Copyright © 2018 Tech Science Press CMC, vol.55, no.2, pp.243-254, 2018

Semi-Supervised Learning with Generative Adversarial Networks on Digital Signal Modulation Classification

Ya Tu[1], Yun Lin[1], Jin Wang[2, 3, *] and Jeong-Uk Kim[4]

Abstract: Deep Learning (DL) is such a powerful tool that we have seen tremendous success in areas such as Computer Vision, Speech Recognition, and Natural Language Processing. Since Automated Modulation Classification (AMC) is an important part in Cognitive Radio Networks, we try to explore its potential in solving signal modulation recognition problem. It cannot be overlooked that DL model is a complex model, thus making them prone to over-fitting. DL model requires many training data to combat with over-fitting, but adding high quality labels to training data manually is not always cheap and accessible, especially in real-time system, which may counter unprecedented data in dataset. Semi-supervised Learning is a way to exploit unlabeled data effectively to reduce over-fitting in DL. In this paper, we extend Generative Adversarial Networks (GANs) to the semi-supervised learning will show it is a method can be used to create a more data-efficient classifier.

Keywords: Deep Learning, automated modulation classification, semi-supervised learning, generative adversarial networks.

1 Introduction

Nowadays, Modulation classification is a major issue in many communications systems [Wu, Zapevalova, Chen et al. (2018)]. In Cognitive Radio (CR), Akyildiz [Akyildiz (2016)] state without an effective settled spectrum assignment policy, wireless network is more likely to be trapped in serious signal jamming [Jia, Liu, Gu et al. (2017); Zhao, Yu and Leung (2015); Zhao, Yu, Li et al. (2016); Zheng, Sangaiah and Wang (2018)]. Hence, people attach importance to signal detection and classification. AMC has been used for decades and it guarantees friendly signal can be securely transmitted and received [Ding, Wu, Yao et al. (2013); Ding, Wang, Wu et al. (2014); Yang, Ping and Sun (2016); Zhao, Yu, Sun et al. (2016)], while locating, identifying and blocking hostile signals [Zheng, Sangaiah and Wang (2016)]. Therefore, a smart way to classify the digital signal modula-

[1] College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China.

[2] School of Computer & Communication Engineering, Changsha University of Science & Technology, Changsha 410114, China.

[3] School of Information Engineering, Yangzhou University, Yangzhou 225009, China.

[4] Department of Energy Grid, Sangmyung University, Seoul, Korea.

[*] Corresponding author: Jin Wang. Email: jinwang@csust.edu.cn.

CMC. doi:10.3970/cmc.2018.01755 www.techscience.com/cmc

tion is a problem demanding prompt solution.

DL is an emerging field of Machine Learning (ML) method due to the contribution of new algorithm [Liu, Fu, He et al. (2017); Wu, Zapevalova, Chen et al. (2018); Yuan, Li, Wu et al. (2017)], mass data and well-performed hardware [Shi, Zheng, Zhou et al. (2017)]. Compared to traditional ML, Sutskever [Sutskever (2012)] proves DL can extract feature itself to adjust its parameter get a better end to end performance. Nowadays, DL has been applied into various areas, such as IoT [Zhang, Kou and Zhang (2017)], fractal generation [Liu, Pan and Fu (2017)]. The automatic feature extraction mechanism will greatly free people from the complex feature engineering. When it comes to DL classification, mass high-quality labeled training data is expected to push the decision boundaries toward the right direction [Li (2015); Yang, Ping and Sun (2016)]. Unfortunately, mass high-quality labeled training data is so expensive that we could not obtain it every time. Like in radar system, it is common to counter an unprecedented unlabeled data. The unlabeled data will be abandoned in supervised learning, but will not be wasted in semi-supervised learning.

