

Concat Convolutional Neural Network for pulsar candidate selection

Qingguo Zeng,¹ Xiangru Li^{2*} and Haitao Lin¹

¹*School of Mathematical Sciences, South China Normal University, No. 55 West of Yat-sen Avenue, Guangzhou 510631, China*

²*School of Computer Science, South China Normal University, Guangzhou 510631, China*

Accepted 2020 March 30. Received 2020 March 29; in original form 2020 February 21

ABSTRACT

Pulsar searching is essential for the scientific research in the field of physics and astrophysics. With the development of the radio telescope, the exploding volume and growth speed of candidates have brought about several challenges. Therefore, there is an urgent demand for developing an automatic, accurate, and efficient pulsar candidate selection method. To meet this need, this work designed a Concat Convolutional Neural Network (CCNN) to identify the candidates collected from the Five-hundred-meter Aperture Spherical Telescope (FAST) data. The CCNN extracts some ‘pulsar-like’ patterns from the diagnostic subplots using Convolutional Neural Network (CNN) and combines these CNN features by a concatenate layer. Therefore, the CCNN is an end-to-end learning model without any need for any intermediate labels, which makes CCNN suitable for the online learning pipeline of pulsar candidate selection. Experimental results on FAST data show that the CCNN outperforms the available state-of-the-art models in a similar scenario. In total, it misses only 4 real pulsars out of 326.

Key words: methods: data analysis; pulsars: general.

1 INTRODUCTION

Pulsars are rapidly rotating, superdense neutron stars with strong magnetic fields. The rotation of the pulsar causes the beam of electromagnetic radiation field to sweep in and out of our line of sight with an extremely regular period. The theory and observation of the pulsars are of great significance to promote the development of physics and astrophysics, such as the evolution of neutron stars (Helfand & Huang 1987), the equation of state of dense matter (Backer et al. 1982), verification on general relativity (Hulse & Taylor 1975; Lyne et al. 2004), etc. In particular, pulsar timing array (PTA) with dozens of millisecond pulsars can be used to detect and analyse gravitational waves due to their accurate timing properties (van Haasteren et al. 2011; Demorest et al. 2012; Manchester et al. 2013). Therefore, it is essential to discover new pulsars to excavate their enormous potentials for scientific research.

Ever since Jocelyn Bell Burnell and Antony Hewish observed the first pulsar in 1967 (Hewish et al. 1968), more than 2700 pulsars have been discovered (Manchester et al. 2005) by the modern radio telescope survey projects, such as the Parkes Multi-beam Pulsar Survey (PMPS; Manchester et al. 2001), the Pulsar Arecibo L-band Feed Array (PALFA; Deneva et al. 2009) survey, LOw-Frequency ARray (LOFAR) Tied-Array All-Sky Survey (LOTASS; Coenen et al. 2014), etc. However, astronomers prophesied that the total number of potentially observable pulsars in the Galaxy should

be approximately 10 times more than this based on the pulsar population model (Faucher-Giguere & Kaspi 2006). To search for more pulsars, some advanced modern radio telescopes will or have been built, such as the Square Kilometre Array (SKA; Smits et al. 2009) and Five-hundred-meter Aperture Spherical radio Telescope (FAST; Nan et al. 2011). Specifically, FAST began to be constructed in 2011 and started formal operations on 2020 January 11 (Mingmei 2020). It is expected to discover about 1500 new normal pulsars and 200 millisecond pulsars (Yue, Li & Nan 2012). In practice, the FAST 19-beam drift-scan survey generates more than one million pulsar candidates per night (Wang et al. 2019b). However, the proportion of real pulsars among candidates is exceedingly small (approximately 1 in 10 000; Lyon et al. 2013) due to the presence of radio frequency interference (RFI) and noise. Therefore, it is seldom to select the pulsars among the candidates just by using simple metrics like the signal-to-noise (S/N) ratio. Traditionally, pulsar candidates selection is completed through inspecting diagnostic plots of the candidates by human experts, but it is impractical to deal with such extreme volume of candidates in this way. In other words, there exist urgent demands for developing an automatic, accurate, and efficient pulsar candidate selection method.

The goal of the pulsar candidate selection is to minimize the retention of the non-pulsar signals without missing pulsar candidates as much as possible, thereby reducing the labor of further observations. In the past few years, a variety of pulsar candidate selection methods have been proposed. Based on the principles of a method, they can be divided into three categories. The first category is of traditional scoring methods. Lee et al. (2013) ranked the candidates

