Radar Emitter Signal Recognition Based on One-Dimensional Convolutional Neural Network with Attention Mechanism

Bin Wu [1], Shibo Yuan [1,*], Peng Li [1], Zehuan Jing [1], Shao Huang [2] and Yaodong Zhao [2]

[1] School of Electronic Engineering, Xidian University, Xi'an 710071, China; bwu@xidian.edu.cn (B.W.); li_peng001@163.com (P.L.); m18903513779@163.com (Z.J.)

[2] Science and Technology on Electronic Information Control Laboratory, Chengdu 610036, China; huangshaolee@163.com (S.H.); zhyd0921@163.com (Y.Z.)

[*] Correspondence: yuanshibo@stu.xidian.edu.cn; Tel.: +86-136-2341-9404

Abstract: As the real electromagnetic environment grows complex and the quantity of radar signals turns massive, traditional methods, which require a large amount of prior knowledge, are time-consuming and ineffective for radar emitter signal recognition. In recent years, convolutional neural network (CNN) has shown its superiority in recognition so that experts have applied it in radar signal recognition. However, in the field of radar emitter signal recognition, the data are usually one-dimensional (1-D), which takes more time and storage space than by using the original two-dimensional CNN model directly. Moreover, the features extracted from convolutional layers are redundant so that the recognition accuracy is low. In order to solve these problems, this paper proposes a novel one-dimensional convolutional neural network with an attention mechanism (CNN-1D-AM) to extract more discriminative features and recognize the radar emitter signals. In this method, features of the given 1-D signal sequences are extracted directly by the 1-D convolutional layers and are weighted in accordance with their importance to recognition by the attention unit. The experiments based on seven different radar emitter signals indicate that the proposed CNN-1D-AM has the advantages of high accuracy and superior performance in radar emitter signal recognition.

Keywords: radar emitter signal recognition; one-dimensional convolutional neural network; attention mechanism

1. Introduction

Radar emitter signal recognition is a technology used to obtain information about radar systems by intercepting and analyzing their signals. The features of radar signals are always extracted manually based on traditional methods. Much research has been done on feature extraction. Bouchou et al. [1] calculated eight key features, including higher-order cumulants (HOC), and used stacked sparse autoencoder (SSAE) to recognize seven different digital modulation signals. Park et al. [2] used wavelet features and support vector machines (SVM) to recognize eight different digital modulation signals. However, as the real electromagnetic environment grows complex and the quantity of radar signals turns massive, the performance of traditional methods, which require a great deal of prior knowledge and time, is poor when the radar emitter signals are on low signal-to-noise ratio (SNR).

It is expected to develop a generic and effective method that can automatically extract features from radar signals. Deep learning [3] has attracted great attention in the field of artificial intelligence, and convolutional neural network (CNN) [4,5] performs well in recognition. A large amount of research on radar emitter signal recognition has been carried out using CNN. Qu et al. [6] trained a CNN model and deep Q-learning network, which use time-frequency images extracted by Cohen class

time-frequency distribution as the input. Shao et al. [7] proposed a deep fusion method based on CNN, which provides competitive results in terms of classification accuracy. Wang et al. [8] combined the time-frequency maps and instantaneous autocorrelation maps of radar signals and used the joint feature maps as the input of CNN, which overcomes the weakness of a single feature map for the classification. Liu et al. [9] proposed an algorithm of radar emitter signal recognition, which uses the time-frequency images as the input of CNN. Cain et al. [10] combined radar frequency, pulse width and pulse repetition interval and used CNN for individual radar identification. Xiao et al. [11] proposed a method based on CNN, which uses the frequency features of automatic dependent surveillance broadcast (ADS-B) signal. Akyon et al. [12] classify the intra-pulse modulation of radar signals based on feature fusion and CNN.

However, in the field of radar emitter signal recognition, most of the sampled radar signals are one-dimensional (1-D) time-domain sequences. If we use the original two-dimensional (2-D) CNN models directly, it will take more time and storage space to transfer the sequences from 1-D form to 2-D form. Moreover, the dimensional transformation will result in poor real-time performance when the 2-D CNN models are used in practical applications. Although CNN models focus on global information and are able to extract features, the weights of the features are not the same, which means that the redundant and useless features can make recognition accuracy suppressed. Considering these limitations, this paper proposes a novel one-dimensional convolutional neural network with an attention mechanism (CNN-1D-AM) to extract features directly from original radar signals sequence in the time domain and focus on the key information of extracted features for radar emitter signal recognition.

The contribution of this paper can be concluded as follows:

(1) The 1-D convolutional layers can directly extract the feature from the time-domain sequences of radar signals. Moreover, compared with 2-D structure, 1-D convolutional layers save time in the dimensional transformation of radar signals, which makes the model better real-time performance in practical applications.

(2) A unit that employs an attention mechanism [13,14] is added to automatically weight the feature maps given by 1-D convolutional layers so that the important features can obtain more weights and the features which have negative impacts on recognition can be inhibited. The experimental results show that the proposed CNN-1D-AM can achieve high accuracy and has superior performance in radar emitter signal recognition.

This paper is organized as follows: In Section 2, the proposed CNN-1D-AM, which uses 1-D convolution and an attention mechanism, is introduced in detail. The experiments and discussions of the proposed methods and other compared methods are shown in Section 3. The conclusion is presented in Section 4.

