

RESEARCH

Open Access



Specific emitter identification based on ensemble domain adversarial neural network in multi-domain environments

Dingshan Li¹, Bin Yao^{1*}, Pu Sun¹, Peitong Li¹, Jianfeng Yan¹ and Juzhen Wang^{2*}

*Correspondence:
yb8481@126.com;
wangjuzhen@whu.edu.cn

¹ China Ship Research and Development Academy, Chaoyang District, Beijing 100020, China

² School of Electronic Information, Wuhan University, Wuhan 430072, China

Abstract

Specific emitter identification is pivotal in both military and civilian sectors for discerning the unique hardware distinctions inherent to various launchers, it can be used to implement security in wireless communications. Recently, a large number of deep learning-based methods for specific emitter identification have been proposed, achieving good performance. However, these methods are trained based on a large amount of data and the data are independently and identically distributed. In actual complex environments, it is very difficult to obtain reliable labeled data. Aiming at the problems of difficulty in data collection and annotation, and the large difference in distribution between training data and test data, a method for individual radiation source identification based on ensemble domain adversarial neural network was proposed. Specifically, a domain adversarial neural network is designed and a Transformer encoder module is added to make the features obey Gaussian distribution and achieve better feature alignment. Ensemble classifiers are then used to enhance the generalization and reliability of the model. In addition, three real and complex migration environments, Alpine–Montane Channel, Plain–Hillock Channel, and Urban–Dense Channel, were constructed, and experiments were conducted on WiFi dataset. The simulation results show that the proposed method exhibits superior performance compared to the other six methods, with an accuracy improvement of about 3%.

Keywords: Domain adaptation, Domain adversarial neural networks, Ensemble learning, Specific emitter identification, Transformer encoder

1 Introduction

Specific emitter identification (SEI), which refers to the use of existing a priori information to achieve the identification of an individual generating a signal based on its unique characteristics, has been widely studied in cognitive radio [1], self-organizing networks [2], physical layer reliability [3] and the Internet of Things (IoT) [4]. It has gained recognition as a pivotal technology with significant applications in both civilian and military sectors [5, 6].

In the traditional SEI scheme, the system is typically divided into the two parts: signal radio frequency fingerprint (RFF) extraction and classification. Signal extraction can be classified into transient-based and steady-state-based methods. Transient

signals typically manifest during device state transitions, whereas steady-state signals characterize the transmitter's stable operational state [7]. The former has a short duration and is difficult to be extracted accurately. In addition, transient signal-based SEI relies heavily on basic characteristics. Steady-state signals, on the other hand, provide relatively more stable RFF features and are more suitable due to their long duration and low acquisition cost. Integral bispectrum, wavelets, and the Hilbert Huang transform (HHT) [8–11] are widely used in steady-state signal processing. The classification part is to develop appropriate classifiers to achieve accurate identification of target radiation sources using discriminative RFF. Reising [12] and Williams [13] applied Multiple Discriminative Analysis/Maximum Likelihood (MDA/ML) classifiers for feature classification. Brik [14] pioneered the implementation of the Support Vector Machine (SVM) classifier. In [15], SVM utilizes features extracted through Empirical Modal Decomposition (EMD) to accomplish classification. In [16], a feature vector neural network was constructed using pulse signal parameters. However, it has some limitations in data mining.

Nowadays, deep learning (DL) has garnered significant interest across various prominent domains and proved to be effective in various applications. The performance has been improved compared to traditional techniques. A novel long-tail SEI method is proposed in [17], employing decoupled representation (DR) learning. In [18], to overcome data limitations, propose a few-shot SEI (FS-SEI) method based on self-supervised learning and adversarial augmentation. Yao et al. in [19] proposed the use of asymmetric mask auto-encoder (AMAE) with few-shot to solve the few-shot problem. Recently, some scholars have also studied pruning techniques to lighten models and accelerate the inference speed of SEI [20, 21]. In recent years, research in this area has also been developing in the direction of resource optimization [22, 23].

One of the keys to using DL methods is the necessity for a substantial volume of labeled data to train the model and the data is independently and identically distributed. However, in real complex environments, acquiring a substantial number of labeled samples poses a formidable challenge. Therefore, classifying unknown data based on a small number of samples and improving the generalization and robustness of the model have become a key challenge [24]. Transfer learning (TL), which aims at extracting knowledge from the source tasks and applying the knowledge to the target task [25], is regarded as a reliable solution for SEI [26–28]. Unsupervised Domain Adaptation (UDA) is a subcategory of TL, it does not require labeled data from the target domain. Therefore, the use of unsupervised domain adaptation technique for radiation source individual identification can effectively address the issue of limited availability of labeled samples. Unsupervised adaptive methods are subdivided into two categories: self-training-based and adversarial learning-based. The former utilizes pseudo-labels to provide supervised information, while the latter aligns the source and target domain distributions. In complex environments, the pseudo-labels are of poor quality and cannot be accurately domain-aligned, the domain adaptation technique utilizing adversarial training can enhance the model's adaptation to the target domain's data distribution and enhance the model's generalization performance by acquiring the feature mapping between the source and target domains. This has inspired more and more researchers, including us, to devote themselves to exploring SEI based on adversarial domain adaptation.

In this paper, we propose a method for SEI using ensemble domain adversarial neural network. The method consists of a domain adversarial neural network based on the transformer encoder and a classifier based on ensemble learning. Specifically, the former adds a transformer encoder after the feature extraction layer of the domain adversarial neural network, so that the features extracted obey Gaussian distribution after passing through the encoder, which is conducive to feature alignment. The latter utilizes the ensemble learning method of aggregate the outcomes of numerous weak learners to improve recognition performance. This article makes the following principal contributions.

- This paper proposes a method of ensemble domain adversarial neural network for specific emitter identification. We creatively add a transformer encoder based on the domain adversarial neural network to make the extracted features obey Gaussian distribution.
- Using ensemble learning methods to combine multiple weak classifiers into one strong classifier. Adopting a combination strategy of weighted voting method, each weak learner is assigned different weights to improve the quality and accuracy of decision-making.
- Comprehensively consider the signal transmission environment, construct three channel environments: Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel, simulate different signal transmission situations, and reflect the differences in transfer environments.
- Evaluating the method proposed in this article to migration on WiFi data sets in three environments: Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel, and compared with six methods. The simulation results demonstrate that the method proposed in this paper attains state-of-the-art recognition performance.

