Unsupervised Conditional Adversarial Network for Tax Evasion Detection

Rongzhe Wei
Qian Xuesen College, School of Mathematics and Statisitcs
Xi'an Jiaotong University
Xi'an, China
jessonwrz@163.com

Bo Dong
National Engineering Lab of Big Data Analytics,
College of Distance Education
Xi'an Jiaotong University
Xi'an, China
dong.bo@mail.xjtu.edu.cn

Xulyu Zhu
SPKLSTN Lab
Xi'an Jiaotong University
Xi'an, China
zhuxvlv@163.com

Jianfei Ruan
SPKLSTN Lab
Xi'an Jiaotong University
Xi'an, China
xjtu_jfruan@163.com

Qinghua Zheng
SPKLSTN Lab
Xi'an Jiaotong University
Xi'an, China
qhzheng@mail.xjtu.edu.cn

Huan He
SPKLSTN Lab
Xi'an Jiaotong University
Xi'an, China
hehuan@mail.xjtu.edu.cn

***Abstract*—The identification of tax evasion plays an important role in ensuring tax order, promoting the level of tax collection and management, and reducing tax losses. With the advancements in data mining technology, many machine learning techniques have yielded results in identifying tax evasion. However, to realize satisfactory performance, these models require large amounts of human annotated data. In the tax field, unlabeled tax data are abundant, data annotation in a single region is expensive, and the distributions of characteristics differ among regions; these factors pose substantial difficulties in the development of an identification model. Existing tax evasion detection methods are either trained for single-region tasks, in which case they perform poorly on inter-region tax evasion identification due to the discrepancies in feature distributions, or utilize labeled data from both the target-task field and different but related auxiliary fields to reuse and transfer knowledge of the target domain data, in which case they cannot deal with scenarios in which there are no labeled data in target audit tasks. Although current unsupervised transfer learning techniques can train models in labeled regions for unlabeled regions, large intra-class distribution discrepancies cannot be perfectly minimized in tax evasion detection scenarios. To better address the above challenges, this paper proposes a general architecture, namely, the unsupervised conditional adversarial network (UCAN) for tax evasion detection, which, to the best of our knowledge, is the first approach to solve audit tasks in unlabeled target domains via inter-region transfer. Our architecture establishes an adversarial neural network and we add the label information in the distribution adapter, which can granularly adapt the joint probability distribution (JPD) of the data. We introduce a constraint that is based on the conditional maximum mean discrepancy (CMMD) of the extracted features to align the conditional probability distribution (CPD) of the deep representation. Our model is formed by combining the distribution adapter and the label predictor to realize end-to-end learning of unsupervised feature transfer. The experimental results demonstrate the outstanding performance of our model in all migration tasks compared with state-of-the-art approaches.**

***Index Terms*—Unsupervised Transductive Transfer Learning, Tax Evasion Detection, Distribution Adaptation, Adversarial Networks**

I. Introduction

Tax evasion detection is a comprehensive inspection of taxpayers' tax payments by tax authorities to ensure tax revenue and to investigate tax violations. Recently, with the rapid advancements in computer science and Internet technology, the combination of tax collection management and information technology has become a development trend in the era of big data. Under this scenario, the identification of tax evasion, which is necessary for ensuring the comprehensiveness and fairness of tax collection, for maintaining tax order, and for preventing tax losses, has been attracting extensive interest. Due to the development of data mining technology, there has been a tremendous accumulation of tax data over time and many machine learning approaches have yielded important results on tax evasion detection tasks [2]. However, to realize satisfactory performance, these methods require large amounts of labeled data. In the tax field, since a large amount of data lack annotations, the cost of data labeling in a single region is expensive, and the distributions of features in vary substantially among regions, the establishment of an identification model that can both effectively utilize the annotated tax-related data in a single region and transfer to other regions with unlabeled tax data remains a pressing issue.

Currently, tax evasion detection is conducted mainly via traditional detection methods and data-mining-based detection methods. Traditional detection methods, which consist of manual case selection, whistle-blowing-based selection, and computer-based case selection methods, are either time- and labor-consuming or are typically based on simple rules and judgment conditions, which fail to perform well in tax evasion identification as tax data accumulate at a tremendous speed and tax evasion patterns change constantly. In contrast, data-mining-based detection methods adopt state-of-the-art data-mining techniques, utilize existing annotated data for model training and establish identification models, which can overcome the disadvantages of simple judgment conditions in computer detection methods. However, challenges are still encountered with data-mining-based detection methods: First, due to the differences in economic policies and patterns among regions, it is difficult to establish a unified identification model. Second, tax evasion patterns are of various types and a tax evasion behavior recognition model cannot be used directly to identify another tax evasion behavior. Last, in the tax field, a huge amount of data lack annotation and data labeling is labor-intensive. Thus, overcoming these difficulties remains an urgent problem to be solved.

Unsupervised transductive transfer learning (or unsupervised domain adaptation) [23] aims to establish a transfer predictive model where the source region is given with labeled data, while the related target domain with no labeled data. When it comes to tax evasion detection, there are several challenges to be facing. No existing studies on unsupervised domain adaptation have been applied to tax evasion detection. Furthermore, current methods carry on marginal distribution adaptation and cannot effectively align intra-class distributions in tax evasion detection scenarios.

For addressing the above challenges, this paper proposes a general architecture, namely, the unsupervised conditional adversarial network (UCAN) for tax evasion detection, which uses labeled data of other audit tasks to assist target label-sparse audit tasks to establish an audit model and can minimize intraclass distribution discrepancies. Our architecture establishes an adversarial neural network for migration tasks and its feature-extractor part is used to capture the deep representation of the features. We add the label information in the domain adapter to granularly adapt the JPD of the data. We use CMMD adaption to align CPDs of the deep representation. To realize end-to-end learning of unsupervised feature transfer, our architecture is formed by combining the distribution adapter and the label predictor.

