



# Gravitational Wave Detection Based on Squeeze-and-excitation Shrinkage Networks and Multiple Detector Coherent SNR

Rui-Qing Yan<sup>1</sup>, Wei Liu<sup>2</sup>, Zong-Yao Yin<sup>1</sup>, Rong Ma<sup>1</sup>, Si-Ying Chen<sup>1</sup>, Dan Hu<sup>3</sup>, Dan Wu<sup>4</sup>, and Xian-Chuan Yu<sup>1</sup>

<sup>1</sup>School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China; [yuxianchuan@163.com](mailto:yuxianchuan@163.com), [yanruiqing@mail.bnu.edu.cn](mailto:yanruiqing@mail.bnu.edu.cn)

<sup>2</sup>National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences; Beijing 100190, China, [liuwei@ia.ac.cn](mailto:liuwei@ia.ac.cn)

<sup>3</sup>Department of Radiology and BRIC, University of North Carolina at Chapel Hill, Chapel Hill, NC 27599 United States of America; [hd@bnu.edu.cn](mailto:hd@bnu.edu.cn)

<sup>4</sup>National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; [wudan@bao.ac.cn](mailto:wudan@bao.ac.cn)

Received 2022 May 14; revised 2022 July 14; accepted 2022 July 22; published 2022 October 12

## Abstract

Deep learning techniques have been applied to the detection of gravitational wave signals in the past few years. Most existing methods focus on the data obtained by a single detector. However, the signal-to-noise ratio (SNR) of gravitational wave signals in a single detector is pretty low, making it hard for deep neural networks to learn effective features. Therefore, how to use the observation signals obtained by multiple detectors in deep learning methods is a serious issue. We simulate binary neutron star signals from multiple detectors, including the Advanced LIGO and Virgo detectors. We calculate coherent SNR of multiple detectors using a fully coherent all-sky search method and obtain the coherent SNR data required for our proposed deep learning method. Inspired by the principle of attention network Squeeze-and-Excitation Networks (SENet) and the soft thresholding shrinkage function, we propose a novel Squeeze-and-Excitation Shrinkage (SES) module to better extract effective features. Then we use this module to establish a gravitational wave squeeze-and-excitation shrinkage network (GW-SESNet) detection model. We train and validate the performance of our model on the coherent SNR data set. Our model obtains satisfactory classification accuracy and can excellently complete the task of gravitational wave detection.

*Key words:* methods: data analysis – methods: statistical – gravitational waves

## 1. Introduction

As an important prediction in general relativity, gravitational waves are one of the important frontier fields of contemporary physics. Since the interaction between gravitational waves and matter is very weak during the propagation process, matter does not easily change gravitational waves. Thus, gravitational waves can be used as a carrier to transmit useful astronomical information, which is of great significance for people to explore the vast universe (Saulson 1994). In 2016, humans first discovered the gravitational wave signal generated by the merger of two black holes detected by LIGO (Aasi et al. 2015), which currently has found multiple gravitational wave signals generated by the merger of double black holes (Abbott et al. 2016b, 2016a). The discovery of gravitational waves has ushered in a new era of gravitational wave astronomy, and is also promoting the development of multi-messenger astronomy. However, the signal of gravitational waves is very weak, which makes gravitational wave astronomical research need more sensitive detectors and detection technologies, as well as new data mining and analysis methods in the field of information science.

Matched filtering is the most effective traditional method to detect gravitational wave signals (Szabo et al. 2011; Taracchini

et al. 2014). This method matches the observed signal with a large number of theoretical template libraries, and then judges whether the observed signal is a gravitational wave signal according to the matching result. Each parameterized waveform template in the template library corresponds to a set of parameters, which includes star mass and radius. However, even through the template matching method, we will still get a massive amount of gravitational wave signal candidates, and the detection is very slow and consumes a lot of computing resources.

Recently, deep learning methods have performed prominently in many fields, and some of them have been applied to gravitational wave detection (Beheshtipour & Papa 2020; Wang et al. 2020; Wei & Huerta 2021). There are two ways of implementing deep learning methods in detecting gravitational waves. One is to directly detect the time series gravitational wave signals. Gabbard et al. (2018) utilized a convolutional neural network model to detect gravitational wave time series data. Its performance is comparable to that of the matched filtering method under the same data set. This study shows that deep learning methods can extract the features of gravitational wave time series signals. Chatterjee et al. (2019) used the whitened strain values as the input of the artificial neural

network (ANN) model to extract the features of a gravitational wave signal. Another way to study gravitational waves relying on deep learning is to process image data. When the detector detects the gravitational wave signal, it will detect a transient non-Gaussian noise signal with a complex shape. This kind of signal is also called a “glitch,” which can interfere with the detection of gravitational wave signals, so some scholars have also studied the detection of gravitational wave glitches. Razzano & Cuoco (2018) apply image data as the input of a convolutional neural network to achieve efficient classification of glitches on simulated data sets. Colgan et al. (2020) proposed a deep learning method that considers the complete environment situation and the detector’s data stream to identify glitches, which can reduce the false alarm rate of gravitational wave detectors. George et al. (2017) proposed the transfer learning method to classify the spectral images of false gravitational wave time-series signals by using a pre-training model, and showed good results in identifying different categories of false signals. George & Huerta (2018) again proposed employing convolutional neural networks to detect real gravitational wave signals obtained by multiple detectors such as LIGO, and to estimate the parameters of gravitational wave events. This method demonstrates for the first time that the deep learning method can detect and estimate the parameters of real gravitational wave data.

From these studies using deep learning methods to detect gravitational waves, it can be found that most of the methods are for data obtained by a single detector. However, as a weak signal, the signal-to-noise ratio (SNR) of gravitational waves of a single detector is very low, and it is difficult for deep learning methods to learn effective features. Nowadays, an increasing number of gravitational wave detectors are currently being built or have been built, such as Virgo (Acernese et al. 2014), GEO600 (Willke et al. 2002) and KAGRA (Aso et al. 2013). Coherent multiple detector observation and multi-messenger gravitational wave signal detection are a trend of future development. The benefit of coherent observation is that multiple detectors are coupled with each other, which can improve the SNR of the observation signal, which we call coherent SNR (Macleod et al. 2016). However, detecting gravitational waves using observation signals from multiple detectors in deep learning methods is a challenge. Therefore, we propose a deep learning method to solve the above problems by utilizing the observation signals obtained by multiple detectors.

Based on the above analysis, we first simulate binary neutron star (BNS) signals from multiple detectors, including the Advanced LIGO and Virgo detectors. Second, the SNR is improved by calculating the coherent SNR with multiple detectors using a fully coherent all-sky search method. The method of coherent observation can reduce the interference of noise in the signal, thereby improving the SNR of the data. For the obtained gravitational wave coherent SNR data, we design

a one-dimensional convolutional neural network classification model. In order to better learn the features of coherent SNR data, inspired by the attention mechanism network Squeeze-and-Excitation Networks (SENet) (Hu et al. 2018) and the idea of soft thresholding shrinkage function (Donoho & Johnstone 1994; Donoho 1995), we propose a novel feature extraction Squeeze-and-Excitation Shrinkage (SES) module. In this novel SES module, we automatically learn the importance of feature map channels through a small neural network, and then use soft thresholding as a shrinkage function to eliminate noise features, so as to better extract the effective features of gravitational wave signals. Then, we apply this module to design and build a one-dimensional gravitational wave squeeze-and-excitation shrinkage network (GW-SENet) model suitable for gravitational wave coherent SNR data. The model learns the characteristics of the gravitational wave signal by means of neuron connections and carries out the task of detecting gravitational wave signals.

