Fuzzy Refinement Domain Adaptation for Long Term Prediction in Banking Ecosystem

Vahid Behbood, *Student Member, IEEE*, Jie Lu, *Member, IEEE*, and Guangquan Zhang

***Abstract*—Long-term bank failure prediction is a challenging real world problem in banking ecosystem and machine learning methods have been recently applied to improve the prediction accuracy. However, traditional machine learning methods assume that the training data and the test data are drawn from the same distribution, which is hard to be met in real world banking applications. This paper proposes a novel algorithm known as fuzzy refinement domain adaptation to solve this problem based on the ecosystem-oriented architecture. The algorithm utilizes the fuzzy system and similarity/dissimilarity concepts to modify the target instances' labels which were initially predicted by a shift-unaware prediction model. It employs a classifier to modify the label values of target instances based on their similarity/dissimilarity to the candidate positive and negative instances in mixture domains. Thirty six experiments are performed using three different shift-unaware prediction models. In these experiments bank failure financial data is used to evaluate the algorithm. The results demonstrate that the proposed algorithm significantly improves predictive accuracy and outperforms other refinement algorithms.**

***Index Terms*—Bank failure prediction, banking ecosystem, domain adaptation, machine learning.**

I. Introduction

THE RAPID growth of digital world leads to the emergence of virtual environments namely digital ecosystems (DES) composed of multiple and independent entities such as organizations, services, software and applications sharing one or several missions and focusing on the interactions and interrelationships among them. Banking DES is one of important areas. With recent crises, long-term bank failure prediction becomes an important issue in banking ecosystem and new techniques have been used to handle this issue. Although machine learning has attracted remarkable attention among researchers in bank failure prediction, most of them work under a common assumption that the training data and test data have the same feature space and distribution. When the distribution or feature space changes, the models need to be rebuilt and retrained from scratch using newly collected training data. For example, labeled financial data quickly goes out of date and may not follow the same distribution over time; thus, past labeled data cannot be used to reliably predict the financial situation of an organization. In many real world applications such as finance and banking ecosystem, collecting new training data and retraining the model is very expensive or practically impossible. It would be extremely useful if the need to collect new labeled data could be reduced and the data from different time periods or domains could be utilized to assist the current learning task.

A new field of study called *transfer learning* has recently emerged. Transfer learning, which is different from traditional machine learning and semisupervised algorithms [1]–[6], considers that the domains and tasks of training and testing data can be different [7]. The study of transfer learning has been inspired by the fact that human beings can utilize previously-acquired knowledge to solve upcoming similar but not identical problems more quickly and effectively than if they did not have this knowledge. The study of transfer learning began in 1995 under a number of different names, such as learning to learn, life-long learning, meta learning, and multitask learning [8].

According to its definition, which will be explained in the next section, transfer learning can be divided into three main categories: 1) *inductive transfer learning* in which the learning task in the target domain is different from that in the source domain [9]; 2) *unsupervised transfer learning* which focuses on solving unsupervised learning tasks in the target domain such as clustering, dimensionality reduction and density estimation; and 3) *transductive transfer learning* in which the learning tasks are the same in both domains while the source and target domains are different. Transductive transfer learning, which was first proposed by Arnold [10], can be further divided into two subcategories: 1) the first, which is referred in the literature as *domain adaptation* [11]–[14], *sample selection bias* [15], and *covariate shift* [16], assumes that the feature spaces of both domains are similar but that the marginal probability distribution of the data is different; and 2) the second called *cross-domain adaptation* [17] assumes that the feature spaces are different in both domains but that they have some features in common. This study aims to solve the problem which can be categorized as domain adaptation.

Almost all studies in transductive transfer learning can be categorized into two groups [8]: 1) transferring the knowledge of instances: this approach is motivated by the importance of samples and an attempt is made to find an optimum weight for each instance to learn a precise model for the target domain. Papers in this category can be found in a recently published book by Quionero–Candela *et al.* [18]; and 2) transferring the knowledge

Manuscript received May 01, 2012; revised July 18, 2012; accepted November 30, 2012. Date of publication January 24, 2013; date of current version May 02, 2014. This work was supported by Australian Research Council (ARC) under Discovery Projects under Grant DP110103733. Paper no. TII-12-0333.

The authors are with Decision Systems and E-Service Intelligence Research Laboratory (DESI), Centre for Quantum Computation and Intelligent Systems (QCIS), School of Software, Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney, NSW 2007, Australia (e-mail: vahid.behbood@uts.edu.au; jie.lu@uts.edu.au; guangquan.zhang@uts.edu.au).

Digital Object Identifier 10.1109/TII.2012.2232935

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.* [19]–[21] 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.* [22] proposed a coclustering based algorithm to propagate the label information across domains. Xue *et al.* [17] presented a cross-domain text classification known as TPLSA to integrate target instances and source instances into a unified probabilistic model.

Of the many researches in this field, most have used statistical models with crisp-value features. In this paper, a novel domain adaptation algorithm is proposed using the fuzzy sets concept, which has been proved to be a triumphant soft computing technique in machine learning [23]–[25]. Likewise, most existing methods aim to refine the decision boundary and models, whereas the proposed algorithm aims to solve the domain adaptation problem using the given test and training data and refining the pseudo initial labels in the target domain by a similarity/dissimilarity-based learning. The similarity/dissimilarity-based learning has been proved to be successful technique in many machine learning applications [26]–[29]. The proposed algorithm constructs a local-based classifier in the mixed distribution of the training and test data as a bridge to transfer the feature distribution from the source domain to the target domain. The output of this classifier is utilized to refine the initial labels in the target domain reported by a shift-unaware model. This ability makes the algorithm more practical and independent from the shift-unaware prediction method. The closest study to our research is the method called bridge refinement [30], which uses similar instances to modify the prediction labels in the target domain. However, the authors have not discussed the conditions and definitions of the similarity function that guarantees the similar instances have similar labels.

Digital ecosystems are based on dynamic interaction of entities which evolves their capabilities and roles over time.

DESs are based on dynamic interaction of entities which evolves their capabilities and roles over time. As a matter of fact, it is evident that dynamism and flexibility are keys in a DES [31] and service oriented architectures (SOA), the prime supporting architectural style of DES, cannot address these features properly [32]. Ecosystem-oriented architecture (EOA) emerged to tackle these issues by considering that there is not any reference model at all and letting the ecosystem entity to converge in dependence with the time, in a model. EOA assert that the components have to be flexible and adaptable enough to support the highly dynamic nature of DES instead of fighting and controlling it. Transfer learning, particularly domain adaptation, which is a new machine learning and data mining framework, is significantly compatible with the EOA approach and can be implemented in many novel applications in DESs. Whereas, most studies to date have been done in the areas of text classification and reinforcement learning, and there is a lack of published novel applications of transfer learning in other areas [33].