The main contributions of this paper can be summarized as follows:

- **We establish a general architecture that is based on unsupervised adversarial learning in tax evasion detection scenarios. Our architecture can be trained in a region with annotated labels and applied to completely unlabeled regions for audit tasks. Via adversarial learning, our architecture better captures deep representations of features.**
- **We propose a novel method for inter-region tax evasion detection in which label information is added to the domain classifier of the adversarial neural network for JPD adaption. In unsupervised learning conditions, the source and target regions contain labeled and unlabeled data, respectively, and we use true labels for source region input data and pseudo labels for target region input data to more granularly adapt the JPD of the data.**
- **We provide a unified framework that uses a CMMD criterion on extracted features to align the CPD. CMMD measures the distribution distance between samples in a class and by minimizing CMMD, our model can shorten the intraclass distribution distance**

Our proposed model is trained on real tax dataset for two provinces in China. Comprehensive experimental results demonstrate that our proposed architecture outperformed the state-of-the-art approaches in all thirteen transfer scenarios in five regions in terms of both the inter-region tax evasion detection accuracy and the interpretability. Furthermore, since UCAN is a general architecture, it can be applied not only to tax evasion detection migration tasks but also to other migration tasks.

The remainder of this paper is structured as follows: Section 2 summarizes the related work regarding current tax evasion detection and transfer learning methods. In Section 3, we introduce our identification architecture. Thereafter, we describe our tax dataset, experiments, and results in Section 4. Section 5 presents the conclusions of our work.

II. Related Work

In this section, we briefly review the related work regarding current tax evasion detection and transfer learning methods.

A. *Tax Evasion Detection Methods*

Currently, main tax evasion detection methods are either traditional detection or data-mining based detection methods. Traditional detection methods include manual case selection, whistle-blowing-based selection, and computer-based case selection methods [1], which are either time-and-labor consuming or based on simple detection rules, which cannot be applied to complicated tasks.

Data-mining based methods that learn an automatic architecture from existing tax evasion data and without expert experience are considered the most promising approach to be used by tax administrations for detecting tax evasion. Current studies on data-mining based methods include association analysis [4] [5], cluster analysis [6], genetic algorithm [10], classification [8], reinforcement learning [11] [12], and simulation [13] [14]. Wu et al. [4] applied association rules on tax database to enhance the performance and productivity of value-added tax evasion detection. Matos et al. [5] proposed a method to detect frequent fraud patterns by using association rules and rank the taxpayers according to their potential for fraud. Assylbekov et al. [7] applied SOM algorithm to identify potential tax fraud anomalies. Xia et al. [6] proposed an approach that combines the support vector machine (SVM) and self-organizing feature map (SOM) neural network for tax evasion detection, which uses SVM to classify the taxpayers and SOM to cluster suspect information and selects the potential suspects for detection. Zhu et al. [2] proposed IRTED-TL, an effective and explainable transfer architecture for label-sparse tasks that integrates transfer adaboost (TrAdaBoost) [15], transfer component analysis (TCA) [16], and lightGBM [17].

However, since data-labeling is a really time-consuming task in tax field, and few regions have enough labeled data. The aforementioned approaches are either trained for single-region tax evasion detection, which performs poorly when it comes to migration tasks owing to discrepancies of feature distributions cross-regions, or require the target region in a migration task to have a few annotated data.

B. Unsupervised Domain Adaptation

Domain adaptation (DA), or transfer learning, utilizes labeled data in one or more source regions to assist tasks in a target domain [23] [25]. There are different settings of domain adaptation, and in this paper, we focus on unsupervised domain adaptation, which deals with the migration tasks that the labeling information from the target domain is not available and commonly assumes the existence of a domain invariant feature space [24]. So far, numerous DA approaches which can take in unsupervised domain adaptation settings have been proposed [22]. Pan et al. [16] proposed TCA for domain adaption, which uses maximum mean discrepancy (MMD) to measure the distance between the source and target domains in a reducing kernel Hilbert Space (RKHS). Long et al. [20] proposed joint distribution adaption (JDA), which can adapt the marginal probability distribution (MPD) and JPD simultaneously to minimize multi distribution distance. With the advent of generative adversarial nets (GAN), new architectures embedded with adversarial learning come into being. Yaroslav Ganin et al. [21] proposed domain adversarial neural network (DANN), which is the earliest model to embed adversarial mechanism into neural network. More recently, Long et al. [26] proposed joint adaptation networks (JAN) based on joint maximum mean discrepancy (JMMD) criterion to adapt JPD across domains.

In unsupervised domain adaptation, the aforementioned DA methods with adversarial learning technics outperform other architectures in most transfer learning scenarios and are consider to be the most promising methods. However, these adversarial learning methods fail to take into account the label information of data, which may lead to the fact that although the MPDs are aligned in two domains, their CPDs differs greatly. Existing DA methods that can adapt conditional distributions cannot effectively align intra-class distribution differences in tax evasion detection scenarios. For these purposes, the proposed UCAN architecture adds label information in the domain adapter of adversarial neural network and uses CMMD criterion to adapt the marginal, joint and CPDs simultaneously between two domains.

III. Proposed Architecture

In this paper, we proposed an architecture, namely, UCAN, that comprehensively addresses the following challenges in tax evasion detection: (i) establishing an efficient transfer identification model for tax evasion when there are no labeled data in the target region and (ii) adapting both the marginal and conditional probability distributions of tax data simultaneously between two domains. Our UCAN method is based on unsupervised conditions and utilizes an adversarial transfer learning technique to extract deep tax-evasion-related knowledge from regions with sufficient annotated and unlabeled data. Since most current adversarial transfer learning networks focus on aligning the marginal probability distributions, which may lead to underalignment of the CPDs between domains (in Fig.1, the DANN model is considered as an example). Our method adds the true and pseudo labels from the source and target domains, respectively, to the domain adaptor and imposes a CMMD constraint on the extracted features to reduce the intra-class distribution discrepancies to address the problem.

