Deep Transfer Cooperative Sensing in Cognitive Radio

Lusi Li, He Jiang, and Haibo He, *Fellow, IEEE*

***Abstract*—In this letter, we propose a deep transfer cooperative sensing (DTCS) approach in cognitive radio networks, where multiple secondary users (SUs) cooperate to detect the presence of signals from a primary user (PU) in a shared frequency band. DTCS is a cooperative spectrum sensing (CSS) framework based on unsupervised deep transfer learning. It operates on energy vectors, whose each element is a sensing result by an energy detector from individual SU. It learns the knowledge by combining the sensing results from all SUs in one radio frequency environment and transfers it to another one. This approach is applicable for detecting the presence of arbitrary unknown signals, which enhances the generalization ability and robustness of the framework. Simulation results demonstrate the effectiveness of DTCS.**

***Index Terms*—Cognitive radio, cooperative spectrum sensing, unsupervised deep transfer learning, generalization, robustness.**

I. Introduction

COGNITIVE radio (CR) has been a promising technology to enhance the utilization of limited spectrum in wireless communications. Specifically, CR senses the radio frequency (RF) environment and allows unlicensed/secondary users (SUs) to opportunistically utilize the licensed spectrum, which is owned by licensed/primary users (PUs). To maximize spectrum utilization without interfering with the PUs' usages, SUs should have the ability of spectrum sensing (SS) to detect the presence of primary signals from PUs [1].

Since the sensing performance of individual SU may degrade due to the issues of shadowing, multipath fading, and receiver uncertainty, cooperative spectrum sensing (CSS) is presented to address this problem [2]. In CSS, SUs situated in different locations cooperate to achieve higher sensing performance than individual SU does [3]. This cooperation is completed by a fusion center, combining sensing information and making decisions [4].

In practice, the SUs often have no prior knowledge of the primary signals. Designing a blind sensing approach becomes an urgent issue. Recently, deep learning (DL) has achieved considerable success in identifying the underlying structures of different objects on various complex tasks, such as computer vision, speech recognition, and wireless communications [5]. Thus it has great potential in SS by exploring the latent structures of the primary signals [6]. Some DL-based signal detectors have been proposed. For example, [7] designs the DetectNet based on convolutional long short-term deep neural networks and it can detect arbitrary types of primary signals in the case of SS. For CSS, it employs the DetectNet for each SU separately and then presents the SoftComninationNet for the fusion center to make decisions. However, most of the existing DL-based methods need a large amount of labeled training data and are tested on the data with a similar distribution. That is to say, when the RF environment changes, directly applying the trained model to detect the new primary signals may have poor performance. For example, PU transmits different types of signals. The bigger the difference between the original and new signals, the poorer the detection performance will be. Moreover, manual annotation is often expensive and time-consuming. Hence, deep transfer learning (DTL) can be adopted to reduce the annotation time and cost by adapting label-rich signal data (i.e., source domain) to related but different label-scarce signal data (i.e., target domain). In [8], the authors propose a deep sensing method with transfer learning for SS. Two cases are considered. One case is an unsupervised transfer learning. The other case is to fine-tune the framework with a small amount of labeled target data.

Motivated by [8], in this letter, we propose a deep transfer cooperative sensing (DTCS) method, which is a novel unsupervised DTL-based CSS scheme for CR. We summarize the contributions as follows:

- We propose DTCS, which is a CSS scheme and applicable for detecting the presence of arbitrary unknown signals from PU in the new communications settings.
- DTCS operates on energy vectors, whose each element is a sensing result by an energy detector from individual SU. It learns the knowledge by combining all sensing results from a source domain and transfers it to a target domain.
- The effectiveness of DTCS is demonstrated in simulations. It can achieve better sensing performance compared with several baseline detectors.

