International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

A Compressive Sampling And Deep Learning-Based Algorithm For Wideband Signal Detection

Xudong Jin[1], **Minya Chen**[2]
Binary Graduate School, Ioi Business Park, NO. 1, 47100 Puchong, Selangor
Binary Graduate School, Ioi Business Park, NO. 1, 47100 Puchong, Selangor

Abstract:
In the field of frequency-hopping signal detection, conventional methods face the dual challenges of high computational complexity and artificially designing detection statistics, which significantly limits their application in resource-constrained scenarios. To solve this problem, we proposed a frequency-hopping signal detection algorithm, stft-cs-shufflecbam, by combining time-graph compression perception and deep learning. Short-time Fourier transform (STFT) transforms frequency-hopping signals into time chart representations, and two-dimensional compression sensing technology and half-tensor product (STP) reduce the computational load by dimensioning the time chart down. A lightweight network model, ShuffleNet, is then constructed and a convolutional block attention module (CBAM) is introduced to enhance the feature extraction capability of the frequency hopping signal. Finally, the classification and identification of the compressed time chart enabled efficient detection of the frequency-hopping signal. The goal of this research is to provide a solution that achieves both accuracy and computational efficiency for wideband signal detection in resource-constrained environments such as mobile devices and embedded systems.
In conventional research, the detection accuracy of frequency-hopping signals has been improved by deep learning and the data dimension has been reduced by compression perception. However, in the existing methods, the attenuation of the frequency hopping signal in the time frequency domain cannot be fully utilized, thus increasing the computational complexity and relying on the manual design of the detection statistics. . Furthermore, some studies have attempted to combine compression perception with deep learning, but there are limitations in maintaining detection performance in low SNR environments. In the existing method, it is easy to generate the problem that the detection probability decreases due to the loss of feature information under the condition of low s/n ratio. In addition, in existing studies, the design optimization of the compression perception measurement matrix was insufficient, and it was difficult to achieve both information retention and computational efficiency in feature extraction. Therefore, innovative methods that efficiently adapt to noise environments and improve detection accuracy while reducing computational complexity are required.
The specific process of this research is to generate a time chart of the frequency-hopping signal by using TFR Includes. In experiments, when using a 512 length humming window, the stft-shufflecam method achieves a signal-to-noise ratio (SNR) of -10 At dB, it is shown that the detection probability is improved by approximately 69.5% compared with the stft-shufflecam method and 42% compared with the wt-shufflecam method. Second, by integrating the shfflenet and CBAM modules, the proposed stft-cs ShuffleCBAM model can greatly improve the detection performance while maintaining lower computational complexity. Specifically, in a low SNR environment with an SNR of -14 dB, the detection probability of stft-s-shufflecbam was 1, 0.16 for the stft-resnet50 method, and 0.9 for the cnn-st method. For actual applications, a signal acquisition system for a rp-n310 receiver and a portable radio station was constructed to verify the robustness of the algorithm. When the sample coefficient is compressed to M=64, the network model is fine-tuned to increase the detection probability from 0.58 to 0.92. In addition, the calculation complexity of stft-cs-shufflecbam was found to be significantly lower than that of hcrnn-6 or cnn-st. For example, if M=4, the time complexity is 1.4110×10^7 FLOPs, and the space complexity is 2.4777×10^5. It is 92.7% and 95.3% lower than hcrn-6 respectively. In this research, we propose a practical solution that effectively overcomes the limitations of conventional methods by optimizing compression perception and deep learning framework, and combines the accuracy of frequency hopping signal detection with low computational complexity.
Keywords: *Frequency-Hopping Signal Detection, STFT*

1. INTRODUCTION

Frequency-hopping signal is a kind of wideband signal, characterized by a wide working bandwidth but a narrow instantaneous occupied bandwidth, and showing sparsity in the time-frequency domain. In view of the problems such as high computational complexity in the existing deep learning-based frequency-hopping signal detection algorithms and the need for manual design of detection statistics in the detection algorithms based on compressive sensing, this chapter makes full use of the sparse characteristics of frequency-hopping signals in the time-frequency domain and proposes a frequency-hopping signal detection algorithm combining time-frequency graph compressive sampling and deep learning. It is abbreviated as STFT-CS-ShuffleCBAM. Firstly, the frequency-hopping signal is converted into a time-frequency graph representation form through the short-time Fourier transform; Secondly, compressive sensing and semi-tensor product techniques are adopted to perform data dimension reduction on the time-frequency graph, thereby reducing the computational burden. Then, a lightweight network model, ShuffleNet, was constructed, and a Convolutional Block Attention Module (CBAM) was introduced to enhance the model's ability to express the characteristics of frequency-hopping signals. Then, the STFT-CS-ShuffleCBAM method is used to classify and identify the time-frequency graph after compression sampling; Finally, the performance of the proposed algorithm was evaluated through computer simulation and the actual collected frequency-hopping signals.

3. Frequency-hopping signal detection model based on time-frequency analysis

3.1 Time-domain model

Frequency hopping is a multi-carrier transmission technology that achieves signal transmission by switching between multiple frequencies. The frequency of the frequency-hopping signal varies within the preset frequency set under the control of pseudo-random codes and briefly lingers at different frequencies. The mathematical expression of the frequency-hopping signal is as follows:

$$x(t) = a(t)\sum_{i=1}^{O} g_h(t - iT_h)\cos(2\pi f_i(t)t + \varphi_0)$$

Figure 3.1

In the signal detection stage, the existence of frequency-hopping signals can be modeled as a binary hypothesis testing problem. The binary hypothesis testing problem represented by sampled signals is:

$$r(n) = \begin{cases} w(n) & H_0 \\ h(n) * x(n)e^{j\theta_0} + w(n) & H_1 \end{cases}$$

Figure3.2

3.2 Time-frequency domain model

Time-frequency analysis, by jointly analyzing the signal in both the time and frequency dimensions, provides the spectral information of the signal varying with time. The commonly used time-frequency analysis methods at present include short-time Fourier Transform (STFT), wavelet transform and S-transform, etc. Among them, the short-time Fourier transform is a fundamental and widely used time-frequency analysis method. In this chapter, STFT is adopted to conduct time-frequency analysis on frequency-hopping signals(Park,et al.2023). STFT segments the signal by introducing a sliding window function and applies the Fourier transform to each segment, thereby obtaining the time-spectrum information of the signal. STFT can be expressed as:

$$STFT_x(t,f)=\int_{-\infty}^{+\infty}x(\tau)v(\tau-t)e^{-j2\pi f\tau}d\tau$$

Figure3.3
In practical applications, signals are usually discrete, so it is necessary to discretize the continuous STFT. Discrete STFT can be expressed as:

$$STFT_x(i,k)=\sum_{n=0}^{L_v-1}x(n)v^*(n-i)e^{-j2\pi kn/L_v}$$

Figure3.4
To ensure that the analysis focuses on the signals near the current time point, which helps to reflect the instantaneous power information of the signals more accurately. This chapter adopts the TFRSTFT algorithm (Auger,et al.1996) to implement the short-time Fourier transform.
The modulus value of time-frequency analysis is called the time-frequency graph. The time-frequency graphs of the frequency-hopping signals obtained by using the three methods of STFT, wavelet transform and S-transform are shown in Figure 3.1. It can be clearly seen from the time-frequency graph of the frequency-hopping signal in Figure 3.1 that the energy of the frequency-hopping signal is mainly concentrated at a few time-frequency points. Because the frequency of the frequency-hopping signal undergoes sudden changes at different time points, a dispersed and sparse energy distribution is presented on the time-frequency graph. The sparsity of this energy distribution provides feasibility for the application of compressive sensing technology. With the help of compressive sensing, the data dimension can be effectively reduced, thereby reducing the computational complexity and achieving effective detection of frequency-hopping signals.