We use a novel application, named long-term bank failure prediction, to evaluate and compare the proposed domain adaptation algorithm. Long-term bank failure prediction is a challenging real world application in banking ecosystem and many machine learning methods have been employed to construct accurate computational systems to predict the probability of bank failures. However, they assume a static reference model, in which the test data and training data have the same distribution, to predict the financial failure. Whereas the financial failure is a highly dynamic and time-dependent event in which the data distribution changes over a longer period. The prediction model, which is an entity of banking ecosystem, has to be adequately flexible and adaptable to support the dynamic nature of DES. We apply the EOA approach to handle this problem and create a flexible and adaptable model using proposed fuzzy domain adaptation algorithm.

The rest of the paper is organized as follows. In Section II, preliminary concepts including the definition of transfer learning and similarity/dissimilarity functions and related theorems are addressed. Section III presents the fuzzy refinement domain adaptation algorithm and Section IV describes the experimental illustration and results. Section V concludes the paper and outlines future research.

II. Reliminaries

In this section, the definition of transfer learning and domain adaptation in particular is introduced [8]. The definitions and related theories of similarity and dissimilarity functions and the classifier [28], [34] which is created using these functions are presented.

Definition 2.1: A Domain consists of two components: 1) feature space X; and 2) marginal probability distribution $P(X)$ where $X = \{x_1, \ldots, x_n\}$ domain is denoted by $D = \{X, P(X)\}$.

Definition 2.2: A Task consists of two components: 1) a label space $Y = \{x_1, \ldots, y_m\}$ and an objective predictive function f which is not observed and has to be learned by pairs (x_i, y_j) where $x_i \in X$ and $y_j \in Y$, task is denoted by $T = \{Y, f\}$.

Definition 2.3: 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 in D_t using the knowledge in D_s and T_s, where $D_s \neq D_t$ or $T_s \neq T_t$.

Definition 2.4: Transductive transfer learning is a category of transfer learning in which $T_s = T_t$ and $D_s \neq D_t$ which implies that either $X_t \neq X_s$ or $P_t(X) \neq O_s(X)$.

Definition 2.5: Domain Adaptation is a category of transductive transfer learning in which $X_t = X_s$ but $P_t(X) \neq P_s(X)$.

Definition 2.6: [34]: Let $z_a, z_b, z_c \in X \times \{-1, 1\}$, similarity Si and dissimilarity Di functions are $(\boldsymbol{\varepsilon}, \boldsymbol{\gamma}, \boldsymbol{B}, \boldsymbol{\pi})$**-good** for learning problem if"

a) there are two conditional $pdfs$ such that $\tilde{p}(x|y = 1)/p(x|y = 1) \leq \sqrt{B}$, $\tilde{p}(x|y = -1)/p(x|y = -1) \leq \sqrt{B}$ hold for at least $1 - \pi$ mass probability of instances z;

b) there is a threshold $v(x_b, x_c)$ such that at least $1 - \epsilon$ probability mass of examples z satisfy

$$\tilde{p}\left(Si(x_a, x_b) - Si(x_a, x_c) > v(x_b, x_c) | y_a = y_b, y_a = -y_c\right) \geq 0.5 + \gamma/2$$

or

$$\tilde{p}\left(Di(x_a, x_b) - Di(x_a, x_c) > v(x_b, x_c) | y_a = y_b, y_a = -y_c\right) \geq 0.5 + \gamma/2.$$

Theorem 2.1 [34]: If Si and Di are $(\varepsilon, \gamma, B, \pi)$-good functions, then $n = (16B^2/\gamma^2)\ln(2/\delta)$ positive x_{b_i} and $n = (16B^2/\gamma^2)\ln(2/\delta)$ negative x_{c_i} instances are sufficient so that with the $probability \geq 1 - \delta$, there exists a convex combination classifier $f(x)$ of n base classifiers $h_i(x_a)$

$$f(x_a) = \sum_{i=1}^{n} \omega_i h_i(x_a), \quad \sum \omega_i = 1, \quad \omega_i \geq 0$$

where

$$h_i(x_a) = sgn\left[Si\left(x_a, x_{b_i}\right) - Si\left(x_a, x_{c_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right]$$

or

$$h_i(x_a) = sgn\left[Di\left(x_a, x_{c_i}\right) - Di\left(x_a, x_{b_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right]$$

such that the error rate of the combined classifier at margin $\gamma/2B$ is at most $\boldsymbol{\varepsilon} + \boldsymbol{\delta}$, provided $\boldsymbol{\pi} \leq \gamma^2\delta/\mathbf{64}B^2\ln(2/\delta)$ and the threshold is known.

The proof of the abovementioned Theorem and related examples can be found in [34].

III. Fuzzy Refinement Domain Adaptation Algorithm

The refinement algorithm [30] is motivated by the PageRank algorithm [35]. It assumes that the conditional probability of a specified label C given an instance d, does not vary among different distributions: $P_{D_s}(C|d) = P_{D_{s\cup t}}(C|d) = P_{D_t}(C|d)$ although the marginal probability of instance, $P(d)$, varies. The reason is based on the fact that if an identical instance appears in the target domain and the source domain, the predicted label should be the same. Furthermore, the more similar the instances are in the target domain, the greater is the probability that they have the same label. This situation creates a mutual reinforcement relationship between instances in the target domain and the source domain and can be used to correct the predicted labels. Not only is this assumption considered, but a complementary idea is also applied. It is assumed that the more different the instances are in the target domain, the less is the probability that they have the same label. For instance, in a two class problem, significantly dissimilar instances are located in the opposite classes, while significantly similar instances are located in the same class. In the other words, the similarity and dissimilarity between instances can indicate their class labels. However, the similarity and dissimilarity functions play an important role and need to be defined well enough to map the instances and then discriminate the instances accurately. Wang and Balcan [28], [29] have developed theories for good similarity and dissimilarity functions and have given sufficient conditions for the functions to allow good learning. Hence, the definitions and conditions can be used to select similarity and dissimilarity functions such that there is a high probability that the classifier, which is created using these functions, will indicate the label of a given instance accurately. The definition and theory (Definition 2.5 and Theorem 2.1) proposed by Wang and Balcan [28], [34] are used to define and construct the classifier based on the similarity and dissimilarity measures. The fuzzy approach is utilized in the proposed algorithm to tackle the imprecise nature of financial forecasting in uncertain environment. Using fuzzy approach assists to handle the vagueness in financial features which is the result of uncertainty in finance industry. Also, by applying fuzzy approach the output of proposed algorithm is the pseudo labels which show the membership value of target organization in each class. These labels clearly demonstrate the level of financial healthiness of the organization instead of simply allocating a binary crisp-value label to the organization.