The remainder of this article is organized as follows. Section 2 introduces the system model, signal model, channel model and problem description. Section 3 explains the method proposed in this paper in detail. Section 4 analyzes the experimental results. Section 5 draws conclusions.

2 System model, signal model, channel model and problem formulation

2.1 System model

This article mainly implements individual recognition of WiFi signals in different environments. The SEI system used for identification of different WiFi transmitters is shown in Fig. 1. It mainly consists of three parts: data generation, data reception and preprocessing, and model training. The data generation department consists of 7 different WiFi signal transmitters, and then performs signal attenuation operations through a channel simulator, and finally consists of three channel environments of Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel through mathematical modeling. SM200B is used to receive signals, and perform signal detection and preprocessing operations. Data preprocessing steps include signal

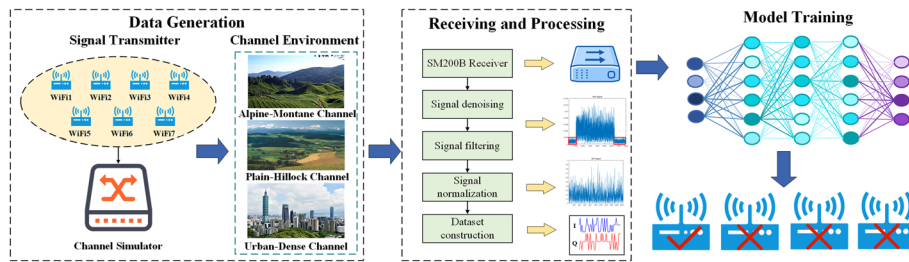


Fig. 1 The system structure of WiFi individual identification

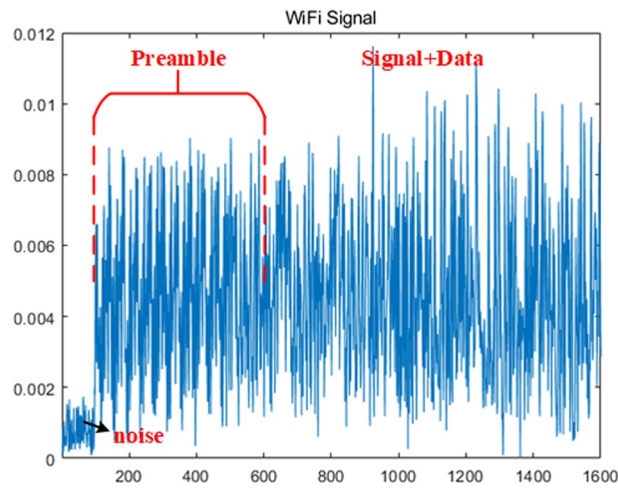


Fig. 2 WiFi signal waveform

denoising, signal filtering and signal normalization. The data is then fed into model training, and finally the signals are individually identified.

Considering the differences in the three channel environments, it is difficult to deploy the model trained with data collected from the original channel environment to the new channel environment. Specifically, there are at least two issues with rapid deployment: (1) Obtaining a substantial volume of labeled data for updating models in new channel environments is a challenging endeavor; (2) The model trained in the original environment exhibits limited generalization capabilities in the new environment. Therefore, in the case of significant differences between the two domains and limited data samples, ensemble adversarial domain adaptation is used to achieve SEI.

2.2 Signal model

A WiFi signal is a radio wave whose frequency is usually around 2.4–5 GHz. Its signal waveform is shown in Fig. 2. The WiFi signal emitted by the k th device can be described using the following formula.

$$r_k(t) = s_k(t) * h_k(t) + n_k(t), k = 1, 2, \dots, K \tag{1}$$

where $r_k(t)$ represents the received WiFi signal, $s_k(t)$ stands for the transmitted WiFi signal, $h_k(t)$ represents the channel response, which characterizes the influence of the

signal in the frequency domain when the signal is transmitted from the sending end to the receiving end through the channel. This typically includes multipath propagation, fading, and other channel characteristics, $n_k(t)$ denotes the channel noise, which is due to random interference introduced by the environment and communication equipment.

2.3 Channel model

In communication systems, there may be various types of interference in the channel, which can affect the transmission and reception of signals. Taking into account the noise interference, fading, and path loss in the channel, we model three channel environments: Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel. The general formula for path loss is

$$L = L0 + 10 * \lambda * \log_{10}(d) + \sigma * randn \quad (2)$$

where L is the path loss, $L0$ is a constant, λ is the path loss coefficient, d is the distance, and σ is the standard deviation of Gaussian random variables. $randn$ usually refers to random numbers drawn from the standard normal distribution. The following is an introduction to the three channel environments.

2.3.1 Alpine–Montane channel model

In the Alpine–Montane Channel model, set $L0$ for 40, λ for 3, d for 100, and σ for 4. Adding Rayleigh fading, the probability density function(PDF) of the Rayleigh distribution is:

$$f_R(r) = \frac{r}{\sigma^2} e^{-\frac{r^2}{2\sigma^2}}, r > 0 \quad (3)$$

where r represents the value of a random variable. σ^2 It is a scale parameter of the distribution, which controls the degree of diffusion of the distribution. Finally add path loss and the channel model after Rayleigh fading as follows:

$$y_{Alpine_Montane} = x * 10^{-\frac{L}{20}} * rayleigh + n \quad (4)$$

where $y_{Alpine_Montane}$ represents the signal output after passing through the Alpine–Montane Channel channel model, x is the transmission signal, $rayleigh$ is Rayleigh fading, and n represents additive white Gaussian noise (AWGN).

2.3.2 Plain-Hillock channel model

In the Plain-Hillock Channel channel model, set $L0$ for 35, λ for 2.8, d for 100, and σ for 3.5. Then add path loss and the channel model after Rayleigh fading as follows:

$$y_{Plain_Hillock} = x * 10^{-\frac{L}{20}} * rayleigh + n \quad (5)$$

where $y_{Plain_Hillock}$ represents the signal output after passing through the Plain-Hillock Channel channel model, x is the transmission signal, $rayleigh$ is Rayleigh fading, and n represents AWGN.

