Data Augmentation with Conditional GAN for Automatic Modulation Classification

Mansi Patel
Department of Computer Science, California State University, Sacramento
Sacramento, CA, USA
mansipatel@csus.edu

Xuyu Wang
Department of Computer Science, California State University, Sacramento
Sacramento, CA, USA
xuyu.wang@csus.edu

Shiwen Mao
Department of Electrical and Computer Engineering, Auburn University
Auburn, AL, USA
smao@ieee.org

ABSTRACT

Deep learning has great potential for automatic modulation classification (AMC). However, its performance largely hinges upon the availability of sufficient high-quality labeled data. In this paper, we propose data augmentation with conditional generative adversarial network (CGAN) for convolutional neural network (CNN) based AMC, which provides an effective solution to the limited data problem. We present the design of the proposed CGAN based data augmentation method, and validate its performance with a public dataset. The experiment results show that CNN-based modulation classification can greatly benefit from the proposed data augmentation approach with greatly improved accuracy.

CCS CONCEPTS

• **Machine Learning** → **Adversarial Learning**; • **Neural Networks** → *Generative Adversarial Networks*; • **Automatic Signal Detection** → Wireless Networks.

KEYWORDS

Automatic modulation classification (AMC), cognitive radio (CR), deep learning, convolutional neural networks (CNN), conditional generative adversarial network (CGAN)

ACM Reference Format:
Mansi Patel, Xuyu Wang, and Shiwen Mao. 2020. Data Augmentation with Conditional GAN for Automatic Modulation Classification. In *ACM Workshop on Wireless Security and Machine Learning (WiseML'20), July 13, 2020, Linz (Virtual Event), Austria.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3395352.3402622

1 INTRODUCTION

In wireless communications, the scarce, depleting spectrum resource and on the other hand, the inefficient use of allocated spectrum have driven the vibrant research in Cognitive Radio (CR) [1]. Automatic modulation classification (AMC) is an essential component of CR to detect the nearby emitters to avoid radio inference and to improve spectrum efficiency [2]. AMC aims to classify the modulation types of received signals without a priori information of the signal and channel, with great applications for spectrum sensing and access, spectrum anomaly detection, classification security, and transmitter identification [3–5].

Traditional modulation recognition schemes can be classified into likelihood-based [6] and feature-based [7] categories. likelihood-based methods use Bayesian estimation for modulation classification assuming prior information such as channel and noise models. They usually have high computational complexity and are not suited for highly dynamic environments. Feature-based methods use handcrafted features to classify modulations. However, they largely depend on reliable features and manual selection. Recently, deep learning models have been leveraged for AMC without assuming prior information such as channel models [8, 9]. For example, a convolutional neural network (CNN) is used to classify 11 different modulations in the RadioML2016.10A dataset, where up to 70% cassification accuracy is achieved [2]. To further improve the accuracy, other deep learning models have also been proposed, such as recurrent neural networks (RNN) [10] and fusion methods [11].

Although deep learning-based methods can achieve satisfactory modulation classification accuracy, a massive amount of training samples are required and their performance hinges upon the quality of the samples [12]. However, it is costly and challenging to obtain labeled training samples, which greatly limits the application of deep learning for AMC and other wireless communications and networking tasks. To this end, we propose *data augmentation* to address the above problem. Currently, although sample augmentation methods, such as rotation, flip, and Gaussian noise have been used for modulation classification, only a small improvement of 2.5% has been achieved [13]. Some researchers have applied generative adversarial networks (GAN) to generate high-quality image dataset [14] and to augment training data for a wireless jammer [15]. GAN has also used for spectrum sensing [16], spectrum generation [17, 18], and wireless signal spoofing [19]. Moreover, auxiliary classifier GAN (ACGAN) is employed for modulation recognition with a simulated and ideal dataset that does not consider channel and hardware effects [20]. As a result, the accuracy improvement is limited for the ideal dataset.

In this paper, we propose to utilize conditional GAN (CGAN) [21] for data augmentation that takes into account the real features of wireless hardware, such as I/Q imbalance, and the impact of wireless channel, which are considerably more challenging than the previous ideal case. CGAN includes a generator and a discriminator that are conditioned on auxiliary information such as class labels, making them highly useful for synthesizing labeled data. The main idea

WiseML'20, July 13, 2020, Linz (Virtual Event), Austria

ACM ISBN 978-1-4503-8007-2/20/07…$15.00
https://doi.org/10.1145/3395352.3402622

is to exploit the superior learning power of CGAN to synthesize high quality, labeled wireless modulation data from a small set of available real data. The augmented dataset will greatly benefit CNN-based AMC to achieve greatly improved classification accuracy.

