Neurocomputing 596 (2024) 127948

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semi-supervised heterogeneous domain adaptation for few-sample credit risk classification

Zhaoqing Liu, Guangquan Zhang, Jie Lu *

University of Technology Sydney, Broadway, Sydney, 2007, NSW, Australia

ARTICLE INFO

Communicated by N. Zeng

Keywords:
Transfer learning
Semi-supervised learning
Credit risk classification

ABSTRACT

Credit risk classification is a crucial area in machine learning-enhanced financial decision support systems, and numerous studies have achieved significant progress. However, data in the modern financial landscape is inherently complex, characterized by a lack of labeled data, cross-domain heterogeneity, and class imbalance. These three major problems hinder the achievement of credit risk classification. Therefore, we propose a semi-supervised heterogeneous domain adaptation method and an imbalanced data augmentation method to overcome these challenges with a single neural network-based model. Experimental results obtained from four representative benchmark datasets confirm the superior performance of the proposed method.

1. Introduction

Out-of-control risk management can lead to non-performing assets, bringing substantial financial losses to financial institutions, and many stakeholders, including honest borrowers with good credit histories, will not be able to stay out of it. Before the 2008 recession, neglect of risk controls significantly triggered the collapse of collateralized debt obligations, leading to massive losses and bankruptcies of many financial and non-financial institutions worldwide [1]. The resulting recession further depressed demand and buying activities in the economy to extremely low levels, resulting in severe unemployment. The entire economy could suffer if financial market participants do not adequately assess current and future losses arising from risk. For this reason, in many countries and market economies, financial market participants have developed credit risk models in accordance with Basel I, Basel II, Basel III, and the International Financial Reporting Standard 9 (IFRS 9) [1]. The significance of credit risk modeling mainly lies in three aspects: (1) Reminding borrowers to identify risks early and adjust their financial activities and strategies; (2) Helping financial institutions to determine the financial risks of borrowers and make reasonable loan decisions; (3) Assisting financial regulators to grasp the financial situation of borrowers in a timely manner and strengthen supervision to maintain the stability of the financial market.

Machine learning (ML) and judgmental methods are the two main approaches to credit risk modeling [2], with ML methods are currently considered more promising than traditional judgmental approaches, as financial institutions need to make real-time credit decisions to remain competitive in the digital world. Among the various methods for credit risk classification (CRC), the most popular techniques include Logistic Regression, Survival Analysis, Random Forests, Gradient Boosting, Markov Chain Modeling, and Neural Networks [2,3]. The advantages of ML methods include automatic and timely decisions [4], high accuracy driven by big data [5], impartiality [6], and low likelihood of artificial dishonesty [3].

However, CRC often faces challenging learning environments in the new financial world. Three primary issues are lack of labeled data, cross-domain heterogeneity, and class imbalance. Firstly, many financial institutions have limited labeled data to train ML models due to the costly process of labeling vast amounts of data [7]. Some new financial institutions or applications may even lack labeled data altogether. Secondly, while transfer learning can mitigate the shortage of labeled data by leveraging knowledge from related domains and enhancing learning performance in the current domain [7,8], cross-domain heterogeneity poses a significant barrier to achieving successful transfer. Lastly, the data in CRC applications is often inherently imbalanced [9], easily leading to a trained model exhibiting high accuracy in predicting the majority class but poor accuracy in predicting the minority class. In other words, such a biased model would overfit the majority class, and the correlations between the data features and classes it captures would be false. This has significant implications for financial institutions, as mistakenly accepting borrowers with high credit risk is far more detrimental than wrongly rejecting borrowers with low credit risk. Thus, for CRC under class imbalances, minorities carry more critical information than majorities, although they are smaller in number and less representative. Accordingly, the primary

* Corresponding author.
E-mail addresses: zhaoqing.liu-1@student.uts.edu.au (Z. Liu), guangquan.zhang@uts.edu.au (G. Zhang), jie.lu@uts.edu.au (J. Lu).

https://doi.org/10.1016/j.neucom.2024.127948
Received 21 January 2024; Received in revised form 29 April 2024; Accepted 27 May 2024
Available online 28 May 2024

learning goal of credit risk modeling is usually the ability to classify those minorities correctly. Training a model on imbalanced data results in three inherent flaws [10]: (1) Biased: The model tends to be biased towards the majority class and struggles to recognize patterns and characteristics in the minority class. (2) Over-fitting: Due to the first flaw, the model may over-fit the majority class, assuming there are no minority classes in most cases and prioritizing the prediction of the majority class. (3) False correlations: The model fails to capture valid correlations between classes and features in imbalanced data, which is crucial for credit risk models to understand how each feature affects both the minority and majority classes.

In the field of CRC, ML methods outperform traditional statistical methods in performance, especially when dealing with nonlinear patterns [11]. In prior studies, supervised ML methods are still the most popular, and the performance of ensemble or hybrid algorithms is superior to that of standalone algorithms [11,12]. To the best of our knowledge, semi-supervised heterogeneous domain adaptation that attempts to solve CRC problems with a lack of labeled data, cross-domain heterogeneity, and class imbalance is a relatively emerging research area. The above three challenges often coexist in CRC applications; however, existing ML methods can simultaneously solve only some of these problems. Therefore, we propose a semi-supervised heterogeneous domain adaptation approach called Semi-Supervised Transfer Adaptive Neural Forests (STANF) and an Imbalanced Data Augmentation (IDA) method to address all these problems with a single neural network-based model. The primary improvement of STANF over current CRC methods is its innovative adoption of semi-supervised heterogeneous domain adaptation to solve the few-sample CRC problem. When transferring knowledge from the source domain to improve learning from the target domain, STANF can identify and strengthen the related cross-domain information, equivalent to weakening the unrelated cross-domain information to mitigate the negative effect of cross-domain heterogeneity. Moreover, STANF can utilize the information available in unlabeled data to aid learning from a small amount of labeled data through predictive and structural consistency. Another significant improvement of STANF is that it addresses the inherent class imbalance prevalent in financial data. A cost-sensitive loss function is proposed to improve the quality of knowledge acquisition from class-imbalanced source data. IDA is also proposed to compensate for the lack of attention to the minority class when learning from class-imbalanced target data. The contributions of the paper are summarized as follows:

- It enhances semi-supervised heterogeneous domain adaptation methods for few-sample credit risk classification applications under various few-label and class-imbalanced regimes.
- By integrating heterogeneous transfer learning and imbalanced learning techniques, the proposed method can solve the problems of lack of labeled data, cross-domain heterogeneity and class imbalance with only one model. Experiments on four representative benchmark datasets in credit risk classification demonstrate the method's superiority.
- It also suggests a novel imbalanced data augmentation method for semi-supervised heterogeneous domain adaptation. This method has three main advantages: (1) It can mitigate data bias mainly caused by fewer labels and imbalanced classes; (2) It can improve the generalization performance of models; (3) It is compatible with end-to-end learning and can be easily extended to other learning methods. Extensive experiments indicate the efficacy of the method.

2. Related work

This section provides an overview of existing methods for semi-supervised heterogeneous domain adaptation and related methods for imbalanced classification and imbalanced data augmentation.

2.1. Semi-supervised heterogeneous domain adaptation

Semi-supervised heterogeneous domain adaptation is one of the transfer learning settings [8,13]. In the process of using the knowledge from the source domain to improve the classification performance of the target domain, the source and the target domains are heterogeneous, but the labels are shared, and the source domain has the labels while the target domain only has a few labels for each class. Semi-supervised heterogeneous domain adaptation methods can be roughly divided into four types: geometric or statistical alignment, instance reweighting, pseudo-label strategies, and feature augmentation [14]. For example, the instance-reweighting method, Cross-Domain Landmark Selection [15], learns two linear feature transformations and estimates the source and target data weights. Another example is the pseudo-label-strategy method, Generalized Joint Distribution Adaptation [16], which projects source data and target data into a latent space by learning feature transformations and then eliminates the difference between projected cross-domain heterogeneous data. In addition, the hybrid of neural networks and ensemble trees has also made successful progress in dealing with Semi-supervised heterogeneous domain adaptation problems, in which the Transfer Neural Tree [17] method selects the appropriate feature mapping from the latent space extracted by neural networks to jointly solve cross-domain feature mapping, adaptation, and classification problems.

2.2. Imbalanced classification

Imbalanced classification refers to a classification predictive modeling problem where the number of samples over each class in the training dataset is not balanced (i.e., the class distribution is not equal or close to equal but biased or skewed) [18]. The existing methods for imbalance classification can be roughly summarized into three categories: rebalancing sampling, cost-sensitive learning, and algorithm improvement. Rebalancing sampling aims to balance the number of training samples for each class during model training [18]. It includes random over-sampling and under-sampling techniques [10]. The former provides balanced data for model training by increasing the probability of the minority class being selected through various sampling strategies, while the latter achieves this by decreasing the probability of the majority class being selected [10]. Cost-sensitive learning techniques induce model training by balancing the degree of attention to each class, specifically reducing the contribution of the majority class to the loss function while increasing the contribution of the minority class [18]. In cost-sensitive learning, class-level reweighting adjusts the loss values of different classes according to the label frequency of the training samples. The class-level re-margining deals with class imbalance by adjusting the minimum distance between the features learned by other classes and the classifier's decision boundary [18]. From the perspective of algorithm improvement, imbalanced classification methods mainly include modes: (1) improving the feature extractor by optimizing representation learning, (2) enhancing the classifier by designing imbalanced model training methods, (3) refining the learning process of the feature extractor and classifier by decoupling training, and (4) improving the robustness and generalization of the network architecture by ensemble learning [19].