2. One-Dimensional Convolutional Neural Network with Attention Mechanism (CNN-1D-AM)

2.1. One-Dimensional Convolution

CNN are usually designed to process 2-D data, especially images. As radar emitter signals are mainly in 1-D form and dimensional transformation is time-consuming, this paper proposed 1-D convolutional layers for feature extraction. The 1-D convolutional layers decrease the number of parameters compared with traditional 2-D convolutional layers. Moreover, the 1-D signals in the time domain are no longer converted into 2-D feature maps, which saves time and storage space.

Given the 1-D signal sequences $\{x_i\}_{i=1}^N$ where x_i is the i_{th} sample and N is the number of sequences. Assume that there are K filters in the first 1-D convolutional layer and L is the length of one signal sequence, which is the same as the input shape of the layer. Then the output of the filter in 1-D convolutional layer can be written as follows:

$$y_i^k = f\left(w^k * x_i + b^k\right) \tag{1}$$

where y_i^k denotes the output of the k_{th} filter, $f(\cdot)$ is the activation function, w^k and b^k are the weight and bias of the k_{th} filter, and '$*$' means convolution computation. When padding the edge of output result with zero, the output of 1-D convolutional layer can be written as $Y \in R^{L \times K}$.

Similar to 2-D CNN, a pooling layer is connected after the convolutional layers in 1-D CNN. The output of 1-D pooling layer can be written as $\widetilde{Y} \in R^{\frac{L}{r} \times K}$, where r is the rate of downsampling. A typical structure of CNN can be written as follows:

$$x_i \rightarrow Y^1 \rightarrow \widetilde{Y^1} \rightarrow Y^2 \rightarrow$$
$$\widetilde{Y^2} \rightarrow \dots \rightarrow Y^i \rightarrow \widetilde{Y^i} \rightarrow \dots \tag{2}$$

where Y^i denotes the output matrix of the i_{th} convolutional layer and $\widetilde{Y^i}$ is the output matrix of the i_{th} pooling layer.

2.2. Attention Unit

In recent years, Woo et al. [15] proposed the convolutional block attention module (CBAM) in a 2-D CNN. CBAM has proven that the order of the channel attention first and the spatial attention later performs better. This paper proposes the one-dimensional attention unit (AU-1D), which is similar to the order of the original CBAM. The AU-1D is added between the last pooling layer and the first full connection layer, where the unit helps to capture the essential features and suppress the less important information. The structure of the proposed AU-1D is shown in Figure 1.
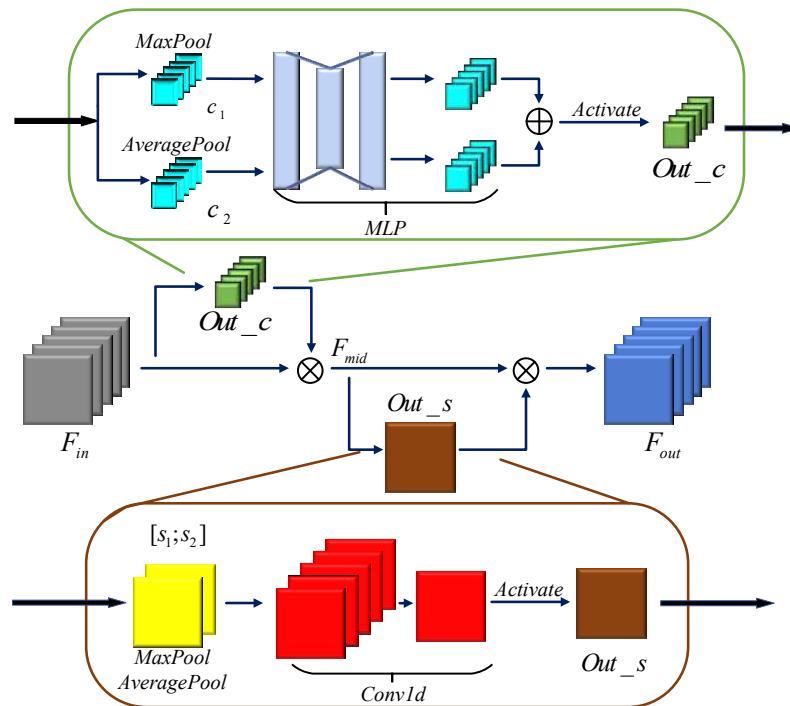

Figure 1. The structure of a one-dimensional attention unit (AU-1D).

Given a feature map $F_{in} \in R^{W \times C}$, where W is the length of the map, and C is the number of channels. AU-1D first extracts the channel features by two ways of pooling. The max-pooling function and average-pooling function in the channel domain can be written as follows:

$$c_1 = MaxPool(F_{in}) = \max(F_{in}(1 \leq i \leq W, C)) \tag{3}$$

A

$$c_2 = AveragePool(F_{in}) = \frac{1}{W} \sum_i^W F_{in}(i, C) \tag{4}$$

where $c_1 \in R^{1 \times C}$ and $c_2 \in R^{1 \times C}$ are two different vectors calculated by different ways of pooling. Then, a multilayer perceptron (MLP) is used to extract features from c_1 and c_2 further. By activating the vector which is merged by two output feature vectors from MLP, the map of channel attention $Out\_c \in R^{1 \times C}$ is produced. This process is shown as follows:

$$Out\_c = Activate(MLP(c_1) + MLP(c_2)) \tag{5}$$

The map of channel attention can be considered as a feature detector [16]. It refers to the weight for each channel in the feature map. Different convolutional kernels extract different information in the channel domain. The map of channel attention refers to the weight of each channel. The more useful information the channel brings, the more weight the channel obtains.