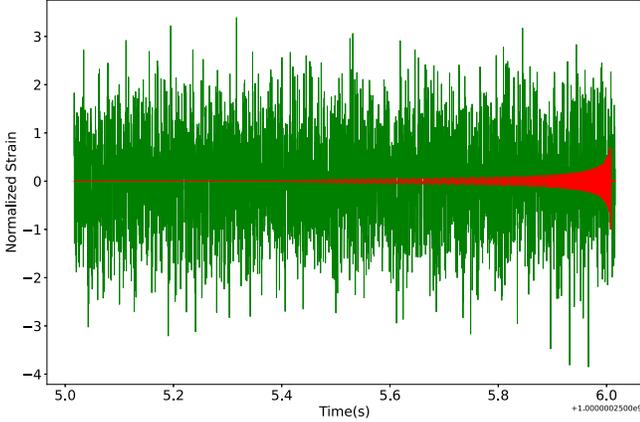
Our paper is structured as follows. The data and preprocessing are provided in Section 2. In Section 3, we introduce the relevant basis algorithms and describe the model structure of our gravitational wave detection in detail. Section 4 discusses experimental results obtained from the model proposed from Section 3. Finally, conclusions are presented in Section 5.

## 2. Data and Data Preprocessing

### 2.1. Data

This section mainly introduces the establishment process of multi-detector coherent SNR data. We first generate the signal of a single detector, and we use three detectors H, L and V, referring to the Advanced LIGO-Hanford, Advanced LIGO-Livingston and advanced Virgo detectors respectively (Fan et al. 2019). For each detector, we apply the *TaylorT4* waveform model (Buonanno et al. 2003) to simulate BNS signals with component masses in the range 1.3 to 1.5  $M_{\odot}$ . Each signal is given a random R.A. and decl. on the sky, and R.A. and polarization phase runs from 0 to  $2\pi$ , while decl. from  $\pi/2$ . to  $-\pi/2$ . The distance is set to 100 Mpc. We simulate the noise of the three real L, H and V detectors from designed sensitivity of power spectral density (psd) respectively. The sample rate of both noise time series and waveform signals is 4096 Hz. Then the BNS signals are inserted into the noise. The duration of the noise time series is 512 s. The peak amplitude of the BNS waveform is positioned in the range of 255.95–256.05 s of the time series. An example of a one second time series around the peak amplitude is shown in Figure 1. During the generation of the data sets, we used the LIGO Algorithm Library (LALSuite (LIGO Scientific Collaboration 2018)) along with the PyCBC software package (<https://ligo-cbc.github.io>).

Then there is the calculation of the multiple detector coherent SNR. The waveforms of gravitational wave signals



**Figure 1.** A simulated time series of a BNS gravitational wave signal with component masses  $m_1 = 1.4 M_\odot$  and  $m_2 = 1.4 M_\odot$  in Advanced LIGO.

obtained by different detectors are different, such as amplitude and phase. The distance, binary inclination, polarization and coalescence phase are the key parameters that affect them. When performing coherent calculations, the amplitude of the gravitational wave signal is decomposed into two polarizations (Harry & Fairhurst 2011; Macleod et al. 2016)

$$\begin{aligned} h_+(t) &= A^1 h_0(t) + A^3 h_{\frac{\pi}{2}}(t) \\ h_\times(t) &= A^2 h_0(t) + A^4 h_{\frac{\pi}{2}}(t), \end{aligned} \quad (1)$$

where  $h_0(t)$  and  $h_{\frac{\pi}{2}}(t)$  represent the phases of the two gravitational wave waveforms, respectively, which are usually considered to be orthogonal;  $A^i$  corresponds to the amplitude with the following equation

$$\begin{aligned} A^1 &= \frac{D_0}{D} \frac{(1 + \cos^2 \iota)}{2} \cos 2\phi_0 \cos 2\psi \\ &\quad - \frac{D_0}{D} \cos \iota \sin 2\phi_0 \sin 2\psi \\ A^2 &= \frac{D_0}{D} \frac{(1 + \cos^2 \iota)}{2} \cos 2\phi_0 \sin 2\psi \\ &\quad + \frac{D_0}{D} \cos \iota \sin 2\phi_0 \cos 2\psi \\ A^3 &= -\frac{D_0}{D} \frac{(1 + \cos^2 \iota)}{2} \sin 2\phi_0 \cos 2\psi \\ &\quad - \frac{D_0}{D} \cos \iota \cos 2\phi_0 \sin 2\psi \\ A^4 &= -\frac{D_0}{D} \frac{(1 + \cos^2 \iota)}{2} \sin 2\phi_0 \sin 2\psi \\ &\quad + \frac{D_0}{D} \cos \iota \cos 2\phi_0 \cos 2\psi, \end{aligned} \quad (2)$$

where  $D$  represents distance;  $D_0$  signifies scaling distance to normalize the waveform  $h_{0,\frac{\pi}{2}}$ ;  $\iota$  stands for inclination angle;  $\phi_0$  corresponds to coalescence phase;  $\psi$  means polarization angle. The gravitational wave signal observed by detector  $X$  is represented as a combination of two polarizations weighted by the detector antenna response,  $F_{\{+, \times\}}^X$  (Jaranowski et al. 1998),

$$h^X(t) = F_+^X h_+(t^X) + F_\times^X h_\times(t^X), \quad (3)$$

where the arrival time at detector  $X$  depends on the sky location of the source relative to the detector and the time of arrival at a fiducial location (Harry & Fairhurst 2011; Macleod et al. 2016). The matched filtering can be expressed as the inner product between the gravitational wave template waveform  $h$  and the detector data  $s$ , and the equation is as follows

$$(a^X|b^X) = 4 \operatorname{Re} \int_0^\infty \frac{\tilde{a}^X(f) \cdot [\tilde{b}^X(f)]^*}{S_h^X(f)}, \quad (4)$$

where  $a^X$  and  $b^X$  represent two time series;  $S_h^X(f)$  is the noise psd in detector  $X$ . For multiple detectors, we define their inner product as the sum of the inner products of individual detectors,

$$(a|b) \equiv \sum_{X=1}^D (a^X|b^X), \quad (5)$$

where  $D$  signifies the number of detectors. We compute the log-likelihood of multiple detectors as follows (Harry & Fairhurst 2011)

$$\ln \Lambda = (s|h) - \frac{1}{2} (h|h) = A^\mu (s|h_\mu) - \frac{1}{2} A^\mu M_{\mu\nu} A^\nu, \quad (6)$$

where  $h_\mu = (F_+ h_0, F_\times h_0, F_+ h_{\frac{\pi}{2}}, F_\times h_{\frac{\pi}{2}})$ , and  $M_{\mu\nu} \equiv (h_\mu|h_\nu)$ . Maximizing log-likelihood of the  $A^i$  values, the coherent SNR is defined as follows

$$\rho_{coh}^2 \equiv 2 \ln \Lambda|_{\max} = (s|h_\mu) M^{\mu\nu} (s|h_\nu). \quad (7)$$

For easier comparison with the coincidence search, we rewrite Equation (7), for which we introduce the complex SNR  $z^X$  in detector  $X$  as follows:

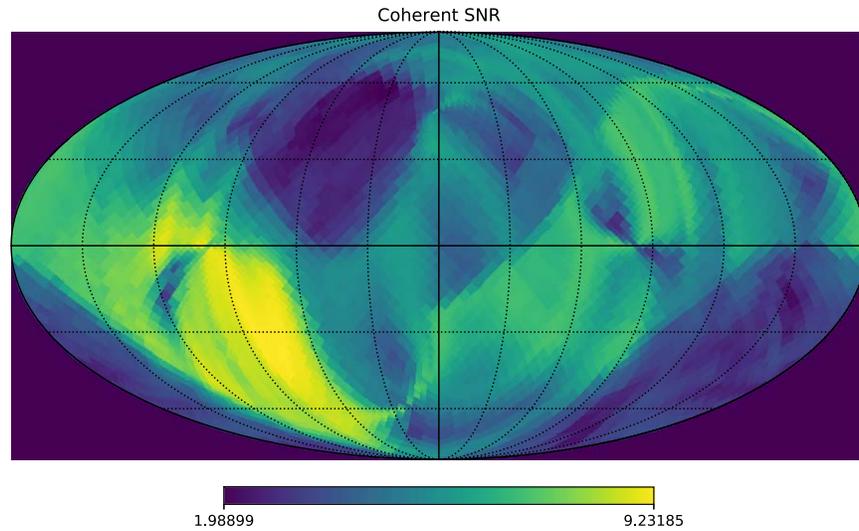
$$z^X = (s^X|h_0^X) + i(s^X|h_{\pi/2}^X). \quad (8)$$

Then the coherent SNR can be rewritten as

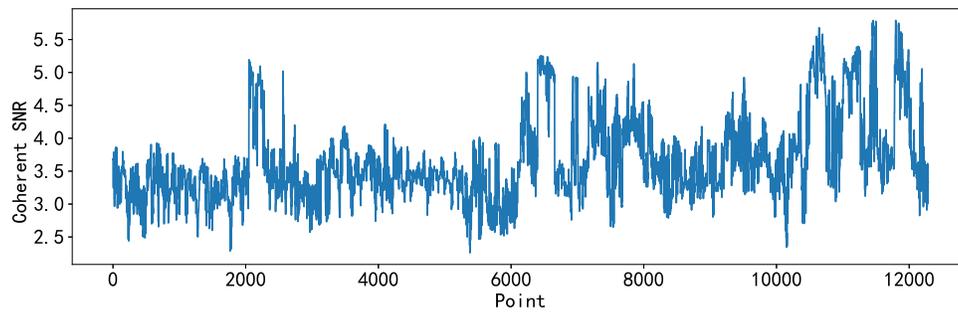
$$\begin{aligned} P_{XY} &= \left[ \frac{(\sigma^X F_+^X)(\sigma^Y F_+^Y)}{\sum_Z (\sigma^Z F_+^Z)^2} + \frac{(\sigma^X F_\times^X)(\sigma^Y F_\times^Y)}{\sum_Z (\sigma^Z F_\times^Z)^2} \right] \\ \rho_{coh}^2 &= \sum_{X,Y=1}^D z^X P_{XY} z^Y \end{aligned} \quad (9)$$

where  $\sigma^X = \sqrt{(h_0|h_0)_X}$  represents the sensitivity encoding for each detector.  $P_{XY}$  stands for the projection of the coherent SNR into a two-dimensional signal space. For more details, refer to Macleod et al. (2016). Figure 2 displays the sky-map of coherent SNR projected into two-dimensional space.

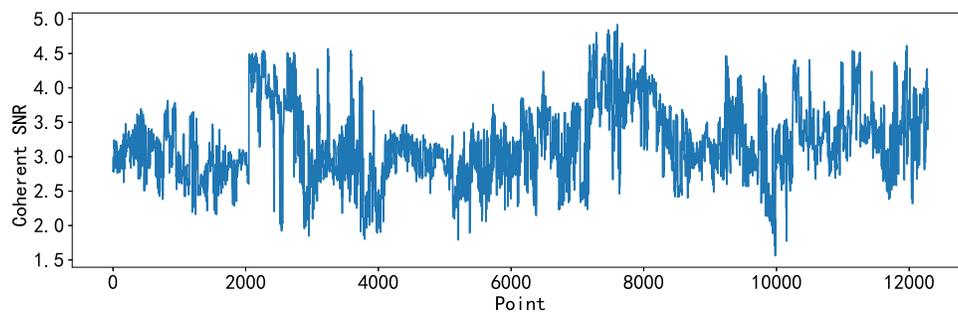
This is a fully-coherent all-sky search method (Macleod et al. 2016), and our calculated coherent SNR can be projected onto the sky-map. The coherent SNR we get is a one-dimensional array, and each value in it can be mapped to each pixel/point of the sky map. We use this one-dimensional coherent SNR data as the input to our deep neural network model. When all three detectors have obtained gravitational wave signals at the same time, we consider this to be a gravitational wave signal candidate. We call this type a signal real, as depicted in



**Figure 2.** Coherent SNR sky-map. A sky map can be generated by projecting coherent SNR data into a two-dimensional signal space.



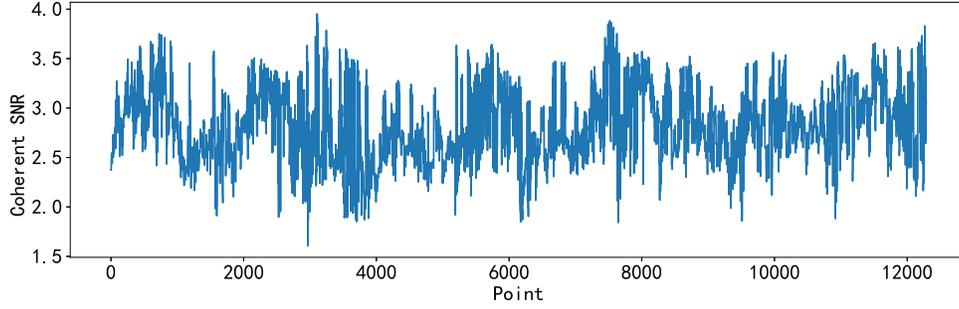
**Figure 3.** The gravitational wave signal of real type which is a random sample from coherent SNR data sets. The X-axis represents the corresponding points projected by coherent SNR onto the sky-map.



**Figure 4.** The gravitational wave signal of glitch type which is a random sample from coherent SNR data sets. The X-axis represents the corresponding points projected by coherent SNR onto the sky-map.

Figure 3. However, there will be false gravitational wave signals during the actual observation process of the equipment. For example, sometimes only one detector will obtain a signal with high SNR, and other detectors do not detect similar signals. This situation is likely caused by equipment noise. We

construct this type of glitch signal by simulating the time lag between the arrival of gravitational wave signals to the different detectors. We call this type of signal a glitch, as shown in Figure 4. Moreover, we simulated noise signals, that is, without injecting gravitational wave waveforms into the noise. We call



**Figure 5.** The gravitational wave signal of noise type which is a random sample from coherent SNR data sets. The X-axis represents the corresponding points projected by coherent SNR onto the sky-map.

this type of signal noise, as displayed in Figure 5. In Figure 3 to Figure 5, the X-axis represents the corresponding pixels/points projected by coherent SNR onto the sky map, which we call point here.

## 2.2. Data Preprocessing

Before calculating the multi-detector coherent SNR, although we have performed noise reduction operations such as whitening and filtering on the gravitational wave data of a single detector, the amplitude of the one-dimensional gravitational wave data generated by calculating the coherent SNR still has a certain fluctuation range. To make the model converge faster during training, we normalized the data using the Min-Max standardized method. It is also called the dispersion standardization method, which maps the data to the interval with the following equation

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}}, \quad (10)$$

where  $x_{\max}$  and  $x_{\min}$  represent the maximum and minimum values in the sample data, respectively.  $x^*$  signifies the standardized result of the input data. Since the categories input by the deep learning model are numerical, this paper uses One-Hot coding (Buckman et al. 2018) to “binarize” the category labels. Performing One-Hot on the label can be regarded as the establishment of different vectors of  $N$  categories. The dimension of each vector is  $N$ , of which only one-dimension is one, and the other dimensions are zero.

