

RESEARCH

Open Access



An automatic modulation classification network for IoT terminal spectrum monitoring under zero-sample situations

Quan Zhou¹, Ronghui Zhang¹, Fangpei Zhang² and Xiaojun Jing^{1*}

*Correspondence:

jxiaojun@bupt.edu.cn

¹ School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China
Full list of author information is available at the end of the article

Abstract

Rely on powerful computing resources, a large number of internet of things (IoT) sensors are placed in various locations to sense the environment we live. However, the proliferation of IoT devices has led to the misuse of spectrum resources, and many IoT devices occupy the frequency band without permission. As a consequence, the spectrum regulation has become an essential part of the development of IoT. Automatic modulation classification (AMC) is a task in spectrum monitoring, which senses the electromagnetic space and is carried out under non-cooperative communication. Generally, deep learning (DL)-based methods are data-driven and require large amounts of training data. In fact, under some non-cooperative communication scenarios, it is challenging to collect the wireless signal data directly. How can the DL-based algorithm complete the inference task under zero-sample conditions? In this paper, a signal zero-shot learning network (SigZSLNet) is proposed for AMC under the zero-sample situations. The semantic descriptions and the corresponding semantic vectors are designed to generate the feature vectors of the modulated signals. The generated feature vectors act as the training data of zero-sample classes. The experimental results demonstrate the effectiveness of the proposed SigZSLNet. The accuracy of one unseen class and two unseen classes exceeds 90% and 76%, respectively. Simultaneously, we show the generated feature vectors and the intermediate layer output of the model.

Keywords: Internet of things, Automatic modulation classification, Zero-shot learning, Generative adversarial network

1 Introduction

The rapid development of the Internet of Things (IoT) and mobile networks is to meet users' demands for performance, speed, seamless connectivity, intelligence, and portability [1–6]. As communication technology continues to evolve to meet application needs [7–9], the 6th Generation Mobile Communication Technology (6G) is the next generation of telecommunication standards to ensure the needs of ever-growing intelligence communication applications. The IoT is a potential candidate for leveraging the resources of communication networks. It is estimated that there are currently over 50 billion devices connected to the internet. Based on these IoT devices to sense the surrounding environment and provide better services. Spectrum is the most

valuable resource in wireless networks, and the increase in IoT end devices may lead to spectrum resource abuse. Therefore, regulating the spectrum becomes an important task, and it is a non-cooperative communication method. Consequently, it is imperative to classify receiver modulation types under non-cooperative communication conditions. Some researchers have proposed an automatic classification modulation scheme called automatic modulation classification (AMC), which contributes to signal recognition, threat assessment, and spectrum monitoring [10].

For the urgent needs of spectrum regulation in IoT, AMC has attracted much attention in recent years. The conventional modulation recognition algorithms can be divided into likelihood-based (LB) methods and feature-based (FB) methods [11]. LB determined the modulation mode of the received signal by the likelihood comparison, which infers the labels by the Bayesian optimization. FB regarded AMC as a pattern recognition issue, and it yielded suboptimal classification accuracy although it was less sensitive to the uncertainties. The algorithm extracts the statistical features [12–14], instantaneous time features [15] and wavelet features [16] from the original modulated signal, and these features are used as input to the machine learning algorithm, such as Support Vector Machine (SVM), Lightboost, Decision Tree (DTree), K-Nearest Neighbors (KNN), [17–20], and infer the class to which they belong as a consequence. However, LB methods are usually of considerable computational complexity, prior knowledge-based accompanying sensitivity to channel noise, and poor robustness. FB methods are expert feature-based and suboptimum classification accuracy. Accordingly, both LB and FB are poorly adapted to non-cooperative communication environments.

Recently, Convolutional Neural Networks (CNNs) have been successfully applied to AMC by learning valid representations from complex data [21]. Reference [22] leveraged CNN network on AMC, and high classification accuracy was obtained, which proved the CNN-based methods are better than traditional feature-based methods. Reference [23] employed CNN, residual network (ResNet), Inception network, and Convolutional Long Short Term Memory Deep Neural Network (CLDNN) to process modulated signals, respectively, which demonstrated that the temporal feature was effective in improving the classification accuracy. Reference [24] considered feature interactions and combined the advantages of CNN and Long Short Term Memory (LSTM) to solve the AMC issue. Researchers propose a CNN-LSTM dual-stream structure to explore feature interactions and the spatial-temporal properties of the original signal. Simultaneously, the signal constellation was applied to AMC for the classification accuracy in [25]. Progressively, for better robustness and decent inference, a grid constellation matrix-based algorithm was considered in [26]. For the few-sample conditions, a host of researchers have made some explorations. Bu et al. [27] proposed an adversarial transfer learning architecture (ATLA) to reduce the differential on sampling rates, with 10% of the dataset for training, and achieved a favorable classification result. Li et al. [28] exploited a capsule network (CapsNet) to extract features from signals, with only 3% of the training data set, and obtained a promising inference result. Nevertheless, the schemes mentioned above work on the assumption that all categories of training sets have data, whether sufficient or few. In fact, these methods cannot work under class-imbalance situations, in which the training set misses data for some classes.

The data augmentation method that based on Generative Adversarial Networks (GAN) [29] is a possible way to address the issue of missing data for some classes, where the generator generates a specified data to fool the discriminator, and then the discriminator attempts to distinguish the real data from the generated data accurately. In the literature [30], the unstable convergence of GAN was solved, and an efficient approximation of the Wasserstein distance was provided for WGAN. For WGAN-GP [31], the gradient disappearance and the weight clipping were considered by an improved gradient penalty. Note that GAN-based data augmentation methods still require the corresponding data for the training and the augmentation. Under the zero-sample conditions, GAN-based data augmentation methods are not applicable anymore for the data imbalance issue. For the classes with entirely missing data, the zero-sample learning scheme is always considered by expert-set linguistic descriptions, transfer learning, or matrix transformations [32, 33]. However, the linear matrix transformation methods have a particularly low inferential accuracy under complex data scenes. Furthermore, the modulated signal does not have a semantic expert dataset. As a result, the methods based on semantic space mapping do not have modulated semantics.

In this paper, in order to better regulate the spectrum usage of IoT devices, a novel signal zero-shot learning network (SigZSLNet) is proposed to solve the issue of zero-sample mentioned above for AMC firstly, where the mapping relations between the classes are established by the semantic space mapping, and the expert-set linguistic descriptions are conducted for the modulated modes. GAN drives the generation of unseen classes, which enriches the training set. In summary, the contributions of this article can be concluded as

- (1) The semantic descriptions are designed for different modulated signals by their properties, and the semantic vectors are obtained based on the generation module of the semantic attribute vector.
- (2) The WGAN-GP based method is employed to generate the feature vectors of the modulated signals under the guidance of the semantic vectors. After that, a complete training dataset is constructed with the WGAN-GP module, including the real feature vector and the synthetic feature vector.
- (3) The complete training dataset is input to the classifier, and the estimation result of the AMC is obtained. Experimental results indicate that the proposed SigZSLNet can solve the zero-sample issue for AMC effectively.

To better present this paper, the remainder of this paper is organized as follows. Section II introduces the motivation and the description of the model. Section III presents the experiments of various groups, and the result is analyzed in detail. We discuss the result in section IV. Finally, Section V presents the conclusion of this article.