The main contributions of this paper are summarized as follows.

- To the best of our knowledge, this is the first work to utilize CGAN for data augmentation, using the CNN-based AMC problem as an example. The proposed data augmentation technique is quite general, and has a great potential to benefit many deep learning based wireless communications and networking studies.
- We discuss the system model, including the signal model, modulation dataset, and CNN-based modulation classification. We also present the detailed design of the proposed CGAN-based data augmentation approach.
- Using a public dataset, our experimental study validates the efficacy of the proposed method on synthesizing high-quality labeled wireless modulation data, which greatly improve the modulation classification accuracy.

The remainder of this paper is organized as follows. The system model is discussed in Section 2. We present the CGAN-based data augmentation in Section 3 and validate its performance in Section 4. Section 5 summarizes this paper.

2 SYSTEM MODEL

2.1 Wireless Signal Model

In wireless spectrum sensing and access, AMC is an important task, which is indispensable for detecting wireless communication types such as radio, radar users, and voice radios. It can also be leveraged to detect nearby emitters, thus avoiding potential radio interference. For modulation recognition, the received wireless spectrum signal $r(t)$ is usually given by

$$r(t) = h(t) * s(t) + n(t), \tag{1}$$

where $s(t)$ is the transmitted signal, $n(t)$ is the additive white Gaussian noise (AWGN), and $h(t)$ is channel impulse response (CIR). The in-phase and quadrature (I/Q) components of the wireless signal can be obtained by sampling the received complex signal $r(t)$, which will then be used for signal modulation recognition.

2.2 Modulation Dataset

In real wireless environments, the channel model captures the transmission impairments a wireless signal experiences, such as multipath, fading, center frequency offset (CFO), and sampling clock offset (SCO). As a result, the received I/Q data will distribute differently from that of the ideally modulated signal.

In this paper, we use a signal modulation dataset called RadioML2016.10A obtained by GNU Radio [2]. This dataset considers many radio channel effects, which is close to real wireless signal data. In particular, the RadioML2016.10A dataset also contains synthetic distributions with 11 different modulations, including 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM. Moreover, there are 220,000 signal samples for the 11 modulations, each of which has 20,000 signal samples. Moreover, each modulation is sampled at 20 different SNR values, ranging from -20dB to 18dB, with 1,000 samples for each SNR level. Each radio signal sample consists of 128 consecutive I/Q data units. By observing the samples, it is evident that in the high-SNR regime, the sample distribution of every modulation technique is comparatively more distinct than that in the low-SNR regime.

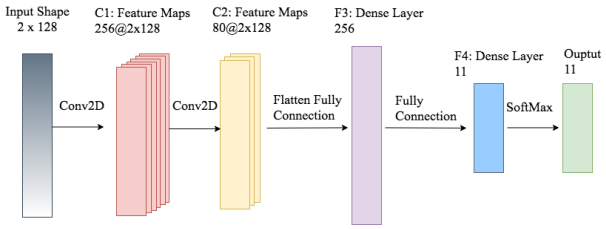


Figure 1: The CNN model used as benchmark in this paper.

2.3 CNN-based Modulation Classification

Modulation classification problems can be considered as a mapping from input signal samples (i.e., the I/Q data) to different modulation categories. The estimated modulation category can be obtained by choosing the output with the largest probability. Modulation classification can be implemented with different deep learning methods.

In this paper, we consider the CNN model as the benchmark for modulation classification proposed by O'Shea et al. in [2]. Fig. 1 illustrates the CNN architecture, which is a 4-layer network with two convolutional layers and two dense layers. The size of the input shape is 2×128, and the size of the output is 11 (corresponding to the 11 modulation techniques in the RadioML2016.10A dataset). In addition, each hidden layer incorporates dropout with probability 0.5. The rectified linear unit (ReLU) activation function is employed for activation. The CNN model is trained by using the Adam optimizer with a categorical cross entropy loss function. Although the CNN-based modulation classification can obtain satisfactory results, its performance is limited by the small dataset (i.e., only 1,000 signal samples in each category) and fixed data features. To improve the classification accuracy, we propose data augmentation using CGAN for modulation classification in this paper.