2.3. Imbalanced data augmentation

Imbalanced data augmentation introduces additional information into model training to enhance the quantity and quality of training data to improve imbalanced learning performance [20]. Among them, various methods [20–22] based on Mixup [23] make the synthesized samples closer to the minority class by adjusting the parameters of Mixup, thus expanding the domain covered by the training set. Such methods enable a model to learn the results of simple linear interpolation of labels in the in-between domain not covered by the original training samples, thereby improving the model's generalization performance beyond the training samples. [24] explains Mixup from the perspective of regularization, arguing that linear interpolation is based on the simplified spatial assumption that linear weights of inputs can be mapped to linear weights of outputs. This assumption acts as a priori to regularize the model training to make the decision boundaries smoother and further away from the high-density domain of the samples. Various imbalanced data augmentation methods based on Mixup can be implemented as on-the-fly modules to be transparently used in a network architecture instead of explicitly using other imbalanced learning methods that introduce complexity. This is especially desirable for standard machine learning methods that assume roughly equal samples over each class.

3. Problem formulation

3.1. Basic definitions

Before giving the problem setup, let us introduce the relevant definitions. Let $\mathcal{X} \subset \mathbb{R}^d$ be a feature space and $\mathcal{Y} = \{1, \ldots, C\}$ be a label space. $P(X, Y)$ is a joint distribution, where $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$ are random variables.

Definition 1 (*Few-Label Classification*)**.** Given a training dataset D consisting of two subsets called labeled and unlabeled data:

$$D_L = \{(\mathbf{x}_1^l, y_1^l), \ldots, (\mathbf{x}_{n^l}^l, y_{n^l}^l)\} \sim P(X, Y) \;\; i.i.d.,$$
$$D_U = \{\mathbf{x}_1^u, \ldots, \mathbf{x}_{n^u}^u\} \sim P(X) \;\; i.i.d.,$$

where $D_L \cap D_U = \emptyset$, n^l and n^u are the number of labeled and unlabeled samples, and $n^l \ll n^u$. Few-label classification refers to a classification predictive modeling problem that seeks to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ from D to accurately classify data from $\mathcal{X}$.

Definition 2 (*Imbalanced Classification*)**.** Given a training dataset $D = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\} \sim P(X, Y)$ *i.i.d.*, where each sample has a corresponding label (i.e., $n = \sum_{c=1}^{C} |D_c|$, here D_c is a set of samples belonging to class c), and $n_{Min} \ll n_{Maj}$, here $n_{Min} = \min_{c\in\mathcal{Y}} \{|D_c|\}$ and $n_{Maj} = \max_{c\in\mathcal{Y}} \{|D_c|\}$ represent the largest and smallest class sizes over all classes (i.e., the number of the minority and majority samples). Imbalanced classification refers to a classification predictive modeling problem that aims to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ from D with an imbalanced class distribution to accurately classify data from $\mathcal{X}$ (especially those belonging to minority classes). In an imbalanced classification problem, the degree of class imbalance (DCI) is defined in terms of the maximum imbalance ratio between class sizes, i.e.,

$$\mathrm{DCI} = \frac{n_{\mathrm{Maj}}}{n_{\mathrm{Min}}} = \frac{\max_{c\in\mathcal{Y}} \{|D_c|\}}{\min_{c\in\mathcal{Y}} \{|D_c|\}}. \tag{1}$$

A multi-class problem can be transformed into a set of binary classification problems through class decomposition. Accordingly, when $\mathcal{Y} = \{0, 1\}$, Definition 2 degenerates into imbalanced binary classification, as follows:

Definition 3 (*Imbalanced Binary Classification*)**.** Given a training dataset D consisting of two sets of samples called positive and negative data:

$$D_P = \{(\mathbf{x}, y) \in D | y = 1\},$$
$$D_N = \{(\mathbf{x}, y) \in D | y = 0\},$$

where $D_P \cup D_N = D$, $D_P \cap D_N = \emptyset$, and $|D_P| \ll |D_N|$. Thus, D_P and D_N are also called the minority and majority samples (or simply the minorities and majorities), and positive and negative classes are also referred to as the minority and majority classes. Imbalanced binary classification refers to a binary classification predictive modeling problem that aims to learn a classifier $f : \mathcal{X} \to \mathcal{Y}$ from D with an imbalanced class distribution to accurately classify data from $\mathcal{X}$ (especially those belonging to the minority class). In an imbalanced binary classification problem, the degree of class imbalance (DCI) is defined as the ratio of the majority to minority class sizes, as follows:

$$\mathrm{DCI} = \frac{n_{\mathrm{Maj}}}{n_{\mathrm{Min}}} = \frac{|D_N|}{|D_P|}. \tag{2}$$

3.2. Problem setup

The credit risk classification studied in this paper can be summarized as a transfer learning problem in which the source and target domains are heterogeneous, the available labeled data from the target domain is small, and the class distributions of source and/or target domains are imbalanced. We call it a semi-supervised heterogeneous domain adaptation problem under a few-label and class-imbalanced regime (SHDA-under-FL&CI). The complexity of SHDA-under-FL&CI can be seen from the fact that source learning and/or target learning are imbalanced classification problems, and target learning is also a few-label classification problem. Let $\mathcal{X}^s \subset \mathbb{R}^{d_1}$, $\mathcal{X}^t \subset \mathbb{R}^{d_2}$ be feature spaces and $\mathcal{Y} = \{1, \ldots, C\}$ be a label space. *Source domain* and *target domain* are two different joint distributions $P(X^s, Y^s)$ and $P(X^t, Y^t)$, where $X^s \in \mathcal{X}^s$, $X^t \in \mathcal{X}^t$ and $Y^s, Y^t \in \mathcal{Y}$ are random variables. Then, we propose SHDA-under-FL&CI as follows:

Definition 4 (*SHDA-under-FL&CI*)**.** Given sets of samples called the *labeled source*, *labeled target* and *unlabeled target* data drawn from heterogeneous domains ($\mathcal{X}^s \neq \mathcal{X}^t$):

$$D_S = \{(\mathbf{x}_1^s, y_1^s), \ldots, (\mathbf{x}_{n^s}^s, y_{n^s}^s)\} \sim P(X^s, Y^s) \;\; i.i.d.,$$
$$D_L = \{(\mathbf{x}_1^l, y_1^l), \ldots, (\mathbf{x}_{n^l}^l, y_{n^l}^l)\} \sim P(X^t, Y^t) \;\; i.i.d.,$$
$$D_U = \{\mathbf{x}_1^u, \ldots, \mathbf{x}_{n^u}^u\} \sim P(X^t) \;\; i.i.d.,$$

where $n^l \ll n^u$, $n^l \ll n^s$, $n^s_{Min} \ll n^s_{Maj}$ and/or $n^l_{Min} \ll n^l_{Maj}$. The degree of class imbalance in D_L and D_S can be calculated by Eq. (1). A semi-supervised heterogeneous domain adaptation problem under few labels and class imbalance is to train a target classifier $f : \mathcal{X}^t \to \mathcal{Y}$ using data D_S, D_L and D_U such that f classifies the unlabeled target data accurately.

4. Proposed methods

This section begins with an introduction to the proposed model architecture, followed by elaborations on how the model works in forward and backpropagation. It then elucidates the operation of the imbalanced data augmentation method. Finally, it summarizes the learning framework.

4.1. Model architecture

Considering the robust representation learning and generalization performance of neural networks [14,25,26] as well as the potential of Random Forests in selecting hidden features [27], our focus is on investigating hybrid model architectures that combine neural networks and random forests. Building on the insights from prior studies [17,28], we propose a novel model architecture for SHDA-under-FL&CI. Instead of using traditional techniques such as softmax, we further propose Adaptive Neural Forests (ANF), a variant of Deep Neural Decision Forests [28], as the output layer of the model architecture.

Figure 1 shows an example of the proposed architecture, where NN represents a conventional neural network, which we call the representation learning layer; ANF is a neural network-based ensemble model consisting of a set of neural trees (NTs), which we refer to as the classification learning layer. Each NT in ANF is a neural network-based tree model that seamlessly integrates a fully connected layer (as shown by the intersection layer of NN and NT in Figure 1) and a decision tree, making end-to-end learning possible.

Notably, NN and ANF are specifically designed to be loosely coupled, whereby we have the flexibility to replace NN with any other

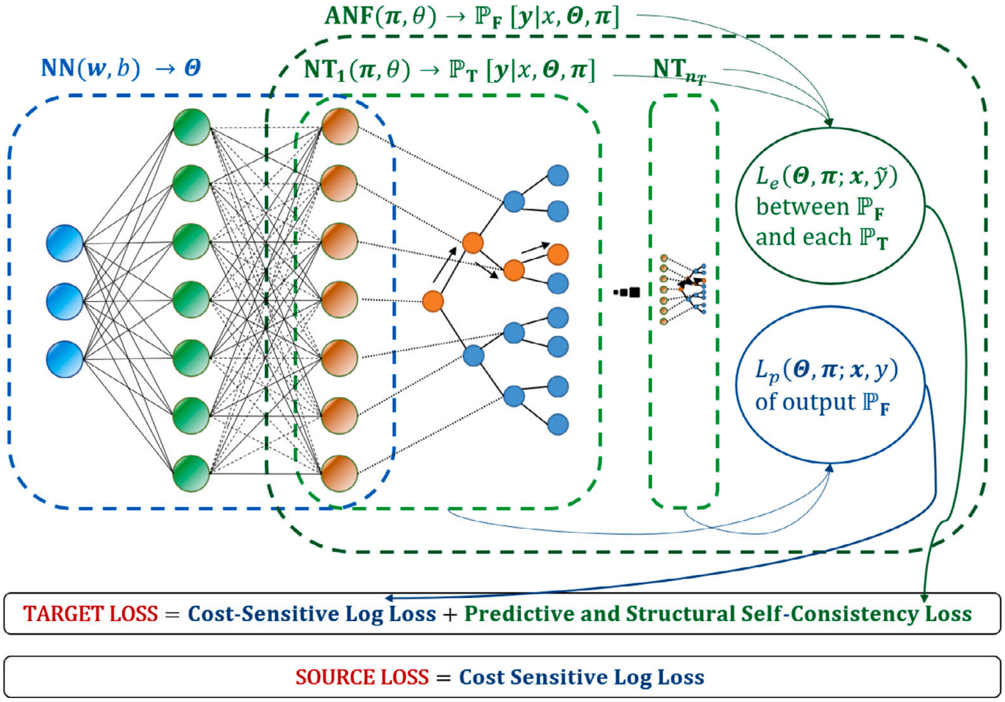
$\mathbf{ANF}(\boldsymbol{\pi},\theta) \to \mathbb{P}_{\mathbf{F}}\,[\boldsymbol{y}|x,\boldsymbol{\Theta},\boldsymbol{\pi}]$
$\mathbf{NN}(\boldsymbol{w},b) \to \boldsymbol{\Theta}$
$\mathbf{NT_1}(\boldsymbol{\pi},\theta) \to \mathbb{P}_{\mathbf{T}}\,[\boldsymbol{y}|x,\boldsymbol{\Theta},\boldsymbol{\pi}]$
$\mathbf{NT}_{n_T}$
$L_e(\boldsymbol{\Theta},\boldsymbol{\pi};\boldsymbol{x},\tilde{y})$ between $\mathbb{P}_{\mathbf{F}}$ and each $\mathbb{P}_{\mathbf{T}}$
$L_p(\boldsymbol{\Theta},\boldsymbol{\pi};\boldsymbol{x},y)$ of output $\mathbb{P}_{\mathbf{F}}$
TARGET LOSS = Cost-Sensitive Log Loss + Predictive and Structural Self-Consistency Loss
SOURCE LOSS = Cost Sensitive Log Loss