Throughout the paper the labeled instances are presented by $\boldsymbol{z} = (\boldsymbol{x}, \boldsymbol{y}) \in \boldsymbol{X} \times \boldsymbol{Y}$, where $\boldsymbol{x} \in \boldsymbol{X}$ is in fuzzy sets A_k, $k = 1, \ldots, N$ and $\boldsymbol{y} \in \boldsymbol{Y}$ is in fuzzy sets $B_k, k = 1, 2$. $\mu_{A_k}(x)$ is the membership value of instance z in fuzzy set A_k and $\mu_{B_1}(y)$ and $\mu_2(y)$ are the membership values of instance z in positive and negative classes respectively.

Definition 3.1: Let $z_{b_i}, z_{c_i} \in X \times Y \subseteq D$ where D is a given domain, $n = \lfloor(16B^2/\gamma^2) \in (2/\delta)\rfloor$, Si and Di are $(\varepsilon, \gamma, B, \pi)$-good similarity and dissimilarity functions. $\forall z_a \in X \times Y \subseteq D_t$, the classifier $G(x_a)|D = \{G_1(x_a), G_2(x_a)\}$ which is constructed using instances of domain D, is defined as follow:

$$G_1(x_a) = \begin{cases} \left[\left(\frac{1}{n}\right) \sum\limits_{i}^{n} \mu_{B_1}\left(y_{b_i}\right)\right] & if\ f(x_a) > 0 \\ 1 - \left[\left(\frac{1}{n}\right) \sum\limits_{i}^{n} \mu_{B_2}\left(y_{c_i}\right)\right] & if\ f(x_a) < 0 \end{cases}$$

and

$$G_2(x_a) = \begin{cases} 1 - \left[\left(\frac{1}{n}\right) \sum\limits_{i}^{n} \mu_{B_2}\left(y_{c_i}\right)\right] & if\ f(x_a) > 0 \\ \left[\left(\frac{1}{n}\right) \sum\limits_{i}^{n} \mu_{B_2}\left(y_{c_i}\right)\right] & if\ f(x_a) < 0 \end{cases}$$

where

$$f(x_a) = \sum_{i=1}^{n} \omega_i h_i(x_a), \quad \sum \omega_i = 1, \quad \omega_i \geq 0$$

and

$$h_i(x) = sgn\left[Si\left(x_a, x_{b_i}\right) - Si\left(x_a, x_{c_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right]$$

or

$$h_i(x) = sgn\left[Di\left(x_a, x_{b_i}\right) - Di\left(x_a, x_{c_i}\right) + v\left(x_{b_i}, x_{c_i}\right)\right].$$

$G_1(x_a)$ and $G_2(x_a)$ indicate the membership values of the instance z_a in positive and negative classes, respectively. The proposed classifier $G(x_a)$ returns the mean of membership values of positive (negative) instances in the positive (negative) class based on the output of the $f(x_a)$. The proposed classifier and the computed label value are applied to refine the initial label value which is predicted by a shift-unaware model.

Example 3.1: A simple example is presented to illustrate the Definition 3.1. Given D is a 3-dimensional feature space with

TABLE I
POSITIVE SAMPLES IN EXAMPLE 3.1

	x_{b_1}	x_{b_2}	x_{b_3}	x_{b_4}	x_{b_5}	x_{b_6}	x_{b_7}	x_{b_8}	x_{b_9}	$x_{b_{10}}$
f_1	0.86	0.85	0.90	0.86	0.98	0.87	0.85	0.99	0.92	0.86
f_2	0.30	0.25	0.28	0.35	0.25	0.35	0.29	0.29	0.27	0.33
f_3	0.62	0.57	0.52	0.51	0.56	0.50	0.58	0.52	0.51	0.62
μ_{B_1}	0.87	0.84	0.80	0.79	0.83	0.88	0.75	0.76	0.81	0.75
μ_{B_2}	0.12	0.15	0.19	0.20	0.16	0.11	0.24	0.23	0.18	0.24

TABLE II
NEGATIVE SAMPLES IN EXAMPLE 3.1

	x_{c_1}	x_{c_2}	x_{c_3}	x_{c_4}	x_{c_5}	x_{c_6}	x_{c_7}	x_{c_8}	x_{c_9}	$x_{c_{10}}$
f_1	0.45	0.56	0.54	0.45	0.45	0.51	0.55	0.52	0.46	0.45
f_2	0.45	0.46	0.54	0.53	0.51	0.48	0.45	0.49	0.46	0.46
f_3	0.50	0.54	0.52	0.50	0.55	0.60	0.56	0.51	0.58	0.59
μ_{B_1}	0.29	0.28	0.25	0.29	0.26	0.29	0.34	0.27	0.26	0.31
μ_{B_2}	0.70	0.71	0.74	0.70	0.73	0.70	0.65	0.72	0.73	0.68

feature vector: $F = \{f_1, f_2, f_3\}$, Si and Di are $(\varepsilon = 0.05, \gamma = 0.1, B = 0.04, \pi = 0.001)$-good similarity and dissimilarity functions, $\delta = 0.05$ and $v = 0$. So the number of samples is computed as

$$N = \left\lfloor \left(\frac{16B^2}{\gamma^2}\right) ln \left(\frac{2}{\delta}\right) \right\rfloor = \left\lfloor \left(\frac{16 \times 0.04}{0.1^2}\right) ln \left(\frac{2}{0.05}\right) \right\rfloor = 9.$$

It means that we need 9 samples of negative and positive classes to construct the classifier $G(x_a)|D$. Positive and negative samples are presented in Tables I and II, respectively.

Given $x_a = (0.7, 0.3, 0.5)$, Euclidean distance is dissimilarity function Di, $Si = e^{-Di^2/10}$ is similarity function and the equal weight $w_i = 1/9$ is considered. Hence, the value of $f(x_a)$ is computed as

$$\begin{aligned} f(x_a) &= \frac{1}{9}\sum_{i=1}^{9} h_i(x_a) \\ &= \frac{1}{9}\sum_{i=1}^{9} sgn\left[Si\left(x_a, x_{b_i}\right) - Si\left(x_a, x_{c_i}\right)\right] = 0.78 > 0 \end{aligned}$$

or

$$\begin{aligned} f(x_a) &= \frac{1}{9}\sum_{i=1}^{9} h_i(x_a) \\ &= \frac{1}{9}\sum_{i=1}^{9} sgn\left[Di\left(x_a, x_{c_i}\right) - Di\left(x_a, x_{b_i}\right)\right] = 0.78 > 0. \end{aligned}$$

Using $f(x_a)$, the classifier $G(x_a)|D = \{G_1(x_a), G_2(x_a)\}$ is constructed as

$$G_1(x_a) = \left[\left(\frac{1}{n}\right)\sum_{i}^{n} \mu_{B_1}\left(y_{b_i}\right)\right] = 0.81,$$

$$G_2(x_a) = \left[1 - \left(\frac{1}{n}\right)\sum_{i}^{n} \mu_{B_1}\left(y_{b_i}\right)\right] = 0.18.$$

So that, $G(x_a)|D = \{0.81, 0.18\}$ which addresses that x_a belongs to positive class with the error rate at most $\varepsilon + \delta = 0.05 + 0.05 = 0.1$.