2 Method

2.1 Motivation

With the rapid development of 5G technology, more and more sensors and devices are connected to the internet, and the internet is transmitted through the wire and air network, satellite network, etc. As the number of participating end terminal devices

becomes more and more extensive, the information received by the central server becomes more and more complex. As the wireless bands become more compact, monitoring and management of wireless signals turn critical and challenging. As shown in Fig. 1, the scenarios of wireless spectrum management are considered in this paper, where the receiver does not have prior knowledge of the modulated signal at the transmitter. Accordingly, the classes of the modulated signal at the receiver may be different from the pre-collected data, and we also need to learn the details of the received signal. The task of non-cooperative communication is to recognize the modulation types of the wireless signal under the above-mentioned scenarios. Additionally, the training data collected in advance are called seen classes, and training data that are not available are called unseen classes. In this way, the available classes of the modulated signal at the receiver is $C_{\text{seen}} = \{c_1, c_2, \dots, c_{m-k}\}$, and the unavailable classes $C_{\text{unseen}} = \{c_{m-k+1}, c_{m-k+2}, \dots, c_m\}$. The wireless signal at receiver can be represented as

$$R(t) = s(t) * G(t) + n(t), \tag{1}$$

where $s(t)$ is the modulated signal from the transmitter, $G(t)$ is the channel, and $n(t)$ is the Additive White Gaussian Noise (AWGN). The goal of the whole work is to complete the training of the classifier based on the data of seen categories, making the classifier able to complete the inference of seen and unseen categories.

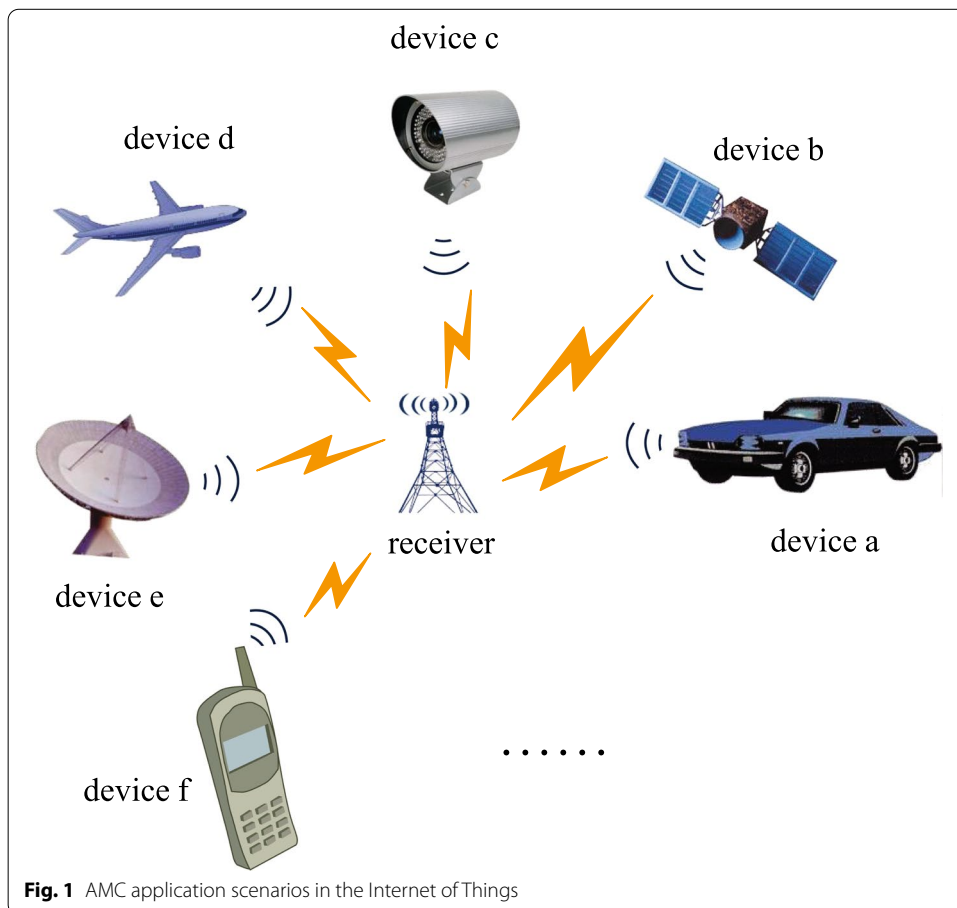


Fig. 1 AMC application scenarios in the Internet of Things

$$L = f(T_{\text{seen}}, T_{\text{unseen}} | C_{\text{seen}}), \quad (2)$$

where the T_{seen} and T_{unseen} are the test signal data of seen and unseen classes, respectively. L is the label of the test data.

Generally, the common DL-based methods [21] can infer the received data with an adequate training data set. To make the DL-based method work under zero-sample conditions, this paper mainly concerns how to generate the missing data C_{unseen} accurately with WGAN-GP and how to conduct AMC on the basis of the seen data and the generated data.

Algorithm 1 Semantic Vector Generation

Input: $C_{\text{seen}}, D_{\text{seen}}, D_{\text{unseen}}$

Output: $S_{\text{seen}}, S_{\text{unseen}}$

```

for  $i = 0$  to  $n_1$  do
   $y_1 = f_{\text{cov}}(C_{\text{seen}})$ ;
   $y_2 = f_{\text{enc}}(D_{\text{seen}})$ ;
  Optimizing  $Loss = L_A(y_1, y_2)$ 
end for
loop
   $y_{\text{seen}} = f_{\text{enc}}(D_{\text{unseen}})$ ;
   $y_{\text{unseen}} = f_{\text{enc}}(D_{\text{unseen}})$ ;
   $S_{\text{seen}} = y_{\text{seen}}$ ;
   $S_{\text{unseen}} = y_{\text{unseen}}$ 
end loop

```

Algorithm 2 Algorithm for Unseen Classes Generation

Input: $C_{\text{seen}}, S_{\text{seen}}, S_{\text{unseen}}$

Output: y_{unseen}

```

1: for  $i = 0$  to  $n_2$  do
2:    $y_G = f_G(S_{\text{seen}}, G_{\text{noise}})$ ;
3:    $y_{\text{seen}} = f_{\text{CNN}}(C_{\text{seen}})$ ;
4:    $\{0, 1\} = f_D(y_{\text{seen}}, y_G)$ ;
5: end for
6: loop
7:    $y_{\text{unseen}} = f_G(D_{\text{unseen}})$ ;
8:    $C_{\text{unseen}} = y_{\text{unseen}}$ 
9: end loop

```

Algorithm 3 Algorithm for Classifier Training

Input: $\{C_{\text{seen}}, C_{\text{unseen}}\}, \{l_{\text{seen}}, l_{\text{unseen}}\}$

Output: $\theta_{\text{classifier}}$

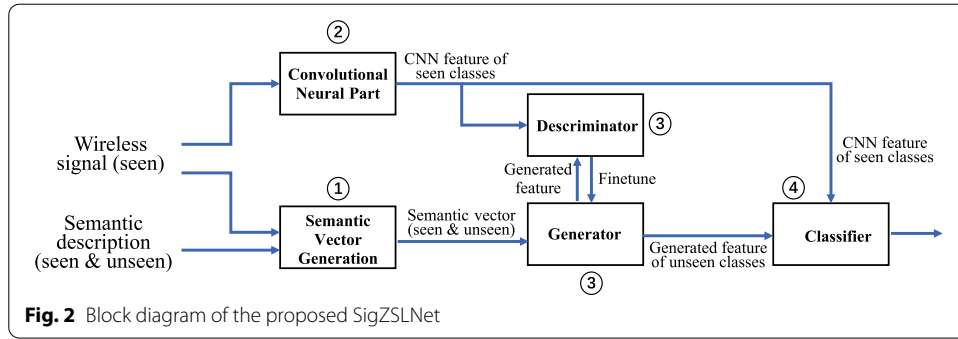
```

1: for  $i = 0$  to  $n_3$  do
2:    $y_{\text{label}} = f_{\text{classifier}}(C_{\text{seen}}, C_{\text{unseen}} | \theta_{\text{classifier}})$ ;
3:    $loss = L_C(y_{\text{label}}, \{C_{\text{seen}}, C_{\text{unseen}}\})$ ;
4:   Optimizing  $loss$ ;
5: end for

```

2.2 Model description

The overall process of our proposed method is shown in Fig. 2, which consists of three subsystems: semantic vector generation, generated module, and classification subsystems.



