Frontiers of Physics
https://doi.org/10.1007/s11467-021-1150-1
Front. Phys.
17(5), 54501 (2022)

RESEARCH ARTICLE

Identify real gravitational wave events in the LIGO-Virgo catalog GWTC-1 and GWTC-2 with convolutional neural network

Meng-Qin Jiang[1], Nan Yang[2], Jin Li[1,†]

[1] *College of Physics, Chongqing University, Chongqing 401331, China*
[2] *Department of Electronical Information Science and Technology, Xingtai University, Xingtai 054001, China*
Corresponding author. E-mail: [†] cqujinli1983@cqu.edu.cn
Received October 20, 2021; accepted January 11, 2022

In recent years, machine learning models have been introduced into the field of gravitational wave (GW) data processing. In this paper, we apply the convolutional neural network (CNN) to LIGO O1, O2, O3a data analysis to search the released 41 GW events which are emitted from binary black hole (BBH) mergers (here we exclude the events from binary neutron star (BNS) mergers, and the events that are not detected simultaneously by Hanford (H) and Livingston (L) detectors), and use time sliding method to reduce the false alarm rate (FAR). According to the results, the 41 confirmed GW events of BBH mergers can be classified successfully by our CNN model. Furthermore, through restricting the number of consecutive prewarning from sequential samples intercepted continuously in LIGO O2 real time-series and vetoing the coincidences of noise from H and L, the FAR is limited to be less than once in 2 months. It is helpful to promote LIGO real time data processing.

Keywords convolutional neural network, gravitational wave events, false alarm rate

1 Introduction

In 2016, the Laser Interferometer Gravitational Wave Observatory (LIGO) and Virgo announced the first detection of gravitational wave (GW) event GW150914 [1, 2], which is emitted from a binary black hole (BBH) merger. The successful observation of the GW event opened a new window for observing the universe. During the first observing run (O1) and second observing run (O2), the LIGO and Virgo scientific collaboration have reported 11 GW events from the compact binary coalescences (CBC) [3], including 10 BBH events and the first event from a binary neutron star (BNS) merger, GW170817 [4]. Until the first half of the third observing run (O3a), 50 real GW events from CBC have been successfully detected [5–11]. All the detections are based on the matched filtering method [12–15]. Since the theoretical waveform of GW from CBC is clear, matched filtering technology has a good performance on searching such weak signals in the noisy data.

On the other hand, high latency is inevitable to determine an event, because the matched filtering method requires a full search in the template bank to match the signal [16]. In addition, the completeness and accuracy of the GW waveform templates are prerequisites for ensuring the work of the matched-filtering method [17]. With the continuous expansion of the theoretical waveform in the ever-expanding parameter space, the search space of the matched-filtering also increases, which means that we may undergo the risk of losing GW events exceeding the theoretical expectation [16].

Recently, deep learning (DL) as a branch of machine learning (ML) is bringing about a revolution in data analysis in various fields [18–22]. Methods based on deep neural networks have become the focus and been widely explored as tools for rapid GW detections and inferences [23]. For classification tasks, DL algorithms perform better than traditional ML algorithms in terms of model accuracy and complexity. It greatly reduces the amount of calculation in the online process and meets the needs of real-time detection [20]. At present, DL has achieved some successes in processing GW data. In 2017, George and his colleagues [24] applied convolutional neural network (CNN) to the GW events detection, which opened the era of GW data processing using ML. Some works for the signal classification and parameter estimation on the simulated GW datasets have supported the feasibility of ML in GW data processing [25–34, 37–40]. Meanwhile the glitch classification for GW detections based on Gravity Spy project [41–44], has combined DL and citizen science to identify the considerable variety of glitches in the twin Advanced LIGO detectors.

As the most concerned issue, observing the real GW events by DL has been discussed intensively. The team of

* This article can also be found at http://journal.hep.com.cn/fop/EN/10.1007/s11467-021-1150-1.

the National for Supercomputing, University of Illinois at Urbana-Champaign designed a DL set that can identify the real GW events generated by the merger of a BBH in O2 and O3 data. This method can identify real events in advanced LIGO data, and the FAR is controlled effectively [45]. The residual network has been applied to the LIGO/Virgo the real GW events detection and the parameter estimation [31, 46]. Significantly the authors of [34] applied the CNN model to the real GW data of LIGO, and first proposed using the signal-to-noise ratio (SNR) after matching the data with the template bases as the input of CNN, which is able to improve the efficiency of network operation. Their results show the network having strong robustnesscan to find 11 events of O1 and O2 with different real noises. At the same time, they also found more than 2000 candidates through the CNN model, which needs to be further analyzed [34].

Motived by these works, we plan to use CNN with the input as Ref. [34] to detect the latest real GW events for BBH in corresponding LIGO/Virgo real data. As the main differences from Ref. [34], we analyze the time-series of two detectors(H and L) and further discuss how to reduce the FAR through the threshold value of the consecutive positive sample number and time shift between the two detectors to veto the coincidences of loud noise. The plan of this paper is as follows: In Section 2 the generation of GW datasets for training and testing are described. In Scetion 3 the structure and hyperparameters of our CNN are determined. In Section 4 based on our adjusted CNN architecture,we apply our trained network for the real GW events in the LIGO/Virgo data. the performance on the testing datasets and FAR for the real data are discussed. At last Section 5 is devoted to a summary and some remarks.

