance, which has been initialized by the shift-unaware model. As a result, the refined fuzzy label matrix for all unlabeled instances of target domain MR^n is formed as follows:

$$MR^n = \begin{bmatrix} mr_{11}^n & \cdots & mr_{1L}^n \\ \vdots & \ddots & \vdots \\ mr_{|D_t|1}^n & \dots & mr_{|D_t|L}^n \end{bmatrix}$$

Each row of this matrix shows the membership value of one instance to all label classes. To find the final label for each instance, the following expression is used:

Label $(x_i) = argmax_i\{mr_{ij}^n | j = 1, \dots, L\}$. [21, 22]

V. EXPERIMENTS

In this section, we report on the experiments completed for the widely used 20*Newsgroup* data set. Different settings of the proposed algorithm are described and the data set specifications are explained. The baselines are also introduced for benchmarking purposes. Finally, the experimental results are obtained and analyzed.

A. Settings

The algorithm was realized using 9 different settings based on different mixture domains D_w and the number of steps in Phase 2 (Refinement presented in Table IV). All settings are divided into three categories. Categories 1 and 2 refer to the settings with one and two steps of refinement, respectively. Category 3 contains the settings with three and $\boldsymbol{n}$ steps of refinement. Hence, each category indicates the number of iterations of the refinement process with different possible mixture domains. Table IV shows that Settings 8 and 9 are similar; however, in Setting 9, we use a small number of labeled instances (5% of samples) of the target domain in mixture domains to examine the influence of labeled target data on the performance of the proposed algorithm. These labeled data are used for training all baselines as well as the refinement process in the proposed algorithm.

TABLE IV
DIFFERENT SETTINGS OF MFBRDA ALGORITHM USED IN EXPERIMENTS

Category	Setting ID	N.O Steps (Iterations)	Mixture Domain (D_w)
1	1	1	Step1: $D_m \subset D_s$
	2	1	Step1: $D_m \subset D_t$
	3	1	Step1: $D_m \subset D_s \cup D_t$
2	4	2	Step 1: $D_m \subset D_s$
			Step 2: $D_m \subset D_t$
	5	2	Step 1:$D_m \subset D_s$
			Step2: $D_m \subset D_s \cup D_t$
	6	2	Step 1: $D_m \subset D_s \cup D_t$
			Step 2: $D_m \subset D_t$
3	7	3	Step 1: $D_m \subset D_s$
			Step 2: $D_m \subset D_s \cup D_t$
			Step 3: $D_m \subset D_t$
	8	n	$D_w = (1-\mu_w)\lvert D_s\rvert + \mu_w\lvert D_t\rvert$
	9	n	$D_w = (1-\mu_w)\lvert D_s\rvert + \mu_w\lvert D_t\rvert$ with labeled target data

B. Data set and Preparation

We validate the proposed algorithm by using a commonly used data set, namely *20Newsgroup* (http://people.csail.mit.edu/jrennie/20Newsgroups/). The different settings mentioned above are investigated. This data collection was not originally designed for transfer learning, so some modification was necessary to make the distribution between the training data and the test data different. The data set has a two-level hierarchical structure. Suppose A and B are two root categories in the data collection, and A1, A2 and B1, B2 are sub-level categories of A and B, respectively. We form the training and test data as follows. Let A.A1, B.B1 be the positive and negative examples in the training data respectively. Let A.A2, B.B2 be the positive and negative examples in the test data, respectively. The target categories are fixed, as A and B, but the distributions of the training data and the test data are different yet still similar enough for the evaluation of the proposed algorithm in transfer learning. There are seven top level categories of which three have no sub-categories. We compose six data sets from the remaining four categories. The detailed composition of these data sets is provided in Table V.

We pre-process the raw data by putting all letters in lower case, stemming words using the Porter stemmer [30], and removing all stop words. According to [31], the DF Thresholding can achieve comparable performance to Information Gain or CHI, but it is much easier to implement and less costly both in time and space requirements. We therefore use it to cut down the number of words/features and speed up the classification. The words that occur in fewer than three documents are removed. Each document is then converted into a bag of-words presentation in the remaining feature space. Each value of the feature is the term frequency of that word in the document, weighted by its IDF (log N/DF).

To examine the performance of the MFBRDA algorithm, we select three different shift-unaware prediction models: Fuzzy Neural Network (FNN) [9]; Support Vector Machine (SVM) [7]; Multi-Layer Perception Neural Network (MLPNN) [8] and K-Nearest Neighbor (KNN) [32]. In SVM, we use the implementation SVMlight [33] with a linear kernel and all options set by default. In MLPNN, we choose the optimum parameters reported in [34] which include a three layer structure neural network with 15 hidden nodes along with a singular value decomposition (SVD) technique for dimension reduction. In KNN, different values of k= 5, 15, 25, 35, 45, 55, 65; have been tried to ensure that the experimental results faithfully reflect the performance of the algorithm. The value of k=45 is the optimum for the k-NN algorithm. *Discrete Incremental Clustering* (DIC) [28], which is a novel self-organizing clustering technique, is applied to create the fuzzy features. DIC applies a fully automated clustering that forms the membership functions according to numerical data distribution on each problem region attribute. The main reason for choosing this method is that it creates the membership functions by a mechanism that can depict the data distribution of numerical data to fuzzy membership functions. Although the data used in the experiments are numeric and MLPNN, SVM, FNN and KNN are trained using these data, our study provides a cluster-based representation of the domain region by fuzzy clustering (based on the fact that the fuzzy sets and membership functions are formed by DIC) rather than distribution-based representation of data. In the proposed MFBRDA algorithm, we use the fuzzy similarity/dissimilarity measure presented in [27] to compute the similarity between samples in the fuzzy cluster-based decision region.