* E-mail: xiangru.li@gmail.com

according to their scores, which are the linear combinations of six well-designed quality factors. The second category improved the methods by applying machine learning (ML) algorithms to learn how to combine the pre-designed quality factors (usually called features in ML) instead of the artificial combination (Eatough et al. 2010; Bates et al. 2012; van Leeuwen et al. 2013; Morello et al. 2014; Lyon et al. 2016). In these methods, pulsar candidate selection was served as a binary classification problem. One of the important factors affecting the classification result is the feature design that relies heavily on human experience. An incomprehensive feature design scheme may have a bad effect on the performance of the models. For example, some methods extracted six features just from the pulse profile and dispersion measure (DM) curve. As a result, it is likely to mistakenly identify some RFI candidates as pulsars. These misclassified candidates are often generated by RFI within several frequency channels so that they have the ‘pulsar-like’ appearance in both the pulse profile and DM curve. In practice, human experts can identify the pulsars from the candidates just by observing the diagnostic plots successfully. Under this inspiration, the third category attempts to directly utilize the diagnostic plots as the inputs into the model instead of hand-crafted features (Zhu et al. 2014; Guo et al. 2019; Wang et al. 2019a,b). These methods prompt the model to learn the ‘pulsar-like’ patterns from the diagnostic subplots by themselves through data-driven learning. Zhu et al. (2014) and Wang et al. (2019b) proposed a two-layer ensemble model to identify the pulsars. For example, the model in Wang et al. (2019b) is composed of five classifiers totally, including two Residual Neural Networks (ResNets), two Support Vector Machines (SVMs), and one Logistic regression (LR). Two ResNets are used to determine whether the time versus phase plot and frequency versus phase plot are ‘pulsar-like’, respectively. Two SVMs evaluate how ‘pulsar-like’ the pulse profile and DM curve are, respectively. Finally, the LR classifies the candidates based on the output scores from the first four classifiers. The first four classifiers constitute a layer of data processors and this layer is referred to as the first layer. The LR constitutes the second layer, which receives the outputs from the first layer. *However, the label of the first layer (the labels of each diagnostic subplot) may not be in accordance with the candidates’ label. In other words, the subplots of some RFI candidates may be the same as those of the pulsars. As a result, we have to manually label whether each of the four subplots is ‘pulsar-like’ individually for every training data, leading to a lot of extra labour.*

In this work, we propose a novel deep learning scheme, Concat Convolutional Neural Network (CCNN), for the pulsar candidate selection based on Convolutional Neural Network (CNN). In this proposed model, a concatenate layer is introduced to replace the second layer of the PICS or PICS-ResNet for overcoming the problem of the non-correspondence between the candidate’s nature (the candidate is pulsar or not) and the labels of each diagnostic subplot (the subplot is ‘pulsar-like’ or not). In addition, the CCNN extracts the features from four diagnostic subplots only using the CNN: one-dimensional (1D) CNN for the pulse profile and DM curve while two-dimensional (2D) CNN for the time versus phase plot and frequency versus phase plot. In application, 2D-CNN has shown its outstanding ability to deal with image pattern recognition and, at the same time, 1D-CNN has been proved that it is adept at signal processing and recognition (Huang et al. 2019). Therefore, the CNN for extracting features rather than the traditional ML models has a great potential to promote the performance of the model to identify the pulsar with the diagnostic subplots as the input into the model. This scheme belongs to

an end-to-end learning model. The complex relationship between the target (the identification results of the candidates) and the inputs (four diagnostic subplots) can be directly described by just one single layer without any intermediate processes or intermediate labels. Therefore, the end-to-end learning makes the CCNN suitable for the online learning pipeline of the pulsar candidate selection. By the way, for an online learning pipeline, the newly confirmed candidates can be directly appended to the training data set to continuously improve the classification accuracy of the model.

The rest of this paper is organized as follows: the experimental data and data pre-processing methods are described in the next section. In Section 3, we presented the components and detailed structure of the CCNN. According to the direction of the concatenate operation, CCNN can be subdivided into Horizontal CCNN (H-CCNN) and Vertical CCNN (V-CCNN). Their performances are investigated and compared with the available methods in Section 4. We conclude and discuss the future work of pulsar candidate selection for FAST in the final section.

2 DATA

The work is conducted for the Commensal Radio Astronomy Fast Survey (CRAFTS; Li et al. 2018). CRAFTS is a drift-scan survey that aims at observing the entire visible sky of the FAST for H I emission and search for the new pulsars utilizing the FAST L-band Array of 19 beams (FLAN; Zhang et al. 2019). The early observation data with labels (Wang et al. 2019b) for pulsar searching from CRAFTS has been public on https://github.com/dzuwhf/FAST_label_data. This work uses this data set to train and test our model.

2.1 The information of the data set

The data set has been split into the training set and test set. The training set consists of 837 real pulsars and 998 RFI candidates, and these samples will be utilized to construct the classification model for pulsar candidate selection. At the same time, the performance of the model will be evaluated on the test set that contains 326 pulsar samples and 13 321 RFI samples.

Each sample is processed by PRESTO (Pulsar Exploration and Search Toolkit; Ransom 2001; Ransom, Eikenberry & Middleditch 2002), which is a typical software for pulsar search and analysis. After that, the dedispersed and folded three-dimensional (3D; time interval, phase, channel frequency) data are stored in a pfd format file as well as some data descriptions. Summing the data along the frequency channels and time intervals generates the time versus phase plot and frequency versus phase plot. Meanwhile, summing the data along both the time intervals and frequency channels generates the pulse profile histogram. In addition, the last diagnostic subplot is the DM curve, which is a plot of the DM trials against the corresponding reduced χ^2 values. Fig. 1 presents the diagnostic subplots of a pulsar candidate and a non-pulsar candidate, respectively. For a real pulsar, there should be usually one or more vertical lines in the time versus phase plot and frequency versus phase plot, which indicates a broadband and pulsed signal lasted during the observation time. At the same time, the profile usually contains one or more peaks and the DM curves should peak at a non-zero value. In general, these four diagnostic subplots constitute the fundamental information for the experts to classify the candidates. As a result, they serve as inputs into our model.

