Deep Adversarial Neural Network Based on Transformer Encoder for Specific Emitter Identification Under Varying SNR

Chang Liu[1], Zhigang Li[1], Haoran Zha[1], Qiao Tian[2], and Meiyu Wang[1](✉)

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin, China
hrbeu_meiyu@hrbeu.edu.cn

[2] College of Computer Science and Technology, Harbin Engineering University, Harbin, China

Abstract. Specific Emitter Identification (SEI) is a technology that distinguishes between the unique hardware differences inherent in different emitters. In practical applications, due to the lack of labeled datasets, transfering labeled source domains to unlabeled target domains is critical, however, individual signals of different emitters will be disturbed by different degrees of noise during the propagation process, the model performance degrades due to differences between domains caused by different noises. To solve this challenge, we introduce unsupervised domain adaptation (UDA) to SEI of different noises, the main principle of UDA is to reduce the difference between the labeled source domain and the unlabeled target domain, and learn domain invariant features between the two domains. In this paper, we propose to use domain adversarial neural network (DANN) based on transformer encoder (DANN-Transformer) for SEI of different noises, this domain adaptation behavior achieves adversarial effects by adding new gradient reversal layers, the transformer encoder can better extract the contextual relevance of signals, and provide deeper transferable features. Finally, experiment on the real ADS-B dataset, when the SNR is between -20dB and -5dB, DANN-Transformer shows superior performance compared to other baseline models. In addition, it also has good anti-noise performance and the performance of more than 95% can still be achieved when the number of target domain samples is 200.

Keywords: unsupervised domain adaptation · specific emitter identification · domain adversarial neural network · transformer encoder

This work is supported by the National Natural Science Foundation of ChinaNo: 62201172), the National Key Research and Development Program of China2022YFE0136800). This work is also supported by Key Laboratory of AdvancedMarine Communication and Information Technology, Ministry of Industry and Information Technology, Harbin Engineering University, Harbin, China.

J. Wang (Ed.): MobiMedia 2023, LNICST 576, pp. 374–386, 2025.
https://doi.org/10.1007/978-3-031-60347-1_31

1 Introduction

Specific emitter identification (SEI) is applied in various aspects of real life [1]. It is mainly used in auxiliary authentication, smart radio, ad hoc network, Internet of Things security and so on.

SEI mainly uses the inherent essential difference information about the transmitting source hardware or system extracted from the received signal, that is the Radio Frequency(RF) fingerprint characteristics of different transmitters. There are two main methods of identification:(i) feature engineering, extracting various features of the signal, and selecting an appropri- ate classifier for recognition, and (ii) end-to-end recognition based on deep learning. Traditional feature extraction methods extract RF fingerprint features through instantaneous features, permutation entropy(PE), short-time Fourier transform(STFT), power spectrum density(PSD), and Hilbert transform, and select Support Vector Machine(SVM), K-NearestNeighbor(KNN) and other classifiers for SEI identification. There are also some methods that convert the signal into a graph for representation, and then extract features for SEI [2]. Deep learning technology has many advantages in application and has higher performance compared with traditional technology. Chen [3] introduced the residual block and multi-branch module on the basic CNN architecture to realize the classification and recognition of civil aviation ACARS signals, and the recognition accuracy rate reached 92% under the condition of 10dB. Pan [4] convert the radiation source signal into a Hilbert grayscale image and input it to ResNet for feature extraction, and learns complex high-dimensional features from the original image. In addition, more advanced networks have been studied, such as complex-valued networks [5] and complex convolutional network [6] for SEI. When the training data set is large enough and the labels are sufficient, the neural network-based method can achieve good performance, but in practical applications, most of them are non-cooperative scenarios, and it is impossible to obtain enough data sets with high-quality labels, so In this case, it is necessary to choose unsupervised SEI. In the process of wireless communication, there will be different degrees of noise influence, which will lead to poor classification effect. However, unsupervised SEI seldom considers this situation, so it is necessary to introduce domain adaptation into the field of unsupervised SEI. [7] introduces the latest research progress and future challenges of domain adaptation in the field of 6G wireless communication.