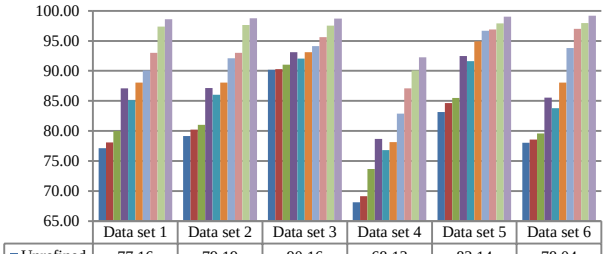
TABLE V
20NEWSGROUP DATA COLLECTION AND ITS DETAILED COMPOSITION DATA SETS

Data set	Train/Test data	Positive	Negative	Number of samples
1	Train	rec.autos rec.motorcycles	talk.politics.guns talk.politics.misc	3669
	Test	rec.sport.baseball rec.sport.hockey	talk.politics.mideast talk.religion.misc	3561
2	Train	rec.autos rec.sport.baseball	sci.med sci.space	3961
	Test	rec.motorcycles rec.sport.hockey	sci.crypt sci.electronics	3954
3	Train	comp.graphics comp.sys.mac.hardware comp.windows.x	talk.politics.mideast talk.religion.misc	4482
	Test	comp.os.ms-windows.misc comp.sys.ibm.pc.hardware	talk.politics.guns talk.politics.misc	3652
4	Train	comp.graphics comp.os.ms-windows.misc	sci.crypt sci.electronics	3930
	Test	comp.sys.ibm.pc.hardware comp.sys.mac.hardware comp.windows.x	sci.med sci.space	4900
5	Train	comp.graphics comp.sys.ibm.pc.hardware comp.sys.mac.hardware	rec.motorcycles rec.sport.hockey	4904
	Test	comp.os.ms-windows.misc comp.windows.x	rec.autos rec.sport.baseball	3949
6	Train	sci.electronics sci.med	talk.politics.misc talk.religion.misc	3374
	Test	sci.crypt sci.space	talk.politics.guns talk.politics.mideast	3828

C. *Empirical Results Analysis*

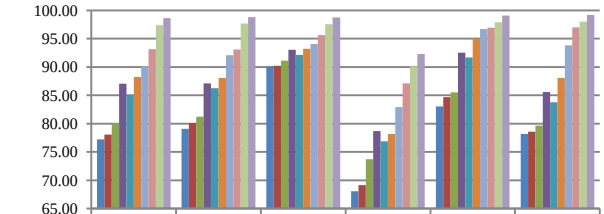
The experimental results show that in all cases the proposed algorithm improves accuracy. As shown in Fig. 11-14, the greatest increase in accuracy is noted for settings 8 and 9 with multiple iterations of refinement and mixture domains of target and source domains in each step. This demonstrates that multi-step refinement significantly improves accuracy and produces better results than other settings with fewer refinement iterations. It can be concluded that the number of refinement steps has a positive influence on performance and is beneficial for boosting accuracy. In the following section, we focus on Setting 9 of the proposed algorithm, which is the most successful of the alternatives being considered. Its results are compared with the unrefined results.

Fig. 11 demonstrates the accuracy of KNN (denoted as Unrefined) compared with nine settings of MFBRDA using KNN as shift-unaware on all 20Newsgroup data sets. The proposed algorithm shows an improvement in all data sets, particularly data set 4, in which the relative increase is 26.1%. The accuracy in Setting 9 increases by 18.9% on average. Fig. 12 shows the accuracy of different settings of the MFBRDA algorithm using SVM on all 20Newsgroup data sets compared with the accuracy of the unrefined results. In all data sets, the proposed algorithm improves accuracy, particularly in data sets 5 and 6, in which the relative increases are 26.9% and 27.1% respectively. The average relative increase of accuracy in Setting 9 over all six data sets is 25%. Fig. 13 reports the results of the proposed algorithm with MLPNN viewed as a shift-unaware classifier. The highest relative enhancements of the accuracy are achieved on data sets 5 and 6, being 26.5% and 26.8% respectively. The average relative increase in Setting 9 is 24.8%. Fig. 14 displays the result of the refinement of the FNN results using the proposed algorithm. The greatest relative increases in accuracy are achieved for data sets 5 and 6, with 26.9% and 27.3% respectively. The average relative increase in Setting 9 is 25.6%.



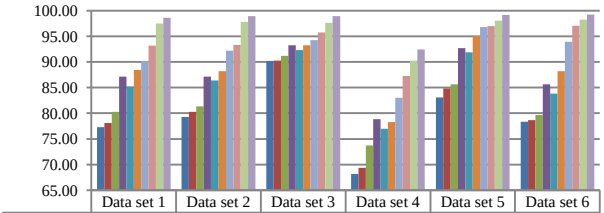
	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5	Data set 6
Unrefined	77.16	79.19	90.16	68.13	83.14	78.04
Setting 1	78.12	80.21	90.27	69.17	84.63	78.58
Setting 2	80.01	81.01	91.04	73.68	85.48	79.60
Setting 3	87.09	87.13	93.09	78.66	92.49	85.57
Setting 4	85.15	86.04	92.05	76.84	91.63	83.77
Setting 5	88.05	88.06	93.11	78.14	94.93	88.07
Setting 6	90.07	92.10	94.14	82.89	96.66	93.79
Setting 7	92.99	92.98	95.61	87.07	96.88	96.99
Setting 8	97.38	97.62	97.52	90.08	97.90	97.97
Setting 9	98.57	98.76	98.69	92.25	99.05	99.16

Fig. 11. Accuracy of the MFBRDA algorithm when using K-NN as a shift-unaware classification model under 9 settings



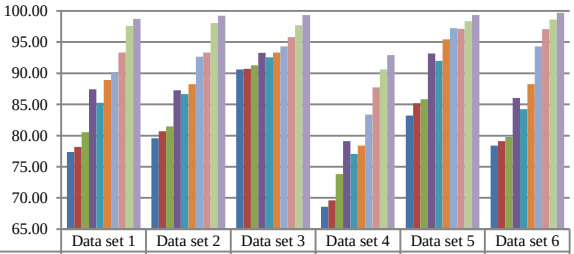
	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5	Data set 6
Unrefined	77.21	79.11	90.09	68.10	83.02	78.18
Setting 1	78.09	80.12	90.20	69.17	84.65	78.60
Setting 2	80.12	81.23	91.10	73.71	85.52	79.63
Setting 3	87.03	87.12	93.03	78.68	92.51	85.59
Setting 4	85.05	86.24	92.15	76.87	91.68	83.78
Setting 5	88.24	88.07	93.22	78.17	94.97	88.09
Setting 6	90.08	92.10	94.03	82.93	96.71	93.81
Setting 7	93.13	93.10	95.63	87.08	96.92	97.01
Setting 8	97.40	97.66	97.54	90.12	97.91	98.02
Setting 9	98.60	98.80	98.73	92.28	99.08	99.18

Fig. 12. Accuracy of the MFBRDA algorithm when using SVM as a shift-unaware classification model under 9 settings



	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5	Data set 6
Unrefined	77.26	79.29	90.21	68.20	83.09	78.35
Setting 1	78.13	80.29	90.23	69.36	84.84	78.68
Setting 2	80.29	81.35	91.21	73.74	85.65	79.67
Setting 3	87.12	87.16	93.22	78.83	92.69	85.64
Setting 4	85.18	86.41	92.34	76.96	91.88	83.85
Setting 5	88.42	88.21	93.23	78.26	95.06	88.18
Setting 6	90.16	92.19	94.22	82.99	96.77	93.96
Setting 7	93.15	93.29	95.73	87.26	97.01	97.06
Setting 8	97.48	97.81	97.60	90.24	98.03	98.21
Setting 9	98.61	98.92	98.90	92.43	99.20	99.22