In the former subsystem, there are two inputs: the wireless signal (seen) and semantic description (seen and unseen). Accordingly, the data of seen signal and the semantic description of seen classes are leveraged to train the generation module of the semantic attribute vector. Consequently, the semantic vector of unseen classes is obtained by the semantic description of unseen classes. Then, in step 2, the original wireless signal of seen types and the pre-trained convolutional neural part are employed to yield the CNN feature vector of seen classes. Next, the generated semantic vector in step 1 and the CNN feature in step 2 are used to generate CNN features of unseen types in step 3. Finally, the CNN feature of seen classes and the generated feature vector of unseen classes form the training set, which is employed to classification subsystem in step 4.

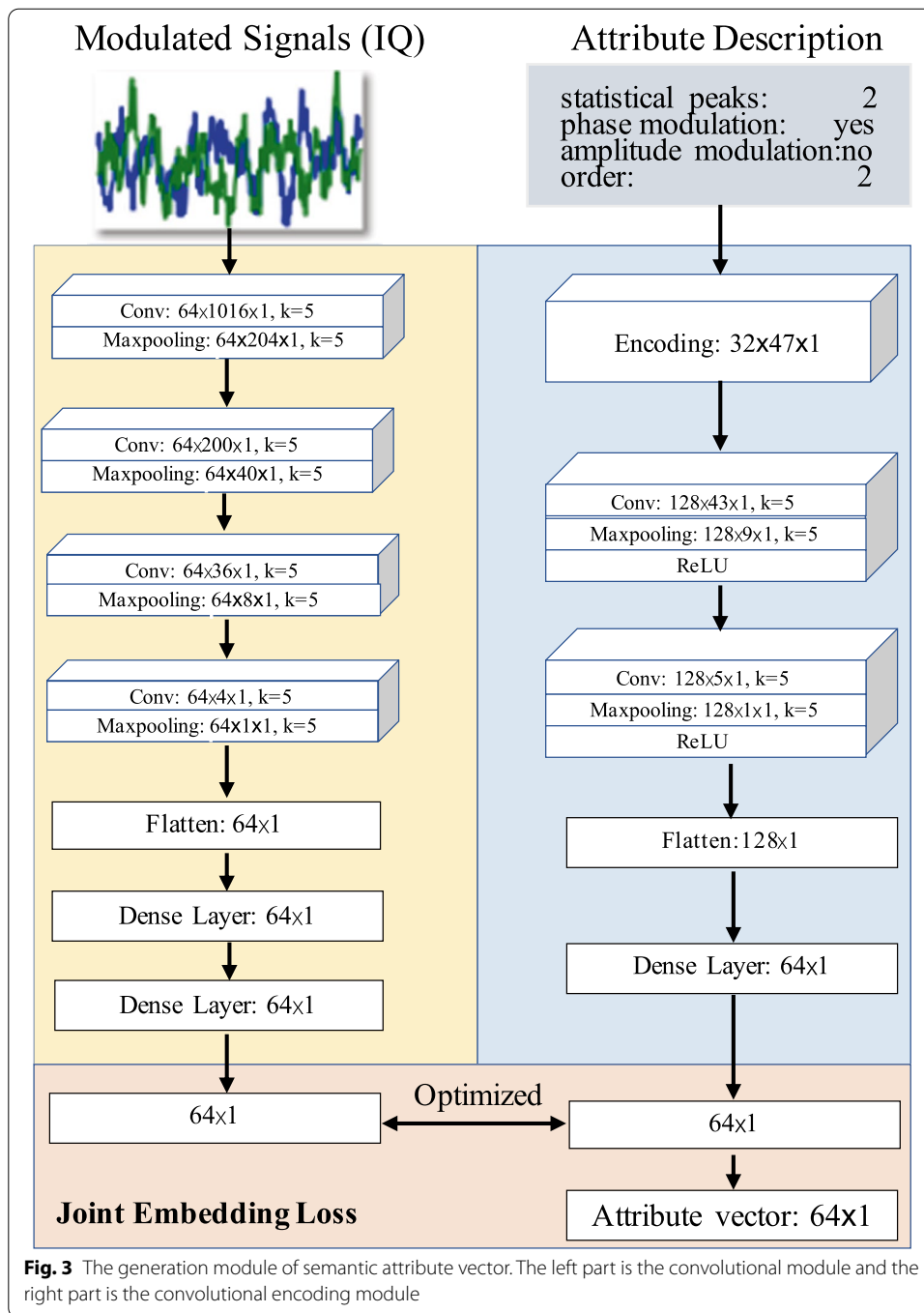
The generation module of the semantic attribute vector is shown in Fig. 3. The network architecture of the proposed SigZSLNet is exhibited in Fig. 4. From Fig. 4, SigZSLNet contains three modules, the CNN module, the WGAN-GP module, and the Classifier module.

2.2.1 Semantic generation

As it is shown in Fig. 3, the generation module of the semantic attribute vector contains the convolutional part and the encoding part, where the convolutional part is to extract the signal data features, and the convolutional encoding part is to encode semantic descriptions and extract semantic features. The layer of semantic generation module and output dimensions of each layer are represented in Table 1. More specifically, various modulated signals are related to their property descriptions. As a result, the property descriptions of different modulated signals are quantified with one-hot encoding in this letter, and the quantification results form the specific semantic vectors, which can guide the WGAN-GP to generate specific data. After the loss function optimization, a semantic feature vector is obtained that can be used as the input to the generator to generate the specific data.