Fig. 1. An example of the model architecture.

type of neural network that can efficiently extract representations depending on the characteristics of the data. It is also important to note that the fully connected layer of each NT in ANF implements the Random Forests' random feature selection strategy, making each NT a random adapter. This property is the basis of ANF's adaptability across heterogeneous domains. Although some NTs in ANF may share some parameters of NN, the random feature selection strategy can still retain the independence of each NT.

4.2. Forward propagation

4.2.1. Representation learning layer

The representation learning layer of the model extracts representations from the input feature space into the feature mapping space. Depending on the characteristics of the data, various neural networks can be flexibly selected to perform representation learning. The NN illustrated in Fig. 1 is an example of a fully connected layer with a hyperbolic tangent function as its activation function. In other words, the representation learning layer operates in the same forward propagation manner as a conventional neural network. In this paper, we use $\mathcal{F}_S$ and $\mathcal{F}_T$ to represent the representational learning layers of the source and target models, respectively. Both have a set of trainable parameters denoted by $\boldsymbol{\Theta}$.

4.2.2. Classification learning layer

As the classification learning layer of the model, ANF (denoted by $\mathcal{G}$) adopts the Random Forests learning paradigm to randomly route the feature maps extracted by the representation learning layer and produce the final output. Traditional deterministic decision trees cannot be seamlessly integrated with neural networks because they are incompatible with backpropagation algorithms. Therefore, we utilize the Neural Tree from dNDF [28] as the base learner of $\mathcal{G}$. Such a design allows us to leverage the random routing and differentiable properties of neural trees to learn all decision and leaf nodes through backpropagation. However, neural trees are not easily extended to semi-supervised heterogeneous domain adaptation tasks. To address this limitation, we introduce stochastic pruning and embedding loss techniques from [17] into ANF. Also, we propose a cost-sensitive loss function combined with rebalancing sampling to handle class imbalance.

ANF initially distributes the input, the feature maps output by the representation learning layer, to each of its neural trees to continue forward propagation. Within each tree, propagation proceeds as follows: (1) the fully connected layer first randomly selects a portion of the input based on a predetermined fraction and then outputs it one-to-one to the decision nodes (except the leaf nodes) of the decision tree. This implies that the number of neurons in the fully connected layer is exactly equal to the number of the decision nodes in the decision tree; (2) the decision nodes are responsible for randomly routing the input to the leaf nodes, which is equivalent to hierarchically dividing the decision space by hyperplanes to build a high-dimensional manifold decision space; (3) the leaf nodes finally give a probability distribution of the input over all classes as the output of the individual tree.

Given an adaptive neural forest $\mathbf{F}$ composed of n_T neural trees, each tree $\mathbf{T}$ consists of $\mathcal{N}$ internal decision nodes and $\mathcal{L}$ terminal leaf nodes.

Decision Nodes in Trees. Let $\mu_\ell(\mathbf{x}|\boldsymbol{\Theta})$ be the stochastic routing function of decision node $n \in \mathcal{N}$ for the probability of a sample $\mathbf{x}$ reaching leaf node $\ell \in \mathcal{L}$, which is defined as

$$\mu_\ell(\mathbf{x}|\boldsymbol{\Theta}) = \prod_{n\in\mathcal{N}} d_n(\mathbf{x};\boldsymbol{\Theta})^{\mathbb{1}_{\ell\swarrow n}}\bar{d}_n(\mathbf{x};\boldsymbol{\Theta})^{\mathbb{1}_{\ell\searrow n}}, \tag{3}$$

where $d_n(\mathbf{x};\boldsymbol{\Theta})$ denotes the decision function that determines whether the sample $\mathbf{x}$, at decision node n, will be directed to the left or right node at the subsequent level, i.e., with the sigmoid function $\sigma = (1+e^{-x})^{-1}$ and the decision node n with the architecture weights θ_n,

$$d_n(\mathbf{x};\boldsymbol{\Theta}) = \sigma(f_n(\mathbf{x};\boldsymbol{\Theta})), \tag{4}$$

here

$$f_n(\mathbf{x};\boldsymbol{\Theta}) = \theta_n^T\mathbf{x}; \tag{5}$$

and the indicator functions $\mathbb{1}_{\ell\swarrow n}$ and $\mathbb{1}_{\ell\searrow n}$ represent the decisions made when traversing a path along $\mathbf{T}$.

Leaf Nodes in Trees. Let π_{ℓ_y} be the class-label distribution for sample $\mathbf{x}$ in leaf node ℓ, which is updated at iteration $t+1$ by

$$\pi_{\ell_y}^{(t+1)} = \frac{1}{Z_\ell^{(t)}}(p_d + \sum_{(\mathbf{x},y')\in D_S}\frac{\mathbb{1}_{y=y'}\,\mu_\ell(\mathbf{x}|\boldsymbol{\Theta})\pi_{\ell_y}^{(t)}}{\mathbb{P}_{\mathbf{T}}[y|\mathbf{x},\boldsymbol{\Theta},\boldsymbol{\pi}^{(t)}]}), \tag{6}$$

where $\boldsymbol{\pi}^{(t)}$ represents the derived class-label distribution at iteration t, and $Z_{\ell}^{(t)}$ is the normalizing factor ensuring that $\sum_{(y)} \pi_{\ell_y}^{(t+1)} = 1$. The starting point $\pi^{(0)}$ can be arbitrary as long as each member is positive. Note that p_d is a pruning term [17], which can suppress the extreme disparity in the class-label distribution when updating each leaf node. Using Eq. (6) is equivalent to a stochastic pruning process, which can improve the adaptability of $\mathcal{G}$ in the transfer learning across heterogeneous domains [17].

Prediction of Trees. The final prediction made by each individual tree $\mathbf{T}$ in forest $\mathbf{F}$ for sample $\mathbf{x}$ is given by

$$\mathbb{P}_{\mathbf{T}}[y|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}] = \sum_{\ell \in \mathcal{L}} \mu_{\ell}(\mathbf{x}|\boldsymbol{\Theta}) \pi_{\ell_y}. \tag{7}$$

Prediction of Forests. With the predictions for sample $\mathbf{x}$ given by all individual trees, the entire forest $\mathbf{F}$ gives the final overall prediction by

$$\mathbb{P}_{\mathbf{F}}[y|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}] = \frac{1}{n_T} \sum_{\mathbf{T} \in \mathbf{F}} \mathbb{P}_{\mathbf{T}}[y|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]. \tag{8}$$

4.3. Backpropagation

With the compatibility of ANF (i.e., $\mathcal{G}$) for backpropagation algorithms, models can be trained end-to-end by optimizing the total loss in source and target learning. However, source and target learning differ according to the objective function in their optimization task.

4.3.1. Source learning

The source model is trained on labeled source data D_S in a supervised manner to prepare adaptive and unbiased classification knowledge for training the target model. The loss function for labeled training sample $(\mathbf{x}, y)$ in learning is determined by the log-loss term, which is as follows:

$$L(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y) = -\log(\mathbb{P}_{\mathbf{T}}[y|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]). \tag{9}$$

Then, our goal is to minimize the following risk term:

$$L_p(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y) = \frac{1}{n_T} \sum_{\mathbf{T} \in \mathbf{F}} \sum_{i=1}^{n^s} \frac{1}{\sum_{i=1}^{n^s} w_{y_i}} L_p(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y), \tag{10}$$

where w_{y_i} is the weight assigned to class y_i to which the ith training sample belongs. Note that the term $1/\sum_{i=1}^{n^s} w_{y_i}$ increases the contribution of the minority class in imbalanced data to the loss, making parameter updates more sensitive to the minority class and vice versa for the majority class. We refer to Eq. (10) as a cost-sensitive loss function (CS-loss). The objective function based on such loss function can guide the learning to pay more attention to the minority class, thereby reducing the risk of the model being biased towards the majority class. With Eq. (10) as the overall objective function of optimization, when learning from D_S, both the representation learning layer $\mathcal{F}_S$ and classification learning layer $\mathcal{G}$ of the source model is updated by

$$\min_{\mathcal{F}_S, \mathcal{G}} \sum_{(\mathbf{x}, y) \in D_S} L_p(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y). \tag{11}$$

Accordingly, in back-propagation, the gradient of the loss L_p with respect to θ can be decomposed by the chain rule as follows:

$$\frac{\partial L_p}{\partial \theta}(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y) = \sum_{n \in \mathcal{N}} \frac{\partial L_p(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y)}{\partial f_n(\mathcal{F}(\mathbf{x}), \boldsymbol{\Theta})} \frac{\partial f_n(\mathcal{F}(\mathbf{x}), \boldsymbol{\Theta})}{\partial \theta}, \tag{12}$$

where $\mathcal{F}$ includes $\mathcal{F}_S$ and $\mathcal{F}_T$.

