Hindawi
Security and Communication Networks
Volume 2021, Article ID 8921485, 10 pages
https://doi.org/10.1155/2021/8921485

Research Article

A Lightweight Modulation Classification Network Resisting White Box Gradient Attacks

Sicheng Zhang, Yun Lin, Zhida Bao, and Jiangzhi Fu

Harbin Engineering University, College of Information and Communication Engineering, Harbin 150000, China

Correspondence should be addressed to Jiangzhi Fu; fujiangzhi@hrbeu.edu.cn

Received 27 June 2021; Revised 24 August 2021; Accepted 11 September 2021; Published 12 October 2021

Academic Editor: Xin Liu

Improving the attack resistance of the modulation classification model is an important means to improve the security of the physical layer of the Internet of Things (IoT). In this paper, a binary modulation classification defense network (BMCDN) was proposed, which has the advantages of small model scale and strong immunity to white box gradient attacks. Specifically, an end-to-end modulation signal recognition network that directly recognizes the form of the signal sequence is constructed, and its parameters are quantized to 1 bit to obtain the advantages of low memory usage and fast calculation speed. The gradient of the quantized parameter is directly transferred to the original parameter to realize the gradient concealment and achieve the effect of effectively defending against the white box gradient attack. Experimental results show that BMCDN obtains significant immune performance against white box gradient attacks while achieving a scale reduction of 6 times.

1. Introduction

The Internet of Things (IoT) is an open and comprehensive network of intelligent objects. It is deployed in different environments through various sensor devices to realize real-time collection and interaction of different monitored, connected, and interactive objects or processes [1, 2]. The IoT has been widely used in all aspects of life, including smart transportation, smart homes, smart cities, smart factories, emergency, medical care, and logistics transportation, bringing great convenience and benefits to human life, and there are still a large number of technologies that improve the efficiency of the Internet of Things constantly emerging [3, 4]. However, compared with the gradual expansion of the application range of the IoT devices on a global scale, the development of its security technology is far behind. If not defended, active malicious physical layer attacks such as deception and interference will disrupt the original communication and transmission order of the IoT, greatly reduce the communication performance of legitimate users, and even infringe on the privacy of users, harm personal safety, and affect industrial production [5–7]. Therefore, the physical layer security of the IoT has been brought to an unprecedentedly important position. As the basis of software wireless, cognitive radio, and spectrum detection, automatic modulation recognition has become an effective means to deal with physical layer security issues [8, 9]. In addition, automatic modulation classification technology is also used in various civil and military fields, such as user legitimacy detection, spectrum detection and management, interference and identification, electronic exhibition, and threat analysis [10–13].

In recent years, artificial intelligence technology has made many achievements in the fields of image recognition and natural language processing, and it is also widely used in communication-related topics, including metalearning for channel estimation, reinforcement learning for resource scheduling, and transfer learning for stations/access points switching [14–17]. More and more researchers have combined artificial intelligence technology with signal processing technology and have achieved many very valuable results for the subject of automatic modulation classification [18]. Literature [19] proposes an adaptive extensible neural network for modulation classification in the multipath fading channel by dividing the network into the amplitude, phase, and frequency weight subnetwork.

Literature [20] proposes that the complex network is used to discover the deep features of the modulated I/Q signal for modulation signal, which has achieved superior performance. Literature [21] proposes a transfer learning-based semisupervised modulation classification to address the problem of a small number of samples that are labelled and a large number that are unlabeled in real communication scenarios. Literature [22] proposes a signal recognition and reconstruction convolutional neural networks, the first zero-shot learning work, by studying the representation of signal semantic feature space. These research results have played a huge role in promoting the subject of modulation signal classification.

However, the automatic modulation classification method based on artificial intelligence technology has the following two obvious shortcomings. (1) While the model classification performance is improved, its scale tends to increase with it, which will increase the storage complexity and computational complexity of the model to the point where the edge devices of the IoT cannot afford it [23]. (2) The deep learning model is to fit high-dimensional data, rather than truly understand the data. This leads to the existence of adversarial samples, which only add a slight disturbance to the data that humans cannot detect, which makes the model produce a very outrageous output. This will also become a threat to the security of the IoT [24]. Literature [25] constructs a convolutional neural network to classify the modulated signal and proposes an index of activation maximization to evaluate the importance of the filter in the network. The network can still obtain the same or higher accuracy when the compression rate is 80%. Literature [26] converts the modulated signal into the form of a constellation diagram and constructs a binary convolutional neural network to classify it. While maintaining the same or higher classification accuracy rate, the model storage compression can be achieved by 26 times. Literature [27] introduces a scaling factor for each neuron in CNN and enforces scaling factors sparsity via compressive sensing to screen out redundant neurons and then these neurons are pruned. Literature [28] designs a communication signal adversarial sample by adding a carefully designed counter disturbance, which can ensure that the communication performance is not damaged, while reducing the intruder's modulation classification accuracy. Literature [29] launched white box and black box attacks on the modulation signal classification model. Both attack methods can significantly reduce the classification accuracy of the model. This literature further proves that the model classification confidence is inversely proportional to the attack success rate. Literature [30] uses the generative adversarial networks for semisupervised learning, which improves the model's robustness and generalization ability for modulated signal classification. In addition, a lot of work has been carried out, focusing on the evaluation of adversarial perturbation in communication signals such as invisibility of adversarial examples, the effectiveness of adversarial attacks on signals, and fitting difference to measure the perturbed waveforms [31–33]. Therefore, the research on the automatic modulation classification model with both lightweight and antiattack has important research value for the whole problem of the IoT with a large number of edge devices.