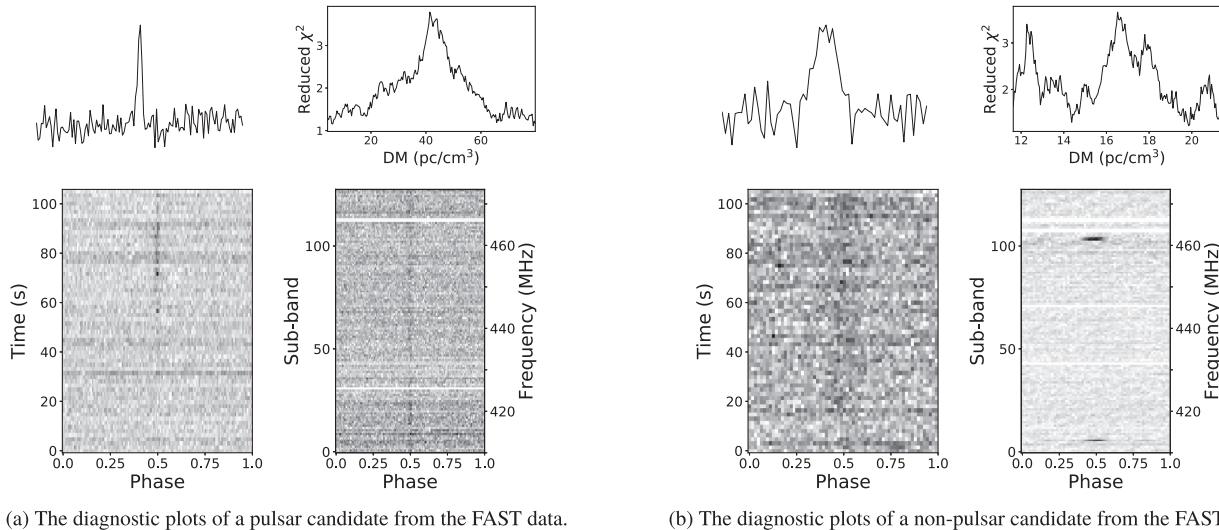


Figure 1. Two examples of the pulsar and non-pulsar candidates. For the pulsar candidate, there is a narrow peak in the pulse profile plot and a persistent vertical line both in the time versus phase plot and the frequency versus phase plot. And meanwhile, DM curve peaks at a non-zero value. For the non-pulsar candidate, there are a broad peak in the pulse profile plot, and what is more, the pulse only appears in several frequency channels, which indicates that this signal is the RFI.

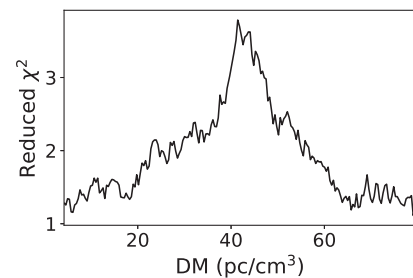
2.2 Data preprocessing

Before feeding the diagnostic subplots into the model, we have to process the data since there is inconsistency among the candidates, such as the size, scale, and so on. These inconsistent factors are useless for identifying the pulsar candidates. What is worse, they have the potential to make negative effects on the training process and performance of the model. Therefore, it is necessary to eliminate these factors before training the model.

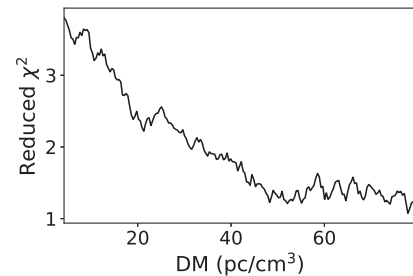
Considering the phase-related bias resulting from the peak far away from the centre of the plot (Zhu et al. 2014), we shift the strongest peak to the centre phase within the subplots except for the DM curve since the position of the peak is an important pattern for pulsar candidate selection. As a result, the model can pay more attention to the presence of the patterns regardless of their position, which is not the necessary factor for identification.

Four diagnostic subplots are all saved as 1D or 2D data arrays, but the size of these arrays vary from candidate to candidate for a certain type of subplots. For the majority of ML algorithms, the size of the inputs should be fixed. Therefore, we have to resize the data arrays to a uniform size: 64 for the pulse profile, 64×64 for the time versus phase plot and frequency versus phase plot, and 200 for the DM curve. The plots whose size is smaller than the uniform size are interpolated and those with larger size are scrunched instead of being downsampled to avoid losing important information. In addition, we normalize the data so that they range from 0 to 1 by using min–max normalization. The normalization can accelerate the convergence of the gradient descent during training (Ioffe & Szegedy 2015). On the other hand, normalization does not do any harm to the performance of the model in theory because we just want to extract some certain patterns (e.g. peaks, stripes, and so on) from the plots, regardless of the exact values in curves or images.