The measured relations between the semantic description and the signal are optimized by Joint Embedding Loss [34], which is formulated as

$$\begin{aligned}
 L_A &= \frac{1}{N} \sum_{n=1}^N \max l_s(s_n, t_n, y_n) + l_t(s_n, t_n, y_n) \\
 &= \frac{1}{N} \sum_{n=1}^N \max(0, \Delta(y_n, y) + \mathbb{E}[F(s_n, t) - F(s_n, t_n)]) \\
 &\quad + \max(0, \Delta(y_n, y) + \mathbb{E}[F(s, t_n) - F(s_n, t_n)])
 \end{aligned} \tag{3}$$



$$\Delta(a, b) = \begin{cases} 1, & \text{if } a \neq b \\ 0, & \text{if } a = b \end{cases} \quad (4)$$

where N denotes the total number of the signal and text pairs; l_s and l_t are the loss of the signal and the semantic, respectively; Δ signifies the 0–1 loss; y denotes the class label; t is the subset of text descriptions, and y_n is the n -th text descriptions; s signifies the subset of signals; $\mathbb{E}[\cdot]$ denotes the mean operation; $F(s, t) = e(s)^T \varphi(t)$, where $e(s)$ is the

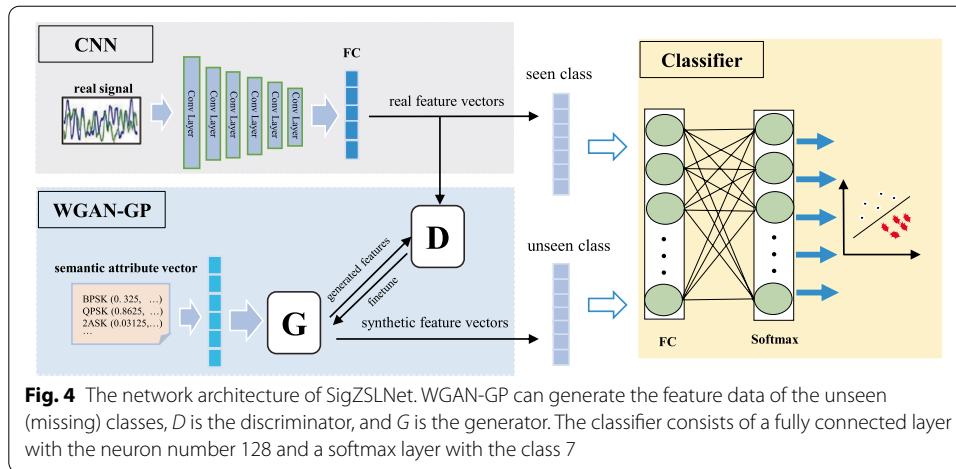


Table 1 Layers of semantic generation module and output dimensions of each layer

Modules	Layers	Output	
Convolutional part	Input	$2 \times 1024 \times 1$	
	Conv1D (filters 64, size 1×5)	$64 \times 1016 \times 1$	
	Maxpooling (size 1×5)	$64 \times 204 \times 1$	
	Conv1D (filters 64, size 1×5)	$64 \times 200 \times 1$	
	Maxpooling (size 1×5)	$64 \times 40 \times 1$	
	Conv1D (filters 64, size 1×5)	$64 \times 36 \times 1$	
	Maxpooling (size 1×5)	$64 \times 8 \times 1$	
	Conv1D (filters 64, size 1×5)	$64 \times 4 \times 1$	
	Maxpooling (size 1×5)	$64 \times 1 \times 1$	
	Flatten	64×1	
	Fully connected	64×1	
	Fully connected	64×1	
	Encoding part	Encoding	$32 \times 47 \times 1$
		Conv1D (filters 128, size 1×5)	$128 \times 43 \times 1$
Maxpooling (size 1×5)		$128 \times 9 \times 1$	
ReLU		–	
Conv1D (filters 128, size 1×5)		$128 \times 5 \times 1$	
Maxpooling (size 1×5)		$128 \times 1 \times 1$	
ReLU		–	
Flatten		128×1	
Fully connected	64×1		

encoder of the input signals, and $\varphi(t)$ is the encoder of the input attribute description. The algorithm for semantic generation module is shown in Algorithm 1, where C_{seen} is the signal of seen classes, D_{seen} is the semantic description of C_{seen} , D_{unseen} represents the semantic description of unseen classes. S_{seen} and S_{unseen} are the output semantic vector of seen classes and unseen classes.

2.2.2 CNN module

The CNN module is leveraged to obtain the feature vectors, which consists of five convolutional layers, one fully connected layer. In detail, the convolution layer

includes a convolution with the stride of 1×1 and the convolution kernel of 1×8 , an activation function $f_{\text{relu}} = \max(0, x)$, and a maximum pooling with the stride of 1×2 and the kernel of 1×2 . The output of the last fully connected layer is 128 dimensions. The details are depicted in Table 2. In general, the main role of CNN module is to extract spatial features of modulated signal data. Similarly, for CNN's ability to extract spatial features, the open source pre-trained CNN model with convolutional module can be used as the CNN module in the proposed SigZSLNet. The pre-training weights of the model proposed in [22] can be found at GitHub. In this way, we utilize this model as a CNN module of SigZSLNet, using the output of the first fully connected layer as the feature vector of the signal.

2.2.3 GAN module

The main motivation of the GAN module is to generate the feature vectors of the unseen (missing) classes, which contains the generator (G) part and the discriminator (D) part. The generator consists of two fully connected layers with 256 neurons in the middle layer and 128 outputs, and the activation function in each layer is *leaky_relu*. Similarly, the discriminator consists of two fully connected layers with 256 neurons, and the activation function in the first layer is *leaky_relu*, while the second layer has no activation function. Table 2 shows the detailed structure of WGAN-GP.

The input to the generator is a semantic attribute vector of the modulated signal, which is generated in the semantic generation module. The input of the discriminator is a 128-dimensional attribute vector generated by the generator. The generator and the discriminator are playing a game called Min–Max, where the generator tries to generate data from the semantic attribute vector that can be judged as true by the discriminator. Simultaneously, the discriminator tries to distinguish between the real feature data from the CNN module and the synthetic data generated by the generator. After several epochs, the generator generates modulated signal data that makes the discriminator difficult to distinguish. In this way, the generated feature data can substitute for the real data. The Wasserstein GAN with gradient penalty (WGAN-GP) objective function is employed in the process we trained, which is proposed by Gulrajani et al. [35]. The optimization objective is expressed as

$$\begin{aligned} \min_G \max_D L_{GAN} = & -E_{x \sim p_r}[D(x)] + E_{x \sim p_g}[D(x)] \\ & + \lambda E_{x \sim p_g}[\|\nabla_x D(x)\|_2 - 1]^2 \end{aligned} \quad (5)$$

where $x \sim p_r$ denotes the real data and $x \sim p_g$ signifies the generated data from the generator. $[\|\nabla_x D(x)\|_2 - 1]^2$ is the gradient penalty, while λ serves as the penalty coefficient with the default $\lambda = 10$. **Algorithm 2** describes the generation process of the missing signal classes, where $f_G(\cdot)$ denotes the generator, $f_D(\cdot)$ denotes the discriminator, $f_{CNN}(\cdot)$ denotes the CNN part, y_{unseen} is the generated feature vector of unseen classes.