2 Construction of data

Template bases: As mentioned in Section 1, our input is the SNR series which is a match between simulation data and template bases. We use some specific template bases to express the features of the candidate GW signals as much as possible. We select the template bases based on the following points: cover the parameter space as much as possible. For the real GW events, the total masses of sources are in the range of 20–150 $M_\odot$, and 78% of the total masses of sources are distributed in the range of 20–74 $M_\odot$, so for the template bases, we choose the total masses are $M = 20+3nM_\odot$ $(n = 0, 1, 2, \cdots, 18)$. And the luminosity distances of sources are in the range of 320–4450 Mpc, so the luminosity distances of the template bases are set to be in the range of 300–5000 Mpc. extracting the features of the GW signals with fixed mass ratio. Although mass ratio plays an important role in the GW waveforms,but for the non-extreme mass ratio cases,the features of the GW signals with various mass ratio on the bases could also be emboied. Therefore we fix 57 templates bases, which have parameters shown in Fig. 1.

The basic principle of matched filtering is to measure the correlation of the data strain $s(t) = n(t) + h(t)$ and the (astrophysical) signal $h(t)$. So the matched filtering signal-to-noise ratio (SNR) is defined to quantify the consistency between the template base $T(t)$ given in the template bank and the strain $s(t)$ recorded at t_0 as

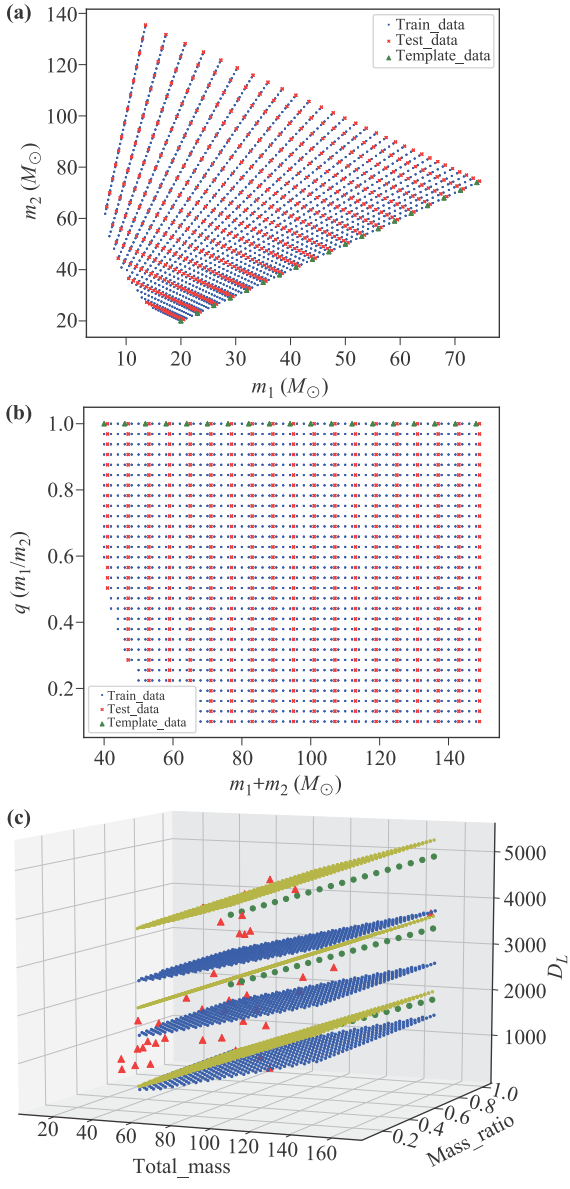

Fig. 1 **(a)** The component mass distribution of training/testing datasets and template bases. The blue, red and green dots represent the component mass distribution of training, testing datasets and template bases respectively. **(b)** The total mass and mass ratio of training/testing datasets and template bases. **(c)** Total mass, mass ratio and luminosity distance distribution in space of training (yellow dot)/testing datasets (blue dots), template bases (green dots) and 41 GW events of O1, O2, and O3 (red triangles) .

$$SNR(t_0) := \int_{-\infty}^{\infty} \frac{\widetilde{s(f)} \cdot \widetilde{T}^*(f) \cdot e^{2\pi i f t_0}}{S_n(f)} df. \quad (1)$$

According to this formula, we try to calculate the correlation of the strain data $s(t) = n(t) + h(t)$ and the template bases rather than all the templates in the bank. Considering two detectors Hanford (H) and Livingston (L), we get a 2×57 two-dimensional SNR series instead of time-series as input so that the calculation speed is enhanced greatly explained in section 5. Although many details in the time-series will be ignored, the correlation of data strain and our template bases are recorded. Furthermore the bases cover the parameter space of the possible GW signals, the signals information will be keeped as possible. It should be noted that the number of base vectors will increase rapidly when considering more parameters. In the current situation, if 9 parameters are added, then the number of base vectors will become $19 \times 3^7 = 41\,553$ (19 comes from the number of the mass bases; 3 represents that the previous bases are tripled and the new parameters of the bases are made to be evenly distributed in the parameter space.), a very large number indeed. However, with better computer equipment it can still be coped with. On the other hand, if only the mission of detecting gravitational waves is involved, it might not be necessary to take all of the parameters into account. For instance, we only choose the template with mass ratio of 1 as the bases in the manuscript and get the results. Certainly we are planning to consider this issue in the future work.