3.3 Frequency-hopping signal detection algorithm based on time-frequency graph compression sampling and lightweight network

3.3.1 Overall framework of the algorithm

This chapter proposes a frequency-hopping signal detection algorithm named STFT-CS-ShuffleCBAM. This algorithm not only makes full use of compressed sampling to reduce complexity while retaining the signal features of the time-frequency graph as much as possible, but also gives full play to the powerful feature extraction ability of the lightweight network. By taking advantage of the differences between the time-frequency graph of the frequency-hopping signal and the time-frequency graph of the noise, the network model can adaptively learn the features of the input time-frequency graph and perform classification, thereby achieving signal detection.The STFT-CS-ShufflecBAM algorithm first acquires the time-frequency graph of the frequency-hopping signal through STFT and takes it as the input of the lightweight network model. This time-frequency graph not only retains the key features of the original signal, but also facilitates the extraction of hidden features by subsequent neural networks. In order to reduce the overall complexity of the model, this algorithm introduces compressive sensing technology to downsample the time-frequency graph(Urkowitz,1967). For two-dimensional signal data such as time-frequency graphs, the two-dimensional compressive sensing method is adopted. In the process of compressive sensing, in order to capture as many time-frequency domain features as possible, an orthogonal complex Gaussian measurement matrix was constructed. Meanwhile, in order to reduce the storage overhead brought by the random observation matrix during the compression process, the half-tensor product matrix multiplication was further adopted to optimize the calculation process.In terms of network structure design, this algorithm selects the lightweight and efficient ShuffleNet model. ShuffleNet is a high-performance image classification network specially designed for mobile devices and embedded systems. It effectively reduces the computational load and improves the overall efficiency through group convolution and channel rearrangement techniques. To further reduce the number of model parameters, this algorithm only uses the first six layers of modules of the ShuffleNet network. Meanwhile, in order to reduce parameters while maintaining good detection performance, a lightweight convolutional attention module (CBAM) is introduced in the framework, enabling the model to focus more on the key areas in the time-frequency graph, thereby further improving the accuracy of signal detection.

3.3.2 Compressed sensing

3.3.2.1The design of the measurement matrix

In most compressive sensing schemes, the real number measurement matrix is usually adopted, while the application of complex number matrices is relatively rare. However, studies have shown that the complex measurement matrix can capture the feature information in the time-frequency graph more effectively (Wang,2024). Therefore, the complex number measurement matrix is constructed in this chapter, and its specific construction method is as follows.

First, generate two random Gaussian matrices with a mean of 0 and a variance of 1, which are expressed as:

$$\boldsymbol{M}_1 \sim \mathcal{N}(0,1)$$
$$\boldsymbol{M}_2 \sim \mathcal{N}(0,1)$$

Figure3.5

Then, perform Gram-Schmidt orthogonalization on these two matrices respectively to obtain the orthogonalization results of matrices Q_1 and Q_2 as matrices c and d respectively, thereby eliminating the correlation between them. Finally, taking matrices a and b as the real and imaginary parts of the complex matrix respectively, the orthogonal complex Gaussian measurement matrix h is constructed, namely:

$$\boldsymbol{\Phi} = \boldsymbol{Q}_1 + j\boldsymbol{Q}_2$$

Figure3.6

In compressive sensing, the compression process of an image can be regarded as a process of feature extraction. The adoption of orthogonal complex Gaussian measurement matrices can effectively enhance the ability of feature extraction. When using the orthogonal complex Gaussian measurement matrix for compressive sampling, compared with the traditional real number measurement matrix, the complex measurement matrix can extract richer image feature information, which is equivalent to obtaining double the feature dimension, thereby demonstrating stronger performance in the subsequent signal detection link.

3.3.2.2 Half-tensor product

The Semi-Tensor Product (Cheng,et al.2007) (STP) is an extended matrix multiplication. It is a new operation between traditional matrix multiplication and Tensor Product multiplication, allowing two matrices with mismatched dimensions to perform multiplication operations and enabling effective compression of downsampling time-frequency graphs.

Suppose that:

$$\boldsymbol{X} \in R^{d\times n}, \quad \boldsymbol{Y} \in R^{p\times q}$$

Figure3.7

If n is a multiple of p or p is a multiple of n, then there is

$$\boldsymbol{Z} = \boldsymbol{X} \ltimes \boldsymbol{Y}$$

Figure3.8

$$\boldsymbol{Z} = \left\{ Z^{il} \right\} = Row_i\left(\boldsymbol{X}\right) \ltimes Col_l(\boldsymbol{Y})$$

Figure3.9

Let p be a multiple of n, that is, p=nt. Then the half-tensor product of matrices X and Y can be expressed as:

$$\boldsymbol{Z} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dn} \end{bmatrix} \ltimes \begin{bmatrix} y_{11} & \cdots & y_{1q} \\ \vdots & \ddots & \vdots \\ y_{p1} & \cdots & y_{pq} \end{bmatrix} = \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{d1} & \cdots & x_{dn} \end{bmatrix} \times \begin{bmatrix} \boldsymbol{Y}_{11} & \cdots & \boldsymbol{Y}_{1q} \\ \vdots & \ddots & \vdots \\ \boldsymbol{Y}_{n1} & \cdots & \boldsymbol{Y}_{nq} \end{bmatrix}$$

Pigure3.10

The half-tensor product compressive sensing model described in this section is defined as follows:

$$\boldsymbol{y} = \boldsymbol{\Phi} \ltimes TFR_r \ltimes \boldsymbol{\Phi}^{\mathrm{T}}$$

Figure3.11

3.3.2.3 Time-frequency graph downsampling

The time-frequency graph is calculated using the TFRSTFT algorithm. To ensure the consistency of the time-frequency graph and facilitate subsequent processing, when saving the time-frequency graph sample, the imresize function is called to adjust the size of the time-frequency graph to

$$N_{fh} \times N_{fh}\text{，}\quad N_{fh}=224$$

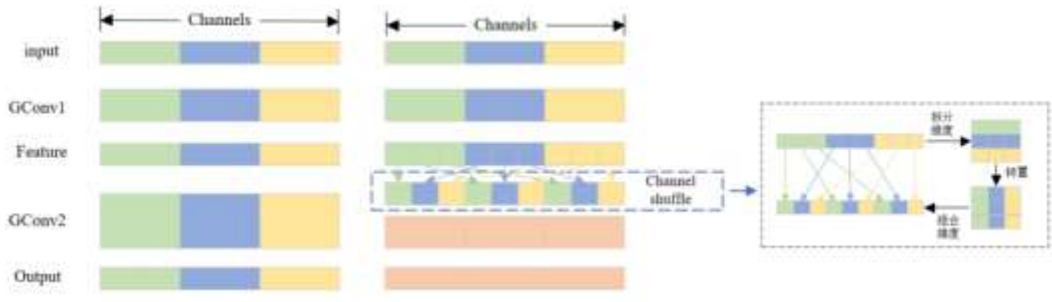
Firstly, the time-frequency graph is subjected to grayscale processing. Then, two-dimensional compression downsampling is performed on the grayscale time-frequency graph using Equation (3.11). The signal dimension after compressive sensing downsampling depends on the downsampling factor M. Specifically, the time-frequency graph is simultaneously reduced by the same multiple M in both dimensions. Therefore, the size of the time-frequency graph after compression and downsampling is

$$\left(N_{fh} / M\right) \times \left(N_{fh} / M\right)$$

By conducting downsampling in this way, not only is the dimension of the data effectively reduced, but also the main features of the signal are retained, providing high-quality input data for the training of deep learning models, thereby enhancing the training efficiency and processing speed of the models. This method is particularly effective when dealing with large-scale datasets, significantly reducing the consumption of computing resources.

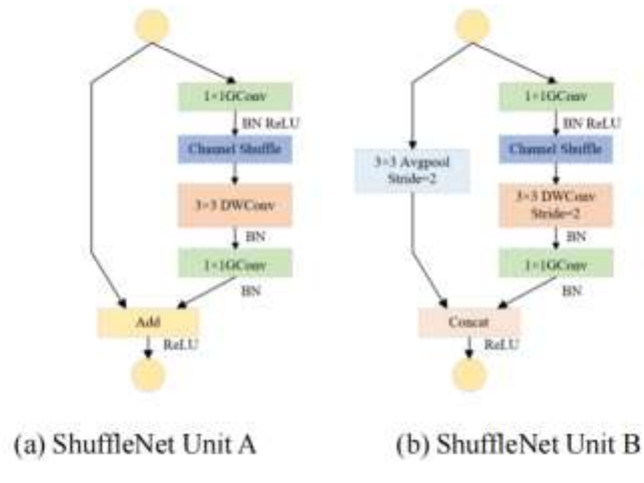
3.3.3 Network Structure

This chapter adopts a lightweight ShuffleNet network structure for the detection task of frequency-hopping signals. ShuffleNet is a lightweight deep learning model specifically designed for mobile and embedded devices. Its core idea lies in significantly reducing the number of model parameters without sacrificing model performance through Group Convolution and Channel Shuffle techniques. Group convolution can effectively reduce the computational complexity and parameter scale of convolution operations. However, this method also brings about the problem of limited information interaction among different groups. To solve this problem, ShuffleNet introduces the Channel Shuffle layer to enhance the feature interaction among channels. The specific Shuffle operation is shown in picture 3.1, where GConv1 and GConv2 represent two stages of grouped convolution operations, and Channel Shuffle represents the process of rearranging the channels.