Table 2 Layers of SigZSLNet and the output dimensions of each layer

Module	Layer	Output
CNN	Input Layer	1024 × 2
	Conv1D (filters 64, size 1 × 8)	1024 × 64
	Maxpooling (size 1 × 2)	512 × 64
	Conv1D (filters 64, size 1 × 8)	512 × 64
	Maxpooling (size 1 × 2)	256 × 64
	Conv1D (filters 64, size 1 × 8)	256 × 64
	Maxpooling (size 1 × 2)	128 × 64
	Conv1D (filters 64, size 1 × 8)	128 × 64
	Maxpooling (size 1 × 2)	64 × 64
	Conv1D (filters 64, size 1 × 8)	64 × 64
	Maxpooling (size 1 × 2)	32 × 64
	Flatten	2048
	Fully connected layer	128
	G	Input layer
Fully connected layer		256
Leaky_relu		–
Fully connected layer		128
D	Input layer	64
	Fully connected layer	256
	Leaky_relu	–
	Fully connected layer	128
Classifier	Input layer	128
	Fully connected layer	128
	PReLU	–
	SoftMax	7

G is the generator module, and D is the discriminator module

2.2.4 Classifier module

The classifier module contains a fully connected layer and a *Softmax* layer. Specifically, the fully connected layer has 128 neurons. *PReLU* is used as the activation functions in the fully connected layer, and *Softmax* is the judicial function of the last layer, which are, respectively, formulated as

$$f_{\text{PReLU}}(x_i) = \begin{cases} x_i, & \text{if } x_i > 0 \\ \lambda x_i, & \text{if } x_i \leq 0 \end{cases} \quad (6)$$

$$f_{\text{Softmax}}(x_i) = \frac{e^{x_i}}{\sum_i e^{x_i}} \quad (7)$$

where $\lambda \in (0, 1)$ is a variable that can be learned by a backpropagation algorithm, and adjust to the most appropriate slope value.

The training stage focuses on training the classifier to have the ability to classify seen and unseen classes. First, the semantic attribute vector is generated based on the semantic description. In this way, the semantic vectors of the seen and unseen categories are obtained. Then, WGAN-GP is trained by the seen classes and their semantic vectors. Consequently, synthetic feature vectors for the unseen categories are generated from the

generator based on their semantic attribute vectors. Finally, both the real feature vectors and the synthetic feature vectors are input to the Classifier module. After processing by the full connection layer and the *Softmax* layer, the prediction result of AMC is obtained. The final loss function is formulated as

$$L_C = h(y, f(x_{\text{seen}}, x_{\text{unseen}} | \theta)) \quad (8)$$

where h is the cross entropy, which is defined as $h(p, q) = -\sum_i p_i \log(q_i)$. y denotes the true label and f signifies the classifier; θ is the weight of the fully connected layer, and x_{seen} is the CNN feature of the seen classes; x_{unseen} denotes the unseen class.

The test stage is concerned with determining of the category to which the received modulated signal data belongs. Generally, In an end-to-end AMC system, the modulated signals received by the receiver may belong to the seen class or the unseen class of modulated signals. The feature vector is obtained after feature extraction by the CNN module, and then inferred by the classifier module to obtain the inference results.

3 Experimental

3.1 Dataset and settings

The experimental dataset of the in-phase and quadrature (IQ) samples is obtained based on the MATLAB 2019b platform. The dataset consists of seven modulated modes, Group = {BPSK, QPSK, 2ASK, 4ASK, 16QAM, 32QAM, 64QAM} (SNRs range from 0 to 10 dB, with the stride 1 dB). For each class, there are 400 modulated signals for each SNR, where 300 of them are for training and the rest are for the test. To sum up, the training set contains 23,100 modulated signals, while the test set contains 7700 modulated signals. The semantic description of each modulated mode is shown in Table 3, where the “statistical peaks” is the peak number of the modulated signal.

Four groups of the seen classes are considered in the experiments, which are presented in Table 4. For Group 1, we do not have the class of *BPSK* for training. Consequently, we utilize the data of seen classes {QPSK, 2ASK, 4ASK, 16QAM, 32QAM, 64QAM} and the description semantic of all classes to generate unseen classes. Group 2 is similar. Group 3 and Group 4 are tested under two unseen classes conditions. Similarly, for Group 3, we do not have {QPSK, 4ASK}, and leverage {BPSK, 2ASK, 16QAM, 32QAM, 64QAM} to obtain unseen classes. For Group 4, we do not have {QPSK, 4ASK}.

The attribute feature vector is normalized in this letter because it can accelerate the training and prevent overfitting. Simultaneously, the Gaussian Noise is added to the input of the generator because the Gaussian data are easier to map into the CNN feature distribution. Additionally, the harmonic mean is considered to reflect the inference ability, and it is defined as

$$H = \frac{2su}{s + u} \quad (9)$$

where s denotes the average classification accuracy for the seen classes, and u is for the unseen classes. The implementation of the SigZSLNet is based on Tensorflow.

Table 3 The semantic description of each category

Classes	Statistical peaks	Phase modulation	Amplitude modulation	Order
BPSK	2	Yes	No	2
QPSK	2	Yes	No	2
2ASK	4	No	Yes	2
4ASK	4	No	Yes	4
16QAM	4	Yes	Yes	16
32QAM	6	Yes	Yes	32
64QAM	8	Yes	Yes	64

Table 4 The semantic description of each category

Groups	Seen classes	Unseen classes
Group 1	QPSK, 2ASK, 4ASK, 16QAM, 32QAM, 64QAM	BPSK
Group 2	BPSK, QPSK, 4ASK, 16QAM, 32QAM, 64QAM	2ASK
Group 3	BPSK, 2ASK, 16QAM, 32QAM, 64QAM	QPSK, 4ASK
Group 4	BPSK, QPSK, 2ASK, 16QAM, 32QAM	4ASK, 64QAM

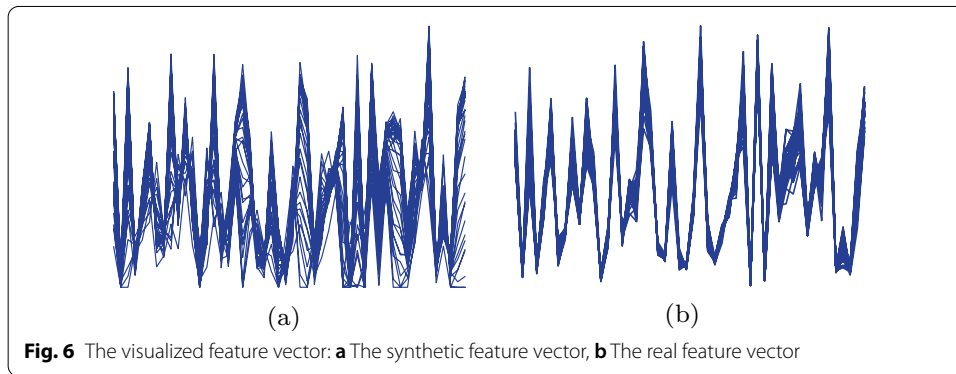
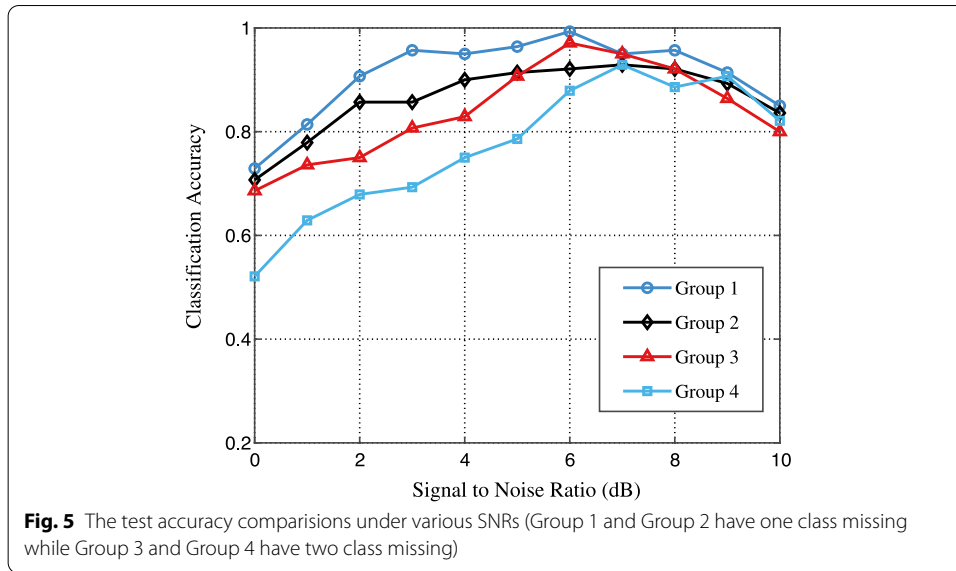
Table 5 The performance comparison among various methods