4.3.2. Target learning

Knowledge learned from the source data is transferred by reusing the trained classification learning layer $\mathcal{G}$ of the source model in the target model. Meanwhile, the target model is trained on labeled target data D_L and unlabeled target data D_U in a semi-supervised manner, which means that only the representation learning layer $\mathcal{F}_T$ of the target model is updated in the backpropagation, while the reused $\mathcal{G}$ is fixed.

Heterogeneous domain adaptation. Reusing $\mathcal{G}$ enables the target learning to observe and maintain the predictive and structural consistency between labeled cross-domain data, thus achieving heterogeneous domain adaptation. Furthermore, as shown in Eq. (6), the stochastic pruning process [17] in $\mathcal{G}$ allows the target learning to recognize and adapt to the representative neurons involved in the cross-domain data. Moreover, as described in Section 4.1, the random feature selection strategy in $\mathcal{G}$ further enhances the adaptability of the target learning to heterogeneous transfer learning.

Semi-supervised learning. In order to learn the target model from D_L and D_U in a semi-supervised manner, instead of updating $\mathcal{F}_T$ by Eq. (11) alone, we introduce the loss term from [17], which is defined as

$$\begin{aligned} L_e(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}) &= \frac{1}{n_T} \sum_{\tilde{y}} -\mathbb{P}_{\mathbf{T}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}] \frac{\mathbb{P}_{\mathbf{F}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]}{\mathbb{P}_{\mathbf{F}}[\tilde{y}|\boldsymbol{\Theta}, \boldsymbol{\pi}]} \\ &= -\sum_{\tilde{y}} \frac{\mathbb{P}_{\mathbf{F}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]^2}{\mathbb{P}_{\mathbf{F}}[\tilde{y}|\boldsymbol{\Theta}, \boldsymbol{\pi}]}, \end{aligned} \tag{13}$$

where $\tilde{y}$ represents the output label of input $\mathbf{x} \in \{D_L, D_U\}$, and $\mathbb{P}_{\mathbf{F}}[\tilde{y}|\boldsymbol{\Theta}, \boldsymbol{\pi}]$ is computed by

$$\mathbb{P}_{\mathbf{F}}[\tilde{y}|\boldsymbol{\Theta}, \boldsymbol{\pi}] = \frac{1}{n^l + n^u} \sum_{\mathbf{x} \in \{D_L, D_U\}} \mathbb{P}_{\mathbf{F}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}], \tag{14}$$

which can be regarded as a normalized factor. Accordingly, the optimization task when learning from the target data becomes

$$\min_{\mathcal{F}_T} \sum_{(\mathbf{x}, y) \in D_L} L_p(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y) + \lambda \sum_{(\mathbf{x}, y) \in D_L} L_e(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}), \tag{15}$$

where λ is a regularization parameter to adjust the learning rate corresponding to L_e. The derivatives of L_p and L_e with respect to Θ can then be used to update the model parameters in backpropagation. The derivative of L_p is the same as Eq. (12), while the partial derivative of L_e with respect to f_n is calculated as follows:

$$\frac{\partial L_e}{\partial \Theta}(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x}, y) = \sum_{\tilde{y}} \frac{\partial L_e(\boldsymbol{\Theta}, \boldsymbol{\pi}; \mathbf{x})}{\partial \mathbb{P}_{\mathbf{T}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]} \frac{\partial \mathbb{P}_{\mathbf{T}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]}{\partial f_n(\mathcal{F}(\mathbf{x}), \boldsymbol{\Theta})}. \tag{16}$$

The details of the above derivations can be found in [17].

4.4. Imbalanced data augmentation

Neither cost-sensitive learning nor rebalancing sampling methods can fully capture the true distribution of the data with few labels and imbalanced classes, nor can they effectively solve information lack or loss [29]. The lack of information usually causes the model to misclassify the sample space as smaller than the actual one, and the loss of information often results in the sample space learned by the model deviating from the actual one. In other words, none of these methods can fully solve the dilemma that a small group of samples has difficulty representing the true sample space. To address the above data bias, we propose an Imbalanced Data Augmentation (IDA) method. It enables STANF to use various combinations of data augmentation and rebalancing sampling techniques. The basic strategy of IDA is to enhance the dominance of the minority class in training and finally make the decision boundary given by the model move towards the majority sample space. IDA prefers to synthesize mixed samples closer to the minority class, thereby strictly constraining the majority sample space from which the model may learn. Moreover, our IDA has the advantage of typical data augmentation methods, which regularize the trained model by adding convex perturbations, a.k.a. out-of-manifold samples [30], to the training data, thereby improving the model's generalization performance. Benefiting from recent advances in imbalanced data augmentation techniques [20–22], we also use Mixup [23] as the base mixer for synthesizing samples in IDA.

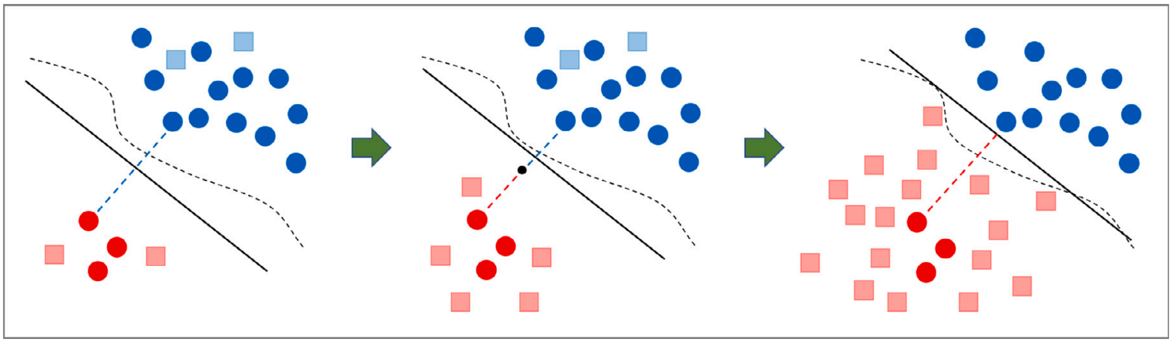

Fig. 2. An illustration of how IDA changes the marginal space from which the mixed samples are synthesized and how this affects the decision boundaries for classification. The blue and red dots represent majority and minority samples, respectively, and the light blue and light red blocks represent the augmented majority and minority samples, respectively. Let the black dashed curves be the true boundaries between the majority and the minority sample spaces, and the black solid lines be the decision boundaries provided by the classifier. The blue dashed lines span the possible marginal spaces for synthesizing both majority and minority samples, while the red dashed lines span only the possible marginal spaces for synthesizing minority samples.

Given an imbalanced dataset $D = \{(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\}$, IDA synthesizes dataset $D^{IDA} = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i), \ldots, (\tilde{\mathbf{x}}_m, \tilde{y}_m)\}$ from D, where $m = r \times n$ and $r \in \mathbb{N}$ represents how many times the sample size is to be increased. Mixed samples are specifically synthesized by convex combinations of pairs of random samples and their labels, i.e., each mixed sample $(\tilde{\mathbf{x}}, \tilde{y})$ is constructed by linear interpolation as follows:

$$\begin{aligned} \tilde{\mathbf{x}} &= \lambda_x \mathbf{x}_i + (1-\lambda_x)\mathbf{x}_j, \\ \tilde{y} &= \lambda_y y_i + (1-\lambda_y) y_j, \end{aligned} \tag{17}$$

where $\mathbf{x}_i$ and $\mathbf{x}_j$ are the features of two samples randomly drawn from D, y_i and y_j are the labels corresponding to $\mathbf{x}_i$ and $\mathbf{x}_j$, λ_x and λ_y are regulatory factors for mixing features and labels, respectively. More specifically, each λ_x is sampled from the beta distribution in the same way as Mixup, i.e.,

$$\lambda_x \sim Beta(\alpha, \alpha), \ \alpha > 0, \tag{18}$$

here, α determines the dispersion of the distribution. However, the acquisition of each λ_y varies, and it is relaxed by the following piecewise function so that more of the resultant samples belong to the minority class:

$$\lambda_y = \begin{cases} 0 & \text{for} \quad \lambda_x < \tau, \ y_i = 0, \ y_j = 1 \\ 1 & \text{for} \quad 1 - \lambda_x < \tau, \ y_i = 1, \ y_j = 0 \\ \lambda_x & \text{for} \quad \text{otherwise} \end{cases}, \tag{19}$$

where $\tau \in [0, 0.5]$ is the threshold that determines the strength of preference for the minority class when assigning mixed labels, and 0 and 1 represent the majority and minority classes, respectively. From the above, we can see that $\lambda_x \in (0, 1)$, and $\lambda_y \in [0, 1]$.

Fig. 2 illustrates how IDA changes the marginal space from which the mixed samples are synthesized and how this affects the decision boundaries for classification. In the three diagrams from the left to the right, τ gradually increases from 0 to 0.5, and accordingly, IDA synthesizes more samples that approximate the minority class (light red dots), indicating that the possible marginal space for synthesizing minority samples gradually expands, as indicated by the gradual increase of the red dashed line. Thus, our method eventually moves the classifier's decision boundary to the majority sample space. Algorithm 1 presents the operation process of IDA.

Since the learnability of the datasets to be learned often varies, the hyperparameter τ of IDA needs to be dynamically adjusted during training. There are three main reasons for the variety of learnability: (1) Under certain conditions, the negative impact of class imbalance on the model can be negligible [31]. For example, if both majority and minority classes are well represented and come from non-overlapping distributions, desired learning outcomes can be achieved regardless of the class imbalance. For another example, the sensitivity of learning methods to class imbalance increases with the complexity of the classification problem, while non-complex linear separable problems are not affected by different degrees of class imbalance. However, the extent to which a classification problem satisfies such ideal learning conditions is difficult to determine prior to learning. (2) For large-scale datasets, the total number of minority samples is more important than the degree of class imbalance [31]. For example, a dataset with a minority sample size of only 1% and a total sample size of 1 million, 10,000 minority samples is still large for model training, although the class imbalance is severe. (3) The marginal space between the minority and majority sample spaces is uncertain.