Semi-supervised Learning is a technique in which both labeled and unlabeled data are used to train a classifier. The goal of this classifier is to combine a few labeled data and a much larger amount of unlabeled data to learn a decision function, which is able to get desirable outcome. In this paper, we will make full use of semi-supervised learning in digital signal modulation classification based on GANs.

The rest of this paper will be presented as follow: Section II offers a brief introduction to how to obtain the dataset in this paper. Section III provides an overview of semi-supervised ACGANs architecture. Section IV gives an intuitive understanding of how Generative Images Quality influence the usage of unlabeled data and Section V will give illustration to the trick we used in semi-supervised GANs and the result about performance comparison between semi-supervised GANs and baseline result. Section VI states the future work.

2 Contour stella image dataset

2.1 Constellation diagram

Nowadays, DL model mainly accepts three data format: Images, Sequence, and Text. We choose Images to be our data format and Constellation Diagram to present the statistic information of digital signal modulation. That means our dataset consists of images.

Constellation Diagram is a complex plane. By using the amplitude information from Inphase channel and Quadrature channel, Constellation Diagram will reveal modulated signal's amplitude and phase information. For example, QPSK can be depicted in Eq. 1

$$s(t_i) = e^{j2\pi f_c t_i + \pi \frac{2c_i+1}{4}} + n(t_i), c_i \in 0,1,2,3 \tag{1}$$

Where t_i is time node, $n(t_i)$ is noise, f_c is carrier wave's frequency and c_i is the signal phase.

Since QPSK signal has one amplitude and four different phases, so constellation diagram of QPSK at 4 dB will be shown as Fig. 1:

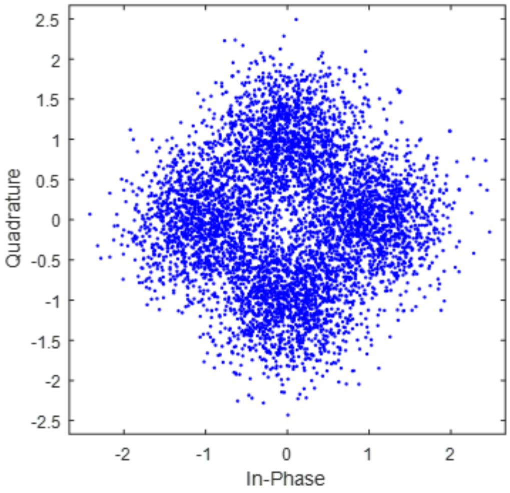

Figure 1: QPSK constellation diagram at SNR=4 dB

2.2 Contour stella image

In our last work [Tang, Tu, Zhang et al. (2018)], we utilize a trick in data preprocessing, which will significantly boost the performance of classifier. In Constellation Diagram, classifier will almost depend on the shape feature of Constellation Diagram to decide the type of modulated signal. We believe additional auxiliary information will help the classifier work well.

In Constellation Diagram, one dot means a sample point in modulated signal. Different area in Constellation Diagram has different point density. Depending on the dot density, we add auxiliary information, color, into Constellation Diagram.

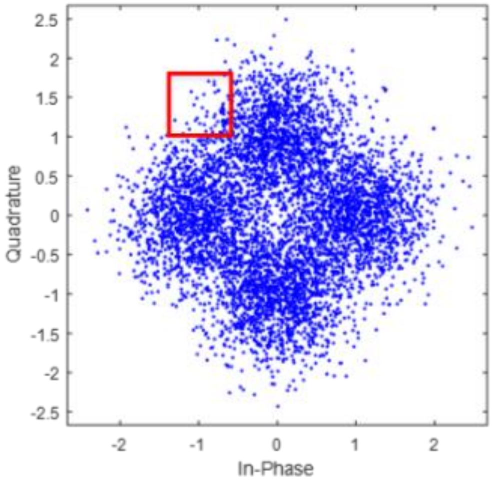

Figure 2: Convert QPSK constellation diagram at SNR=4 dB to contour stella image

As is depicted in Fig. 2, firstly, we choose a window function (the red square in Fig. 2), called Density Window Function, to slide on Constellation Diagram. Density Window Function will count how many dots in its field. After sliding on Constellation Diagram, we mark Relatively Dot Destiny value to different area in Constellation Diagram by using Eq. 2:

$$\rho(i,j)=\frac{\sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} dots(i,j)}{\sum_{x1=W_0}^{W_1}\sum_{y1=H_0}^{H_1}\sum_{i=x_1}^{x_2}\sum_{j=y_1}^{y_2} dots(i,j)} \tag{2}$$