3 DATA AUGMENTATION USING CGAN

GAN is a generative machine learning model, which aims to generate samples that are indistinguishable from the real data [14]. The GAN framework includes two neural network models: a generative model G that is trained to produce new samples, and a discriminative model D that estimates the probability that a sample is from training data rather than G. The generator usually uses random noise as input to generate samples and the discriminator aims to distinguish generated samples from training samples. These two modals compete with each other, so that the generator produced samples become more and more indistinguishable from training samples and the discriminator identifies the generated samples more and more accurately. The discriminator D is trained to maximize the probability of assigning correct labels to the two types of samples; the generator G is trained to minimize $\log(1 - D(G(z)))$. The GAN framework can be modeled as a minimax two-player

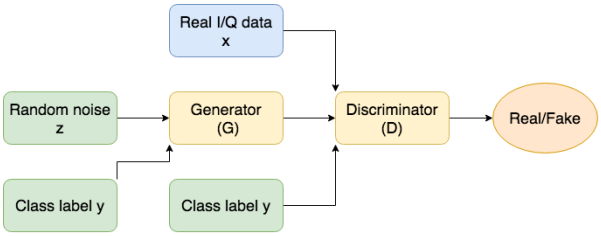


Figure 2: The CGAN architecture.

game with value function $V(G, D)$, given by

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \propto p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \propto p_z(z)} [\log(1 - D(G(z)))], \quad (2)$$

where $p_z(z)$ is a prior on the input noise. For any pair of functions G and D, there is a unique solution, where function G can recover the training data distribution and function D is equal to 0.5.

The original GAN model is unsupervised learning, and thus cannot generate labeled data. In other words, GAN does not have control over the models of data to be generated, which can only learn a mapping from random noise z to a target modulation data x. However, many spectrum sensing tasks, such as AMC, only work with labeled data. In this paper, we propose to use the CGAN approach to effectively solve this problem, which involves the generation of data conditioned on a class label, thus allowing targeted generation of data of a given type.

CGAN is a machine learning framework where both the generator and discriminator are conditioned on auxiliary information such as class labels y that act as an extension to the latent space z to generate and discriminate synthesized data [21]. Consequently, CGAN can learn a mapping from a random noise vector z to the output modulation data x conditioned on a class label y. CGAN can also be modeled as a minimax two-player game, where the value function is given by,

$$\min_G \max_D V(D,G) = \mathbb{E}_{x \propto p_{data}(x)} [\log D(x|y)] + \mathbb{E}_{z \propto p_z(z)} [\log(1 - D(G(z|y)))], \quad (3)$$

where $D(x|y)$ and $G(z|y)$ are the discriminator and generator functions for given label y, respectively. Fig. 2 presents the architecture of CGAN, which adds extra labels to the generator and discriminator for training the networks.

The detailed procedure of CGAN is as follows. The discriminator $D(x|y)$ is a classifier that determines whether the given modulation data is real and fake, which is usually represented by a value of 0 or 1, respectively. It also uses one-hot vector of the label to condition the discriminator output. In addition, adding class labels y can control the output and guide the generator $G(z|y)$ to figure out what to generate. Then, the generator $G(z|y)$ takes a randomly generated noise vector as input data and feedback from the discriminator $D(x|y)$, to generate new modulation data that are as close to real modulation data as possible. These two models compete with each other, each becoming stronger through the process. The generator $G(z|y)$ keeps on creating new modulation data and refining its process until the discriminator $D(x|y)$ can no longer tell the difference between a generated data and the real training data.

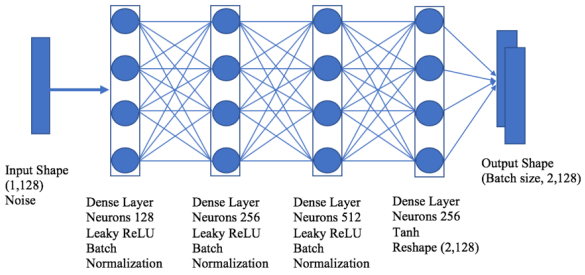


Figure 3: The CGAN generator model.

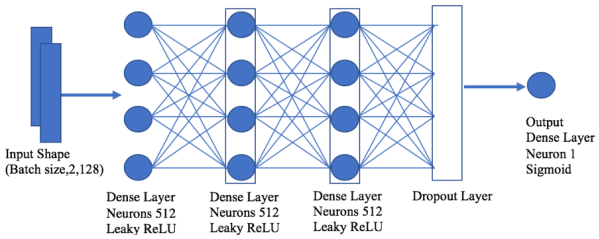


Figure 4: The CGAN discriminator model.