Algorithm 1 IDA.

```
Input: Dataset $D = \{(\mathbf{x}_i, y_i), \ldots, (\mathbf{x}_n, y_n)\}$; Multiplier $r$; Parameter $\alpha$;
    Threshold $\tau$
Output: Augmented dataset $D^{IDA} = \{(\tilde{\mathbf{x}}_i, \tilde{y}_i), \ldots, (\tilde{\mathbf{x}}_m, \tilde{y}_m)\}$, $m = r \times n$
for all $i \in \{1, \ldots, r \times n\}$ do
    Randomly draw sample pair $(\mathbf{x}_i, y_i; \mathbf{x}_j, y_j)$ from $D$
    Sample $\lambda_x$ according to Eq. (18)
    Compute $\lambda_y$ by Eq. (19)
    Mix sample $(\tilde{\mathbf{x}}, \tilde{y})$ by Eq. (17)
    Load $(\tilde{\mathbf{x}}, \tilde{y})$ to $D^{IDA}$
end for
```

The advantage of IDA over conventional rebalancing sampling or data augmentation methods is that it can improve the learning performance of the minority class by synthesizing additional information about the minority class without losing the learning performance of the majority class. To be compatible with the end-to-end learning model, we designed IDA to be plug-and-play. As a result, it can be enabled before the entire training session or at the beginning of each training batch. However, for SHDA-under-FL&CI, we do not recommend using it in batches in mini-batch learning mode due to the particularity of imbalanced data. The main reason is that there are few or no minority samples in each batch, in which case IDA adds more noise than helpful information, and it is conceivable that the model learned from such data is more likely to become worse than better.

4.5. Learning framework

Thanks to the loose coupling property of the proposed model architecture, we can train its classification and representation learning layers

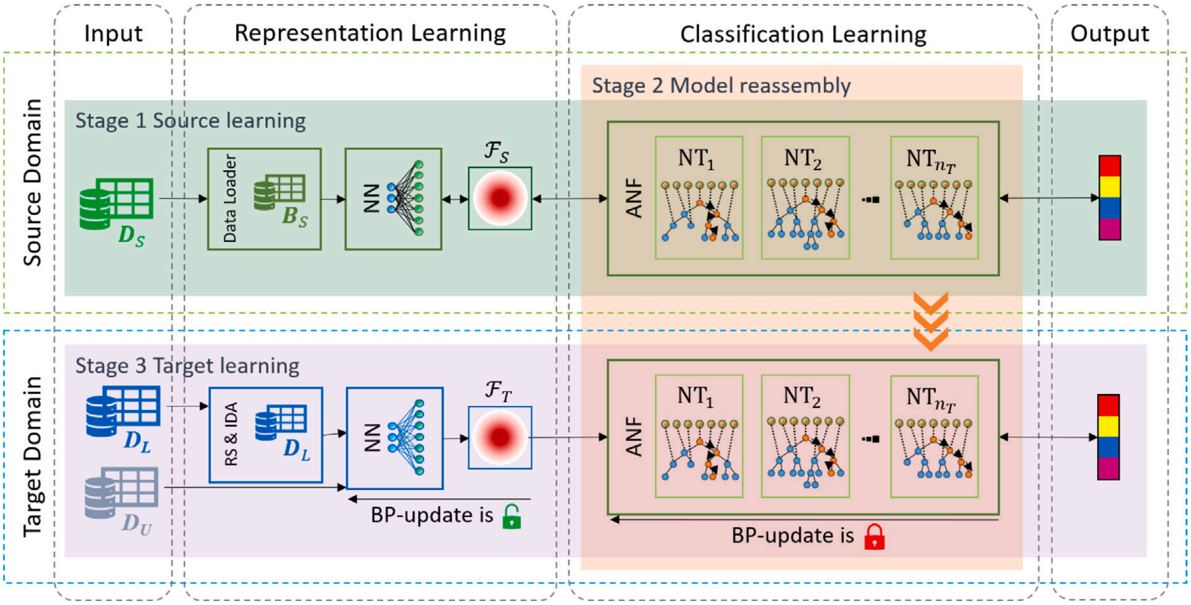

Fig. 3. The learning framework of STANF for SHDA-under-FL&CI.

in a phased manner. The learning framework based on the proposed methods STANF and IDA consists of three main stages: *source learning*, *model reassembly*, and *target learning*. Fig. 3 illustrates the learning stages, data flows, built-in components and their dependencies in the framework.

The learning procedure of STANF can be summarized as follows: (1) In the *source learning*, $\mathcal{F}_S$ and $\mathcal{G}$ are trained on D_S with mini-batch gradient descent based on the proposed CS-loss until convergence conditions are met, or all epochs are iterated. (2) In the *model reassembly*, The model is reassembled for heterogeneous domain adaptation, i.e., $\mathcal{G}$ of the trained source model is reused in the target model, but $\mathcal{F}_S$ is replaced with a new one $\mathcal{F}_T$. (3) In the *target learning*, heterogeneous domain adaptation is achieved by fixing $\mathcal{G}$ of the source model, which means that $\mathcal{G}$ is not updated but only $\mathcal{F}_T$ is trained. D_L is first replaced by a relatively class-balanced dataset synthesized through rebalancing sampling and Algorithm 1, and then $\mathcal{F}_T$ is trained in a semi-supervised manner on augmented D_L and D_U until convergence conditions are satisfied or all epochs are completed. Algorithm 2 is pseudocode for the learning procedure of STANF.

It is worth noting that we use the softer CS-loss in source learning rather than the harder IDA or rebalancing sampling. The main reason is that the total number of minority samples available is more at stake than its ratio or percentage [31]. In the case of relatively abundant source data, the error information introduced by synthesizing minority samples may be more likely to harm the model's performance than the scarcity or under-representation of the minority class. In other words, even a high degree of class imbalance is not the only criterion for adopting IDA or rebalancing sampling. Also, rebalancing sampling and algorithm 1 can be used in various combinations, either by replacing or incorporating the original data with augmented data or by replacing or incorporating the rebalanced samples from the original data with augmented data. including (1) replacing the original data with augmented data, (2) merging the augmented data into the original data, (3) replacing the rebalanced samples drawn from the original data with augmented data, (4) merging the augmented data into the rebalanced samples drawn from the original data, or even more complex combinations.

Algorithm 2 STANF for SHDA-under-FL&CI.

```
Input: Labeled source data $D_S = \{(\mathbf{x}_1^s, y_1^s), \ldots, (\mathbf{x}_{n^s}^s, y_{n^s}^s)\}$; Labeled target
    data $D_L = \{(\mathbf{x}_1^l, y_1^l), \ldots, (\mathbf{x}_{n^l}^l, y_{n^l}^l)\}$; Unlabeled target data $D_U =
    \{\mathbf{x}_1^u, \ldots, \mathbf{x}_{n^u}^u\}$
Output: Classifier $C$ with network weights $\boldsymbol{\Theta}$ and class-label distribution $\pi$
Randomly initialize network weights $\boldsymbol{\Theta}$ and class-label distribution $\pi$
repeat
    Sample $(\mathbf{x}, y)$ from $D_S$
    Update $\pi$ in $\mathcal{G}$ by Eq. (6)
    Update $\boldsymbol{\Theta}$ in $\mathcal{F}_S$ and $\mathcal{G}$ of $C$ by the gradient of Eq. (11)
until $\boldsymbol{\Theta}$ and $\pi$ converge
Sample class-balanced dataset $D_L^{RS}$ from $D_L$
Augment $D_L$ to $D_L^{IDA}$ with Algorithm 1
Replace $D_L$ with a combination of $D_L^{RS}$ and $D_L^{IDA}$
repeat
    Take all $(\mathbf{x}, y)$ in augmented $D_L$ and all $\mathbf{x}$ in $D_U$
    Update $\boldsymbol{\Theta}$ in $\mathcal{F}_T$ of $C$ by the gradient of Eq. (15)
until $\boldsymbol{\Theta}$ converges
```

5. Experiments

This section presents the benchmark datasets, baseline algorithms and experimental setup, model configuration, experimental results, ablation studies, and discussions of the main experimental results.

5.1. Benchmark datasets

The data employed in credit risk classification applications is typically structured (i.e., tabular) [12], which motivated us to select three real-world datasets: German Credit Data (GC), Australian Credit Approval (AC), and Japanese Credit Screening (JC) from the UCI Machine Learning Repository [32], as well as Credit Card (CC) [33] as the benchmark datasets for our experiments. All datasets are suitable for identifying which individuals described by multiple attributes have good or bad credit. The rationale for selecting such datasets is that each is a representative credit risk classification dataset, especially GC, AC, and JC, which are the most popular ones [11,12]. Moreover, these datasets are heterogeneous to each other and are suitable for the problem setting in this study. After data pre-processing, i.e., converting all categorical features into indicator features, the datasets have 61, 42, 46, and 32-dimensional features to feed into the model, respectively. We do not continue with any other data pre-processing. Also, we randomly sample from each dataset but vary the proportion of minority samples to construct imbalanced datasets with different DCIs. The term DCI represents the degree of class imbalance in the datasets, i.e., the ratio of majority to minority samples within them, as defined in Section 3.

5.2. Baselines algorithms

We compare the proposed methods with existing methods to evaluate the performance of our algorithm. Specifically, Support Vector Machines (SVM), Neural Networks (NN), and Random Forests (RF) are non-transfer learning algorithms, and Transfer Neural Trees (TNT) [17] and our STANF are semi-supervised heterogeneous domain adaptation algorithms. The supervised machine learning paradigm dominates existing credit risk classification methods [11,12], and thus, the primary purpose of the comparison experiments is to validate the advantages of STANF employing a transfer learning paradigm over state-of-the-art supervised machine learning methods. In prior studies, ensemble classifiers have significantly outperformed other individual classifiers [11, 12]. Therefore, one of the most popular ensemble learning methods, RF, is chosen as one of the baseline algorithms in the experiments. Two other representative supervised learning methods, NN and SVM, are also among the baseline algorithms. In addition, although few studies have introduced semi-supervised heterogeneous domain adaptation methods into credit risk classification, there is still a need to examine the superiority of STANF over state-of-the-art semi-supervised heterogeneous domain adaptation methods. Considering that data in current credit risk classification applications are usually structured (i.e., tabular), TNT [17], which has a version of semi-supervised heterogeneous domain adaptation applicable to tabular data, is selected as one of the baseline algorithms. In contrast, other semi-supervised heterogeneous domain adaptation methods focusing on unstructured data, such as text and images, are not within the scope of this work.