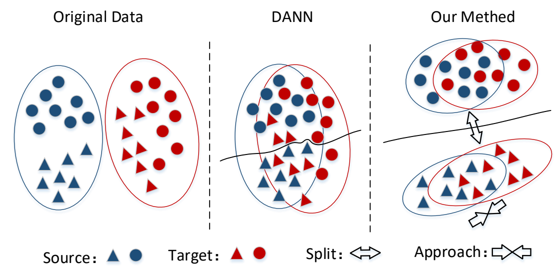

Fig. 1. DANN's shortcoming and UCAN's improvement strategy. Left: There is a data set offset problem between the source domain and the target domain. Middle: The source domain data and the target domain data are adapted at the domain level. The category label of the sample is ignored, thereby resulting in the incorrect adaption between the sample of a class and the sample of another class. Right: If the category label is added, the distance between classes can be increased and similar samples can be more effectively adapted.

A. Framework of UCAN

UCAN is an end-to-end deep network model. An overview illustration of the whole architecture is shown in Fig.2, which is composed of four parts: a feature extractor $G = f_g(x;\theta_g)$, a label predictor $C = f_c(x;\theta_c)$, a domain classifier $D = f_d(x;\theta_d)$, and a distribution adapter M.

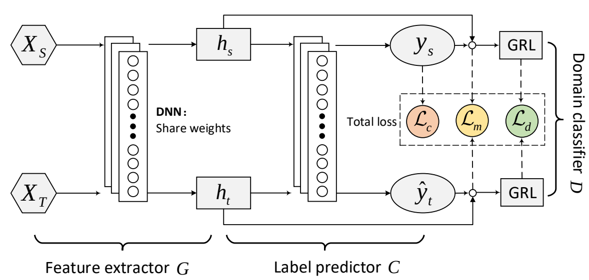

Fig. 2. UCAN architecture. X_S and X_T represent the source and target domain data, respectively. By sharing weights in the feature extractor G, we obtain feature representations h_s and h_t. Then, the label predictor C outputs the predicted labels. In the architecture, y_s stands for the true label in the source region while $\hat{y}_t$ is the pseudo label in the target region. $\mathcal{L}_c$, $\mathcal{L}_m$, and $\mathcal{L}_d$ denote the label predictor loss, distribution adaption loss, and domain classifier loss, respectively. GRL refers to the gradient reverse layer.

The feature extractor G aligns the data distributions of the source and target domains and generates domain-invariant features $h = f_g(x)$. This process of feature generation is realized by a deep neural network (DNN). The input to the domain classifier D is the combination of feature h and label y, which are combined as $h' = h \circ y$, where $\circ$ denotes the combination operation, namely, the concatenation operation or the Kronecker product. The UCAN architecture that uses the concatenation operation for combination is denoted as UCAN-C and the architecture that uses the Kronecker product is denoted as UCAN-K. The distribution adapter M adapts the CPD distance between the source and target domains by using a CMMD constraint and minimizing $P\left(\mathrm{x}_s|y_s=c\right)$ and $P\left(\mathrm{x}_t|y_t=c\right)$.

B. Domain Classifier

In previous adversarial learning methods, the domain classifier uses only a generated feature h to identify the source of the samples, which do not exhibit high interclass discrimination since JPD still has large differences. In this paper, a domain classifier D for joint adaptation of features and labels is proposed and label information is added to feature h, which can adapt the JPDs of the source and target domains. However, in unsupervised transfer learning problems, it is impossible to directly obtain the labels of the target domain to adapt the JPDs $P\left(\mathrm{x}_s, y_s\right)$, and $P\left(\mathrm{x}_t, y_t\right)$. In this case, we use pseudo labels $\hat{y}_t$ in the neural network to estimate the JPD of the target domain:

$$\hat{y}_t^j=\begin{cases} 1 & \text{if } j=\arg\max f_c(f_g(x)) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

where $f_c\left(f_g\left(x_t\right)\right) \in \mathbb{R}^{|C|}$ is a one-dimensional matrix, $|C|$ is the number of categories in label set $\mathcal{Y}$, and $\hat{y}_t$ is the category that corresponds to the largest value in the prediction vector.

After the label information has been obtained, UCAN combines feature h and label information y to form a joint feature $h'=h\circ y$. Two approaches for this combination are designed:
(I) Vector Concatenation: the label information is spliced directly behind the feature. (UCAN-C model)

$$h'=[h,y] \tag{2}$$

(II) Kronecker Product: each element in h is multiplied by each element in y to form a new element. The vector of the $|h|\times|y|$ dimension is finally formed. (UCAN-K model)

$$h'=h\otimes y \tag{3}$$

Finally, the optimization objective of domain classifier is defined as:

$$\begin{aligned}\min_{\theta_d}\max_{\theta_g} L_d\left(f_d, f_g\right)=&-\mathbb{E}_{(x,y)\sim\mathbb{P}_s(X^s,Y^s)}\log\left[f_d\left(f_g(x)\circ y\right)\right]\\ &-\mathbb{E}_{(x,y)\sim\hat{\mathbb{P}}_t(X^t,Y^t)}\log\left[1-f_d\left(f_g(x)\circ y\right)\right]\end{aligned} \tag{4}$$

C. Distribution Adapter

The distribution adapter is the $\mathcal{L}_m$ module in the architecture and uses the CMMD distance to adapt the CPDs of the source target domains. CMMD is derived from MMD, which measures the distance between MPDs of data in the source and target domains and is formulated as:

$$D\left(P\left(\phi(X_S)\right), P\left(\phi(X_T)\right)\right)=\left\|\frac{1}{n_s}\sum_{i=1}^{n_s}\mathbf{A}^{\mathrm{T}}\mathbf{x}_i-\frac{1}{n_t}\sum_{j=n_s+1}^{n_s+n_t}\mathbf{A}^{\mathrm{T}}\mathbf{x}_j\right\|^2 \tag{5}$$

As for CMMD, it first divides the data into different parts according to the true label of source domain and pseudo label of target domain, and then calculates the MMD distance for samples in the same category and combines the MMD distance on all categories to get CMMD:

$$D\left(P\left(\mathrm{x}_s|y_s\right), P\left(\mathrm{x}_t|y_t\right)\right) \tag{6}$$

$$=\sum_{c=1}^{|C|}\left\|\frac{1}{n_s^{(c)}}\sum_{\mathbf{x}_i\in D_s^{(c)}}\mathbf{A}^{\mathrm{T}}\mathbf{x}_i-\frac{1}{n_t^{(c)}}\sum_{\mathbf{x}_j\in D_t^{(c)}}\mathbf{A}^{\mathrm{T}}\mathbf{x}_j\right\|^2 \tag{7}$$

Where, $\mathbf{A}$ is the mapping matrix and $|C|$ is the number of categories.