In order to further improve the reliability of the application of artificial intelligence technology in the security of the IoT, this paper designs a binary modulation signal classification defense network (BMCDN). In the forward propagation of the binarized convolutional layer, the network parameters and input are quantized from 32-bit floating-point type to 1-bit integer type, which reduces the storage overhead of the model. In the original convolution operation, the 32-bit floating-point multiplication operation was replaced by the bit operation, and the accumulation operation was replaced by the counting operation, which reduced the calculation time of the model. In the backward propagation process of the binarized convolutional layer, the gradient of the quantized input and parameters are directly passed to the original input and parameters to update the network. At this time, the gradient obtained by the input is not its true gradient, but the gradient of its quantized value, which has the effect of gradient masking and can effectively defend against white box gradient attacks.

This paper is organized as follows. In Section 2, typical white box gradient attack methods are briefly introduced. In Section 3, the proposed defense framework BMCDN is presented and analyzed. In Section 4, the comprehensive experiments are described to verify the advantages of model scale and immunity to white box gradient attacks. Section 5 draws the conclusions.

1.1. Attacks. According to how much information the attacker has on the target model, attack methods can be divided into white box attacks, gray box attacks, and black box attacks. Among them, white box attacks are the most commonly used method to evaluate defense performance [34]. In a white box attack, the attacker knows the network architecture, parameters, training data, etc., of the model. This paper selects fast gradient sign method (FGSM) and projected gradient descent (PGD) in the white box attack to attack the design model and evaluate its defense performance.

1.2. FGSM. FGSM is a typical single-step attack. According to whether the attack target has a specific category, it can be divided into untargeted attack and targeted attack [35]. The FGSM algorithm is shown in the following formula:

$$\text{untargeted: } \mathbf{x}' = \mathbf{x} + \varepsilon \cdot \operatorname{sign}\left(\nabla_x J\left(f(\mathbf{x}), y_{\text{true}}\right)\right), \tag{1}$$

$$\text{targeted: } \mathbf{x}' = \mathbf{x} - \varepsilon \cdot \operatorname{sign}\left(\nabla_x J\left(f(\mathbf{x}), y_{\text{target}}\right)\right), \tag{2}$$

where $f(\cdot)$ donates the target model. Its input data is the modulated signal $\mathbf{x}$, and output is the classification result; $J(f(\mathbf{x}), y)$ donates the cost function between model output and label; $\nabla_{\mathbf{x}} J(f(\mathbf{x}), y)$ donates the partial derivative of the cost value with respect to x; $\operatorname{sign}(\cdot)$ will get the direction of the gradient; ϵ donates the step size of the perturbation; formula (1) indicates that perturbation is added to x to increase the loss value between the model output and the real label, while formula (2) indicates that perturbation is added

to x to decrease loss value between the model output and the target label to get the adversarial sample $\mathbf{x}^{'}$, respectively.

1.3. PGD. PGD is a typical iterative attack, which can be seen as a combination of random perturbation and an iterative version of FGSM [36]. PGD is considered to be the most adversarial attack algorithm, under the same perturbation intensity. The PGD algorithm is shown in the following formula:

$$\begin{aligned} x_0' &= x + n, \\ \text{untargeted: } x_{i+1}' &= \text{clip}_{\varepsilon,x}\left(x_i' + \alpha \cdot \text{sign}\left(\nabla_x J\left(g\left(x_i'\right), y_{\text{true}}\right)\right)\right), \\ \text{targeted: } x_{i+1}' &= \text{clip}_{\varepsilon,x}\left(x_i' - \alpha \cdot \text{sign}\left(\nabla_x J\left(f\left(x_i'\right), y_{\text{target}}\right)\right)\right), \end{aligned} \tag{3}$$

where n is a random perturbation; α is the perturbation step length of each step. Usually, the number of iteration steps is greater than the ratio of disturbance intensity ϵ to α; $\text{clip}_{\epsilon,\mathbf{x}}(\cdot)$ will clip the adversarial example $\mathbf{x}'$ to maintain it in the ϵ−neighbourhood of original data.

2. Methods

In order to further improve the reliability of the application of artificial intelligence technology in the security of the IoT, this paper designs a binary modulation signal classification defense network. Extremely low bit width will bring storage and computational advantages. The extremely low weight bit width will give it storage and calculation advantages in forward operations. The direct return of the binarized gradient to the original parameters will also achieve the effect of gradient concealment and achieve the effect of effectively defending against white box gradient attacks.

2.1. Forward Propagation. The network binarization method selected in this paper is the deterministic binarization method among the naive binarization methods [37]. In the deterministic binarization method, the parameters of the binarized convolutional layer and the input quantization rules are as follows:

$$w^b = \begin{cases} +1, & w \geq 0, \\ -1, & \text{otherwise}, \end{cases} \tag{4}$$

where w is the original full-precision parameter in the network, and w^b is the binarized parameter obtained after binarization. The sign function $\text{sign}(\cdot)$ can be used to obtain the sign of the parameter and input to realize the binarization.

In actual computer storage, setting 0 bit represents parameter value −1, and setting 1 bit represents parameter value −1. The truth value table of the multiplication of 1-bit weight and the computer variable operation after quantization is shown in Table 1.

From Table 1, we can find that the ±1 weight multiplication operation obtained by the above quantization rule is equivalent to the "XNOR" operation of the computer's 1-bit variable. Therefore, the accumulation operation after the convolution kernel operation can also be replaced by the "bitcount" counting operation. The operation rules of the binarized convolutional layer are shown in Figure 1.

Table 1: The truth value table of the multiplication of 1-bit weight and the computer variable operation.