In order to focus the attention of the concatenate layer on the difference between the real pulsar and non-pulsar candidates, we generate some negative samples by replacing only one of the subplots of the pulsar candidates from the training data set with a corresponding ‘non-pulsar subplot’. For example, we firstly choose a pulsar candidate from the training data set randomly. Secondly,



(a) The original DM curve



(b) The generating DM curve

Figure 2. An example for generating a new false positive sample by modifying the DM curve coming from a pulsar candidate, while other subplots remains unchanged.

the DM curve is modified by removing the part of the curve before the peak and interpolate the rest part of the curve to the uniform size without modifying the other diagnostic subplots (e.g. Fig. 2). In this way, the generating DM curve peaks at the zero and the newly generated sample belongs to the non-pulsar category. In practice, we can generate the new non-pulsar candidate by modifying any one of the diagnostic subplots of the pulsar candidates except the

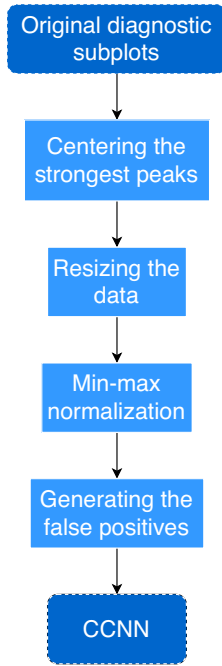


Figure 3. The flowchart of the preprocessing.

pulse profile since it can be obtained by summing the frequency versus phase plot over the time intervals.

In summary, the flowchart of the data preprocessing is shown in Fig. 3. After that, the processed diagnostic subplots are served as the inputs into the proposed CCNN.

3 MODEL

In this section, the related fundamental components of CCNN are first reviewed, including the fully connected layer, convolutional layer, global pooling layer, and then the whole architecture of the model is introduced in detail.

3.1 Fully connected layer

Fully connected layer is a basic component of the Artificial Neural Network (ANN), which is inspired by the biological neural networks, the fundamental element of animal brains (Chen et al. 2019). An example architecture of this type of layer is shown in Fig. 4.

Generally, a fully connected layer can be considered to be a function composed of some simple mathematical operations, which receive an array $\mathbf{x} \in \mathcal{R}^D$ as the input and output another array $\mathbf{h} \in \mathcal{R}^N$. In detail, all the neurons in the input layer are multiplied by weights (also called synapses) and summed together and then transformed via an activation function (Haykin 1998). Then, the i th output can be expressed as

$$h_i = \sigma(\mathbf{w}^\top \mathbf{x} + b_i), \quad (1)$$

where $\mathbf{w} \in \mathcal{R}^d$ is a weight vector that shows the importance of each input neuron, $b_i \in \mathcal{R}$ is a bias value that allows to shift the function up or down, and $\sigma(\cdot)$ is an activation function that decides whether a neuron should be activated or not. Its motivation is to introduce non-linearity into the output of a neuron. And then the neural network is able to learn and represent the complex relationship between input

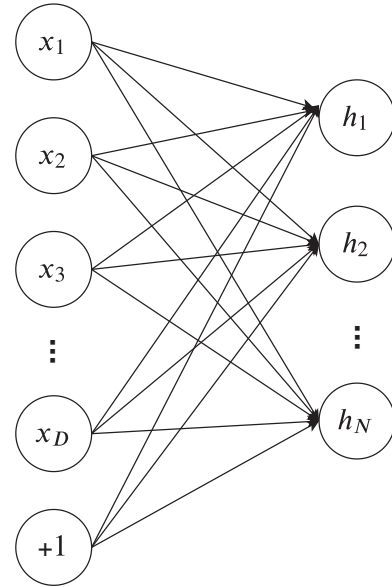


Figure 4. An example architecture of the fully connected layer. The left-hand layer represents its input layer and the right-hand layer denotes the output layer. The lines connecting these two layers with the arrows pointing from the left- to right-hand side indicates the weights.

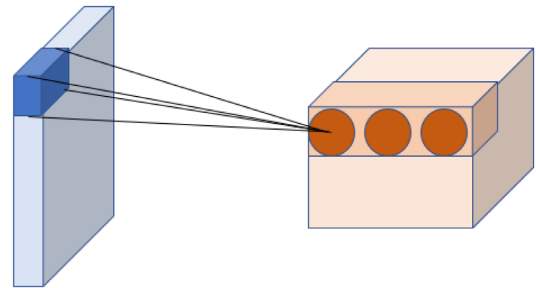


Figure 5. An example architecture of the 2D convolutional layer. The light blue cube represents the input feature map, the blue small cube denotes the filter, and the light orange cube represents the output feature maps. Each neuron in the output layer is obtained by multiplying the corresponding elements of the input feature maps and filter.

data and output target. The activation is triggered by a high similarity between the input data and the patterns stored in weights. Actually, the patterns are unknown or are difficult to be exactly described by humans. Therefore, the weights are initialized randomly. In order to find the correct patterns between the input data and their expected output, it is necessary to adjust the weights. The process of weights adjustment is referred to as learning (Yegnanarayana 1994). The goal of learning is to minimize the difference between the neural network outputs and their expected labels. Usually, this process is implemented by a back-propagation (BP) algorithm (Rumelhart, Hinton & Williams 1986).

The structure of the model (e.g. the number of the neurons of fully connected layer and the type of activation function) is chosen by the user and is built by using a simplest one to precisely describe the relationship between the inputs and targets in order to avoid overfitting (Sarle 1996).

3.2 Convolutional layer

Zhu et al. (2014) creatively introduced 2D-CNN to the pulsar candidate selection. The design of 2D-CNN (shown in Fig. 5) is inspired by the work of Hubel and Wiesel who discovered that the cats' visual cortexes contain neurons that individually respond to edges and bars of particular orientations within a small region of the visual field (Hubel & Wiesel 1959). The ability of neurons to recognize patterns is unaffected by position shifts.