## 3. Relevant Algorithms and the Proposed Method

### 3.1. Relevant Algorithms

Soft thresholding was often used in signal denoising methods in the past (Donoho & Johnstone 1994; Donoho 1995). It has also been implemented in some optimization methods in recent years. Its purpose is to set features whose absolute value is lower than a certain threshold to zero, and adjust other features toward zero, i.e., shrinkage. The threshold is a



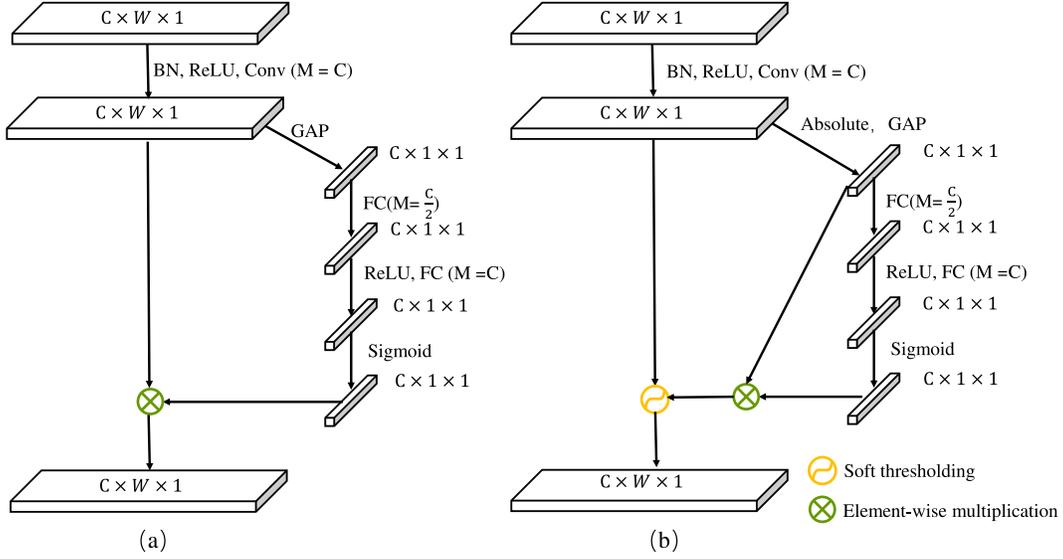
**Figure 6.** The soft thresholding function image, where the threshold value is 2.

parameter that needs to be preset, and its value has a direct impact on the result of signal denoising. The soft thresholding function is as follows

$$\text{soft}(x, \tau) = \begin{cases} x + \tau, & x < -\tau \\ 0, & |x| \leq \tau \\ x - \tau, & x > \tau \end{cases}, \quad (11)$$

where  $x$  is the original input signal,  $\tau$  is the threshold and  $\text{soft}(x, \tau)$  is the signal after denoising. We can find that the function values of the features within the threshold are all zero, so the threshold cannot be too large or too small. The image of the soft thresholding function is plotted in Figure 6, where we set the threshold  $\tau$  to 2. We can see that the soft thresholding is a nonlinear transformation. It has similar properties to the activation function ReLU, where the gradient is zero or one. So, some neural networks use soft thresholding as the activation function to prevent the gradient from vanishing and exploding. The difficulty in using the soft thresholding method for denoising is how to select a suitable threshold. Zhao et al. (2019) proposed a method to automatically learn the threshold by the neural network model.

When building a network model, selecting an appropriate optimizer algorithm can better adjust the weight and bias



**Figure 7.** (a) The structure of SE module. (b) The structure of SES module. Here,  $W$  and  $C$  denote the width and channel of the feature map, respectively; GAP means GAP layer; FC signifies fully connected layer.

parameters of the model, so that the value of the loss function is smaller. It can also better learn features and speed up the convergence of the model. The learning rates of optimizer algorithms currently used in neural networks are usually adaptive, including methods such as Adam (Kingma & Ba 2014), RMSprop (Hinton et al. 2012), etc., which all use a warm-up technique (He et al. 2016) to reduce variance. However, warmup, as a hyperparameter, can only find suitable warmup parameters by artificially continuously adjusting parameters. Based on this, Liu et al. (2019) proposed a new optimizer algorithm, Rectified Adam (RAdam). The algorithm self-adaptively adjusts the learning rate by using a rectifier, so that the model has better convergence and stability without tuning hyperparameters. Details of the RAdam algorithm are given in Algorithm 1.

**Algorithm 1.** Rectified Adam. All operations are element-wise.

---

**Input:**  $\{\alpha_t\}_{t=1}^T$ : step size,  $\{\beta_1, \beta_2\}$ : decay rate to calculate the moving average and moving 2nd moment,  $\theta_0$ : Initial parameters,  $f_t(\theta)$ : stochastic objective function.

**Output:**  $\theta_t$ : resulting parameters

- 1:  $m_0, v_0 \leftarrow 0, 0$  (Initialize moving 1st and 2nd moment)
- 2:  $\rho_\infty \leftarrow 2/(1 - \beta_2) - 1$  (Compute the maximum length of the approximated SMA)
- 3: **while**  $t = \{1, \dots, T\}$  **do**
- 4:  $g_t = \Delta_\theta f_t(\theta_{t-1})$  (Calculate gradients w.r.t. stochastic objective at time-step  $t$ )
- 5:  $m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$  (Update exponential moving 1st moment)
- 6:  $v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$  (Update exponential moving 2nd moment)
- 7:  $\hat{m}_t = m_t / (1 - \beta_1^t)$  (Compute bias-corrected moving average)
- 8:  $\rho_t = \rho_\infty - 2t\beta_2^t / (1 - \beta_2^t)$  (Compute the length of the approximated SMA)
- 9: **If** the variance is tractable, i.e.,  $\rho_t > 4$  **then:**

(Continued)

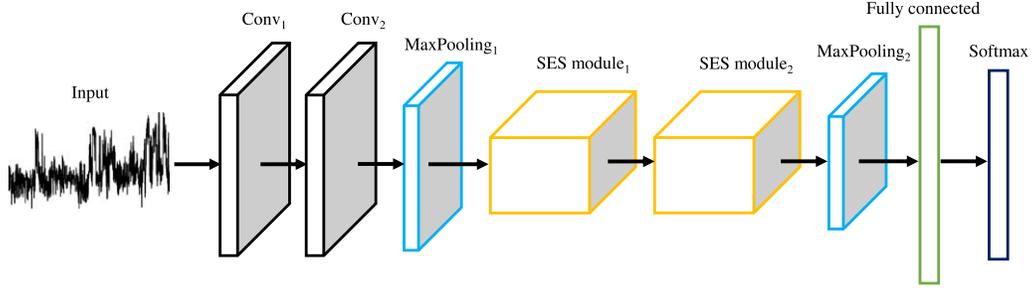
---

- 10:  $\hat{v}_t = \sqrt{v_t / (1 - \beta_2^t)}$  (Compute adaptive learning rate) **then:**
- 11:  $r_t = \sqrt{\frac{(m_t - 4)(m_t - 2)\rho_\infty}{(\rho_\infty - 4)(\rho_\infty - 2)\rho_t}}$  (Compute the variance rectification term)
- 12:  $\theta_t = \theta_{t-1} - \alpha_t r_t \hat{m}_t / \hat{v}_t$  (Update parameters with adaptive momentum)
- 13: **else:**
- 14:  $\theta_t = \theta_{t-1} - \alpha_t \hat{m}_t$  (Update parameters with un-adapted momentum)
- 15: **end if**
- 16: **end while**
- 4: **return**  $\theta_t$ .