5.3. Model configuration and experimental setup

The representation learning layer of all neural network-based models has the same backbone. Accordingly, in TNT and our STANF, both $\mathcal{F}_S$ and $\mathcal{F}_T$ use three-layer neural networks to learn nonlinear patterns. They utilize the hyperbolic tangent as the activation function, apply batch normalization, and produce feature maps of dimension 256. The classification learning layer of both models consists of 20 neural trees, each with a depth of 7. Additionally, each tree randomly selects 20% of the feature map. We set the pruning term p_d in Eq. (6) to a fixed value of 0.001, and the regularization parameter λ in Eq. (15) gradually increases from 0 to 1. Also, we set the hyper-parameters of other baseline algorithms, including RF, NN, and SVM, to their default optimal values. For IDA, we set the multiplier r and the parameter α in Eq. (18) and the threshold τ in Eq. (19) to fixed values of 2, 0.1, and 0.5, respectively. The above model configurations and corresponding hyperparameter settings were kept constant throughout the experiments.

We repeat each experiment 10 times to ensure fair comparisons using random sampling.

5.4. Experimental results

We conduct experiments to compare the performance of our STANF with non-transfer learning and semi-supervised heterogeneous domain adaptation baseline methods when learning from few-label and class-imbalanced datasets. Each experiment consists of 9 tasks, which vary depending on the source and target data. For example, the SHDA classification problems with datasets German Credit Data and Australian Credit Approval as source and target data and datasets Japanese Credit Screening and Credit Card as source and target data are different tasks. We use F1 score and Area Under the Receiver Operating Characteristic Curve (AUROC) rather than accuracy as classification metrics to evaluate the model performance in all experimental results. In the case of training models on class-imbalanced datasets, F1 score and AUROC can quantify predictive performance for majority and minority classes more comprehensively than accuracy because they are harmonic averages of precision and recall.

Tables 1 and 2 presents the main experimental results, revealing the following insights:

- The transfer learning algorithms are significantly superior to the non-transfer learning algorithms regarding overall F1 score and AUROC. This suggests that the learning performance in the target domain can be further improved by leveraging the knowledge gained from the relevant source domain, making transfer learning methods more competitive.
- STANF outperforms the baseline methods in overall F1 score and AUROC. Moreover, STANF can achieve up to a 6.14% advantage in F1 score and up to a 7.15% advantage in AUROC compared to transfer learning baseline methods. This demonstrates the effectiveness of STANF in improving the learning performance in the target domain by transferring unbiased knowledge learned from the source data, as well as its significant superiority in regulating the level of attention paid to minority and majority classes in response to imbalanced learning.
- Even in cases where the degree of class imbalance gradually increases, such as the DCI shown in the table increasing from 70 : 30 to 90 : 10, STANF remains steadily superior in handling class-imbalanced data, while the performance of baseline methods declined significantly. This indicates that our approach can steadily transfer unbiased knowledge from imbalanced source data and adapt to imbalanced target data. Notably, STANF achieves this advantage without requiring further fine-tuning of the parameters.
- STANF can train high-performance target models in a simple and direct semi-supervised manner. It thus provides the possibility for further development of learning potential by continuing with other semi-supervised learning methods, such as phased or progressive ones.

5.5. Ablation studies

Ablation studies examine the core design of a system by removing a component to investigate its effect on system function. We conduct ablation experiments to verify whether the proposed cost-sensitive loss and imbalanced data augmentation methods in the proposed semi-supervised heterogeneous domain adaptation approach contribute to imbalanced transfer learning. Taking the experimental results on the CC → AC task as an example, as shown in Table 3, the findings are summarized as follows:

- Compared with the STANF, which does not use any imbalanced learning techniques, the STANF, which uses the cost-sensitive loss technique only in source learning, can help to learn unbiased source knowledge, thereby improving target learning by transferring the learned knowledge.

Table 1
The experimental results for comparing STANF and baselines. ↑ indicates greater values are preferred. The bold represents the best performance.

Task	DCI	Non-Transfer Learning Algorithm						Transfer Learning Algorithm			
		SVM		NN		RF		TNT		STANF (Ours)	
		F1 ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑
GC → AC	70:30	0.2899	0.5796	0.4299	0.6175	0.6634	0.7622	0.7350	0.7952	**0.7882**	**0.8562**
AC → GC		0.0884	0.5153	0.0000	0.5000	0.3014	0.5663	0.6155	0.6198	**0.6494**	**0.6384**
GC → JC		0.1745	0.5473	0.3950	0.5989	0.6435	0.7468	0.7482	0.8165	**0.7756**	**0.8409**
JC → GC		0.0884	0.5153	0.0000	0.5000	0.3014	0.5663	0.6186	0.6244	**0.6537**	**0.6451**
CC → AC		0.2899	0.5796	0.4299	0.6175	0.6634	0.7622	0.7402	0.8270	**0.7946**	**0.8694**
AC → JC		0.1745	0.5473	0.3950	0.5989	0.6435	0.7468	0.7611	0.8269	**0.7903**	**0.8385**
JC → AC		0.2899	0.5796	0.4299	0.6175	0.6634	0.7622	0.7574	0.8318	**0.8133**	**0.8772**
CC → JC		0.1745	0.5473	0.3950	0.5989	0.6435	0.7468	0.7615	0.8242	**0.8087**	**0.8842**
CC → GC		0.0884	0.5153	0.0000	0.5000	0.3014	0.5663	0.6040	0.6162	**0.6207**	**0.6357**
Avg		0.1842	0.5474	0.2750	0.5721	0.5361	0.6918	0.7046	0.7536	**0.7438**	**0.7873**
GC → AC	75:25	0.1164	0.5299	0.3556	0.5944	0.5845	0.7214	0.7750	0.7891	**0.7952**	**0.8097**
AC → GC		0.0226	0.5050	0.0000	0.5000	0.1336	0.5225	0.6617	0.6087	**0.6797**	**0.6217**
GC → JC		0.2673	0.5719	0.3435	0.5908	0.5076	0.6794	0.7665	**0.8102**	**0.7998**	0.8001
JC → GC		0.0226	0.5050	0.0000	0.5000	0.1336	0.5225	0.6625	0.6047	**0.6756**	**0.6177**
CC → AC		0.1164	0.5299	0.3556	0.5944	0.5845	0.7214	0.7604	0.8324	**0.7831**	**0.8396**
AC → JC		0.2673	0.5719	0.3435	0.5908	0.5076	0.6794	**0.7858**	0.8250	0.7841	**0.8356**
JC → AC		0.1164	0.5299	0.3556	0.5944	0.5845	0.7214	0.7692	0.8004	**0.8069**	**0.8719**
CC → JC		0.2673	0.5719	0.3435	0.5908	0.5076	0.6794	0.7408	0.8255	**0.8021**	**0.8681**
CC → GC		0.0226	0.5050	0.0000	0.5000	0.1336	0.5225	0.5918	0.6130	**0.6800**	**0.6767**
Avg		0.1354	0.5356	0.2330	0.5617	0.4086	0.6411	0.7237	0.7454	**0.7563**	**0.7712**

Table 2
The experimental results for comparing STANF and baselines. ↑ indicates greater values are preferred. The bold represents the best performance.

Task	DCI	Non-Transfer Learning Algorithm						Transfer Learning Algorithm			
		SVM		NN		RF		TNT		STANF (Ours)	
		F1 ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑	*F1* ↑	*AUROC* ↑
Avg		0.	0.	0.	0.	0.	0.	0.	0.	**0.**	**0.**
GC → AC	80:20	0.0740	0.5182	0.2611	0.5585	0.4432	0.6598	0.7754	**0.7924**	**0.8163**	0.7841
AC → GC		0.0247	0.5050	0.0000	0.5000	0.1080	0.5190	0.7043	0.5684	**0.7109**	**0.6239**
GC → JC		0.1313	0.5344	0.3470	0.5991	0.5229	0.6959	0.7737	**0.7842**	**0.8038**	0.7081
JC → GC		0.0247	0.5050	0.0000	0.5000	0.1080	0.5190	0.6924	0.5668	**0.7043**	**0.6186**
CC → AC		0.0740	0.5182	0.2611	0.5585	0.4432	0.6598	0.7525	0.8122	**0.7958**	**0.8464**
AC → JC		0.1313	0.5344	0.3470	0.5991	0.5229	0.6959	0.8092	0.8098	**0.8281**	**0.8576**
JC → AC		0.0740	0.5182	0.2611	0.5585	0.4432	0.6598	**0.8150**	0.8341	0.8029	**0.8610**
CC → JC		0.1313	0.5344	0.3470	0.5991	0.5229	0.6959	0.7658	0.7901	**0.8031**	**0.8314**
CC → GC		0.0247	0.5050	0.0000	0.5000	0.1080	0.5190	0.6587	0.5837	**0.7062**	**0.6179**
Avg		0.767	0.5192	0.2027	0.5525	0.3580	0.6249	0.7497	0.7268	**0.7746**	**0.7499**
GC → AC	90:10	0.0242	0.5054	0.1255	0.5389	0.1579	0.5455	0.8392	0.7485	**0.8818**	**0.7486**
AC → GC		0.0000	0.5000	0.0000	0.5000	0.0125	0.5026	0.8141	0.5507	**0.8537**	**0.5669**
GC → JC		0.0622	0.5152	0.1722	0.5606	0.1801	0.5559	0.8304	0.7187	**0.8757**	**0.7676**
JC → GC		0.0000	0.5000	0.0000	0.5000	0.0125	0.5026	0.8114	**0.5630**	**0.8728**	0.5286
CC → AC		0.0242	0.5054	0.1255	0.5389	0.1579	0.5455	0.8273	0.7496	**0.8547**	**0.7560**
AC → JC		0.0622	0.5152	0.1722	0.5606	0.1801	0.5559	0.8623	**0.7650**	**0.8901**	0.7040
JC → AC		0.0242	0.5054	0.1255	0.5389	0.1579	0.5455	0.8428	0.7282	**0.8937**	**0.7699**
CC → JC		0.0622	0.5152	0.1722	0.5606	0.1801	0.5559	0.8425	0.7378	**0.8511**	**0.7851**
CC → GC		0.0000	0.5000	0.0000	0.5000	0.0125	0.5026	0.7993	**0.5813**	**0.8276**	0.5259
Avg		0.288	0.5069	0.1411	0.5332	0.1168	0.5347	0.8299	0.6825	**0.8668**	**0.6836**

- The STANF that uses the rebalancing sampling only in target learning and the one that uses the cost-sensitive loss technique only in source learning can improve imbalanced transfer learning, and the latter can achieve better overall performance than the former in most cases.
- The STANF, using the cost-sensitive loss and rebalancing sampling techniques, achieves the best overall classification performance in the ablation experiment. This confirms that combining the cost-sensitive loss and rebalancing sampling techniques is superior to using one or the other alone.