Methods	Groups	Seen	Unseen	<i>H</i>	Accuracy
SigZSLNet	1	0.8924	1	0.9431	0.9078
	2	0.9364	0.8515	0.8919	0.9243
	3	0.88	0.7114	0.7868	0.8318
	4	0.7991	0.6932	0.7424	0.7688

3.2 Results

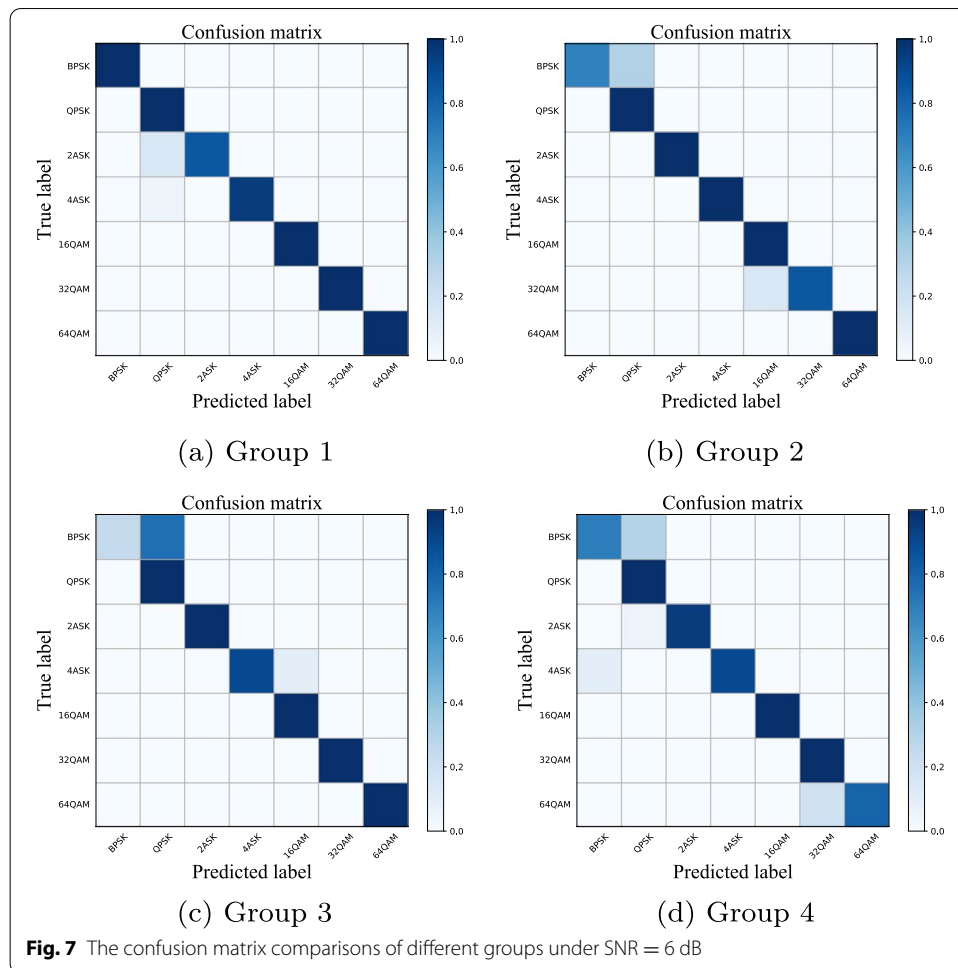
Table 5 provides the performance comparisons of SigZSLNet among various groups. In addition, there is one missing class in Group 1 and Group 2, while Group 3 and Group 4 have two classes of data missing. The recognition accuracy in Table 5 is the average value of five experiments. In fact, the convolutional neural part of the pre-trained model can employ in the CNN part of the proposed SigZSLNet. In the experiment, we leverage the ResNet pre-trained on rml2018.a [22] to serve as the CNN module of the SigZSLNet. As shown in Table 5, the average classification accuracy of the proposed SigZSLNet exceeds 76%. In detail, the accuracy of Group 1 and Group 2 exceeds 85%, which indicates that SigZSLNet can effectively conduct AMC under zero-sample situations. Besides, compared with the recognition accuracy of Group 1 and Group 2, the performance of Group 3 and Group 4 declines. With the increase in missing categories, the data quality of generating different missing categories decreases.

The reason for the above result is the limited generation ability of the proposed two fully connected layers. With the increase in missing categories, the data quality of generating decreases. As a result, each class of experiment data has its unique embedding space, and it is difficult for generate models to map various different distribution



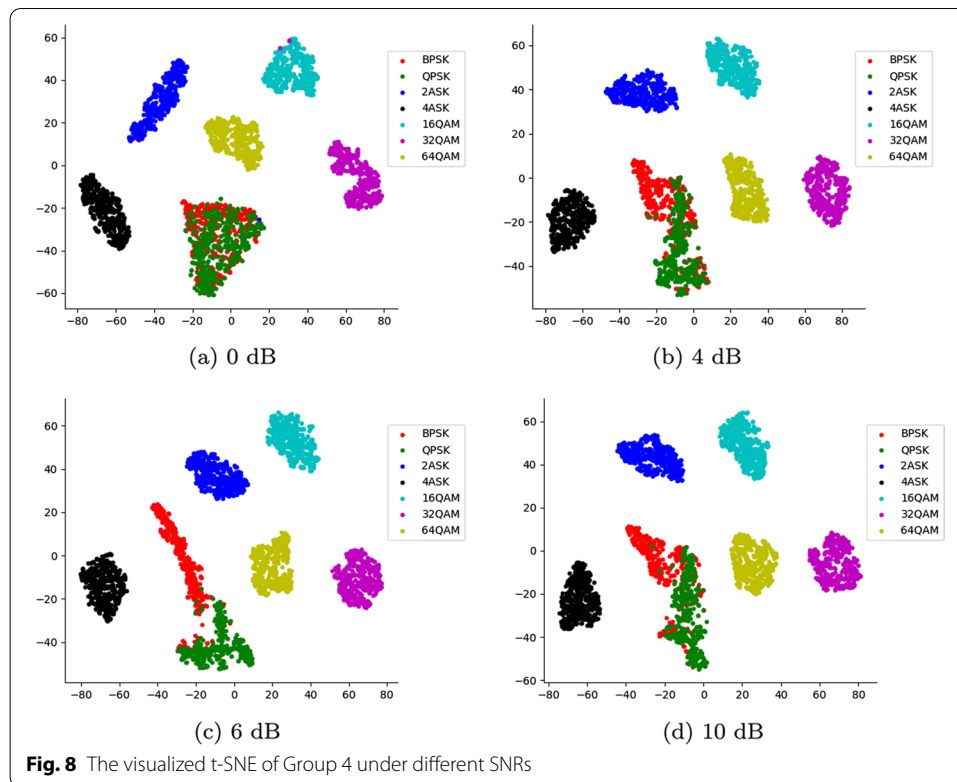
spaces at the same time. Simultaneously, the modulated signal is highly susceptible to the signal-to-noise ratio. Under low SNR conditions, the signal is strongly disturbed, the features are not obvious, and the features learned by the generator are not obvious. In experiments, we mixed low SNR and high SNR data, and the model was unable to learn a valid feature representation. Consequently, the recognition accuracy decreases due to the fall of the data generation quality. In total, Table 5 states that the proposed SigZSLNet is an AMC scheme more applicable to real scenarios.