Multiplication of 1-bit weight			Equivalent computer operation		
$\mathbf{W_m}$	$\mathbf{W_n}$	$\mathbf{W_m} \times \mathbf{W_n}$	V_m	V_n	$V_m \odot V_n$
1	1	1	1	1	1
1	−1	−1	1	0	0
−1	1	−1	0	1	0
−1	−1	1	0	0	1

2.2. Backward Propagation. The gradient during training of the binarized network is still obtained through backward propagation. The most critical step is the binarization operation, which directly affects the gradient calculation of the parameters and input before the binarization. The curve of the $\text{sign}(\cdot)$ function used in the binarization process is shown in Figure 2.

From the figure, we can see that the derivative of the function is almost zero everywhere. This will cause the gradient to be blocked when it is propagated here. In order to avoid the problem of the gradient propagation being blocked, this experimental design transfers the binarized parameters and the input gradient directly to the original parameters and input, respectively. Among them, the gradient of the original data exceeding [−1, 1] is set to zero. After an update step is completed, original data will be clipped to the interval of [−1, 1]. A very simple example is listed, as shown in Figure 3.

In Figure 3, from left to right, a process in which weights and inputs are binarized and convolution is calculated. The lower right corner of each data indicates the respective gradient value after backpropagation. The gradient propagated to output through the previous layer is assumed to be 1. It can be seen from Figure 3 that the gradient obtained from the original input and weight is the gradient of the binarized input and weight. Although the gradients they get are not their own gradients, this at least ensures that the network can be updated and optimized. Obviously, the gradient obtained by the weight and input is not its optimal gradient, which makes the update and optimization direction in each update step not optimal.

However, this gradient propagation method will achieve a very good gradient masking effect. The core idea of the white box gradient attack is to add a disturbance to the input, which will increase the cost of the output and the label the fastest. This direction is actually the reverse of the optimal gradient of the input data. In the binary convolutional layer, this optimal gradient is hidden. As a result, it will gain high immunity against white box gradient attacks. The principle of its immunity to the white box gradient attack is shown in Figure 4.

Figure 4 shows a simplified two-dimensional contour map of the cost value of network output and label for one

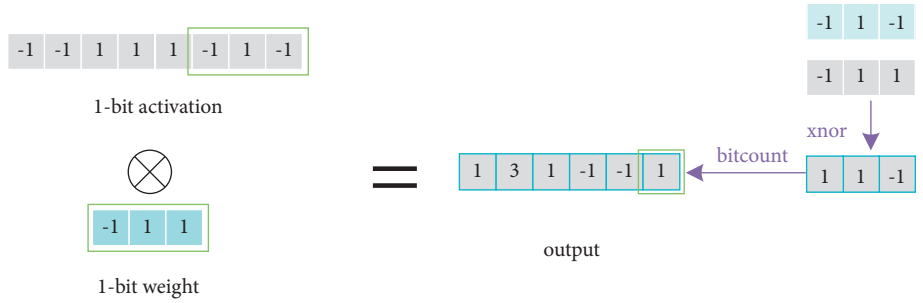

Figure 1: Schematic diagram of binary convolutional layer operation rules.

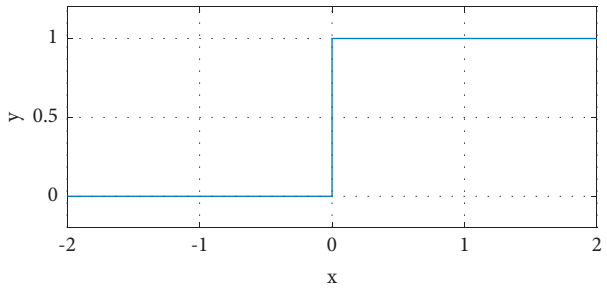

Figure 2: The curve of the sign(·) function.

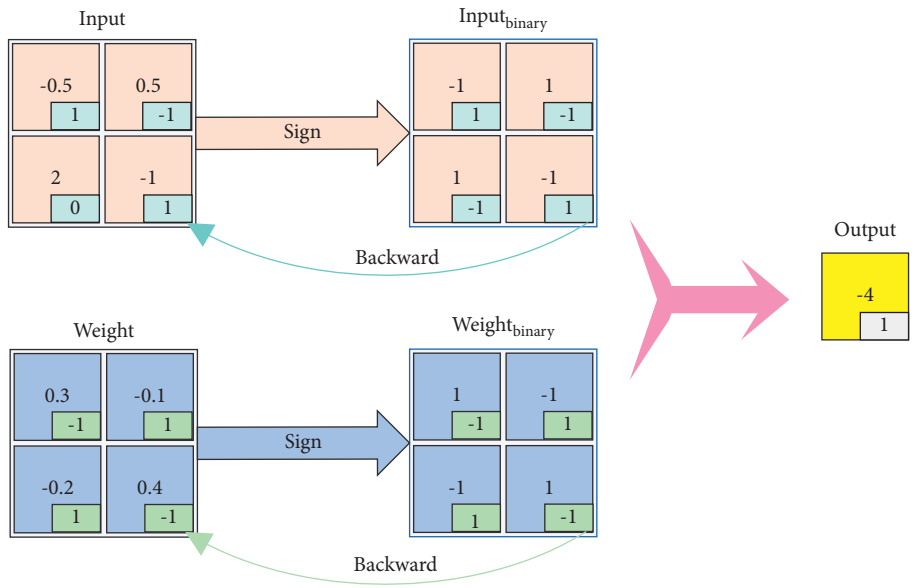

Figure 3: Schematic diagram of the gradient backpropagation rule of the binary convolutional layer.