II. System Model

We consider a CR network, where a frequency channel is shared by N_{su} SUs and a single PU. For CSS, to detect the presence of primary signals, each SU employs an energy detector (ED) to estimate the energy level at a sensing time period τ and sensing bandwidth ω. Specifically, we consider the sampling frequency as the Nyquist rate $f_s = 2\omega$. Thus each SU obtains $K = 2\omega\tau$ received signals at each time period. The

Manuscript received December 19, 2020; revised February 23, 2021; accepted March 12, 2021. Date of publication March 19, 2021; date of current version June 9, 2021. This work was supported by the National Science Foundation under Grant ECCS 1731672. The associate editor coordinating the review of this article and approving it for publication was W. Cheng. *(Corresponding author: Haibo He.)*

The authors are with the Department of Electrical, Computer and Biomedical Engineering, University of Rhode Island, Kingston, RI 02881 USA (e-mail: lusi_li@uri.edu; jiang_he@uri.edu; haibohe@uri.edu).

Digital Object Identifier 10.1109/LWC.2021.3067508

k-th ($k = 1, \ldots, K$) received signal of SU_i ($i = 1, \ldots, N_{su}$) can be denoted by

$$x_i(k) = \begin{cases} h_i(k)z(k) + w_i(k), & \text{if PU is present} \\ w_i(k), & \text{if PU is absent} \end{cases} \quad (1)$$

where $h_i(k)$ is the channel coefficient from PU to SU_i; $z(k)$ is the signal data transmitted by PU; $w_i(k)$ is the noise at SU_i. The channel coefficient $h_i(k)$ is described by the fading and path-loss components as follows:

$$h_i(k) = \gamma e_i^{-\varepsilon/2} \quad (2)$$

where γ is the fading component; e_i is the Euclidean distance between PU and SU_i; ε is the path-loss component.

By the energy detector, the estimated energy level of SU_i can be obtained by

$$d_i = \frac{1}{K} \sum_{k=1}^{K} |x_i(k)|^2 \quad (3)$$

We let $d = (d_1, \ldots, d_{N_{su}})^{\mathrm{T}}$ be an energy vector of the accumulated energy levels of all SUs in a single interval τ.

In this letter, we assume the PU can alternate between the present and absent states. Each energy vector d can be considered as an input vector of our proposed DTCS method. Given d, DTCS aims to predict the presence of the primary signals from PU, which is denoted by y.

$$y = \begin{cases} 1, & \text{if PU is present} \\ 0, & \text{if PU is absent.} \end{cases} \quad (4)$$

III. Related Work

In cooperative sensing, the SUs report the sensing results to the fusion center for decision making [9]. The fusion center has two fusion strategies for the sensing results: hard fusion and soft fusion. For the hard fusion strategies, the SUs report only one-bit decision information to the fusion center, which represents whether the received energy level d_i from the i-th SU is greater than a particular threshold. For example, the OR rule, the AND rule, and Maximum Ratio Combining (MRC) rule are commonly used for CSS.

For the soft fusion strategies, the SUs report the entire sensing results to the fusion center without performing the local decisions. Machine learning (ML) technologies are widely used soft fusion strategies, where the CSS problem can be considered as a binary classification problem, and the energy vectors are as the input. For example, the authors in [2] propose several CSS algorithms, including Support Vector Machine (SVM) and Multi-Layer Perceptron (MLP).

It is shown that soft fusion strategies perform better than hard fusion strategies based on energy detection [10]. The ML methods as the soft fusion strategies are trained and tested under the same set of communications conditions. They may not work well when wireless communications settings change. To this end, the authors in [8] propose a sensing method with Transfer Component Analysis (TCA) for SS. However, it only considers the difference between domains and ignores the differences between classes. Moreover, it is designed for SS and not for CSS. Unlike them, our proposed method incorporates deep transfer learning to CSS by adapting the learned method to the new communications settings. We attempt to improve the generalization ability of the sensing model by reducing the differences in terms of both domains and classes.