The test accuracy comparisons under various SNRs are shown in Fig. 5, where the settings of Group 1-4 are the same as that of Table 5. From Fig. 5, the recognition accuracy improves with the rise of SNR firstly and then slightly declines with the rise of SNR when the SNR is above 6 dB. It indicates that SNR is an important influence factor for the recognition accuracy of AMC. Particularly, the average classification accuracy exceeds 85% of various groups when SNR varies from 6 dB to 9 dB because GAN balances the quality of the generated feature vector. In addition, as discussed in Table 5, the recognition accuracy of the proposed SigZSLNet gradually deteriorates with the rise of the unseen classes.



In Fig. 6, 50 synthetic feature vectors and 50 real feature vectors of the random 2ASK modulation signal are visualized. From Fig. 6, although there is much difference between the synthetic feature vector and the real feature vector, their similarity is at a high level, especially for the waveform variation trend. The generated feature vectors have great similarity with the real feature vectors. This means that SigZSLNet can generate the feature vectors of zero-sample classes accurately.

As a supplement, the confusion matrix comparisons of various groups are made under SNR = 6 dB in Fig. 7, where the BPSK signal is missing in Group 1 and exists in Group 2–4. Figure 7 indicates that the classification accuracy of Group 1 is obviously higher than that of Group 2–4 for the BPSK signal, which manifests the generation quality of the synthetic feature vectors by the GAN module outperforms that of the real feature vectors by the CNN module. This states that the proposed SigZSLNet can effectively fill the missing classes and improve the accuracy of AMC under zero-sample situations. Simultaneously, for the computational complexity of the proposed model, the generator and discriminator separately consist of two fully connected layers, and the WGAN module and the classifier contain 91,009 parameters, which make the network converge easily. For the training process, the total number of training data is 20900, and the training process takes 0.628 seconds in each epoch.



In Fig. 8, we show the visualized results of the feature vectors before input to the classifier module. The 128-dimensional features are downscaled into two-dimensional coordinate vectors by the t-SNE algorithm [36]. Figure 8a–d shows the visualization of the seen and unseen classes of Group 4 with different signal-to-noise ratios for the feature vectors. The classification accuracy of the classifier is highest at the SNR of about 6 dB, as described in the previous experimental results. For the results of dimensionality reduction visualization, at a signal-to-noise ratio of 6 dB, different classes are clustered together, and each class is easier to distinguish. When SNRs = {0 dB, 4 dB, 10 dB}, the reduced-dimensional features of BPSK and QPSK are mixed together and not easily distinguished. However, for 4ASK and 64QAM, which are unseen categories, the generated data are distinguishable. It is concluded that the algorithm has the highest classification accuracy at a signal-to-noise ratio of 6 dB because the CNN feature vectors of BPSK and QPSK are easily confused. Thus, the result of difficulty distinguishing between BPSK and QPSK is that the pre-trained CNN feature vector outputs low-quality feature vectors.

4 Discussion

Simulation results show that the SigZSLNet proposed in this paper can generate data of missing modulation signal to make up for that data in the training set and solve zero-sample in AMC. In the groups where one category was missing, and two categories were missing, the accuracy rate exceeded 76% in the task of classifying seven categories. However, it is known from the simulation results that the

generated modulated signals lack diversity and can only generate data of modulated signals with a single SNR. The model needs to be further improved in terms of SNR diversity to generate rich modulated signal SNRs. In addition, the poor robustness of the open-source CNN model, which is only trained with 24 categories of modulated signals, makes it difficult to excel in our experiments. In future work, we will collect rich modulated signal data, making a dataset like Imagenet [37], and CoCo [38], to train more robust pre-trained models.

5 Conclusion

The increasing number of IoT devices means that more traffic will occupy the scarce available spectrum in the future, so it becomes extremely important to regulate and recognize the observed signals. However, in complex electromagnetic environments, some classes of modulated signals cannot be collected in advance to train the classifier, which requires us to find a way to address the recognition of signals in zero-sample conditions. In this paper, we first propose the method SigZSLNet to implement AMC under zero-sample conditions. Based on the semantic feature vector, the feature vectors of the missing modulated signals are generated with WGAN, the classification accuracy of unseen classes has been greatly improved.

Simultaneously, the various groups' experimental results validate the effectiveness of the proposed SigZSLNet. The proposed method obtains an average accuracy of over 85% in the missing category when one modulation signal is missing, corresponding to an accuracy of over 76% for the classification over seen and unseen classes. While two modulation signals are missing, the proposed algorithm obtains an average accuracy of more than 69% in the missing category, corresponding to an accuracy of more than 76% for the seven classification tasks. We visualized the generated feature vectors for comparison. Also, We visualized the data clustering for each category based on the PCA algorithm and t-SNE algorithm to prove the validity of the generated data. In conclusion, the experimental results show that our proposed method effectively solves the AMC task of spectrum resource management for IoT terminal devices.

Abbreviations

AMC: Automatic modulation classification; IoT: Internet of things; SigZSLNet: Signal zero-shot learning network; GAN: Generative adversarial network; 6G: The 6th generation mobile communication technology; LB: Likelihood-based; FB: Feature-based; CNNs: Convolutional neural networks; DL: Deep learning; CLDNN: Convolutional long short term memory deep neural network; ATLA: Adversarial transfer learning architecture; CapsNet: Capsule network; SVM: Support vector machine; DTree: Decision tree; LightGB: Lightgbost; KNN: K-nearest neighbors; IQ: In-phase and quadrature; BPSK: Binary phase shift keying; ASK: Amplitude shift keying; QPSK: Quadrature phase shift keying; QAM: Quadrature amplitude modulation; SNR: Signal-to-noise ratio; PReLU: Parametric rectified linear unit; t-SNE: t-distribution domain embedding algorithm.

Authors' contributions

Q.Z was in charge of the major theoretical analysis, algorithm design, experimental simulation, and paper writing. X.J.J and Q.Z had contributions to paper writing. F.P.Z and R.H.Z had contributions to theoretical analysis and gave suggestions of the organization. All authors read and approved the final manuscript.

Funding

This research was solely the work of the authors, funded by no authority.

Availability of data and materials

Data sharing not applicable to this article as no datasets were generated or analyzed during the current study.

Declarations

Competing interests

The authors declare no conflict of interest.

Author details

¹School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China. ²Information Science Research Institute of China Electronics Technology Group Corporation, Beijing, China.