Fig. 3 and Fig. 4 plot the CGAN generator and discriminator networks, respectively. The generator network consists of four dense layers with 128, 256, 512, and 256 neurons, respectively. LeakyReLU is used as the activation function with a 0.05 learning rate, which is followed by batch normalization in each dense layer, as shown in Fig. 3. The discriminator network in Fig. 4 has three dense layers, each with 512 neurons. The last layer has only one node to determine if the modulation data is real and fake. The activation function is Sigmoid in the last layer. LeakyReLU is also used as activation function with a learning rate of 0.05. The dropout layer has dropout rate of 40%. In addition, both networks are followed by an embedded layer with class label. The RadioML2016.10A dataset has 11 classes, with class labels from 0 to 10. Thus, CGAN can be used for data augmentation for generating high-quality and diversity modulation data with class labels, which will be useful for improving CNN-based modulation classification accuracy.

4 EXPERIMENTAL STUDY

4.1 Experiment Configuration

We use the RadioML2016.10A dataset with 11 different modulations, including 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM. For each modulation there are 20 different SNR levels ranging from -20dB to 18dB. Timothy J O'Shea et al. [2] applied CNN for AMC with this dataset. Our approach is to augment the RadioML2016.10A dataset by generating more synthesized data using the proposed CGAN model and use the same CNN model to evaluate the improvement in modulation classification. The training and testing data are divided by 80%:20%.

The number of samples for each of modulations for a given SNR value is 1000 in the original dataset, which have different distributions. CGAN is applied with a batch size of 128 to generate synthesized modulation data with a similar distribution. Note that it requires multiple runs to achieve the desired results.

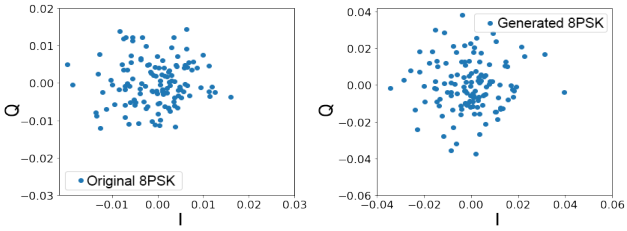


Figure 5: 8PSK (SNR=16dB): (left) original data; (right) synthesized data.

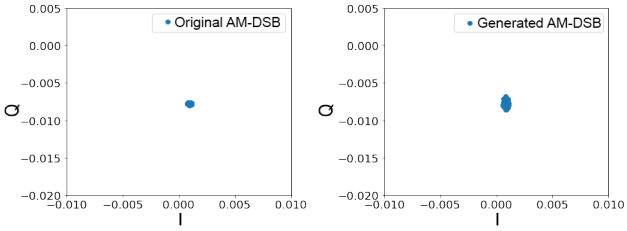


Figure 6: AM-DSB (SNR=16dB): (left) original data; (right) synthesized data.

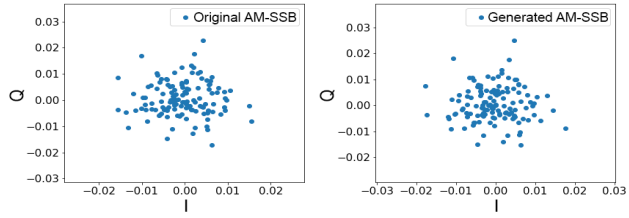


Figure 7: AM-SSB (SNR=16dB): (left) original data; (right) synthesized data.

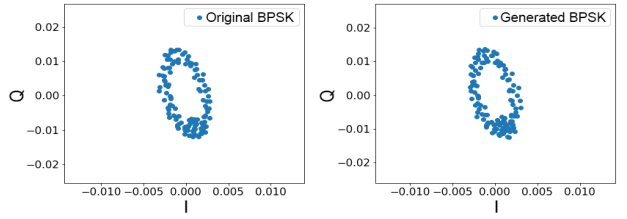


Figure 8: BPSK (SNR=16dB): (left) original data; (right) synthesized data.

We utilize the TensorFlow framework, Keras, and Scikit-Learn libraries for training the CGAN and CNN models, and use Google Colab as a free cloud service to train these models with graphics processing unit (GPU). To evaluate the classification accuracy, we leverage the $F1$ score as a function of the counts of true positives t_p, false positives f_p, and false negatives f_n, which is defined as

$$F1 = \frac{2t_p}{2t_p + f_p + f_n}. \tag{4}$$

In the reminder of this section, we will present and discuss the experimental results with the proposed approach. The experimental results are obtained for different numbers of synthesized data samples, i.e., 1000, 2000, 3000, 4000, and 5000 for each modulation at each SNR level. Hence, for each SNR value, 11000, 22000, 33000, 44000, and 55000 data samples are added by data augmentation.