CMMD measures the distribution distance of intra-class samples and this distance is reduced by minimizing CMMD to align the distributions of the same-category samples in the source and target domains. The optimization objective of the distribution adapter can be defined as:

$$\min_{\theta_g}\mathcal{L}_m=D\left(P\left(\mathrm{x}_s|y_s\right), P\left(\mathrm{x}_t|y_t\right)\right) \tag{8}$$

$$=\sum_{c=1}^{|C|}\left\|\frac{1}{n_s^{(c)}}\sum_{\mathbf{x}_i\in D_s^{(c)}}\mathbf{A}^{\mathrm{T}}\mathbf{x}_i-\frac{1}{n_t^{(c)}}\sum_{\mathbf{x}_j\in D_t^{(c)}}\mathbf{A}^{\mathrm{T}}\mathbf{x}_j\right\|^2 \tag{9}$$

D. Joint Loss and Training Stage

The joint loss function of UCAN is composed of the label predictor loss $\mathcal{L}_c$, domain classifier loss $\mathcal{L}_c$, and distribution adapter loss $\mathcal{L}_m$. $\mathcal{L}_c$ is defined as:

$$\mathcal{L}_c\left(f_c, f_g\right)=\frac{1}{n_s}\sum_{i=1}^{n_s}L\left(f_c\left(f_g\left(x_i^s\right)\right), y_i^s\right) \tag{10}$$

$$=-\frac{1}{n_s}\sum_{i=1}^{n_s}\sum_{j=1}^{k}\mathbb{I}\left(y_i^s=j\right)\cdot\log f_c\left(f_g\left(x_i^s\right)\right) \tag{11}$$

where $\mathbb{I}\left(y_j^s=j\right)$ represents the indicator function.

The total loss for our UCAN architecture is defined as:

$$\mathcal{L}\left(\theta_g,\theta_c,\theta_d\right)=\mathcal{L}_c\left(\theta_g,\theta_c\right)+\alpha\mathcal{L}_m\left(\theta_g\right)-\beta\mathcal{L}_d\left(\theta_g,\theta_d\right) \tag{12}$$

Based on the above total loss function, the optimization target of parameters θ_g, θ_c, and θ_d can be given as:

$$\hat{\theta}_g,\hat{\theta}_c=\arg\min_{\theta_g,\theta_c}\mathcal{L}\left(\theta_g,\theta_c,\hat{\theta}_d\right) \tag{13a}$$

$$\hat{\theta}_d=\arg\max_{\theta_d}\mathcal{L}\left(\hat{\theta}_g,\hat{\theta}_c,\theta_d\right) \tag{13b}$$

The pseudo-code of the learning procedure of parameters θ_g, θ_c, and θ_d is provided in Algorithm **1**. [!ht] UCAN:

training procedure Source data $D_s = (X_s, y_s)$; Target data $D_t = (X_t, y_t)$; Loss weights α,β; Minibatch size m; Training step n; Learning rate μ.

Initialize feature extractor, label predictor, domain classifier with random weights θ_g,θ_c,θ_d.

$t = 1, \ldots, n$ Sample minibatch $\{x_i^s, y_i^s\}_{i=1}^{m/2}$, $\{x_i^t\}_{i=1}^{m/2}$ from D_S and D_T

Compute $\mathcal{L}_c$ loss by Eq. (10)

Compute $\mathcal{L}_m$ loss by Eq. (8)

Compute $\mathcal{L}_d$ loss by Eq. (4)

Combine loss by Eq. (13)

$\theta_g \leftarrow \theta_g - \mu \left(\frac{\partial L_c}{\partial \theta_g} + \alpha \frac{\partial L_m}{\partial \theta_g} - \beta \frac{\partial L_d}{\partial \theta_g} \right)$

$\theta_c \leftarrow \theta_c - \mu \frac{\partial L_c}{\partial \theta_c}$

$\theta_d \leftarrow \theta_d - \mu \frac{\partial L_d}{\partial \theta_d}$

θ_g, θ_c, θ_d

IV. Experimental Results

To evaluate the performance of the proposed UCAN architecture, we introduce our experimental design. Then, we investigate the following research questions:

Question 1: Can our UCAN architecture realize higher classification accuracy than state-of-the-art approaches? For this, we compared the classification accuracies in thirteen tax evasion migration scenarios between the proposed and state-of-the-art methods to evaluate its effectiveness.

Question 2: Can our method capture deep invariant feature representation cross-regions and realize superior distribution adaption? To illustrate this, we visualized the generated features by using t-SNE techniques to map high-dimensional features into a two-dimensional plane for display, which reveals the distribution adaption performances of the approaches.

Question 3: (3) How sensitive is our architecture to hyperparameter values? To investigate this, we varied two key hyperparameters in UCAN, namely, the loss weight α of the distribution adapter and the loss weight β of the domain classifier, and analyzed the influences of these hyperparameters on the classification accuracy to evaluate the sensitivity of our method to the hyperparameter values.