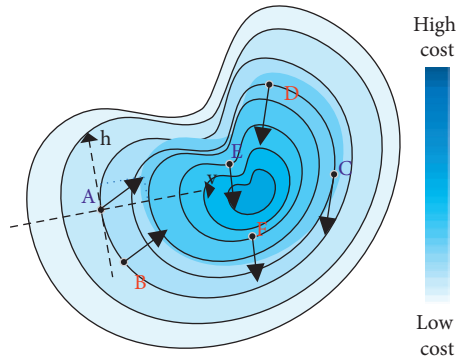

Figure 4: Two-dimensional contour map of the cost value.

data. The darker part has a higher cost value. The points on the same circle have the same cost value. Point A reaches point B after binarization, and point C and point D and point E and point F are the same. A data will inevitably transform from itself to its binary state. Therefore, point A and point B have the same cost value. After back-propagation, point A obtains the gradient of point B. The attack generated by the attack algorithm only produces an attack effect in the v direction, and there is no attack effect in the h direction, so the attack effect will be weakened or invalidated as point C and point D. It even produces the opposite effect of the attack as point E and point F. The training of the binary network is a process of multiple iterations. The gradient value obtained through back-propagation at one time may not be conducive to training, but the next time, a gradient that is conducive to training may be obtained. After multiple iterations, it can be ensured that the network as a whole is updated in a direction that fits the data well.

3. Architecture

This paper designs and constructs a BMCDN that directly processes the waveform domain modulation signal. In each layer of the network, we use the block structure shown in Figure 5. Among them, Float/Binary Layer is used for feature extraction and classification, BatchNormal Layer is used to normalize data to speed up training, and MaxPooling Layer implements feature fusion to enhance the antinoise performance of the network.

By analyzing the data of the modulated signal, we can find that there are a large number of fine-grained features in the signal, which enable them to carry different information in different ways. If the binarization process is carried out too early, these features will be filtered out or submerged. Therefore, the first two layers in the network designed in this paper still maintain the float-blocks to extract fine-grained features in the signal. After passing through the float-blocks, the one-dimensional data features are mapped to the high-dimensional network space, which can ensure that the subsequent binarization processing will not bring too much information loss. After the float-blocks, binary-blocks with filters of different sizes are designed to extract coarse-grained features from the data. For the feature classification step, this paper designs to use a binarized convolutional layer with a 1×1 size filter as the classification layer. Between the coarse-grained feature extraction layer and the classification layer, a transition layer is designed to be inserted to match the data size. After the classification layer, a Float Layer with a size of 1×1 convolution kernel combined with the softmax layer is designed as an output layer to obtain classification probabilities of different classifications. The structure of the BMCDN is shown in Figure 6.

4. Results and Discussion

The dataset used in this study is the public modulation signal dataset RML2018.10A published by [38]. In this dataset, 24 types of single-carrier modulation signals that are widely used are collected, including OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QB-WC-AM-AM-WC, AM-DSB-SC, FM, GMSK, and OQPSK. The SNR ranges from −20 dB to 30 dB in 2 dB steps. The parameters of the pulse shaping filter, the multipath fading of the channel and other system, and environmental parameters are all taken into consideration when collecting data. In this paper, the dataset is divided into training set, validation set, and test set according to the ratio of 8:1:1. The experimental environment in this paper is a computer with GTX2080 GPU resources and Windows 10 operating system. The experiment is based on the PyTorch deep learning framework and the DeepRobust adversarial sample attack and defense platform [39].

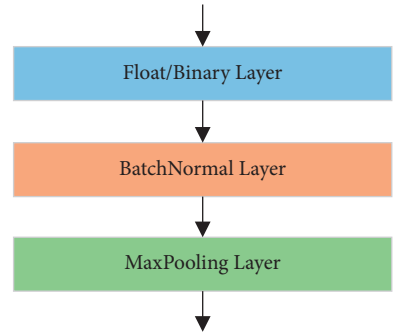

Figure 5: Schematic diagram of block structure.

4.1. Analysis of the Classification Effect of the Model. In order to obtain a clear positioning of the classification performance of the constructed BMCDN, the experimental design constructs a full-precision network with the same architecture for comparison, float modulation classification defense network (FMCDN). The two networks use the same training environment and parameters, the loss function is the cross-entropy loss function, the optimizer is adaptive moment estimation, the batch size is 256, and the learning rate is $5e-5$. The test classification accuracy rates of the two trained models under modulated signal data with different SNR are shown in Figure 7.

From Figure 6, we can find that the classification accuracy of FMCDN and BMCDN is improved with the increase of SNR and tends to stabilize around 14 dB. The accuracy of FMCDN is stable around 96%, while the accuracy of BMCDN is stable around 90%. In the two networks with the same architecture, the parameters of FMCDN are all floating-point data, and the fitting ability of the network is obviously better than that of BMCDN, so the difference in accuracy is within expectations and acceptable.