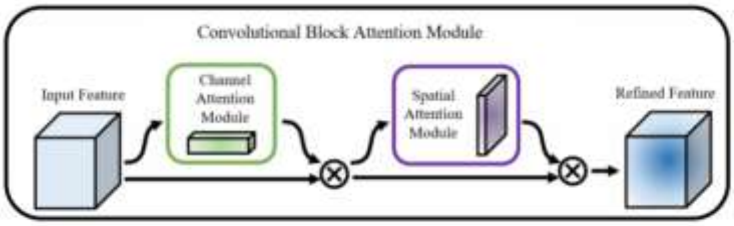
Picture 3.1 Shuffle Operation

ShuffleNet draws on the design concept of ResNet, starting from the basic ResNet bottleneck residual module and gradually evolving into ShuffleNet units. Subsequently, by stacking multiple ShuffleNet units, a complete ShuffleNet module is constructed.The ShuffleNet unit is shown in Picture 3.2.Picture 3.2(b) shows the main path using depthwise convolution, where the convolution kernel size is 3x3 and the step size is 2, in order to reduce the resolution of the feature map. This method can reduce the size of feature maps while maintaining computational efficiency, which is conducive to improving the processing speed and performance of the model.The bypass adds an AveragePooling layer, where the size of the pooling kernel is 3x3 and the step size is 2, synchronously reducing the resolution of the feature map with the depth convolution on the main path(Burel,et al.2001). To maintain the dimension unchanged, the concatenation operation was ultimately adopted instead of the addition operation to merge the feature maps from different channels. This not only retains the feature information obtained through different processing paths, but also effectively increases the number of channels of the feature map and enhances the learning ability of the model.



Picture3.2ShuffleNet unit

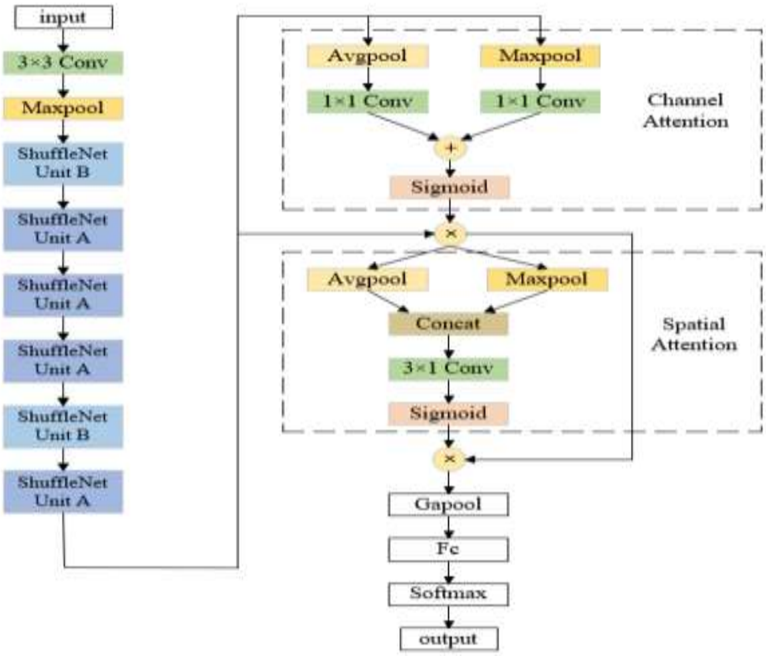
To further reduce the number of parameters, this chapter only uses the first six layers of the ShuffleNet network modules. The network composed of these first six layers of modules is denoted as shufflenet-6. Meanwhile, in order to maintain detection performance while reducing parameters, the Attention model CBAM (Convolutional Block Attention Module) is introduced in this chapter. As shown in Picture 3.3, CBAM is a lightweight convolutional attention module that can enhance the model's performance by focusing on important feature information without significantly increasing the computational burden. This combination not only helps to enhance the model's detection capability, but also ensures computational efficiency and the effective utilization of resources.



Picture 3.3 CBAM Module(woo,et al.2018)

CBAM consists of two sub-modules, namely CAM (Channel Attention Module) and SAM (Spatial Attention Module). CBAM features high efficiency and low resource consumption. Meanwhile, it has a flexible structure and can be conveniently integrated as a plugin into the existing network architecture. Among them, the CAM module, without changing the channel dimension, compresses the spatial dimension to highlight the significant information in the input features. The SAM module, while maintaining spatial resolution, reduces the channel dimension, thereby enhancing the expression of information related to the target position. By introducing the CBAM module into the neural network, the overall performance of the model can be effectively enhanced.

The network structure (ShuffleCBAM) proposed in this chapter is shown in Picture 3.4.



Picture 3.4 ShuffleCBAM Network Structure

3.3.4 Model Training and Testing

During the model training phase, the labeled dataset

$$D_{FH} = \{(s^{(1)}, I^{(1)}), (s^{(2)}, I^{(2)}), \dots, (s^{(I)}, I^{(I)})\}$$

is input into the proposed ShuffleCBAM network. The output of this network can be expressed as:

$$G_\delta(s^{(i)}) = [G_{\delta|H_1}(s^{(i)}), G_{\delta|H_0}(s^{(i)})]$$

Figure3.12

This chapter uses the cross-entropy function as the loss function to train the network.

$$\text{Loss} = -\frac{1}{N_B}\sum_{i=1}^{N_B}(l^{(i)}\log G_{\delta|H_1}(s^{(i)})) + (1-l^{(i)})\log(1-G_{\delta|H_1}(s^{(i)}))$$

$$\nabla\delta \leftarrow \frac{\partial \text{Loss}}{\partial \delta}$$

$$\delta \leftarrow optimizer(\delta, \nabla\delta)$$

Figure3.13

In the model inference stage, the existence of frequency-hopping signals is usually determined based on the following criteria:

$$H_1 : \eta > \gamma$$
$$H_0 : \eta \leq \gamma$$

Figure3.14

3.4 Analysis of Detection Performance of Computer Simulation Frequency-hopping Signals

3.4.1 Dataset Generation and Experimental Parameter Setting

In the experimental design stage, the frequency-hopping signal is generated first. The specific process is as follows: Generate a 16-bit random binary sequence, each consisting of 512 sampling points, and then perform BPSK modulation on it to obtain a modulated signal. The frequency range of 5 MHz is uniformly divided into 128 frequency points as the frequency set, and a frequency is randomly selected within each bit period. For each bit, generate the corresponding carrier signal and multiply it with the BPSK modulation signal of that bit to obtain the frequency-hopping signal corresponding to that bit. Through bit-by-bit processing, a complete frequency-hopping signal with a length of 8192 is finally synthesized. Next, noise addition and power normalization processing will be carried out. To simulate the actual channel environment, additive white Gaussian noise (AWGN) is introduced. The signal-to-noise ratio (SNR) setting range is from -30 dB to 0 dB, with a step size of 2 dB. The time-frequency graph of the noisy signal is generated by using the TFR-STFT method, and the time-frequency graph samples are saved. Under each signal-to-noise ratio condition, 500 samples are generated and divided into the training set, validation set and test set in a ratio of 2:1:2. It should be noted that the training set contains time-frequency graph samples of signals under different signal-to-noise ratio conditions, aiming to enhance the model's adaptability to various noise environments. Meanwhile, in addition to the frequency-hopping signal samples, the training set and the test set also contain an equal amount of pure noise time-frequency graph samples to enhance the authenticity and challenge of the classification task.