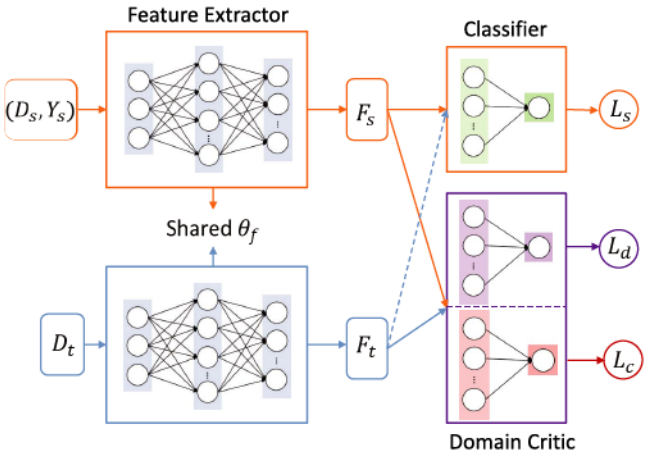


Fig. 1. The architecture of the proposed DTCS framework.

IV. Deep Transfer Cooperative Sensing

Motivated by [8], in this letter, we aim to enhance the generalization ability and robustness of the CSS scheme in the absence of the knowledge of the CR network parameters, such as the types of PU signals and the SNR λ_i at SU_i. For CSS, we are given a labeled source data set $(D_s, Y_s) = \{(d_s^i, y_s^i)\}_{i=1}^{n_s}$ with and an unlabeled target data set $D_t = \{d_t^j\}_{j=1}^{n_t}$, where d_s^i and d_t^j are the estimated energy vectors from source and target domains, respectively; n_s and n_t are the numbers of the energy vectors from source and target domains, respectively. Our proposed DTCS framework aims to utilize a feature extractor, a domain critic net, and a classifier to learn the domain-invariant features and predict the labels of unknown target data.

A. Framework of DTCS

Fig. 1 shows our proposed DTCS framework. Inspired by the previous work [11], [12], DTCS is designed with three components: a feature extractor with shared weights θ_f extracts the source features F_s and target features F_t; a classifier is trained by the labeled F_s and predicts labels for F_t; a domain critic net aligns both domains by reducing the differences in terms of domain and class.

We are given the extracted source features $f_s = F(d_s)$ and target features $f_t = F(d_t)$ from the feature extractor F with shared weights θ_f.

Firstly, we train a classifier C_y using the labeled source samples by the supervised loss function:

$$\mathcal{L}_s(d_s, y_s) = \frac{1}{n_s} \sum_{i=1}^{n_s} L(C_y(f_s^i(d_s^i)), y_s^i) \quad (5)$$

where L is a cross-entropy loss function; $f_s^i(d_s^i)$ represents the feature extracted from the feature extractor. The parameters θ_s of C_y can be learned by minimizing the softmax cross entropy

Algorithm 1 Deep Transfer Cooperative Sensing

Input: Source data (D_s, Y_s), target data D_t, the minibatch size m, training step of classifier T, training step of domain critic A, learning rate ϕ.

Output: The presence of target data Y_t.

Initialize feature extractor, classifier, and domain critic with random parameters θ_f, θ_s, θ_d, and θ_c.
repeat
  Sample minibatch $\{(d_s^i, y_s^i)\}_{i=1}^m$ from (D_s, Y_s)
  Sample minibatch $\{d_t^i\}_{i=1}^m$ from D_t
  for $t = 1, 2, \ldots, T$ **do**
    $\theta_s \leftarrow \theta_s - \phi \nabla_{\theta_s} \mathcal{L}_s$
  end for
  $f_s \leftarrow F(d_s)$, $f_t \leftarrow F(d_t)$, $\widehat{Y_t} \leftarrow C_y(f_t)$
  Sample z_f as the random representations between pairs of f_s and f_t
  Sample z_c^k as the random representations between pairs of f_s^k and f_t^k $(k = \{0, 1\})$
  for $a = 1, 2, \ldots, A$ **do**
    $\theta_d \leftarrow \theta_d + \phi \nabla_{\theta_d} \mathcal{L}_d$
    $\theta_c \leftarrow \theta_c + \phi \nabla_{\theta_c} \mathcal{L}_c$
  end for
  $\theta_f \leftarrow \theta_f - \phi \nabla_{\theta_f} [\mathcal{L}_s + \mathcal{L}_d + \mathcal{L}_c]$
until θ_s, θ_d, θ_c, and θ_f converge
Return $Y_t = C_y(F(D_t))$

$\mathcal{L}_s$. Then we can use the trained C_y to obtain the predicted labels of the target samples by $\widehat{Y_t} = C_y(f_t(d_t))$.