---

### 3.2. Methods

Attention mechanism is widely used in deep learning methods and has achieved excellent results in many applications. The purpose of using the attention mechanism is to highlight important features, which is its basic idea. Since there are important features, there are also unimportant features, which can also be called noise features. Our idea is to use the attention mechanism to obtain important features, and then remove those unimportant or noise features, so that the model can extract more effective features. The soft thresholding as a shrinkage function can set the input data below the threshold to zero, and the data above the threshold will also shrink toward zero, which can achieve this purpose. So, the attention mechanism can be used to obtain important features, and the unimportant or noise features can be removed by the soft thresholding function. A typical method of using the attention mechanism in a convolutional neural network is SENet (Hu et al. 2018), whose module structure is shown in Figure 7(a).



**Figure 8.** The structure of GW-SESNet.

Therefore, we propose an SES feature extraction module based on the SENet and the soft thresholding shrinkage function, as illustrated in Figure 7(b). Since our input data are one-dimensional, the dimension of the feature map input by the SES module is  $C \times W \times 1$ . The feature map is transformed into a vector through the global average pooling (GAP) layer. Because the SES module will learn a threshold, the absolute value of the feature map is taken. After that, it goes through two fully connected (FC) layers. The first FC layer halves the number of channels, and the second FC layer restores the number of channels. A sigmoid function is used to get the weights of the feature map channels. The SE module directly multiplies this weight onto the feature map. But in our SES module, the threshold is obtained by multiplying this weight by the average value of the feature map calculated by GAP. This threshold is used to control the feature map output by the SES module.

We apply the SES module to a one-dimensional convolutional neural network and propose our GW-SESNet model. Figure 8 features the model structure. The model adopts a one-dimensional convolutional neural network structure as a whole, which consists of three parts, the input part, the feature extraction part and the category probability prediction part. In the feature extraction part, we used our proposed SES module to extract more efficient features. The input size of the network is 12288-dimensional gravitational wave coherent SNR data. We first use two convolutional layers with a kernel size of  $1 \times 32$  to extract features. After the convolution, a batch normalization (BN) layer is utilized to normalize the data, which makes the model converge faster. A max pooling layer is used to extract the salient features in a region. After that, we connect two SES modules to apply a soft thresholding as shrinkage functions to remove the features of the noisy signal through the learning of the feature channel threshold. Finally, a fully connected layer is used to classify the one-dimensional coherent SNR data. The specific parameters of the network model are displayed in Table 1. The establishment of the GW-SESNet model consists of two steps, the first is the training process of the model, and then the testing. So, we split the

**Table 1**  
The Structure and Setting Parameters for GW-SESNet Networks

No.	Layer	Feature Map	Kernel Size	Activation Function
1	<i>Input</i>	1	$1 \times 12288$	...
2	<i>Conv<sub>1</sub></i>	128	$1 \times 32$	<i>ReLU</i>
3	<i>Conv<sub>2</sub></i>	128	$1 \times 32$	<i>ReLU</i>
4	<i>MaxPooling<sub>1</sub></i>	128	$1 \times 4$	...
5	<i>Conv<sub>3</sub></i>	64	$1 \times 16$	<i>ReLU</i>
6	<i>GAP<sub>1</sub></i>	64	$1 \times 3072$	...
7	<i>FC<sub>1</sub></i>	...	$1 \times 16$	<i>ReLU</i>
8	<i>FC<sub>2</sub></i>	...	$1 \times 16$	<i>ReLU</i>
9	<i>Conv<sub>4</sub></i>	64	$1 \times 16$	<i>ReLU</i>
10	<i>GAP<sub>2</sub></i>	64	$1 \times 3072$	...
11	<i>FC<sub>4</sub></i>	...	$1 \times 16$	<i>ReLU</i>
12	<i>FC<sub>5</sub></i>	...	$1 \times 16$	<i>ReLU</i>
13	<i>MaxPooling<sub>2</sub></i>	64	$1 \times 4$	...
14	<i>FC<sub>7</sub></i>	...	$1 \times 1024$	<i>ReLU</i>
15	<i>Output</i>	...	$1 \times 3$	<i>Softmax</i>

coherent SNR data set into training and test sets. In the training phase, we set the batchsize hyperparameter to determine the number of samples for each training of the network, and we iteratively train the network model by setting the epoch hyperparameter so that the model can converge to a stable state. After the model training is completed, the well-trained model parameters are saved, and then we use the test set to verify the performance of the model. The specific description of the GW-SESNet is as described in Algorithm 2.

**Algorithm 2.** The training process of GW-SESNet

**Input:**

The number of training sets,  $N_{train}$ ;

The number of test sets,  $N_{test}$ ;

The number of samples taken from the training set for each training, *batchsize*;

The number of training times in the full sample of the training set, *epoch*;

**Output:**

classification result,  $C$ ;

1: **for** The loop count  $< epoch$ ;

2: Choose *batchsize* samples from training set.

(Continued)

---



---

```

3: Save GW-SESNet model.
4: Save weights and parameters.
5: end for
6: Test trained GW-SESNet.
7: return  $C$ .
    
```

---



---

We use multi-class cross-entropy as the loss function of our model. The output of the model normally is a number, representing a certain category. Since it is a multi-classification task, it is impossible to design a loss function, so we usually use the softmax function to turn the output of the model into a probability value, then use cross entropy to calculate the distance between the predicted value and the ground truth. The formula of softmax is as follows

$$p_i = \frac{e^{x_i}}{\sum_{j=1}^N e^{x_j}}, \quad (12)$$

where  $N$  represents the number of categories, here three;  $x_i$  means the prediction result of the GW-SES model for a certain sample;  $p_i$  signifies the probability distribution of the prediction result in the three categories of real, glitch and noise. Then the cross-entropy cost function is used to calculate the distance between it and the true probability distribution, which can be expressed as

$$L = \frac{1}{N} \sum_{i=1}^N L_i = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^M y_{ij} \log(p_{ij}), \quad (13)$$

where  $N$  represents the number of samples;  $M$  means the number of categories, here three;  $y_{ij}$  is the label, sign function, if the category of sample  $i$  is  $j$ , then  $y_{ij} = 1$ , otherwise  $y_{ij} = 0$ ;  $p_{ij}$  signifies the probability that the  $i$ th sample is predicted to be the  $j$ th label value; After calculating the cross-entropy error, we employ the RAdam algorithm for parameter optimization.

#### 4. Experimental Analysis

To evaluate the effectiveness of our proposed method for gravitational wave detection, we conduct a series of experiments on coherent SNR data sets. Experimental results demonstrate that the proposed method achieved better results compared with several other classical methods. We also conduct experiments under different SNRs, and the results show that the proposed method has good performance even at lower SNRs.

##### 4.1. Experimental Datasets

In our experiment, we simulated 2500 pieces of data for three types of signals, real, glitch and noise, respectively, with a total of 7500 pieces of data as training and test data. For each

**Table 2**  
The Experimental Data Sets of Gravitational Wave Coherent SNR

Category	Number of Instances	Training	Testing	Label
real	2500	2000	500	2
glitch	2500	2000	500	1
noise	2500	2000	500	0

type, 2000 pieces of data are used for training and 500 pieces of data are used for testing, as shown in Table 2. The real type of gravitational wave signal category is labeled as real, the false type of gravitational wave signal category is labeled as glitch, and the noise type of signal is labeled as noise.