2.3.3 Urban-Dense channel model

In the urban channel model, set $L0$ for 45, λ for 3.5, d for 100, and σ for 6. Then add Rice fading, the PDF of the Rice distribution is:

$$f_L(r) = \frac{2r}{\sigma^2} e^{-\frac{r^2 + \alpha^2}{\sigma^2}} I_0\left(\frac{2\alpha r}{\sigma^2}\right) \quad (6)$$

where r is a random variable, λ is a scale parameter, α is the non central parameter of the distribution, and $I_0(\cdot)$ is the modified Bessel function. The final signal output formula is:

$$y_{Urban_Dense} = x * 10^{-\frac{L}{20}} * rice + n \quad (7)$$

where y_{Urban_Dense} is the signals after urban offense and defense channels, x is the transmission signal, $rice$ is rice fading, and n represents AWGN.

2.4 Problem formulation

In this article, x represents the input signal sample, which comprises an IQ format signal from the RF device; y corresponds to the category of the respective RF device. $\mathbf{D} = \{(x_1, y_1), \dots, (x_n, y_n)\}$ represents a dataset containing WiFi signal samples and corresponding labels. \mathcal{X} represents the sample space, \mathcal{Y} represents category space. $P(y | x)$ represents the conditional probability distribution, $P(x, y)$ represents the joint probability distribution.

2.4.1 SEI problem

The SEI problem can be expressed as maximum a posterior probability (MAP) problem, which solves by comparing the posterior probabilities of different categories to find the category with the maximum posterior probability. In individual recognition problems, this can be understood as finding the category of individuals most likely to correspond to the given data, represented by the following formula

$$\hat{y} = \arg \max_{y \in \mathcal{Y}} f_S(y | \mathbf{x}; \mathbf{W}_S) \quad (8)$$

where $f_S(\cdot)$ represents the mapping function, \mathbf{W}_S represents the hyperparameter set, and \hat{y} represents the predicted result. The objective of SEI is to determine optimal hyperparameters $W \in \mathcal{W}$ to achieve mapping from data space \mathcal{X} to label space \mathcal{Y} . And minimize the expected error ε_{ex} as follows

$$\min_{W_S \in \mathcal{W}} \varepsilon_{ex} = \min_{W_S \in \mathcal{W}} E_{(x,y) \sim P_{(x,y)}} \mathcal{L}(\hat{y}, y) \quad (9)$$

where $\mathcal{L}(\hat{y}, y)$ represents the loss compared to the real category and the predicted category. Usually $P_{(x,y)}$ is unknown, and the minimum empirical error ε_{em} is usually used to replace ε_{ex} , which can be represented as follows

$$\min_{W_S \in \mathcal{W}} \varepsilon_{em} = \min_{W_S \in \mathcal{W}} E_{(x,y) \sim D} \mathcal{L}(\hat{y}, y) \quad (10)$$

2.4.2 Adversarial domain adaptation for SEI problem

Adversarial domain adaptation is an UDA method, where we define the dataset of the source domain as $D_S = \{(x_1^s, y_1^s), \dots, (x_n^s, y_n^s)\}$ and the dataset of the target domain as $D_T = \{x_1^t, \dots, x_n^t\}$, the target domain has no label. The task is to find the target prediction function $f(\cdot)$ through x^s, y^s and x^t to predict the label.

In a cooperative scenario, the data in the D_S and D_T are independent and the distribution is the same for both. In this situation, we learn $P_S(y | x)$ through x^s and y^s to build a classifier that is also similar to $P_T(y | x)$, 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 $P_S(y | x)$ is different from $P_T(y | x)$, and the adaptability on the target domain is relatively poor. Therefore, the adversarial domain adaptation method introduces x^t and trains together with x^s and y^s , alleviating the difference in domain adaptation and increasing the generalization performance.

3 The method proposed in this article

3.1 Overview of the framework

The overall framework is shown in Fig. 3. Firstly, bootstrap sampling is performed on the labeled source domain and unlabeled target domain data to obtain N different sampling sets. Then, a weak classifier is used, where we use a domain adversarial neural network (DANN). And it has been improved by adding a transformer encoder to the feature extractor, which can better align features and provide deeper transferable features. Add a Gradient Reverse Layer (GRL) to achieve the classifier and domain discriminator backpropagation optimize the gradient in the direction of optimizing the classifier’s performance. This can meet the needs of both discriminators and classifiers simultaneously. Finally, the results are integrated through a weighted voting method.

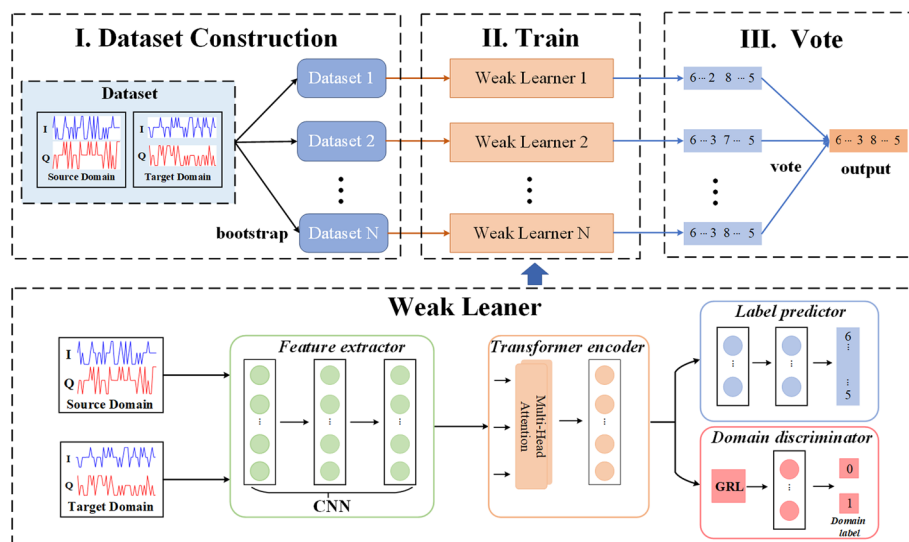


Fig. 3 Overview of the proposed E-DANN method

3.2 Dataset sampling

Bootstrap sampling is a widely utilized random sampling method in statistics and machine learning. It mainly involves estimating population parameters, constructing confidence intervals, and conducting hypothesis testing. Its main idea is to extract multiple self-service samples from the original samples by putting them back, and then perform statistical analysis on these self-service samples. Here are the detailed steps of Bootstrap sampling:

1. Original Dataset: First, there's an original dataset with n observation samples, which can be data points obtained from experiments, surveys, or data collection.
2. Sampling with Replacement: Bootstrap sampling generates multiple random samples of size n from the original dataset. This is done with replacement, meaning that in each sampling, the same sample can be selected multiple times, while others may not be selected at all. This process simulates independent repeated random sampling from the population. The probability of each data not being sampled is:

$$p = \lim_{n \rightarrow +\infty} \left(1 - \frac{1}{n}\right)^n = \frac{1}{e} \approx 0.368 \quad (11)$$

Bootstrap sampling is highly advantageous for dataset sampling within this work, primarily because of its independence from data distribution assumptions. This versatility makes it suitable even for non-normally distributed data. By generating multiple Bootstrap samples, it facilitates uncertainty estimation by enabling the calculation of variance and confidence intervals, providing valuable insights into estimate uncertainty. This adaptability extends to estimating statistical parameters, conducting hypothesis tests, constructing confidence intervals, and supporting ensemble methods such as Bagging in machine learning. In summary, Bootstrap sampling serves as a powerful statistical tool, adept at estimating parameters and managing uncertainty without being confined by data distribution or assumptions. These attributes make it a well-suited choice for dataset sampling, as discussed in this work.

3.3 Domain adversarial neural network based on transformer encoder

Domain adversarial neural network (DANN) is a deep learning method used to solve domain adaptive problems. The main idea behind DANN is to reduce distribution differences between different domains through adversarial training, thereby improving the model's generalization ability. Compared with the generation of confrontation network, the difference between them is 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 network architecture is shown in the weak classifier in Fig. 3, which mainly consists of the following four parts:

- **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.
- **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.

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 GRL can achieve the effect of confrontation. The domain classifier is trained to maximize its errors between the source and target domains, effectively forcing the features to become indistinguishable across domains. This is achieved by minimizing a specific loss function, often referred to as the Domain Adversarial Loss. Below is the calculation of the domain adversarial loss function.

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

$$G_y(G_f(x); \mathbf{V}, \mathbf{c}) = \text{softmax}(\mathbf{V}\mathbf{G}_f(x) + \mathbf{c}) \quad (12)$$

Among them, \mathbf{V} represents the weight matrix. \mathbf{c} represents the bias parameter. When a given data point (x_i, y_i) , The loss of the label predictor is:

$$\mathcal{L}_y(G_y(G_f(x_i)), y_i) = \log \frac{1}{G_y(G_f(x))} y_i \quad (13)$$

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

$$\min_{\mathbf{W}, \mathbf{b}, \mathbf{V}, \mathbf{c}} = \left[\frac{1}{n} \sum_{i=1}^n \mathcal{L}_y^i(\mathbf{W}, \mathbf{b}, \mathbf{V}, \mathbf{c}) + \lambda \cdot R(\mathbf{W}, \mathbf{b}) \right] \quad (14)$$

Among them, \mathcal{L}_y^i represents the label prediction loss of the i th sample, λ is an artificially set regularization parameter, \mathbf{W} represents the weight matrix, \mathbf{b} represents the bias vector, 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:

$$G_d(G_f(\mathbf{x}); \mathbf{u}, z) = \text{sigm}(\mathbf{u}^\top G_f(\mathbf{x}) + z) \quad (15)$$

Among them, \mathbf{u} represents a set of network parameters. Then, the domain discriminator loss $G_d(\cdot)$ is defined as follows:

$$\begin{aligned} \mathcal{L}_d(G_d(G_f(\mathbf{x}_i)), d_i) \\ = d_i \log \frac{1}{G_d(G_f(\mathbf{x}_i))} + (1 - d_i) \log \frac{1}{G_d(G_f(\mathbf{x}_i))} \end{aligned} \quad (16)$$

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

$$R(\mathbf{W}, \mathbf{b}) = \max_{\mathbf{u}, z} \left[-\frac{1}{n} \sum_{i=1}^n \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) - \frac{1}{n'} \sum_{i=n+1}^N \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) \right] \tag{17}$$

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:

$$E(\mathbf{W}, \mathbf{V}, \mathbf{b}, \mathbf{c}, \mathbf{u}, z) = \frac{1}{n} \sum_{i=1}^n \mathcal{L}_y^i(\mathbf{W}, \mathbf{b}, \mathbf{V}, \mathbf{c}) - \lambda \left(\frac{1}{n} \sum_{i=1}^n \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) + \frac{1}{n'} \sum_{i=n+1}^N \mathcal{L}_d^i(\mathbf{W}, \mathbf{b}, \mathbf{u}, z) \right) \tag{18}$$

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

3.4 Combining strategies

In the prediction of classification problems, the commonly used method is to combine weak classifier using a voting strategy. This involves voting on the output results of each weak classifier to determine the final prediction of the classifier. The combination strategy adopted in this article is the weighted voting method. It assigns different weights to each weak classifier, where each weak classifier’s prediction is multiplied by a weight. The final category is determined by summing the weighted votes for each category, and the category with the highest cumulative score is designated as the final category. The mathematical expression for the weighted voting method is as follows.

$$H(x) = c_{\arg \max_j} \sum_{i=1}^T w_i h_i^j(x) \tag{19}$$

where $H(x)$ is the weighted voting result, and $h_i^j(x)$ is the predicted probability result of the i -th learner for the j -th class, w_i is the weight of the i -th learner.

In addition, there is a majority voting method, where the absolute majority voting method requires that the predicted number of votes for the category must exceed half of the total, while the plurality voting method selects the highest number of votes as the final output result. Below is the mathematical expression for the plurality voting method.

$$H(x) = \arg \max_j \sum_{i=1}^T h_i^j(x) \tag{20}$$

where $H(x)$ is the plurality voting result, and $h_i^j(x)$ is the predicted probability result.

The plurality voting method is suitable when there is little difference in the recognition of a single learner and there is no prior knowledge indicating that a certain learner

is more important. The weighted voting method is suitable for learners with different performances.

3.5 Algorithm process

Algorithm 1 describes the overall training process of the method proposed in this article. Firstly, the dataset is sampled using bootstrap sampling. During training, parameters are updated in the opposite direction of the gradient as per Eq. 18 for minimization and in the direction of the gradient for maximization. In words, the neural network and the domain regressor engage in an adversarial competition, striving to optimize the objective function defined in Eq. 18. Finally, a combination strategy of weighted voting method is adopted for the training results.

