Research Article

Domain Adaptation-Based Automatic Modulation Recognition

Tong Li¹ and Yingzhe Xiao²

¹Department of Computer Science and Information, Shanxi Engineering Vocational College, Taiyuan 030032, China

²College of Information and Computer, Taiyuan University of Technology, Taiyuan 030600, China

Correspondence should be addressed to Yingzhe Xiao; xiaoyingzhe@tyut.edu.cn

Received 10 August 2021; Revised 21 August 2021; Accepted 24 August 2021; Published 11 October 2021

Academic Editor: Bai Yuan Ding

Copyright © 2021 Tong Li and Yingzhe Xiao. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Deep learning-based Automatic Modulation Recognition (AMR) can improve the recognition rate compared with traditional AMR methods. However, in practical applications, as training samples and real scenario signal samples have different distributions in practical applications, the recognition rate for target domain samples can deteriorate significantly. This paper proposed an unsupervised domain adaptation based AMR method, which can enhance the recognition performance by adopting labeled samples from the source domain and unlabeled samples from the target domain. The proposed method is validated through signal samples generated from the open-sourced Software Defined Radio (SDR) GNU Radio. The training dataset is composed of labeled samples in the source domain and unlabeled samples in the target domain. In the testing dataset, the samples are from the target domain to simulate the real scenario. Through the experiment, the proposed method has a recognition rate increase of about 88% under the CNN network structure and 91% under the ResNet network structure.

1. Introduction

Automatic Modulation Recognition (AMR) plays an important role in the wireless communication field. For example, in cognitive radio tasks [1, 2], the transmission source can adaptively change modulation scheme according to the current channel state to improve communication efficiency. Therefore, the receiver needs to do AMR as a basic procedure prior to demodulation. In spectrum sensing tasks [3, 4], modulation information based on AMR is considered as basic information for describing the spectrum conditions, which can provide references for further spectrum management.

Traditional AMR methods have adopted carefully handcrafted features for classification [5, 6]. Normally, there are two procedures. Firstly, features with explicit meanings are extracted to form a feature vector or feature space. And then, according to the features extracted from the baseband sample signals, the modulation scheme is recognized through some machine learning based classification methods. The features from the first phase include the following: (1) spectral based feature, which can exploit

spectral properties from different signal components; (2) wavelet based features from wavelet transformation; (3) high-order statistics, which are effective for classification of digital modulations; (4) cyclic features from cyclostationary analysis. In the second phase, many existing classifiers are adopted, including Support Vector Machine (SVM), Decision Tree (DT), and k Nearest Neighbor (KNN) [7–9]. These traditional methods are well described and proved to be effective in AMR tasks. However, many recent researches have shown that Deep Learning (DL) based methods can also be adopted to the AMR tasks [10, 11]. Even with a simple few-layer Convolutional Neural Network (CNN), the classification performance is better than the traditional handcrafted-feature based methods.

Deep learning based methods adopt training samples to enable automatic learning of effective features for modulation classification. These methods can be categorized according to different aspects including samples and network models. From the aspect of samples adopted, some methods adopt In-phase and Quadrature (IQ) samples in the time domain, while others adopt spectrograms. For the first type, as the modulation