Then, the middle-regained feature map F_{mid} is obtained through the process of multiplying Out_c and the original feature map F_{in}. This process is shown as follows:

$$F_{mid} = F_{in} \otimes Out\_c = F_{in} \otimes \sigma(W_{MLP}(c_1) + W_{MLP}(c_2)) \tag{6}$$

where $\otimes$ stands for multiply computation, σ denotes the sigmoid function, W_{MLP} denotes the weights of MLP.

In spatial feature extraction, there are two ways of pooling whose pooling-axes [17] are different from that in channel feature extraction. The max-pooling function and average-pooling function in the spatial domain can be written as follows:

$$s_1 = MaxPool(F_{mid}) = \max(F_{mid}(W, 1 \leq j \leq C)) \tag{7}$$

$$s_2 = AveragePool(F_{mid}) = \frac{1}{C} \sum_j^C F_{mid}(W, j) \tag{8}$$

where $s_1 \in R^{W \times 1}$ and $s_2 \in R^{W \times 1}$ are two different vectors calculated by different ways of pooling. s_1 and s_2 are concatenated into a fusion vector $s \in R^{W \times 2}$. The Conv1d unit extracted information from s. By activating the output of the Conv1d unit, the map of spatial attention $Out\_s \in R^{W \times 2}$ is produced. This process is shown as follows:

$$s = [s_1; s_2] \tag{9}$$

$$Out\_s = Activate(conv1d(s)) \tag{10}$$

where $conv1d(\cdot)$ is the computation of 1-D convolution.

The map of spatial attention reflects the importance of features in different areas. Not all areas in the feature map are equally important to the recognition, but the areas which are relevant to the task of recognition should be concerned more.

Finally, the regained feature map F_{out} is obtained through the process of multiplying Out_s and the original feature map F_{mid}. This process is written as follows:

$$F_{out} = F_{mid} \otimes Out\_s = F_{mid} \otimes \sigma(W_{conv1d}([s_1; s_2])) \tag{11}$$

where W_{conv1d} denotes the weights of convolutional layers.

Through the AU-1D, the feature maps extracted from the 1-D convolutional layers will be weighted. The most useful information in the feature maps weights higher, and the useless information will be suppressed. In this way, the network can extract more effective features and improve the performance of recognition.

2.3. CNN-1D-AM

According to the analysis of the 1-D convolution and attention unit, the structure of the CNN-1D model with attention mechanism (CNN-1D-AM), this paper proposed is shown in Figure 2.
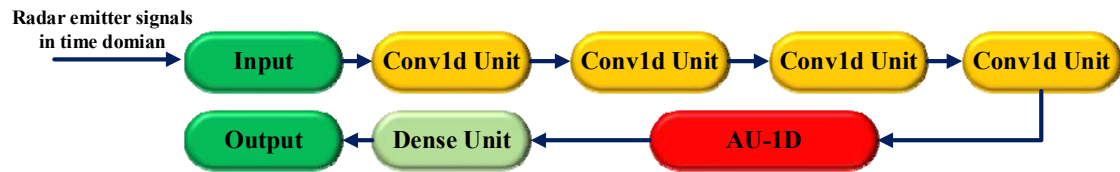

Figure 2. The structure of one-dimensional convolutional neural network with an attention mechanism (CNN-1D-AM).

In Figure 2, 'Input' is the layer, which uses the sequence of radar emitter signals in the time domain. 'Output' is the layer with a certain number of neurons, which refers to the number of signal types. 'Conv1d Unit' contains one convolutional layer, one max-pooling layer and one batch-normalization layer. The size of the convolutional kernels is 33 in four 'Conv1d Units,' and the number of filters is 32, 64, 128, 256 in turns. 'Dense Unit' contains one full connection layer.

To reduce the influence of different amplitudes on recognition, the amplitude normalization for the original data is needed. The original data are the radar emitter signals in the time domain. The expression of amplitude normalization is shown as follows:

$$d(i,j) = \frac{r(i, 1 \leq j \leq H)}{max(abs(r(i, 1 \leq j \leq H)))}, 1 \leq i \leq N \tag{12}$$

where $r \in R^{N \times H}$ are the original data sequences in the time domain, $d \in R^{N \times H}$ are the normalized data sequences in the time domain, N is the number of samples, and H is the length of each sample. The result of amplitude normalization is the input of the CNN-1D-AM model for recognition.

The activation function in the last layer is the 'SoftMax' function so that the probability for each type of signal in recognition can be obtained. The final probability for each type of signals is shown as follows:

$$\hat{y}^i = P(y = i|out) = \frac{e^{out^i}}{\sum_{i=1}^{T} e^{out^i}} \tag{13}$$

where $\hat{y} = [\hat{y}^1, \hat{y}^2, \ldots, \hat{y}^T]$, $out^i = [out^1, out^2, \ldots, out^T]$. $\hat{y}^i$ refers to the probability that the input data are recognized as class i. out^i is the output of the i_{th} neuron in the final output layer, which contains T neurons in total. The category corresponding to the maximum $\hat{y}$ is the classification result of CNN-1D-AM.