Algorithm 1 Training of the proposed model.

Input: Labeled source data $D_S = \{(x_i, y_i)\}_{i=1}^n$, unlabeled target data $D_T = \{(x_i)\}_{i=1}^n$, adaptation parameter λ , learning rate μ , number of base learners T , number of iterations B .

Training process

- 1: Using bootstrap sampling method to sample D_S and D_T datasets.
- 2: **for** $i = 1$ to T **do**
- 3: **for** $b = 1$ to B **do**
- 4: **#Forward propagation**
- 5: $G_f(x_i) \leftarrow \text{sigm}(\mathbf{b} + \mathbf{W}x_i)$
- 6: $G_y(G_f(x_i)) \leftarrow \text{softmax}(\mathbf{c} + \mathbf{V}G_f(x_i))$
- 7: **#Backward propagation**
- 8: $\Delta \mathbf{c} \leftarrow -(\mathbf{e}(y_i) - G_y(G_f(x_i)))$
- 9: $\Delta \mathbf{v} \leftarrow \Delta \mathbf{c} G_f(x_i)^T$
- 10: $\Delta \mathbf{b} \leftarrow (\mathbf{V}^T \Delta \mathbf{c}) \odot G_f(x_i) \odot (1 - G_f(x_i))$
- 11: $\Delta \mathbf{w} \leftarrow \Delta \mathbf{b} \cdot (x_i)^T$
- 12: **#Domain adaptation regularizer: D_S**
- 13: $G_d(G_f(x_i)) \leftarrow \text{sigm}(d + \mathbf{u}^T G_f(x_i))$
- 14: $\Delta d \leftarrow \lambda(1 - G_d(G_f(x_i)))$
- 15: $\Delta \mathbf{u} \leftarrow \lambda(1 - G_d(G_f(x_i))) G_f(x_i)$
- 16: $\text{tmp} \leftarrow \lambda(1 - G_d(G_f(x_i)))$
 $\times \mathbf{u} \odot G_f(x_i) \odot (1 - G_f(x_i))$
- 17: $\Delta \mathbf{b} \leftarrow \Delta \mathbf{b} + \text{tmp}$
- 18: $\Delta \mathbf{w} \leftarrow \Delta \mathbf{w} + \text{tmp} \cdot (x_i)^T$
- 19: **# D_T is the same as D_S**
- 20: **end for**
- 21: **end for**
- 22: return $h(x), w_i$

Output: $H(x) = c_{\arg \max_j} \sum_{i=1}^T w_i h_i^j(x) = 0$

4 Experiment

4.1 Dataset description

We used the data collected by the proposed system to validate the identific performance of the SEI method, which is a WiFi signal similar to that collected in real Alpine-Montane Channel, Plain-Hillock Channel, and Urban-Dense environments. We denoise,

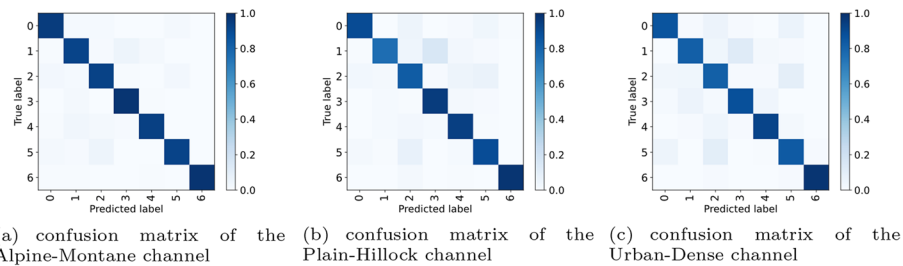


Fig. 4 Recognition effects under three channel environments

Table 1 Performance comparison of different domain adaptation methods

Methods	Alpine-Montane to Plain-Hillock (%)	Alpine-Montane to Urban-Dense (%)	Plain-Hillock to Alpine-Montane (%)	Plain-Hillock to Urban-Dense (%)	Urban-Dense to Alpine-Montane (%)	Urban-Dense to Plain-Hillock (%)
TCA	66.60	66.84	66.74	63.2	63.96	61.09
ARDA	79.91	75.31	79.95	76.79	85.50	79.53
ADDA	80.39	76.45	84.60	77.14	78.60	82.74
DAAN	86.25	83.54	90.42	83.42	87.75	87.89
DANN	89.85	86.50	91.16	86.57	91.12	89.53
E-DANN(ours)	92.89	90.32	94.53	91.00	94.39	92.35

filter, and normalize the collected signals, and then merge them into IQ channels to achieve the data format required for the experiment. The signal length is 2048 sampling points, with a total of 7 types of WiFi signals. The size of the source domain training dataset is 7000, the size of the target domain training dataset is 7000, the size of the validation dataset is 4200, and the size of the test dataset is 2800.

4.2 Baseline

We conducted a comparative analysis between the proposed E-DANN method and six other existing methods. Among them, we use convolutional neural networks (CNN) as the backbone network, and Table 1 provides an introduction to CNN related structures. The following is an introduction to the other six methods.

- Transfer component analysis (TCA) [29]: Perform marginal distribution alignment.
- Manifold embedded distribution alignment (MEDA) [30]: This represents the initial endeavor to implement dynamic distribution alignment in manifold domain adaptation.
- Domain adversarial neural networks (DANN) [31]: Achieve efficient domain transfer by enhancing it with a small number of standard layers and new gradient inversion layers.
- Dynamic adversarial adaptation network (DAAN) [32]: The conditional domain discriminant block and integrated dynamic adjustment factors are introduced.
- Adversarial discriminative domain adaptation (ADDA) [33]: Combines discriminative modeling, unrestricted weight sharing, and GAN loss.

- Adversarial representation domain adaptation (ARDA) [34]: This method measures distribution divergence by introducing Wasserstein GAN.

4.3 Recognition performance in different channel environments

In the first half of this article, three channel models were introduced, and the recognition performance of SEI largely depends on the channel environment. Therefore, this experiment verifies the recognition performance of SEI in three channel environments, using the backbone network CNN. The experimental results are depicted in Fig. 4.