Although unsupervised domain adaptation (UDA) for SEI has gradually received attention, they are rarely applied to SEI. [8] performed continuous wavelet transform (CWT) on the original signal, and then combined with DANN, and achieved good results on SEI at different frequencies. In addition, there is also the application of DANN to specific emitter modulation recognition. In this paper, instead of extracting traditional features from the signal, we directly input it into DANN [9] for end-to-end recognition. This is more realistic than other transfer learning methods, transforming raw signal data into domain space may result in loss of features and distortion.

The main contributions of this paper are the following three points:

– Aiming at the problem of poor domain adaptive performance under different noise environments, adversarial transfer learning is introduced into the field of SEI.
– The proposed SEI framework introduces the Transformer encoder into the feature extraction network of DANN, which can better extract the contextual correlation of signals, learn deep-level transferable features, and have strong robustness.
– Experimental evaluation on the real ADS-B dataset, and the experimental results show that DANN-Transformer has a great improvement over other baseline models at low SNR. We also explore the anti-noise performance of DANN and the effect of the number of samples on SEI identification.

The rest of this paper is organized as follows. In Sect. 2, we mainly introduce the principle and research status of the UDA method. In Sect. 3, we describe the problem to be solved by applying SEI under different noises in the field of UDA. In Sect. 4, we introduce the basic framework of DANN-Transformer, the calculation of loss function, and the basic architecture of the network. In Sect. 5, we evaluate the performance of the DANN-Transformer method in several aspects. In Sect. 6, conclusions and directions for future research are given.

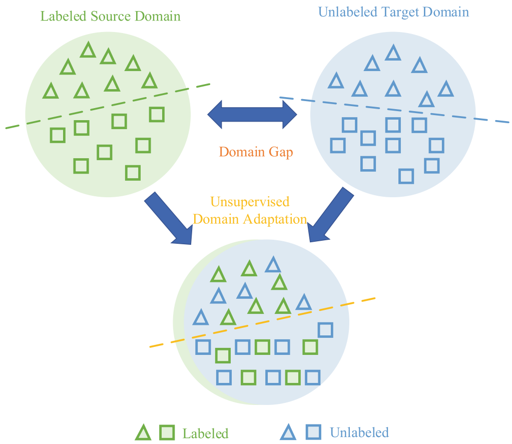


Fig. 1. Unsupervised domain adaptation

2 Unsupervised Domain Adaptation

Compared with traditional signal processing methods, supervised deep learning realizes an end-to-end learning mode, and is the most widely used method in various research fields, but its performance depends on two conditions: (i) high quality labeled training dataset, (ii) the data are identically distributed.

However, reliable labeling of SEI is difficult. Therefore, we need to transfer the model to solve the task that there is not enough labeled data set. When two domain distributions are inconsistent, the performance will drop significantly. For example, when two domains are disturbed by different degrees of noise, the performance of the model after migration cannot be guaranteed.

Regarding the issue above, UDA is a good solution that it solves the problem of no labels in the target domain. DANN is a classic unsupervised domain adaptation method, which introduces the idea of confrontation learning into domain adaptation for the first time. Fig. 1 is a diagram of the unsupervised domain adaptation process. UDA is mainly divided into three types: based on style transfer [10,11], self-training [12,13], based on adversarial learning [14,15].

3 Problem Description

In this section, we define the notation in UDA. We define the dataset as $\mathbf{X}$ and the label set as $\mathbf{Y}$. The source domain of the tag is $\mathbf{D}_{\mathrm{s}}$ and the target domain without labels is $\mathbf{D}_{\mathrm{t}}$. In the $\mathbf{D}_{\mathrm{s}}$, the dataset $X^s = \{(x_1^s, y_1^s), \ldots, (x_n^s, y_n^s)\}$ have n_s labeled data. In the $\mathbf{D}_{\mathrm{t}}$, the data of the environment with different signal-noise ratio (SNR) from the $\mathbf{D}_{\mathrm{s}}$, is denoted as $X^t = \{x_1^t, \ldots, x_n^t\}$, but the label is unknown. The task is to find the target prediction function $f(\cdot)$ through X^s, Y^s and X^t to predict the label of the $\mathbf{D}_{\mathrm{t}}$.