Where x_1, y_1 stands for top-left corner of Destiny Windows Function currently coordinate, x_2, y_2 stands for bottom-right corner of Destiny Windows Function currently coordinate, W_0, H_0 stands for Constellation Diagram top-left corner coordinate, W_1, H_1stands for bottom-right corner coordinate of Constellation Diagram, $\rho(i,j)$ is Relatively Point Destiny, and $dots(i,j)$ means one sample point, whose coordinate is (i,j) in Constellation Diagram.

Then, we will map the area's Relatively Dot Destiny into a color bar based on its normalized value, where Yellow means higher density area, Green means middle density area and Blue means lower density area.

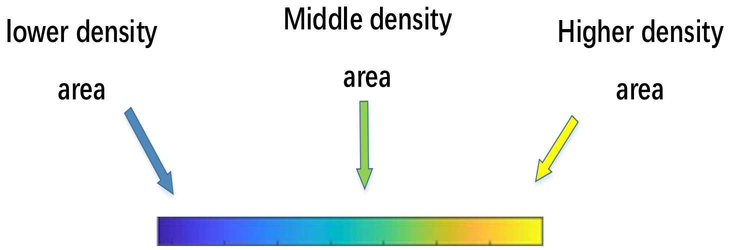

Figure 3: Relatively dot destiny colorbar

We consider 8 modulation categories in this paper, including BPSK, 4ASK, QPSK, OQPSK, 8PSK, 16QAM, 32QAM, and 64QAM. We give the 8 modulated signals at SNR=4 dB mentioned above Contour Stella Image in Fig. 4:

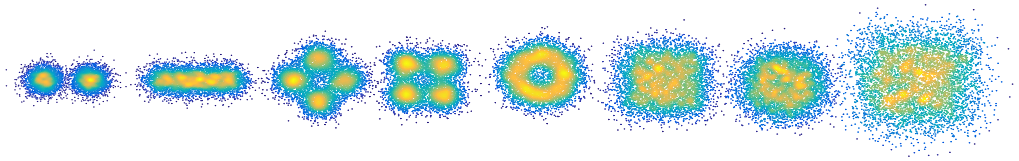

Figure 4: Contour stellar images of 8 signals at SNR=4 dB. The category of modulated signal in Fig. 4 from left to right is: BPSK, 4ASK, QPSK, OQPSK, 16QAM, 32QAM, 64QAM

In our work [Tang, Tu, Zhang et al. (2018)], we have proved this data preprocessing trick indeed helps classifier works well.

In this paper, we create Contour Stella Image dataset for semi-supervised learning. The dataset includes 8 categories modulated signal at 14 dB: BPSK, 4ASK, QPSK, OQPSK, 16QAM, 32QAM, 64QAM. Each modulated signal will be distributed 10000 labeled data for training data and 1000 labeled data for testing data, which means there will be 80000 labeled training dataset and 8000 labeled test dataset in total.

3 Semi-supervised learning based on ACGANs

Work on generating images with Generative Adversarial Networks (GANs) has shown promising results. GANs has two basic components: a generative net *G* and a discriminator *D*. *D* is trained to find out the images taken in whether it comes from *G*, while *G* is trained to let *D* make a mistake. When learning a generative model, we also consider solving a semi-supervised signal modulation classification task.

