

“© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# A Novel Fuzzy Neural Network for Unsupervised Domain Adaptation in Heterogeneous Scenarios

Feng Liu, Guangquan Zhang, Jie Lu

Decision Systems and e-Service Intelligence (DeSI) Lab,  
Centre for Artificial Intelligence, Faculty of Engineering and Information Technology  
University of Technology Sydney  
Sydney, Australia

Feng.Liu-2@student.uts.edu.au; {Guangquan.Zhang, Jie.Lu}@uts.edu.au

**Abstract** — How to leverage knowledge from labelled domain (source) to help classify unlabeled domain (target) is a key problem in the machine learning field. Unsupervised domain adaptation (UDA) provides a solution to this problem and has been well developed for two homogeneous domains. However, when the target domain is unlabeled and heterogeneous with the source domain, current UDA models cannot accurately transfer knowledge from a source domain to a target domain. Benefiting from development of neural networks, this paper presents a new neural network, shared fuzzy equivalence relations neural network (SFER-NN), to address the heterogeneous UDA (HeUDA) problem. SFER-NN transfers knowledge across two domains according to shared fuzzy equivalence relations that can simultaneously cluster features of two domains into several categories. Based on the clustered categories, SFER-NN is constructed to minimize the discrepancy between two domains. Compared to previous works, SFER-NN is more capable of minimizing discrepancy between two domains. As a result of this advantage, SFER-NN delivers a better performance than previous studies using two public datasets.

**Keywords**—transfer learning, domain adaptation, fuzzy relation, machine learning

## I. INTRODUCTION

The performance of a classification model trained on data from a specific domain typically degrades when applied to a related but different one. Although labelling many instances from the new domain would address this issue, it is often too expensive or impractical. Domain adaptation and transfer learning have therefore emerged as solutions to this problem. Domain adaptation model leverages labelled data from a source domain, in which it is abundant, to train a classification model to operate in a target domain, in which it is either sparse or even lacking altogether [1]–[3]. For example, the model trained to recognize trucks can be used to recognize cars; credit records collected from Germany can be applied to improve the assessment accuracy of Australia’s credit records which suffer from data shortage; or bank failure data from the United States can be used to help predict Australian banks’ failures. These examples clearly show that classification accuracy on a target domain will be improved if we use domain adaptation models to transfer knowledge from a source domain to the target domain [4]–[6]. The amount of data in the current era is growing fast, which means that data labelling comes at a high cost or is an impossible mission. How to using the existed knowledge (labeled data/source

domain) to explore unknown domain (unlabeled but related data/target domain) has thus become extremely important.

To date, five types of domain adaptation models have been well studied: homogeneous supervised/semi-supervised/unsupervised domain adaptation [7]–[9] and heterogeneous supervised/semi-supervised domain adaptation [10]–[12]. In each of these, “homogeneous” represents source domains and target domains have the same number of related features, and “heterogeneous” represents source domains and target domains have a different number of related features.

Supervised and semi-supervised domain adaptation models need labeled instances in the target domain, which is an obstacle to applying these models to successfully accomplish a classification task in an unlabeled target domain. To address unsupervised domain adaptation problem, existing models are limited to source domains that have the homogeneous feature space as the target domain, which does not apply to some scenarios in the current environment. Therefore, it is necessary to consider another type of domain adaptation: heterogeneous unsupervised domain adaptation (HeUDA). This is a common scenario in the current era because it is difficult to find a homogeneous source domain for a specific target domain. Recently, domain adaptation has been investigated in the context of deep learning with promising results [13]–[15]. The simplest approach involves fine-tuning a convolutional neural network (CNN) pre-trained on the source domain using instances in the target domain. However, current deep domain adaptation mainly focus on the five types mentioned above rather than the heterogeneous unsupervised domain adaptation.

In our previous work [16], we proposed a shared fuzzy equivalence relations (SFER) model to address the heterogeneous unsupervised domain adaptation problem. But the SFER model still is a shadow domain adaptation model, which means that it has limited ability to minimize the discrepancy between source and target domains compared to deep domain adaptation models. To overcome the shortcoming of SFER model and propose a neural-network-based HeUDA model, this paper presents a novel model called SFER neural network (SFER-NN), which comprises two parts: 1) SFER model to construct the first two layers of the neural network; and 2) the maximum mean discrepancy (MMD) regularizer to minimize discrepancy between two domains in a hidden layer. After training SFER-NN using labelled instances from the source domain and unlabeled

instances from the target domain, knowledge can be transferred from the source domain to the target domain. The main contributions of this paper are as follows:

- 1) A novel neural network, SFER-NN, is designed to address the heterogeneous unsupervised domain adaptation problem;
- 2) SFER-NN is more capable of reducing the discrepancy between source and target domains than current heterogeneous unsupervised domain adaptation models.