Fig. 13. Accuracy of the MFBRDA algorithm when using MLPNN as a shift-unaware classification model under 9 settings



	Data set 1	Data set 2	Data set 3	Data set 4	Data set 5	Data set 6
Unrefined	77.37	79.58	90.59	68.57	83.20	78.38
Setting 1	78.20	80.69	90.72	69.61	85.14	79.09
Setting 2	80.53	81.49	91.27	73.81	85.82	79.76
Setting 3	87.42	87.25	93.25	79.11	93.17	86.03
Setting 4	85.24	86.65	92.53	77.03	91.99	84.25
Setting 5	88.90	88.26	93.33	78.37	95.40	88.26
Setting 6	90.19	92.64	94.31	83.34	97.21	94.30
Setting 7	93.29	93.31	95.78	87.70	97.11	97.06
Setting 8	97.55	98.06	97.65	90.59	98.34	98.61
Setting 9	98.69	99.23	99.32	92.89	99.29	99.66

Fig. 14. Accuracy of the MFBRDA algorithm when using FNN as a shift-unaware classification model under 9 settings

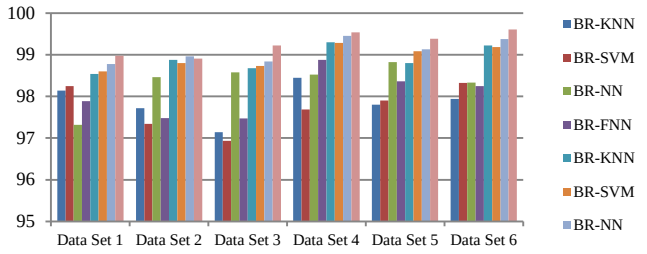


Fig. 15. Accuracy of MFBRDA and BR algorithms using K-NN, SVM, MLPNN and FNN classifiers with 6 data sets

Additionally, we compare the performance of the proposed MFBRDA algorithm (Setting 9) with another refinement algorithm, Bridge Refinement (BR) [7]. This is an algorithm which considers the features with numeric values and uses the Cos function expressed as $((i,) = 1 - c(di,dj))$ as the corresponding distance. The results of the comparison demonstrate the impact of the fuzzy set-based approach on the quality of the results obtained. In the benchmark, four different shift-unaware classifiers; KNN, SVM, MLPNN and FNN are used to determine the initial labels. Fig 15 shows the benchmark results by reporting the accuracy of the FBR and BR algorithms on the Newsgroup data sets. The results clearly show that the FBR algorithm outperforms the BR algorithm in all data sets when different classifiers are used. For instance, the average increase in accuracy achieved by MFBRDA on data sets 3, 4 and 6 is 1.4%, 1.44% and 1.41%, respectively. Similarly, the average increase in accuracy gained by MFBRDA using KNN, SVM, MLPNN and FNN classifiers is 1.3%, 1.2%, 1.1% and 1.0%, respectively. All in all, the fuzzy set-based approach applied to the MFBRDA algorithm significantly improves the refinement performance and boosts accuracy.

VI. Case Study: Long-term Bank Failure Prediction

We demonstrate how the proposed MFBRDA algorithm improves long-term bank failure prediction in a case study of US bank data (1979-2000). We carry out a set of experiments to examine the algorithm's performance in transferring knowledge from different time periods using different features of US bank data. In the experiments, we benchmark the FBR with the bridge refinement (BR) reported in [7] using three different shift-unaware prediction models which compute the initial labels.

A. Case Description

The data set and financial variables are extracted from Call Report Data, downloaded from the website of the Federal Reserve Bank of Chicago .The status of each bank is described according to the Federal Financial Institutions Examination Council. The dataset, shown in Table VI, includes the observation period of the surviving banks of 21 years from Jun 1980 to Dec 2000, based on the history of each bank according to the Federal Financial Institutions Examination Council (FFIEC). There are 548 failed banks and 2,555 banks that survived, as presented in [21, 29]. Although Tung et al. [28] used nine financial features, according to their statistical significance and correlation, it is observed that a model with three features has fewer rules, less computational load, and greater prediction accuracy. Each feature is ranked based on the importance of a feature as a result of a future selection process, and the three features with the highest grade are selected [35]. The definition of all features and their expected impact on bank failure are described in Table VII. The MFBRDA algorithm is run for nine and three features (indicated by *) separately and the results are then compared in terms of the long-term prediction accuracy.

TABLE VI
Number of Available Records in US Data Sets for Each Scenario

Year	Total Number of Banks	Survived banks	Failed banks
1990	2156	1843(85.48%)	313 (14.52%)
1995	2539	2192(86.34%)	347(13.66%)
1998	2943	2585(87.84%)	358(12.16%)
2000	3103	2555(82.34%)	548(17.66%)

TABLE VII
Definition of Variables and Their Impact on Bank Failure

Financial Variable	Expected Effect of Failure
*CAPADE: average total equity capital /average total assets	The higher the ratio, the smaller the probability of failure
OLAQLY: average (accumulated) loan loss allowance/average total loans and leases, gross	The smaller the ratio, the smaller the probability of failure
PROBLO: Loans 90+ days late/average total loans & leases, gross	The higher the ratio, the higher the probability of failure
*PLAQLY: Loan loss provision /average total loans and leases, gross	The higher the ratio, the higher the probability of failure
NIEOIN: noninterest expense/operating income	The higher the ratio, the higher the probability of failure
NINMAR: (total interest income – interest expense)/average total asset	The higher the ratio, the smaller the probability of failure
*ROE: (net income after tax applicable income taxes)/average total equity capital	The higher the ratio, the smaller the probability to failure
LIQUID: (average cash & average federal funds sold)/(average total deposit & average fed funds purchased & average banks' liability on acceptance & average other liabilities)	The higher the ratio, the higher the probability of failure
GROWLA: (total loans & leases (t) – total loans & leases (t-1))/ total loans & leases (t-1)	The higher the ratio, the smaller the probability of failure

B. Case Design and Processing

The source instances are selected from a data set to year 1990. This data set is used as the training data to train shift-unaware prediction models. The target instances are selected as test data from the records for 1995, 1998 and 2000, i.e., 5, 8 and 10 years following 1990. In the domain adaptation problem, the marginal distribution of data on each covariate between the source and target domains is different; hence the main reason for using 1990 data as the source (training) domain and 1995, 1998 and 2000 data as the target (test) domain is to have a significant difference in marginal distribution when the time gap between the source data (1990) and the target data (1995, 1998 and 2000) is big enough (5, 8 and 10 years respectively). We perform the refinement algorithms on the labels predicted by the shift-unaware prediction models, from which we receive the unrefined (initial) labels of target instances. To examine the performance of the MFBRDA algorithm, we select three different shift-unaware prediction models: Fuzzy Neural Network (FNN) [9], Support Vector Machine (SVM) [7], and Multi-Layer Perception Neural Network (MLPNN) [8]. Additionally, we compare the proposed algorithm with another refinement algorithm, Bridge Refinement (BR) [7]. BR is a non-fuzzy algorithm, which applies features with numeric and Euclidian distance as similarity measures. Comparing MFBRDA with BR demonstrates the impact of the fuzzy approach we apply in FBR.