The working mechanism of the convolutional layer is like one uses a flashlight to slide over a big image from the left- to right-hand side, and top to bottom. Technically, this flashlight is referred to as filters, which actually is a collection of weight vectors, and the area shot by the flashlight is called the receptive field. The output from a neuron is obtained by multiplying the elements of the filter with the image pixels the values in the filter with the original pixel values of the image within the corresponding receptive field and adding all these multiplications together. The activation of the output neuron is triggered if the particular pattern (e.g. the peak and the stripes in diagnostic subplots) are detected from the corresponding receptive field. When the sliding is over, the feature maps (or called activation maps) are obtained. The different filters are used to detect different and simple patterns. As the network going deeper, the patterns extracted by CNN become more complex.

In this work, we not only use the 2D-CNN to extract patterns on the time versus phase plot and frequency versus phase plot, but also process the pulse profile and DM curve using 1D-CNN. The 1D-CNN has been widely applied in time-series data, 1D astronomical signals, etc. Pearson, Palafox & Griffith (2018) show the prominent ability of the 1D-CNN, and Zhu et al. (2014) demonstrate the outstanding performance of 2D-CNN for the time versus phase plot and frequency versus phase plot in pulsar candidate selection. Therefore, the application of 1D-CNN for the other subplots instead of the traditional ML models has the potential of improving the performance of pulsar candidate selection theoretically.

3.3 Global pooling layer

There is usually a pooling layer in the back of a convolutional layer. The intuitive reasoning behind the pooling layer is that the emphasis of the CNN is to detect the existence of some specific patterns within the image regardless of their exact positions for a classification task. The application of the pooling layer contributes to reducing the number of the parameters thereby decreasing the computational costs. On the other hand, it can effectively alleviate the overfitting problem.

Pooling layer downsamples feature maps by summarizing each map. Two common pooling methods are average pooling and max pooling. They summarize the average presence of a feature and the most activated presence of a feature, respectively.

In this work, we apply the global pooling, which samples the entire feature map to a single value instead of down sampling patches of the input feature map as the traditional pooling operation does. On the one hand, the global pooling further reduced the number of the training parameters to improve the calculating speed and mitigate overfitting. On the other hand, after the traditional pooling layer, there is usually a reshape operation that transforms the multidimensional arrays into the one-dimensional vectors before being input into the fully connected layer. This reshape operation may destroy the spatial information in the feature maps. In contrast, the global pooling is more native to the convolution structure, and Lin, Chen & Yan (2013) have demonstrated that the global

pooling has a better performance in some classification tasks than the traditional pooling.

3.4 Concat Convolutional Neural Network

In this work, we use four CNNs to extract features respectively from four diagnostic subplots: two 1D-CNNs, respectively, for the pulse profile and DM curve, and two 2D-CNNs, respectively, for the time versus phase plot and frequency versus phase plot. Considering that the extracted features, such as peaks or the vertical stripes, have some simple pattern, the CNN for each subplot in this work consists of only three convolutional layers and is followed by a global max pooling layer to summarize the information of the feature maps and output a 1D vector. And then, a concatenate layer is applied to merge the information coming from four different diagnostic subplots. The proposed scheme is referred to as CCNN, and this work used two examples of the CCNN: H-CCNN (Fig. 6a) and V-CCNN (Fig. 6b).

The difference between H-CCNN and V-CCNN is their concat type and subsequent layers. In detail, the former model concatenates the four vectors extracted from the four diagnostic subplots one after another in a horizontal direction to form a long 1D vector, while the latter concatenates them in a vertical direction to generate a 2D matrix. After that, the final two layers of H-CCNN are the fully connected layer and the activation of the last one is a sigmoid function. The sigmoid activation function computes the probability of a candidate being pulsar. The subsequent layers of V-CCNN are several convolutional layers and one global average pooling layer that averages each feature maps, and the resulting vector of the global average pooling layer is fed into a softmax layer (Lin et al. 2014). Two neurons in the output layer stand for the probabilities that the candidate is of pulsar and non-pulsar, respectively. The choice of activation function for the last layer and the parameters for the convolutional layers of H-CCNN and V-CCNN were determined by their F_1 score performance. This work investigated the sigmoid and softmax functions for finding the appropriate activation. And the output of sigmoid function is a real value, denoted by p for convenience, between 0 and 1. This value represents the probability of a candidate being pulsar. Therefore, the probability of the candidate being a non-pulsar can be calculated by $1 - p$. Brief architectures of CCNN (H-CCNN and V-CCNN) are presented in Fig. 6.

The configuration of CCNN (e.g. the number and size of the filters, the type of the global pooling, the number of the neurons in the fully connected layer, and the type of the activation function, etc.) is determined by grid search based on 10-fold cross-validation (James et al. 2013). First, the training data set is shuffled randomly and split into 10 groups. Each unique group is taken as the validation set one after another and the corresponding remaining data are served as the training set that is used to train the CCNN. The performance is evaluated by computing the F_1 score on the validation set. Finally, the hyperparameters are determined based on the model with the highest F_1 score. As a result, the optimal structure of CCNN is shown in Fig. 6, and the model is trained by an Adam optimizer (Kingma & Ba 2014) with a learning rate of 0.001 and batch size of 64.