In a cooperative scenario, the data in the $\mathbf{D}_{\mathrm{s}}$ and $\mathbf{D}_{\mathrm{t}}$ are independent and the distribution is the same for both. In this situation, We learn $P_s(y \mid x)$ through X^s and Y^s to build a classifier that is also similar to $P_t(y \mid x)$, and the test effect in the target domain is also very good. However, in non-cooperative scenarios, the two are in different distributions. At this point, we only use X^s and Y^s to learn the classifier constructed by $P_s(y \mid x)$ is different from $P_t(y \mid x)$, and the adaptability on the target domain is relatively poor. Therefore, the DANN method introduces X^t and trains together with X^s and Y^s, alleviating the difference in domain adaptation and increasing the generalization performance.

4 Proposed Framework

This paper mainly uses DANN to study the transfer problem under different noise domains. Compared with the generation of confrontation network, the difference between them that the samples in the target domain are fake samples in the generation of confrontation network. Therefore, the feature extractor in DANN mainly plays a role of feature extraction. It mainly extracts common transferable features between the two domains, the features learned by the discriminator are very similar and cannot be distinguished accurately, and the discriminative ability of the discriminator is continuously enhanced, to achieve better classification performance. In addition, a transformer encoder is added to the feature extraction network to extract the contextual correlation of signals and learn deeper transferable features. The main architecture of DANN-Transformer is shown in Fig. 2.

The four main parts in DANN-Transformer are introduced as follows:

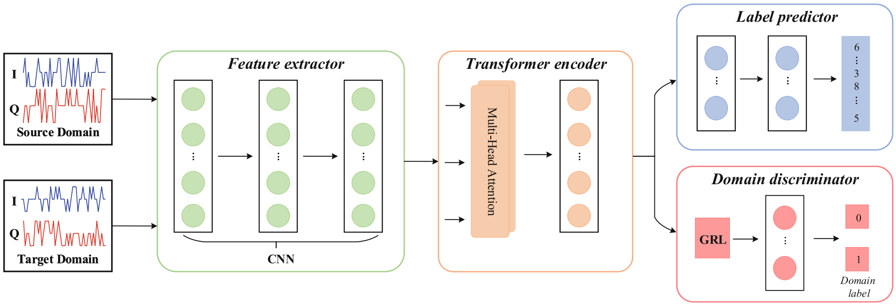


Fig. 2. The overall framework of the proposed algorithm

- **Feature extractor:** It mainly consists of a basic CNN network for feature extraction for label predictors to optimize classification performance and domain discriminators to optimize discrimination performance.
- **Transformer encoder:** Embed the extracted features and add position information, then make a residual connection with the new vector generated by the multi-head self-attention layer, and finally pass through the feed forward neural network (see Fig. 3).
- **Label predictor:** As far as possible to separate the correct label.
- **Domain classifier:** Distinguish as much as possible from which domain the extracted transferable features come from.

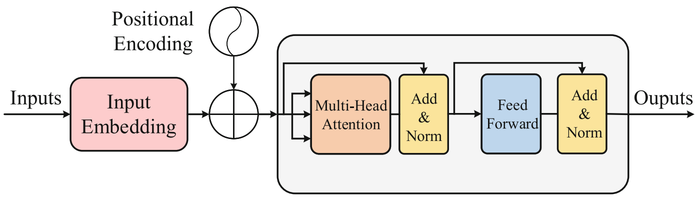


Fig. 3. Transformer encoder

The original I/Q signal is first passed through the feature extractor, and then the transferable features are extracted by the transformer encoder, finally classified through the label predictor. Adding a gradient reversal layer (GRL) can achieve the effect of confrontation. The loss function and network structure are described in detail in the following subsections.