When it comes to traditional GANs, the discriminator *D* outputs a probability that the image comes from the real dataset. In semi-supervised learning task, we extend *D* output to $\mathrm{N}+1$. *N* units correspond to image class and 1 unit corresponds to the source where the image comes from.

In this paper, we choose ACGAN [Odena, Christopher and Jonathon (2016)] to finish our semi-supervised learning task and call it SSACGAN (Semi-Supervised ACGAN). ACGAN is proposed by Odena et al. [Odena, Christopher and Jonathon (2016)] and not very different from CGAN [Isola, Zhu and Zhou (2016)]. The generator *G* in ACGAN will use the concatenated information, corresponding class label *c* and noise *z*, as the input to generator. The discriminator *D* will give a prediction on the source of image and the class it should belongs to. In other word, a classifier is embedding in discriminator *D*. The objective for ACGAN is: Generator G generates specified labeled image to fool discriminator *D*, while discriminator *D* tries to give the correct label about the class and the source. ACGAN can be depicted in Fig. 5:

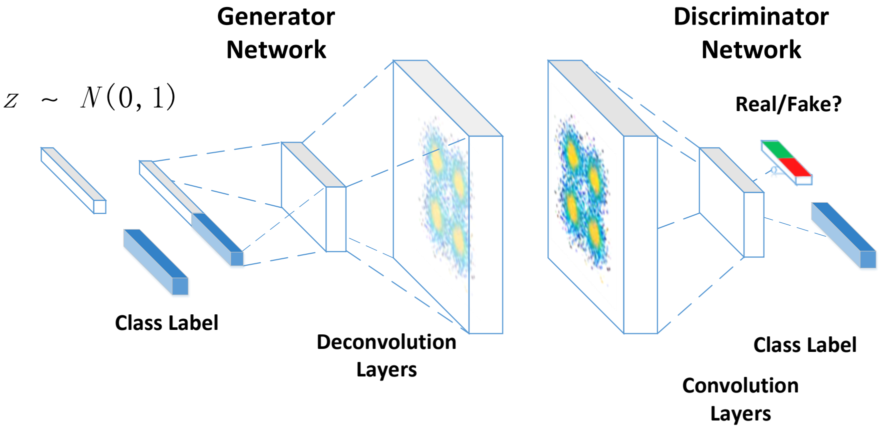

Figure 5: ACGAN model

In semi-supervised learning, ACGANs will learn how to handle three different sources of training data:

1. Real images with labels. Discriminator *D* will tune its parameter to tell this image correct label and the source of this data is from real data.

2. Real images without labels. Discriminator *D* will only try to predict this data is from real data, but we do not care about the class it should belong to.
3. Images from the generator. Discriminator *D* will learn how to classify them as fake data, but we do not care about the class it should belong to.

The combination of these different sources of data will help the classifier in discriminator *D* to figure out which feature is useful and which feature is useless.

From Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)], the object of training phase is to minimize Eq. 3:

$$L^f = L^f_{\sup} + L^f_{un\sup} \tag{3}$$

Where:

$$L^f_{\sup} = -E_{(x,y)\sim p_{d(x,y)}} \log(softmax(y = i \mid x, y \le k)) \tag{4}$$

$$L^f_{un\sup} = -E_{z\sim p_g(x)} \log(p_c(y = k+1|x)) - E_{x\sim p_d(x)} \log(1 - p_c(y = k+1|x)) \tag{5}$$

Where $p_c(y = k + 1|x)$ is the probability of x being a fake example and $1 - p_f(y = k + 1|x)$ is the probability of *x* being a real example which is assigned by discriminator.

From Eq. 3, Eq. 4 and Eq. 5, discriminator *D* plays a decisive role. A good discriminator D should handle both problem mentioned below:

1. Push the generator to produce realistic images. Discriminator *D* should be smart to tell real and fake samples.
2. Using the generator's images, labeled and unlabeled data, discriminator *D* is expected to classify the dataset.