Given $x_a = (0.65, 0.35, 0.35)$, Euclidean distance is dissimilarity function Di, $Si = e^{-Di^2/10}$ is similarity function and we consider equal weight $w_i = 1/9$. Hence, the value of $f(x_a)$ is computed as

$$\begin{aligned} f(x_a) &= \frac{1}{9}\sum_{i=1}^{9} h_i(x_a) \\ &= \frac{1}{9}\sum_{i=1}^{9} sgn\left[Si\left(x_a, x_{b_i}\right) - Si\left(x_a, x_{c_i}\right)\right] = -0.56 < 0 \end{aligned}$$

or

$$\begin{aligned} f(x_a) &= \frac{1}{9}\sum_{i=1}^{9} h_i(x_a) \\ &= \frac{1}{9}\sum_{i=1}^{9} sgn\left[Di\left(x_a, x_{c_i}\right) - Di\left(x_a, x_{b_i}\right)\right] = -0.56 < 0. \end{aligned}$$

Using $f(x_a)$, the classifier $G(x_a)|D = \{G_1(x_a), G_2(x_a)\}$ is constructed as

$$G_1(x_a) = 1 - \left[\left(\frac{1}{n}\right)\sum_{i}^{n} \mu_{B_2}\left(y_{c_i}\right)\right] = 0.29,$$

$$G_2(x_a) = \left[\left(\frac{1}{n}\right)\sum_{i}^{n} \mu_{B_2}\left(y_{c_i}\right)\right] = 0.71.$$

So that, $G(x_a)|D = \{0.29, 0.71\}$ which addresses that x_a belongs to negative class with the error rate at most $\varepsilon + \delta = 0.05 + 0.05 = 0.1$.

Definition 3.2: Let $z_a \in X \times Y \subseteq D_t$, $U(x_a)|D_s = \{U_1(x_a), U_2(x_a)\}$ be the initial label value for instance z_a which is computed by fuzzy shift-unaware model U trained by instances from source domain D_s, $G(x_a)|D = \{G_1(x_a), G_2(x_a)\}$ is the label value for instance z_a which is computed by classifier G constructed by instances from a given domain D. The refined label value $R(x_a) = \{R_1(x_a), R_2(x_a)\}$ is defined as follows:

$$R(x_a) = \alpha U(x_a)|D_s + (1-\alpha)G(x_a)|D, \quad \alpha \le 1$$

where $\boldsymbol{\alpha}$ is a tradeoff coefficient.

Example 3.2: A simple example is introduced to clarify the Definition 3.2. We apply the values in Example 3.1 to illustrate the meaning of refinement. Given $U\|D_s$ is fuzzy shift-unaware model trained by instances from source domain D_s and $U(x_a)|D_s$ is initial label value for $x_a = (0.7, 0.3, 0.5)$ as follows:

$$U(x_a)|D_s = \{U_1(x_a), U_2(x_a)\} = \{0.5, 0.5\}$$

It is assumed that the shift-unaware model predicts equal membership value in each class. Hence, it is not possible to allocate x_a to any class based on shift-unaware model output. The classifier G is used to refine the label and specify the class that x_a belongs to. Given $\alpha = 0.3$, the value of the refined label would be as follows:

$$\begin{aligned} R_1(x_a) &= 0.3 \times 0.5 + 0.7 \times 0.81 = 0.717, \\ R_2(x_a) &= 0.3 \times 0.5 + 0.7 \times 0.18 = 0.276. \end{aligned}$$

So we have

$U(x_a)|D_s = \{0.717, 0.276\}$ which addresses that x_a belongs to positive class.

Similarly, it is assumed that the initial label value computed by $U(x_a)|D_s$ for $x_a = (0.65, 0.35, 0.35)$ is as follows:

$$U(x_a)|D_s = \{U_1(x_a), U_2(x_a)\} = \{0.5, 0.5\}$$

Again, the classifier G is used to refine the label and specify the class that x_a belongs to. Given $\alpha = 0.3$, the value of the refined label would be as follows:

$$\begin{aligned} R_1(x_a) &= 0.3 \times 0.5 + 0.7 \times 0.29 = 0.353, \\ R_2(x_a) &= 0.3 \times 0.5 + 0.7 \times 0.71 = 0.647. \end{aligned}$$

So we have

$U(x_a)|D_s = \{0.0.353, 0.647\}$ which addresses that x_a belongs to negative class.

The defined refinement (Definition 3.2) is applied in the proposed algorithm through a multistep path. The given instance comes from the target domain while the classifier is trained by positive and negative instances which come from a set of mixture domains composed of target and source domains in each step. The first mixture domain is composed of labeled instances in the source and target domains. Then through the steps, the number of source instances reduces and the number of target instances increases. The classifier computes the label value of the given target instance based on the positive and negative instances of mixture domains which transform from the source domain to the target domain and bridge the gap between the two domains.

Given $\tilde{A}^s = \{\widetilde{A_1}^s, \ldots, \widetilde{A_N}^s\}$ and $\tilde{A}^t = \{\widetilde{A_1}^t, \ldots, \widetilde{A_N}^t\}$ are the fuzzy sets for source domain $D_s = X^S \times Y$ and target domain $D_t = X^t \times Y$ respectively, where $\widetilde{A_k}$ is a fuzzy trapezoidal-shaped membership functions for each feature. Given $\tilde{B} = \{\tilde{B}_1, \tilde{B}_1\}$ is the fuzzy label set, which is the same for both domains. It is assumed that the number of these features for both target and source domains is the same, but the membership functions of these fuzzy sets are different. This assumption implies a domain adaptation problem in which the feature space is the same, but the distributions are different. Given $D_t = D_{t_0} \cup D_{t_1}$ is the target domain which includes labeled instances D_{t_0} and unlabeled instances D_{t_1}. The fuzzy refinement domain adaptation (FRDA) algorithm is described as follows:

<Fuzzy Refinement Domain Adaptation algorithm>

Input: *Source domain*: D_s

Target domain: $D_t = D_{t_0} \cup D_{t_1}$
Fuzzy shift-unaware prediction model trained by source domain: $U|D_s$
Similarity/dissimilarity based classifier constructed by given domain: $G|D$
Coefficient parameter: α
Number of steps: m

[Begin]

Step 1: Fuzzify the crisp-value of instances from both domains using the Singleton fuzzifier as follow:

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

where $\tilde{x}_i$ is the fuzzified equivalent of crisp input x_i.