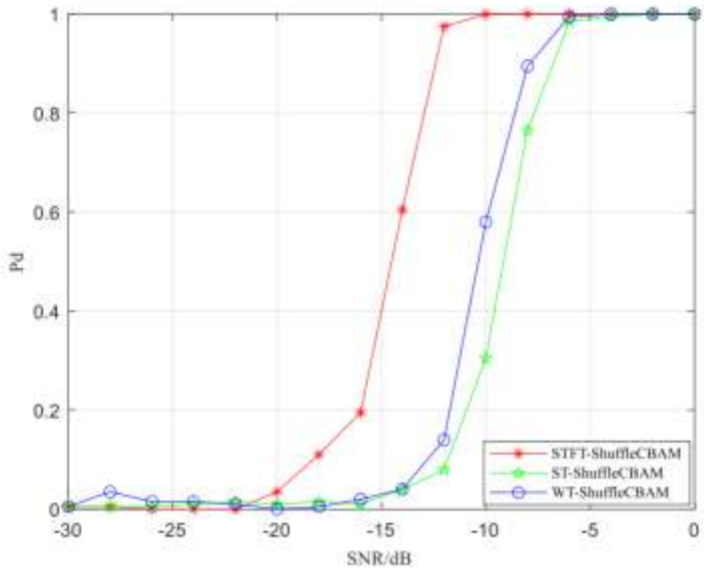
3.4.2 Experimental Results and Performance Analysis

Based on the understanding of the STFT-CS-ShuffleCBAM algorithm, a strict simulation experiment was carried out next to complete the detection of the performance of frequency-hopping signal detection, in order to determine whether the detection of the algorithm is accurate and its robustness in the noise environment with differences.

(1) The influence of time-frequency transformation on algorithm performance

To further identify whether the short-time Fourier transform in the STFT-ShuffleCBAM algorithm is effective, this experiment rejects the introduction of compressive sensing (CS) and instead uses the S-transform and wavelet transform to generate time-frequency graphs respectively. During this process, the ST-ShuffleCBAM and WT-ShuffleCBAM (Gu, et al.2023) algorithms were used. Among them, the scale is set to 512 in the wavelet transform; In STFT, a Hamming window of length 512 is adopted. In a Gaussian white noise environment, the detection probability curves of the three methods - STFT-ShuffleCBAM, ST-ShuffleCBAM and WT-ShuffleCBAM - for frequential-hopping signals are shown in

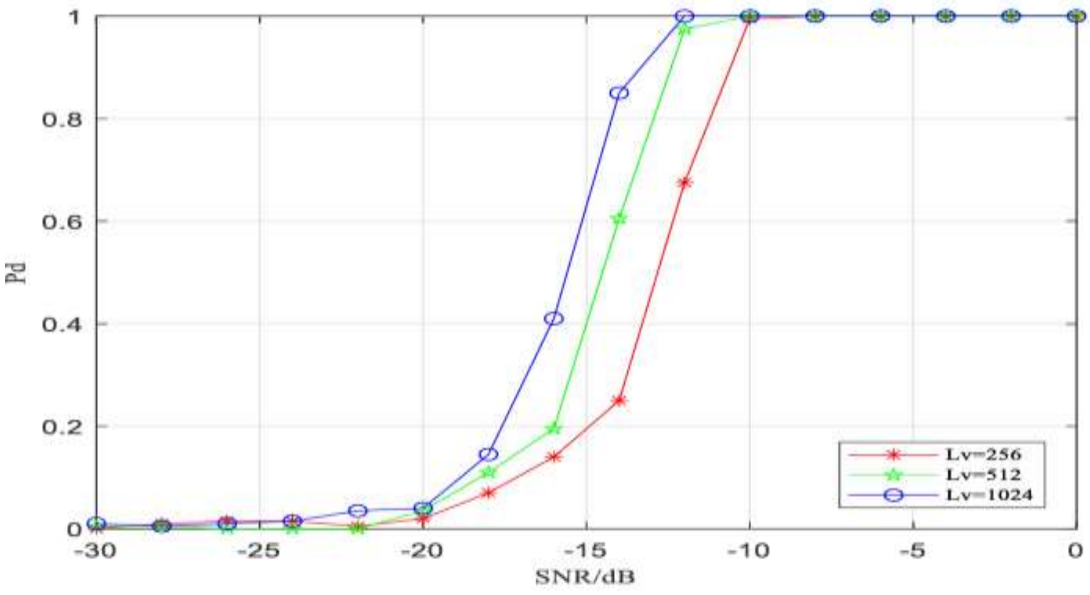
Picture 3.7. It can be seen from the figure that STFT-ShuffleCBAM demonstrates the best performance in frequency-hopping signal detection. When the signal-to-noise ratio (SNR) is -10 dB, the detection probabilities of ST-ShuffleCBAM, WT-ShuffleCBAM and STFT-ShuffleCBAM are 0.305, 0.58 and 1 respectively. It can be seen from this that the time-frequency analysis method based on STFT significantly outperforms the other two methods in detection performance. Therefore, STFT is selected as the main time-frequency analysis means in the subsequent research of this chapter.



Picture 3.7 Detection performance of different time-frequency transformation methods

(2) The influence of window length on algorithm performance

Under the conditions where the window lengths (Lv) are set to 256, 512 and 1024 respectively, the detection performance of the STFT-ShuffleCBAM method for frequency-hopping signals is shown in Figure 3.8. It can be seen from the figure that as the window length increases, the detection effect of STFT-ShuffleCBAM gradually improves, but at the same time, the computational complexity also increases accordingly. Therefore, on the basis of weighing the detection performance against the computational cost, a window length of 512 was selected for further analysis in the subsequent experiments.



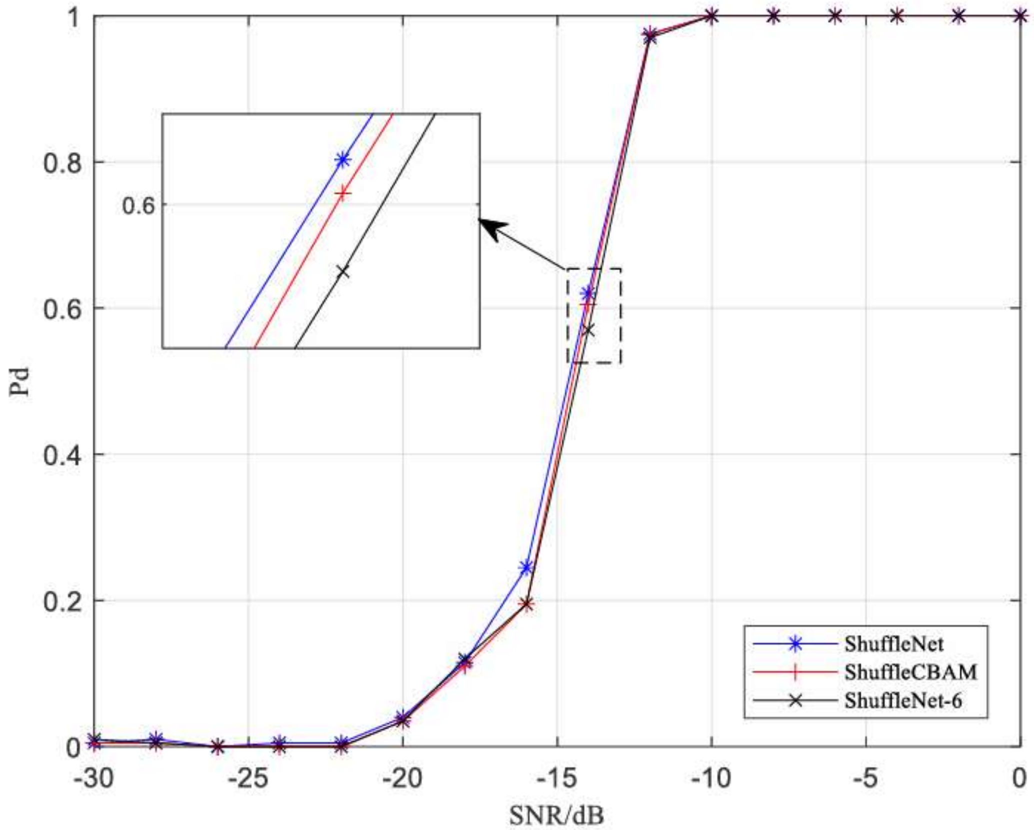
Picture 5.8 Detection performance under different window lengths