4 The effect of generative images quality on the usage of unlabeled data

In this section, we attempt to give an intuitive understanding of how Generative Images Quality influence the usage of unlabeled data. From Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)], the result of *softmax(x)* function can be rewritten as Eq. 6:

$$\exp[l_j(x)] = c(x) softmax(y = j \mid x), \forall j \le k \tag{6}$$

Where $c(x)$ are some undetermined scaling function.

For $p_c(y = k + 1|x)$, we can also rewrite it as Eq. 7 :

$$\exp[l_{k+1}(x)] = c(x) p_c(y = k+1 \mid x) \tag{7}$$

As has been stated in Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)], $l_i(x)$ does not change the output of the softmax if it subtracts a general function $d(x)$ from each output logit:

$$l_j(x) \leftarrow l_j(x) - d(x), \forall j \tag{8}$$

Where $d(x)$ is a value function.

Suppose we set the (k +1)'th logit to 0 by subtracting $d(x)$. The output of discriminator D is Eq. 9:

$$D(x)=\frac{V(x)}{V(x)+1} \tag{9}$$

Where $V(x)=\sum_{i=1}^{k}\exp(l_i(x))$

From Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)], the unsupervised-learning loss for semi-supervised learning GANs, is:

$$L_{unsupervised}=-\{E_{x\sim p_d(x)}\log D(x)+E_{x_g\sim p_g}\log(1-D(G(z)))\} \tag{10}$$

Where $G(z)$ is the output of generator G, $D(G(z))$ is the output of the prediction from discrimination D.

Combined with Eq. 9, Eq. 5 can be rewritten as followed:

$$\begin{aligned}L_{unsupervied}&=E_{x_g\sim p_g}\log(1+\sum_{i=1}^{k}e^{l_i(x_g)})\\ &-E_{x\sim p_d}[\log\sum_{i=1}^{k}e^{l_i(x)}-\log(1+\sum_{i=1}^{k}e^{l_i(x)})]\end{aligned} \tag{11}$$

Taking Eq. 11 derivative with respect to discriminator D parameter θ, we will get the gradient for updating discriminator D:

$$\begin{aligned}&E_{x_g\sim p_g}\sum_{i=1}^{k}p_c(y=i\,|\,x_g)\nabla l_i(x_g)-\\ &E_{x\sim p_d}\sum_{i=1}^{k}softmax(y=i\,|\,x)p_c(y=k+1\,|\,x)\nabla l_i(x)\end{aligned} \tag{12}$$

Where $p_c(y=i|x_g)$ is equal to Eq. (9) and $p_c(y=k+1|x)$ is equal to $1-\mathrm{D(x)}$.

Minimizing $L_{unsupervised}$ will tune the parameters θ to decrease $l_i(x_g)$ and increase $l_i(x)$.

4.1 Bad image quality

In this case, generator *G* produces such rubbish images that discriminator D can easily find out the source where the images comes from. In that way, $p_c(y=k+1|x)\approx 0$. $p_c(y=k+1|x_g)$ is close to 1, which will lead $p_c(y=i|x_g)$ is almost equal to 0. Eq. 12 is almost equal to 0, which will prevent model from exploiting unlabeled data.

4.2 Good image quality

In this case, $p(x_g)$ is very close to $p(x)$, this will make $p_c(y=k+1|x_g)$ be close to 0.5, since discriminator D will randomly predict x_g source. Therefore, three data source will become almost two source at this time, Real images without labels has no difference to Images from the generator. The model will be more likely to over-fitting around the unlabeled examples, which will hurt the test performance.

4.3 *Moderated image quality*

In this case, the distance between x and x_g is sufficiently far apart. The model can exploit three data source to make a smooth decision function. What is more, $p_c(y = k + 1|x)$ will be less when it compares to Good Image Quality case. This will increase $p_c(y = k + 1|x_g)$ while decrease $p_c(y = i|x_g, y \neq k + 1)$ to model learn more lesson from supervised-learning part, which will boost test performance somehow.