This paper is organized as follows. Section II reviews papers in the unsupervised domain adaptation field. Section III introduces the SFER-NN model. Section IV demonstrates the classification results of the SFER-NN model and benchmarks. Section V concludes the paper and discusses future work.

## II. RELATED WORK

We briefly review unsupervised domain adaptation (UDA) models in this section. When there is no labeled instance in the target domain, the knowledge transfer process is regarded as UDA. There is one basic assumption: two domains are related and have similar prediction tasks. Since it is difficult to transfer knowledge from heterogeneous source domains to an unlabeled target domain, the principal UDA technologies deal with the issue as if the source domain and target domain are homogeneous. This kind of technology is known as homogeneous unsupervised domain adaptation (HoUDA). The representative HoUDA models include transfer component analysis (TCA) [7], geodesic flow kernel (GFK), information-theoretical learning (ITL) [17], transfer deep network [18], optimal transport [19], scatter component analysis [20], conditional transferable components [21] and correlation alignment [22]. Recently, the generative adversarial network is applied to develop HoUDA models and has satisfied performance in the computer vision field [23], [24].

Unsupervised domain adaptation models based on homogeneous feature spaces have been widely researched. However, HeUDA models are rarely discussed due to two shortcomings of current domain adaptation models: the feature spaces must be homogeneous, and there must be at least some labeled instances in the target domain (or there must be a parallel set in both domains). Hybrid heterogeneous transfer learning model [25] uses the information of the parallel set of both domains to transfer knowledge across domains. Domain specific feature transfer [26] is designed to address HeUDA problem when two domains have common features. Kernel canonical correlation analysis (KCCA) [27] was proposed to address HeUDA problems when there are paired instances in source and target domains, but this model is not valid when there are no paired instance in both domains. In our previous works, we successfully develop several models to transfer knowledge from a heterogeneous source domain to an unlabeled target domain without requirements of common features or paired instances [16], [28], [29]. However, these models are shadow domain adaptation model, which means that they have limited ability to minimize the discrepancy between two domains.

## III. HETEROGENEOUS UNSUPERVISED DOMAIN ADAPTATION VIA FUZZY NEURAL NETWORK

This section presents the proposed neural network, SFER-NN, based on shared fuzzy equivalence relations. The notations used in this section will be first introduced, and the second subsection details the motivation of our paper. The shared fuzzy equivalence relations will be briefly reviewed in subsection C. The final subsection presents the SFER-NN to address the situation in which the target domain is unlabeled and has a heterogeneous feature space with the source domain.

### A. Notations and problem statement

We assume that there is one source domain  $D_S$  and one target domain  $D_T$  with different feature spaces, in which  $D_S = \{X_{si}, y_{si}\}$  and  $D_T = \{X_{ti}\}$ , where  $X_{si} \in \mathbb{R}^{N_s}$ ,  $X_{ti} \in \mathbb{R}^{N_t}$ ,  $y_{si} \in \{1, 2\}$ ,  $N_s \neq N_t$  and  $i = 1, 2, \dots, L$  ( $L$  is the number of instances in each domain). We denote  $S_s = \{X_{si}\}$  and  $S_t = \{X_{ti}\}$ ,  $i = 1, 2, \dots, L$ . We aim to label each instance in  $D_T$  using the knowledge from  $D_S$ . It is important to clarify that there are no labeled instances in the target domain.

### B. Motivation

Because the source domain and the target domain are heterogeneous, the model trained by the source domain cannot be directly applied to label instances of the target domain. A common idea is to map two domains onto a latent feature space so that the mapped domains have the same dimensionality. In our previous work [16], SFER model is proposed to cluster features of each domain so that features of two domains can be clustered as several categories and the numbers of the categories of two domains are the same. Fig. 1 gives an example about how a SFER model to cluster features of two heterogeneous domains into several categories. After obtaining these categories, SFER model applies a z-score function to minimize the discrepancy between two domains. Although the z-score function is easy to implement and has a low computational cost, its ability to minimize the discrepancy between two domains is very limited. Because of the recent development of deep domain adaptation models [14], [15], [30], we are motivated to apply a neural network to overcome the drawback of SFER model.