We also conduct ablation studies focusing on the proposed imbalanced data augmentation method. Table 4 lists the F1 scores in the corresponding experimental results, from which the following findings can be obtained:

Table 3
The experimental results of the ablation study for the cost-sensitive loss and rebalancing sampling techniques ($DCI = 70 : 30$ for all datasets). Naive stands for the STANF without any imbalanced learning techniques, w/ CS represents the STANF with only the cost-sensitive loss, w/ RS represents the STANF with only the rebalancing sampling and w/ CS+RS refers to the STANF with both techniques. ↑ indicates greater values are preferred. The bold represents the best performance.

Task	*Metric*	*Naive*	*w/ CS*	*w/ RS*	*w/ CS+RS*
CC → GC	F1 ↑	0.5993	0.6256	0.6469	**0.6514**

- Compared with the STANF without the proposed IDA, the STANF using IDA can achieve better classification performance. This suggests that IDA can collaborate with the cost-sensitive loss and rebalancing sampling techniques to help further improve model performance.

Table 4
The F1 scores in the experimental results of the ablation study for the imbalanced data augmentation method. w/o IDA represents STANFs that do not use the imbalanced data augmentation method, and w/ IDA represents STANFs that do. ↑ indicates greater values are preferred. The bold represents the best performance.

Task	DCI=70:30		DCI=80:20		DCI=90:10	
	w/o IDA	w/ IDA	w/o IDA	w/ IDA	w/o IDA	w/ IDA
GC → AC	**0.8112**	0.7882	**0.8283**	0.8163	0.8699	**0.8818**
AC → GC	0.6369	**0.6494**	0.6882	**0.7109**	**0.8751**	0.8537
GC → JC	**0.7828**	0.7756	**0.8225**	0.8038	0.8590	**0.8757**
JC → GC	0.6314	**0.6537**	0.6809	**0.7043**	0.8675	**0.8728**
CC → AC	**0.7946**	**0.7946**	0.7853	**0.7958**	0.8299	**0.8547**
AC → JC	0.7872	**0.7903**	0.8125	**0.8281**	0.8704	**0.8901**
JC → AC	0.8058	**0.8133**	0.7864	**0.8029**	0.8706	**0.8937**
CC → JC	0.7723	**0.8087**	0.7897	**0.8031**	0.8208	**0.8511**
CC → GC	**0.6460**	0.6207	0.6877	**0.7062**	0.8110	**0.8276**
Avg	0.7409	**0.7438**	0.7646	**0.7746**	0.8527	**0.8668**

- When the class imbalance of a dataset is not severe, as shown by DCI=70 : 30 in the table, the model performance improvement due to IDA is not as significant. This result is intuitive because, in cases where minority samples are less scarce, IDA may use convex combinations that are too biased towards minority classes, introducing unnecessary noise. Using such data perturbations to regularize training is more likely to compromise model performance.

5.6. Discussion

5.6.1. Semi-supervised heterogeneous domain adaptation capability

The experiment has a total of 36 tasks, and the experimental results shown in Tables 1 and 2 demonstrate the effectiveness of STANF in associating and recognizing cross-domain data. We first validate the effectiveness of STANF in transferring knowledge from different source domains to the same target domain. The results show that STANF can reach knowledge transfer at the architectural level by reusing the classification learning layer of the model learned from the source domain. The adaptive neural forest structure in the classification learning layer based on the random routing mechanism can capture and maintain the structural consistency between labeled data across domains. Furthermore, random pruning further allows the classification learning layer to identify and strengthen neurons that represent related cross-domain data, leading to more refined knowledge transfer. In other words, the classification learning layer ignores leaf nodes with insufficient adaptation ability, thereby successfully associating cross-domain heterogeneous data and improving recognition performance. In addition, the embedding loss enables STANF to learn new representation learning layers for the model from the source domain in a semi-supervised manner, further alleviating the reliance on labeling information.

Why can Eq. (13) be used as a consistency regularization in semi-supervised learning? According to Eq. (13), for the same sample, the more inconsistent the prediction given by each individual tree is with that given by the whole forest, the greater the embedding loss. Minimizing Eq. (13) is equivalent to maximizing the structural consistency between the output $\mathbb{P}_{\mathbf{T}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]$ of each individual tree and the output $\mathbb{P}_{\mathbf{F}}[\tilde{y}|\mathbf{x}, \boldsymbol{\Theta}, \boldsymbol{\pi}]$ of the entire forest. In other words, adding Eq. (13) to the loss function is applying a consistency constraint to force unlabeled data D_U to traverse similar paths along the tree as labeled data D_L. Note that such a constraint relies on a Random Forests-like bagged ensemble model and multiple Decision Tree-like inference hierarchies to take full advantage of the model's convergence of individual- and population-level predictions for the same sample. Therefore, we refer to Eq. (13) as the *predictive and structural self-consistency regularization* (PSSCR) term of the loss function, which can be used as a consistency regularization technique for semi-supervised bagged ensemble models. Intuitively, the constraint effect produced by PSSCR is filtering out useless or harmful information extracted from unlabeled data.

5.6.2. Robustness

The experimental results show that the advantage of STANF is stable in the case of transferring knowledge from different source domains to the same target domain. For example, the STANF model performs best regardless of the semi-supervised heterogeneous domain adaptation from GC to AC, JC to AC, or CC to AC, as shown in Tables 1 and 2. In addition, STANF can reliably learn from datasets with different degrees of category imbalance (both in the target and source domains). As shown in Tables 1 and 2, the 36 tasks in the experiments can be categorized into three groups using the degree of class imbalance as the grouping criterion. Regardless of the degree of class imbalance, the TANF model consistently maintains the performance advantage.

All IDA parameters are set manually and empirically tuned on the datasets during training. Such a search process can be called the empirical optimal method, which usually yields relatively optimal parameters. Then, the computational complexity of the empirical optimal method can be simply expressed as $\mathcal{O}_{SOP} = N_{ITER}\ \mathcal{O}_{STANF}$, where $\mathcal{O}_{SOP}$ denotes the computational complexity of searching the optimal parameters of IDA, N_{ITER} refers to the number of search iterations, and $\mathcal{O}_{STANF}$ denotes the computational complexity of STANF. Thanks to the robustness of STANF, the value of N_{ITER} is generally manageable, and in this work, $N_{ITER} < 10$ is guaranteed. According to the empirical results, the fewer samples used to train the model, the higher the risk that IDA introduces harmful noise. Therefore, in a few-sample CRC problem, it is recommended to adopt such an optimal parameter search strategy for IDA: the optimal value of parameter r is searched from small to large; on the contrary, the optimal values of parameters α and τ are searched from large to small. This strategy can somewhat avoid introducing the hazardous noise that misleads the model to learn a false sample space.

Note that the competitive experimental results, as shown in Tables 1 and 2, are achieved despite using only empirically optimal values of the IDA parameters obtained after a few search iterations. The results demonstrate, to some extent, the robustness of STANF. In other words, with more tuning effort, STANF could potentially achieve a more significant advantage over the baseline algorithms.

5.6.3. Potential challenges in practical applications

Potential challenges in applying the proposed approach in real-world credit risk classification scenarios are of concern. Some of the typical aspects are listed below.

Considering that the benchmark datasets in this work may not be representative of all credit risk classification datasets, the generalizability of the proposed method to industry-specific datasets, especially large-scale datasets, needs to be re-examined thoroughly. In particular, financial data is usually multi-sourced and thus prone to carry defects, such as missing values and dirty data. Accordingly, selecting appropriate data pre-processing before training the STANF model must be analyzed case-by-case to ensure the model works correctly. In addition, although existing credit risk classification applications mainly use structured data [12], attempts to further improve the performance of credit risk prediction through unstructured data like text and images have been unceasing. Therefore, the scalability of the proposed method to unstructured datasets is also a challenge.

Continuous data growth is ubiquitous in credit risk classification applications, and it is the main trigger for data drift and concept drift that degrade model performance. These issues lead to an increase in the frequency of model updates, which raises the demand for the method in terms of computational resources and training time. Therefore, it is challenging for financial institutions with limited infrastructure to implement the proposed method in credit risk classification applications. In addition, data security and privacy protection, which are crucial in credit risk classification applications, may hinder data access, thus affecting the proposed method's applicability.

6. Conclusion and future work

This paper proposes a semi-supervised heterogeneous domain adaptation method called STANF for few-sample credit risk classification under few-label and class-imbalanced regimes. The paper also suggests an imbalanced data augmentation method called IDA to enhance STANF further. A brief review of the literature on the techniques involved in credit risk classification and their limitations is also presented. The primary innovation of STANF is that it adopts semi-supervised heterogeneous domain adaptation to solve few-sample credit risk classification problems. It also improves learning from class-imbalanced domains through cost-sensitive loss and IDA. Four real-world credit risk classification datasets are used to validate the effectiveness of the proposed method. Different performance metrics, such as F1-score and AUROC, are used to evaluate the performance of the proposed method. Experiments, including 36 tasks, are conducted to compare STANF with the baseline algorithms. The experimental results show that STANF outperforms existing credit risk classification methods regarding F1-score and AUROC. The significance of this work is that STANF can effectively mitigate the dependence on labeling information and the harm caused by class imbalance.