5 Experimental results

To achieve a better classification result, we also apply some tricks in this model to improve discriminator *D* performance:

1. Feature Matching. Feature Matching is a GANs training technique propose by Ian Goodfollow et al. [Sailmans, Goodfellow, Zaremba et al. (2016)]. The concept of Feature Matching can be depicted as Eq. 13:

$$\left| f_{train} - f_{generate} \right| \tag{13}$$

Where f_{train} is the average value of features on the training data, $f_{generate}$ is the average value of features on the generated samples.

We extract the intermediate feature in discriminator *D* and tell generator to minimize the distance. As has been mentioned in Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)], feature matching will make model generate blurry and distorted samples, which is close to “Moderated Image Quality” mentioned above.

2. Adjusting Dropout Hyper Parameter. In semi-supervised learning, due to the limited label in training dataset, the classifier is prone to over-fitting. So we adjust dropout hyper parameter to combat with over-fitting.
3. Using LogSumExp (LSE) Function to avoid numerical problems. LSE can be related to Eq. 14:

$$LSE(x_1, x_2, ..., x_n) = \sum_{i=1}^{n} \log(x_i) \tag{14}$$

This trick will alleviate numerical problems when facing very extreme numerical.

4. Minibatch Discrimination. This method is proposed by Sailmans et al. [Sailmans, Goodfellow, Zaremba et al. (2016)] will greatly avoid model collapse in GANs so that we could get better generated images. We introduce this trick to obtain Good Image Quality mentioned above to conduct comparison experiment.

In semi-supervised leaning experiment, one key metric of interest is the number of labeled data, we will use tiny portion of labeled data. For those we want to treat as unlabeled data, we will ignore its supervised loss.

We select ACGAN’s ImageNet hyper parameter announced in Odena et al. [Odena, Olah and Shlens (2016)], modify the dropout portability to 0.3, and change generator G loss function according to trick 1. Baseline result play the role as an isolated classifier on restricted training sets. To train the baseline, we train ACGAN without updating generator *G* [Odena (2016)]. The classification accuracy comparison between baseline and semi-supervised learning ACGAN is related to how much we shrink the training set,

In this experiment, we shrink dataset to 500, 1000, 2000, and 5000 labeled data for per modulation category signal. That means we have 4000 labeled data, 8000 labeled data, 16000 labeled, 40000 labeled data for the overall dataset. For baseline result, we just train ACGAN discriminator *D* without updating generator *G*. After conducting semi-supervised learning based on ACGAN which has been applied Minibatch Discrimination (SSACGAN-MD) or Feature Matching (SSACGAN-FM) in model, we got the comparison result:

Table 1: Classification average accuracy between different method

Labeled Data	Baseline (%)	SSACGAN-MD (%)	SSACGAN-FM (%)
4000	64.89	66.12	70.12
8000	72.86	72.99	75.36
16000	88.45	87.31	88.42
40000	97.32	93.44	95.64

From Tab. 1, some conclusions can be drawn. (1) The amount of dataset plays an important in DL classification task. (2) More labeled data will improve classifier accuracy. (3) SSACGAN outperforms CNN which shares the same classifier architecture in tiny labeled dataset. In tiny labeled dataset, there exists three data source for SSGAN classifier's training. In larger labeled dataset, the unavoidable noise in generated images will prevent SSGAN classifier to learn the correct feature. (4) Semi-Supervised Learning GANs will perform better when it produce "Moderated Image Quality".