Training and testing datasets: The training datasets are used to optimize the weights of network. Usually, the network is trained multiple times on the training datasets, and each training round is called a epoch. After multiple epochs, the network can learn the abstract features of training datasets until the network is trained well (i.e., the loss function getting the minimum value). Generally, in order to avoid overfitting, a part of training datasets are set to be validation datasets in training process for evaluating the ability of generalization. Testing datasets should be generated independently of the training and validation datasets, and used to test the performance of the trained CNN model.

In this paper, the signals of training and testing datasets are simulated from CBC waveforms using PyCBC with SEOBNRv4 approximation [34–36]. For simplicity, we only consider spinless black holes and quasi-circular binary stars without orbital eccentricity. For the training datasets, the total mass of the BBH ranges from 40 to 150 ($M_\odot$), where the range of m_1 is 3 to 74 $M_\odot$, and the step size is $3M_\odot$, and the mass ratio $q = m_1/m_2$ from 0.1 to 1, and the step size is 0.1. Regarding to be the luminosity distance D_L, it ranges in 300–5000 Mpc. For the testing datasets, the total mass of BBH ranges in 41–151 $M_\odot$, and the training datasets and testing datasets without are joint values.

The noises of training and testing datasets are obtained from real advanced LIGO strain data, which can be downloaded from the Gravitational Wave Open Science Center (GWOSC) [31]. The selected segments come from the real strain data without GW events, which contain 42 756 s successive strain data from the O2 run as noise data. Then split the noise data into 21378 background noise datasets and each one contains 2 seconds time-series. All the time-series are whitened.

Each sample includes two 2 seconds time-series with a sampling rate of 4096 Hz corresponding to H and L detectors respectively, which contains a signal immersed in the background noise (i.e., positive sample, labeled "1") or pure noise without any simulated GW signal (i.e., negative sample, labeled "0"). For the training datasets, there are 4746 simulated GW waveforms. Then we superimpose two different noise datasets on each GW waveform, and get 9492 positive samples for training. We select another 8192 pure background noise datasets as the negative samples for training. For the testing datasets, there are 1646 simulated GW waveform. We add 1646 background noise datasets to the simulated pure GW waveforms as positive samples in the testing datasets, and another 2048 pure background noise datasets as negative samples. It should be noted that the samples in the testing datasets and the training datasets are completely disjointed. The distribution of component masses for template bases and training, testing datasets are shown in Fig. 1.

3 Structure of neural network

We still choose a CNN and build appropriate input and output dimensions for the layers. The following figure [Fig. 2(b)] describes our network structure. The input is 2×57 dimensions SNR series from matching each sample and template bases [cf. Fig. 2(a)]. The filter sizes of dense layers are 128, 64, 2 respectively. The sizes of the convolutional layers are 32, 64, 128, 256, 128 respectively, we use kernel size of 1×3 for the convolution layers. The pooling layers use maximum pooling with a pooling window of 2 and a step size of 2, which means that the feature sequence is down-sample to half the length of the original sequence and retain the feature with the maximum value. The activation function is chosen to be ReLU, which is applied to the output of every convolutional and dense layer. Subsequently, the feature sequence extracted by the convolutional layers is sent into the fully connected layers to finish classification. In this model, we use the softmax function to map the output s of CNN into the range of (0,1):

$$p = \frac{1}{1 + e^{-s}} \quad (2)$$

and set the classification threshold to be p_c =0.5. Samples with a discriminant value greater than p_c are judged as positive samples (i.e., including GW signals), otherwise they are judged as negative samples (i.e., noise).