Step 2: Perform antecedent matching of fuzzyfied inputs $x_i \in D_s$ and D_t against fuzzy features $\tilde{A}^s$ and $\tilde{A}^t$ respectively, the input membership value in each feature is computed as follows:

$$\mu_{\widetilde{A_k}}(\tilde{x}_i) = \begin{cases} 0, & if \quad x_i \le l_{\widetilde{A_k}} \\ \frac{x_i - l_{\widetilde{A_k}}}{u_{\widetilde{A_k}} - l_{\widetilde{A_k}}}, & if \quad l_{\widetilde{A_k}} \le x_i \le u_{\widetilde{A_k}} \\ 1, & if \quad u_{\widetilde{A_k}} \le x_i \le v_{\widetilde{A_k}} \\ \frac{r_{\widetilde{A_k}} - x_i}{r_{\widetilde{A_k}} - u_{\widetilde{A_k}}}, & if \quad v_{\widetilde{A_k}} \le x_i \le r_{\widetilde{A_k}} \\ 0, & if \quad x_i \ge r_{\widetilde{A_k}}, \end{cases} \quad j \in \{1, 2, \ldots, N\}.$$

Step 3: Train shift-unaware fuzzy prediction model $U|D_s$ by the instances of source domain.

Step 4: Calculate the fuzzy initial label matrix for unlabeled target domain instances $MU_{|D_{t_1}| \times 2}$ by the prediction model which is trained by source domain data in step 3.

```
For $i = 1$ to $|D_{t_1}|$
    For $j = 1$ to 2
        $MU(i,j) = U_j(x_{a_i})|D_s$
    Next j
Next $i$
    Step 5: Compute the fuzzy refined label matrix for
    unlabeled target domain instances $MR_{|D_{t_1}|\times 2}$ using
    the label matrix $MG_{|D_{t_1}|\times 2}$ which is calculated by
    similarity/dissimilarity classifier constructed by mixture
    domain $D_w$.
For $w = 1$ to m
    $D_w = (1 - (w/m))|D_s| + (w/m)|D_{t_1}| + |D_{t_0}|$
    Construct $G|D_w$ by the positive and negative instances
    from domain $D_w$
    For $i = 1$ to $|D_{t_1}|$
        For $j = 1$ to 2
            $MG(i,j) = G_j(x_{a_i})|D_w$
            $MR(i,j) = \boldsymbol{\alpha} MU(i,j) + (\mathbf{1} - \boldsymbol{\alpha})MG(i,j)$
        Next $j$
    Next $i$
Next $w$
[End]
Output: Refined fuzzy label matrix of unlabeled target
instances $MR_{|D_{t_1}|\times 2}$
```

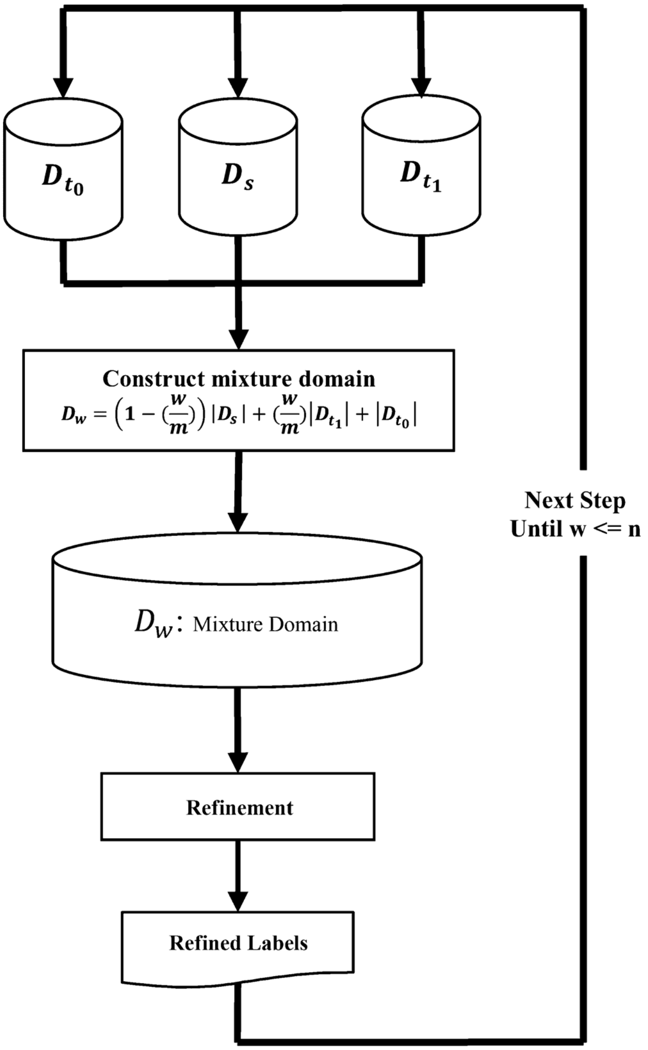

Fig. 1. Fuzzy refinement domain adaptation (FRDA) algorithm.

As can be seen in Step 5, the refinement is based on the fact that the label of similar or dissimilar instances to the target instance is used to modify the initial label of the target instance. The refinement process moves from the source domain (D_s) toward the target domains $(D_{t_1}\ or\ D_{t_0})$ through m steps using the parameter (w/m), which indicates the percentage of instances of the source domain and target domain in the mixture domain in each step of refinement. As w increases, (w/m) becomes greater, and consequently, the contribution of source domain data in the mixture domain becomes less; conversely, the portion of target domain data increases. The resultant neighboring mixture domains are similar to one another and smoothly transfer from the source domain toward the target domain. Through the multistep process, a bridge is created and label structure between the source and target domains is transferred more accurately and easily. As a result of this algorithm, a fuzzy label matrix for all unlabeled instances of the target domain is achieved. Each row of this matrix indicates the membership values of one instance to all classes. To find the final label for each instance, the following equation can be used as a defuzzifier.

Label $(x_{a_i}) = \arg max_j\{\widetilde{MR}_{i,j}|j = 1,2\}$ Fig. 1 demonstrates the proposed Fuzzy Refinement Domain Adaptation (FRDA) algorithm.

IV. Experiments

In this section, we validate the proposed algorithm using real world financial data in different settings. The task in this experiment is bank failure prediction and the prediction label has two classes: Failed and Survived. We perform the experiments to examine the algorithm's performance in transferring knowledge from different time periods using different settings and features.