4.1 Loss Functions

For the label predictor,softmax as an activation function, its output is:

$$G_y\left(G_f(x);\mathbf{V},\mathbf{c}\right) = softmax\left(\mathbf{V}\mathbf{G}_{\mathrm{f}}(\mathrm{x}) + \mathbf{c}\right) \tag{1}$$

When a given data point (x_i, y_i), The loss of the label predictor is:

$$\mathcal{L}_y\left(G_y\left(G_f\left(x_i\right)\right), y_i\right) = \log \frac{1}{\mathrm{G_y}\left(\mathrm{G_f(x)}\right)} y_i \tag{2}$$

Therefore, on the source domain, our training optimization goal is:

$$\min_{\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}} = \left[\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_y^i(\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}) + \lambda \cdot R(\mathbf{W},\mathbf{b})\right] \tag{3}$$

Among them, $\mathcal{L}_y^i$ represents the label prediction loss of the ith sample, λ is an artificially set regularization parameter, and $\lambda \cdot R(\mathbf{W},\mathbf{b})$ can reduce the phenomenon of over-fitting.

The core of DANN is the domain discriminator, sigmoid as an activation function, its output is:

$$G_d\left(G_f(\mathbf{x}); \mathbf{u}, z\right) = sigm\left(\mathbf{u}^\top G_f(\mathbf{x}) + z\right) \tag{4}$$

Then, the domain discriminator loss $G_d(\cdot)$ is defined as follows:

$$\mathcal{L}_d\left(G_d\left(G_f\left(\mathbf{x}_i\right)\right), d_i\right) = d_i \log \frac{1}{G_d\left(G_f\left(\mathbf{x}_i\right)\right)} + (1 - d_i)\log \frac{1}{G_d\left(G_f\left(\mathbf{x}_i\right)\right)} \tag{5}$$

Among them, d_i represents which domain the sample comes from. At this point, the optimization objective of the domain discriminator is:

$$R(\mathbf{W},\mathbf{b}) = \max_{\mathbf{u},z}\left[-\frac{1}{n}\sum_{i=1}^{n} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) - \frac{1}{n'}\sum_{i=n+1}^{N} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z)\right] \tag{6}$$

The total loss of the trained network mainly composed of label predictor loss (source domain) and domain discriminator loss (source domain, target domain). So we get the total objective function as:

$$\begin{aligned} &E(\mathbf{W},\mathbf{V},\mathbf{b},\mathbf{c},\mathbf{u},z) = \tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mathcal{L}_y^i(\mathbf{W},\mathbf{b},\mathbf{V},\mathbf{c}) \\ &-\lambda\left(\tfrac{1}{n}\textstyle\sum_{i=1}^{n} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z) + \tfrac{1}{n_i'}\textstyle\sum_{i=n+1}^{N} \mathcal{L}_d^i(\mathbf{W},\mathbf{b},\mathbf{u},z)\right) \end{aligned} \tag{7}$$

Among them, during training, the parameter optimization of label predictor and domain discriminator can be achieved by minimizing and maximizing objective functions respectively.

4.2 Network Structure

As shown in Table 1 are the specific parameters of the network structure. Suppose 64 samples are input to the neural network each time, Then the input tensor size is $64 * 2 * 4800$, the output size is $64 * 10$. The label classifier uses three fully connected layers, followed by softmax for classification. The domain discriminator uses three fully-connected layers, followed by Logsoftmax for source and target domain classification.

Table 1. network structure of feature extractor

name	output	param
Conv1d	(64,512,4800)	3584
BatchNorm1d	(64,512,4800)	1024
MaxPool1d	(64,512,4800)	–
ReLU	(64,512,2400)	–
Conv1d	(64,256,2400)	393472
BatchNorm1d	(64,256,2400)	512
MaxPool1d	(64,256,1200)	–
ReLU	(64,256,1200)	–
Conv1d	(64,128,1200)	163968
BatchNorm1d	(64,128,1200)	256
MaxPool1d	(64,128,600)	–
ReLU	(64,128,600)	–
Conv1d	(64,64,600)	41024
BatchNorm1d	(64,64,600)	128
MaxPool1d	(64,64,300)	–
ReLU	(64,64,300)	–

5 Experimental Results

5.1 Dataset

Automatic Dependent Surveilance-Broadcast (ADS-B) is an integrated monitoring and communication system that can be used to transmit flight parameters. It can realize the sharing of flight information and broadcast flight information regularly, which is necessary for the implementation of regional air management. We use the dataset proposed in paper [16] to evaluate our proposed method. It was acquired in a real world noisy environment. We select 10 types of signals in the ADS-B dataset, each type of signal has 400 signal samples, and the length of each signal sample is 4800. The samples in the two fields are randomly selected, the number of samples is 2000, and the sample categories are 10 categories.