##### 4.2. Experimental Results

In order to evaluate the classification performance, we utilize the accuracy and F1 score as the evaluation indicators of different types in the experimental results. F1 score is a measure of the accuracy of a classification model that considers both precision and recall. For a multi-classification task, there will be multiple different confusion matrices between different categories, and the recall and precision are also calculated on multiple confusion matrices. Therefore, there are two ways to calculate the recall and precision of multiple categories, namely macro-average and micro-average. The macro-average is to first obtain the precision and recall of different categories, then sum and average. This paper uses the macro-averaged F1 score, which is defined as follows:

$$\begin{aligned}
 P_i &= \frac{TP_i}{TP_i + FP_i} \\
 P &= \frac{\sum_{i=1}^N P_i}{N} \\
 R_i &= \frac{TP_i}{TP_i + FN_i} \\
 R &= \frac{\sum_{i=1}^N R_i}{N} \\
 F1 &= 2 \frac{PR}{P+R}, \quad (14)
 \end{aligned}$$

where  $P_i$  and  $R_i$  denote the precision and recall of each category respectively;  $N$  the number of categories;  $P$  and  $R$  signify the multi-category precision and recall, respectively, calculated using the macro-average; TP, FP, TN and FN correspond to true positive, false positive, true negative and false negative, respectively. Based on FP and TN, the false positive rate (FPR) can be obtained, which is defined as follows

$$\begin{aligned}
 FPR(i) &= \frac{FP_i}{FP_i + TN_i} \\
 FPR &= \frac{\sum_{i=1}^N FPR(i)}{N}, \quad (15)
 \end{aligned}$$

where  $FPR(i)$  denotes the false positive rate of each category;  $N$  stands for the number of categories; Here, FPR is also called false alarm rate, which is a common indicator in gravitational

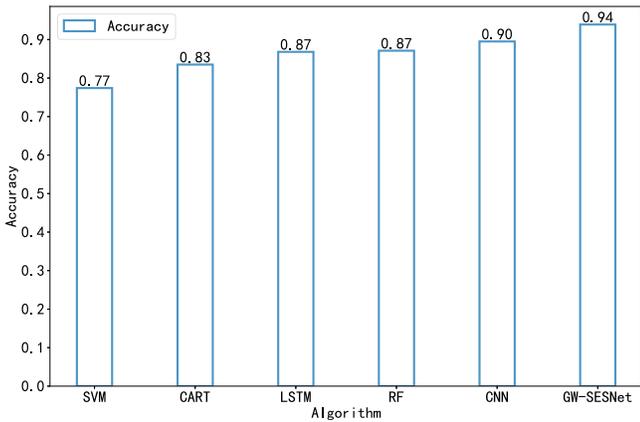
**Table 3**  
Comparison of Different Algorithms in Gravitational Wave Coherent SNR Classification

Algorithm	Precision			Recall			F1 score			Average F1 score
	Real	Glitch	Noise	Real	Glitch	Noise	Real	Glitch	Noise	
SVM	0.90	0.80	0.85	0.70	0.94	0.96	0.73	0.72	0.86	0.77
RF	0.99	0.80	0.85	0.66	0.94	0.99	0.80	0.87	0.92	0.86
CART	0.99	0.70	0.92	0.50	0.99	0.99	0.66	0.83	0.96	0.81
LSTM	0.84	0.82	0.91	0.73	0.86	0.99	0.78	0.84	0.95	0.86
CNN	0.97	0.85	0.89	0.71	0.89	0.99	0.82	0.91	0.93	0.89
GW-SESNet	0.97	0.95	0.88	0.84	0.98	0.98	0.90	0.97	0.93	0.93

**Table 4**

The AUC Results of Different Algorithms are Compared on the Gravitational Wave Coherent SNR

Algorithm	AUC			Average AUC
	Real	Glitch	Noise	
SVM	0.875	0.870	0.958	0.903
RF	0.956	0.982	0.991	0.968
CART	0.744	0.894	0.978	0.872
LSTM	0.916	0.958	0.990	0.956
CNN	0.943	0.986	0.985	0.973
GW-SESNet	0.975	0.995	0.989	0.989



**Figure 9.** Comparison of the accuracy of different algorithms in gravitational wave coherent SNR classification.

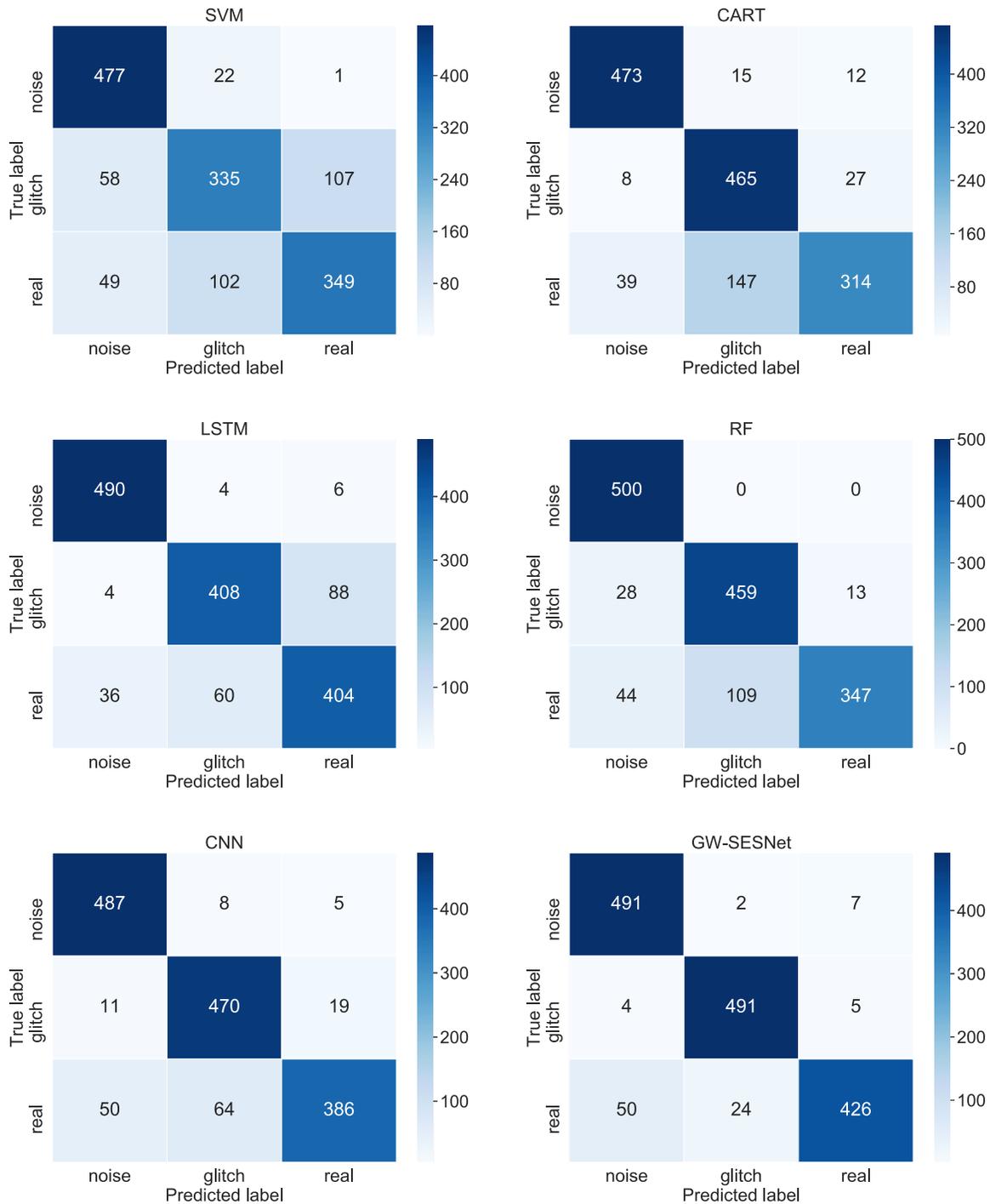
wave detection. The recall  $R$  in Equation (14) is also called true positive rate (TPR), and it is also called sensitivity in other literatures (Li et al. 2020). The receiver operating characteristic (ROC) curve, which is defined by TPR and FRP, is also used as an evaluation metric for our proposed method.