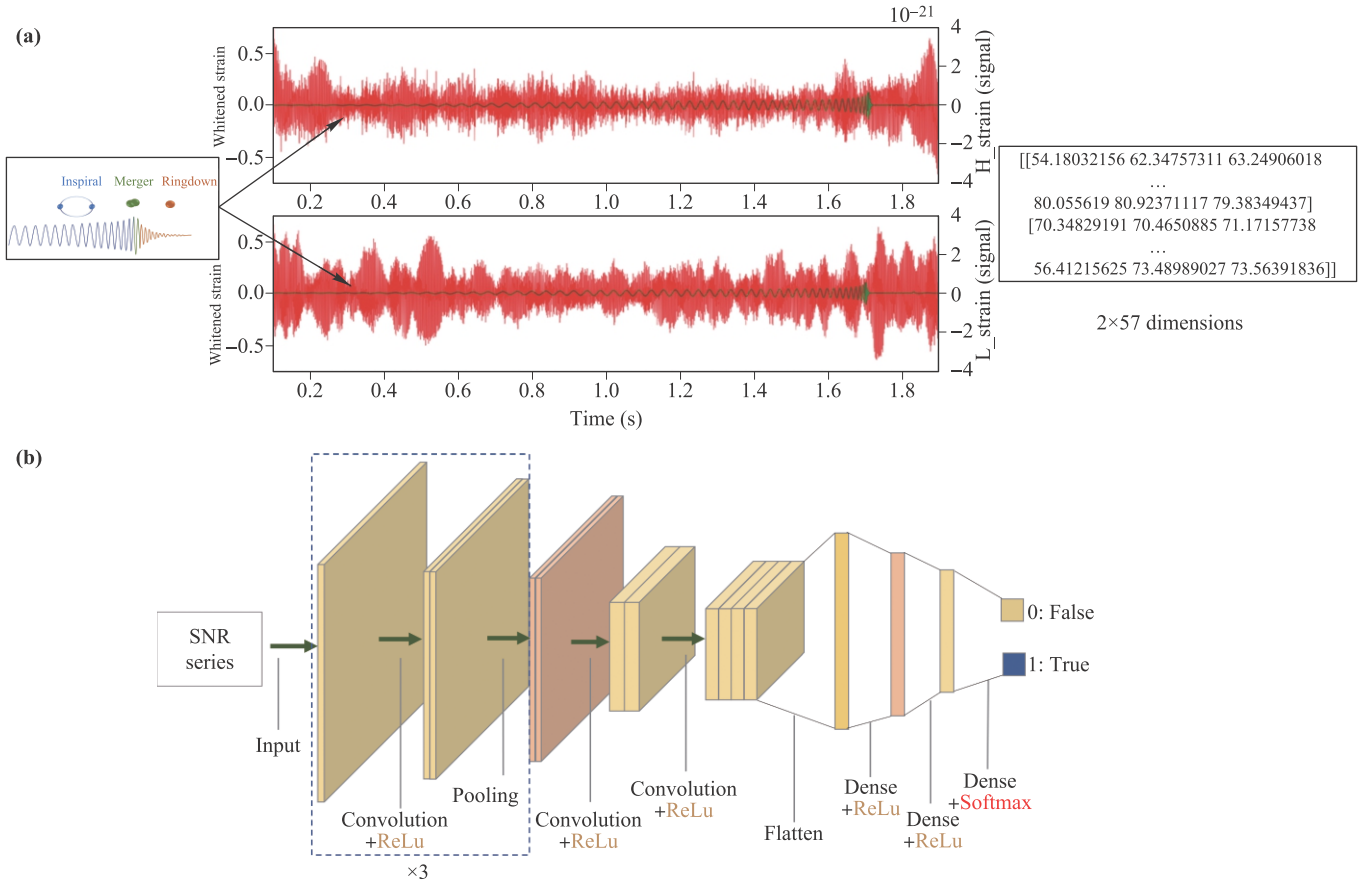

Fig. 2 The diagram of a SNR series generation and the CNN classifying process. **(a)** Strain data including signals in H and L detectors resulting in a 2×57 dimensions matched filtering SNR series of H and L detectors. **(b)** The CNN classifying process for a sample.

We use the binary cross-entropy loss function to evaluate the deviation between the label and the actual output value of training datasets. The calculated loss is back-propagated through the network to update the weights and deviations, to make the loss value reach minimum as fast as possible. In this process, we choose Adam as the optimizer for updating the weights of neural network. The diagram of the key processes are illuminated in Fig. 2. We set the batch size to be 64 and the learning rate to be 0.006, which have been tested resulting in the highest accuracy on the model [cf. Fig. 3]. At the same time, we also use the EarlyStopping module of Keras, which monitors the loss on the training and validation datasets, and forcibly terminates the training when the verified loss fails to be improved after a certain number of times.

In order to prevent overfitting, batch normalization and dropout layers are used in our CNN model. Moreover we shuffle the training datasets and extract a part of them as validation datasets at a ratio of 0.25, which is used to verify the effect of model during the training process. In each training round, all the training datasets are randomly shuffled. At the end of each round, the model is evaluated on the validation datasets, and the corresponding loss is calculated to measure the generalization of the model. In our work, all implementations of the current work are coded in Python based on the Keras framework. During the training process, the accuracy of the model reaches 98% after 200 epochs. Then inputting the testing datasets into the trained CNN, the result shows that our network can successfully detect 95% of the data. For evaluating the whole performance of the network, we calculate the corresponding true alarm probability and false alarm probability [cf. the Receiver Operating Characteristic (ROC) curves in Fig. 3]. The figure shows that when the batchsize is chosen to be 64, the accuracy is best given the FAR. When batchsize equals to 64 and the optimizer is Adam, the highest accuracy is achieved.