### C. Shared fuzzy equivalence relations

In this subsection, we briefly introduce the SFER model. Due to the length limitation, we mainly focus on input and output of SFER model, which facilitates readers to understand SFER-NN. We use  $F_s = \{f_{s1}, \dots, f_{sN_s}\}$  to denote the set of features of the source domain and  $F_t = \{f_{t1}, \dots, f_{tN_t}\}$  to denote the set of features of the target domain. Then, SFER model will map  $F_s$  and  $F_t$  to be  $F_s^{SFER}$  and  $F_t^{SFER}$ :

$$[F_s^{SFER}, F_t^{SFER}] = SFER(F_s, F_t),$$

where  $F_s^{SFER}$  and  $F_t^{SFER}$  only contains  $N_l$  element. The value of  $N_l$  will be automatically confirmed by SFER model. In Fig.

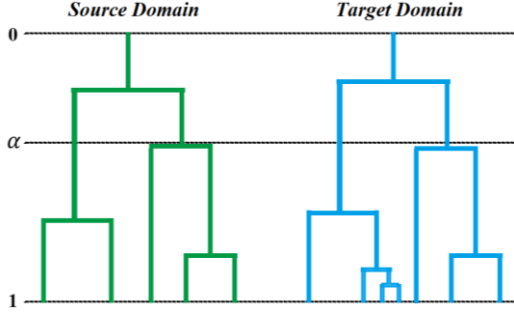


Fig. 1. Shared fuzzy equivalence relations (SFER) to cluster features of two heterogeneous domains. In this figure, there are 5 features in the source domain and 7 features in the target domain. SFER model will find the best alpha to cluster two domains' features into 2 categories. Using SFER, we actually map two heterogeneous feature spaces onto a latent feature space where its dimensionality equals to 2.

1,  $N_s$  is 5,  $N_t$  is 7 and  $N_l$  is 2, which means that SFER model map two heterogeneous domains onto a 2-dimension latent feature space.

From the view of neural network, SFER model is a non-fully-connected two-layer neural network. Each input-layer neuron (e.g.,  $f_{s1}$  or  $f_{t1}$ ) only connects with one neuron (e.g.,  $f_{s1}^{SFER}$  or  $f_{t1}^{SFER}$ ) in the output layer, and the weights between two layers are 1. From this view, we can easily construct SFER-NN in a following subsection.

#### D. Shared fuzzy equivalence relations neural network

This subsection first presents the structure of SFER-NN and then shows how to train a SFER-NN.

##### 1) Structure of SFER-NN

SFER-NN is a five-layer neural network with two branches (one branch from the source domain and another one from the target domain) and its structure is illustrated in Fig. 2. We will introduce the structure of SFER-NN layer by layer.

**Layer 1 (Input layer):** the input layer receives two instances:  $X_{si}$  and  $X_{ti}$ .  $X_{si}$  is a  $N_s$ -by-1 vector from the source-domain branch and  $X_{ti}$  is a  $N_t$ -by-1 vector from the target-domain branch.

**Layer 2 (SFER layer):** the SFER layer has  $2 \times N_l$  neurons,  $N_l$  neurons for each branch. The values of neurons in the source-domain branch forms a  $N_l$ -by-1 vector and is denoted by  $a_2^{si}$  (corresponding to the  $i^{th}$  instance in the source domain). Similarly, we have another  $N_l$ -by-1 vector  $a_2^{ti}$  for the target-domain branch (corresponding to the  $i^{th}$  instance in the target domain). They are calculated by following equations,

$$a_2^{si} = W_1^s X_{si}, a_2^{ti} = W_1^t X_{ti}, \quad (1)$$

where  $W_1^s$  is a  $N_l$ -by- $N_s$  matrix and  $W_1^t$  is a  $N_l$ -by- $N_t$  matrix. Each element in  $W_1^s$  and  $W_1^t$  can only be set to 0 or 1 and  $W_1^s$  and  $W_1^t$  have following constrains.

$$\begin{aligned} \sum_{i=1}^{N_l} (W_1^s)_{ij} &= 1, \sum_{i=1}^{N_s} \sum_{j=1}^{N_l} (W_1^s)_{ij} = N_s, \\ \sum_{i=1}^{N_l} (W_1^t)_{ij} &= 1, \sum_{i=1}^{N_t} \sum_{j=1}^{N_l} (W_1^t)_{ij} = N_t. \end{aligned}$$

This means that the input layer and the SFER layer are not fully connected in each branch, and each input neuron only connect with one neuron in the SFER layer. SFER code is available in [https://github.com/alexwolfluiuf/SFER\\_code](https://github.com/alexwolfluiuf/SFER_code).