The CCNN is implemented using KERAS (Chollet et al. 2015) with the TENSORFLOW backend (Abadi et al. 2015).¹ KERAS is a high-level neural network API written in PYTHON and it focuses on enabling fast experimentation instead of coding ability. This

¹<https://github.com/xrli/CCNN/>

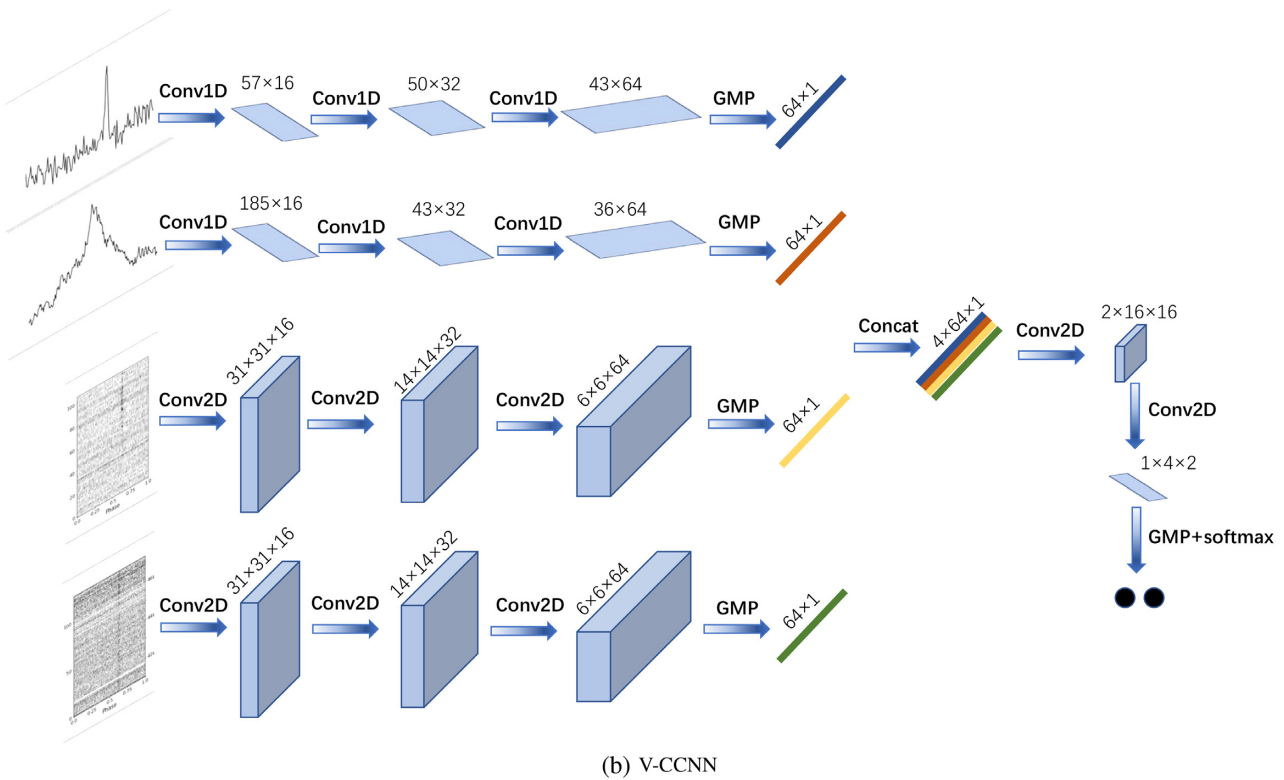
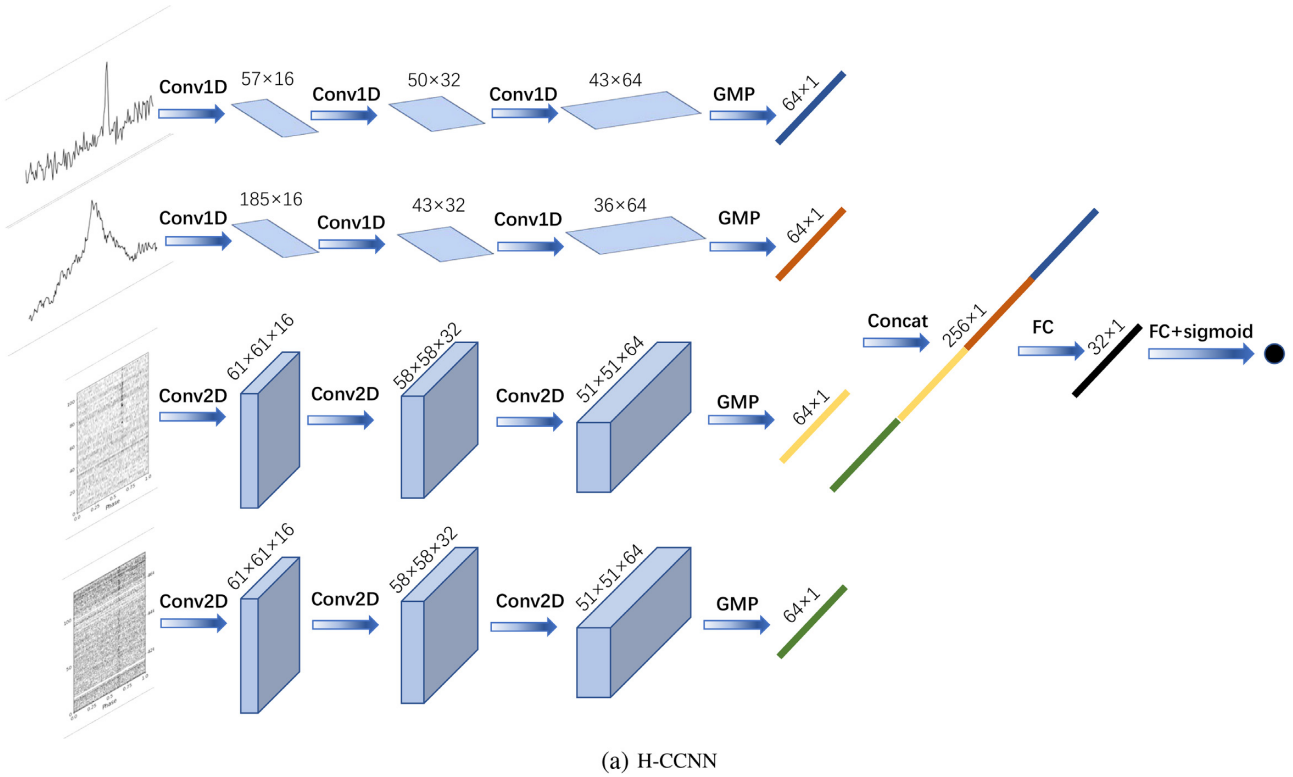