information including phase and amplitude information in the IQ samples remains intact, it is in principle capable of classifying any modulation types. Also, this type of methods has the advantages of less input data volume, smaller set of network parameters, and so on, while, for the methods adopting spectrograms as the input, limitations lie in the classification of Phase Shift Keying (PSK) signals, due to the loss of phase information from the spectrograms. However, it is much easier to transfer the training models or parameters in image recognition tasks to AMR tasks, because spectrograms can be regarded as special image samples. The authors in [12] proposed a method that has a two-classification-stage structure. In the first stage, the spectrograms are adopted to discriminate PSK signals and other modulation signals. Then, in the second stage, IQ baseband samples are adopted for modulation classification within the PSK signals. From the aspect of network models, many try to enhance performance by adopting different network models. The authors in [13] have proposed a simple CNN network based framework, which is proved to have better performance than traditional handcrafted feature based methods. The authors in [14] adopted a Recurrent Neural Network (RNN) model for AMR tasks. Paper [15] gave comprehensive performance comparisons between many popular network models in AMR. The models include Residual Network (ResNet), inception network, and Convolutional Long-short Deep Neural Network (CLDNN). Among these networks, the authors conclude that the CLDNN model has the best recognition rate. [16, 17] proposed an optimized ternarized CNN, which is implemented in FPGA based hardware design. The method can bring a high data throughput with low classification latency at the cost of a few recognition rate reduction. The authors in [18] proposed a complex-valued network, which redefine the convolution layer, pooling layer, and so on from Euclidean spaces to non-Euclidean signal spaces utilizing weighted Fréchet mean. The complex-valued model can achieve state-of-the-art performance with much less parameter size (less than 1% of CNN). The network can significantly accelerate training process and can have less computational burden. The authors in [19] proposed a radio transformer network, which added a transformer structure to a common CNN network to cope with the problem of time, frequency, and phase offsets in the IQ baseband samples. The method basically introduced the notion of synchronization to design the transformer, which in principle can be regarded as a way to adopt a priori knowledge for classification. A fine-tuning based transfer learning method is proposed in [20], which has adopted partially labeled target domain data samples for tuning the parameters of the trained model with source domain samples. A complex-valued network is proposed in [21] for modulation recognition. Compared with its real-valued counterparts, which process the data in the Euclidean space, the complex-valued network proposes to process the data on the Riemannian manifold, which can better preserve the geometric structure of the complex-

valued signals. The experimental results show that it can enhance performance compared to real-valued networks. However, it is much more difficult to train than real-valued networks.

The mentioned methods, although effective, do not take into consideration that the signal samples can have significantly different distributions among training and actual classification. The reasons are twofold: (1) the channel varies between the training samples and samples in real implementation; (2) the estimated parameters differ between training samples and samples in real implementation; e.g., the estimated bandwidth and center frequency can affect the process of sampling. The different distributions between the training samples and the samples in real implementation can significantly deteriorate the performance. Therefore, strategies should be added to alleviate the problem. There are two basic strategies: (1) trying to enhance the training dataset by introducing signal samples with new channels and estimation parameters; (2) adopting transfer learning to transfer existing network to adapt new data with different distributions. For the first strategy, as the properties in new channels can be unknown, it is very hard to ensure that the enhanced dataset has the same distribution with the samples in real implementation. Also, many redundancies can be introduced to the dataset, which can make it significantly harder for the training process to converge. As for the second strategy, transfer learning does not necessarily need a large number of labeled samples with new distribution. Therefore, transfer learning is much appropriate for solving the mentioned problem.

This paper proposed an unsupervised domain adaptation based AMR method, which can enhance the recognition performance by adopting labeled samples from the source domain and unlabeled samples from the target domain. The proposed method can cope with the problem that, in real modulation recognition tasks, the channels and parameters (including the band width and center frequency) are varying. The proposed method has the following advantages: (1) the network structure has only added a small-scale subnetwork (the domain discriminator); thus, the changes along with the complexity increase compared with the original neural network are minor; (2) the method is unsupervised, meaning that, for the samples in the target domain, the label information is exemplified. This is especially suitable for real implementations, where samples without labels can be acquired much easier than those with labels; (3) the proposed method is compatible with existing network and, thus, can inherit the favorable features of existing network structure. The proposed method is validated adopting signal samples generated from the open-sourced Software Defined Radio (SDR) framework GNU Radio. In the training dataset, there are labeled samples in the source domain and unlabeled samples in the target domain. In the testing dataset, the samples are from the target domain to simulate the real scenario. Through the experiment, the proposed method has a recognition rate increase of about 88% under the CNN network structure and 91% under the ResNet network structure.