The cross-entropy (CE) function is selected as the cost function. The CE function is written as follows:

$$L(\theta) = -\sum_{i=1}^{T} y^i \ln(\hat{y}^i) = -\sum_{i=1}^{T} y^i \ln(g(\theta, x)^i) \tag{14}$$

where y is the one-hot coded result of data label, $g(\theta, x)$ denotes the output of CNN-1D-AM with x as the input, θ is the weights of the model, $L(\theta)$ is the result of the CE function.

Adaptive moment estimation (ADAM) [18] is chosen as the optimization algorithm. According to (14), this algorithm can be written as follows:

$$g \leftarrow \nabla_\theta L(\theta) \tag{15}$$

$$m \leftarrow \beta_1 m + (1 - \beta_1)g \tag{16}$$

$$v \leftarrow \beta_2 v + (1 - \beta_2)g^2 \tag{17}$$

$$m \leftarrow m/\left(1 - \beta_1^T\right) \tag{18}$$

$\sqrt{}$

$$\theta \leftarrow \theta - \alpha \cdot m / \left(\sqrt{v} + \varepsilon \right) \tag{19}$$

where g is the gradient of $L(\theta)$ by its gradient operator ∇_θ, m and v are the moment vectors with 0 as their initial value, β_1 and β_2 are constants, usually set to 0.9 and 0.999, α is the learning rate, ε is a smoothing parameter, typically set to 10^{-8}.

3. Experiments and Discussions

The experiment platform parameters for algorithm implementation are shown in Table 1.

Table 1. Experiment platform parameters.

Project	Parameter
CPU	Intel Silver 4110
GPU	P400 + P40
RAM	64 GB
System Version	Centos 7
Simulation Software	MATLAB2020a, Python3.7, Keras 2.2.4

3.1. Dataset

Seven different varieties of radar emitter signals were used to validate the effectiveness of the proposed algorithm, namely, continuous wave (CW), linear frequency wave (LFM), nonlinear frequency wave (NLFM), binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), binary frequency shift keying (BFSK) and quadrature frequency shift keying (QFSK). These seven different types of modulation are commonly used in radar systems. The specific parameters of the signals are shown in Table 2. The carrier frequency and frequency bandwidth change within a certain range, which meets the changing characteristics of the electromagnetic environment.

Table 2. Specific parameters of seven types of radar emitter signals.

Signal Type	Carrier Frequency	Parameter
CW	200 MHz~220 MHz	None
LFM	200 MHz~220 MHz	Frequency bandwidth: 50 MHz to 60 MHz
NLFM	200 MHz~220 MHz	Frequency of modulation signal ranges from 10 MHz to 12 MHz
BPSK	200 MHz~220 MHz	13-bit Barker code Width of each symbol is 0.038 us
QPSK	200 MHz~220 MHz	16-bit Frank code Width of each symbol is 0.03 us
BFSK	200 MHz~220 MHz 300 MHz~320 MHz	13-bit Barker code Width of each symbol is 0.038 us
QFSK	100 MHz~110 MHz 150 MHz~160 MHz 200 MHz~210 MHz 250 MHz~260 MHz	16-bit Frank code Width of each symbol is 0.03 us

Note 1: The pulse width for each type signal is 0.5 us; Note 2: Sampling frequency is 2 GHz.

The datasets in the experiment were produced like this:

(1) First, we generated seven types of radar emitter signals with different values of SNR. The type of noise was Gaussian white noise, and the passband ranged from 90 MHz to 340 MHz. The SNR for each type of signal ranged from −10 dB to 0 dB with 1 dB step, totaling 11 values. The number of samples for each type of signal with each value of SNR was 7000.

(2) Second, we divided the samples into three different datasets. As (1) shows, 7000 samples for each type of signal with each value of SNR were divided into training dataset with 1600 samples, validation dataset with 400 samples and testing dataset with 5000 samples.

(3) Third, we made the final datasets. The final training dataset with 123,200 samples, the final validation dataset with 30,800 samples and the final testing dataset with 385,000 samples were combined by the datasets in (2).

3.2. Experiments of CNN-1D-AM

The model CNN-1D-AM was trained based on the preprocessed data in Section 3.1. The number of parameters and training time per epoch for CNN-1D-AM is shown in Table 3.

Table 3. Quantity of parameters and training time per epoch for CNN-1D-AM.

Model	CNN-1D-AM
Quantity of parameters	3,554,504
Time per epoch	55 s

As shown in Table 3, the training time of CNN-1D-AM for each epoch with 123,200 samples was less than one minute, which means that the model was lightly designed and was on low incremental resource consumption.