In the domain critic, with f_s and f_t, the improved Wasserstein distance [13] (IWD) is used to measure the difference in terms of domain by

$$\mathcal{L}_d(d_s, d_t) = \frac{1}{n_s} \sum_{d_s \in D_s} h_d(f_s) - \frac{1}{n_t} \sum_{d_t \in D_t} h_d(f_t) - \lambda_d (|| \bigtriangledown_{z_f} h_d(z_f)||_2 - 1)^2 \quad (6)$$

where h_d is a function that maps the feature f to a real number with parameter θ_d; λ_d is a balancing coefficient; $(|| \bigtriangledown_{z_f} h_d(z_f)||_2 - 1)^2$ is a gradient penalty for θ_d to void the issues of gradient vanishing or exploding in the training procedure. The IWD between domains can be estimated by maximizing the domain loss $\mathcal{L}_d$.

With (f_s, Y_s) and $(f_t, \widehat{Y_t})$, similarly, the IWD can also be utilized to measure the differences between classes by maximizing the class loss $\mathcal{L}_c$.

$$\mathcal{L}_c(d_s, d_t) = \sum_{k=0}^{1} \{ \frac{1}{n_s^k} \sum_{d_s \in D_s^k} h_c^k(f_s^k) - \frac{1}{\widehat{n_t^k}} \sum_{d_t \in \widehat{D_t^k}} h_c^k(\widehat{f_t^k}) - \lambda_c^k (|| \bigtriangledown_{z_c^k} h_c^k(z_c^k)||_2 - 1)^2 \} \quad (7)$$

where k is the label; h_c^k is a function that maps the feature f^k to a real number with parameter θ_c^k; $\theta_c = \{\theta_c^0, \theta_c^1\}$.

The overall objective of our proposed DTCS method is as follows:

$$\min_{\theta_f, \theta_s} \{ \mathcal{L}_s + \max_{\theta_d} \mathcal{L}_d + \max_{\theta_c} \mathcal{L}_c \}. \quad (8)$$

B. Training of DTCS

We show the detailed training procedure of DTCS in Algorithm 1. We can achieve the overall objective by the standard back-propagation training approach with a two-step iteration. Firstly, the classifier is trained by minimizing $\mathcal{L}_s$. Secondly, the trained classifier is applied to predict the labels of target data. Moreover, the domain critic aligns the two domains by maximizing $\mathcal{L}_d$ and $\mathcal{L}_c$. We update the feature extractor by the combination of the above three losses. With each iteration of DTCS, the errors introduced from wrongly predicted target labels can be alleviated. For the testing procedure, the trained feature extractor can learn the features of target test data and the trained classifier can predict the corresponding labels.

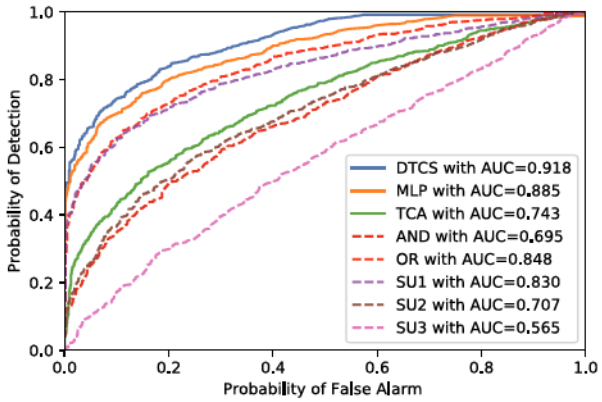
In addition, the computational complexity of DTCS with respect to the number of SUs N_{su} is $O(N_{su})$. The reason is that the number of SUs corresponds to the number of nodes in the input layer. Therefore, if we assume the numbers of nodes in the hidden layers are fixed for a pre-selected architecture, then the number of SUs will only affect the size of the weight matrix from the input layer to the first hidden layer of the neural network structure. Hence, the computational complexity of the DTCS is linearly correlated with the number of SUs.