4 Results and analysis

Applying the network to analyze the real GW events, we use the data from GWTC-1 and GWTC-2 detected by H and L. Our classification results have listed in Table 1, which shows that all the signals have been high likelihood as positive samples. It means our network can classify the

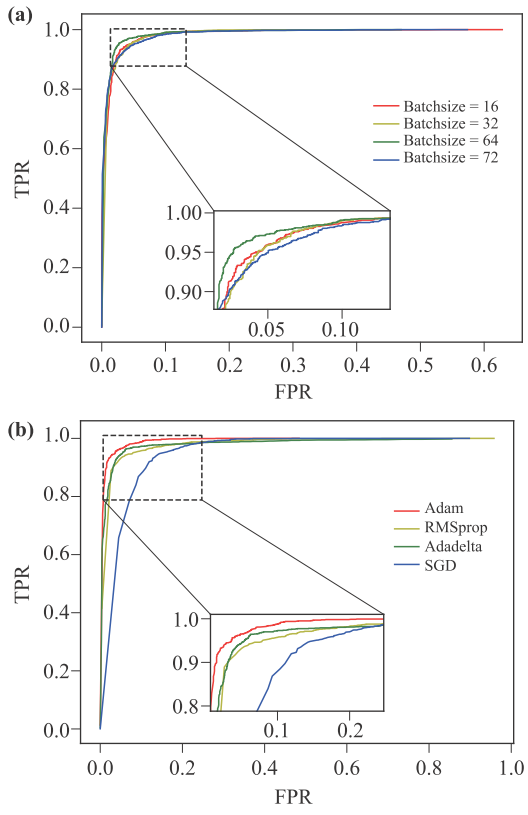

Fig. 3 The ROC curves for the network classification with different batchsizes **(a)** and optimizers **(b)**.

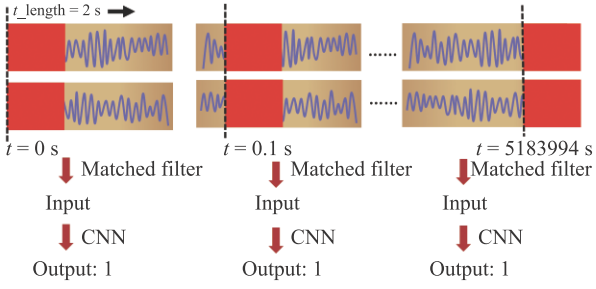

Fig. 4 This figure shows the schematic diagram of time sliding method. The red slider in the figure represents the sliding window. The window length is 2 seconds. The step can be set to be $\Delta t = 0.1, 0.2, 0.4$. The window slides on the H and L detectors at the same time, which is matched with the template bases to calculate the SNR series, and then it is applied to the CNN network for testing.

real data including GW events accurately.

However, from Fig. 3, the FAR continues to rise after ensuring the accuracy. In order to reduce the FAR of events as much as possible and improve the confidence of GW detection, we use TimeSlide to another 1 month strain data without GW events in O2 run. The specific steps are as follows:

Step 1: Set a 2-second sliding window and the initial sliding step to be 0.1 s.

Table 1 The classification results of the latest 41 GW events from BBH mergers. Here p is the probability value described by Eq. (2).

Event	p	Event	p
GW150914	0.9999634	GW190421_213856	0.9950676
GW151012	0.9999993	GW190408_181802	0.9984047
GW151226	0.9220735	GW190503_185404	0.9887100
GW170104	0.9999992	GW190512_180714	0.9999985
GW170608	0.9752168	GW190513_205428	0.9947618
GW170729	0.9959053	GW190514_065416	0.9999822
GW170809	0.9987486	GW190517_055101	0.9999939
GW170814	0.9682219	GW190519_153544	0.9999993
GW170818	0.9971951	GW190521_074359	0.9999612
GW170823	0.9946213	GW190527_092055	0.9997705
GW190521	0.9999878	GW190602_175927	0.9981902
GW190412	0.5045002	GW190701_203306	0.9999898
GW190413_134308	0.9986176	GW190413_052954	0.5045002
GW190706_222641	0.9979181	GW190707_093326	0.9991100
GW190719_215514	0.9999954	GW190720_000836	0.9999867
GW190727_060333	0.9999378	GW190728_064510	1
GW190731_140936	1	GW190803_022701	0.9988020
GW190828_063405	0.8163405	GW190828_065509	0.9999771
GW190909_114149	0.9984491	GW190915_235702	0.9999991
GW190924_021846	0.9999932	GW190929_012149	0.9999999
GW190706_222641	0.9979181	GW190706_222641	0.9979181

Step 2: Use above sliding window to intercept 1 month successive strain data in O2 run. Take $0 \to 2$ s as the first set, then slide the window for 0.1 s. Take $0.1 \to 2.1$ s as the second set. Continue the window sliding, and take $0.2 \to 2.2$ s as the third set, and so on, until the whole strain data are slided (note: strian data from H and L detectors are operated simultaneously). Figure 4 shows the operation intuitively. All the sets constitute new testing datasets.

Step 3: Then apply our network (with trained weights) to the new testing datasets (here $p_c = 0.5$), and output a series of "0" and "1".

Step 4: Test the outputs. Set n_p as the consecutive number of "1" and n_c as the threshold value of n_p for judging a positive sample. Once $n_p \geq n_c$ appears, it is considered to be a signal. Record the number of false alarmed signals varing with n_c.

Step 5: Change the sliding step size of the window, and repeat steps 1 to 4. Then record the number of false alarmed signals varing with n_c.