Draw a confusion matrix based on the recognition results. The rows of the confusion matrix represent the actual categories and the columns represent the predicted categories of the model. The diagonal elements of the confusion matrix represent the number of samples correctly classified by the model, while the off-diagonal elements represent misclassifications by the model. The accuracy can be calculated from the value of the confusion matrix, which is the ratio of the number of correctly classified samples to the total number of samples. The matrix has the lowest degree of chaos in the Alpine–Montane channel environment, with an accuracy rate of 94.55%. In the Plain-Hillock channel environment, the accuracy rate is 91.21%, and in the Urban-Dense channel environment, the degree of matrix chaos is the highest, with an accuracy rate of 88.08%. Because there are fewer buildings and flat terrain in a Alpine–Montane Channel environment, it is less prone to reflection or diffraction, resulting in fewer multipath effects and relatively less noise interference. In Plain-Hillock Channel environment, there are not many obstacles or height differences, and there are moderate multipath effects and noise interference. There are many obstacles in the Urban-Dense environment, which can cause signal reflection and multipath effects, and there is a lot of noise interference in the Urban-Dense Channel. From this, it can be seen that our simulated channel environment is similar to the real channel environment, with significant differences among the three channel environments, which sets a good experimental scenario for the subsequent transfer experiment.

4.4 Performance comparison of different transfer methods

A model trained in one channel environment and tested in another channel environment often does not perform well, which requires domain adaptation technology to solve. Therefore, it is necessary to verify the effectiveness of specific emitter recognition under different domain adaptation methods. Based on recognition experiments in different channel environments, we observed the differences in the three channel environments of Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel. Therefore, we have set up six different migration scenarios, including Alpine–Montane Channel to Plain-Hillock Channel, Alpine–Montane Channel to Urban-Dense Channel, Plain-Hillock Channel to Alpine–Montane Channel, Plain-Hillock Channel to Urban-Dense Channel, Urban-Dense Channel to Alpine–Montane Channel, and Urban-Dense Channel to Plain-Hillock Channel. We validated the performance of the proposed method and other advanced domain adaptation methods through these migration scenarios. The performance comparison of different domain adaptation methods is shown in Table 1.

From Table 1, we can see that the E-DANN method proposed in this article has high accuracy in all six transfer scenarios. Relative to the other six methods, there is an improvement in accuracy of approximately 3%. E-DANN is an enhancement of DANN, designed to acquire domain-invariant feature representations by incorporating domain classifiers and domain adversarial losses, which is simple and effective. In addition, an ensemble classifier is introduced to improve the accuracy in each scenario by integrating the performance of different classifiers. We can also see that the recognition performance in the migration scenario from Plain-Hillock Channel to Alpine–Montane Channel is superior to other migration scenarios, because the signals in these two channel environments are highly similar and can provide more similar transferable features. In addition, we can also see that deep domain adaptive methods are superior to non deep domain adaptive methods, as deep domain adaptive methods can learn deeper transferable features.

4.5 The impact of the number of samples in the source domain and target domain on migration performance

Transfer learning cannot always assume that there are many samples in the source domain, therefore, it is essential to investigate the influence of the sample size in the source domain on domain adaptation methods. In this experiment, we set the sample quantity within the source domain to {1000, 2000, 3000, 4000, 5000, 6000, 7000}, while maintaining a constant number of samples in the target domain.

Figure 5 shows the recognition performance of different methods under different sample sizes in the source domain. It is evident that the accuracy rises with the increasing number of source domain samples. And the E-DANN method outperforms other methods in recognition performance under different sample sizes. Therefore, changes in sample size have an impact on domain adaptation effectiveness. However, the change in sample size has little impact on the E-DANN method, and even with a sample size of 1000, the accuracy of 90% can still be achieved. The TCA method is greatly affected by

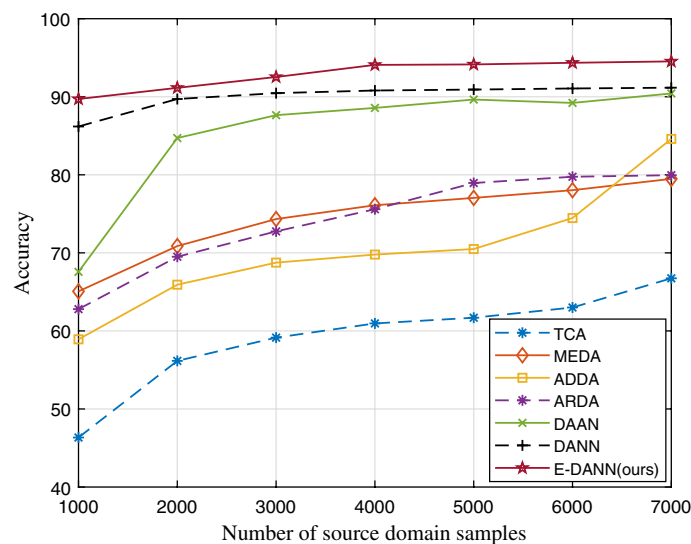


Fig. 5 Recognition performance under different sample sizes in source domain

the number of samples, as it cannot effectively find the shared space between the source and target domain when the number of samples in the source domain is small. MEDA is more complex than other shallow transfer methods, and it needs to deal with pop learning and dynamic distribution alignment. MEDA can maintain the intrinsic geometric and topological structure of the data and maintain the intrinsic properties of the data through manifold learning. MEDA introduces a dynamic distribution alignment mechanism that can adjust the alignment strategy based on feedback from the classifier. This is the reason why the performance of MEDA is higher than that of shallow migration method, even ADDA.

The quantity of samples within the target domain is also a key factor affecting the effectiveness of transfer learning. Therefore, it is also very important to study the impact of the number of unlabeled samples in the target domain on recognition performance. In this experiment, we set the number of samples in the target domain to {1000, 2000, 3000, 4000, 5000, 6000, 7000}, while maintaining a constant number of samples in the source domain.

Figure 6 shows the recognition performance of different methods under different sample sizes in the target domain. We can see that as the sample size increases, the recognition performance of different methods is basically not affected. The recognition performance of E-DANN method is higher than other methods under different sample sizes. When the number of samples in the target domain is 1000, an accuracy rate of over 90% can still be achieved. This indicates that good performance can also be achieved with a small number of samples.

4.6 Ablation experiment

This experiment mainly verifies the superiority of adding transformer encoder. We will refer to the DANN that adds a transformer encoder as DANN-Transformer. Figure 7 shows the performance comparison of DANN method and DANN-Transformer method for mutual migration in three different channel environments.