**Layer 3 (Hidden layer):** the Hidden layer has  $4 \times N_l$  neurons,  $2 \times N_l$  neurons for each branch. The values of neurons in the source-domain branch forms a  $2N_l$ -by-1 vector and is denoted by  $a_3^{si}$ . Similarly, we have another  $2N_l$ -by-1 vector  $a_3^{ti}$  for the target-domain branch. They are calculated by following equations,

$$a_3^{si} = g(W_2 a_2^{si} + b_2), a_3^{ti} = g(W_2 a_2^{ti} + b_2),$$

where  $W_2$  is a  $2N_l$ -by- $N_l$  matrix,  $b_2$  is a  $2N_l$ -by-1 vector,  $a_2^{si}$  and  $a_2^{ti}$  are calculated by Eq. (1) and  $g(\cdot)$  is a rectifier linear unit (ReLU) function,  $g(x) = \max(0, x)$ . It is clear that two branches share the weight matrix  $W_2$ .

**Layer 4 (Representation layer):** the Representation layer has  $4 \times N_l$  neurons,  $2 \times N_l$  neurons for each branch. Similar with the Hidden layer, values of neurons in this layer are calculated by following equations,

$$a_4^{si} = g(W_3 a_3^{si} + b_3), a_4^{ti} = g(W_3 a_3^{ti} + b_3), \quad (2)$$

where  $W_3$  is a  $2N_l$ -by- $2N_l$  matrix,  $b_3$  is a  $2N_l$ -by-1 vector and  $g(\cdot)$  is a ReLU function. Two branches share the  $W_3$  in this layer.

**Layer 5 (Output layer):** the Output layer has 4 neurons, 2 neurons for each branch (binary classification problem). Similar with the Hidden layer, values of neurons in this layer are calculated by following equations,

$$a_5^{si} = W_4 a_4^{si} + b_4, a_5^{ti} = W_4 a_4^{ti} + b_4, \quad (3)$$

where  $W_4$  is a 2-by- $2N_l$  matrix and  $b_4$  is a 2-by-1 vector. Two branches share the  $W_4$  in this layer.

##### 2) Cost function and learning process of SFER-NN

In this section, we introduce the cost function of SFER-NN and how to learn parameters of SFER-NN. In this paper, cross-entropy is used as the supervised loss, which is defined as follows.

$$\ell_c(a_5^{si}, y_{si}) = - \sum_{j=1}^2 1_{y_{si}=j} \log(h_j(X_{si})), \quad (4)$$

where  $h_j(X_{si}) = (a_5^{si})_j / \sum_{j'} (a_5^{si})_{j'}$ , is the softmax function that computes the probability of predicting instance  $X_{si}$  for  $j^{th}$  class. Then, the cost function of SFER-NN is defined as follows.

**Definition 1 (cost function of SFER-NN).** Given  $D_S = \{X_{si}, y_{si}\}$  and  $D_T = \{X_{ti}, y_{ti}\}$ , the cost function of the SFER-NN is defined as follows:

$$J(D_S, D_T) = \sum_{i=1}^{N_s} \ell_c(a_5^{si}, y_{si}) + \lambda \text{MMD}(A_4^s, A_4^t), \quad (5)$$

where  $\ell_c(a_5^{si}, y_{si})$  is defined in Eq. (4),  $A_4^s = [a_4^{s1} \dots a_4^{sL}]$  is a  $2N_l$ -by- $L$  matrix,  $A_4^t = [a_4^{t1} \dots a_4^{tL}]$  is a  $2N_l$ -by- $L$  matrix,  $\lambda$