V. Simulations

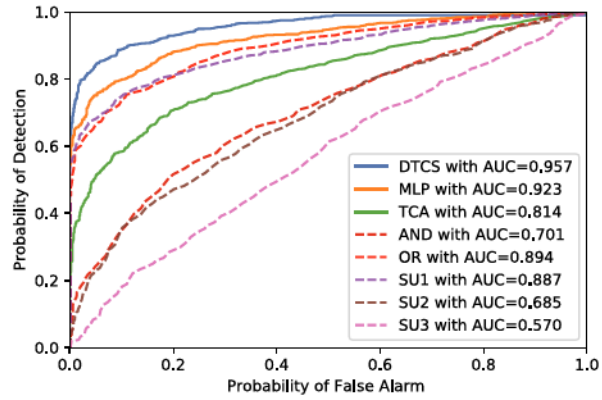
A. Setup

Sensing robustness in the unsupervised learning fashion was shown to be a challenging problem [8]. Given labeled source PU signals in one RF environment, we evaluate the generalization ability and robustness of our proposed DTCS method on the unlabeled target PU signals in the new RF environment. In this letter, we generate 4 kinds of digitally modulated signals in additive white Gaussian noise (AWGN) with different noise power spectral densities (PSDs) as positive samples and the corresponding AWGN as negative samples. For the simulation parameters, we have $K = 2\omega\tau = 2 * (5MHz) * (10\mu s) = 100$; the PU activate probability 0.5; the PU transmission power 0.1 mW; the QPSK signal data experience the fading component $\gamma = 2$ and the path-loss exponent $\varepsilon = 3$ under the noise PSDs $\delta_q = -150$dBm/Hz; the Gaussian signal data experience $\gamma = 1$ and $\varepsilon = 4$ under $\delta_g = -153$dBm/Hz; PAM4 signals experience $\gamma = 2$ and $\varepsilon = 3$ under $\delta_p = -150$dBm/Hz; 16-PSK signals experience $\gamma = 1$ and $\varepsilon = 4$ under $\delta_{16} = -153$dBm/Hz. In the simulations, we vary the number of SUs N_{su}, the number of training samples n_s, and the type of PU signals to examine our proposed DTCS method.

The following baselines are compared with our proposed DTCS methods: OR rule, AND rule, MRC, TCA, Support Vector Machine (SVM), and MLP. We implement all our experiments by TensorFlow. For TCA, we tune the parameters and report the average best results. MLP has two hidden layers of 500 and 100 nodes. MLP is trained only by the labeled source data and tested on the target data directly. For DTCS, the feature extractor includes two hidden layers of 500 nodes and 100 nodes; the classifier is designed with one hidden layer of 100 nodes, relu activation function, and softmax output function; the domain critic net has one hidden layer of