5.2 Baseline

In this section, we introduce other domain adaptation methods for comparison of the performance of DANN-Transformer methods.

- **CNN** [17], the source domain is extracted and classified through the same CNN network as DANN, the model is trained, and then tested on the target domain.
- Transfer Component Analysis (**TCA**) [18], it only does marginal distribution alignment.

– Joint distribution alignment (**JDA**) [19], it performs marginal distribution and conditional distribution alignment.
– Weighted Balanced Distribution Adaptation (**W-BDA**) [20], it can adapt the importance of marginal distribution and conditional distribution, and adaptively change the weight of each class
– Transfer Joint Matching (**TJM**) [21], it aims to reduce the domain difference by jointly matching the features and reweighting the instances across domains in a principled dimensionality reduction procedure.
– Manifold Embedded Distribution Alignment (**MEDA**) [22], it proposes manifold feature transformations to reduce data drift between domains.

5.3 Performance Comparison of Different Methods

In this paper, additive Gaussian white noise (AWGN) with different SNRs is added to two domains to carry out migration experiments. In this experiment, we verify the superiority of DANN-Transformer method. We add -5dB noise on the $\mathbf{D}_s$, and then the SNR ranges from -20dB to 20dB on the $\mathbf{D}_t$, and the rest of the conditions remain unchanged, results are shown in Fig. 4.

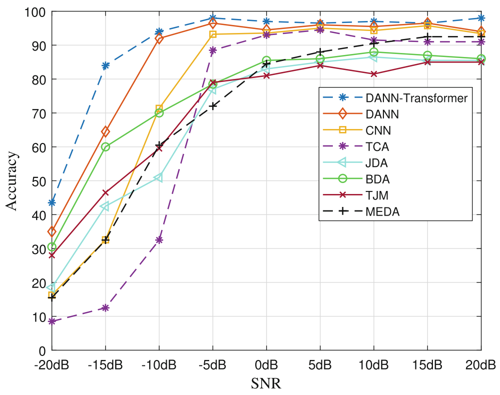
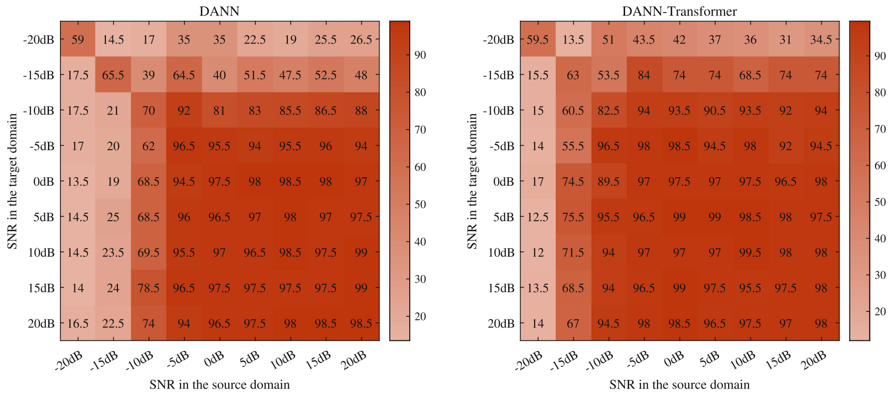


Fig. 4. Effect of SNR Variation in the Target Domain on Recognition

In Fig. 4 is a comparison of the accuracy of different UDA algorithms, we can see the change of the accuracy curve with the increase of SNR. Compared with the other seven methods, the DANN-Transformer method has obvious advantages at -20dB to -5dB, and becomes relatively stable after -5dB, and the accuracy rate is not much different, because the characteristics of the samples are not very different at high SNR. And we can also see that the deep transfer methods are generally better than the non-deep transfer methods, because the deep transfer method can learn the deep features of the sample. In addition, we can see that when the transformer encoder module is added to DANN, it shows excellent performance at low SNRs, because it can capture deeper transferable features.