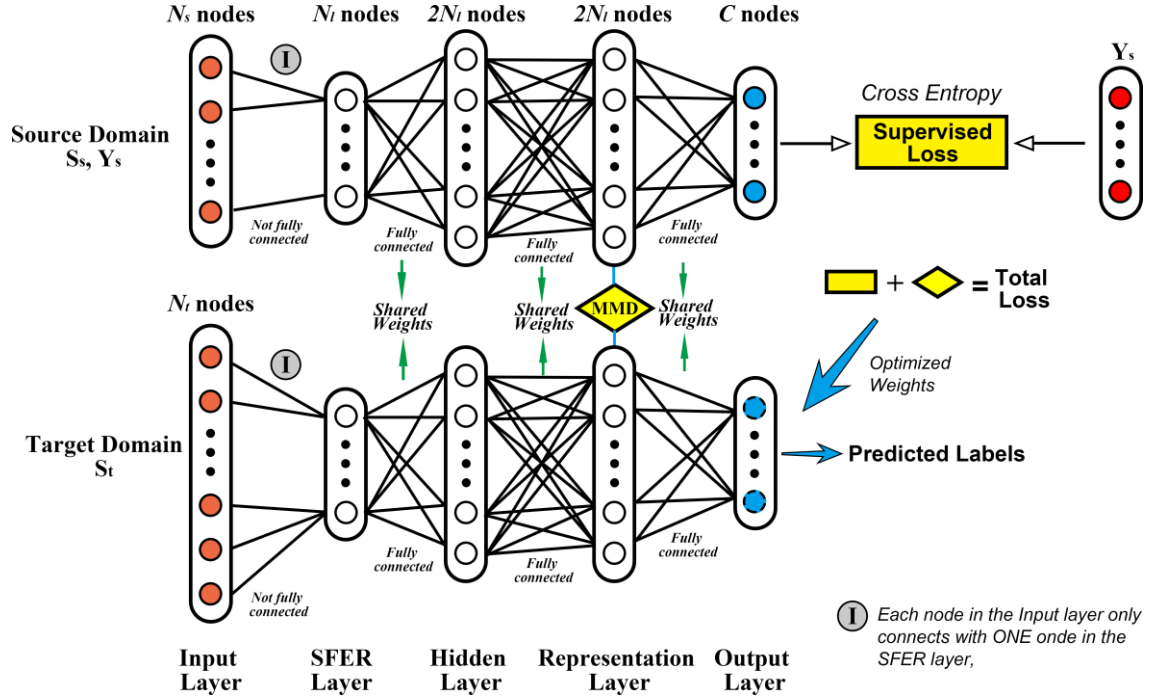


Fig. 2. The structure of SFERNN. SFERNN contains five layers and two branches: one from the source domain and another one from the target domain. Except for the weights connecting the first two layers, two branches share the weights. SFER directly confirms the weights connecting the first two layers and these weights will not be optimized in the learning process of the neural network. In the fourth layer, MMD is added as a regularizer to minimize the discrepancy between the fourth layers of two branches. Number of neurons are decided by the SFER model because they are related to  $N_l$ .

is the penalty parameter of  $MMD(A_4^s, A_4^t)$  and  $MMD(A_4^s, A_4^t)$  is calculated according to Eq. (3) in [31].

$W_1^s$ ,  $W_1^t$  and  $N_l$  will be directly confirmed by the SFER model, so  $W_1^s$  and  $W_1^t$  can be regarded as constant matrix in the cost function  $J(D_s, D_t)$  and there is no need to manually adjust structure of the SFER-NN (if  $N_l$  is fixed, the structure of this network is fixed). We use the Adam gradient decent algorithm to optimize the cost function  $J(D_s, D_t)$  to obtain the best  $\{W_i^*\}_{i=2}^4$  and  $\{b_i^*\}_{i=2}^4$ . Using  $W_1^s$ ,  $W_1^t$ ,  $\{W_i^*\}_{i=2}^4$  and  $\{b_i^*\}_{i=2}^4$ , instances in the target domain can be labelled by the SFER-NN. Algorithm 1 briefly presents the procedures of SFER-NN.

#### Algorithm 1. SFER-NN

##### Input:

$D_s$ : source domain;  $F_s$ : feature set of  $D_s$ ;  
 $D_t$ : target domain;  $F_t$ : feature set of  $D_t$   
 $\lambda$ : penalty parameter

##### Output:

$\hat{y}_{ti}$ : predicted class for  $X_{ti}$

- 1  $[F_s^{SFER}, F_t^{SFER}] = SFER(F_s, F_t)$ ;
- 2 Using  $F_s^{SFER}$  and  $F_t^{SFER}$  to construct  $W_1^s, W_1^t$ ;
- 3 Using Adam optimizer to get  $\{W_i^*\}_{i=2}^4$  and  $\{b_i^*\}_{i=2}^4$ ;
- 4 Using  $W_1^s, W_1^t, \{W_i^*\}_{i=2}^4$  and  $\{b_i^*\}_{i=2}^4$  to predict  $\hat{y}_{ti}$ ;
- 5 **Return** FFM<sub>S</sub> and FFM<sub>T</sub>.