4.2. Analysis of Model Size. In order to comprehensively analyze the performance of the two networks, this paper further evaluates the scale of the model from both the model file size and the running speed. The test environment selected for the experiment is JD AI's DABNN, a binary network reasoning framework highly optimized for ARM

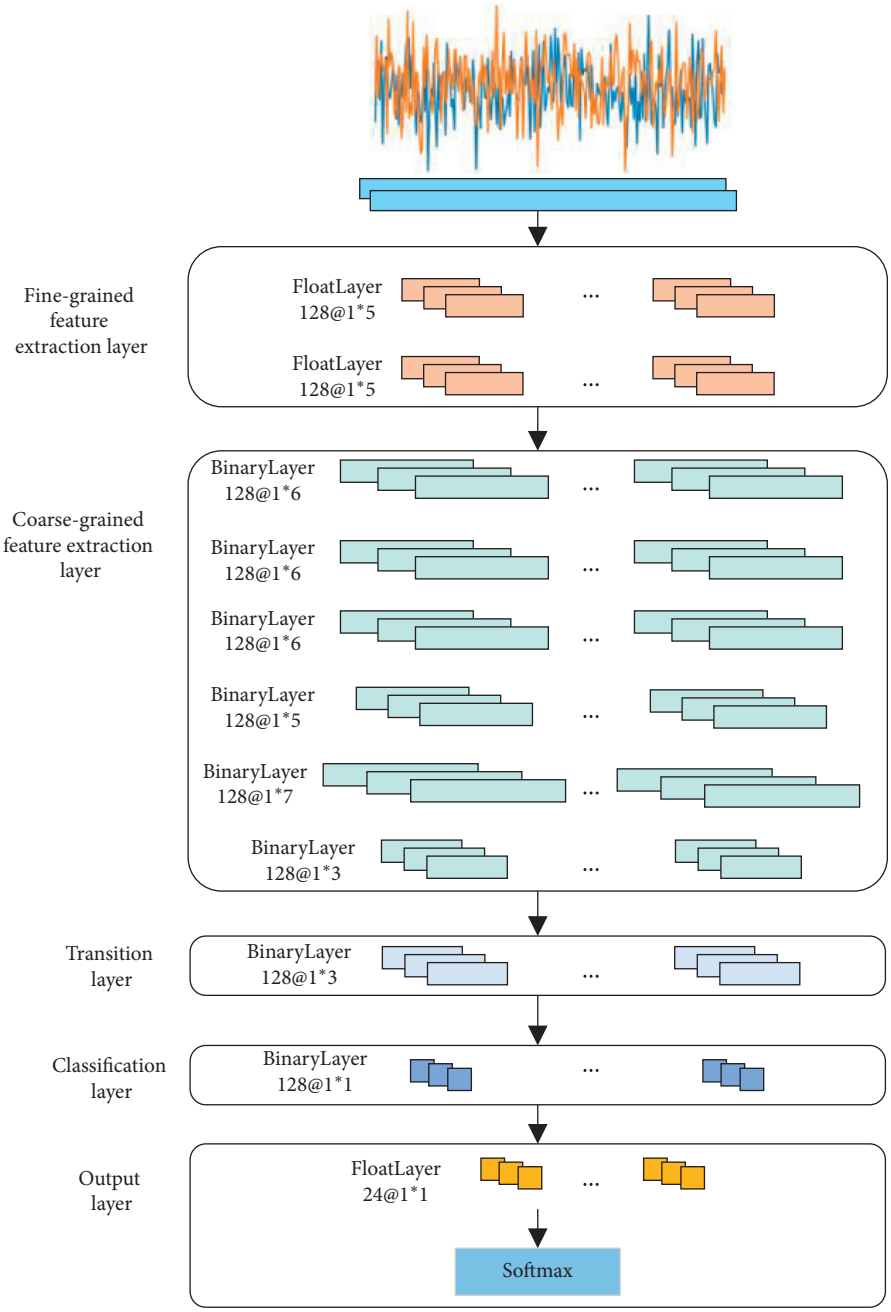

Figure 6: The structure of the BMCDN.

instruction set [40]. In order to obtain a more accurate and stable difference, the selected platform is a Raspberry Pi 4B with a single ARM core and a frequency of 1.5 GHz. After half an hour of startup, the model is run for 100 times to compare its average running time. The model file size and runtime difference are shown in Figure 7.

From Figure 8, we can see that the file size of the BMCDN model is less than one-sixth of the FMCDN model, and the running time is less than three-quarters of the FMCDN. In theory, the size of the binarization layer is one-third of the full-precision layer. In terms of computational complexity, floating-point multiplication operations will also be replaced with bit operations to achieve speed-up effects. The increase in speed is also related to the floating-point unit of the device. In the BMCDN, both the first two layers and the last layer remain full-precision layers, which causes the model file size to fail to reach the optimal compression limit of 32 times smaller, and some

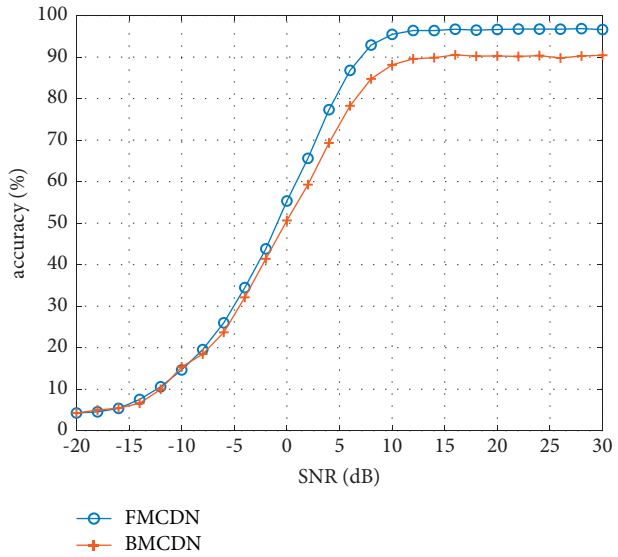

Figure 7: Test results of the two models under different SNR.