A. Data Set

The data set and financial variables are extracted from Call Report Data, which is downloaded from the website of the Federal Reserve Bank of Chicago, and the status of each bank is identified according to the Federal Financial Institutions Examination Council. The dataset, shown in Table III, includes the observation period of the surviving banks of 21 years from Jun 1980 to Dec 2000. There are 548 failed banks and 2555 banks that survived, as presented by [36], [37]. Although, Tung *et al.* [36] used nine financial features, according to their statistical significance and correlation, it is observed that the model with three features has fewer created rules, less computational load and greater prediction accuracy. Each feature is ranked based on the importance of a feature as a result of a feature selection process, and the three features with the highest grade are selected [37]. The definition of all features and their expected impacts on bank failure are described in Table IV. The FRDA

TABLE III
NUMBER OF AVAILABLE RECORDS IN SECOND DATA SET FOR EACH SCENARIO

Year	Banks Total	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 IV
DEFINITION OF VARIABLES AND THEIR IMPACTS ON BANK FAILURE (THREE SELECTED VARIABLES ARE MARKED WITH *)

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 of 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

algorithm is run by nine and three features separately and the results are then compared.

B. Research Design and Preprocessing

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 respectively from 1990. We perform the refinement algorithms on the labels predicted by shift-unaware prediction models, from which we receive the unrefined (initial) labels of target instances. To ensure that the FRDA algorithm is sufficiently robust, we select three different shift-unaware prediction models: fuzzy neural network (FNN) [38]; support vector machine (SVM) [30]; multilayer perception neural network (NN) [39].

Additionally, we compare the proposed algorithm with two other refinement algorithms: refinement domain adaptation (RDA); bridge refinement (BR) [30]. The RDA, which is the nonfuzzy version of the FRDA, applies the features with the crisp-value, the classifier defined by Definition 2.5 and the Euclidian distance as similarity/dissimilarity measure. Comparing FRDA with RDA demonstrates the impact of the fuzzy approach we apply in FRDA. We use the BR, which is the closest study to our research, with the same implementation of bridge refining in [30].

To establish the influence of the labeled target instances on the performance of FRDA, it is examined and analyzed using two different settings: Setting 1: Call FRDA algorithm where $D_{t_0} = \emptyset$; Setting 2: Call FRDA algorithm where $D_{t_0} \neq \emptyset$. The evaluation is also performed using two categories of feature sets: nine variables and; three variables. In conclusion, we perform 36 different experiments. Since there is not a meaningful difference between the results of the experiments with nine and three features, we compute the average accuracy of these two categories.

To reduce the influence of the imbalanced data-sets problem, the Synthetic Minority Over-sampling Technique (SMOTE) [40] is applied to the training data sets. The number of failed banks increases 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 splits into two pools: 1) failed banks denoted with output 1; and 2) survived banks denoted with output-1. There are five groups, which include instances of both pools randomly to form the training set. The training sets of the five groups are mutually exclusive. The shift-unaware prediction 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 cross validation groups' accuracy, is calculated using the Geometric Mean of sensitivity and specificity. This metric is applied because both are expected to be high simultaneously.

Discrete incremental clustering (DIC), which is a novel self organizing clustering technique, is applied to create the fuzzy features. DIC is a dynamic clustering technique which avoids drawbacks such as *stability-plasticity* and *inflexibility* found in other methods and computing trapezoidal-shaped fuzzy sets [36]. We use the fuzzy similarity/dissimilarity measure addressed by [41] in the FRDA algorithm. To construct the basic classifier $f(x)$, the approach introduced by [34] is applied.

C. Experiment Results Analysis

In this section, the results gained from the experiment are reported in Tables V–VII. To ensure that the FRDA makes a significant improvement in accuracy, we perform our algorithm on unrefined labels predicted by three different shift-unaware prediction models: FNN; SVM; NN. In all scenarios, the proposed algorithm increases the accuracy. The maximum relative growth is gained in refining the initial labels predicted by SVM and FNN for year 2000, being 26.14% and 22.98%, respectively. Furthermore, in all scenarios, FRDA achieves better accuracy on Setting 2 than on Setting 1. There is on average 2.98% more improvement in accuracy when we use labeled target instances in our algorithm. Hence, it is concluded that the labeled instances in the target domain have a constructive influence on the predictive accuracy of the algorithm. Additionally, the proposed

TABLE V
ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON FNN

Year	FNN	FRDA		RDA		BR	
		Setting1	Setting 2	Setting1	Setting2	Setting1	Setting2
1995	0.8236	0.9184	0.9476	0.9163	0.9303	0.9115	0.9153
		11.51%	**15.06%**	11.26%	12.96%	10.67%	11.13%
1998	0.7615	0.8920	0.9112	0.8911	0.9037	0.8722	0.8911
		17.14%	**19.66%**	17.02%	18.67%	14.54%	17.02%
2000	0.7329	0.8647	0.9013	0.8600	0.8908	0.8507	0.8821
		17.98%	**22.98%**	17.34%	21.54%	16.07%	20.36%

TABLE VI
ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON SVM

Year	SVM	FRDA		RDA		BR	
		Setting1	Setting 2	Setting 1	Setting 2	Setting 1	Setting 2
1995	0.8063	0.9154	0.9431	0.9082	0.9283	0.9011	0.9132
		13.53%	**16.97%**	12.64%	15.13%	11.76%	13.26%
1998	0.7501	0.8951	0.9055	0.8923	0.8983	0.872	0.8881
		19.33%	**20.72%**	18.96%	19.76%	16.25%	18.40%
2000	0.7093	0.8685	0.8947	0.8598	0.8702	0.8414	0.8545
		22.44%	**26.14%**	21.22%	22.68%	18.62%	20.47%

TABLE VII
ACCURACY AND RELATIVE INCREASE IN ACCURACY ACHIEVED BY REFINEMENT ALGORITHMS ON NN

Year	NN	FRDA		RDA		BR	
		Setting1	Setting 2	Setting 1	Setting 2	Setting 1	Setting 2
1995	0.8117	0.9129	0.9288	0.8994	0.9194	0.8804	0.9035
		12.47%	**14.43%**	10.80%	13.27%	8.46%	11.31%
1998	0.7664	0.8801	0.9022	0.8731	0.8947	0.8625	0.8733
		14.84%	**17.72%**	13.92%	16.74%	12.54%	13.95%
2000	0.7251	0.8695	0.8812	0.8631	0.8734	0.8488	0.8654
		19.91%	**21.53%**	19.03%	20.45%	17.06%	19.35%

TABLE VIII
HOLM TEST TO COMPARE THE REFINEMENT ALGORITHM PERFORMANCE

	Comparison	$z = (R_0 - R_i)/SE$	P	α-Holm	Hypothesis
1	FRDA vs. BR	6.197	5.762E-8	0.017	Rejected
2	FRDA vs. RDA	4.647	3.358E-4	0.025	Rejected
3	RDA vs. BR	3.098	0.002	0.05	Rejected

algorithm demonstrates more augmentation when the gap between the source domain and the target domain becomes larger. For instance, if SVM is applied for initial prediction, FRDA enhances the predictive accuracy for 1995, five years following 1990, by 13.53%, and it improves the accuracy by 22.44% in year 2000, 10 years following 1990.