5.4 Domain Adaptation Under Different SNRs

Through the previous section, we analyzed that DANN-Transformer and DANN have great advantages over other baseline algorithms. In this summary, we analyze the mutual transfer under different SNRs. The SNR of the source domain changes from -20dB to 20dB, and the target domain SNR varies from -20dB to 20dB. Fig. 5 shows the domain adaptation results of DANN at different SNRs.



(a) DANN (b) DANN-Transformer

Fig. 5. Accuracy of domain adaptation under different SNRs

In Fig. 5, we can see that when the DANN method transfer from data with a SNR greater than -5dB to a SNR greater than -5dB, the accuracy rate can reach more than 90%. When the SNR is lower than -10dB, the signal will be greatly affected, and the recognition accuracy will be greatly reduced. However, DANN-Transformer can still achieve an accuracy rate of 90% when the SNR is -10dB, indicating that it has stronger robustness. When the noise environment is the same between the two domains, the characteristics of the two are similar, and the effect of domain adaptation is better. For example, -20dB SNR in both domains, it is better than when the target domain is other SNRs.

5.5 Effect of Sample Size and Category

In this experiment, we study the effect of sample size on classification accuracy in two domains. We fixed the SNR of the $\mathbf{D}_s$ at 10dB, and fixed the SNR of the $\mathbf{D}_t$ at -5dB, and kept the rest of the conditions unchanged.

Transfer learning cannot always assume that there are many samples in the $\mathbf{D}_s$, therefore, it is necessary to explore the impact of the number of samples in the $\mathbf{D}_s$ on the domain adaptation algorithm. In this experiment, the number of samples in this experiment varies from 200 to 2000 with an interval of 200. The number of unlabeled samples in the $\mathbf{D}_t$ is fixed at 1600. Table 2 shows the

Table 2. classification accuracy rate of source domain sample change

Samples in D_s	**Methods**						
	TCA	**JDA**	**TJM**	**MEDA**	**BDA**	**DANN**	**DANN-Transformer**
200	16.0	50.5	39.5	71.0	**76.0**	59.0	70.0
400	14.0	53.0	51.5	76.0	80.0	68.5	**80.5**
600	15.5	65.5	64	81.5	78.0	79.0	**84.5**
800	16.0	73.5	77	82.5	79.0	86	**86.5**
1000	17.0	77.5	82.5	83.5	84.0	90.0	**90.5**
1200	17.0	84.5	86.0	83.5	86.5	90.5	**91.0**
1400	17.0	84.5	85.5	84.0	86.0	**93.5**	93.0
1600	16.5	85.0	86.0	85.0	87.0	91.5	**95.0**
1800	16.0	85.5	84.0	85.5	88.5	95.0	**97.5**
2000	17.5	85.5	81.5	84.5	89.0	95.5	**98.0**

accuracy of the source domain sample size under different methods. Except for TCA, the accuracy of other methods increases with the increase of the sample size. The accuracy of TCA is basically unchanged, which may be due to the poor transfer effect in this noisy environment. To achieve an accuracy rate of more than 90%, it is necessary to ensure that the number of samples in the $\mathbf{D}_\mathrm{s}$ is more than 1000. It shows that the number of samples in the $\mathbf{D}_\mathrm{s}$ has an impact on the classification effect.