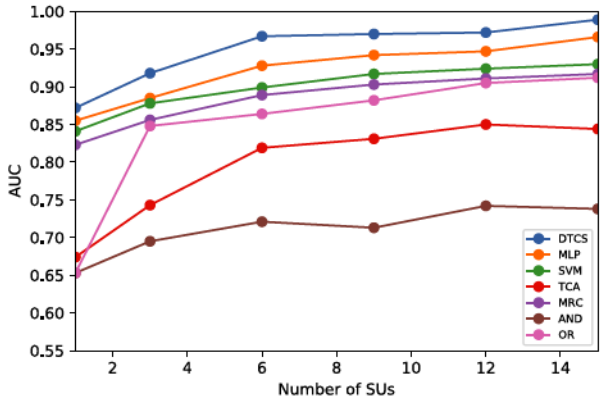


(a) Transfer task: QPSK → Gaussian

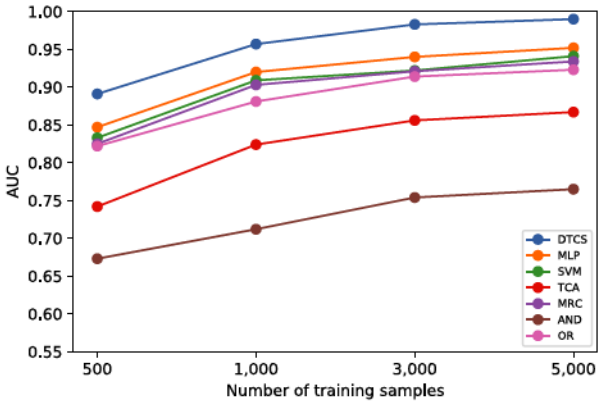


(b) Transfer task: Gaussian → QPSK

Fig. 2. Sensing performance of the CSS scheme with $N_{su} = 3$ for two transfer tasks.



(a) N_{su} vs. AUC when $n_s = 2,000$



(b) n_s vs. AUC when $N_{su} = 12$

Fig. 3. Sensing performance on the transfer task QPSK → Gaussian.

100 nodes for domain alignment and one hidden layer of 100 nodes for class alignment. The minibatch size m is 128. The training steps are $T = A = 5$. The learning rate is $\phi = 10^{-4}$. Two metrics are used to measure the performance of all the models: AUC and ROC curve. The ROC curve is the curve of probability of false alarm P_{fa} vs. probability of detection P_d. We report the average AUC values of each transfer task over five experiments.

B. Simulation Results

In Fig. 2, we show the sensing performance by DTCS, the baselines, and the ED for each single SU on two transfer tasks: QPSK → Gaussian and Gaussian → QPSK.

In this CSS scenario, $N_{su} = 3$; the distances between PU and 3 SUs (SU_1, SU_2, and SU_3) are 500m, 750m, and 1000m, respectively. We ran Monte-Carlo simulations with 2,000 realizations for source and target domains. Thus, source and target data sets have the same size $n_s = n_t = 2,000$. Specifically, Fig. 2(a) shows the ROC curves and average AUC values on the transfer task QPSK → Gaussian, where we use QPSK signals as the source domain and Gaussian signals as the target domain. Fig. 2(b) shows the results on the transfer task Gaussian → QPSK. From the results, DTCS outperforms the other compared methods. It improves the generalization ability of CSS compared with MLP due to the transfer learning strategy. MLP performs better than TCA, which demonstrates the superiority of neural networks. In terms of traditional methods, OR rule has a better performance than AND rule and the ED of SU_1 closer to PU performs better than that of the other SUs.

In Fig. 3, we show the average AUC vs. the number of SUs and the number of training samples in the source domain on the transfer task QPSK → Gaussian.

The single PU and multiple SUs are randomly deployed in a 2000m × 2000m area. For Fig. 3(a), we fix $n_s = n_t = 2,000$. Roughly speaking, it can be observed that the AUC value increases as the number of SUs increases. It highlights the benefits of the CSS scheme. DTCS shows better performance than the other baselines in a large margin. For Fig. 3(b), we fix $N_{su} = 12$ and $n_t = 2,000$. We can observe that the AUC value increases as the number of training samples in the source domain increases. Given larger amounts of labeled source data, DTCS can have better generalization ability and robustness by exploring the underlying structural information across domains.

In order to further verify the ability to detect the presence of arbitrary unknown signals, we evaluate DTCS on the transfer tasks for different PU signals. We fix $N_{su} = 12$ and $n_s = n_t = 2,000$. DTCS is tested on different transfer tasks shown in Table I. DTCS achieves the best performance on all transfer tasks. Note that the performance is different when we interchange the source and target domains. We can observe that some kinds of signals can make effective guidance in transfer learning, such as Gaussian signals.