The performance of the proposed FRDA algorithm is compared with that of RDA and BR. As can be seen from Tables V–VII, FRDA outperforms other refinement algorithms in all scenarios. To confirm the growth in accuracy, the Holm test [42], which is a nonparametric statistical test, is applied to specify whether the improvement is significant or not. The test is performed on 12 scenarios in which a particular shift-unaware prediction model is applied. As a result, we perform the Holm test three times to benchmark the performance of the refinement algorithms when FNN, SVM and NN are employed respectively. The results of the statistical tests, which are presented in Table VIII, reject the hypothesis of equality of the accuracy in 0.05 level of confidence. This demonstrates that the FRDA outperforms RDA and BR significantly and shows that RDA has remarkably better performance than BR. From Test 2, it is concluded the fuzzy approach we apply in our algorithm significantly enhances the accuracy. Moreover, Tests 1and 3 imply that the proposed algorithm significantly outperforms the BR, regardless of whether the feature values are fuzzy or not, and whether the shift-unaware prediction model is FNN, SVM or NN.

The proposed algorithm includes several parameters which need to be set which is an important implementation issue. The influence of parameters on the performance of algorithm is investigated empirically. To do this, the average accuracy of the algorithm on one data set is examined using different values of these parameters on two settings and two prediction models: NN; SVM. According to the results of parameter settings, the optimum values for parameters are: $\varepsilon = 0.05$, $\delta = 0.05$, $\gamma = 0.09$, $B = 0.1$, $\pi = 0.001$, $\alpha = 0.3$, and $m = 25$. Since, the proposed algorithm applies local learning for domain adaptation; it gains a low computational complexity. In the step 4 of the algorithm the complexity is of $O(|D_{t_1}|)$: the number of unlabeled instances in target domain. The complexity of Step 5 is of $O(|D_{t_1}| \times m \times n)$ where $n = 72$ and $m = 25$ are the parameters of algorithm. n is the number of samples to construct the classifier and m is the number of steps of refinement. Hence, overall computational complexity of the proposed algorithm depends on the number of unlabeled instances in target domain which is highly low in comparison with existing domain adaptation methods.

V. Conclusion

In this paper, the FRDA algorithm is proposed to solve the domain adaptation problem based on the ecosystem-oriented approach. The algorithm concentrates on the modification of the instances' labels in the target domain using a classifier that is trained based on similar/dissimilar instances in the sets of mixture domains. It uses fuzzy concepts to handle the vagueness of feature values and improve predictive accuracy. Three different shift-unaware prediction models including FNN, SVM, and NN are applied to report the initial labels for unlabeled target instances. According to the empirical results, the proposed algorithm achieves a remarkable improvement in performance. It demonstrates a significant increase in predictive accuracy, particularly when the algorithm utilizes a few labeled target data along with the source data for training (Setting 2). The results show that the FRDA achieves even better enhancement when it is applied to a greater time period prediction. In contrast to other domain adaptation methods, which apply crisp-value, the FRDA algorithm applies a fuzzy concept to modify the predicted labels and consequently achieves better results. FDR is relatively independent from the predictive function and can be applied with other shift-unaware prediction methods. According to the empirical results, it can successfully transfer knowledge over a long time period to predict failure in banking ecosystem 10 years ahead. The algorithm can be considered as an applicable prediction model which does not need to be retrained in every determined period. Additionally, it can be applied in instances in which insufficient recent training data is available in the banking ecosystem. A comparison of the proposed algorithm using other prediction models, such as fuzzy case base reasoning, can be conducted in future research. Developing an algorithm based on the proposed algorithm which can extract the relevant features to reduce the difference between domains is also desirable for future work. Finally, comparing the performance of the proposed algorithm with that of other statistical domain adaptation methods and types of data sets would also provide an interesting topic for future study.

References

[1] D. P. Filev, R. B. Chinnam, F. Tseng, and P. Baruah, "An industrial strength novelty detection framework for autonomous equipment monitoring and diagnostics," *IEEE Trans. Ind. Inf.*, vol. 6, no. 4, pp. 767–779, Nov. 2010.

[2] D. Hunter, H. Yu, I. M. S. Pukish, J. Kolbusz, and B. M. Wilamowski, "Selection of proper neural network sizes and architectures: A comparative study," *IEEE Trans. Ind. Inf.*, vol. 8, no. 2, pp. 228–240, May 2012.

[3] X. Zhu, Semi-Supervised Learning Literature Survey Computer Science, University of Wisconsin-Madiso. Madison, WI, USA.

[4] K. Nigam, A. K. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Mach. Learn.*, vol. 39, pp. 103–134, 2000.

[5] T. Joachims, "Transductive inference for text classification using support vector machines," in *Proc. 16th Int. Conf. Mach. Learn.*, Bled, Slovenia, 1999, pp. 200–209.

[6] S. Maiti, V. Verma, C. Chakraborty, and Y. Hori, "An adaptive speed sensorless induction motor drive with artificial neural network for stability enhancement," *IEEE Trans. Ind. Inf.*, vol. 8, no. 4, pp. 757–766, Nov. 2012.

[7] G. P. C. Fung, J. X. Yu, L. Hongjun, and P. S. Yu, "Text classification without negative examples revisit," *IEEE Trans. Knowledge Data Eng.*, vol. 18, no. 1, pp. 6–20, Jan. 2006.

[8] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge Data En.*, vol. 22, no. 10, pp. 1345–1359, Oct. 2010.

[9] R. Luis, L. Sucar, and E. Morales, "Inductive transfer for learning Bayesian networks," *Mach. Learn.*, vol. 79, pp. 227–255, 2010.

[10] A. Arnold, R. Nallapati, and W. W. Cohen, "A comparative study of methods for transductive transfer learning," in *Proc. 7th IEEE Int. Conf. Data Mining Workshops*, Omaha, NE, USA, 2007, pp. 77–82.

[11] E. W. Xiang, C. Bin, D. H. Hu, and Y. Qiang, "Bridging domains using world wide knowledge for transfer learning," *IEEE Trans. Knowledge Data Eng.*, vol. 22, no. 6, pp. 770–783, Jun. 2010.

[12] G. Bo, Y. Linjun, X. Chao, and H. Xian-Sheng, "Ranking model adaptation for domain-specific search," *IEEE Trans. Knowledge Data Eng.*, vol. 24, no. 4, pp. 745–758, Apr. 2012.

[13] D. Lixin, X. Dong, and I. W. Tsang, "Domain adaptation from multiple sources: A domain-dependent regularization approach," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 23, no. 3, pp. 504–518, Mar. 2012.

[14] D. Lixin, I. W. Tsang, and X. Dong, "Domain transfer multiple kernel learning," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. 34, no. 3, pp. 465–479, Mar. 2012.

[15] B. Zadrozny, "Learning and evaluating classifiers under sample selection bias," in *Proc. 21st Int. Conf. Mach. Learn.*, Bnaff, Canada, 2004, pp. 114–121.