In the future, we will investigate combining STANF with other semi-supervised learning methods to exploit the potential of unlabeled data further. We will also investigate the superiority of STANF over other semi-supervised heterogeneous domain adaptation methods applicable to unstructured data such as text and images. In addition, considering the different learning conditions of various datasets, giving a universal convex combination suitable for all datasets for the proposed IDA is difficult. While IDA can improve model performance without much effort in tuning parameters, it usually requires a search process to get its optimal parameters. Therefore, we will also explore transforming the parameters of IDA into trainable ones to learn the optimal convex combinations for the training samples. Compared to the manual search method for optimal IDA parameters used in this study, the new optimization approach may reduce the computational complexity and increase the robustness of the proposed method. On the other hand, the new optimization approach may pose the challenge of putting more effort into giving interpretability for data augmentation.

CRediT authorship contribution statement

Zhaoqing Liu: Writing – original draft, Software, Methodology, Data curation, Conceptualization. **Guangquan Zhang:** Supervision. **Jie Lu:** Writing – review & editing, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgment

The work presented in this paper was supported by the Australian Research Council (ARC) under Discovery Project DP220102635.

References

[1] B. Engelmann, A simple and consistent credit risk model for basel II/III, IFRS 9 and stress testing when loan data history is short, 2021.
[2] M. Leo, S. Sharma, K. Maddulety, Machine learning in banking risk management: A literature review, Risks 7 (1) (2019) 29.
[3] S. Bhatore, S. Lalit Mohan, Y.R. Reddy, Machine learning techniques for credit risk evaluation: a systematic literature review, J. Bank. Financial Technol. 4 (1) (2020) 111–138.
[4] Z. Liu, A. Liu, G. Zhang, J. Lu, An empirical study of fuzzy decision tree for gradient boosting ensemble, in: AI 2021: Advances in Artificial Intelligence - 34th Australasian Joint Conference, in: Lecture Notes in Computer Science, Vol. 13151, Springer, 2022, pp. 716–727.
[5] J. Lu, L. Niu, G. Zhang, A situation retrieval model for cognitive decision support in digital business ecosystems, IEEE Trans. Ind. Electron. 60 (3) (2013) 1059–1069.
[6] J. Lu, X. Yang, G. Zhang, Support vector machine-based multi-source multi-attribute information integration for situation assessment, Expert Syst. Appl. 34 (2) (2008) 1333–1340.
[7] H. Suryanto, A. Mahidadia, M. Bain, C. Guan, A. Guan, Credit risk modeling using transfer learning and domain adaptation, Front. Artif. Intell. 5 (2022) 868232.
[8] J. Lu, H. Zuo, G. Zhang, Fuzzy multiple-source transfer learning, IEEE Trans. Fuzzy Syst. 28 (12) (2020) 3418–3431.
[9] S. Shi, R. Tse, W. Luo, S. D'Addona, G. Pau, Machine learning-driven credit risk: a systemic review, Neural Comput. Appl. 34 (17) (2022) 14327–14339.
[10] M.Z. Abedin, C. Guotai, P. Hajek, T. Zhang, Combining weighted SMOTE with ensemble learning for the class-imbalanced prediction of small business credit risk, Complex Intell. Syst. 9 (4) (2022) 3559–3579.
[11] A. Bhattacharya, S.K. Biswas, A. Mandal, Credit risk evaluation: a comprehensive study, Multimedia Tools Appl. 82 (12) (2023) 18217–18267.
[12] X. Zhang, L. Yu, Consumer credit risk assessment: A review from the state-of-the-art classification algorithms, data traits, and learning methods, Expert Syst. Appl. 237 (Part B) (2023) 121484.
[13] Z. Fang, J. Lu, F. Liu, J. Xuan, G. Zhang, Open set domain adaptation: Theoretical bound and algorithm, IEEE Trans. Neural Netw. Learn. Syst. 32 (10) (2021) 4309–4322.
[14] Z. Fang, J. Lu, F. Liu, G. Zhang, Semi-supervised heterogeneous domain adaptation: Theory and algorithms, IEEE Trans. Pattern Anal. Mach. Intell. 45 (1) (2023) 1087–1105.
[15] Y.H. Tsai, Y. Yeh, Y.F. Wang, Learning cross-domain landmarks for heterogeneous domain adaptation, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2016, pp. 5081–5090.
[16] Y. Hsieh, S. Tao, Y.H. Tsai, Y. Yeh, Y.F. Wang, Recognizing heterogeneous cross-domain data via generalized joint distribution adaptation, in: IEEE International Conference on Multimedia and Expo, IEEE Computer Society, 2016, pp. 1–6.
[17] W. Chen, T.H. Hsu, Y.H. Tsai, M. Chen, Y.F. Wang, Transfer neural trees: Semi-supervised heterogeneous domain adaptation and beyond, IEEE Trans. Image Process. 28 (9) (2019) 4620–4633.
[18] K. Cao, C. Wei, A. Gaidon, N. Aréchiga, T. Ma, Learning imbalanced datasets with label-distribution-aware margin loss, in: Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, 2019, pp. 1565–1576.
[19] B. Kang, S. Xie, M. Rohrbach, Z. Yan, A. Gordo, J. Feng, Y. Kalantidis, Decoupling representation and classifier for long-tailed recognition, in: 8th International Conference on Learning Representations, 2020.
[20] H. Chou, S. Chang, J. Pan, W. Wei, D. Juan, Remix: Rebalanced mixup, in: Computer Vision - ECCV 2020 Workshops, in: Lecture Notes in Computer Science, Vol. 12540, Springer, 2020, pp. 95–110.
[21] A. Galdran, G. Carneiro, M.Á.G. Ballester, Balanced-mixup for highly imbalanced medical image classification, in: Medical Image Computing and Computer Assisted Intervention - MICCAI 2021 - 24th International Conference, in: Lecture Notes in Computer Science, Vol. 12905, Springer, 2021, pp. 323–333.

[22] Z. Xu, Z. Chai, C. Yuan, Towards calibrated model for long-tailed visual recognition from prior perspective, in: Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, 2021, pp. 7139–7152.
[23] H. Zhang, M. Cissé, Y.N. Dauphin, D. Lopez-Paz, Mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, 2018.
[24] L. Carratino, M. Cissé, R. Jenatton, J. Vert, On mixup regularization, J. Mach. Learn. Res. 23 (2022) 325:1–325:31.
[25] J. Lu, J. Xuan, G. Zhang, X. Luo, Structural property-aware multilayer network embedding for latent factor analysis, Pattern Recognit. 76 (2018) 228–241.
[26] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, G. Zhang, Transfer learning using computational intelligence: A survey, Knowl.-Based Syst. 80 (2015) 14–23.
[27] X. Li, W. Chen, Q. Zhang, L. Wu, Building auto-encoder intrusion detection system based on random forest feature selection, Comput. Secur. 95 (2020) 101851.
[28] P. Kontschieder, M. Fiterau, A. Criminisi, S.R. Bulò, Deep neural decision forests, in: S. Kambhampati (Ed.), Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI/AAAI Press, 2016, pp. 4190–4194.
[29] K. Li, B. Wang, Y. Tian, Z. Qi, Fast and accurate road crack detection based on adaptive cost-sensitive loss function, IEEE Trans. Cybern. 53 (2) (2023) 1051–1062.
[30] H. Guo, Y. Mao, R. Zhang, Mixup as locally linear out-of-manifold regularization, in: The Thirty-Third AAAI Conference on Artificial Intelligence, Vol. 33, (01) AAAI Press, 2019, pp. 3714–3722.
[31] J.M. Johnson, T.M. Khoshgoftaar, Survey on deep learning with class imbalance, J. Big Data 6 (1) (2019) 1–54.
[32] D. Dua, C. Graff, UCI machine learning repository, 2017.
[33] H. William, et al., Econometric analysis fifth edition, 2003.

Zhaoqing Liu is pursuing his Ph.D. with the Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. His research interests are in machine learning-enhanced decision support systems and transfer learning.

Guangquan Zhang is an Australian Research Council (ARC) QEII Fellow, Associate Professor and the Director of the Decision Systems and e-Service Intelligent (DeSI) Research Laboratory at the Australian Artificial Intelligence Institute, University of Technology Sydney, Australia. He received his Ph.D. in applied mathematics from Curtin University, Australia, in 2001. From 1993 to 1997, he was a full Professor in the Department of Mathematics, Hebei University, China. His main research interests lie in fuzzy multi-objective, bilevel and group decision making, fuzzy measures, transfer learning and concept drift adaptation. He has published six authored monographs and over 500 papers including some 300 articles in leading international journals. He has supervised 40 Ph.D. students to completion and mentored 15 Postdoc fellows. Prof Zhang has won ten very competitive ARC Discovery grants and many other research projects. His research has been widely applied in industries.

Jie Lu (F'18) is an Australian Laureate Fellow, IFSA Fellow, ACS Fellow, Distinguished Professor, and the Director of Australian Artificial Intelligence Institute (AAII) at the University of Technology Sydney, Australia. She received a Ph.D. degree from Curtin University in 2000. Her main research expertise is in transfer learning, concept drift, fuzzy systems, decision support systems and recommender systems. She has published over 500 papers in IEEE Transactions and other leading journals and conferences. She is the recipient of two IEEE Transactions on Fuzzy Systems Outstanding Paper Awards (2019 and 2022), NeurIPS2022 Outstanding Paper Award, Australia's Most Innovative Engineer Award (2019), Australasian Artificial Intelligence Distinguished Research Contribution Award (2022), Australian NSW Premier's Prize on Excellence in Engineering or Information & Communication Technology (2023), and the Officer of the Order of Australia (AO) 2023.