2. Related Works

In this part, firstly, we give a simplified introduction about the signal models and different factors, which can affect the performance for AMR. Then, some classical AMC methods including feature based and deep learning based methods are introduced.

2.1. Simplified Signal Model. A simplified received signal model can be described as follows:

$$r(n) = e^{j2\pi f_{\text{bias}} nT * n_{\text{crw}}(t)} \sum_{i=1}^{N-1} s(i)h(nT - iT - \epsilon T) + w(n), \quad (1)$$

where f_{bias} denotes the frequency offset when down converting to baseband, T denotes the symbol interval, $n_{\text{crw}}(t)$ denotes the time varying residual carrier random walk, $s(i)$ denotes the sending symbol sequence, $h(t)$ denotes the channel impulse response, ϵ denotes the symbol rate deviation, and $w(n)$ denotes white Gaussian noise. Noting that the mentioned model is only a simplified one. In real scenario applications, the signal propagation model can be very complex and can be time-frequency varying. Therefore, it is analytically hard to have a closed form of signal model. Therefore, when evaluating an AMR method, it is necessary to take into consideration whether changing factors will affect the robustness of the method. For traditional handcrafted feature based AMR methods, as the features are inferred from a enforced simple signal model assumption, the performance can deteriorate significantly when the assumption is not fulfilled or encountered changing factors.

In this paper, we have considered the four factors that can affect the IQ signal samples. The four factors are listed in detail in [22]. Here, only a simplified description is given. The four factors are as follows: (1) sample rate offset, which denotes the sample clock offset in the receiver side; (2) center frequency offset, which normally resulted from the errors in carrier frequency estimation; (3) selective fading model, which can be regarded as effects of multipath propagation; (4) white noise, which affects the received signal noise ratio.

2.2. Classical Deep Learning-Based Modulation Recognition. As deep learning can enable feature learning from training data, it can also be adopted in learning signal sample features for AMR applications. Compared with handcrafted feature based methods, the features learned from training samples are more adopted for classification and may not clearly related to expert features. Here, some classical deep learning based methods are listed, which has shown better classification performance than traditional handcrafted feature based methods.

A classical CNN-based method is proposed in [13], which apply a CNN network for classification. Its network structure is shown in Figure 1, which has two convolution layers and two dense layers. The convolutional layers in the method can be regarded as “matched filter” in the receiver. If the training samples consisted of samples with different

FIGURE 1: The CNN structure (left) and the CLDNN structure (rights) for AMR.

pulse shaping filter parameters, such as different roll-off factors and different types of shaping filter, the convolutional layers can cope with changes and recover constellation information. The convolutional layer can also recover the signals samples from different channel effects (act as compensation filter) if there are enough training samples from different channels.

Another classical deep architecture for modulation recognition is the Convolutional Long-short term Deep Neural Networks (CLDNN) network [15]. As shown in Figure 1, the network consisted of two convolutional layers and two recurrent layers (adopting Long Shot Term Memory LSTM cells). The advantage for this structure is that it has considered not only the feature extraction in different scales, but also features extraction along time. This structure can correspond to procedures for common demodulation. The convolutional layers can serve as the matched filter, and the LSTM cells can serve as the synchronizer and sampler in time based processing. The new structure with layers corresponding to standard demodulators is proved to have better recognition performance than the CNN structure mentioned previously. However, the cost is a significant increase in training time.

In paper [19], a radio transformer based network is proposed. The radio transformer, inspired by spatial transformer network adopted for image recognition tasks, has added a parameter regression substructure to a simple CNN network. This substructure, noted as transformer, is designed to estimate the received radio signal related parameters, such as time offset, sampling offset, phase offset,

and frequency offset. The transformer, then, is adopted to compensate these offsets to enable better classification. As the radio transformer network has adopted known models, including radio signal synchronization and normalization, the recognition performance is enhanced.