Figure 5 illuminates the number of false alarmed samples in the 2 months testing strain data and the real GW events classified correctly varing with n_c. It can be found that when n_c chosen to be 4 and time sliding step to be 0.4 s, the number of false alarmed signals will be diminished significantly, meanwhile our network is able to identify 41 real GW events correctly. Furthermore, the coincidence of loud noise in multiple detectors

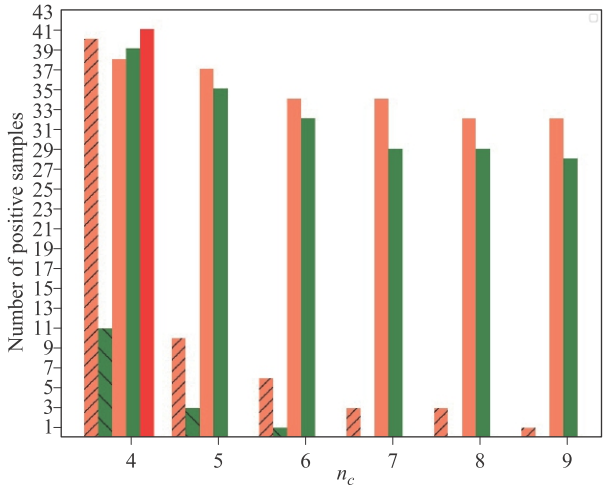

Fig. 5 The number of alarmed positive samples varing with n_c. The different colors represent the cases with different time sliding steps (orange: $\Delta t = 0.1$, green: $\Delta t = 0.2$, red: $\Delta t = 0.4$), and the shaded bars with diagonal lines represent the number of false alarms samples in testing datasets. The bars with pure colors show the number of real GW events classified correctly in the way.

should be vetoed. Therefore the time-series of one detector are shifted in time with respect to the other detector by a time much longer than the expected time interval of the GW signal (which should be 10 ms for H and L) and look for the coincidences. Focus on the samples judged as positive by the above steps, the following operations should be executed: (i) shift the time-series of H and L with 100 ms, 1000 ms, 1 s; (ii) repeat the above steps 1–5 and fix $n_c = 4$, $\Delta t = 0.4$; (iii) veto the samples still judged as positive. Finally the false alarm ratio can be controlled to be <1 in 2 months.

5 Conclusions and supplements

As a target search, we attempt to confine the FAR with the threshold value of the consecutive positive number n_c =4, time sliding $\Delta t = 0.4$ s, and reduce it by further time sliding between the two detectors to veto the coincidence of loud noise. The corresponding results show it is available to limit the FAR to be less than 1 in 2 months. The basic thought comes from matched filtering, but the matching is just operated for the specific positive samples and time sliding only happens within a sample (2 seconds). Therefore, the time sliding is ensued, and in our whole procedure (from SNR series generation to time sliding judgment), it takes nearly 21 minutes to process a 4096 s strain data with 24 G RTX3090 GPU.

As the input of ML, the SNR distribution of a few templates has been used to replace the original time-series in this work. The main reasons are: (i) the quantity and the quality of the data are both vital to obtain a reasonable result in the process of data mining by machine learning. Therefore, with sufficient data one of the methods that are usually utilized to boost the performance of CNN is feature pre-extraction using expert knowledge. The reason for that benefit comes from the structure of deep learning by Neural Networks. The structure of CNN includes convolutional layers for feature extraction and fully connected layers for function fitting, and the feature pre-extraction can largely improve the working efficiency of convolutional layers. For instance, according to the works of Zhou and others, the feature pre-extraction is used to improve the precision of their models [48, 49]. (ii) For our case, the feature pre-extraction is also very helpful. Otherwise, one needs to input the time-series directly into the convolutional layers, and with gradient descent algorithm and tedious iterations only few valuable kernels can be found. Usually, these kernels are far from enough to obtain precise results. Furthermore, another fact is that the calculation of SNR itself is also a kind of feature extraction from time-series. Therefore, we use the feature pre-extraction strategy by replacing the first convolutional layer with 57 pre-set templates from traditional expert knowledge. From another point of view, the set of gravitational time-series can be abstracted as a linear space, and these templates are its base vectors. Thus, the convolution between a time-series and a template is the projection of this time-series on this template, and the set of those projects on all of templates represents the feature of this time-series.

Applying ML to analyze the real data of detector is a hotspot of GW observation. As a preliminary study, using simulated GW signals of BBH with real noise to train a CNN model can effectively identify all the existing GW signals of BBH and find some candidates. Considering the real noise is non-stationary and non-Gaussian, the robustness of the trained CNN is guaranteed through performing strong recognition ability for signals with different noises. Meanwhile although the training datasets exclude the BBH with spin, the CNN still find this GW events. To a certain extent, it shows the generalization of ML.

Acknowledgements We are thankful to postdoctoral of He Wang for many helpful discussions. This work was supported by the National Key Research and Development Program of China (Grant No. 2021YFC2203004), the National Natural Science Foundation of China (Grant No. 11873001), and Key Research Program of Xingtai 2020ZC005.