## IV. EXPERIMENTS

In this section, we apply SFER-NN to two public datasets to test its classification performance on them. We begin by introducing the relevant datasets. Then the related benchmarks

are introduced. Lastly, the classification results will be analyzed.

### A. Datasets for HeUDA

The same datasets with our previous works are selected in this paper, from the UCI Machine Learning Repository (UMLR, <http://archive.ics.uci.edu/ml/index.html>). Both datasets are used to test the SFER-NN and benchmarks. They are related to personal credit assessment, namely German Credit Data and Australian Credit Approval. They have totally different feature spaces. One of our tasks is to use labelled instances in the German data to label instances in the Australian data, referred as G2A. Another task is A2G. Detailed descriptions of both tasks can be found in [26].

Table I gives the details of two HeUDA tasks. It should be pointed out that the unsupervised domain adaptation model works in these instances (G2A or A2G) because of the similarity between two domains (both are datasets for evaluating personal credit).

TABLE I. DETAILS OF THE TWO HEUDA TASKS

Source Domain	Target Domain	Labels	Task Name
German Credit Data	Australian Credit Approval	1: Good	G2A
Australian Credit Approval	German Credit Data	1: Good	A2G

### B. Benchmarks and implementation details

To demonstrate the superiority of the SFER-NN for addressing HeUDA problem, several benchmarks are selected for comparison purposes. Dimensional reduction

technology can be applied to force the two domains to have the same number of features. We denote this method as dimension reduction geodesic flow kernel (DG) where the dimensions of the mapped features are set as  $r$ . KCCA, as a HeUDA model in the literature, is selected as another benchmark model. Other benchmarks include our previous models: FFF-GFK, UFFFG and SFER. DG, KCCA, FFF-GFK, UFFFG and SFER are regarded as transfer models. Following previous works [16], [29], [32], we select two non-transfer models: all\_1 (A1) and the clustering method (CM). A1 labels all instances as “1” and CM clusters the instances and randomly labels the categories using the  $k$ -means algorithm. Because there is no labeled instance in the target domain, CM naturally gives extremely unstable prediction results.

The order of features and instances in each dataset was randomly permuted before training the SFER-NN. Following [16], [29], [32], 600 unbiased instances from the German Credit dataset are randomly selected for every experiment and we ran the experiment 50 times for each model and each task. Since there is no labeled instance in the target domain, it was impossible to automatically tune the optimal parameters for the target classifier using cross-validation. As a result, we used the Adam optimizer provided in the Pytorch 0.4.0 with default parameters to train our SFER-NN. We use the cuckoo search algorithm to train UFFFG model with recommend parameters in [29]. Because there is no existing pair for the two transfer learning tasks, we randomly matched instances from each domain as pairs for the KCCA model. We set  $\lambda$  to 1 for MMD regularizer.

*Accuracy* was used as the test metric as it has been widely adopted in the literature [7], [10], [20]. The definition follows.

$$Accuracy = \frac{|\{\mathbf{x}: \mathbf{x} \in \mathcal{X}_T \wedge g(\mathbf{x}) = y(\mathbf{x})\}|}{|\{\mathbf{x}: \mathbf{x} \in \mathcal{X}_T\}|},$$

where  $y(\mathbf{x})$  is the ground truth label of  $\mathbf{x}$ , while  $g(\mathbf{x})$  is the label predicted by domain adaptation models. All experiments were conducted on an Intel(R) Core(TM) i7-4770 CPU at 3.40Ghz with a memory of 64 GB running Windows 7 professional 64-bit operating system and Pytorch 0.4.0. We ran the experiments 50 times, preprocessing the instances with the zscore function.

### C. Classification results and analysis

The classification results are listed in Table II. From this table, it is clear that SFER-NN outperforms the other models. KCCA has a better result than DG, CM and A1 for task G2A. CM outperforms A1, DG and KCCA for A2G, meaning that KCCA and DG are invalid for task A2G. SFER is better than the other models, mainly due to the strong ability to construct homogeneous representations of two heterogeneous domains. The constructed representations can avoid the negative transfer. For example, KCCA has high Max accuracy when the task is G2A but it often has the situation of negative transfer. This results in its average accuracy being much lower and having higher standard derivation. The UFFFG model has better performance than FFF-GFK for task G2A because the Australian data has better clustering results. SFER-NN has a better classification results than SFER,

which means SFER-NN is more capable of reducing discrepancy between two domains. This results reflect that setting MMD as a regularizer in SFER-NN will improve the classification performance on the target domain.