The average recognition rates for the training dataset and validation dataset during the training session are shown in Figure 3.
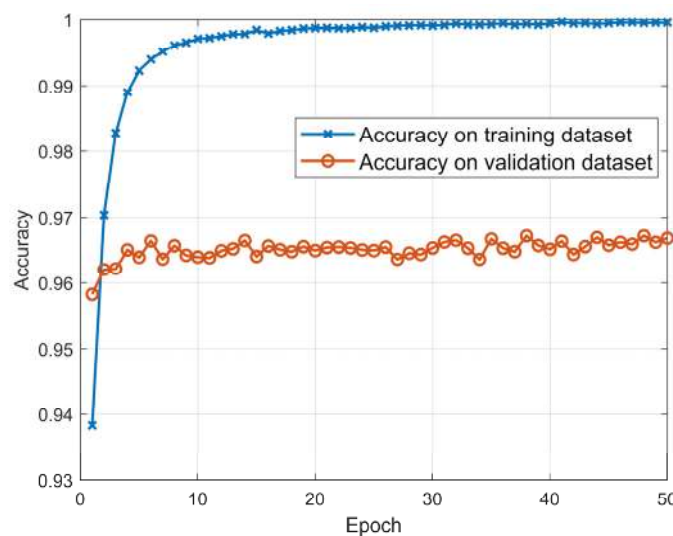

Figure 3. The average recognition rates of CNN-1D-AM on the training dataset and validation dataset with different quantity of training epochs.

Figure 3 shows that after training 50 epochs, the recognition accuracy of CNN-1D-AM on the training dataset reached nearly 100%. Moreover, the recognition accuracy of the model on the validation dataset was over 96%, which denotes that the model converged.

The weights of the neural network with the highest recognition rate on the validation dataset were saved. Under this circumstance, the recognition rate of CNN-1D-AM with 11 values of SNR on the validation dataset is shown in Figure 4.
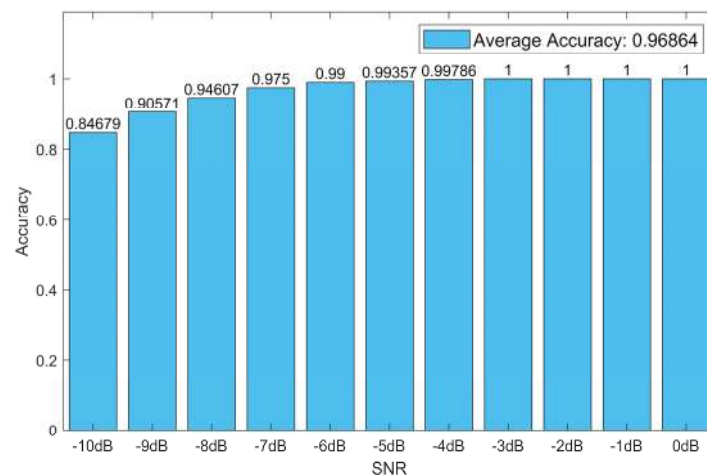
Figure 4. The recognition rates of CNN-1D-AM with 11 values of signal-to-noise ratio (SNR) on the validation dataset.

Figure 4 indicates that the model acquired nearly 100% accuracy when the SNR was above −6 dB. Moreover, the accuracy was less than 90% only when SNR was lower than −9 dB.

In the real applications, the number of samples which need to be tested is always larger than that on the validation dataset. Therefore, the testing dataset with large-scale samples was used to validate the exact real performance of the model. The recognition rate of CNN-1D-AM with 11 values of SNR on the testing dataset is shown in Figure 5.
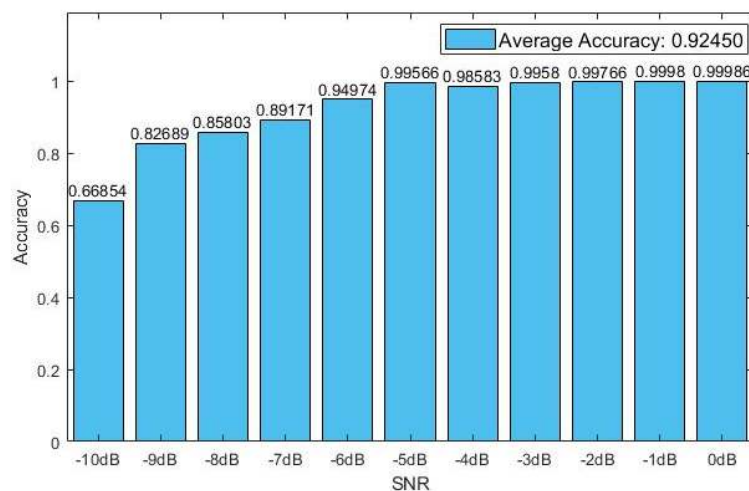
Figure 5. The recognition rates of CNN-1D-AM with 11 values of SNR on the testing dataset.

As shown in Figure 5, the average recognition rate of CNN-1D-AM decreased compared with Figure 4. This is because the number of samples on the testing dataset was about 12.5 times more than that on the validation dataset and 3.125 times more than that on the training dataset. This is equivalent to the situation that a model is trained with fewer samples and is tested with a huge number of samples. When SNR was above −5 dB, the accuracy of recognition on the testing dataset was still close to 100%. Interestingly, the recognition rate fell nearly 1% when the SNR rose from −5 dB to −4 dB.