TABLE I
SENSING PERFORMANCE (AUC) ON DIFFERENT TRANSFER TASKS

Transfer Task	OR	AND	MRC	TCA	SVM	MLP	DTCS
QPSK → Gaussian	0.905	0.742	0.911	0.850	0.924	0.947	**0.972**
Gaussian → QPSK	0.921	0.754	0.924	0.861	0.937	0.956	**0.983**
PAM4 → Gaussian	0.769	0.556	0.775	0.587	0.786	0.780	**0.827**
Gaussian → PAM4	0.923	0.684	0.904	0.759	0.892	0.935	**0.966**
PAM4 → 16-PSK	0.883	0.661	0.891	0.735	0.885	0.897	**0.929**
16-PSK → PAM4	0.885	0.650	0.877	0.856	0.893	0.888	**0.917**
PAM4 → QPSK	0.858	0.685	0.861	0.763	0.858	0.872	**0.901**
QPSK → PAM4	0.897	0.640	0.902	0.883	0.896	0.904	**0.956**
Average	0.880	0.672	0.881	0.787	0.884	0.897	**0.931**

TABLE II
AVERAGE RUNNING TIME ON DIFFERENT TRANSFER TASKS

Time (s)	OR	AND	MRC	TCA	SVM	MLP	DTCS
Training	-	-	-	2.340	1.062	2.186	2.773
Test	0.490	0.498	0.417	0.121	0.085	**0.054**	**0.054**
Sum	0.490	0.498	**0.417**	2.461	1.147	2.240	2.827

TABLE III
SENSING PERFORMANCE (AUC) IN TERMS OF DIFFERENT ε AND FIXED $\gamma = 1$ ON THE TRANSFER TASK OF 16-PSK → 16-PSK

Parameters	OR	AND	MRC	TCA	SVM	MLP	DTCS
$\varepsilon = 4 \rightarrow \varepsilon = 2$	0.892	0.694	0.902	0.859	0.914	0.916	**0.948**
$\varepsilon = 4 \rightarrow \varepsilon = 4$	0.883	0.661	0.878	0.842	0.895	0.901	**0.962**
$\varepsilon = 4 \rightarrow \varepsilon = 6$	0.810	0.653	0.815	0.826	0.856	0.923	**0.957**
$\varepsilon = 4 \rightarrow \varepsilon = 8$	0.598	0.591	0.643	0.584	0.525	0.658	**0.819**

TABLE IV
SENSING PERFORMANCE (AUC) IN TERMS OF DIFFERENT γ AND FIXED $\varepsilon = 4$ ON THE TRANSFER TASK OF 16-PSK → 16-PSK

Parameters	OR	AND	MRC	TCA	SVM	MLP	DTCS
$\gamma = 1 \rightarrow \gamma = 1$	0.883	0.661	0.878	0.842	0.895	0.901	**0.962**
$\gamma = 1 \rightarrow \gamma = 2$	0.884	0.728	0.841	0.834	0.867	0.894	**0.955**
$\gamma = 1 \rightarrow \gamma = 3$	0.895	0.792	0.912	0.816	0.853	0.887	**0.946**
$\gamma = 1 \rightarrow \gamma = 4$	0.913	0.795	0.919	0.672	0.725	0.873	**0.942**

To compare the computational burden complexity of each technique, we obtain the running time for each method during the training and testing phases on the eight transfer tasks in Table I. Table II shows the average running time in terms of training, test, and the sum values averaged over five experiments. OR, AND, and MRC methods directly work on the test data set without training running time. From the sum values, it can be observed that the traditional CSS methods (OR, AND, and MRC) outperform the machine learning methods (TCA, SVM, MLP, DTCS). However, in the test phase, it is worth noting that machine learning methods perform better than the traditional CSS methods, and DTCS performs the best. They can directly apply the trained models to detect the presence of signals. In practical cases, we can train the machine learning models offline and perform the detection online.