To reduce the influence of the imbalanced data-sets problem, the SMOTE technique is applied to the training data sets. The number of failed banks is increased to the number of survived banks to achieve a balanced data set, which improves the accuracy of prediction without losing important information. In each scenario, the training data set is split into two pools: (1) failed banks denoted by output -1, and (2) survived banks denoted by output +1. There are five groups, each of which randomly includes instances of both pools to form the training set. The training sets of the five groups are mutually exclusive. The model is trained using the training data sets and then evaluated with the testing data sets. The accuracy of the experiment in each scenario, which is the mean value of the accuracy of the cross validation group, is calculated using the Geometric Mean of *sensitivity* and *specificity* [36]. This metric is applied because both are expected to be high. To form fuzzy features, the discrete incremental clustering (DIC) is used [28]. We use the fuzzy similarity/dissimilarity measure addressed by [27] in the MFBRDA algorithm.

C. Analysis of Results

The results of the experiments are reported in Tables VIII-X that include the accuracy and relative increases achieved by the refinement algorithms in long-term bank failure prediction. To compare the performance of the MFBRDA for transfer learning, it is examined using three different shift-unaware prediction models: FNN, SVM, and MLPNN. MFBRDA improves the predictive accuracy of shift-unaware prediction models in all settings. The algorithm is evaluated according to nine different settings to examine the influence of the refinement iterations and mixture domain on the performance of MFBRDA. These settings consider different situations ranging from a scenario in which the source data is only used in a single iteration to the situation in which a number of labeled target data in multiple iterations are utilized. Using different settings, we determine how the different steps and labeled target data can improve the algorithm's performance.

The total relative increases when using MFBRDA after refining the initial labels produced by FNN, SVM and MLPNN are 15.29%, 19.36% and 17.10%, respectively. This shows that the influence of the proposed algorithm becomes more significant once a prediction horizon becomes longer and consequently the difference between the target domain and the source domain becomes greater. For instance, the average relative increase is 11.98%, 17.29% and 19.15% on 1995, 1998 and 2000 data sets, respectively (see Table VIII, where FNN is the shift-unaware prediction model). This increase is even more apparent in Table IX where SVM is the shift-unaware prediction model and the accuracy is improved by 11.68%, 18.59% and 25.74 % on the prediction of a 5, 8 and 10 year time horizon respectively.

TABLE VIII
ACCURACY AND RELATIVE INCREASE OF FBR AND BR USING FNN
(BOLDFACE NUMBERS INDICATE THE AVERAGE ACCURACY AND RELATIVE INCREASE OF EACH CATEGORY)

Category	Setting	1995		1998		2000	
		MFBRDA	BR	MFBRDA	BR	MFBRDA	BR
1	1	0.8315	0.8246	0.7722	0.7618	0.7507	0.7412
		0.96%	0.12%	1.41%	0.04%	2.43%	1.13%
	2	0.8463	0.8377	0.7911	0.7762	0.76	0.7495
		2.76%	1.71%	3.89%	1.93%	3.70%	2.26%
	3	0.8584	0.8483	0.812	0.7903	0.8047	0.7841
		4.23%	3.00%	6.63%	3.78%	9.80%	6.99%
Average		**0.8454**	**0.8368**	**0.7917**	**0.7761**	**0.7718**	**0.7582**
		2.65%	**1.61%**	**3.97%**	**1.92%**	**5.31%**	**3.46%**
2	4	0.8532	0.8473	0.8281	0.805	0.7945	0.7892
		3.59%	2.88%	8.75%	5.71%	8.40%	7.68%
	5	0.8783	0.8711	0.8533	0.8432	0.8302	0.8167
		6.64%	5.77%	12.06%	10.73%	13.28%	11.43%
	6	0.8831	0.8742	0.8555	0.8474	0.8447	0.8239
		7.22%	6.14%	12.34%	11.28%	15.25%	12.42%
Average		**0.8715**	**0.8642**	**0.8456**	**0.8318**	**0.8231**	**0.8099**
		5.82%	**4.93%**	**11.05%**	**9.24%**	**12.31%**	**10.51%**
3	7	0.9153	0.9104	0.8911	0.8721	0.8821	0.8638
		11.13%	10.54%	17.02%	14.52%	20.36%	17.86%
	8	0.9303	0.9196	0.9037	0.8845	0.8908	0.8759
		12.96%	11.66%	18.67%	16.15%	21.54%	19.51%
	9	0.9476	0.9279	0.9112	0.8895	0.9013	0.8862
		15.06%	12.66%	19.66%	16.81%	22.98%	20.92%
Average		**0.9310**	**0.9193**	**0.9020**	**0.8820**	**0.8914**	**0.8753**
		13.05%	**11.6%**	**18.45%**	**15.8%**	**21.63%**	**19.43%**
Unrefined (FNN)		0.8236	0.8236	0.7615	0.7615	0.7329	0.7329

TABLE IX
ACCURACY AND RELATIVE INCREASE OF FBR AND BR USING SVM
(**BOLDFACE** NUMBERS INDICATE THE AVERAGE ACCURACY AND RELATIVE INCREASE OF EACH CATEGORY)