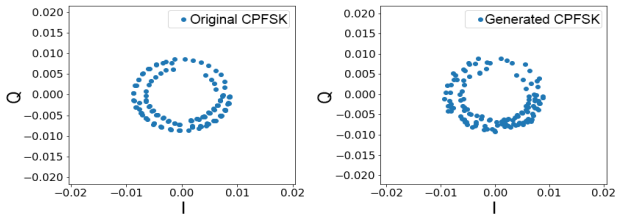


Figure 9: CPFSK (SNR=16dB): (left) original data; (right) synthesized data.

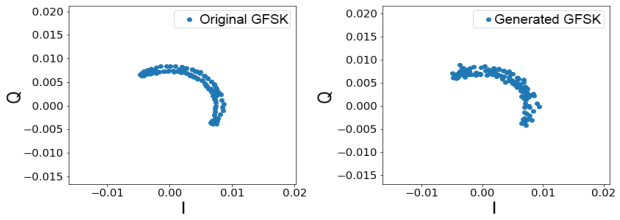


Figure 10: GFSK (SNR=16dB): (left) original data; (right) synthesized data.

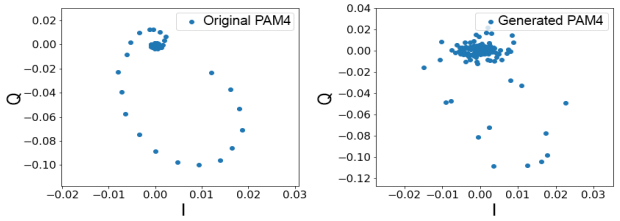


Figure 11: PAM4 (SNR=16dB): (left) original data; (right) synthesized data.

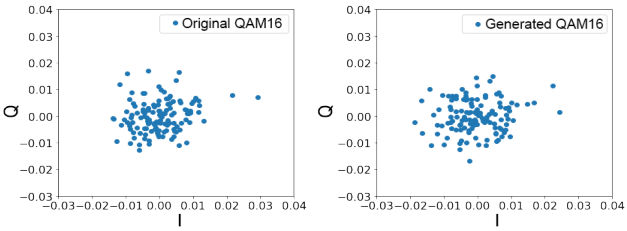


Figure 12: QAM16 (SNR=16dB): (left) original data; (right) synthesized data.

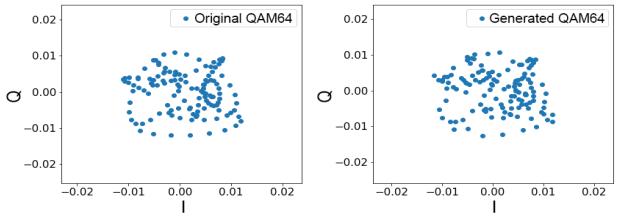


Figure 13: QAM64 (SNR=16dB): (left) original data; (right) synthesized data.

4.2 Performance Comparison

In Fig. 5 to Fig. 15, we plot the modulation I/Q data from the original dataset and the CGAN synthesized data side-by-side for the 11 modulations when SNR=16dB. It is easy to notice that the synthesized modulation data is very similar to the original modulation data with only small differences. More important, we can see that for complex modulations such as QAM64, the proposed CGAN method can achieve a better synthesis performance.

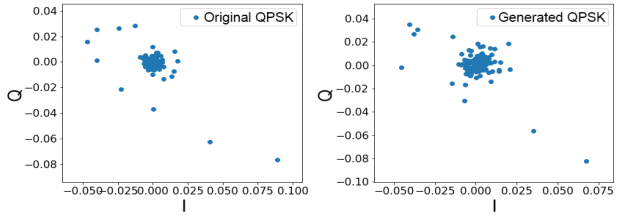


Figure 14: QPSK (SNR=16dB): (left) original data; (right) synthesized data.

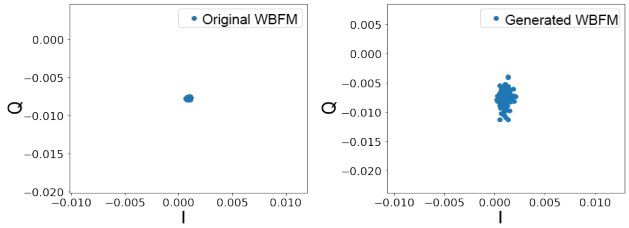
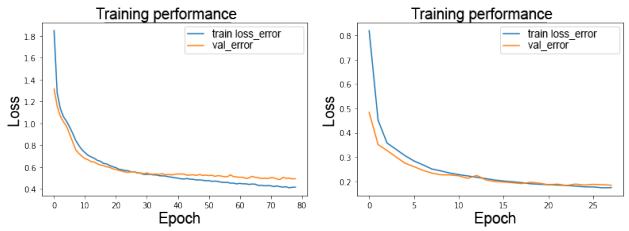


Figure 15: WBFM (SNR=16dB): (left) original data; (right) synthesized data.