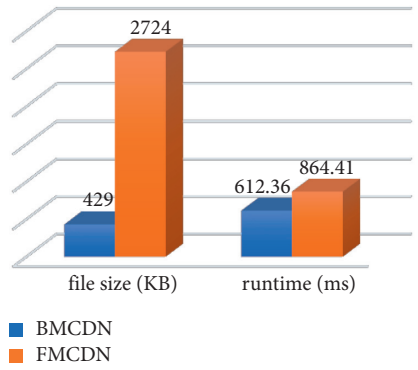

Figure 8: The attack effect of FGSM and PGD and the defensive effect after defensive training.

intermediate operations in the binarization layer are also floating-point operations, and it is reasonable to obtain the above runtime compression. The compression ratio of the model file size and runtime will increase as the proportion of the binarization layer increases.

4.3. Analysis of the Model's Antiattack Performance. This paper uses two typical white box attack methods, single-step attack FGSM, and iterative attack PGD to evaluate the antiattack performance of BMCDN and FMCDN. The perturbation intensity of the two attack methods ranges from 0 to 0.03, with a step size of 0.005. Among them, the number of iterations of PGD is the quotient of the disturbance intensity and the step size plus 5 to ensure the attack intensity. According to the classification accuracy rate in Figure 6, this paper selects representative data of SNR −8 dB, 0 dB, 8 dB, and 16 dB to attack to evaluate antiattack performance of the model. The attack effects of the two attack algorithms are shown in Figures 9(a) and 9(b).

In order to further improve the performance of the constructed model against attacks, this paper designs the use of training sets with attack disturbances to conduct defense training on the model. The perturbation intensity of the training set is 0.02. The training method and parameters of defense training are the same as the above experiment. The same attack as before is applied to the model after defensive training, and the attack effect is shown in Figures 9(c) and 9(d).

When horizontally comparing Figures 9(a) and 9(b) or Figures 9(c) and 9(d), we can find that (1) the accuracy rate decreases as the attack intensity increases for the model weather before defensive training or after defensive training. (2) For data whose classification effect is poor by the model itself, the attack algorithm is not effective against this type of data. (3) Under the same disturbance intensity, the iterative attack method PGD is more aggressive than the single-step attack method FGSM for FMCDN, while FGSM and PGD have almost the same attack effect on BMCDN. (4) Under the same disturbance intensity, BMCDN has stronger

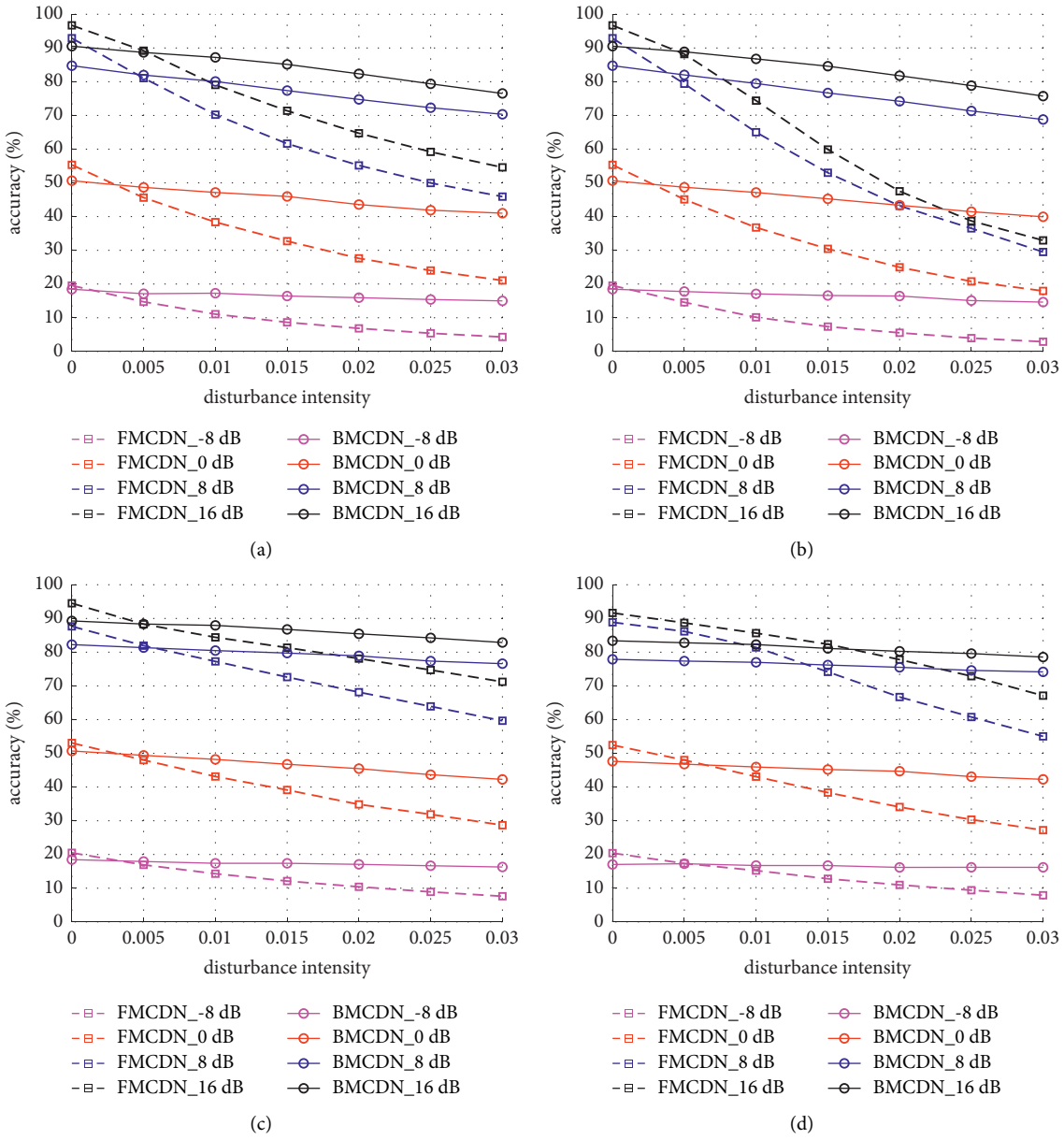

Figure 9: The attack effect of FGSM and PGD and the defensive effect after defensive training. (a) FGSM attack. (b) PGD attack. (c) FGSM defense. (d) PGD defense.

antiattack performance compared with FGSM. The gradient concealment of BMCDN makes the data not modified in the direction of the fastest increase in loss every time the antiperturbation is resisted. In addition, the weak disturbance is overwhelmed by the binarization operation. Therefore, BMCDN obtains a strong antiattack performance against white box gradient attacks.