3. Materials and Methods

As mentioned previously, in actual modulation recognition tasks, the received signal samples may encounter channel effects, which are different from those of the training sample. In other words, the distribution of the training samples is different from that of the testing samples. Here, we call the labeled signal samples the source domain, while the unlabeled signal samples encountered in real implementation will be called the target domain. The source domain differs from the target domain due to different radio propagation channels. In this paper, we propose a domain adaptation based network, which can adopt labeled source domain samples and unlabeled target domain samples for modulation recognition. In the context of this paper, the relationship of source domain and target domain is shown in Figure 2. In this section, the proposed method is introduced from the following three parts: the model details, the training and optimization process, and the dataset generation details.

3.1. Model Description. The proposed network structure has a model shown in Figure 3. As can be seen, the network is composed of three substructures: the feature extraction subnetwork, the modulation prediction subnetwork, and the domain classifier subnetwork. The feature extraction along with the modulation prediction subnetwork can be regarded as a traditional neural network, where the feature extraction subnetwork can be composed of several convolutional layers, and the modulation prediction subnetwork can be composed of a dense layer with softmax layer. Except for the layers mentioned above, the feature extraction together with the modulation prediction subnetwork can make up any classical neural network, such as a ResNet and CLDNN network. Inspired by the method proposed in [23], the domain classifier subnetwork is added to the existing neural network, which can be adopted for domain adaptation. Note that the goal of domain adaptation is that the features extracted are less domain sensitive. Equivalently, the extracted features are shared within both the source domain and the target domain. We will discuss the three subnetworks in the following descriptions.

- (1) The feature extraction network. The subnetwork can be denoted as $H_f(\cdot; \theta_f)$, where the subindex f denotes that it is related to feature extraction, and θ_f denotes the network parameters in it. In our implementation, the structure of the feature extractor can be similar to any existing networks. For example, it can make up of a few convolutional layers (similar to CNN network), a few residual diagram structures (similar to ResNet), and a few convolutional layers along with a few LSTM cells (similar to CLDNN).

FIGURE 2: Source and target domain in the context of modulation recognition.

FIGURE 3: The network structure of the proposed model.

- (2) The modulation prediction network. This can be denoted as $H_m(\cdot; \theta_m)$, where the subindex m denotes that it is related to modulation prediction, and θ_m denotes the network parameters for prediction. The modulation prediction structure is similar to many neural networks, with some linear dense layers and softmax layer for prediction. Here, we have adopted the classical cross-entropy loss L_m as the loss function from this structure, which can be denoted as

$$L_m = - \sum c_i \log p_i, \quad (2)$$

where c_i denotes the one-hot encoded results adopting known labels, while p_i denotes the output of the softmax layer. As the cross-entropy loss is commonly adopted in many deep learning based methods, only a simple introduction is given here.

- (3) The domain classifier substructure. It can be denoted as $H_d(\cdot; \theta_d)$, where θ_d denotes the network parameters. This substructure can be added to many existing neural networks for domain adaptation. In this structure, we want the network to tell whether the input data are from the source or target domain,

so that the network can be diverged when training. However, we also want the domain classifier to feedback to the feature extraction subnetwork, so that the features extracted are shared in both the source and target domains. To make the mentioned two goals happen at the same time, a gradient reverse layer is added to the domain classifier. Mathematically, this means finding the optimal θ_d , which makes the domain classifier loss L_d minimum, while finding the optimal θ_f , which makes the samples domain invariant. L_d denotes the domain classifier loss function. It is also a cross-entropy loss similar to L_m . The difference is that, for L_m , the classification types equal the number of modulation types, while, for L_d , the classification type number is 2, denoting the source domain type or the target domain type. The detailed information for the reverse gradient layer will be introduced in the next section.