We compare our proposed method with several other methods. In order to demonstrate the advantages of a deep learning convolutional neural network in processing gravitational wave coherent SNR data, we compare with some algorithms in machine learning methods, including SVM (Hearst et al. 1998), RF (Svetnik et al. 2003) and CART

(Loh 2011). Because our data are one-dimensional, we also compare with the LSTM (Hochreiter & Schmidhuber 1997) method. For demonstrating the effectiveness of our proposed SES module, we compare with the one-dimensional CNN method. SVM, RF and CART use the methods in the scikit-learn module, a third-party machine learning module commonly used in Python, for training and testing. The parameters in the model use the default parameters set in the scikit-learn module. The one-dimensional CNN model uses the same convolutional layers, pooling layers and BN layers as the GW-SESNet model, but does not use the SES module. The GW-SESNet model relies on a small part of the training data as the validation set during training, in order to observe the training situation of the network during training. If there is a problem with the model, the training can be stopped at any time according to the accuracy of the validation set. In this experiment, the batch size of the GW-SESNet model is 32 and the epoch is 100.

In order to compare the classification effects of different methods, the experiment first compares the precision, recall and harmonic average F1 scores of different methods, as displayed in Table 3. We show metrics for each category. From the table we can see that the precision of the GW-SESNet model in the real category is as high as 0.97. Comparing all models, the recall of the real category is the best on the model GW-SESNet, and in the same category, the difference between the recall and precision of the GW-SESNet model is relatively small, and the value is the highest. Observing the RF method and CART method, in the real category, the recall and precision are significantly different. For example, the precision of the CART method is 0.99, but the recall is only 0.50. It shows that the machine learning algorithm has poor classification performance on the coherent SNR data of gravitational waves. Although the F1 score of RF is 0.86, the precision and recall are quite different on the real category.

We also utilize the AUC value to evaluate the comprehensive classification performance of our experimental results, as shown in Table 4. The GW-SESNet model displays best results for the AUC values of the three different categories and their average AUC values. The AUC value on the real category is 0.975, and the average AUC value of the three categories is

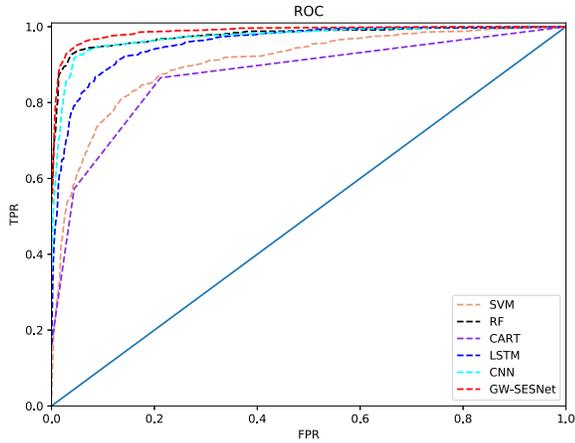


**Figure 10.** The confusion matrices for three categories of real, glitch and noise of the different algorithms.

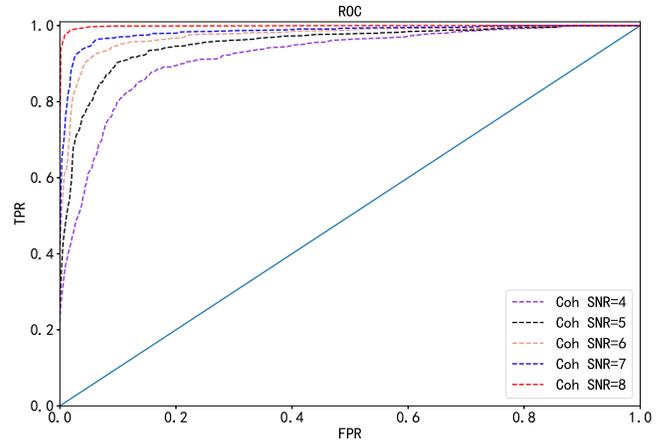
0.989. Observing the machine learning method, the AUC value is also relatively low. This also verifies that our method has certain advantages over machine learning methods on this data set from different metrics. We use histograms to visually show the accuracy values of each model, as featured in Figure 9. It

can be ascertained from the experimental results that the GW-SESNet model used in this paper has the highest accuracy, which is 0.94, higher than other methods.

At the same time, the confusion matrix and multi-class ROC curve were used to demonstrate the classification effect



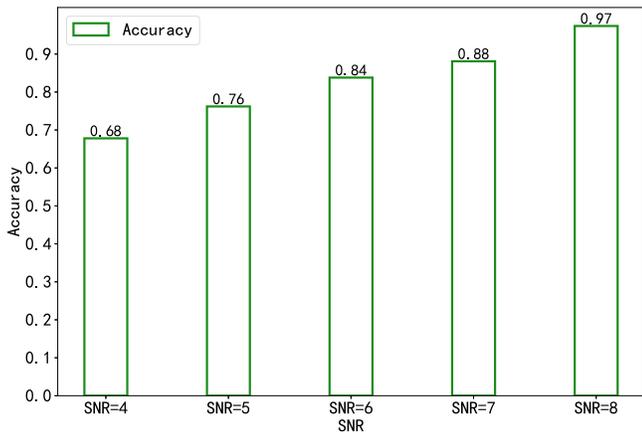
**Figure 11.** The ROC curves of different model algorithms are compared on the gravitational wave coherent SNR.



**Figure 13.** The ROC curves of the GW-SESNet model with different SNRs on the gravitational wave coherent SNR.

**Table 5**  
Comparison of Different SNRs of GW-SESNet Model in Gravitational Wave Coherent SNR Classification

Coherent SNR	Precision			Recall			F1 Score			Average F1 Score
	Real	Glitch	Noise	Real	Glitch	Noise	Real	Glitch	Noise	
Coh-SNR = 4	0.86	0.96	0.53	0.31	0.74	0.98	0.46	0.83	0.68	0.66
Coh-SNR = 5	0.93	0.96	0.60	0.48	0.83	0.98	0.63	0.89	0.75	0.76
Coh-SNR = 6	0.96	0.95	0.70	0.62	0.92	0.98	0.75	0.93	0.82	0.83
Coh-SNR = 7	0.97	0.94	0.78	0.72	0.94	0.98	0.83	0.94	0.87	0.88
Coh-SNR = 8	0.99	0.97	0.97	0.95	0.99	0.98	0.97	0.98	0.98	0.97



**Figure 12.** The accuracy results of the GW-SESNet model with different SNRs on the gravitational wave coherent SNR.

of different contrasting methods. Figure 10 depicts the confusion matrices for the six methods used in the experiments. Figure 11 plots the ROC curves of different methods on the test data set. It can be seen from the ROC curve that the method GW-SESNet (red line) implemented in this paper is closest to the upper left corner, the area enclosed by the

**Table 6**  
Comparison Results of Sensitivities Under Different Coherent SNRs when FPR is 0.001, 0.01 and 0.1

Coherent SNR	FPR = 0.001	FPR = 0.01	FPR = 0.1
Coh-SNR = 4	0.267	0.371	0.799
Coh-SNR = 5	0.318	0.485	0.903
Coh-SNR = 6	0.468	0.618	0.947
Coh-SNR = 7	0.555	0.767	0.968
Coh-SNR = 8	0.942	0.977	0.999

coordinate axis is the largest and the effect is the best. By comparing the experimental results of the one-dimensional CNN model and the GW-SESNet model, it is affirmed that the SES module proposed in this paper can better learn the effective features of gravitational wave coherent SNR data. From the different indicators of the experimental results, we can see that the performance of the GW-SESNet network model is better than other methods. The one-dimensional convolutional neural network based on the SES module proposed in this paper displays good performance on the multiple detector coherent SNR data. It can achieve satisfactory classification of real, glitch and noise gravitational wave data.