Figure 6. The architecture of CCNN. The input subplots of CCNN from the top to bottom panel are individually the pulse profile, DM curve, the frequency versus phase plot, and the time versus phase plot whose sizes are 64×200 , 64×64 , and 64×64 , respectively. The output sizes of the 1D convolutional layer are $L \times N$ and the output sizes of the 2D convolutional layer are $H \times W \times N$, where L denotes the length of the tensors, H and W are the height and width of the tensors, and N is the number of the feature maps. The output layer of the H-CCNN and V-CCNN are different due to the dissimilar operation in concatenate layer and their subsequent layers. The former model outputs the probability of a candidate being pulsar, while the output layer of the latter model contains two neurons, respectively, representing the probabilities of a candidate being pulsar and non-pulsar. GMP means global max pooling layer, Concat represents the concatenate layer and FC is the abbreviation of fully connected layer.

characteristic of KERAS makes it suitable for the process and analysis of the astronomical data.

4 EXPERIMENTAL INVESTIGATION

To investigate the effectiveness of the CCNN, some quantitative evaluations and comparisons are conducted on FAST observations for pulsar candidate selection. In this section, we first introduce evaluation metrics, and then present and analyse the experimental results.

4.1 The evaluation metrics

In pulsar candidate selection, the pulsars and their harmonic signals are served as positive samples, while the remaining non-pulsars are regarded as negative samples. Therefore, this problem can be considered as a binary classification task. The common evaluation metrics for the binary classification problem include the Accuracy, Precision, Recall, and F_1 score. All the metrics utilized in this work are defined as follows:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (2)$$

$$\text{Precision} = \frac{TP}{TP + FP}, \quad (3)$$

$$\text{Recall} = \frac{TP}{TP + FN}, \quad (4)$$

$$F_1 \text{ score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \quad (5)$$

where TP, FP, TN, and FN, respectively, denote the number of true positive (both ground-true label and prediction are pulsars), false positive (ground-true label is the non-pulsar and prediction is pulsar), true negative (both ground-true label and prediction are non-pulsar), and false negative (ground-true label is pulsar and prediction is non-pulsar).

Accuracy indicates the fraction of the correct predictions on the whole. Precision and Recall are inversely proportional, respectively, to the FP and FN. Therefore, the Precision and Recall indicate the severity of false detection and missed detection, respectively. The false positives will waste labor and time for the further observation while the false negatives make negative effects on the search for the new pulsars. Considering the test set is heavily imbalanced, F_1 score is also used to assess the performance of the model (Jeni, Cohn & De La Torre 2013). F_1 score is defined as the harmonic mean of Precision and Recall and is served as a trade-off between them. All of the metrics range in [0, 1]. The smaller the FPR and FNR, the better the RFI detection scheme. In contrast, a higher value of accuracy and F_1 score are more satisfying.

4.2 The experiment on FAST data

The experimental data collected from the FAST drift-scan survey have been split into the training set and test set when they were public, where there are only 1835 candidates in total, among which 837 are pulsars or their harmonics and 998 are non-pulsars.

For a deep learning model, less training data are more likely to make it overfitting. In statistics, overfitting refers to an analysis that matches a particular and known data set exactly but fails to fit the

other data well or predict future observations satisfying (Leinweber 2007). In ML, an overfitting model usually possesses too many parameters or heavily complex structures for the limited data (Tetko, Livingstone & Luik 1995). As the epochs of training increases, the performance of an over-fitting model keeps improving on the training set but will degrade after a period of growth on the test set. To overcome this problem, we adopt early stopping during the training process. First, the original training set is randomly split into two groups: 80 per cent for training and 20 per cent for performance validation. Secondly, the performance of the model is evaluated on the validation set at the end of each epoch. At the same time, the loss at the current epoch is compared to the previous one or that of the saved model; the model with smaller loss will be saved. And then, stopping the training process will be triggered when the loss score of the validation set increases for successively five times.