A. Dataset

In this paper, we use real inter-region tax detection data from two provinces in China, which were obtained from five regions, namely, G_1, G_2, G_3, G_4, and S, and are divided into 13 groups of migration tasks, as listed in Table I. In the transfer scenario $G_1 \rightarrow G_2$, G_1 represents the source region and G_2 represents the target region; other migration scenarios are represented in the same way. D_S, $|D_S|$ represent the detection data and the number of samples in the source region, respectively. Similarly, D_T, $|D_T|$ denote the detection data and the number of samples in the target region, respectively. In this experiment, all source region samples are used as the training set and all target region samples as the test set. The labels of the detection samples are divided into two categories: fraudulent invoicing enterprises and normal enterprises. The enterprises in the first class intended to evade tax and defraud tax authorities by submitting fraudulent value-added-tax invoices.

B. Comparison Methods

To verify the performance of the UCAN, we adopt up to seven both traditional and state-of-the-art transfer learning approaches as comparison methods in the experiments, namely the Source Only, TCA, GFK, DDC, DAN, JAN, and DANN. The Source-Only method only used source domain data for training. TCA and GFK are traditional non-deep unsupervised transfer learning methods. DDC, DAN and JAN are the deep transfer methods based on distribution divergence. These methods optimize network parameters by minimizing distribution distance and adapt source domain and target domain. DDC method maps the features of a hidden layer in the network to the reproducing kernel Hilbert space (RKHS), and uses MMD distance as the loss to optimize network. Multiple hidden layers in the neural network were mapped to the reproducing kernel Hilbert space by the DAN method, and the MK-MMD distance is used as the loss to optimize network. JAN combines hidden layer features and labels to optimize the JMMD distance between source and target domains. DANN is a deep transfer method based on the adversarial network. After the feature generator, the domain classifier network is added. It uses adversarial method to train data. In the experiments, our UCAN architecture is divided into UCAN-C and UCAN-K two models according to the manner of combination of features and labels.

C. Effectiveness of UCAN

1) Performance Calculation using Different Approaches: To evaluate the performance of different approaches on every migration task, we calculate the classification accuracy (ACC) and area under curve (AUC) on all thirteen transfer scenarios.

Table II shows the ACC of different methods, where UC-C and UC-K represent UCAN-C and UCAN-K respectively. In the most transfer tasks, the performance of UCAN-C and UCAN-K methods are better than the comparison approaches, especially in $G \rightarrow S$ task with great difficulty in transfer. The UCAN has the accuracy rate of 78.5% which is obviously better than other comparison methods. Only in $G_1 \rightarrow G2$ and $G_4 \rightarrow G_3$ migration tasks, the ACC of UCANs (UCAN-C and UCAN-TEK-K) are slightly lower than that of DAN or JAN. The ACC of traditional non-deep transfer learning methods TCA and GFK are lower than that of the Source-Only method, indicating that negative transfer occurs in the process of migration. The reason is that the non-deep transfer method will reduce the ACC of the Source domain through unsupervised feature mapping. Even if the mapping is to the same distribution, the accuracy of the target domain will not be ideal. Deep transfer method including DANN, DDC, DAN, JAN, and UCAN are all better than the benchmark method the Source Only, which shows that the deep transfer learning can effectively improve the detection accuracy of the

Scenarios	$G_1 \to G_2$	$G_1 \to G_3$	$G_1 \to G_4$	$G_2 \to G_1$	$G_2 \to G_3$	$G_2 \to G_4$	$G_3 \to G_1$
$\|D_S\|$	3388	3388	3388	1630	1630	1630	1580
$\|D_T\|$	1630	1580	646	1630	1580	646	1580
Scenarios	$G_3 \to G_2$	$G_3 \to G_4$	$G_4 \to G_1$	$G_4 \to G_2$	$G_4 \to G_3$	$G \to S$	
$\|D_S\|$	1580	1580	646	646	646	8940	
$\|D_T\|$	1580	646	646	646	646	2768	

TABLE I
THIRTEEN TRANSFER SCENARIOS

target domain. The reason for this is that model can learn features with more transferable by the way of deep network domain adaptive embedding. Furthermore, DAN method is an improved version of DDC method. It uses multi-layer multi-core mapping to take place of the original single-layer single-core mapping, so DAN is superior to DDC in most tasks. The ACC of UCANs are better than that of DANN method, since DANN only considers the difference between domains, but does not consider the intra-class distance between same classes. UCANs add JPDs of pseudo label adaptation of samples into the domain classifier which can improve the effect of the models.

Table III shows the AUC score of different methods on thirteen migration tasks. In dataset with unbalanced categories, the AUC score often reflects the real effect of the model more than the ACC. In $G \to S$ task with great difficulty in transfer, UCANs reaches the AUC score of 0.835, which is 11.1% higher than the sub-optimal JAN method. In other transfer tasks, the AUC scores of UCAN-C and UCAN-K are all higher than 0.9, and the average of the AUC score reached 0.96.

2) Feature Visualization: To evaluate distribution adaptation performance of our architecture, we visualized the generated features in particular $G \to S$ transfer task on methods: the Source-Only, DANN, UCAN-C, and UCAN-K. Specifically, as shown in Fig.**??**, the mapping features came from the output of the feature extractor's last layer h (the third layer of the network) and we used t-SNE technics to map from high-dimensional features to two-dimensional features for plane display. There are four types of dots in the figure. Dark red and magenta dots represent enterprises with fraudulent invoices in the source and target regions respectively, while dark blue and cyan dots represent normal enterprises in the source and target regions respectively. As illustrated in the figure, 50 dots of each type are sampled and displayed. Fig.3 shows the visualization of features in the Source-Only method. Since no domain adaptation is added, the source and target domain samples are completely separated, and the data distribution between the fields is quite different, in this case the source domain trained model has a large generalization error in the target domain. Fig.4 shows the visualization of DANN method features. It can be seen that the dark blue and cyan dots, dark red and magenta dots have initially gathered into two clusters. The normal dots of the left cluster accounted for the majority, while the fraudulently invoiced enterprises of the right cluster accounted for the majority. However, there are still some magenta dots in the normal enterprise cluster and some cyan dots in the fraudulently invoiced enterprise cluster. The reason is that DANN only adapts the probability of marginal distribution and does not take into account the label information of samples. In Fig.5, the UCAN-C method has basically completely separated the blue node from the red node, and the red and blue nodes can be classified by simple logistic regression only relying on the two-dimensional information on the image. However, there are many dots in the classification plane of the UCAN-C method, and the classification plane is not very clear. The UCAN-K method in Fig.6 not only distinguishes the red and blue nodes, but also has fewer dots on the classification plane and larger inter-class distance between the categories.