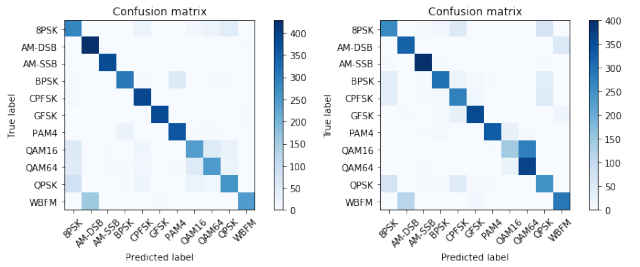
(a) 1000 synthesized samples. **(b) 5000 synthesized samples.**

Figure 16: Training performance when SNR=16 dB with (a) 1000 and (b) 5000 synthesized modulation data samples.

Fig. 16 presents the training performance of CNN-based AMC when SNR=16dB by adding 1000 and 5000 synthesized samples. We can see that the training loss values for the two cases are 1.8 and 0.8 at the 1st epoch, respectively. Furthermore, the training loss curves converge in 30 epochs and 10 epochs to values 0.25 and 0.6, respectively. These results clearly demonstrate that the CNN-based training can greatly benefit from the CGAN augmented data with fast convergence and a smaller training loss.

Fig. 17 shows a comparison of confusion matrices of different SNR values with 1000 augmented data. We can see that the classification accuracy when SNR=10dB is slightly better than that when SNR=-4dB. We also find the synthesized data are more helpful to improve the accuracy for lower SNR values. Moreover, the confusion matrix for SNR=-4dB exhibits a major confusion between the QAM64 and QAM16 modulations and a minor confusion by miss classifying data of the 8PSK, CPFSK, and QPSK modulations.

Fig. 18 shows the confusion matrices for modulation classification when SNR=16dB with 1000, 2000, 3000, 4000, and 5000 synthesized data for each modulation, as well as the original data without CGAN. An accurate classification performance throughout all the classes can be seen in these plots, indicated by the clear diagonal blocks with only very few miss classified data (e.g., see Figs. 18b and 18c). Fig. 18a has a darker diagonal, meaning it has better classification results. Fig. 18f shows the classification performance



(a) SNR=10dB **(b) SNR=-4dB**

Figure 17: Comparison of the confusion matrices for (a) SNR=10dB and (b) SNR=-4dB with 1000 synthesized samples.

without CGAN. We can see a worse performance as indicated by the scattered confusion matrix. Also, it seems to be confused in the classification of 8PSK, QAM16, QAM64, QPSK, and WBFM modulations. Moreover, QAM64 is miss-classified with different modulations such as QAM16, CPFSK, and QPSK.

Fig. 19 shows a comparison of the F1 scores of the original data without CGAN and augmented data using CGAN at different SNR levels. It can be seen that the proposed approach outperforms the CNN-only approach with the original data for all the SNR levels. We also find that adding more synthesized data always achieves a higher F1 value (i.e., a better performance). When augmented with 5000 synthesized samples, the F1 curve remains close to 0.93 from SNR=0dB to SNR=18dB. Compared with the case without data augmentation, the proposed approach achieves an approximately 25% gain in F1. In the low SNR regime when SNR is below -16dB, an approximately 16% gain in F1 is achieved. This experiment validates that the proposed CGAN based data augmentation can effectively improve the accuracy of CNN-based modulation classification.

5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a CGAN-based data augmentation approach for CNN-based AMC. The idea was to leverage CGAN to generate high-quality, labeled data with a small amount of seed data, thus overcoming the high cost and challenge associated with obtaining wireless datasets. Through experiments with a public dataset, we showed that CNN-based modulation classification could greatly benefit from the proposed data augmentation approach. For future work, we will investigate other deep learning models such as Long Short-term Memory (LSTM) and Bidirectional LSTM for AMC with CGAN-based data augmentation.