To figure out the specific recognition results of CNN-1D-AM, the confusion matrix for average recognition performance based on the testing dataset is shown in Figure 6. It was found that the part of low recognition rates could be attributed to the classification of BFSK signals. A portion of the BFSK signals was mainly misidentified as CW signals and BPSK signals. Apart from this, the average recognition rates of the other six types of signals were over 93.5% by calculating.
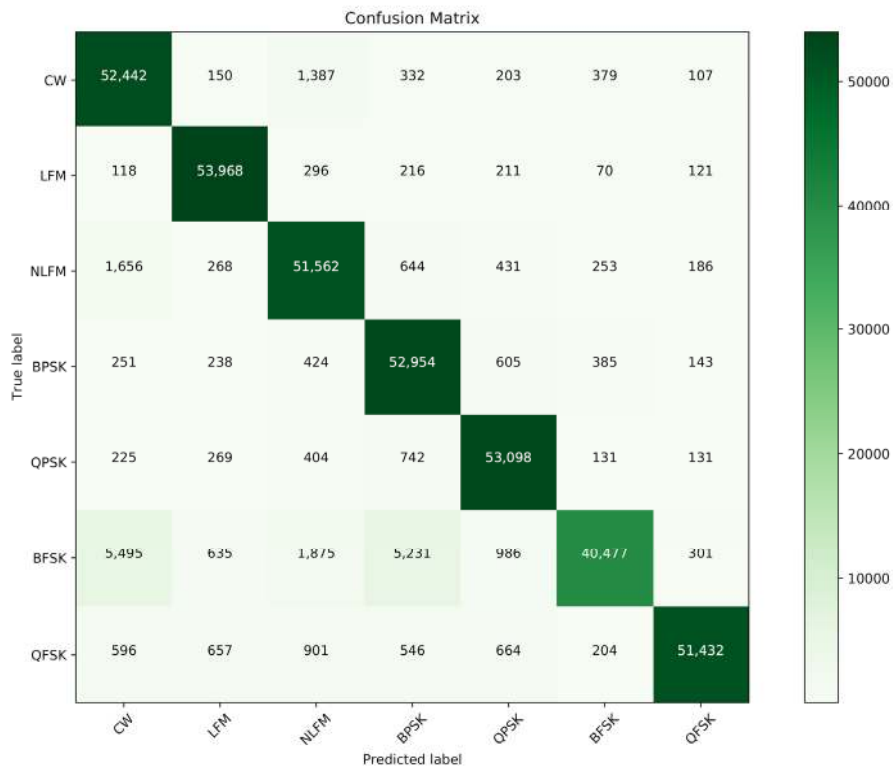
Figure 6. The confusion matrices of CNN-1D-AM, based on average recognition rates.

3.3. Learned Features

In this section, the extracted features of signals by the proposed CNN-1D-AM were investigated. Specifically, a sample from the testing dataset was sent to the CNN-1D-AM model. Some features filtered by the layer before the attention unit and weighted by the attention unit are plotted in Figures 7 and 8. The weights of the attention unit are also shown in Figure 9.

Figures 7 and 8 indicate that the features in different channels and different positions of space were weighted by the attention unit. The relative values of features in some certain channels and in some positions of space turned to zero. Moreover, Figure 9 shows that the features in different positions of space and channels gained weights differently based on the attention unit.
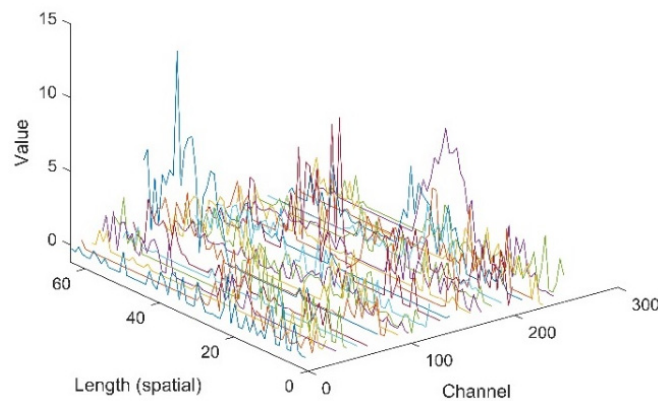

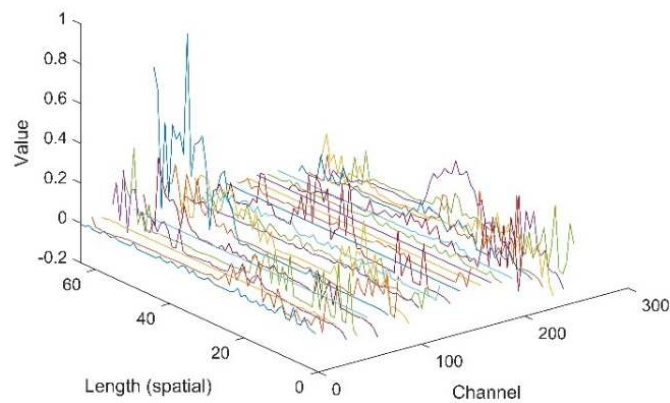**Figure 7.** The features filtered by the layer before the attention unit.

Figure 8. The features weighted by the attention unit.
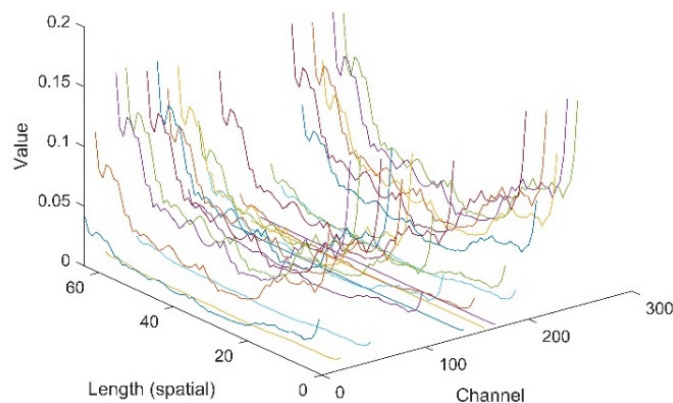

Figure 9. The weights of the attention unit.