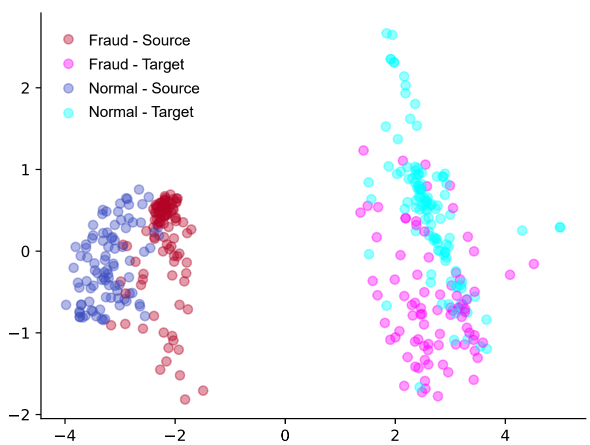

Fig. 3. Feature Visualization Analysis on Source-Only

3) Parametric Sensitivity Analysis: Two key hyper-parameters of UCAN architecture are selected for sensitivity analysis, which are the loss weight α of the distribution adapter and loss weight β of the domain classifier. The influence of parameter settings of α and β on accuracy is analyzed in the experiment. Fig. 7 and Fig. 8 illustrate the ACC changes of UCAN-C and UCAN-K in migration task $G \to S$. Fig. 7 analyzes the sensitivity of parameter α, which β is fixed to 0.2 in the experiment and we respectively take $\alpha \in$ 0.01, 0.02, 0.05, 0.1, 0.2, 0.5. It is observed that UCAN-C and UCAN-C increase first and then decrease, and the optimal value is obtained when $\alpha = 0.2$. The graph is displayed as a bell curve, which proves the effectiveness of the distribution adapter.

Method	Source	TCA	GFK	DANN	DDC	DAN	JAN	**UC-C**	**UC-K**
$G \rightarrow S$	0.637	0.604	0.646	0.673	0.703	0.674	0.691	**0.785**	0.760
$G_1 \rightarrow G_2$	0.848	0.771	0.795	0.880	0.879	**0.885**	0.801	0.853	0.828
$G_1 \rightarrow G_3$	0.657	0.773	0.713	0.674	0.907	0.892	0.641	**0.915**	0.863
$G_1 \rightarrow G_4$	0.896	0.827	0.791	0.892	0.899	0.895	0.901	**0.929**	0.902
$G_2 \rightarrow G_1$	0.800	0.754	0.748	0.875	0.827	0.856	0.865	0.886	**0.913**
$G_2 \rightarrow G_3$	0.844	0.704	0.765	0.897	0.917	0.898	0.937	0.897	**0.941**
$G_2 \rightarrow G_4$	0.788	0.754	0.762	0.850	0.820	0.810	0.867	**0.878**	0.875
$G_3 \rightarrow G_1$	0.875	0.633	0.680	0.944	0.900	0.924	0.904	**0.953**	0.940
$G_3 \rightarrow G_2$	0.645	0.665	0.807	0.860	0.879	0.872	0.817	0.899	**0.912**
$G_3 \rightarrow G_4$	0.834	0.618	0.740	0.882	0.830	0.896	0.861	**0.929**	0.921
$G_4 \rightarrow G_1$	0.875	0.701	0.594	0.893	0.857	0.902	0.891	**0.925**	0.895
$G_4 \rightarrow G_2$	0.716	0.598	0.604	0.822	0.816	0.844	0.831	**0.848**	0.818
$G_4 \rightarrow G_3$	0.757	0.694	0.544	0.854	0.794	0.816	**0.878**	0.820	0.850
avg	0.783	0.700	0.707	0.846	0.848	0.859	0.837	**0.886**	0.878

TABLE II
ACC SCORE OF TAX EVASION DETECTION WITH DIFFERENT METHODS

Method	Source	TCA	GFK	DANN	DDC	DAN	JAN	**UC-C**	**UC-K**
$G \rightarrow S$	0.636	0.604	0.690	0.723	0.742	0.716	0.724	**0.835**	0.814
$G_1 \rightarrow G_2$	0.938	0.771	0.854	0.939	0.950	**0.950**	0.874	0.929	0.935
$G_1 \rightarrow G_3$	0.781	0.773	0.797	0.772	0.950	0.934	0.728	**0.963**	0.924
$G_1 \rightarrow G_4$	0.937	0.827	0.860	0.939	0.950	0.967	0.937	**0.982**	0.979
$G_2 \rightarrow G_1$	0.900	0.754	0.815	0.953	0.896	0.927	0.940	0.974	**0.975**
$G_2 \rightarrow G_3$	0.879	0.704	0.831	0.952	0.967	0.955	0.958	**0.987**	0.980
$G_2 \rightarrow G_4$	0.872	0.754	0.815	0.917	0.888	0.896	0.920	**0.965**	0.933
$G_3 \rightarrow G_1$	0.926	0.633	0.730	0.978	0.953	0.969	0.942	**0.988**	0.986
$G_3 \rightarrow G_2$	0.721	0.665	0.838	0.925	0.936	0.935	0.899	0.961	**0.966**
$G_3 \rightarrow G_4$	0.884	0.618	0.787	0.922	0.921	0.953	0.911	**0.968**	0.957
$G_4 \rightarrow G_1$	0.946	0.701	0.693	0.955	0.912	0.952	0.949	**0.968**	0.963
$G_4 \rightarrow G_2$	0.828	0.598	0.709	0.901	0.875	0.894	0.911	**0.921**	0.899
$G_4 \rightarrow G_3$	0.853	0.694	0.664	0.922	0.836	0.878	**0.946**	0.926	0.913
avg	0.854	0.700	0.776	0.908	0.906	0.917	0.895	0.**951**	0.940

TABLE III
AUC SCORE OF TAX EVASION DETECTION WITH DIFFERENT METHODS

Similarly, Fig. 8 analyzes the sensitivity of the parameter β where α is set to 0.2 in the experiment and we respectively take $\beta \in$ 0.01, 0.02, 0.05, 0.1, 0.2, 0.5. The bell curve obtains its optimal value when $\beta = 0.2$, which reveals the effectiveness of the domain classifier.