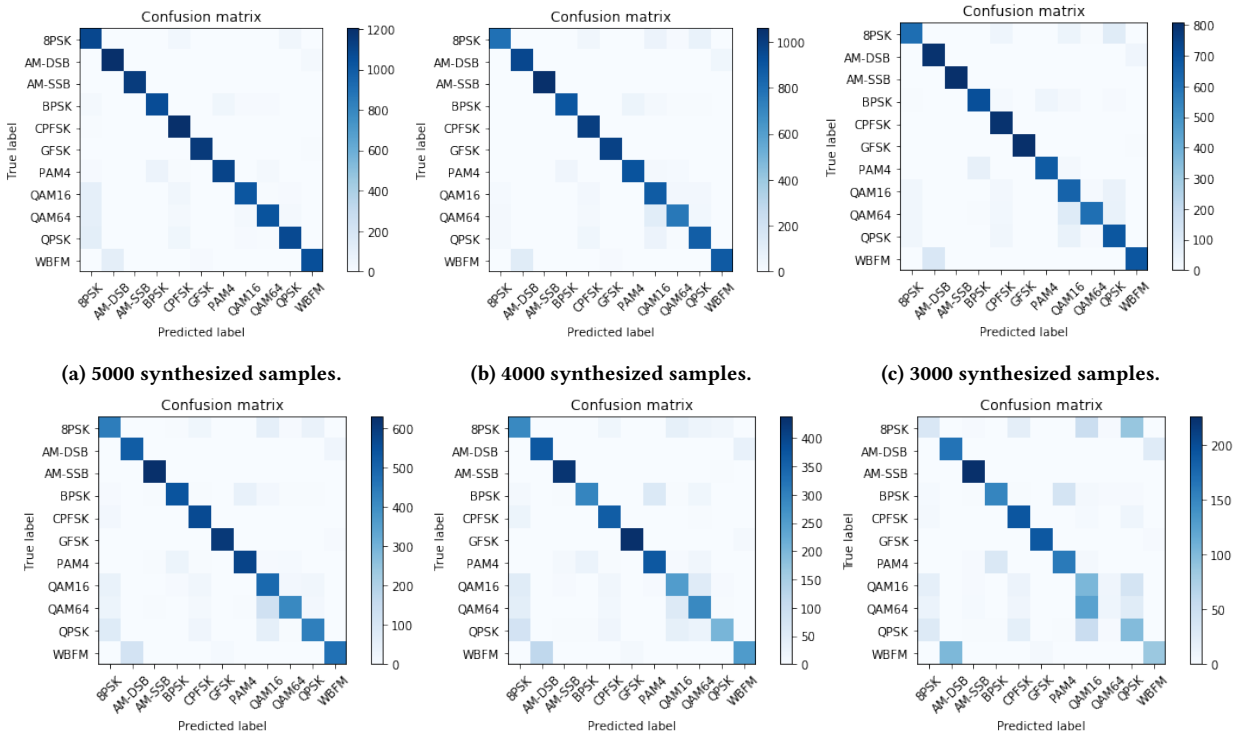
ACKNOWLEDGMENTS

This work is supported in part by the NSF under Grant ECCS-1923717 and the Wireless Engineering Research and Education Center at Auburn University, Auburn, AL, USA.

REFERENCES

[1] Muhammad Amjad, Mubashir Husain Rehmani, and Shiwen Mao. 2018. Wireless multimedia cognitive radio networks: A comprehensive survey. *IEEE Communications Surveys and Tutorials* 20, 2 (Second Quarter 2018), 1056–1103.

[2] Timothy J O'Shea, Johnathan Corgan, and T Charles Clancy. 2016. Convolutional radio modulation recognition networks. In *Proc. 2016 Int. Conf. Engineering Appl. Neural Netw.* Springer, Aberdeen, Scotland, 213–226.



(a) 5000 synthesized samples. **(b) 4000 synthesized samples.** **(c) 3000 synthesized samples.**

(d) 2000 synthesized samples. **(e) 1000 synthesized samples.** **(f) Without CGAN.**

Figure 18: Confusion matrices for modulation classification when SNR=16dB with different amount of synthesized data.

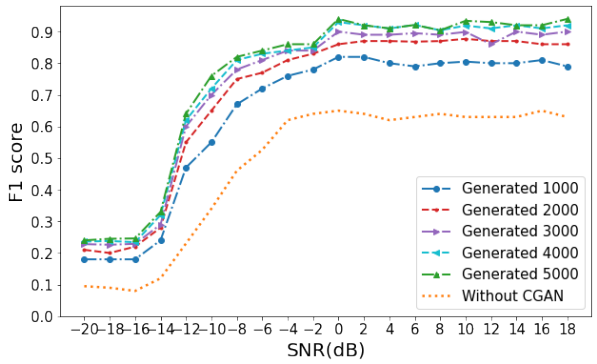


Figure 19: Classification accuracy with different amount of augmented data.