When vertically comparing Figures 9(a) and 9(c) or Figures 9(b) and 9(d), we can find that (1) after defensive training, the defense performance of the two models for the two attack algorithms is improved, and the improvement of FMCDN is greater than that of BMCDN. (2) However, the antiattack performance of the FMCDN model after defensive training is weaker than that of the BMCND model before defensive training. Therefore, the optimization effect of defense training on the decision boundary is not as good as the defense performance of gradient mask against white box gradient attacks.

5. Conclusions

Aiming at the problem of high computational complexity and vulnerability to adversarial sample attacks when artificial intelligence is applied to the issue of IoT security, this paper designs a modulation classification defense model BMCDN that combines fast inference and antiwhite box gradient attacks to detect the malicious attacks and interference in the IoT. In the BMCDN model, the binarization operation enables the multiplication and accumulation operations in the forward inference of the network to be

replaced by bit operations, which greatly reduces the amount of parameters and the computational complexity. In addition, the gradient masking in backpropagation also enables it to have the performance of resisting white box gradient attacks far exceeding the full-precision network model with the same architecture.

While obtaining the above results, we have been keeping these points in mind: (1) there is no free lunch. The model obtains the characteristics of a smaller scale and resisting white box gradient attacks, so it will inevitably show the vulnerability to certain attacks. Further research on the defense performance of BMCDN is necessary. (2) We believe that there are still redundant connections in the BMCND model, and further pruning may result in a more lightweight physical layer defense model of the IoT. (3) More complex channel models should be considered, and corresponding attacks and defense methods should be studied and designed. (4) We believe that the binarization block is the reason why its network is resistant to white box gradient attacks. Therefore, we introduce a small number of binarization blocks and retain most of the full-precision blocks, which may be able to ensure the classification performance of the model while gaining defensive performance. The research on these points will further improve the deficiencies of this paper and promote the topic of physical layer security of the IoT.

Data Availability

The RML2018.10A dataset used in the experiments is public. Please refer to the corresponding literature for download URL. The source code of the proposed method is available from the corresponding author on reasonable request, http://opendata.deepsig.io/datasets/2018.01/2018.01.OSC.0001_1024x2M.h5.tar.gz.

Conflicts of Interest

The authors declare that are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This project was funded by the General Program of the National Natural Science Foundation of China (61771154) and the Basic Scientific Research Business Fees of Central Universities (3072020CF0813). At the same time, the authors thank the Harbin Engineering University for the funding of the Key Laboratory of Advanced Ship Communication and Information Technology Industry and the Ministry of Information Technology.

References

[1] L. D. Xu, W. He, and S. Li, "Internet of things in industries: a survey," *IEEE Transactions on industrial informatics*, vol. 10, no. 4, pp. 2233–2243, 2014.

[2] B. Wang, X. Zhang, and H. Wu, "A method of ZigBee automatic irrigation," *International Journal of Performability Engineering*, vol. 16, no. 4, pp. 639–646, 2020.

[3] X. Liu and X. Zhang, "NOMA-based resource allocation for cluster-based cognitive industrial Internet of things," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 8, pp. 5379–5388, 2020.

[4] X. Liu, X. B. Zhai, W. Lu, and C. Wu, "QoS-guarantee resource allocation for multibeam satellite industrial Internet of things with NOMA," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 3, pp. 2052–2061, 2021.

[5] Q. Qi, X. Chen, C. Zhong, and Z. Zhang, "Physical layer security for massive access in cellular Internet of Things," *Science China Information Sciences*, vol. 63, no. 2, pp. 1–12, 2020.

[6] S. Hameed, F. I. Khan, and B. Hameed, "Understanding security requirements and challenges in Internet of things (IoT): a review," *Journal of Computer Networks and Communications*, vol. 2019, Article ID 9629381, 14 pages, 2019.

[7] G. Gui, M. Liu, F. Tang, N. Kato, and F. Adachi, "6G: opening new horizons for integration of comfort, security, and intelligence," *IEEE Wireless Communications*, vol. 27, no. 5, pp. 126–132, 2020.

[8] S. Huang, C. Lin, W. Xu, Y. Gao, Z. Feng, and F. Zhu, "Identification of active attacks in Internet of things: joint model-and data-driven automatic modulation classification approach," *IEEE Internet of Things Journal*, vol. 8, no. 3, pp. 2051–2065, 2020.

[9] S. Huang, Y. Yao, Z. Wei, Z. Feng, and P. Zhang, "Automatic modulation classification of overlapped sources using multiple cumulants," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 6089–6101, 2017.

[10] Y. Lin, X. Zhu, Z. Zheng, Z. Dou, and R. Zhou, "The individual identification method of wireless device based on dimensionality reduction and machine learning," *The Journal of Supercomputing*, vol. 75, no. 6, pp. 3010–3027, 2019.