Category	Setting	1995		1998		2000	
		MFBRDA	BR	MFBRDA	BR	MFBRDA	BR
1	1	0.7966	0.7964	0.761	0.7544	0.7117	0.7062
		0.52%	0.49%	4.28%	3.37%	4.62%	3.81%
	2	0.801	0.7986	0.7645	0.7569	0.7166	0.7106
		1.07%	0.77%	4.75%	3.71%	5.34%	4.45%
	3	0.8081	0.8019	0.7712	0.7626	0.7196	0.7131
		1.97%	1.19%	5.67%	4.49%	5.78%	4.82%
Average		**0.8819**	**0.8819**	**0.8773**	**0.8656**	**0.858**	**0.856**
		11.28%	**1.19%**	**0.82%**	**4.90%**	**3.86%**	**5.24%**
2	4	0.805	0.7963	0.764	0.7594	0.7171	0.7102
		1.58%	0.48%	4.69%	4.06%	5.41%	4.40%
	5	0.8494	0.8429	0.8131	0.8037	0.7631	0.7545
		7.18%	6.36%	11.41%	10.13%	12.17%	10.91%
	6	0.8729	0.8591	0.8301	0.8185	0.7795	0.768
		10.15%	8.40%	13.74%	12.15%	14.58%	12.89%
Average		**0.8924**	**0.8924**	**0.8828**	**0.8724**	**0.8639**	**0.8632**
		12.61%	**6.30%**	**5.08%**	**9.95%**	**8.78%**	**10.72%**
3	7	0.8935	0.8856	0.8433	0.8354	0.8154	0.8061
		12.74%	11.75%	15.55%	14.47%	19.86%	18.49%
	8	0.9194	0.8905	0.8947	0.8729	0.8734	0.8619
		16.01%	12.37%	22.60%	19.61%	28.38%	26.69%
	9	0.9288	0.909	0.9022	0.8852	0.8812	0.8682
		17.20%	14.70%	23.62%	21.29%	29.53%	27.62%
Average		**0.9139**	**0.9139**	**0.895**	**0.8901**	**0.8745**	**0.8733**
		15.32%	**15.32%**	**12.94%**	**20.59%**	**18.46%**	**25.92%**
Unrefined (SVM)		0.7925	0.7925	0.7925	0.7298	0.7298	0.6803

TABLE X
ACCURACY AND RELATIVE INCREASE OF FBR AND BR USING NN (BOLDFACE NUMBERS INDICATE THE AVERAGE ACCURACY AND RELATIVE INCREASE OF EACH CATEGORY)

Category	Setting	1995		1998		2000	
		MFBRDA	BR	MFBRDA	BR	MFBRDA	BR
1	1	0.8412	0.8395	0.7767	0.7625	0.7315	0.7143
		0.80%	0.60%	3.11%	1.22%	4.43%	1.97%
	2	0.8469	0.8391	0.789	0.7798	0.7439	0.7321
		1.49%	0.55%	4.74%	3.52%	6.20%	4.51%
	3	0.8504	0.8435	0.7998	0.787	0.7393	0.7222
		1.91%	1.08%	6.17%	4.47%	5.54%	3.10%
Average		**0.8462**	**0.8407**	**0.7885**	**0.7764**	**0.7382**	**0.7229**
		1.40%	**0.74%**	**4.67%**	**3.07%**	**5.39%**	**3.19%**
2	4	0.8466	0.8411	0.7833	0.7786	0.7303	0.7211
		1.45%	0.79%	3.98%	3.36%	4.25%	2.94%
	5	0.8754	0.868	0.8164	0.8057	0.7658	0.7592
		4.90%	4.01%	8.38%	6.96%	9.32%	8.38%
	6	0.9023	0.8952	0.8385	0.8222	0.7838	0.7728
		8.12%	7.27%	11.31%	9.15%	11.89%	10.32%
Average		**0.8748**	**0.8681**	**0.8127**	**0.8022**	**0.7600**	**0.7510**
		4.83%	**4.03%**	**7.89%**	**6.49%**	**8.49%**	**7.21%**
3	7	0.915	0.9068	0.8694	0.8552	0.8066	0.7945
		9.65%	8.66%	15.41%	13.53%	15.15%	13.42%
	8	0.9421	0.9301	0.9077	0.8955	0.8641	0.8512
		12.89%	11.46%	20.50%	18.88%	23.35%	21.51%
	9	0.9475	0.9393	0.9129	0.9061	0.8765	0.8589
		13.54%	12.56%	21.19%	20.28%	25.12%	22.61%
Average		**0.9349**	**0.9254**	**0.8967**	**0.8856**	**0.8491**	**0.8349**
		12.03%	**10.89%**	**19.03%**	**17.56%**	**21.21%**	**19.18%**
Unrefined (NN)		0.8345	0.8345	0.8345	0.7533	0.7533	0.7005

If we compare the accuracy of the settings in each category, a growing trend with respect to categories 1, 2 and 3 can be identified, as shown in Fig. 16. This demonstrates the average relative growth in accuracy for three shift-unaware prediction models. The main reason for this trend is that a single-step refinement is applied in this category, while a multistep refinement mechanism is used in other categories. Likewise, Category 3 outperforms Category 2. It can be concluded that if the number of refinement iterations becomes larger, the settings in each category become more accurate. One reason for this finding is that if more refinement iterations are applied, the structure of the data sets can be transferred from the source domain to the target domain in a more accurate manner.

To examine the influence of the fuzzy set-based approach on the performance of MFBRDA, we compare it with that of BR using the three different shift-unaware prediction models. The proposed algorithm outperforms the BR in all settings of three shift-unaware prediction models. To attest to this improvement, we apply a nonparametric statistical procedure to test whether the accuracy achieved by MFBRDA is significantly better than the accuracy of BR. Table XI summarizes the results of the Holm test [37] completed at the 0.05 confidence level. The hypothesis of equality of the FBR and BR accuracies is rejected.

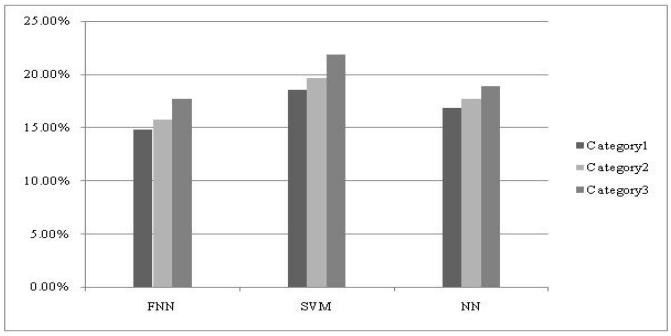


Fig. 16. Average relative increase in accuracy for different categories

TABLE XI
RESULTS OF THE HOLM TEST FOR THE ACCURACIES PRODUCED BY THE FBR AND BR METHODS

Comparison	$z = (R_0 - R_i)/SE$	P	α-Holm	Hypothesis
FBR vs. BR in 1995	5.196	2.035E-7	0.05	Rejected
FBR vs. BR in 1998	5.196	2.035E-7	0.05	Rejected
FBR vs. BR in 2000	5.196	2.035E-7	0.05	Rejected

D. *Sensitivity of Parameters*

The proposed MFBRDA algorithm comes with two parameters, k and α, whose values need to be set. In this section, we investigate the influence of these parameters on the performance of the MFBRDA algorithm. To do this, the performance of the algorithm is examined using different values of the parameters on all settings in each scenario. As an example, the accuracy gained by the MFBRDA algorithm for different values of k in three categories of settings on year 2000 is shown in Fig. 17. This figure shows that although the performance is not highly sensitive with regard to k as long as k is large enough, the settings in category 3 need smaller k than those in categories 1 and 2 which implies lower computational complexity in category 3 than in categories 1 and 2. Furthermore, the average values of 70, 75 and 80 for category 3, 2 and 1, respectively are the optimal values for k and as such were chosen in this research.