As mentioned previously, the model has a three-subnetwork structure, including the feature extraction, modulation prediction, and domain classifier subnetwork. The forward propagation function can be denoted as $H_f(\cdot; \theta_f)$, $H_m(\cdot; \theta_m)$ and $H_d(\cdot; \theta_d)$, respectively. The cost function is listed as follows:

$$F_{\text{loss}}(\theta_f, \theta_m, \theta_d) = L_m(\theta_f, \theta_m) - \eta L_d(\theta_f, \theta_d), \quad (3)$$

which are composed of two terms. The term $L_m(\theta_f, \theta_m)$ denotes the cross-entropy cost function from modulation recognition, and the term $L_d(\theta_f, \theta_d)$ denotes the domain classification error. η is a parameter denoting the respective weight for the two terms responsible for the overall cost function. Worth noting is that there is a minus sign in front of the domain classifier related cost function, which means that we want the domain classification error to grow larger during training. Again, this can make the feature extraction subnetwork prone to favor features shared by both source and target domain. The two terms of the cost function can be expanded as

$$\begin{aligned} L_m(\theta_f, \theta_m) &= \sum_i L_{m,i}(H_m(H_f(\mathbf{s}_i; \theta_f); \theta_m), m_i), \\ L_d(\theta_f, \theta_d) &= \sum_i L_{d,i}(H_d(H_f(\mathbf{s}_i; \theta_f); \theta_d), d_i), \end{aligned} \quad (4)$$

where the subindex i denotes the corresponding training signal sample index, $\mathbf{s}_i$ denotes the training sample, m_i denotes the modulation label the sample, and d_i denotes the domain label of the corresponding sample. As can be seen from the overall cost function, signals samples of different modulation types and domains are all needed for training.

3.2. Training and Optimization. The overall structure of the proposed method is shown in Figure 3. The key for the proposed method is the domain adaptation subnetwork (also denoted as the domain classification subnetwork). As mentioned, to make the network converge, the domain classification loss should be minimized during optimization. However, this will courage the feature extraction

subnetwork to learn features effective to discriminate the data domain, which is not our intention. To make the feature extraction subnetwork learn shared features in source and target domain, while keeping the network converged, a gradient reversal layer is added between the feature extraction subnetwork and the domain classification subnetwork. This layer is a special one with no parameters to train and optimize, while it only has a hyperparameter η . The η parameter will be multiplied with a minus during optimization shown as follows:

$$\theta_f = \theta_f - \mu \left(\frac{\partial L_m}{\partial \theta_f} - \eta \frac{\partial L_d}{\partial \theta_f} \right), \quad (5)$$

$$\theta_m = \theta_m - \mu \frac{\partial L_m}{\partial \theta_m}, \quad (6)$$

$$\theta_d = \theta_d - \mu \frac{\partial L_d}{\partial \theta_d}. \quad (7)$$

Equations (5)–(7) have shown the updates of the parameters from the subnetworks during training, where μ denotes the update rate, θ_f , θ_m and θ_d denote the feature extraction parameters, the modulation classification parameters, and the domain classification parameters, and the partial signs denote the calculated partial gradient of different losses. From the mentioned parameter update equations, we can see that the mathematical expression of the gradient reversal layer can be written as

$$\begin{aligned} \mathbf{R}(\mathbf{x}) &= \mathbf{x}, \\ \frac{\partial \mathbf{R}}{\partial \mathbf{x}} &= -\eta \mathbf{I}. \end{aligned} \quad (8)$$

Note that, here, $\mathbf{I}$ denotes the identity matrix. The backward expression has a partial gradient of $-\eta \mathbf{I}$. The reason is that the parameter η is a hyperparameter and is not related to the losses during optimization.