(3) Effective performance analysis of the CBAM module

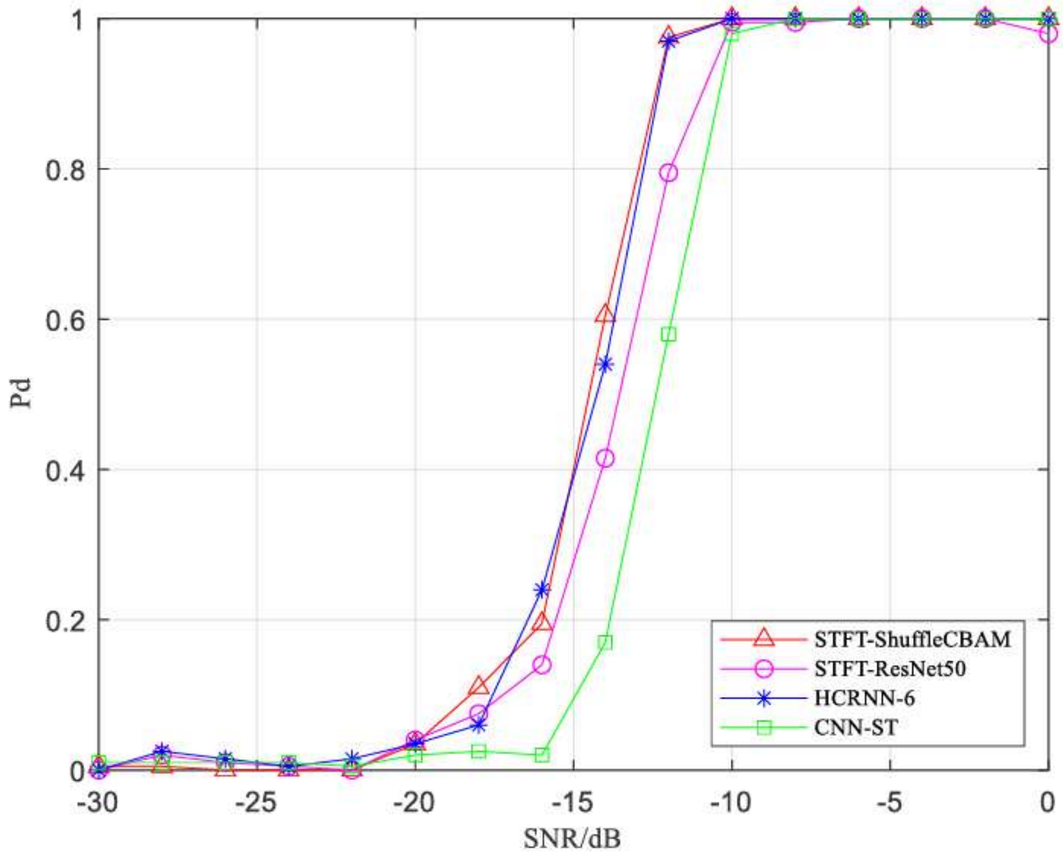
To evaluate the effectiveness of the CBAM attention mechanism in signal detection tasks, this experiment compared three different deep learning network structures: the ShuffleCBAM model proposed in this paper, the standard ShuffleNet(Zhang,et al.2018), and the simplified Shufflenet-6. The designed ShuffleCBAM model has made structural innovations. On the basis of retaining the first six units of the original ShuffleNet used for basic feature extraction, the subsequent ten ShuffleNet units are replaced with CBAM attention modules. Thus, a lightweight network architecture with adaptive feature selection capability is constructed. Figure 3.9 shows the comparison results of the detection performance of the proposed methods when different network models are adopted without introducing compressive sensing technology. As can be seen from the figure, when the signal-to-noise ratio (SNR) is -14 dB, the deep architecture based on ShuffleCBAM has a detection probability approximately 5% higher than that of the model based on ShuffleNet-6. The standard ShuffleNet uses all 16 ShuffleNet units and has the best detection performance, but it also brings the highest model complexity. In contrast, ShuffleCBAM significantly reduces the computational complexity of the model while maintaining high detection performance. Although ShuffleNet-6 has the most streamlined model, its detection effect is the least satisfactory. In conclusion, ShuffleCBAM has successfully achieved a good trade-off between detection accuracy and computational efficiency.

(4) Performance comparison of different algorithms

This experiment aims to evaluate the performance of different algorithms in the frequency-hopping signal detection task without introducing a compressed sampling module. The STFT-ShuffleCBAM detection algorithm proposed in this chapter was compared and analyzed with the STFT-ResNet50 algorithm based on ResNet50, as well as the methods proposed in references(Li,et al.2022). Among them, the algorithm proposed in reference generates time-frequency images by using short-time Fourier transforms with window lengths of 32, 64, 128, 256, 512 and 1024 respectively, and inputs them into a hybrid convolutional neural network/recurrent neural network (CNN/RNN) structure. This method is abbreviated as HCRNN-6; The method adopted in reference is to first extract the time-frequency matrix through S-transformation and then input it into CNN for detection, which is simply referred to as CNN-ST. Figure 3.10 shows the frequency hopping signal detection probability curves of four algorithms, namely STFT-ShuffleCBAM, STFT-ResNet50, HCRNN-6 and CNN-ST, under different signal-to-noise ratio conditions. It can be seen from the figure that the detection performance of STFT-ShuffleCBAM is comparable to that of HCRNN-6, and is significantly better than the STFT-ResNet50 and CNN-ST methods. However, while HCRNN-6 achieves high detection performance, it also brings a large amount of time-frequency data input and higher model complexity. Comprehensively considering the detection accuracy and computing resource consumption, the STFT-ShuffleCBAM algorithm proposed in this chapter has a better balance in overall performance and shows stronger application advantages.



Picture 3.9 Effective detection performance of the CBAM module



Picture 3.10 Detection performance of different algorithms

(5) The influence of measurement matrix and product methods on algorithm performance

In this experiment, three different types of measurement matrices are respectively adopted to construct

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

the compressive sensing model: random orthogonal complex Gaussian measurement matrix, random orthogonalized Gaussian measurement matrix, and ordinary random Gaussian measurement matrix. Corresponding to the proposed STFT-CS-ShuffleCBAM detection algorithm, they are respectively abbreviated as STFT-CS-Shufflecbam-ROCG, STft-CS-Shufflecbam-ROG and STFT-CS-Shufflecbam-RG. In addition, two matrix Product methods were considered: conventional matrix product and Semi-Tensor product (STP). Set the downsampling factor M to 4. Figure 3.11 shows the detection probability curves of frequency-hopping signals under the above different configurations. It can be seen from the figure that when using the conventional matrix product, the STFT-CS-ShuffleCBAM-RG method almost completely loses the key feature information in the original time-frequency graph after completing two-dimensional compression downsampling, resulting in the inability to effectively detect frequency-hopping signals. In contrast, the STFT-CS-ShuffleCBAM-ROCG and STFT-CS-ShuffleCBAM-ROG methods retain most of the characteristics of the frequency-hopping signal during the compression process and exhibit relatively ideal detection performance. To further reduce the storage requirements and computational complexity of the measurement matrix, the half-tensor product (STP) is introduced as the product method. Because STP can significantly reduce the scale of the required measurement matrix while maintaining the same downsampling ratio, it is particularly suitable for resource-constrained application scenarios. Considering that there is a large information loss in the compression process of the STFT-CS-ShuffleCBAM-RG method, experiments are only conducted on the combination of the STFT-CS-ShuffleCBAM-ROCG and STFT-CS-ShuffleCBAM-ROG methods with STP. They are respectively named STFT-CS-ShuffleCBAM-ROCG-STP and STFT-CS-ShuffleCBAM-ROG-STP. The experimental results show that both of these two STPT-based methods exhibit excellent detection performance, among which STFT-CS-ShuffleCBAM-ROCG-STP stands out the most. Compared with STFT-CS-ShuffleCBAM-ROCG without adopting STP, this method not only significantly reduces the storage space and computational overhead of the measurement matrix, but also can still maintain a relatively high detection accuracy. Therefore, in the subsequent chapters (as shown in Figure 3.2), if the method of using a random orthogonal complex Gaussian measurement matrix combined with a half-tensor product is involved, it is uniformly referred to as STFT-CS-ShuffleCBAM.