Table 3. classification accuracy rate of target domain sample change

Samples in D_t	**Methods**						
	TCA	**JDA**	**TJM**	**MEDA**	**BDA**	**DANN**	**DANN-Transformer**
200	17.5	85.5	81.5	84.5	84.5	91.5	**96.0**
400	19.3	85.3	84.0	87.0	86.5	94.5	**97.0**
600	19.5	86.3	84.8	84.0	84.8	95.5	**96.0**
800	21.4	85.5	87.3	87.0	87.0	95.0	**97.0**
1000	23.6	85.5	87.4	85.1	87.8	94.5	**98.0**
1200	23.2	85.7	87.7	85.4	85.0	93.5	**96.0**
1400	23.6	86.6	87.8	85.2	86.5	94.0	**97.0**
1600	24.2	85.5	87.2	85.3	86.9	95.5	**98.5**

The number of samples in the $\mathbf{D}_\mathrm{t}$ is also a key factor affecting the effect of domain adaptation. Therefore, it is important to study the effect of the number of unlabeled samples in the $\mathbf{D}_\mathrm{t}$ on the recognition performance. The number of samples in this experiment varies from 200 to 1600 with an interval of 200.

Source domain samples are fixed at 2000. Table 3 shows the accuracy of the target domain sample size under different methods, we can see that the number of samples in the $\mathbf{D}_\mathrm{t}$ has little effect on the classification accuracy. When the number of samples is 200, the classification accuracy of DANN-Transformer can still reach more than 95%. This shows that the DANN-Transformer method still exhibits good performance when the $\mathbf{D}_\mathrm{t}$ has few samples.

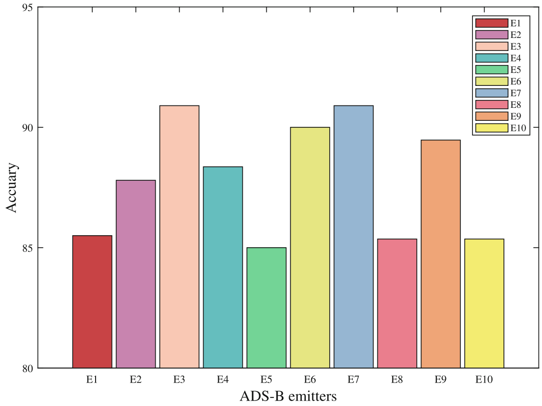


Fig. 6. Classification accuracy under category ratio $\mathbf{D}_\mathrm{t}/\mathbf{D}_\mathrm{s}$ =1/10

In addition, we explore the impact of the category of samples in the $\mathbf{D}_\mathrm{t}$ on the classification accuracy. We assume that there is only one type of emitter signal in the $\mathbf{D}_\mathrm{t}$, and its recognition accuracy mainly depends on whether the type of signal itself is separable. In Figure 6, E represents the emitter, we can see that the separability of E1, E5, E8, and E10 is relatively poor, but the classification accuracy is not much different, which shows that whether the categories between the two fields are aligned will not affect the classification accuracy.

6 Conclusions

In this paper, we propose a DANN method suitable for SEI of different noises, an algorithm for unsupervised domain adaptation, which mainly performs recognition in non-cooperative scenarios. Adding the Transformer encoder module to DANN improves the robustness of the method. We validate the effectiveness of the proposed method for SEI application on the real-world ADS-B dataset, where the proposed method achieves better performance at low SNR (-20dB to -5dB) compared to other baseline methods. We perform migration under different SNR conditions, which shows that the method has good adaptability in different noise environments. In addition, the number of samples in the target domain has little effect on the performance of DANN-Transformer, while the number of samples in the source domain has a certain impact on the performance of the method.

When the number of samples in the target domain is only 200, it can still achieve an accuracy rate of more than 95%. In the future, it is very valuable to study migration under different noise conditions, because the real noise environment is not necessarily just AWGN, and the noise environment is very rich.