To observe the performance in terms of different channel parameters for a given modulation, we conduct the experiments on the transfer task of 16-PSK → 16-PSK, where the source and target domains are 16-PSK signals with different channel parameters. We also fix $N_{su} = 12$ and $n_s = n_t = 2,000$. Table III shows the sensing performance in terms of different ε and fixed $\gamma = 1$. Table IV shows the sensing performance in terms of different γ and fixed $\varepsilon = 4$. From the results of Tables III and IV, it can be noted that DTCS performs better in the case where the source and target domains have the similar parameters than that in the other cases. The smaller differences between the source and target domains, the better the trained models can perform on the target domains.

VI. CONCLUSION

In this letter, we proposed a novel deep transfer cooperative sensing approach named DTCS. DTCS does not require any labeled unknown signals and can transfer the knowledge of known signals to detect the presence of the unknown signals in an unsupervised fashion. By incorporating deep transfer learning into the CSS scheme, we enhance the generalization ability and robustness of DTCS. From the simulations, the effectiveness of DTCS has been demonstrated. In the future, we would like to extend our proposed method to a more practical scenario, where we can have different numbers of secondary users in the CR network.

REFERENCES

[1] L. Li, H. Jiang, and H. He, "Imbalanced learning for cooperative spectrum sensing in cognitive radio networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Waikoloa, HI, USA, 2019, pp. 1–6.
[2] C. H. A. Tavares, J. C. Marinello, M. L. Proenca Jr., and T. Abrao, "Machine learning-based models for spectrum sensing in cooperative radio networks," *IET Commun.*, vol. 14, no. 18, pp. 3102–3109, Nov. 2020.
[3] A. Kumar, S. Saha, and K. Tiwari, "A double threshold-based cooperative spectrum sensing with novel hard-soft combining over fading channels," *IEEE Wireless Commun. Lett.*, vol. 8, no. 4, pp. 1154–1158, Aug. 2019.
[4] Y. Fu and Z. He, "Entropy-based weighted decision combining for collaborative spectrum sensing over Byzantine attack," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1528–1532, Dec. 2019.
[5] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: Cooperative spectrum sensing based on convolutional neural networks," *IEEE Trans. Veh. Technol.*, vol. 68, no. 3, pp. 3005–3009, Mar. 2019.
[6] B. Shang and L. Liu, "Machine learning meets point process: Spatial spectrum sensing in user-centric networks," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 34–37, Jan. 2020.
[7] J. Gao, X. Yi, C. Zhong, X. Chen, and Z. Zhang, "Deep learning for spectrum sensing," *IEEE Wireless Commun. Lett.*, vol. 8, no. 6, pp. 1727–1730, Dec. 2019.
[8] Q. Peng, A. Gilman, N. Vasconcelos, P. C. Cosman, and L. B. Milstein, "Robust deep sensing through transfer learning in cognitive radio," *IEEE Wireless Commun. Lett.*, vol. 9, no. 1, pp. 38–41, Jan. 2020.
[9] H. Jiang, L. Li, H. He, and L. Liu, "Evolutionary search for energy-efficient distributed cooperative spectrum sensing," in *Proc. IEEE Int. Conf. Comput. Netw. Commun. (ICNC)*, Big Island, HI, USA, 2020, pp. 567–571.
[10] J. Ma, G. Zhao, and Y. Li, "Soft combination and detection for cooperative spectrum sensing in cognitive radio networks," *IEEE Trans. Wireless Commun.*, vol. 7, no. 11, pp. 4502–4507, Nov. 2008.
[11] L. Li, H. He, J. Li, and G. Yang, "Adversarial domain adaptation via category transfer," in *Proc. IEEE Int. Joint Conf. Neural Netw. (IJCNN)*, Budapest, Hungary, 2019, pp. 1–8.
[12] L. Li, Z. Wan, and H. He, "Dual alignment for partial domain adaptation," *IEEE Trans. Cybern.*, early access, Apr. 29, 2020, doi: 10.1109/TCYB.2020.2983337.
[13] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of Wasserstein GANs," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2017, pp. 5767–5777.