In this experiment, we also give the generated image of 40000 labeled situation after generator *G* loss converge. Generated image is depicted in Figs. 6 (a), 6(b):

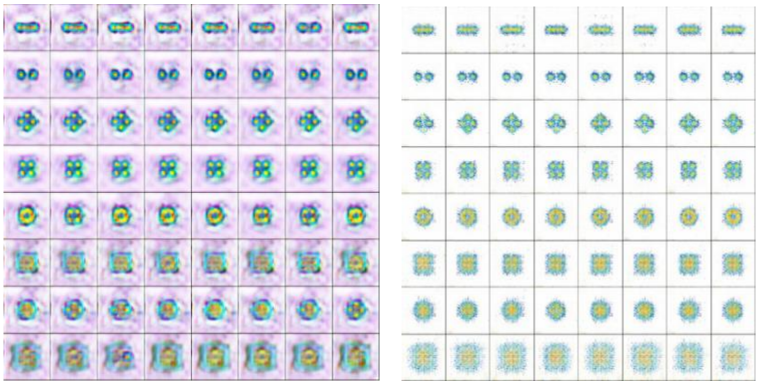

Figure 6: (a) Generated contour stella image dataset from SSACGAN-FM at 4 dB; (b) Generated contour stella image dataset from SSACGAN-MD at 4 dB

Seen from Figs. 6 (a), 6(b), due to the control of label, every row has the same modulation category signal. Compare to Generated Contour Stella Image dataset from

SSACGAN-FM the quality of generated images is not very good, feature-matching loss is responsible for this [Sailmans, Goodfellow, Zaremba et al. (2016)].

5 Conclusion and future work

In this paper, we discuss about how to implement semi-supervised learning with GANs. GANs obtains good classification accuracy for it exploits three data source. A good semi-supervised learning GANs result actually requires a "bad" generator G. The classify result deserves our utter attention, though we do not get "Good Image Quality". What's more, in tiny labeled dataset, we prefer to use semi-supervised learning, while we choose to use supervised learning in larger labeled dataset.

There are many factors still need improvement. We will explore the possible situation that the quality of the generated images and classification accuracy can be both improved. Secondly, the SNR of dataset is not low SNR, we will explore its performance in low SNR situation. Thirdly, this dataset is made up of simple feature images, we believe this model can work better with deeper networks. Last, we will explore more GANs' training tricks mentioned in in future model.

Acknowledgement: This work is supported by the National Natural Science Foundation of China (Nos. 61771154, 61603239, 61772454, 6171101570). Meantime, all the authors declare that there is no conflict of interests regarding the publication of this article. Prof. Jin Wang is the corresponding author. We gratefully thank of very useful discussions of reviewers.

References

Akyildiz, I. F.; Lee, W. Y.; Vuran, M. C.; Mohanty, S. (2016): Next generation/dynamic spectrum access/cognitive radio wireless networks: A survey. *Computer Networks*, vol. 50, no. 13, pp. 2127-2159.

Bin, T.; Ya, T.; Zhaoyue, Z.; Lin, Y. (2018): Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access*.

Dai, Z.; Yang, Z.; Yang, F.; Cohen, W. W.; Salakhutdinov, R. R. (2017): Good semi-supervised learning that requires a bad gan. *Advances in Neural Information Processing Systems*, pp. 6513-6523.

Ding, G.; Wang, J.; Wu, Q.; Yao, Y. D.; Song, F. et al. (2016): Cellular-base-station assisted device-to-device communications in TV white space. *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 1, pp. 107-121.

Ding, G.; Wu, Q.; Yao, Y. D.; Wang, J.; Chen, Y. (2013): Kernel-based learning for statistical signal processing in cognitive radio networks: Theoretical foundations, example applications, and future directions. *IEEE Signal Processing Magazine*, vol. 30, no. 4, pp. 126-136.

Ding, G.; Wang, J.; Wu, Q.; Zhang, L.; Zou, Y. et al. (2014): Robust spectrum sensing with crowd sensors. *IEEE Transactions on Communications*, vol. 62, no. 9, pp. 3129-3143.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D. et al. (2014):

Generative adversarial nets. *Advances in Neural Information Processing Systems*, pp. 2672-2680.