References

1. Bao, Z., Lin, Y., Zhang, S., Li, Z., Mao, S.: Threat of adversarial attacks on dl-based iot device identification. IEEE Int. Things J. **9**(11), 9012–9024 (2021)
2. Lin, Y., Tu, Y., Dou, Z., Chen, L., Mao, S.: Contour stella image and deep learning for signal recognition in the physical layer. IEEE Trans. Cognit. Commun. Netw. **7**(1), 34–46 (2020)
3. Chen, S., Zheng, S., Yang, L., et al.: Deep learning for large-scale real-world ACARS and ADS-B radio signal classification[J]. IEEE Access **7**, 89256–89264 (2019)
4. Pan, Y., Yang, S., Peng, H., et al.: Specific emitter identification based on deep residual networks[J]. IEEE Access **7**, 54425–54434 (2019)
5. Tu, Y., Lin, Y., Hou, C., Mao, S.: Complex-valued networks for automatic modulation classification. IEEE Trans. Veh. Technol. **69**(9) 10 085–10 089 (2020)
6. Hou, C., Liu, G., Tian, Q., Zhou, Z., Hua, L., Lin, Y.: Multi-signal modulation classification using sliding window detection and complex convolutional network in frequency domain. IEEE Int. Things J. (2022)
7. Wang, M., Lin, Y., Tian, Q., Si, G.: Transfer learning promotes 6g wireless communications: Recent advances and future challenges. IEEE Trans. Reliab. **70**(2), 790–807 (2021)
8. Huang, K., Yang, J., Liu, H., Hu, P.: Deep adversarial neural network for specific emitter identification under varying frequency. In: Bulletin of the Polish Academy of Sciences: Technical Sciences, pp. e136 737- e136 737 (2021)
9. Ganin, Y.: Domain-adversarial training of neural networks. J. Mach. Learn. Res. **17**(1), 2096–2030 (2016)
10. Sankaranarayanan, S., Balaji, Y.: Carlos D Castillo, and Rama Chellappa. Aligning domains using generative adver- sarial networks. In CVPR, Generate to adapt (2018)
11. Tasar, O., Giros, A., Tarabalka, Y., et al.: Daugnet: Unsupervised, mul- tisource, multitarget, and life-long domain adaptation for semantic segmentation of satellite images[J]. IEEE Trans. Geosci. Remote Sens. **59**(2), 1067–1081 (2020)
12. Liu, X., Hu, B., Liu, X., Lu, J., You, J., Kong, L.: Energy- constrained self-training for unsupervised domain adap- tation. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7515–7520. IEEE (2021)
13. Zou, Y., Yu, Z., Liu, X., Kumar, BVK.: Confidence regularized self-training. ICCV (2019)
14. Tzeng, E., Hoffman, J., Saenko, K., et al.: Adversarial discriminative domain adaptation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 7167–7176 (2017)
15. Bousmalis, K., Silberman, N., Dohan, D., et al.: Unsupervised pixel-level domain adaptation with generative adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3722-3731 (2017)
16. Ya, T.U., Yun, L.I.N., Haoran, Z.H.A., et al.: Large-scale real-world radio signal recognition with deep learning[J]. Chin. J. Aeronaut. **35**(9), 35–48 (2022)

17. Wong, L.J., Headley, W.C., Andrews, S., Gerdes, R.M., Michaels, A.J.: Clustering learned cnn features from raw I/Q data for emitter identification. MILCOM 2018–2018 IEEE Military Communications Conference (MILCOM), pp. 26-33 (2018)
18. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE Trans. Neural Networks **22**(2), 199–210 (2010)
19. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer feature learning with joint distribution adaptation. In: Proceedings of the IEEE international conference on computer vision, pp. 2200-2207 (2013)
20. Wang, J., Chen, Y., Hao, S., Feng, W., Shen, Z.: Balanced distribution adaptation for transfer learning. In: 2017 IEEE International Conference on Data Mining (ICDM), New Orleans, LA, USA, pp. 1129–1134 (2017)
21. Long, M., Wang, J., Ding, G., Sun, J., Yu, P.S.: Transfer joint matching for unsupervised domain adaptation. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, pp. 1410–1417 (2014)
22. Wang, J., Feng, W., Chen, Y., Yu, H., Huang, M., Yu, P.S.: Visual domain adaptation with manifold embedded distribution alignment. In: Proceedings of the 26th ACM international conference on Multimedia. Association for Computing Machinery, New York, NY, USA, pp. 402–410 (2018)