V. CONCLUSION

In this paper, we proposed a general architecture named UCAN and applies it to inter-region tax evasion detection, which utilizes labeled data of other audit tasks to assist target label-sparse audit tasks and reduces intra-class distribution discrepancies. Our architecture establishes an adversarial neural network for migration tasks. After deep feature representation is captured from the feature extractor, we add label information in the domain classifier to adapt JPD. In the distribution adapter, CMMD distance is introduced to align CPD. By combining the distribution adapter and label predictor, our UCAN architecture realizes end-to-end learning of unsupervised feature transfer. The experimental results reveal the outstanding performance of our architecture in all transfer tasks compared with state-of-the-art approaches.

Since UCAN architecture is a general framework, it can be applied not only to tax evasion detection migration tasks, but also other transfer scenarios as well.

ACKNOWLEDGMENT

This research was partially supported by “The Fundamental Theory and Applications of Big Data with Knowledge Engineering” under the National Key Research and Development

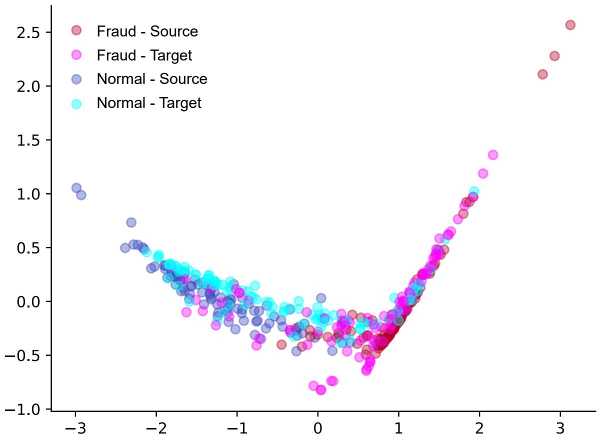

Fig. 4. Feature Visualization Analysis on DANN

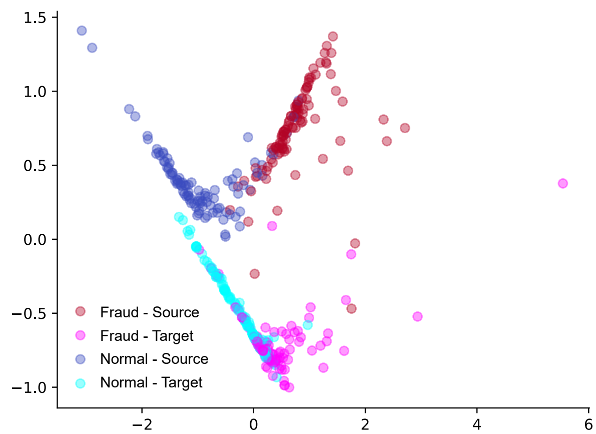

Fig. 5. Feature Visualization Analysis on UCAN-C

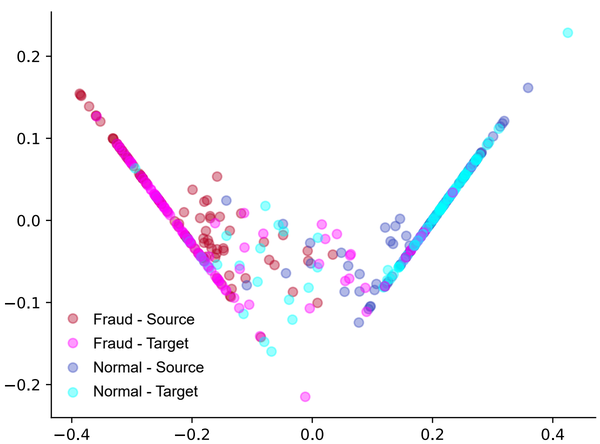

Fig. 6. Feature Visualization Analysis on UCAN-K

Program of China with Grant No. 2018YFB1004500, the MOE Innovation Research Team No. IRT17R86, the National Science Foundation of China under Grant Nos. 61721002, 61532015, and Project of China Knowledge Centre for En-

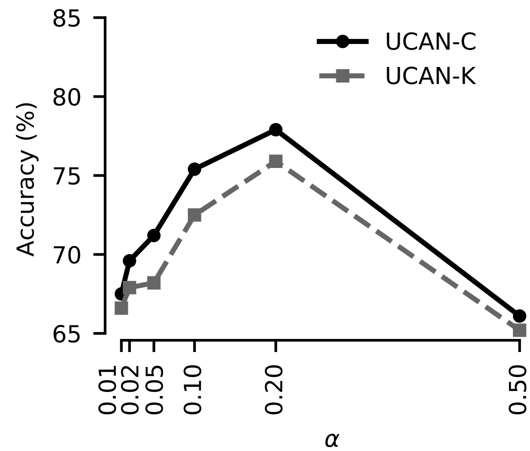

Fig. 7. Parametric Sensitivity Analysis of UCAN-C and UCAN-K on Accuracy w.r.t α

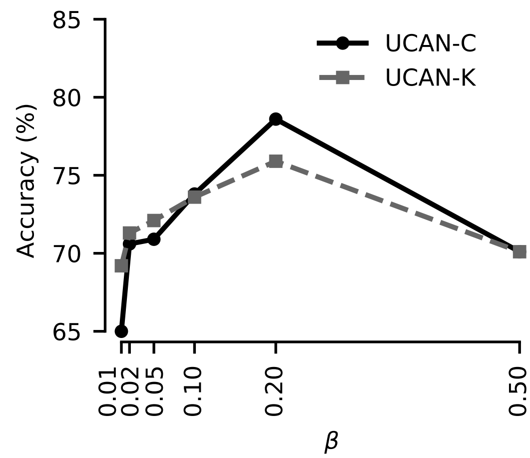

Fig. 8. Parametric Sensitivity Analysis of UCAN-C and UCAN-K on Accuracy w.r.t β

gineering Science and Technology.