References

1. B. P. Abbott, R. Abbott, T. D. Abbott, et al., Observation of gravitational waves from a binary black hole merger, *Phys. Rev. Lett.* 116(6), 061102 (2016)
2. J. Liu, G. Wang, Y. M. Hu, T. Zhang, Z. R. Luo, Q. L. Wang, and L. Shao, GW150914 and gravitational-wave astronomy, *Chin. Sci. Bull.* 61(14), 1502 (2016)

3. B. P. Abbott, R. Abbott, T. D. Abbott, et al., GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs, *Phys. Rev. X* 9(3), 031040 (2019)
4. B. P. Abbott, R. Abbott, T. D. Abbott, et al., GW170817: Observation of gravitational waves from a binary neutron star inspiral, *Phys. Rev. Lett.* 119(16), 161101 (2017)
5. B. P. Abbott, R. Abbott, T. D. Abbott, et al., GWTC-1: A gravitational-wave transient catalog of compact binary mergers observed by LIGO and Virgo during the first and second observing runs, *Phys. Rev. X* 9(3), 031040 (2019)
6. A. H. Nitz, C. Capano, A. B. Nielsen, S. Reyes, R. White, D. A. Brown, and B. Krishnan, 1-OGC: The first open gravitational-wave catalog of binary mergers from analysis of public advanced LIGO data, *Astrophys. J.* 872(2), 195 (2019)
7. R. Abbott, T. D. Abbott, S. Abraham, et al., GW190412: Observation of a binary-black-hole coalescence with asymmetric masses, *Phys. Rev. D* 102(4), 043015 (2020)
8. R. Abbott, T. D. Abbott, S. Abraham, et al., GW190814: Gravitational waves from the coalescence of a 23 solar mass black hole with a 2.6 solar mass compact object, *Astrophys. J.* 896, L44 (2020)
9. B. P. Abbott, R. Abbott, T. D. Abbott, et al., GW190425: Observation of a compact binary coalescence with total mass $\sim 3.4\ M_{\odot}$, *Astrophys. J.* 892, L3 (2020)
10. R. Abbott, T. D. Abbott, S. Abraham, et al., GW190521: A binary black hole merger with a total mass of $150\ M_{\odot}$, *Phys. Rev. Lett.* 125(10), 101102 (2020)
11. R. Abbott, et al., GWTC-2: Compact binary coalescences observed by LIGO and Virgo during the first half of the third observing run, *Phys. Rev. X* 11, 021053 (2021)
12. L. S. Finn, Detection, measurement, and gravitational radiation, *Phys. Rev. D* 46(12), 5236 (1992)
13. K. Cannon, R. Cariou, A. Chapman, M. Crispin-Ortuzar, N. Fotopoulos, M. Frei, C. Hanna, E. Kara, D. Keppel, L. Liao, S. Privitera, A. Searle, L. Singer, and A. Weinstein, Toward early-warning detection of gravitational waves from compact binary coalescence, *Astrophys. J.* 748(2), 136 (2012)
14. S. A. Usman, A. H. Nitz, I. W. Harry, C. M. Biwer, D. A. Brown, M. Cabero, C. D. Capano, T. D. Canton, T. Dent, S. Fairhurst, M. S. Kehl, D. Keppel, B. Krishnan, A. Lenon, A. Lundgren, A. B. Nielsen, L. P. Pekowsky, H. P. Pfeiffer, P. R. Saulson, M. West, and J. L. Willis, The PyCBC search for gravitational waves from compact binary coalescence, *Class. Quantum Grav.* 33(21), 215004 (2016)
15. B. P. Abbott, R. Abbott, T. D. Abbott, et al., Observing gravitational-wave transient GW150914 with minimal assumptions, *Phys. Rev. D* 93(12), 122004 (2016)
16. I. Harry, S. Privitera, A. Bohe, and A. Buonanno, Searching for gravitational waves from compact binaries with precessing spins, *Phys. Rev. D* 94(2), 024012 (2016)
17. R. Smith, S. E. Field, K. Blackburn, C. J. Haster, M. Purrer, V. Raymond, and P. Schmidt, Fast and accurate inference on gravitational waves from precessing compact binaries, *Phys. Rev. D* 94(4), 044031 (2016)
18. A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM* 60, 84C90 (2017)
19. J. Schmidhuber, Deep learning in neural networks: An overview, *Neural Netw.* 61, 85 (2015)
20. I. Goodfellow, Y. Bengio, and A. Courville, Deep Learning, MIT Press, 2016
21. I. Kononenko, Machine learning for medical diagnosis: History, state of the art and perspective, *Artif. Intell. Med.* 23(1), 89 (2001)
22. M. Pirooznia, J. Y. Yang, M. Q. Yang, and Y. Deng, A comparative study of different machine learning methods on microarray gene expression data, *BMC Genomics* 9(S1), S13 (2008)
23. G. Allen, et al., Deep learning for multi-messenger astrophysics: A gateway for discovery in the big data era, arXiv: 1902.00522 (2019)
24. D. George and E. Huerta, Deep neural networks to enable real-time multimessenger astrophysics, *Phys. Rev. D* 97(4), 044039 (2018)
25. M. Chen, Y. H. Zhong, Y. Feng, D. Li, and J. Li, Machine learning for nanohertz gravitational wave detection and parameter estimation with pulsar timing array, *Sci. China Phys. Mech. Astron.* 63(12), 129511 (2020)
26. C. Escamilla-Rivera, M. A. C. Quintero, and S. Capozziello, A deep learning approach to cosmological dark energy models, *J. Cosmol. Astropart. Phys.* 03, 008(2020)
27. D. George, H. Shen, and E. A. Huerta, Classification and unsupervised clustering of LIGO data with deep transfer learning, *Phys. Rev. D* 97(10), 101501 (2018)
28. M. Razzano and E. Cuoco, Image-based deep learning for classification of noise transients in gravitational wave detectors, *Class. Quantum Grav.* 35, 095016 (2018)
29. E. A. Huerta, D. George, Z. Z. Zhao, and G. Allen, Real-time regression analysis with deep convolutional neural networks, arXiv: 1805.02716 (2018)
30. D. George and E. A. Huerta, Deep learning for real-time gravitational wave detection and parameter estimation: Results with advanced LIGO data, *Phys. Lett. B* 778, 64 (2018)
31. G. Allen, I. Andreoni, E. Bachelet, G. B. Berriman, F. B. Bianco, et al., Deep learning for multi-messenger astrophysics: A gateway for discovery in the big data era, arXiv: 1902.00522 (2019)
32. H. Y. Shen, E. A. Huerta, Z. Z. Zhao, E. Jennings, and H. Sharma, Statistically-informed deep learning for gravitational wave parameter estimation, *Mach. Learn. Sci. Tech.* 3, 015007 (2022)
33. C. Chatterjee, L. Wen, K. Vinsen, M. Kovalam, and A. Datta, Using deep learning to localize gravitational wave sources, *Phys. Rev. D* 100(10), 103025 (2019)
34. H. Wang, S. Wu, Z. Cao, X. Liu, and J. Y. Zhu, Gravitational-wave signal recognition of LIGO data by deep learning, *Phys. Rev. D* 101(10), 104003 (2020)