4. Results and Discussion

4.1. Dataset Generation. The dataset adopted here is generated from the open-sourced Software Defined Radio GNU radio. Here, we are inspired by the open-sourced dataset from the authors in [22]. Based on that dataset, we enhanced it with different frequency offset in the carrier, along with different channel responses. By generating data with different carrier frequency offsets and channel responses, the target domain data are produced, which can be adopted to validate the proposed method. The modulation dataset is composed of signal samples with 5 different Signal Noise Ratios (SNR), from 0 to 20 dB. The dataset includes 9 commonly seen modulation types: AM, FM, GFSK, BPSK, QPSK, 8PSK, OQPSK, 8QAM, and 16QAM. Each signal sample is made of 128 baseband IQ samples. As mentioned, the differences between the source and target domain data are twofold as shown in Table 1. The differences between the source and target domain for the dataset mainly lie in three aspects: (1) for the source domain, no carrier frequency

TABLE 1: Source domain and target domain differences.

	Source domain	Target domain
Frequency offset	No	Yes
Channel condition	Gaussian channel	Rayleigh channel
With labels	Yes	No

offset is added, while it is added in the target domain. Specifically, the maximum carrier frequency offset added to the target domain is $0.05 * \pi$ in the digital frequency. The mentioned situation is commonly seen when performing signal detection in a wide band and down convert it to baseband; (2) the source domain is with a typical Gaussian channel, while the target domain is with the Rayleigh channel; (3) during training, the source domain data are with labels, while the target domain data are without labels. The differences between the source and target domain data are listed in Table 1.

Figure 4 has shown some signal samples from the dataset. As can be seen, the signal samples are baseband IQ data. During the preprocessing procedure, all the signal samples are normalized according to the total energy.

4.2. Hyper Parameter Settings. As mentioned previously, both source and target domain samples are adopted in the training processes. The difference is that, for the target domain training samples, the samples are without modulation type labels.

The equation from 1 has a hyperparameter η_p , which denotes the ratio of the two losses: the cost function modulation recognition and the cost function of domain classifier. In this paper, the parameter η_p is set as

$$\eta_p = \frac{2}{1 + \exp(-\varepsilon p)} - 1, \quad (9)$$

where η_p denotes the hyperparameter (it has a value between 0 and 1), and p denotes the real time recognition rate. To set the value of η_p like this makes the network much easier to train. The reason is as follows: in the beginning, when the recognition rate p is near 0, the training process only aims at having higher modulation recognition rate. When p is higher, the network cost function also starts to consider the effects of domain classification. Parameter ε denotes how fast the training goal changes from only higher recognition rate to a balance between higher recognition rate and domain relevance. To choose the best hyperparameter, we have compared the recognition rate and the training time of different values of ε . The results are shown in Figure 5. We can see that the training time reaches the minimum when $\varepsilon = 15$, while the recognition rate reaches its maximum when $\varepsilon = 20$. As a matter of fact, we set the value of ε as 20, to make the network have a maximum recognition rate.

5. Methods Comparisons

In order to fully illustrate the effectiveness of the proposed method, in the experiment, the traditional CNN network and ResNet network were changed. The domain

classification subnetwork is added according to the method described above. During the training of the network, source domain samples with labels and unlabeled target domain samples are used. On the basis of the reconstructed network, the method of non-transfer learning, the supervised transfer learning method, and the fine-tuning based transfer learning method proposed in [20] are compared. The test datasets adopted in the experiment for method comparison are the same, including samples of both the source and target domains. The training sets adopted in the four methods compared are different, and their characteristics are shown in Table 2. Among them, the method proposed in this paper has adopted source domain samples with modulation type labels and target domain samples without modulation type labels for training. The non-transfer learning method has only adopted source domain samples with modulation type labels for training. For the supervised transfer learning method, both labeled source domain samples and labeled target domain samples are adopted. The fine-tuning based transfer learning method firstly adopts labeled source domain data for training and then a small part of labeled source domain data for parameter fine-tuning. Table 2 gives recognition rate comparisons between the mentioned four methods using CNN and ResNet as the basic neural network.