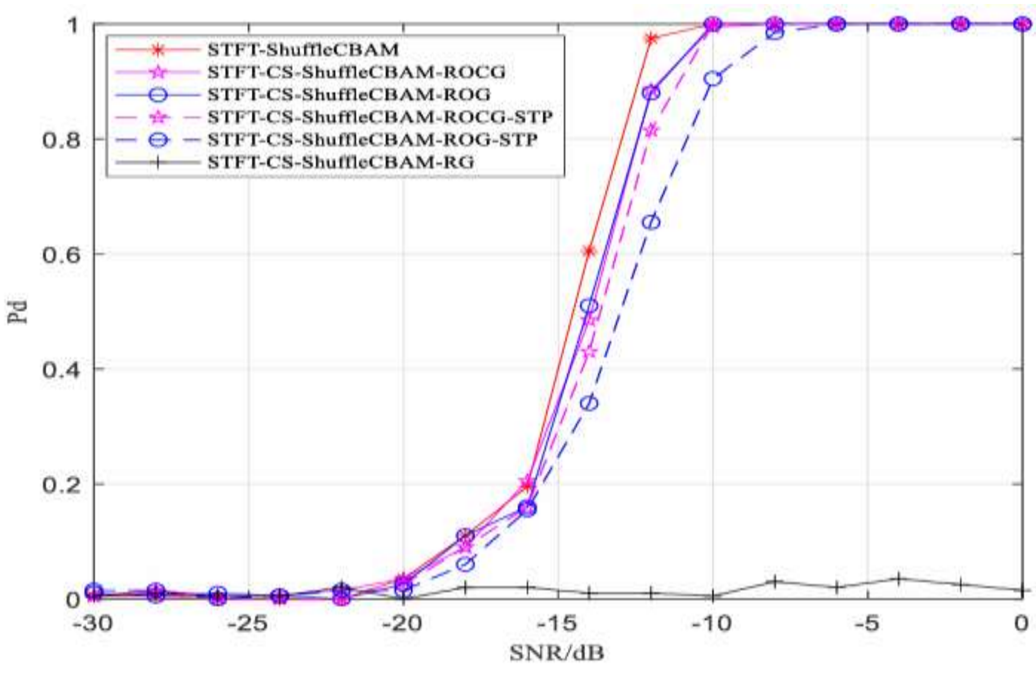


Figure 3.11 Detection performance under different measurement matrices and different product methods

(6) The influence of downsampling factors on algorithm performance

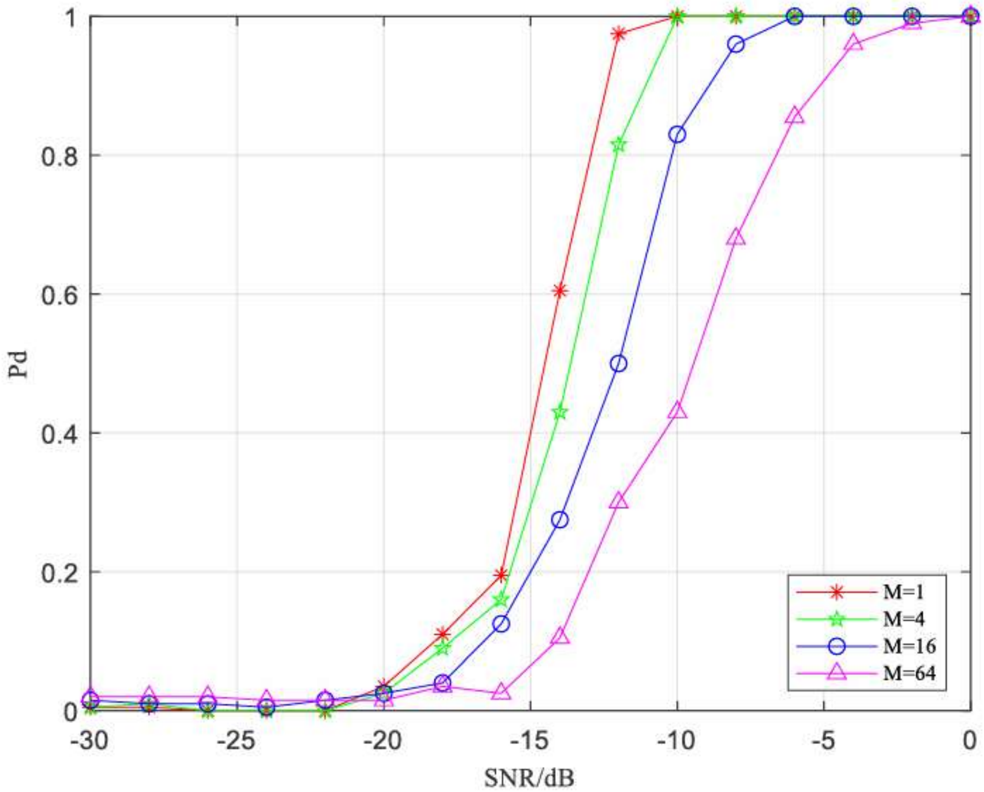
The STFT-CS-ShuffleCBAM algorithm was evaluated by a $2^m \times 2^m$ random orthogonal complex Gaussian measurement matrix, and the sampling reduction factors M were set to 1, 4, 16 and 64 respectively(Wei,et al.2022). Figure 5.12 shows the detection performance of this method for frequency-hopping signals under different downsampling factors. It can be observed from Figure 5.12 that as the downsampling factor M increases and the corresponding time-frequency graph dimension decreases, the detection performance of the algorithm shows a downward trend. Specifically, when the downsampling factor is increased, two-dimensional compression processing of the time-frequency graph using the measurement matrix will lead to partial loss of the features of the time-frequency graph. This feature loss makes it difficult for ShuffleCBAM networks to learn the complete feature representation of frequency-hopping signals from the compressed time-frequency graph, thereby leading to a decline in overall detection performance.

The conclusion that can be drawn is that the adoption of an appropriate downsampling factor must be in harmony with the control of the algorithm's detection performance and its computational complexity. The size of the sampling factor is inversely proportional to the detection accuracy, but for the computational burden, a smaller factor is more perfect. The larger the downsampling factor is, the more it can help reduce computational complexity, but it will sacrifice detection performance.

3.5 Performance Analysis of Detection for Actual Frequency-Hopping Signals

In this study, a special signal detection and acquisition system is constructed, so as to control the detection performance of the STFT-CS-ShuffleCBAM algorithm in frequency hopping signal through the signal transmission source and a signal receiving device of the handheld radio. Among them, the center frequency of the handheld radio is set to 38.5MHz, the frequency hopping bandwidth is 6MHz, the frequency setting size is 256 channels, and the frequency hopping rate is 1000 hops per second. The signal is captured by the USRP-N310 receiving device with a center frequency of 38.5MHz and a sampling frequency of 12.5MHz. The complete acquisition of the frequency hopping signal and the background noise signal with a length of 131072 was completed, and the noise with different signal-to-noise ratios was added on the basis of the original frequency hopping signal.