[16] H. Shimodaira, "Improving predictive inference under covariate shift by weighting the log-likelihood function," *J. Statist. Plann. Inference*, vol. 90, pp. 227–244, 2000.

[17] G.-R. Xue, W. Dai, Q. Yang, and Y. Yu, "Topic-bridged PLSA for cross-domain text classification," in *Proc. 31st Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, Singapore, 2008, pp. 627–634.

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

[19] J. Blitzer, R. McOnald, and F. Pereira, "Domain adaptation with structural correspondence learning," in *Proc. Con. Empirical Methods Natural Language Process.*, Sydney, Australia, 2006, pp. 120–128.

[20] J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Wortman, "Learning bounds for domain adaptation," in *Proc. 21st Annu. Conf. Neural Inf. Process. Syst.*, Cambridge, MA, USA, 2007, pp. 245–252.

[21] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification," in *Proc. 45th Annu. Meeting Assoc. Comput. Ling.*, Prague, Czech Republic, 2007, pp. 440–447.

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

[23] Z. Geng, L. Han-Xiong, and G. Min, "Design a wind speed prediction model using probabilistic fuzzy system," *IEEE Trans. Ind. Inf.*, vol. 8, no. 4, pp. 819–827, Nov. 2012.

[24] L. Weitao, W. Dianhui, and C. Tianyou, "Flame image-based burning state recognition for sintering process of rotary kiln using heterogeneous features and fuzzy integral," *IEEE Trans. Ind. Inf.*, vol. 8, no. 4, pp. 780–790, Apr. 2012.

[25] F. Penedo, R. E. Haber, A. Gajate, and R. M. del Toro, "Hybrid incremental modeling based on least squares and fuzzy K-NN for monitoring tool wear in turning processes," *IEEE Trans. Ind. Inf.*, vol. 8, no. 4, pp. 811–818, Nov. 2012.

[26] T. Du-Ming, I. Y. Chiang, and T. Ya-Hui, "A shift-tolerant dissimilarity measure for surface defect detection," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 128–137, Feb. 2012.

[27] T. Cuong and M. M. Trivedi, "3-D posture and gesture recognition for interactivity in smart spaces," *IEEE Trans. Ind. Inf.*, vol. 8, no. 1, pp. 178–187, Feb. 2012.

[28] L. Wang, C. Yang, and J. Feng, "On learning with dissimilarity functions," in *Proc. Int. Conf. Mach. Learn.*, Corvallis, OR, 2007, pp. 991–998.

[29] M. F. Balcan, A. Blum, and N. Srebro, "A theory of learning with similarity functions," *Mach. Learn.*, vol. 72, pp. 89–112, 2008.

[30] D. Xing, W. Dai, G.-R. Xue, and Y. Yu, "Bridged refinement for transfer learning," in *Proc. 11th Eur. Conf. Principles Practice Knowledge Discovery Databases*, Warsaw, Poland, 2007, pp. 324–335.

[31] P. Ferronato, "Architecture for Digital Ecosystems, beyond Service Oriented Architecture," in *Proc. IEEE Int. Conf. Digital Ecosyst. Technol.*, 2007, pp. 660–665.

[32] P. Ferronato, "Ecosystem Oriented Architecture (EOA) vs. SOA," in *Digital Business Ecosystems*. Luxembourg: European Commission, 2007, pp. 111–118.

[33] Q. Yang, "Transfer learning beyond text classification," in *Proc. 1st Asian Conf. Mach. Learn.: Adv. Mach. Learn.*, Nanjing, China, 2009, pp. 10–22.

[34] L. Wang, M. Sugiyama, C. Yang, K. Hatano, and J. Feng, "Theory and algorithm for learning with dissimilarity functions," *Neural Comput.*, vol. 21, pp. 1459–1484, 2008.

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

[36] W. L. Tung, C. Quek, and P. Cheng, "GenSo-EWS: A novel neural-fuzzy based early warning system for predicting bank failures," *Neural Netw.*, vol. 17, pp. 567–587, 2004.

[37] G. S. Ng, C. Quek, and H. Jiang, "FCMAC-EWS: A bank failure early warning system based on a novel localized pattern learning and semantically associative fuzzy neural network," *Expert Syst. With Appl.*, vol. 34, pp. 989–1003, 2008.

[38] V. Behbood, J. Lu, and G. Zhang, "Adaptive Inference-based learning and rule generation algorithms in fuzzy neural network for failure prediction," in *Proc. IEEE Int. Conf. Intell. Syst. Knowledge Eng. (ISKE)*, Hangzhou, China, 2010, pp. 33–38.

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

[40] N. Chawla, K. Bowyer, L. Hall, and P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[41] W. Wen-June, "New similarity measures on fuzzy sets and on elements," *Fuzzy Sets Syst.*, vol. 85, pp. 305–309, 1997.

[42] S. Holm, "A simple sequentially rejective multiple test procedure," *Scandinavian J. Statist.*, vol. 6, pp. 65–70, 1979.

Vahid Behbood (S'11) received the Ph.D. degree in software engineering form the University of Technology Sydney, Sydney, Australia.

He is currently working as Postdoc Research Fellow in Decision Systems and e-Service Intelligence Research Laboratory of the Centre for Quantum Computation and Intelligent Systems, School of Software, Faculty of Engineering and Information Technology, at the University of Technology, Sydney (UTS), Australia. His research interests include machine learning (particularly transfer learning), fuzzy sets and systems, warning systems, and data mining. He has published over 20 papers in journals and conferences.

Jie Lu (M'06) received the Ph.D. degree from Curtin University, Perth, Australia, in 2000.

She is currently Head of the School of Software at the University of Technology Sydney, Sydney, Australia. Her research interests include the area of decision support systems and e-Service intelligence. She has published five research books and 270 papers.

Dr. Li won five Australian Research Council discovery grants and 10 other grants. She received a University Research Excellent Medal in 2010. She serves as Editor-In-Chief for *Knowledge-Based Systems* (Elsevier), Editor-In-Chief for *International Journal of Computational Intelligence Systems* (Atlantis), Editor for *Intelligent Information Systems* (World Scientific), and delivered six keynote speeches at international conferences.

Guangquan Zhang received the Ph.D. degree in applied mathematics from Curtin University of Technology, Perth, Australia.

He is currently an Associate Professor in the Faculty of Engineering and Information Technology at the University of Technology Sydney (UTS), Sydney, Australia. From 1979 to 1997, he was a Lecturer, Associate Professor and Professor in the Department of Mathematics, Hebei University, China. His main research interests include the area of multiobjective, bilevel, and group decision making, decision support system tools, fuzzy measure, fuzzy optimization, and uncertain information processing. He has published four monographs, four reference books, and over 250 papers including more than 140 refereed journal articles.

Dr. Zhang has won four 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.