35. X. Liu, Z. Cao, and L. Shao, Validating the effective-one-body numerical-relativity waveform models for spin-aligned binary black holes along eccentric orbits, *Phys. Rev. D* 101(4), 044049 (2020)
36. Z. J. Cao and W. B. Han, Waveform model for an eccentric binary black hole based on the effective-one-body-numerical-relativity formalism, *Phys. Rev. D* 96(4), 044028 (2017)
37. W. Wei and E. A. Huerta, Gravitational wave denoising of binary black hole mergers with deep learning, *Phys. Lett. B* 800, 135081 (2020)
38. X. R. Li, W. L. Yu, X. L. Fan, and G. J. Babu, Some optimizations on detecting gravitational wave using convolutional neural network, *Front. Phys.* 15(5), 54501 (2020)
39. J. A. González and F. S. Guzman, Characterizing the velocity of a wandering black hole and properties of the surrounding medium using convolutional neural networks, *Phys. Rev. D* 97(6), 063001 (2018)
40. B. J. Lin, X. R. Li, and W. L. Yu, Binary neutron stars gravitational wave detection based on wavelet packet analysis and convolutional neural networks, *Front. Phys.* 15(2), 24602 (2020)
41. M. Zevin, S. Coughlin, S. Bahaadini, E. Besler, N. Rohani, S. Allen, M. Cabero, K. Crowston, A. K. Katsaggelos, S. L. Larson, et al., Gravity spy: Integrating advanced ligo detector characterization, machine learning, and citizen science, *Class. Quantum Grav.* 34, 064003 (2017)
42. J. C. Driggers, S. Vitale, A. P. Lundgren, M. Evans, K. Kawabe, and E. A. Dwyer, Improving astrophysical parameter estimation via offline noise subtraction for advanced LIGO, *Phys. Rev. D* 99(4), 042001 (2019)
43. G. Vajente, Y. Huang, M. Isi, J. C. Driggers, J. S. Kissel, M. J. Szczepanczyk, and S. Vitale, Machine-learning non-stationary noise out of gravitational-wave detectors, *Phys. Rev. D* 101(4), 042003 (2020)
44. A. Torres-Forné, E. Cuoco, J. A. Font, and A. Marquina, Application of dictionary learning to denoise LIGO's blip noise transients, *Phys. Rev. D* 102(2), 023011 (2020)
45. W. Wei, A. Khan, E. A. Huerta, X. Huang, and M. Tian, Deep learning ensemble for real-time gravitational wave detection of spinning binary black hole mergers, *Phys. Lett. B* 812, 136029 (2021)
46. J. D. Alvares, J. A. Font, F. F. Freitas, O. G. Freitas, A. P. Morais, et al., Exploring gravitational-wave detection and parameter inference using deep learning methods, *Class. Quant. Grav.* 38, 155010 (2021)
47. P. G. Krastev, K. Gill, V. A. Villar, and E. Berger, Detection and parameter estimation of gravitational waves from binary neutron-star mergers in real LIGO data using deep learning, *Phys. Lett. B* 815, 136161 (2021)
48. B. Zhou, et al., Learning Deep Features for Discriminative Localization, 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 27–30 June, 2016, Las Vegas, NV, IEEE, 2016, pp 2921
49. S. Liu, A. J. Davison, and E. Johns, Self-Supervised Generalisation with Meta Auxiliary Learning, NIPS, 2019