[11] H. Wang, J. Li, L. Guo, D. Zheng, Y. Lin, and Z. Ruolin, "Fractal complexity-based feature extraction algorithm of communication signals," *Fractals*, vol. 25, Article ID 1740008, 2017.

[12] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.

[13] Y. Lin, Y. Tu, Z. Dou, L. Chen, and S. Mao, "Contour stella image and deep learning for signal recognition in the physical layer," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 34–46, 2021.

[14] S. Park, H. Jang, O. Simeone, and J. Kang, "Learning to demodulate from few pilots via offline and online meta-learning," *IEEE Transactions on Signal Processing*, vol. 69, pp. 226–239, 2021.

[15] N. Mastronarde and M. van der Schaar, "Fast reinforcement learning for energy-efficient wireless communication," *IEEE Transactions on Signal Processing*, vol. 59, no. 12, pp. 6262–6266, 2011.

[16] M. Wang, Y. Lin, Q. Tian, and G. Si, "Transfer learning promotes 6G wireless communications: recent advances and future challenges," *IEEE Transactions on Reliability*, vol. 70, no. 2, pp. 790–807, 2021.

[17] K. Anupama, Y. C. Rao, and V. K. Gurrala, "A machine learning approach to monitor water quality in aquaculture," *International Journal of Performability Engineering*, vol. 16, no. 12, pp. 1845–1852, 2020.

[18] Q. V. Pham, N. T. Nguyen, T. Huynh-The, L. B. Le, K. Lee, and W. J. Hwang, "Intelligent radio signal processing: a contemporary survey," 2020, https://arxiv.org/abs/2008.08264.

[19] G. Q. Yang, "Modulation classification based on extensible neural networks," *Mathematical Problems in Engineering*, vol. 2017, Article ID 6416019, 10 pages, 2017.

[20] Y. Tu, Y. Lin, C. Hou, and S. Mao, "Complex-valued networks for automatic modulation classification," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10085–10089, 2020.

[21] W. Yu, G. Guan, H. Gacanin, T. Ohtsuki, H. Sari, and F. Adachi, "Transfer learning for semi-supervised automatic modulation classification in ZF-MIMO systems," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 10, no. 2, pp. 231–239, 2020.

[22] Y. Dong, X. Jiang, H. Zhou, Y. Lin, and Q. Shi, "SR2CNN: zero-shot learning for signal recognition," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2316–2329, 2021.

[23] K. Nan, S. Liu, J. Du, and H. Liu, "Deep model compression for mobile platforms: a survey," *Tsinghua Science and Technology*, vol. 24, no. 6, pp. 677–693, 2019.

[24] C. Szegedy, W. Zaremba, I. Sutskever et al., "Intriguing properties of neural networks," 2013, https://arxiv.org/abs/1312.6199.

[25] Y. Lin, Y. Tu, and Z. Dou, "An improved neural network pruning technology for automatic modulation classification in edge devices," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 5, pp. 5703–5706, 2020.

[26] S. Zhang, L. I. N. Yun, T. U. Ya, and S. Mao, "Electromagnetic signal modulation recognition technology based on lightweight deep neural network," *Journal on Communications*, vol. 41, no. 11, pp. 12–21, 2020.

[27] Y. Wang, J. Yang, M. Liu, and G. Gui, "LightAMC: lightweight automatic modulation classification via deep learning and compressive sensing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 3, pp. 3491–3495, 2020.

[28] M. Z. Hameed, A. György, and D. Gündüz, "The best defense is a good offense: adversarial attacks to avoid modulation detection," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1074–1087, 2020.

[29] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in DNN-based modulation recognition," in *Proceedings of the IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 2469–2478, IEEE, Toronto, Canada, July 2020.

[30] H. Xu, Y. Ma, H.-C. Liu et al., "Adversarial attacks and defenses in images, graphs and text: a review," *International Journal of Automation and Computing*, vol. 17, no. 2, pp. 151–178, 2020.

[31] H. Zhao, Q. Tian, L. Pan, and L. Lin, "The technology of adversarial attacks in signal recognition," *Physical Communication*, vol. 43, Article ID 101199, 2020.

[32] Y. Lin, H. Zhao, X. Ma, Y. Tu, and M. Wang, "Adversarial attacks in modulation recognition with convolutional neural networks," *IEEE Transactions on Reliability*, vol. 70, no. 1, pp. 389–401, 2020.

[33] H. Zhao, Y. Lin, S. Gao, and S. Yu, "Evaluating and Improving Adversarial Attacks on DNN-Based Modulation Recognition," in *Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference*, pp. 1–5, IEEE, Taipei, Taiwan, December 2020.

[34] Y. Tu, Y. Lin, J. Wang, and J. Kim, "Semi-supervised learning with generative adversarial networks on digital signal modulation classification," *CMC-Computers Materials & Continua*, vol. 55, no. 2, pp. 243–254, 2018.

[35] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, https://arxiv.org/abs/1412.6572.

[36] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, https://arxiv.org/abs/1706.06083.

[37] H. Qin, R. Gong, X. Liu, X. Bai, J. Song, and N. Sebe, "Binary neural networks: A survey," *Pattern Recognition*, vol. 105, Article ID 107281, 2020.

[38] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[39] Y. Li, W. Jin, H. Xu, and J. Tang, "Deeprobust: a pytorch library for adversarial attacks and defenses," 2020, https://arxiv.org/abs/2005.06149.

[40] J. Zhang, Y. Pan, T. Yao, H. Zhao, and T. Mei, "Dabnn: a super fast inference framework for binary neural networks on arm devices," in *Proceedings of the 27th ACM International Conference on Multimedia*, pp. 2272–2275, Nice, France, October 21-25, 2019.