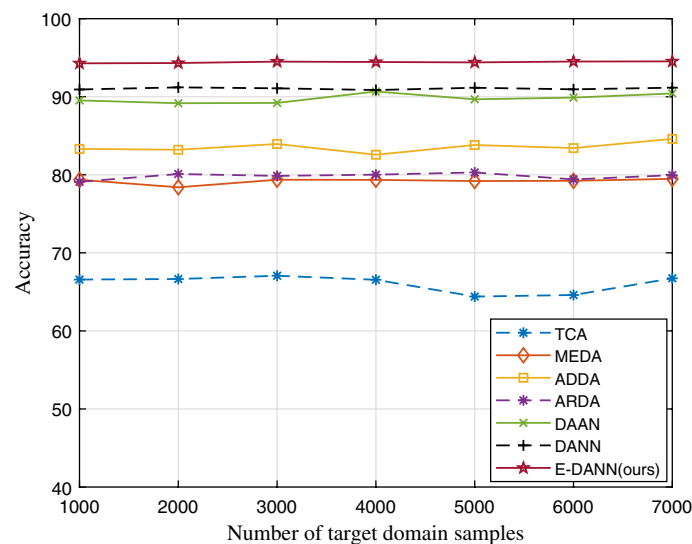


Fig. 6 Recognition performance under different sample sizes in target domain

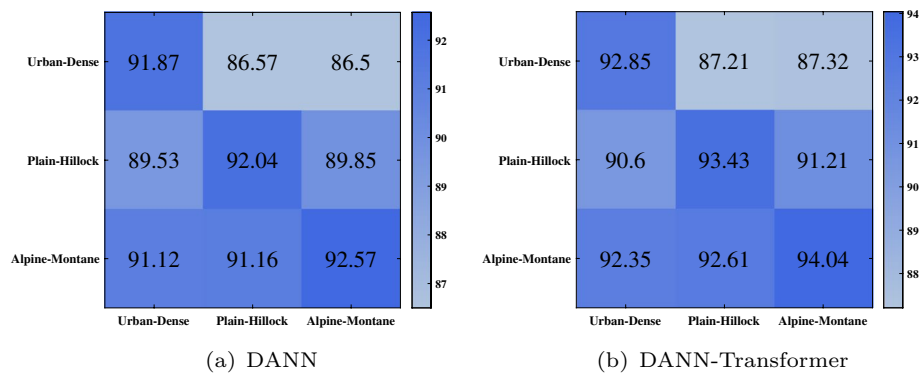


Fig. 7 The results of mutual migration between DANN and DANN Transformer methods in three different channel environments

We can see that the performance in any migration scenario after adding the transformer encoder is higher than that without adding it. Because transformer encoder has strong long-distance feature acquisition ability, it can better extract the contextual relevance of signals. We can also see that the values on the diagonal are the highest, indicating that the recognition performance is the best in the same migration scenario, as the signal features of the two are the most similar. The transformer encoder can make the output features follow a Gaussian distribution, providing deeper transferable features for better feature alignment.

5 Conclusion

In this article, we propose a method for identifying specific emitters using ensemble domain adversarial neural network. This method consists of a domain adversarial neural network based on transformer encoder and an ensemble learning classifier. Specifically, the former adds a transformer encoder after the feature extraction layer of the domain adversarial neural network, so that the extracted features from the source and target domains follow a Gaussian distribution after passing through the encoder, which is conducive to feature alignment. The latter utilizes the ensemble learning method of weighted voting to combine the results of multiple weak learners to improve recognition performance. The migration performance of the proposed method was evaluated in three environments: Alpine–Montane Channel, Plain-Hillock Channel, and Urban-Dense Channel, and compared with the other six methods. The simulation results show that the proposed method exhibits superior performance compared to the other six methods, with an accuracy improvement of about 3%. In addition, the impact of the sample quantity in the source and target domain on the adaptation effect of the migration domain was also analyzed. In the future, we hope to continue studying the impact of feature subspaces on migration performance.

Acknowledgements

The authors would like to acknowledge the anonymous reviewers and editors of this paper for their valuable comments and suggestions.

Author contributions

YB and YJF contributed to the design and writing of the study, and LDS supervised the study and advised on the revision of the manual and provided input on the revision of the draft manuscript. WJZ contributed to the data. LPT made some comments on the manuscript. SP provided some review for the revision of the manuscript. All authors have read and agreed to the manuscript.

Funding

This material is based upon unfunded work.

Availability of data and materials

Please contact author for data requests.

Declarations**Ethics approval**

Not applicable.

Consent for publication

The picture materials quoted in this article have no copyright requirements, and the source has been indicated.

Competing interest

(check journal-specific guidelines for which heading to use) The authors declare no competing interest.