In order to compare the classification performance of the GW-SESNet network model proposed in this paper under different data qualities, we selected five different coherent SNR test data sets. The coherent SNRs are 4, 5, 6, 7 and 8, and 300 samples are produced for each SNR. We compare the precision, recall and their harmonic mean F1 scores under different SNRs, as shown in Table 5. With the increase of the one-dimensional coherent SNR, the difference between the precision and recall of each category decreases, the precision and recall of different categories increase, and the average F1 score of the three categories increases gradually. The F1 score at a coherent SNR of 8 is 97%. It can be shown that the data quality directly affects the experimental results of the model.

Figure 12 shows the accuracy histogram of the GW-SESNet model under different SNRs. When the coherent SNR is 8, the accuracy of the model can be as high as 97%. With the continuous reduction of the coherent SNR, the accuracy of the model classification is also gradually reduced. In order to more intuitively show the detection ability of the GW-SESNet model under different SNRs, this experiment generates the ROC curves, as depicted in Figure 13. All curves are above the diagonal. The red curve represents the curve with an SNR of 8, which is closest to the upper left corner, indicating the best classification effect. The purple curve represents the curve with an SNR of 4, at the bottom, indicating the worst classification effect. According to the ROC curve, we can also see that the quality of the data has a direct impact on the classification effect of the model.

The false alarm rate is an important indicator of concern to astronomers when detecting weak gravitational wave signals. From the confusion matrix in Figure 10, we can see that our proposed method GW-SESNet, out of the 438 gravitational wave signals predicted to be the real type, 12 are false predicted, and the false alarm rate can be calculated to be 0.027. According to the confusion matrix of each algorithm, their false positive rate can be obtained. We calculated the sensitivities at different coherent SNRs at FPRs of 0.001, 0.01 and 0.1, as displayed in Table 6. The results affirm that our proposed method has a very low false alarm rate in gravitational wave detection and exhibits strong performance.

## 5. Conclusions

Since the detection of real gravitational wave signals, how to use automated methods to efficiently detect gravitational wave signals with low SNR has become more and more important. Because gravitational waves are weak signals, it is relatively difficult for us to detect gravitational wave signals obtained by a single detector. With the establishment of detectors around the world, it has become a trend to detect gravitational wave signals by coherent multiple detectors. In this paper, we propose a gravitational wave detection model based on deep convolutional neural networks and multiple detector coherent SNRs. First, we simulate BNS signals from multiple detectors,

including the Advanced LIGO and Virgo detectors. Second, we compute the coherent SNR of observed signals from multiple detectors using a coherent method to improve the SNR. We establish a one-dimensional deep convolutional neural network model for coherent SNR data. To make the model better learn useful features, we propose a novel SES feature extraction module based on soft thresholding and attention network SENet structure. The module can automatically learn a threshold according to which useful features are retained and noisy features are eliminated. Then we design a gravitational wave detection model GW-SESNet based on this module. We train and validate the performance of our model on the coherent SNR data set. It can be seen from the experimental results that our model obtains high classification accuracy on the gravitational wave coherent SNR data. Compared with other classical model algorithms, our method shows better performance in the gravitational wave detection task.

## Acknowledgments

This research was supported by the National Natural Science Foundation of China and Beijing Natural Science Foundation (No. 4224091) and China Postdoctoral Science Foundation (No. 2021M693402). We are grateful for the data provided by the Astronomy Department of Beijing Normal University, Professor Zonghong Zhu.

## References

- Aasi, J., Abbott, B., Abbott, R., et al. 2015, *CQGra*, 32, 074001
- Abbott, B. P., Abbott, R., Abbott, T., et al. 2016a, *PhRvL*, 116, 241103
- Abbott, B. P., Abbott, R., Abbott, T., et al. 2016b, *PhRvL*, 116, 061102
- Acernese, F. a., Agathos, M., Agatsuma, K., et al. 2014, *CQGra*, 32, 024001
- Aso, Y., Michimura, Y., Somiya, K., et al. 2013, *PhRvD*, 88, 043007
- Beheshtipour, B., & Papa, M. A. 2020, *PhRvD*, 101, 064009
- Buckman, J., Roy, A., Raffel, C., & Goodfellow, I. 2018, in Proc. 6th Int. Conf. on Learning Representations (Vancouver, Canada)
- Buonanno, A., Chen, Y., & Vallisneri, M. 2003, *PhRvD*, 67, 104025
- Chatterjee, C., Wen, L., Vinsen, K., Kovalam, M., & Datta, A. 2019, *PhRvD*, 100, 103025
- Colgan, R. E., Corley, K. R., Lau, Y., et al. 2020, *PhRvD*, 101, 102003
- LIGO Scientific Collaboration 2018, LIGO Algorithm Library–LAL Suite (<https://lscsoft.docs.ligo.org/lalsuite/lalsimulation>)
- Donoho, D. L. 1995, *ITIT*, 41, 613
- Donoho, D. L., & Johnstone, J. M. 1994, *Biometrika*, 81, 425
- Fan, X., Li, J., Li, X., Zhong, Y., & Cao, J. 2019, *SCPM*, 62, 1
- Gabbard, H., Williams, M., Hayes, F., & Messenger, C. 2018, *PhRvL*, 120, 141103
- George, D., & Huerta, E. A. 2018, *PhLB*, 778, 64
- George, D., Shen, H., & Huerta, E. 2017, arXiv:1706.07446
- Harry, I. W., & Fairhurst, S. 2011, *PhRvD*, 83, 084002
- He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 770
- Hearst, M. A., Dumais, S. T., Osuna, E., Platt, J., & Scholkopf, B. 1998, *IEEE Intelligent Systems and Their Applications*, 13, 18
- Hinton, G., Srivastava, N., & Swersky, K. 2012, Neural networks for machine learning Lecture 6a Overview of Mini-Batch Gradient Descent ([https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture\\_slides\\_lec6.pdf](https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf))
- Hochreiter, S., & Schmidhuber, J. 1997, *Neural Comput.*, 9, 1735
- Hu, J., Shen, L., & Sun, G. 2018, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition, 7132
- Jaranowski, P., Krolak, A., & Schutz, B. F. 1998, *PhRvD*, 58, 063001

- Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
- Li, X.-R., Yu, W.-L., Fan, X.-L., & Babu, G. J. 2020, *FrPhy*, 15, 1
- Liu, L., Jiang, H., He, P., et al. 2019, arXiv:1908.03265
- Loh, W.-Y. 2011, *Data Mining and Knowledge Discovery*, 1, 14
- Macleod, D., Harry, I., & Fairhurst, S. 2016, *PhRvD*, 93, 064004
- Razzano, M., & Cuoco, E. 2018, *CQGra*, 35, 095016
- Saulson, P. R. 1994, *Fundamentals of Interferometric Gravitational Wave Detectors* (Singapore: World Scientific)
- Svetnik, V., Liaw, A., Tong, C., et al. 2003, *J. Chem. Inf. Comput. Sci.*, 43, 1947
- Szabo, T., Pierpaoli, E., Dong, F., Pipino, A., & Gunn, J. 2011, *ApJ*, 736, 21
- Taracchini, A., Buonanno, A., Pan, Y., et al. 2014, *PhRvD*, 89, 061502
- Wang, H., Wu, S., Cao, Z., Liu, X., & Zhu, J.-Y. 2020, *PhRvD*, 101, 104003
- Wei, W., & Huerta, E. 2021, *PhLB*, 816, 136185
- Willke, B., Aufmuth, P., Aulbert, C., et al. 2002, *CQGra*, 19, 1377
- Zhao, M., Zhong, S., Fu, X., Tang, B., & Pecht, M. 2019, *IEEE Trans. Ind. Inf.*, 16, 4681