TABLE II. PREDICTION RESULTS OF FFF-GFK AND BENCHMARKS

Model	Criteria	A2G	G2A
A1	Average	50%	50%
	Min	50%	50%
	Max	50%	50%
CM	Average	50.96%±5.22%	44.86%±0.43%
	Min	43.67%	43.77%
	Max	57.17%	56.23%
DG	Average	50.91%±1.03%	45.17%±1.95%
	Min	48.50%	41.16%
	Max	53.00%	52.61%
KCCA	Average	50.53%±4.61%	51.08%±9.78%
	Min	41.00%	31.50%
	Max	59.67%	71.00%
FFF-GFK	Average	59.48%±2.91%	69.53%±6.90%
	Min	50.33%	47.83%
	Max	65.00%	77.50%
UFFFG	Average	59.82%±2.88%	72.89%±6.20%
	Min	50.67%	56.00%
	Max	65.00%	83.83%
SFER	Average	60.33%±1.87%	76.85%±1.21%
	Min	56.00%	75.17%
	Max	65.00%	79.00%
SFER-NN	Average	<b>60.84%±1.98%</b>	<b>77.92%±1.26%</b>
	Min	<b>56.17%</b>	<b>56.00%</b>
	Max	<b>65.00%</b>	<b>83.83%</b>

The following table shows the running time of the benchmarks and SFER-NN. It is clear that SFER-NN needs more time to train its parameters.

TABLE III. RUNNING TIME OF EACH MODEL

	DG	FFF-GFK	UFFFG+CS	SFER	SFER-NN
Time (s)	1.8	1.5	344	62	126

## V. CONCLUSIONS AND FURTHER STUDY

This paper presents a novel heterogeneous unsupervised domain adaptation neural network, called shared fuzzy equivalence relations neural network. We first apply shared fuzzy equivalence relation model to confirm the structure of the neural network and assign weights in the first two layers. Then, maximum mean discrepancy is employed to minimize the discrepancy between two domains. Different from previous works, we first apply a neural network to address the heterogeneous unsupervised domain adaptation problem and SFER-NN shows a strong ability to transfer knowledge across two heterogeneous domains. The SFER-NN is tested on two public datasets and delivers satisfactory classification results. Based on the SFER-NN, our future work includes: 1) to develop a neural network to transfer knowledge from multi-source domains to an unlabeled target domain, and 2) to investigate a more effective regularizer to measure the discrepancy between domains.

## ACKNOWLEDGMENT

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Grant DP170101632.