3.4. Comparison of Other Methods

To further evaluate the effectiveness of the proposed method, some traditional methods and state-of-the-art deep learning-based models were used as a comparison.

The traditional methods include SVM [19], which uses seven HOC features as the input; SSAE1, which uses spectral power feature, amplitude feature in the time domain and six HOC features as input. Moreover, the deep learning-based models include CNN and deep neural networks (DNN) [20], stacked autoencoder (SAE) [21].

For the CNN part, the VGG network [22] and ResNet [23] were chosen as the comparison models. As the structure of the proposed CNN-1D-AM is not complicated, for this paper, we chose the specific VGG network, which includes 13 weight layers (VGG13) and the specific ResNet, which includes 18 layers (ResNet18). To make the comparison between methods as fair as possible, both of VGG13 and ResNet18 were transferred from 2-D forms, and the parameters were reset properly according to the literature. Moreover, to investigate the impact of the attention mechanism, a CNN-1D model, which is transferred by deleting the attention unit from the proposed models, was also used as a comparison (CNN-1D-Normal).

For the DNN part, four different models were chosen, and the detail of these models is shown in Table 4. The adjacent layers were fully connected. The differences among the four DNN models were the quantity of layers and the number of neurons in the layers.

In addition, three SAE models were chosen, and their structure is shown in Table 5. The SAE models included at least one autoencoder and one classifier. Moreover, the adjacent layers of autoencoders and the classifier were fully connected.

The datasets used in this session were the same as before. The input of CNN, DNN and SAE models in comparison was the sequences of radar emitter signals in the time domain. Moreover, the input data of SVM and SSAE were calculated according to the same datasets.

Table 4. The detail of four deep neural networks (DNN) models for radar emitter signal recognition.

Neurons of the Layers	DNN1	DNN2	DNN3	DNN4
Input layer	1024			
First hidden layer	512	512	256	512
Second hidden layer	256	256	64	256
Third hidden layer	128	N/A	N/A	128
Fourth hidden layer	N/A	N/A	N/A	64
Output layer	7			

Table 5. The structure of the stacked autoencoder (SAE) model for radar emitter signal recognition.

SAE Model	Parts of SAE	First Auto-Encoder	Second Auto-Encoder	Third Auto-Encoder	Classifier
SAE1	Input layer	1024	512	256	128
	Hidden layer	512	256	128	N/A
	Output layer	1024	512	256	7
SAE2	Input layer	1024	512		256
	Hidden layer	512	256	N/A	N/A
	Output layer	1024	512		7
SAE1	Input layer	1024			512
	Hidden layer	512	N/A		N/A
	Output layer	1024			7

Figure 10 shows the recognition accuracy of different methods and models with each value of SNR on the testing dataset. By analysis, the accuracy of convolutional neural network models was higher than other methods, and the performance of CNN-1D-AM this paper proposed was superior to those of other models above-mentioned. Moreover, the comparison between CNN-1D-AM and CNN-1D-Normal shows that AU-1D could improve the recognition accuracy of the network.
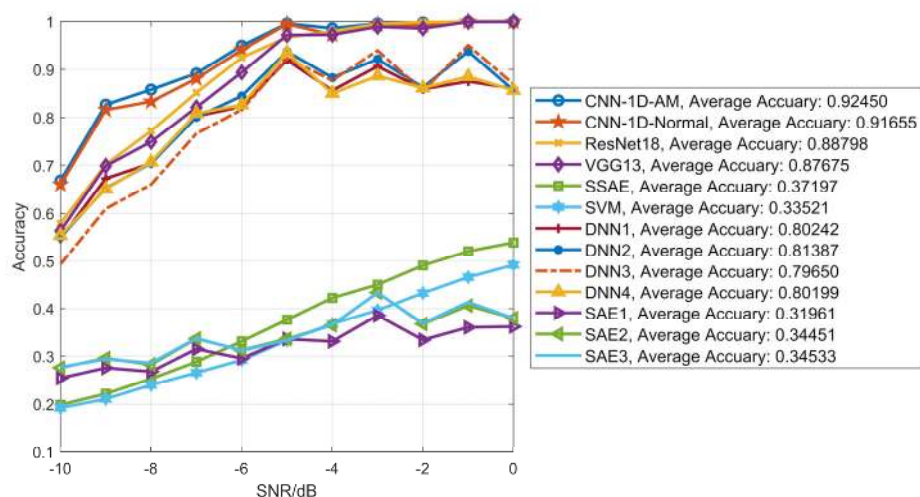

Figure 10. Recognition accuracy of different methods and models (CNN-1D-AM, CNN-1D-Normal, ResNet18, VGG13, SSAE, SVM, DNN1, DNN2, DNN3, DNN4, SAE1, SAE2, SAE3) with each value of SNR on the testing dataset.

Table 6 shows the number of parameters and training time per epoch for convolutional neural network models, which indicated that the CNN-1D-AM model was of higher efficiency and lower consumption of computation.

Table 6. The number of parameters and training time per epoch for convolutional neural network models.