To find the optimal value α in the range [0, 1], the accuracy of MFBRDA is examined for all settings of each category. Fig. 18 shows the average prediction accuracy by applying different values of α for all categories and years. It also shows that the optimum value of α in category 3 is more than it is in category 2. The value in category 2 is also larger than the in category 1. This can be explained as follows: if the input data are more comprehensive, the refinement is more effective and the value of α is higher. The optimal average values of α are 0.65, 0.68, and 0.74 in categories 1, 2 and 3, respectively.

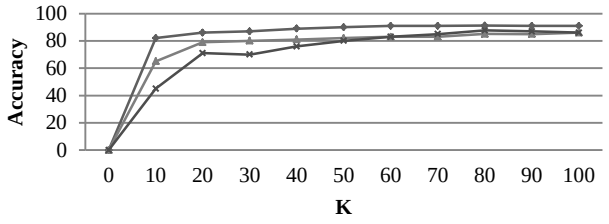


Fig. 17. The accuracy of FNN-RF when using different values of k

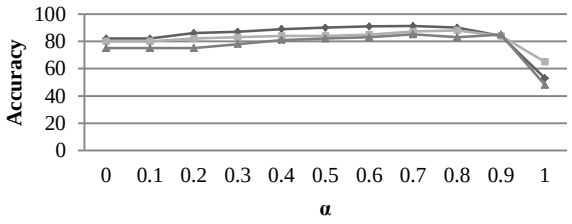


Fig. 18.The accuracy of FNN-RF when using different values of α

VII. Comparative Analysis

In this section, a comprehensive empirical analysis is performed to compare the performance of proposed MFBRDA algorithm with that of the existing commonly used and advanced domain adaptation methods. The experiments are carried out using five different data sets and the results are examined using statistical tests.

A. Data Sets

The experimental analysis is carried out on five different data sets including: 1) 20newsgroup; 2) Bank failure prediction; 3) SRAA3 and 4) Reuters-2157833. 20Newsgroup and Bank failure prediction data sets have been explained in Sections 5 and 6 respectively. In SRAA, there are four discussion groups: simulated autos (simauto), simulated aviation (simaviation), real autos (realauto), and real aviation (realaviation). As shown in Table 11, we compose two data sets from SRAA. In Reuters-21578, there are three major groups of places, peoples and orgs. As shown in Table XII, we compose three data sets from Reuters-21578. Like the 20Newsgroup data set, we pre-process the raw data of SRAA and Reuters-21578 by putting all letters in lower case, stemming words using the Porter stemmer [30], and removing all stop words. According to [31], DF Thresholding achieves comparable performance to Information Gain or CHI, but is much easier to implement and less costly both in time and space requirements. Hence we use it to cut down the number of words/features and speed up classification. The words that occur in fewer than three documents are removed. Each document is then converted into a bag of-words presentation in the remaining feature space. Each value of the feature is the term frequency of that word in the document, weighted by its IDF (log N/DF). In the Email spam data set, there are three email subsets (denoted by User1, User2, and User3, respectively) annotated by three different users. The task is to classify spam and non-spam emails. Since the spam and non-spam emails in the subsets have been differentiated by different users, the data distributions of the three subsets are related but different. Each subset has 2,500 emails, in which one half of the emails are non-spam (labeled as 1) and the other half are spam (labeled as -1). On this data set, we consider three settings: 1) User1 (source domain) and User2 (target domain); 2) User2 (source domain) and User3 (target domain); and 3) User3 (source domain) and User1 (target domain).

TABLE XII
SRAA and Reuters-21578 data collection and its detailed composition data sets

Data set	Train/Test data	Positive	Negative	Number of samples
SRAA 1	train	simauto	simaviation	8000
	test	realauto	realaviation	8000
SRAA 2	train	realaviation	simaviation	8000
	test	realauto	simauto	8000
Reuters 1	train	orgs	places	1078
	test	orgs	places	1080
Reuters 2	train	people	places	1239
	test	people	places	1210
Reuters 3	train	orgs	people	1016
	test	orgs	people	1046

B. Empirical Results Analysis

In our experimental analysis, we aim to compare the performance of the FMBRDA algorithm with that of three existing domain adaptation methods. We compute the accuracy of the best setting (Setting 9) of the BR [7] and FMBRDA algorithms when three different shift-unaware prediction models (FNN, SVM, MLPNN) are used. Then the average accuracy of these three models, indicated by $BR^{-average}$ and $MFBRDA^{-average}$, is calculated for comparison analysis. The performance of $BR^{-average}$ and $MFBRDA^{-average}$ is benchmarked with Sample Selection Bias (SSB) [10] and Domain Adaptation Support Vector Machine (DASVM) [11]. As can be seen from Table XIII, the experiments are performed using four real-world data collections, namely 20Newsgroup, Bank failure prediction, SRAA and Reuters-21578, composed of six, three, two and three data sets respectively. For each method, we randomly sample the training data five times and report the mean and standard deviation of each method. Finally the results are examined by a nonparametric statistical procedure Holm test [37].

Table XIII shows that $MFBRDA^{-average}$ outperforms other existing methods for all data sets. For instance, the average improvements that it achieves in 20Newgroups are 1.15, 1.16 and 0.67 in contrast to $BR^{-average}$, SSB and DASVM respectively. It also outperforms $BR^{-average}$, SSB and DASVM in Bank failure prediction by 1.6, 1.3 and 0.87 respectively, on average. $MFBRDA^{-average}$ improves the average accuracy in SRAA by 1.8, 1.65 and 0.95, and in Reuters-21578, by 1.33, 1.6 and 0.67 in contrast to $BR^{-average}$, SSB and DASVM. To attest to this improvement, we apply the Holm test [37] to test whether the accuracy achieved by $MFBRDA^{-average}$ is significantly better than the accuracy of other domain adaptation methods. The bold face values in the $MFBRDA^{-average}$ column indicate the rejection of the hypothesis of equality of the accuracies at the 0.1 confidence level. As shown, the proposed algorithm significantly outperforms other methods in 12 out of 14 data sets.