Isola, P.; Zhu, J. Y.; Zhou, T.; Efros, A. A. (2016): *Image-to-image translation with conditional adversarial networks.* http://arxiv.org/abs/1611.07004.

Jia, M.; Liu, X.; Gu, X.; Guo, Q. (2017): Joint cooperative spectrum sensing and channel selection optimization for satellite communication systems based on cognitive radio. *International Journal of Satellite Communications and Networking*, vol. 35, no. 2, pp. 139-150.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): Imagenet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, pp. 213-226.

Li, J. C. (2015): A new robust signal recognition approach based on holder cloud features under varying SNR environment. *KSII Transactions on Internet and Information Systems*, vol. 9, no. 12, pp. 4934-4949.

Liu, S.; Fu, W.; He, L.; Zhou, J.; Ma, M. (2017): Distribution of primary additional errors in fractal encoding method. *Multimedia Tools and Applications*, vol. 76, no. 4, pp. 5787-5802.

Liu, S.; Pan, Z.; Fu, W.; Cheng, X. (2017): Fractal generation method based on asymptote family of generalized mandelbrot set and its application. *Journal of Nonlinear Sciences and Applications*, vol. 10, no. 3, pp. 1148-1161.

Odena, A. (2016): *Semi-supervised learning with generative adversarial networks.* https://arxiv.org/abs/1606.01583.

Odena, A.; Olah, C.; Shlens, J. (2016): *Conditional image synthesis with auxiliary classifier GANs*. http://arxiv.org/abs/1610.09585.

Salimans, T.; Goodfellow, I.; Zaremba, W.; Cheung, V.; Radford, A. et al. (2016): Improved techniques for training gans. *Advances in Neural Information Processing Systems*, pp. 2234-2242.

Shi, X.; Zheng, Z.; Zhou, Y.; Jin, H.; He, L. et al. (2017): Graph processing on GPUs: A survey. *ACM Computing Surveys*, vol. 50, no. 6, pp. 1-35.

Wu, C.; Zapevalova, E.; Chen, Y.; Li, F. (2018): Time optimization of multiple knowledge transfers in the big data environment. *Computers, Materials & Continua*, vol. 54, no. 3, pp. 269-285.

Yang, Z.; Ping, S.; Sun, H. (2016): CRB-RPL: A receiver-based routing protocol for communications in cognitive radio enabled smart grid. *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5985-5994.

Yuan, C.; Li, X.; Wu, Q.; Li, J.; Sun, X. (2017): Fingerprint liveness detection from different fingerprint materials using convolutional neural network and principal component analysis. *Computers, Materials & Continua*, vol. 53, no. 3, pp. 357-371.

Zhang, G.; Kou, L.; Zhang, L.; Liu, C.; Da, Q. et al. (2017): A new digital watermarking method for data integrity protection in the perception layer of IoT. *Security and Communication Networks.*

Zhao, N.; Yu, F.; Li, M.; Yan, Q.; Leung, V. C. (2016): Physical layer security issues in interference-alignment-based wireless networks. *IEEE Communications Magzine*, vol. 54, no. 8, pp. 162-168.

Zhao, N.; Yu, F. R.; Leung, V. C. (2015): Opportunistic communications in interference alignment networks with wireless power transfer. *IEEE Wireless Communications*, vol. 22, no. 1, pp. 88-95.

Zhao, N.; Yu, F.; Sun, H.; Li, M. (2016): Adaptive power allocation schemes for spectrum sharing in interference-alignment-based cognitive radio networks. *IEEE Transactions on Vehicular Technology*, vol. 65, no. 5, pp. 3700-3714.

Zheng, Z.; Sangaiah, A. K.; Wang, T. (2018): Adaptive communication protocols in flying ad-hoc network. *IEEE Communication Magazine*, vol. 56, no. 1, pp. 136-142.