As can be seen from Table 3, ResNet, as a whole, has a higher recognition rate than the CNN network, which is consistent with previous publications. Among the CNN-based methods, the method proposed in this paper has increased the recognition rate by about 48% and 6% compared with the method without transfer learning and the method of fine-tuning based transfer learning method. This has demonstrated that the proposed unsupervised transfer learning method (here, the unsupervised notion refers to the source domain data that are unlabeled) can make full use of the distribution of the unlabeled target domain data to adjust the neural network, so that the feature extraction subnetwork can adapt to the input of both source and target domain samples, which are distributed differently. However, the fine-tuning based method only relies on a few labeled target domain data and cannot make full use of all target domain data. Therefore, its recognition rate is lower than that of the method proposed in this paper. Compared with the supervised method, the recognition rate of the method proposed in this paper is only 3% lower, indicating that the method proposed in this paper has almost reached the upper limit of the recognition rate (i.e., both labeled source and target domains are used for training). In practical applications, usually only a small amount of labeled source domain data and a large amount of unlabeled target domain data can be obtained. Therefore, the proposed method is more practical than the supervised method. In the comparison of methods based on ResNet, the method in this paper has improved the recognition rate by about 46% and 8% compared with the method without transfer learning and the method of fine-tuned transfer learning, respectively. Compared with the recognition rate of supervised methods, ours is only 3% lower, which denotes a similar conclusion.

FIGURE 4: Several signal samples of different modulation types in the dataset.

FIGURE 5: The effects of different ϵ over training time and recognition rate.

TABLE 2: The differences between the training dataset including source and target domain in the four methods.

Method	Source domain	Target domain
Proposed method	With label	Without label
Without transfer learning	With label	No data needed
Fine-tuning based method	With label	Partially with labels
Supervised method	With label	With label

TABLE 3: The recognition rate comparisons between the four methods.

Method	CNN based (%)	ResNet based (%)
Proposed method	88	91
Without transfer learning	40	45
Fine-tuning based method	82	83
Supervised method	91	94

6. Conclusions

In the application of deep learning based modulation recognition, the purpose of improving the recognition rate is often achieved by changing the deep neural network structure. However, in practical applications, training samples and real scenario signal samples in practical applications have different distributions due to different channels and different frequency offsets. This can make the neural network achieve a high recognition rate for source domain samples, while having a poor recognition rate for target domain samples. This paper proposed an adaptive unsupervised learning modulation recognition method, which can use labeled data in the source domain and unlabeled target domain data for training and can be directly implemented adopting existing deep learning network with minor structural changes. Through the simulation data set generated by the open source GNU Radio software, it proves the proposed method in this paper, compared with the method without transfer learning and the method with fine-tuning, and the recognition rate has increased by 48% and 6%, respectively, under the condition of CNN as the basic network. Under the condition of ResNet as the basic network, the recognition rate has increased by 46% and 8%, respectively. Moreover, the method proposed in this paper is close to the upper limit of the modulation recognition rate after training adopting both labeled target and labeled source domain data. Under the condition of CNN and ResNet as the basic network, it only drops by 3% and 3%, respectively.

Data Availability

The RadioML dataset is publicly available.

Conflicts of Interest

There are no conflicts of interest among the authors.

Acknowledgments

This work was sponsored in part by 2020 Science and Technology Innovation Project of Colleges and Universities in Shanxi Province (2020L0764).