TABLE XIII
The accuracy and standard deviation of benchmarking domain adaptation methods on four data sets

Data Sets	$BR^{-average}$	SSB	DASVM	$MFBRDA^{-average}$
20Newsgroup				
Data set 1	97.5±0.3	97.3±0.2	97.9±0.1	98.8±0.1
Data set 2	98.2±0.1	98.1±0.4	98.7±0.1	98.9±0.2
Data set 3	97.5±0.4	97.5±0.3	98.0±0.3	98.9±0.2
Data set 4	97.9±0.6	98.0±0.5	98.6±0.7	99.5±0.5
Data set 5	98.8±0.3	98.9±0.2	99.0±0.3	99.2±0.2
Data set 6	97.9±0.7	97.9±0.5	98.5±0.6	99.4±0.8
Bank Failure Prediction				
Data set 1	92.5±0.2	93.1±0.3	93.6±0.1	94.2±0.1
Data set 2	89.3±0.3	89.5±0.3	90.2±0.4	90.9±0.3
Data set 3	87.1±0.5	87.1±0.7	87.3±0.6	88.6±0.5
SRAA				
Data set 1	93.0±0.2	93.3±0.4	93.7±0.1	94.8±0.2
Data set 2	87.2±0.2	87.2±0.5	88.2±0.2	89.0±0.3
Reuters-21578				
Data set 1	63.4±0.2	63.5±0.2	63.8±0.3	65.1±0.3
Data set 2	81.1±0.3	79.8±0.4	81.5±0.2	82.1±0.4
Data set 3	75.3±0.6	75.7±0.3	76.5±0.3	76.6±0.4

VIII. Conclusions and Further Study

The research challenge in this study was to develop a domain adaptation algorithm which can be made independent of the shift-unaware model and work with any given model. The objective of this study was to develop a domain adaptation algorithm that would be able to handle uncertainty of data and deal with vague (non-numeric) values of the features and class labels. The multi-step Fuzzy Bridge Refinement Domain Adaptation (MFBRDA) algorithm was proposed using a fuzzy similarity-based local learning approach. The experimental results obtained show that the proposed MFBRDA algorithm brings about a remarkable improvement in performance. A significant increase in predictive accuracy has been reported, in particular when the algorithm uses three iterations and utilizes a number of labeled target data along with source data and unlabeled target data. The results show that the MFBRDA has even better performance when applied to the long-term prediction horizon.

It is worth noting that compared to an existing refinement method called BR, the MFBRDA algorithm applies fuzzy sets to modify the initial prediction and, according to the empirical results, it substantially outperforms the BR method. The MFBRDA is independent of the prediction model and can be applied with other methods. We have shown that the MFBRDA can successfully transfer knowledge over a long time period to predict bank failure 10 years ahead. The MFBRDA can be considered as an applicable prediction model which does not need to be retrained for every period. Additionally, it can be applied to scenarios in which there is an insufficient number of recent training data.

Our future studies will focus on three tasks. One is to use other prediction or classification models such as fuzzy case-based reasoning and fuzzy rule-based learning models to realize transfer learning. Another is to develop a method, based on the proposed algorithm, which can extract the relevant features to reduce the difference between domains. Finally, an interesting and promising direction could be to examine the performance of the proposed algorithm in contrast to other transfer learning methods, using different real-world data sets.

References

[1] S. J. Pan and Q. Yang. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering, (22)*, pp. 1345-1359.

[2] X. Zhu, "Semi-Supervised Learning Literature Survey," Computer Sciences TR 1530, University of Wisconsin, Madison, 2005.

[3] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning,(39)*, pp. 103-134.

[4] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Eleventh Annual Conference on Computational Learning Theory*, Madison, WI, 1998, pp. 92-100.

[5] T. Joachims, "Transductive inference for text classification using support vector machines," in *Sixteenth International Conference on Machine Learning*, Bled, Slovenia, 1999, pp. 200-209.

[6] G. P. C. Fung, J. X. Yu, L. Hongjun, and P. S. Yu. (2006). Text classification without negative examples revisit. *IEEE Transactions on Knowledge and Data Engineering,(18)*, pp. 6-20.

[7] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged refinement for transfer learning," in *11th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Warsaw, Poland, 2007, pp. 324-335.

[8] C.-T. Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[9] V. Behbood, J. Lu, and G. Zhang, "Adaptive Inference-based learning and rule generation algorithms in fuzzy neural network for failure prediction," in *IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE)*, Hangzhou, China, 2010, pp. 33-38.

[10] J. Huang, A. J. Smola, A. Gretton, K. M. Borgwardt, and B. Scholkopf. (2007). Correcting sample selection bias by unlabeled data. *Advances in Neural Information Processing Systems,(19)*, pp. 601-608, 2007.

[11] L. Bruzzone and M. Marconcini. (2010). Domain adaptation problems: A DASVM classification technique and a circular validation strategy. *IEEE Transactions on Pattern Analysis and Machine Intelligence,(32)*, pp. 770-787.

[12] M. Sugiyama, S. Nakajima, H. Kashima, P. Von Buenau, and M. Kawanabe. (2008). Direct importance estimation with model selection and its application to covariate shift adaptation. *Advances in Neural Information Processing Systems,(20)*, pp. 1433-1440, 2008.

[13] J. Quinonero-Candela, M. Sugiyama, A. Schwaighofer, and N. Lawrence, *Dataset Shift in Machine Learning*. Cambridge: The MIT Press, 2009.

[14] J. Blitzer, R. McOnald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Conference on Empirical Methods in Natural Language Processing*, Sydney, Australia, 2006, pp. 120-128.

[15] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Twenty-First Annual Conference on Neural Information Processing Systems*, Cambridge, MA, 2007, pp. 245-252.

[16] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification," in *45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republi, 2007, pp. 440-447.

[17] W. Dai, G.-R. Xue, Q. Yang, and Y. Yu, "Co-clustering based classification for out-of-domain documents," in *13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,*, San Jose, CA, 2007, pp. 210-219.

[18] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Singapore, Singapore, 2008, pp. 627-634.

[19] Z. Deng, Y. Jiang, F. Chung, H. Ishibuchi, and S. Wang. (2013). Knowledge-leverage based fuzzy system and its modeling. *IEEE Transactions on Fuzzy Systems, 21(4)*, pp. 1-17.

[20] M. Seera, C.P. Lim. (2014). Transfer learning using the online fuzzy min-max neural network neural. *Computing and Applications,25(2)*, pp. 469-480.

[21] M. Seera, C.P. Lim, D. Ishak, H. Singh. (2012). Application of the fuzzy min–max neural network to fault detection and diagnosis of induction motors. *Neural Computing and Applications, 23(1)*, pp. 191-200.

[22] V. Behbood, J. Lu, G. Zhang, "Long term bank failure prediction using fuzzy refinement-based transductive transfer learning", in *IEEE International Conference on Fuzzy Systems*, Taipe, Taiwan, 2011, pp. 2676-2683.