References

[1] F. Tian et al., Mining Suspicious Tax Evasion Groups in Big Data, IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 10, pp. 2651C2664, Oct. 2016.

[2] X. Zhu, Z. Yan, J. Ruan, Q. Zheng, and B. Dong, IRTED-TL: An Inter-Region Tax Evasion Detection Method Based on Transfer Learning, in 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), 2018, pp. 1224C1235.

[3] S. J. Pan and Q. Yang, A Survey on Transfer Learning, IEEE Transactions on Knowledge and Data Engineering, vol. 22, no. 10, pp. 1345C1359, Oct. 2010.

[4] R.-S. Wu, C. S. Ou, H. Lin, S.-I. Chang, and D. C. Yen, Using data mining technique to enhance tax evasion detection performance, Expert Systems with Applications, vol. 39, no. 10, pp. 8769C8777, Aug. 2012.

[5] T. Matos, J. A. F. de Macedo, and J. M. Monteiro, An Empirical Method for Discovering Tax Fraudsters: A Real Case Study of Brazilian Fiscal Evasion, in Proceedings of the 19th International Database Engineering Applications Symposium, New York, NY, USA, 2014, pp. 41C48.

[6] X. Liu, D. Pan, and S. Chen, Application of Hierarchical Clustering in Tax Inspection Case-Selecting, in 2010 International Conference on Computational Intelligence and Software Engineering, 2010, pp. 1C4.

[7] Z. Assylbekov, I. Melnykov, R. Bekishev, A. Baltabayeva, D. Bissengaliyeva, and E. Mamlin, Detecting Value-Added Tax Evasion by?Business Entities of Kazakhstan, in Intelligent Decision Technologies 2016, 2016, pp. 37C49.

[8] Y.-S. Chen and C.-H. Cheng, A Delphi-based rough sets fusion model for extracting payment rules of vehicle license tax in the government sector, Expert Systems with Applications, vol. 37, no. 3, pp. 2161C2174, Mar. 2010.

[9] M. Gupta and V. Nagadevara, Audit Selection Strategy for Improving Tax Compliance C Application of Data Mining Techniques, p. 10.

[10] G. Warner et al., Modeling tax evasion with genetic algorithms, Econ Gov, vol. 16, no. 2, pp. 165C178, May 2015.

[11] N. Abe et al., Optimizing Debt Collections Using Constrained Reinforcement Learning, in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2010, pp. 75C84.

[12] L. Akoglu, H. Tong, and D. Koutra, Graph based anomaly detection and description: a survey, Data Min Knowl Disc, vol. 29, no. 3, pp. 626C688, May 2015.

[13] T. Llacer, F. J. Miguel, J. A. Noguera, and E. Tapia, An agent-based model of tax compliance: an application to the spanish case, Advs. Complex Syst., vol. 16, no. 04n05, p. 1350007, Aug. 2013.

[14] J. A. Noguera, F. J. M. Quesada, E. Tapia, and T. Llcer, Tax Compliance, Rational Choice, and Social Influence: An Agent-Based Model, Revue francaise de sociologie, vol. Vol. 55, no. 4, pp. 765C804, 2014.

[15] W. Dai, Q. Yang, G.-R. Xue, and Y. Yu, Boosting for Transfer Learning, in Proceedings of the 24th International Conference on Machine Learning, New York, NY, USA, 2007, pp. 193C200.

[16] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, Domain Adaptation via Transfer Component Analysis, IEEE Transactions on Neural Networks, vol. 22, no. 2, pp. 199C210, Feb. 2011.

[17] G. Ke et al., LightGBM: A Highly Efficient Gradient Boosting Decision Tree, in Advances in Neural Information Processing Systems 30, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 3146C3154.

[18] B. Tan, Y. Song, E. Zhong, and Q. Yang, Transitive Transfer Learning, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 2015, pp. 1155C1164.

[19] B. Tan, Y. Zhang, S. J. Pan, and Q. Yang, Distant Domain Transfer Learning, in Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[20] M. Long, J. Wang, G. Ding, J. Sun, and P. S. Yu, Transfer Feature Learning with Joint Distribution Adaptation, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2200C2207.

[21] Y. Ganin et al., Domain-Adversarial Training of Neural Networks, in Domain Adaptation in Computer Vision Applications, G. Csurka, Ed. Cham: Springer International Publishing, 2017, pp. 189C209.

[22] Z. Cao, M. Long, J. Wang, and M. I. Jordan, Partial Transfer Learning With Selective Adversarial Networks, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2724C2732.

[23] F. Lv, J. Zhu, G. Yang, and L. Duan, TarGAN: Generating target data with class labels for unsupervised domain adaptation, Knowledge-Based Systems, vol. 172, pp. 123C129, May 2019.

[24] B. Fernando, A. Habrard, M. Sebban, and T. Tuytelaars, Unsupervised Visual Domain Adaptation Using Subspace Alignment, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 2960C2967.

[25] M. Wang and W. Deng, Deep visual domain adaptation: A survey, Neurocomputing, vol. 312, pp. 135C153, Oct. 2018.

[26] 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 - Volume 70, Sydney, NSW, Australia, 2017, pp. 2208C2217.