## REFERENCES

- [1] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 10, pp. 1345–1359, 2010.
- [2] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, "Transfer learning using computational intelligence: A survey," *Knowledge-Based Syst.*, vol. 80, pp. 14–23, 2015.
- [3] B. Da, Y. S. Ong, A. Gupta, L. Feng, and H. Liu, "Fast transfer Gaussian process regression with large-scale sources," *Knowledge-Based Syst.*, vol. 165, pp. 208–218, 2019.
- [4] T. Liu, Q. Yang, and D. Tao, "Understanding how feature structure transfers in transfer learning," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, 2017, pp. 2365–2371.
- [5] V. Behbood, J. Lu, G. Zhang, and W. Pedrycz, "Multistep fuzzy bridged refinement domain adaptation algorithm and its application to bank failure prediction," *IEEE Trans. Fuzzy Syst.*, vol. 23, no. 6, pp. 1917–1935, 2015.
- [6] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Fuzzy regression transfer learning in Takagi-Sugeno fuzzy models," *IEEE Trans. Fuzzy Syst.*, vol. 25, no. 6, pp. 1795–1807, 2017.
- [7] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE Trans. Neural Networks*, vol. 22, no. 2, pp. 199–210, 2011.
- [8] J. Jiang and C. Zhai, "A two-stage approach to domain adaptation for statistical classifiers," in *Proceedings of the 16th ACM Conference on Conference on Information and Knowledge Management*, 2007, pp. 401–410.
- [9] B. Gong, K. Grauman, and F. Sha, "Learning kernels for unsupervised domain adaptation with applications to visual object recognition," *Int. J. Comput. Vis.*, vol. 109, no. 1–2, pp. 3–27, 2014.
- [10] W. Li, L. Duan, D. Xu, and I. W. Tsang, "Learning with augmented features for supervised and semi-supervised heterogeneous domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 6, pp. 1134–1148, 2014.
- [11] M. Xiao and Y. Guo, "Feature space independent semi-supervised domain adaptation via kernel matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 54–66, 2015.
- [12] H. Zuo, G. Zhang, W. Pedrycz, V. Behbood, and J. Lu, "Granular Fuzzy Regression Domain Adaptation in Takagi-Sugeno Fuzzy Models," *IEEE Trans. Fuzzy Syst.*, vol. Accept, 2017.
- [13] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep Transfer Learning with Joint Adaptation Networks," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 2208–2217.
- [14] A. Rozantsev, M. Salzmann, and P. Fua, "Beyond sharing weights for deep domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. Accept, 2018.
- [15] K. Saito, Y. Ushiku, and T. Harada, "Asymmetric Tri-training for Unsupervised Domain Adaptation," in *Proceedings of the 34th International Conference on Machine Learning*, 2017, pp. 2988–2997.
- [16] F. Liu, J. Lu, and G. Zhang, "Unsupervised Heterogeneous Domain Adaptation via Shared Fuzzy Equivalence Relations," *IEEE Trans. Fuzzy Syst.*, vol. 26, no. 6, pp. 3555–3568, 2018.
- [17] Y. Shi, L. Angeles, and F. Sha, "Information-theoretical learning of discriminative clusters for unsupervised domain adaptation," in *Proceedings of the 29th International Conference on Machine Learning*, 2012, pp. 1079–1086.
- [18] M. Long, J. Wang, Y. Cao, J. Sun, and P. S. Yu, "Deep learning of transferable representation for scalable domain adaptation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2027–2040, 2016.
- [19] N. Courty, R. Flamary, D. Tuia, S. Member, and A. Rakotomamonjy, "Optimal transport for domain adaptation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1853–1865, 2017.
- [20] M. Ghifary, D. Balduzzi, W. B. Kleijn, and M. Zhang, "Scatter component analysis: A unified framework for domain adaptation and domain generalization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1414–1430, 2017.
- [21] M. Gong, K. Zhang, T. Liu, D. Tao, C. Glymour, and I. Systems, "Domain adaptation with conditional transferable components," in *Proceedings of the 33rd International Conference on Machine Learning*, 2016, pp. 2839–2848.
- [22] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *Proceedings of the 30th AAAI conference on Artificial Intelligence*, 2016, pp. 2058–2065.
- [23] W. Deng, L. Zheng, G. Kang, Y. Yang, Q. Ye, and J. Jiao, "Image-Image Domain Adaptation with Preserved Self-Similarity and Domain-Dissimilarity for Person Re-identification," *arXiv Prepr. arXiv 1711.07027 [cs.CV]*, pp. 1–10, 2017.
- [24] Z. Zhong, L. Zheng, Z. Zheng, S. Li, and Y. Yang, "Camera Style Adaptation for Person Re-identification," *arXiv Prepr. arXiv 1711.10295 [cs.CV]*, pp. 1–10, 2017.
- [25] J. T. Zhou, S. J. Pan, I. W. Tsang, and Y. Yan, "Hybrid Heterogeneous Transfer Learning through Deep Learning," in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2213–2219.
- [26] P. Wei, Y. Ke, and C. K. Goh, "A General Domain Specific Feature Transfer Framework for Hybrid Domain Adaptation," *IEEE Trans. Knowl. Data Eng.*, vol. PP, no. c, pp. 1–1, 2018.
- [27] Y. R. Yeh, C. H. Huang, and Y. C. F. Wang, "Heterogeneous domain adaptation and classification by exploiting the correlation subspace," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2009–2018, 2014.
- [28] F. Liu, G. Zhang, and J. Lu, "Heterogeneous transfer learning: An unsupervised approach," *arXiv:1701.02511 [cs.LG]*, pp. 1–15, 2017.
- [29] F. Liu, G. Zhang, and J. Lu, "Unconstrained fuzzy feature fusion for heterogeneous unsupervised domain adaptation," in *Proceedings of the IEEE International Conference on Fuzzy Systems*, 2018, pp. 1–8.
- [30] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," in *Proceedings of the 30th Annual Conference on Neural Information Processing Systems*, 2016, pp. 136–144.
- [31] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain Adaptation via Transfer Component Analysis," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 2009, pp. 1187–1192.
- [32] F. Liu, G. Zhang, and J. Lu, "Heterogeneous unsupervised domain adaptation based on fuzzy feature fusion," *Proc. IEEE Int. Conf. Fuzzy Syst.*, pp. 1–6, 2017.