[23] V. Behbood, J. Lu, G. Zhang. (2013). Fuzzy bridged refinement domain adaptation: Long-term bank failure prediction. *International Journal of Computational Intelligence and Applications, 12 (1)*, pp. 135-152.

[24] J. Shell, S. Coupland. (2012). Towards Fuzzy Transfer Learning for Intelligent Environments. *Ambient Intelligence Lecture Notes in Computer Science Volume, (7683)*, pp 145-160.

[25] Z. Deng, K.-S. Choi, Y. Jiang, S. Wang. (2014). Generalized hidden-mapping ridge regression, knowledge-leveraged inductive transfer learning for neural networks, fuzzy systems and kernel methods. *IEEE Transactions on Cybernetics*, doi: 10.1109/TCYB.2014.2311014.

[26] L. Page, S. Brin, R. Motwani, and T. Winograd, "The page rank citation ranking: Bringing order to the Web," in *7th International World Wide Web Conference*, Brisbane, Australia, 1998, pp. 161-172.

[27] W. Wen-June. (1997). New similarity measures on fuzzy sets and on elements. *Fuzzy Sets and Systems,(85)*, pp. 305-309.

[28] W. L. Tung, C. Quek, and P. Cheng. 2004. GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures. *Neural Networks,(17),* pp. 567-587.
[29] H. Le Capitaine, (2012). A Relevance-Based Learning Model of Fuzzy Similarity Measures. *IEEE Transactions on Fuzzy Systems,(20),* pp. 57-68.
[30] M. F. Porter. 1980. An algorithm for suffix stripping. *Program,(14),* pp. 130-137.
[31] Y. Yang and J. O. Pedersen, "A comparative study on feature selection in text categorization," in *Fourteenth International Conference on Machine Learning*, 1997, pp. 412-420.
[32] Mitchell, T.M., *Machine Learning*, McGraw Hill, New York, NY, 1996.
[33] T. Joachims, "Making large-scale support vector machine learning practical," in *Advances in Kernel Methods* Cambridge, MA: MIT Press, 1999, pp. 169-184.
[34] W. Wang and B. Yu. 2009. Text categorization based on combination of modified back propagation neural network and latent semantic analysis. *Neural Computing and Applications,(18),* pp. 875-881.
[35] G. S. Ng, C. Quek, and H. Jiang. 2008. FCMAC-EWS: A bank failure early warning system based on a novel localized pattern learning and semantically associative fuzzy neural network. *Expert Systems with Applications,(34),* pp. 989-1003.
[36] S. Kotsiantis, D. Kanellopoulos, and P. Pintelas. 2006. Handling imbalanced datasets: A review. *GESTS International Transactions on Computer Science and Engineering,(30),* pp. 25-36, 2006.
[37] S. Holm. 1979. A simple sequentially rejective multiple test procedure. *Scandinavian Journal of Statistics,(6),* pp. 65-70, 1979.

V. Behbood (M'2011) received the Ph.D. in software engineering form University of Technology Sydney, Sydney, NSW, Australia, in 2013. His research interests include machine learning, fuzzy sets and systems, data analytic and warning systems.

He is currently working as lecturer in Decision Systems and e-Service Intelligence Research Laboratory of the Centre for Quantum Computation & Intelligent Systems, School of Software, Faculty of Engineering and Information Technology, at the University of Technology Sydney. He has published over 20 papers in Journals and conferences.

J. Lu (SM'2013) received the Ph.D. in Information System from Curtin University of Technology, Perth, WA, Australia in 2001. Her main research interests lie in the area of decision support systems, recommender systems, knowledge-based prediction and warning systems, fuzzy information processing and e-Service intelligence.

She is the Associate Dean Research in the Faculty of Engineering and Information Technology at the University of Technology. She has published five research books and 350 papers in refereed journals and conference proceedings. She has won seven Australian Research Council (ARC) Discovery Project grants and 10 other research grants. She received the first UTS Research Excellence Medal for Teaching and Research Integration in 2010.

Professor Lu serves as Editor-In-Chief for Knowledge-Based Systems (Elsevier), Editor-In-Chief for International Journal on Computational Intelligence Systems (Atlantis), Associate Editor for IEEE Trans on Fuzzy Systems, editor for book series on Intelligent Information Systems (World Scientific), and chairs for ten international conferences as well as having delivered many keynote speeches at international conferences.

G. Zhang (M'2005) received the Ph.D. degree in applied mathematics from Curtin University of Technology, Perth, WA, Australia, in 2001. His main research interests lie in the area of multi-objective, bilevel, and group decision making, decision support system tools, fuzzy measure, fuzzy optimization and uncertain information processing.

He is currently an Associate Professor in the Faculty of Engineering and Information Technology at the University of Technology Sydney, Sydney, Australia. From 1979 to 1997, he was a Lecturer, Associate Professor, and Professor in the Department of Mathematics, Hebei University, China. He has published four monographs, five reference books, and over 300 papers including more than 150 refereed journal articles.

Dr. Zhang has won six Australian Research Council (ARC) discovery grants and many other research grants. He has served, and continues to serve, as a Guest Editor of special issues for four international journals.

W. Pedrycz (F'2009) received the Ph.D. in Electrical Engineering from Silesian Technical University, Gliwice, Poland, in 1980. His main research directions involve computational intelligence, fuzzy modeling and granular computing, knowledge discovery and data mining, fuzzy control, pattern recognition, knowledge-based neural networks, relational computing, and software engineering.

He is Canada Research Chair (CRC) in Computational Intelligence in the Department of Electrical and Computer Engineering, University of Alberta. He has been a member of numerous program committees of IEEE conferences in the area of fuzzy sets and neurocomputing. In 2007 he received a prestigious Norbert Wiener award from the IEEE Systems, Man, and Cybernetics Council. He is a recipient of the IEEE Canada Computer Engineering Medal 2008. In 2009 he has received a Cajastur Prize for Soft Computing from the European Centre for Soft Computing for "*pioneering and multifaceted contributions to Granular Computing*". In 2013 has was awarded a Killam Prize. In the same year he received a Fuzzy Pioneer Award 2013 from the IEEE Computational Intelligence Society. He has published numerous papers in this area. He is also an author of 15 research monographs covering various aspects of Computational Intelligence, data mining, and Software Engineering.

Professor Pedrycz is intensively involved in editorial activities. He is an Editor-in-Chief of Information Sciences and Editor-in-Chief of WIREs Data Mining and Knowledge Discovery (Wiley). He currently serves as an Associate Editor of IEEE Transactions on Fuzzy Systems and is a member of a number of editorial boards of other international journals.