Model	CNN-1D-AM	CNN-1D-Normal	ResNet18	VGG13
Quantity of parameters	3,554,504	3,520,903	4,465,543	5,761,863
Time per epoch	55 s	50 s	101 s	80 s

4. Conclusions

This paper proposes a novel CNN-1D-AM for radar emitter signal recognition. The designed 1-D convolutional layers especially could directly extract features from the time-domain sequences of radar emitter signals. The attention unit was integrated into the CNN-1D model so that the recognition accuracy of a neural network could be improved further. The experimental results indicated that CNN-1D-AM could achieve high accuracy of recognition on seven different radar signals. The comparison results with some traditional methods and deep learning-based models show the superior performance of CNN-1D-AM. In future work, we hope to propose a CNN-1D model with a new attention mechanism, which can increase the accuracy of recognition further.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Bouchou, M.; Wang, H.; Lakhdari, M.E.H. Automatic digital modulation recognition based on stacked sparse autoencoder. In Proceedings of the 2017 IEEE 17th International Conference on Communication Technology (ICCT), Chengdu, China, 27–30 October 2017; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2017; pp. 28–32.
2. Park, C.-S.; Choi, J.-H.; Nah, S.-P.; Jang, W.; Kim, D.Y. Automatic Modulation Recognition of Digital Signals using Wavelet Features and SVM. In Proceedings of the 10th International Conference on Advanced Communication Technology, Gangwon-Do, Korea, 17–20 February 2008; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2008; Volume 1, pp. 387–390.
3. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [CrossRef] [PubMed]
4. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**. [CrossRef]
5. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1–9.
6. Qu, Z.; Hou, C.; Hou, C.; Wang, W. Radar Signal Intra-Pulse Modulation Recognition Based on Convolutional Neural Network and Deep Q-Learning Network. *IEEE Access* **2020**, *8*, 49125–49136. [CrossRef]
7. Shao, G.; Chen, Y.; Wei, Y. Deep Fusion for Radar Jamming Signal Classification Based on CNN. *IEEE Access* **2020**, *8*, 117236–117244. [CrossRef]
8. Wang, F.; Yang, C.; Huang, S.; Wang, H. Automatic modulation classification based on joint feature map and convolutional neural network. *IET Radar Sonar Navig.* **2019**, *13*, 998–1003. [CrossRef]
9. Liu, Z.; Shi, Y.; Zeng, Y.; Gong, Y. Radar Emitter Signal Detection with Convolutional Neural Network. In Proceedings of the 2019 IEEE 11th International Conference on Advanced Infocomm Technology (ICAIT), Jinan, China, 18–20 October 2019; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2019; pp. 48–51.

10. Cain, L.; Clark, J.; Pauls, E.; Ausdenmoore, B.; Clouse, R.; Josue, T. Convolutional neural networks for radar emitter classification. In Proceedings of the 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 8–10 January 2018; pp. 79–83.

11. Xiao, Y.; Wei, X.Z. Specific emitter identification of radar based on one dimensional convolution neural network. *J. Phys. Conf. Ser.* **2020**, *1550*. [CrossRef]

12. Akyon, F.C.; Alp, Y.K.; Gok, G.; Arikan, O. Classification of Intra-Pulse Modulation of Radar Signals by Feature Fusion Based Convolutional Neural Networks. In Proceedings of the 2018 26th European Signal Processing Conference (EUSIPCO), Rome, Italy, 3–7 September 2018; Institute of Electrical and Electronics Engineers (IEEE): Piscataway, NJ, USA, 2018.

13. Chen, L.; Zhang, H.; Xiao, J.; Nie, L.; Shao, J.; Liu, W.; Chua, T.-S. SCA-CNN: Spatial and Channel-Wise Attention in Convolutional Networks for Image Captioning. *arXiv* **2016**, arXiv:1611.05594.

14. Hu, J.; Shen, L.; Albanie, S.; Sun, G.; Wu, E. Squeeze-and-Excitation Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *42*, 2011–2023. [CrossRef]

15. Woo, S.; Park, J.; Lee, J.-Y.; Kweon, I.S. CBAM: Convolutional Block Attention Module. In Proceedings of the European Conference on Computer Vision, Munich, Germany, 8–14 September 2018.

16. Zeiler, M.D.; Fergus, R. Visualizing and Understanding Convolutional Networks. In Proceedings of the European Conference on Computer Vision, Zurich, Switzerland, 6–12 September 2014; Springer: Cham, Switzerland, 2014.

17. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proceedings of the International Conference on Learning Representations, Toulon, France, 24–26 April 2017.

18. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.

19. Yang, C.; He, Z.; Peng, Y.; Wang, Y.; Yang, J. Deep Learning Aided Method for Automatic Modulation Recognition. *IEEE Access* **2019**, *7*, 109063–109068. [CrossRef]

20. Lim, H.-S.; Jung, J.; Lee, J.-E.; Park, H.-M.; Lee, S. DNN-Based Human Face Classification Using 61 GHz FMCW Radar Sensor. *IEEE Sens. J.* **2020**, *20*, 12217–12224. [CrossRef]

21. Yuan, X.; Huang, B.; Wang, Y.; Yang, C.; Gui, W.-H. Deep Learning-Based Feature Representation and Its Application for Soft Sensor Modeling with Variable-Wise Weighted SAE. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3235–3243. [CrossRef]

22. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.

23. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.