[3] Yun Lin, Jicheng Jia, Sen Wang, Bin Ge, and Shiwen Mao. 2020. Wireless device identification based on radio frequency fingerprint features. In *Proc. IEEE ICC 2019.* IEEE, Dublin, Ireland, 1–6.

[4] Yun Lin, Haojun Zhao, Ya Tu, Shiwen Mao, and Zheng Dou. 2020. Threats of adversarial attacks in DNN-based modulation recognition. In *Proc. IEEE INFOCOM 2019.* IEEE, Toronto, Canada, 1–10.

[5] Kemal Davaslioglu and Yalin E Sagduyu. 2019. Trojan attacks on wireless signal classification with adversarial machine learning. In *Proc. DySPAN 2019.* IEEE, Newark, NJ, 1–6.

[6] Jefferson L Xu, Wei Su, and Mengchu Zhou. 2010. Likelihood-ratio approaches to automatic modulation classification. *IEEE Trans. Syst., Man, Cybern. Syst., Part C (Appl. Rev.)* 41, 4 (2010), 455–469.

[7] Domenico Grimaldi, Sergio Rapuano, and Luca De Vito. 2007. An automatic digital modulation classifier for measurement on telecommunication networks. *IEEE Trans. Instrum. Meas.* 56, 5 (2007), 1711–1720.

[8] Gihan J Mendis, Jin Wei, and Arjuna Madanayake. 2016. Deep learning-based automated modulation classification for cognitive radio. In *Proc. IEEE ICCS 2016.* IEEE, Shenzhen, China, 1–6.

[9] Yi Shi et al. 2019. Deep learning for RF signal classification in unknown and dynamic spectrum environments. In *Proc. IEEE DySPAN 2019.* IEEE, Newark, NJ,

[10] Guanhong Tao et al. 2019. Sequential convolutional recurrent neural networks for fast automatic modulation classification. https://arxiv.org/abs/1909.03050

[11] Shilian Zheng, Peihan Qi, Shichuan Chen, and Xiaoniu Yang. 2019. Fusion methods for CNN-based automatic modulation classification. *IEEE Access J.* 7 (2019), 66496–66504.

[12] Yaohua Sun, Mugen Peng, Yangcheng Zhou, Yuzhe Huang, and Shiwen Mao. 2019. Application of machine learning in wireless networks: Key technologies and open issues. *IEEE Communications Surveys and Tutorials* 21, 4 (Fourth Quarter 2019), 3072–3108.

[13] Liang Huang, Weijian Pan, You Zhang, LiPing Qian, Nan Gao, and Yuan Wu. 2019. Data augmentation for deep learning-based radio modulation classification. *IEEE Access J.* 8 (Dec. 2019), 1498–1506.

[14] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Proc. NIPS 2014.* MIT Press, Montréal, Canada, 2672–2680.

[15] Tugba Erpek, Yalin E. Sagduyu, and Yi Shi. 2019. Deep learning for launching and mitigating wireless jamming attacks. *IEEE Trans. Cognitive Commun. Netw.* 5, 1 (Mar. 2019), 2–14.

[16] Kemal Davaslioglu and Yalin E Sagduyu. 2018. Generative adversarial learning for spectrum sensing. In *Proc. IEEE ICC 2018.* IEEE, Kansas City, MO, 1–6.

[17] Francesco Restuccia, Salvatore D'Oro, Amani Al-Shawabka, Bruno Costa Rendon, Kaushik Chowdhury, Stratis Ioannidis, and Tommaso Melodia. 2020. Hacking the waveform: Generalized wireless adversarial deep learning. https://arxiv.org/abs/2005.02270

[18] Tamoghna Roy, Tim O'Shea, and Nathan West. 2019. Generative adversarial radio spectrum networks. In *Proc. 2019 ACM Workshop on Wireless Security and Machine Learning.* ACM, Miami, FL, 12–15.

[19] Yi Shi, Kemal Davaslioglu, and Yalin E Sagduyu. 2019. Generative adversarial network for wireless signal spoofing. In *Proc. 2019 ACM Workshop on Wireless Security and Machine Learning.* ACM, Miami, FL, 55–60.

[20] Bin Tang, Ya Tu, Zhaoyue Zhang, and Yun Lin. 2018. Digital signal modulation classification with data augmentation using generative adversarial nets in cognitive radio networks. *IEEE Access J.* 6 (2018), 15713–15722.

[21] Mehdi Mirza and Simon Osindero. 2014. Conditional generative adversarial nets. https://arxiv.org/abs/1411.1784