References

- [1] H. Anandakumar and K. Umamaheswari, "Supervised machine learning techniques in cognitive radio networks during cooperative spectrum handovers," *Cluster Computing*, vol. 20, no. 2, pp. 1505–1515, 2017.
- [2] F. A. Awin, Y. M. Alginahi, E. Abdel-Raheem, and K. Tepe, "Technical issues on cognitive radio-based Internet of Things systems: a survey," *IEEE access*, vol. 7, pp. 97887–97908, 2019.
- [3] W. Lee, M. Kim, and D.-H. Cho, "Deep cooperative sensing: cooperative spectrum sensing based on convolutional neural networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 3, pp. 3005–3009, 2019.
- [4] M. S. Gupta, K. Kumar, and K. Kumar, "Progression on spectrum sensing for cognitive radio networks: a survey, classification, challenges and future research issues," *Journal of Network and Computer Applications*, vol. 143, pp. 47–76, 2019.
- [5] J. Huang, S. Huang, Y. Zeng, H. Chen, S. Chang, and Y. Zhang, "Hierarchical digital modulation classification using cascaded convolutional neural network," *Journal of Communications and Information Networks*, vol. 6, no. 1, pp. 72–81, 2021.
- [6] D. H. Al-Nuaimi, I. A. Hashim, I. S. Zainal Abidin, L. B. Salman, and N. A. Mat Isa, "Performance of feature-based techniques for automatic digital modulation recognition and classification-A review," *Electronics*, vol. 8, no. 12, p. 1407, 2019.
- [7] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*, John Wiley & Sons, Hoboken, NJ, USA, 2015.
- [8] E. Azzouz and A. K. Nandi, *Automatic Modulation Recognition of Communication Signals*, Springer, New York, NY, USA, 2013.
- [9] F. Wang and X. Wang, "Fast and robust modulation classification via Kolmogorov-Smirnov test," *IEEE Transactions on Communications*, vol. 58, no. 8, pp. 2324–2332, 2010.
- [10] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.
- [11] Y. Tu and Y. Lin, "Deep neural network compression technique towards efficient digital signal modulation recognition in edge device," *IEEE Access*, vol. 7, pp. 58113–58119, 2019.
- [12] X. Zha, X. Qin, Y. Zhou, and H. Peng, "Power of deep learning for amplitude-phase signal modulation recognition," in *Proceedings of the 2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, May 2019.
- [13] T. J. O'Shea, J. Corgan, and T. Charles Clancy, "Convolutional radio modulation recognition networks," in *Proceedings of the International Conference on Engineering Applications of Neural Networks*, Springer, Aberdeen, UK, September 2016.
- [14] C. Yang, Z. He, Y. Peng, Y. Wang, and J. Yang, "Deep learning aided method for automatic modulation recognition," *IEEE Access*, vol. 7, pp. 109063–109068, 2019.
- [15] N. E. West and T. O'Shea, "Deep architectures for modulation recognition," in *Proceedings of the 2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, March 2017.
- [16] S. Tridgell, D. Boland, P. H.W. Leong, and S. Siddhartha, "Real-time automatic modulation classification," in *Proceedings of the 2019 International Conference on Field-Programmable Technology (ICFPT)*, December 2019.
- [17] S. Tridgell, D. Boland, P. H.W. Leong, R. Kastner, A. Khodamoradi, and S. Siddhartha, "Real-time automatic modulation classification using RFSoC," in *Proceedings of the 2020 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, May 2020.

- [18] R. Chakraborty, Y. Xing, and S. Yu, “SurReal: complex-valued learning as principled transformations on a scaling and rotation manifold,” 2019, <https://arxiv.org/abs/1910.11334>.
- [19] T. J. O’Shea, L. Pemula, D. Batra, and T. C. Clancy, “Radio transformer networks: attention models for learning to synchronize in wireless systems,” in *Proceedings of the 2016 50th Asilomar Conference on Signals, Systems and Computers*, November 2016.
- [20] Z. Lei, “Research on deep learning based modulation recognition technologies,” Master thesis, University of Electronic Science and Technology of China, Chengdu, China, 2019.
- [21] R. Chakraborty, J. Wang, and S. X. Yu, “Sur-real: frechet mean and distance transform for complex-valued deep learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, Long Beach, CA, USA, June 2019.
- [22] O’shea, J. Timothy, and W.. Nathan, “Radio machine learning dataset generation with gnu radio,” *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, 2016.
- [23] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in *Proceedings of the International Conference on Machine Learning*, PMLR, Lille, France, July 2015.