Consistent with the signal processing flow in the simulation environment, all signals have undergone power normalization processing, and then the corresponding time-frequency graphs are generated using the TFRSTFT algorithm. Ultimately, for each set signal-to-noise ratio level, we obtained 500 independent signal samples, of which 300 were used to fine-tune the network model, and the remaining 200 were reserved for the testing phase. It is worth noting that the number of noise samples matches that of signal samples, ensuring the balance and representativeness of the dataset. Such an experimental design enables us to comprehensively and precisely evaluate the effectiveness of the STFT-CS-ShuffleCBAM algorithm under real-world conditions and its robustness to noise. In addition, by fine-tuning and verifying the network model, the practical application potential and reliability of the algorithm are also guaranteed.



picture 3.12 Detection performance under different downsampling factors

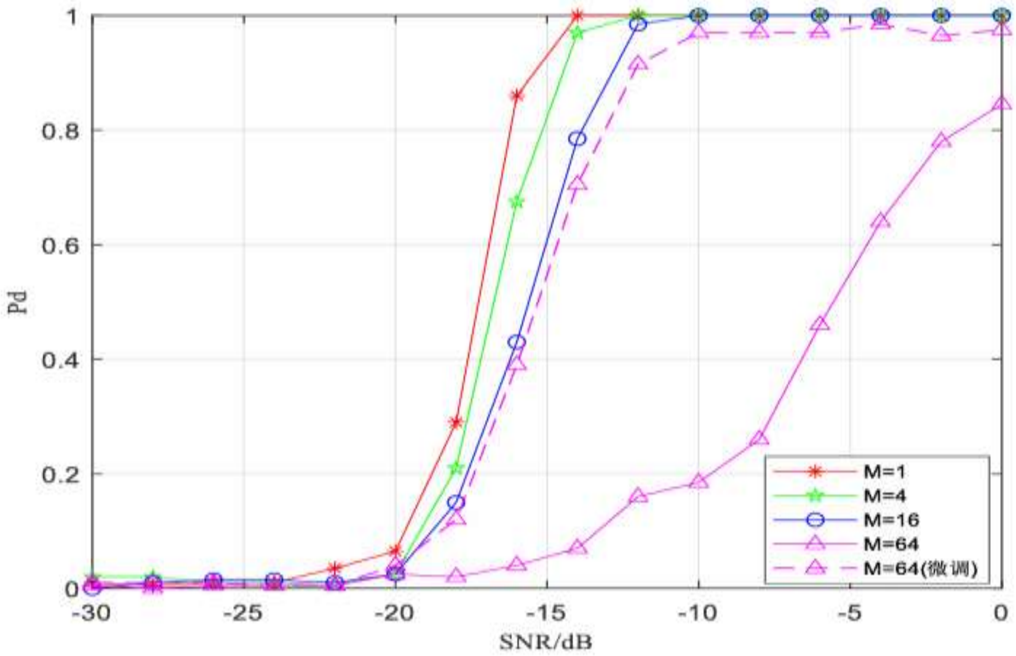
(1) The influence of downsampling factors on algorithm performance

To further verify the effectiveness of the STFT-CS-ShuffleCBAM algorithm in the detection of actual frequency-hopping signals, this paper uses the network model trained based on computer simulation data in Section 3.4.2 (6) to test the actually collected frequency-hopping signals. Under the conditions of setting the false alarm probability P=0.01 and the downsampling factors M to 1, 4, 16 and 64 respectively, Figure 3.13 shows the detection performance of this method at different M values. In addition, the figure also presents the detection results after fine-tuning the network using the actual collected data samples under the condition of M=64. It can be seen from the experimental results that as the downsampling factor M increases and the dimension of the time-frequency graph decreases, the overall detection performance of the algorithm shows a downward trend. When M=4 and M=16, the performance degradation is relatively small. However, when M=64, due to the loss of excessive feature information during the compression process, the detection performance deteriorates significantly. To alleviate this problem, the network was fine-tuned using the actually collected samples. During the fine-tuning process, the learning rate was set at 0.00001 and the training rounds were set at 2. The results show that after fine-tuning, the detection ability of the STFT-CS-ShuffleCBAM method for actual signals has been significantly improved. The above experimental results show that the STFT-CS-ShuffleCBAM algorithm proposed in this chapter not only has a good initial detection ability when facing the radio frequency hopping signals in the real environment, but also can effectively adapt to the changes in the actual signal distribution through fine-tuning with a small amount of real data, demonstrating strong robustness and adaptability.

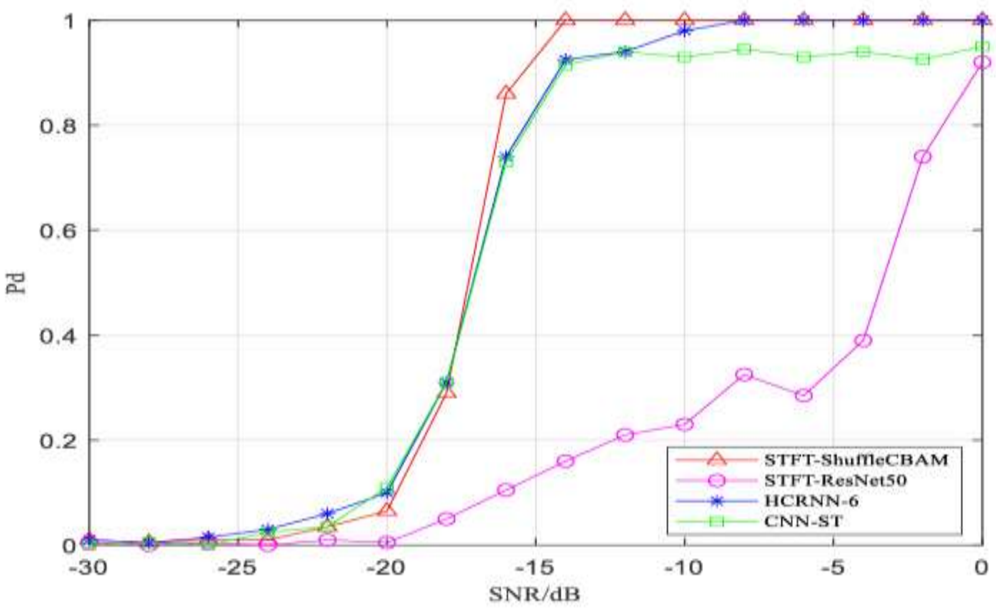
(2)Performance comparison of different algorithms.

To further verify the generalization ability of the method proposed in this chapter in the real environment, the network model trained based on computer simulation data in Section 3.4.2 (4) is adopted to detect the actually collected frequency-hopping signals. Figure 3.14 shows the comparison results of detection performance between the STFT-ShuffleCBAM algorithm and several typical algorithms such as STFT-

ResNet50, CNN-ST, and HCRNN-6 under different signal-to-noise ratios (SNRS) under the condition of false alarm probability p=0.01. It can be seen from the figure that STFT-ShuffleCBAM demonstrates significantly superior performance compared to other methods in actual signal detection tasks. Taking the low signal-to-noise ratio condition with SNR of -14 dB as an example, the detection probabilities of STFT-ResNet50, CNN-ST and HCRNN-6 are 0.16, 0.9 and 0.92 respectively. The STFT-ShuffleCBAM method proposed in this chapter achieves a perfect detection probability - that is, 1. Furthermore, the HCRNN-6 method achieves a detection probability of 1 only when the SNR is -8 dB. However, within the entire test range of SNR from -30 dB to 0 dB, neither STFT-ResNet50 nor CNN-ST can achieve complete detection performance. In conclusion, the STFT-ShuffleCBAM algorithm not only has a higher detection accuracy in actual frequency-hopping signal detection tasks, but also maintains excellent robustness in low signal-to-noise ratio environments, demonstrating superior overall performance compared to existing methods.



Picture 3.13 Detection performance under different downsampling factors



Picture 3.14 Comparison of detection performance of different algorithms

3.6 Complexity Analysis

The time complexity and space complexity of the STFT-CS-ShuffleCBAM algorithm proposed in this chapter are elaborated in detail in Tables 3.3 and 3.4 respectively. Table 3.3 compares the differences in computational complexity between the STFT-ShuffleCBAM algorithm and existing algorithms (including HCRNN-6 in Reference and CNN-ST in reference), evaluated based on the number of floating-point operations (FLOPs) and memory usage.

Between HCRNN-6, CNN-ST and STFT-ShuffleCBAM algorithms, there is no doubt that the third option performs better in terms of time complexity and spatial complexity. Specifically, Table 3.4 shows the states of different stages of the computational complexity of the STFT-CS-ShuffleCBAM algorithm when downsampling, which shows that it is not difficult to achieve high detection performance and optimize the downsampling process at different stages when taking into account the computational complexity. In short, the STFT-CS-ShuffleCBAM algorithm can significantly reduce the computational burden and adapt to practical application scenarios, which is more efficient and feasible than the other two methods (Lim, 2017). Especially when you want to reconcile time and space complexity and avoid the inconvenience of resource-constrained environments, the STFT-CS-ShuffleCBAM algorithm is undoubtedly superior to traditional methods.