Received: 25 September 2021 Accepted: 2 March 2022

Published online: 25 March 2022

References

1. C. Yuanhao, L. Fan, J. Xiaojun, M. Junsheng, Integrating sensing and communications for ubiquitous iot: Applications, trends and challenges. arXiv preprint [arXiv:2104.11457](https://arxiv.org/abs/2104.11457) (2021)
2. M. Junsheng, J. Xiaojun, Z. Yangying, G. Yi, Z. Ronghui, Z. Fangpei, Machine learning-based 5g ran slicing for broadcasting services. *IEEE Trans. Broadcast.* (2021). <https://doi.org/10.1109/TBC.2021.3122353>
3. Z. Ronghui, J. Xiaojun, W. Sheng, J. Chunxiao, M. Junsheng, Y.F. Richard, Device-free wireless sensing for human detection: the deep learning perspective. *IEEE Internet Things J.* **8**(4), 2517–2539 (2021). <https://doi.org/10.1109/JIOT.2020.3024234>
4. W. Zijie, L. Rongke, L. Qirui, S. John, T.K. Michel, Energy-efficient data collection and device positioning in uav-assisted iot. *IEEE Internet Things J.* **7**(2), 1122–1139 (2020). <https://doi.org/10.1109/JIOT.2019.2952364>
5. L. Qiang, S. Songlin, R. Bo, K. Michel, Intelligent reflective surface based 6g communications for sustainable energy infrastructure. *IEEE Wirel. Commun.* **28**(6), 49–55 (2021). <https://doi.org/10.1109/MWC.016.2100179>
6. Z. Ronghui, J. Chunxiao, W. Sheng, Z. Quan, J. Xiaojun, M. Junsheng, Wi-Fi Sensing for joint gesture recognition and human identification from few samples in human-computer interaction. *IEEE J. Sel. Areas Commun.* <https://doi.org/10.1109/JSAC.2022.3155526>
7. C. Na, R. Bo, Z. Xinran, K. Michel, Scalable and flexible massive mimo precoding for 5g h-cran. *IEEE Wirel. Commun.* **24**(1), 46–52 (2017). <https://doi.org/10.1109/MWC.2017.1600139WC>
8. C. Lei, F.Y. Richard, J. Hong, R. Bo, L. Xi, C.M.L. Victor, Green full-duplex self-backhaul and energy harvesting small cell networks with massive mimo. *IEEE J. Sel. Areas Commun.* **34**(12), 3709–3724 (2016). <https://doi.org/10.1109/JSAC.2016.2611846>
9. N. Ahasanun, K. Michel, R. Bo, Fountain coded cooperative communications for lte-a connected heterogeneous m2m network. *IEEE Access* **4**, 5280–5292 (2016). <https://doi.org/10.1109/ACCESS.2016.2601031>
10. M. Junsheng, J. Xiaojun, H. Hai, G. Ning, Subspace-based method for spectrum sensing with multiple users over fading channel. *IEEE Commun. Lett.* **22**(4), 848–851 (2018)
11. O.A. Dobre, M. Oner, S. Rajan, R. Inkol, Cyclostationarity-based robust algorithms for qam signal identification. *IEEE Commun. Lett.* **16**(1), 12–15 (2012)
12. S. Wei, Feature space analysis of modulation classification using very high order statistics. *IEEE Commun. Lett.* **17**(9), 1688–1691 (2013)
13. B. David, S. Balu, A hybrid ica-svm approach to continuous phase modulation recognition. *IEEE Signal Process. Lett.* **16**(5), 402–405 (2009)
14. H. Sai, Y. Yuanyuan, W. Zhiqing, F. Zhiyong, P. Zhang, Low complexity automatic modulation classification based on order-statistics. *IEEE Trans. Wirel. Commun.* **16**, 400–411 (2017)
15. E.E. Azzouz, A.K. Nandi, Automatic identification of digital modulation types. *Signal Process* **47**(1), 55–69 (1995)
16. M. Stephane, Z. Sifen, Characterization of signals from multiscale edges. *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(7), 710–732 (1992)
17. C. Corinna, V. Vladimir, Support-vector network. *Mach. Learn.* **20**(3), 273–297 (1995)
18. B. Leo, Random forests. *IEEE Trans. Broadcast.* **45**(1), 5–32 (2001)
19. T. Cover, P. Hart, Nearest neighbor pattern classification. *IEEE Trans. Broadcast.* **13**(1), 21–27 (1967)
20. N. Friedman, D. Geiger, M. Goldszmidt, Bayesian network classifiers. *Mach. Learn.* **29**(2), 131–163 (1997)
21. L. Yann, B. Yoshua, H. Geoffrey, Deep learning. *Nature* **521**(7553), 436–444 (2015)
22. O. Tim, R. Tamoghna, T.C. Charles, Over-the-air deep learning based radio signal classification. *IEEE J. Sel. Top. Signal Process.* **12**(1), 168–179 (2018)
23. W. Nathan, O. Timothy, Deep architectures for modulation recognition. In: *IEEE International Symposium on Dynamic Spectrum Access Networks*, pp. 1–6 (2017)
24. Z. Zufan, L. Hao, W. Chun, G. Chengquan, X. Yong, Automatic modulation classification using cnn-lstm based dual-stream structure. *IEEE Trans. Veh. Technol.* **69**(11), 13521–13531 (2020)
25. W. Yu, L. Miao, Y. Jie, G. Guan, Data-driven deep learning for automatic modulation recognition in cognitive radios. *IEEE Trans. Veh. Technol.* **68**(4), 4074–4077 (2019)
26. H. Sai, J. Yizhou, G. Yue, F. Zhiyong, Z. Ping, Automatic modulation classification using contrastive fully convolutional network. *IEEE Wirel. Commun. Lett.* **8**(4), 1044–1047 (2019)
27. B. Ke, H. Yuan, J. Xiaojun, H. Jindong, Adversarial transfer learning for deep learning based automatic modulation classification. *IEEE Signal Process. Lett.* **27**, 880–884 (2020)
28. L. LiXin, H. Junsheng, C. Qianqian, Automatic modulation recognition: a few-shot learning method based on the capsule network. *IEEE Trans. Broadcast.* **99**, 1–1 (2020)
29. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. *Neural Inf. Process. Syst.* **27**, 2672–2680 (2014)
30. A. Martin, C. Soumith, L. Bottou, Wasserstein gan (2017)

31. G. Ishaan, A. Faruk, A. Martin, D. Vincent, C. Aaron, Improved training of Wasserstein GANs. *Neural Inf. Process. Syst.* **30**, 5767–5777 (2017)
32. F. Yanwei, H. Timothy, X. Tao, G. Shaogang, Transductive multi-view zero-shot learning. *IEEE Trans. Pattern Anal. Mach. Intell.* **37**, 2332–2345 (2015)
33. R.-P. Bernardino, T. Philip, An embarrassingly simple approach to zero-shot learning, vol. 37, pp. 2152–2161 (2015)
34. R. Scott, A. Zeynep, L. Honglak, S. Bernt, Learning deep representations of fine-grained visual descriptions, pp. 49–58 (2016)
35. M.A.V.D.A.C. Ishaan Gulrajani, Faruk Ahmed: Improved training of wasserstein gans (2017)
36. M. Laurens, H. Geoffrey, Visualizing data using t-sne. *J. Mach. Learn. Res.* **9**(86), 2579–2605 (2008)
37. D. Jia, D. Wei, S. Richard, L. Lijia, L. Kai, L. Feifei, Imagenet: a large-scale hierarchical image database, pp. 248–255 (2009)
38. T. Lin, M. Michael, S. Belongie, Microsoft coco: common objects in context, pp. 740–755 (2014)

Publisher's Note

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

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)