Received: 21 December 2023 Accepted: 26 February 2024

Published online: 28 March 2024

References

1. Y. Tu, Y. Lin, C. Hou, S. Mao, Complex-valued networks for automatic modulation classification. *IEEE Trans. Veh. Technol.* **69**(9), 10085–10089 (2020)
2. S. Zheng, S. Chen, X. Yang, DeepReceiver: a deep learning-based intelligent receiver for wireless communications in the physical layer. *IEEE Trans. Cogn. Commun. Netw.* **7**(1), 5–20 (2020)
3. Y. Lin, H. Zhao, X. Ma, Y. Tu, M. Wang, Adversarial attacks in modulation recognition with convolutional neural networks. *IEEE Trans. Reliab.* **70**(1), 389–401 (2021)
4. Z. Bao, Y. Lin, S. Zhang, Z. Li, S. Mao, Threat of adversarial attacks on DL-based IoT device identification. *IEEE Internet Things J.* **9**(11), 9012–9024 (2022)
5. P. Sui, Y. Guo, H. Li, S. Wang, X. Yang, Wavelet packet and granular computing with application to communication emitter recognition. *IEEE Access.* **7**, 94717–94724 (2019)
6. Y. Lin, Y. Tu, Z. Dou, An improved neural network pruning technology for automatic modulation classification in edge devices. *IEEE Trans. Veh. Technol.* **69**(5), 5703–5706 (2020)
7. K. Tan, W. Yan, L. Zhang, Q. Ling, C. Xu, Semi-supervised specific emitter identification based on bispectrum feature extraction CGAN in multiple communication scenarios. *IEEE Trans. Aerosp. Electron. Syst.* **59**(1), 292–310 (2023)
8. Tao Wan, Hao Ji, Wanan Xiong, Bin Tang, Xueli Fang, Lei Zhang, Deep learning-based specific emitter identification using integral bispectrum and the slice of ambiguity function. *SIVIP* **16**(7), 2009–2017 (2022)
9. Bertoincini Crystal, Rudd Kevin, Nousain Bryan, Hinders Mark, Wavelet fingerprinting of radio-frequency identification (RFID) tags. *IEEE Trans. Ind. Electron.* **59**(12), 4843–4850 (2012)
10. J. Zhang, F. Wang, O.A. Dobre, Z. Zhong, Specific emitter identification via Hilbert–Huang transform in single-hop and relaying scenarios. *IEEE Trans. Inf. Forensics Secur.* **11**(6), 1192–1205 (2016)
11. J. Zhang, F. Wang, Z. Zhong, O. Dobre, Novel Hilbert spectrum-based specific emitter identification for single-hop and relaying scenarios, in *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6 (2015)
12. D.R. Reising, M.A. Temple, M.J. Mendenhall, Improved wireless security for GMSK-based devices using RF fingerprinting. *Int. J. Electron. Secur. Digit. Forensics* **3**(1), 41–59 (2010)
13. M.K.D. Williams, M.A. Temple, D.R. Reising, Augmenting bit-level network security using physical layer RF-DNA fingerprinting, in *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pp. 1–6 (2010)
14. V. Brik, S. Banerjee, M. Gruteser, S. Oh, Wireless device identification with radiometric signatures, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking*, pp. 116–127 (2008)
15. J. Liang, Z. Huang, Z. Li, Method of empirical mode decomposition in specific emitter identification. *Wirel. Pers. Commun.* **96**, 2447–2461 (2017)
16. Gençol Kenan, Kara Ali, At. Nuray, Improvements on deinterleaving of radar pulses in dynamically varying signal environments. *Digit. Signal Process.* **69**, 86–93 (2017)
17. H. Zha, H. Wang, Z. Feng, Z. Xiang, W. Yan, Y. He, Y. Lin, LT-SEI: long-tailed specific emitter identification based on decoupled representation learning in low-resource scenarios. *IEEE Trans. Intell. Transp. Syst.* 1–15 (2023)
18. C. Liu, X. Fu, Y. Wang, L. Guo, Y. Liu, Y. Lin, H. Zhao, G. Gui, Overcoming data limitations: a few-shot specific emitter identification method using self-supervised learning and adversarial augmentation. *IEEE Trans. Inf. Forensics Secur.* **19**, 500–513 (2024)
19. Z. Yao, X. Fu, L. Guo, Y. Wang, Y. Lin, S. Shi, G. Gui, Few-shot specific emitter identification using asymmetric masked auto-encoder. *IEEE Commun. Lett.* **27**(10), 2657–2661 (2023)
20. Y. Lin, H. Zha, Y. Tu, S. Zhang, W. Yan, C. Xu, GLR-SEI: green and low resource specific emitter identification based on complex networks and fisher pruning. *IEEE Trans. Emerg. Top. Computat. Intell.* (2023). <https://doi.org/10.1109/TETCI.2023.3303092>
21. X. Zhang, X. Chen, Y. Wang, G. Gui, B. Adebisi, H. Sari, F. Adachi, Lightweight automatic modulation classification via progressive differentiable architecture search. *IEEE Trans. Cogn. Commun. Netw.* **9**(6), 1519–1530 (2023)
22. X. Liu, Z. Liu, B. Lai et al., Fair energy-efficient resource optimization for multi-UAV enabled Internet of Things. *IEEE Trans. Veh. Technol.* **72**(3), 3962–3972 (2022)

23. X. Liu, B. Lai, B. Lin et al., Joint communication and trajectory optimization for multi-UAV enabled mobile internet of vehicles. *IEEE Trans. Intell. Transp. Syst.* **23**(9), 15354–15366 (2022)
24. N. Yang, B. Zhang, G. Ding, Y. Wei, G. Wei, J. Wang, D. Guo, Specific emitter identification with limited samples: a model-agnostic meta-learning approach. *IEEE Commun. Lett.* **26**(2), 345–349 (2022)
25. M. Wang, Y. Lin, Q. Tian, G. Si, Transfer learning promotes 6G wireless communications: recent advances and future challenges. *IEEE Trans. Reliab.* **70**(2), 790–807 (2021)
26. M. Wang, Y. Lin, H. Jiang, Y. Sun, TESPDA-SEI: tensor embedding substructure preserving domain adaptation for specific emitter identification. *Phys. Commun.* **57**, 101973 (2023)
27. X. Zhang, T. Li, P. Gong, X. Zha, R. Liu, Variable-modulation specific emitter identification with domain adaptation. *IEEE Trans. Inf. Forensics Secur.* **18**, 380–395 (2023)
28. R. Wei, J. Gu, S. He, W. Jiang, Transformer-based domain-specific representation for unsupervised domain adaptive vehicle re-identification. *IEEE Trans. Intell. Transp. Syst.* **24**(3), 2935–2946 (2023)
29. S.J. Pan, I.W. Tsang, J.T. Kwok, Q. Yang, Domain adaptation via transfer component analysis. *IEEE Trans. Neural Netw.* **22**(2), 199 (2011)
30. J. Wang, W. Feng, Y. Chen, H. Yu, M. Huang, Yu, Philip S, Visual domain adaptation with manifold embedded distribution alignment, in *Proceedings of the 26th ACM international conference on Multimedia*, pp. 402–410 (2018)
31. Y. Ganin, E. Ustinova, H. Ajakan et al., Domain-adversarial training of neural networks. *J. Mach. Learn. Res.* **17**(59), 1–35 (2016)
32. C. Yu, J. Wang, Y. Chen, M. Huang, Transfer learning with dynamic adversarial adaptation network, in *2019 IEEE international conference on data mining (ICDM)*, pp. 778–786 (2019)
33. E. Tzeng, J. Hoffman, K. Saenko, T. Darrell, Adversarial discriminative domain adaptation, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* pp. 7167–7176(2017)
34. J. Shen, Y. Qu, W. Zhang, Y. Yu, Adversarial representation learning for domain adaptation. arXiv preprint [arXiv:1707.01217](https://arxiv.org/abs/1707.01217) (2017)

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.