Table 3.3 Computational Complexity of the Three Algorithms

Algorithm	Time complexity	Space complexity
STFT-ShuffleCBAM	5.3727×10^7	7.7740×10^5
HCRNN-6	1.9572×10^9	5.3315×10^7
CNN-ST	6.7740×10^8	3.0540×10^6

Table 3.4 Computational Complexity of STFT-CS-ShuffleCBAM Method

Downsampling factor M	Time complexity	Space complexity
1	5.3727×10^7	7.7440×10^5
4	1.4110×10^7	2.4777×10^5
16	3.7157×10^6	1.1606×10^5
64	1.1040×10^6	8.3130×10^4

Whether it is the computational burden in terms of time or space, it will be determined by the dimension of the input time-frequency matrix. Specifically, if the dimension of the time-frequency matrix is reduced to 50% of the original, the algorithm is liberated in terms of time complexity, effectively reducing it by three times. The result when M=4 in the sampling factor can be referred to below. At the level of space complexity, it is also inversely proportional to the dimension of the input time-frequency matrix. In such circumstances, compressed sampling can not only reduce the computational workload of most models but also release more storage resources, which is particularly convenient for application scenarios with limited resources. In this case, to update the computing rate and effect without compromising the detection performance, it is advisable to consider modifying the size of the input time-frequency graph, ultimately achieving a dual enhancement of resource conservation and work efficiency.

CONCLUSION

In order to adapt to the situation of limited resources and insufficient hardware capabilities in real-world application scenarios, this study proposes a STFT-CS-ShuffleCBAM frequency-hopping signal detection algorithm. By combining the compressive sensing of time-frequency graphs with deep learning technology, it achieves the improvement of computing efficiency and the enhancement of resource efficiency. Firstly, a random orthogonal complex Gaussian measurement matrix was designed. By using two-dimensional compressive sensing and half-tensor product techniques, the size of the time-frequency graph was effectively reduced, thereby significantly alleviating the computational burden. Secondly, the optimized

ShuffleCBAM network model is used to process the downsampled time-frequency graph. Even under low signal-to-noise ratio conditions, this network can efficiently detect frequency-hopping signals. Finally, the experimental results were given. The experimental results show that the algorithm demonstrates remarkable effectiveness both in the simulated environment and in the actual scenarios. Compared with the existing algorithms, this algorithm not only improves the detection accuracy but also has a lower computational complexity.

The sparse characteristics of frequency-hopping signals in the time-frequency domain are rationally applied, and both the preservation of key features and the reduction of the dimension of input data are achieved simultaneously. Random orthogonal complex Gaussian measurement matrices are used, accompanied by compressed sampling of short-time frequency graphs. The results show that the performance of this matrix in feature extraction is significantly better than that of the traditional matrix. Combined with the half-tensor product (STP) technology, the compression process is enhanced and optimized. Therefore, even in scenarios where resources are limited and large-scale datasets need to be dealt with, this model can be used smoothly. Secondly, a lightweight neural network architecture called ShuffleCBAM was designed. Experiments show that at different levels of signal-to-noise ratio (SNR), ShuffleCBAM is always superior to HCRNN-6 and CNN-ST. This result has also been verified in many ways, mainly based on computer-generated frequency-hopping signals to complete a large number of simulation experiments. Experiments show that when the signal-to-noise ratio is set at -14 dB, the detection probability of STFT-CS-ShuffleCBAM achieves perfect accuracy, while the performance of other algorithms is significantly lower (Yuan,et al.2008). In addition, the algorithm was tested on the actually collected frequency-hopping signals. A dedicated signal acquisition system was constructed, including a handheld radio configured for transmitting frequency-hopping signals and a USRP-N310 receiving device for capturing these signals. STFT-CS-ShuffleCBAM still maintains a high detection accuracy when processing actual signals, which proves its feasibility and robustness in practical applications. The computational complexity and storage complexity of the algorithm were analyzed in detail. In conclusion, the STFT-CS-ShuffleCBAM algorithm has multiple advantages in practical applications, achieving high detection accuracy and reducing computational complexity, making it an ideal solution for real-time signal processing applications such as wireless cognitive radio and spectrum monitoring systems. In addition, the flexible adjustment of the downsampling factor enables users to balance detection performance and computational efficiency according to specific application requirements.

REFERENCES

1. Auger, F., Flandrin, P., Gonçalvès, P., & others. (1996). Time-frequency toolbox. CNRS France-Rice University.
2. Burel, G., Bouder, C., & Berder, O. (2001). Detection of direct sequence spread spectrum transmissions without prior knowledge. In Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 2001) (Vol. 1, pp. 236–239). IEEE. https://doi.org/10.1109/GLOCOM.2001.965854
3. Cheng, D., Qi, H., & Xue, A. (2007). A survey on semi-tensor product of matrices. Journal of Systems Science and Complexity, 20(2), 304–322. https://doi.org/10.1007/s11424-007-9025-5
4. Fargues, M. P., Overdyk, H. F., & Hippenstiel, R. (1997). Wavelet-based detection of frequency hopping signals. In Conference Record of the 31st Asilomar Conference on Signals,Systems and Computers(Vol.1,pp.515–519).IEEE. https://doi.org/10.1109/ACSSC.1997.680439
5. Gu, H., Liu, X., Xu, L., & others. (2023). Period estimation of spread spectrum codes based on ResNet. Sensors, 23(15), 7002. https://doi.org/10.3390/s23157002
6. Li, C., Zhao, Z., & Chen, Y. (2022). Detection algorithm of frequency hopping signals based on S transform and deep learning. In Proceedings of the 16th IEEE International Conference on Signal Processing (ICSP) (Vol. 1, pp. 310–313). IEEE. https://doi.org/10.1109/ICSP49669.2022.9757821
7. Lim, C. H. (2017). Energy detection based spectrum sensing for radar signals in the presence of noise power uncertainty. The

Journal of Korean Institute of Communications and Information Sciences, 42(5), 982–984. https://doi.org/10.7840/kics.2017.42.5.982

8. Park, D. H., Jeon, M. W., Shin, D. M., & others. (2023). LPI radar detection based on deep learning approach with periodic autocorrelation function. Sensors, 23(20), 8564. https://doi.org/10.3390/s23208564

9. Urkowitz, H. (1967). Energy detection of unknown deterministic signals. Proceedings of the IEEE, 55(4), 523–531. https://doi.org/10.1109/PROC.1967.5573

10. Wang, Y. (2024). Research on performance optimization of image compressed sensing under vehicle channel conditions (Master's thesis). Zhejiang University of Science and Technology.

11. Wei, F., Zheng, S., Zhou, X., & others. (2022). Detection of direct sequence spread spectrum signals based on deep learning. IEEE Transactions on Cognitive Communications and Networking, 8(3), 1399–1410. https://doi.org/10.1109/TCCN.2022.3169051

12. Woo, S., Park, J., Lee, J. Y., & others. (2018). CBAM: Convolutional block attention module. In V. Ferrari, M. Hebert, C. Sminchisescu, & Y. Weiss (Eds.), roceedings of the European Conference on Computer Vision (ECCV) (pp. 3–19). Springer. https://doi.org/10.1007/978-3-030-01234-2_1

13. Yuan, Y., Mei, W., Wu, S., & others. (2008). Hop period estimation for frequency hopping signals based on Hilbert-Huang transform. In Proceedings of the 2008 Congress on Image and Signal Processing (Vol. 5, pp. 452–455). IEEE. https://doi.org/10.1109/CISP.2008.232

14. Zhang, X., Zhou, X., Lin, M., & others. (2018). ShuffleNet: An extremely efficient convolutional neural network for mobile devices. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6848–6856). IEEE. https://doi.org/10.1109/CVPR.2018.00716

Information

This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions

The manuscript has a single author who was solely responsible for the research design, data collection and analysis, and the drafting and revision of the manuscript.

Conflict of Interest Statement

The author declares no conflicts of interest.

Ethical Statement

This study adhered to all relevant ethical standards for academic research. Where applicable, any research involving humans or animals was conducted in accordance with ethical guidelines.