Fig. 7 shows the trends of the loss and accuracy during the training. It is worth noting that we plot the whole training process with 100 epochs to facilitate the observation and, in fact, the training process has been early stopped at the 46th epoch. The trends of the loss and accuracy curves of both the H-CCNN and V-CCNN share similar characteristics. Therefore, we just take that of H-CCNN for a detailed analysis. The changing trends of the loss curves are consistent with the above analysis, which indicates that both H-CCNN and V-CCNN become over-fitting as the training continues after the trigger-point. Therefore, it is necessary to use early stopping during the training process for alleviating the problem of over-fitting. The curves in Fig. 7 indicate that the model becomes over-fitting after the 46th epoch. The validation accuracy curve peaks at the 41th epoch and keeps oscillating after that, which means that the model achieves its best performance at this epoch and it is saved as the final model used to make the classification on the test set. And two final models are compared to the state-of-the-art ML models: PICS (Zhu et al. 2014) and PICS-ResNet (Wang et al. 2019b), which were trained in the previous works and have been public.²

The results of these models tested on the FAST data are presented in Table 1. On the whole, H-CCNN achieves the best performance on the FAST data, especially on the Accuracy, Precision, and F_1 score. In detail, the recall of the H-CCNN, 0.9635, is a higher than that of PICS and is relatively less than that of PICS-ResNet, 0.9816. However, H-CCNN achieves over 10 percentage points precision higher than PICS as well as PICS-ResNet. It is because that PICS mistakenly identifies 863 non-pulsars as the pulsar signals while the number of the false positives of H-CCNN is approximately half of that. It means that the ability of PICS and H-CCNN to correctly identify the real pulsar signals among all the candidates collected from FAST are almost the same but, at the same time, H-CCNN is far better at excluding false candidates than PICS when identifying the pulsars. As a result, using H-CCNN for pulsar candidate selection in the practical application can reduce the labor and expense for the further observations. On the other hand, V-CCNN achieves the highest recall among all the classifiers and misses only four pulsars in all. In addition, we average the scores from the final H-CCNN and V-CCNN, then the mean scores are served as the final score of H-CCNN+V-CCNN. This operation is served as ensemble learning in the field of ML. Ensemble learning combines the results coming from multiple different learning algorithms to obtain a better performance than that of any single component alone (Rokach 2010). In this way, H-CCNN+V-CCNN inherits high precision

²https://github.com/dzuwhf/PICS-ResNet/tree/master/ubc_AI

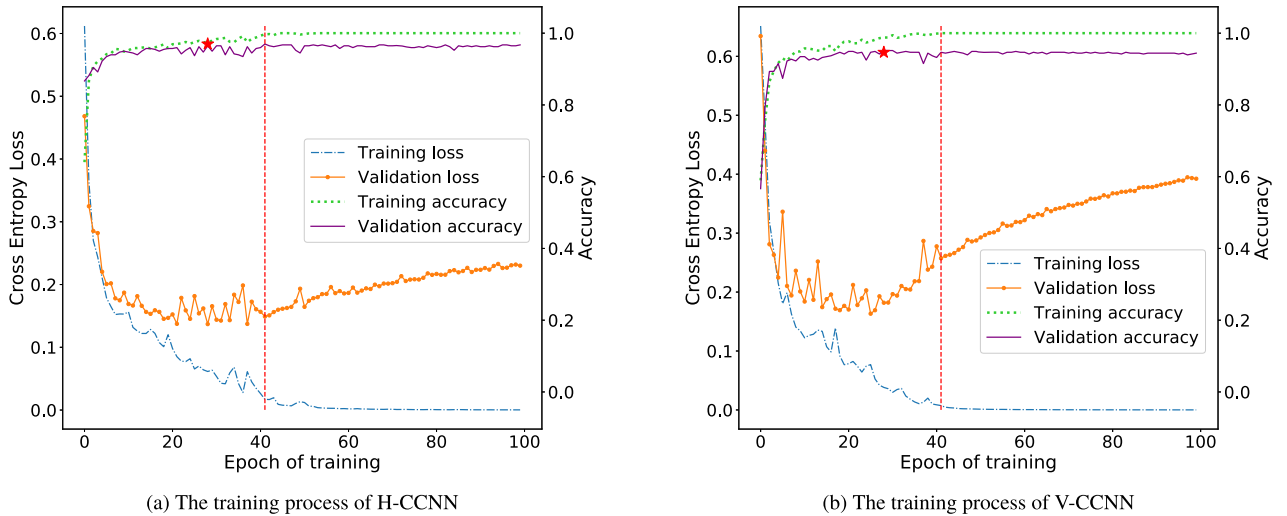


Figure 7. The loss and accuracy curves of training and validation set during the training. The vertical dashed line means the trigger point of early stopping after which the validation loss increases almost continuously. The star-like marker shows the best performance of the model on validation set and locates the point before the trigger point of early stopping.

Table 1. The evaluation results of four classifiers on the FAST data set.

Model	Training data set (no. of training samples)	Accuracy	Precision	Recall	F_1 score	No. of missing pulsars
PICS	FAST + HTRU + FALFA (13 632)	0.9357	0.2649	0.9540	0.4146	15
PICS-ResNet	FAST + HTRU + FALFA (13 632)	0.9332	0.2612	0.9816	0.4126	6
H-CCNN	FAST (1835)	0.9634	0.3920	0.9632	0.5572	12
V-CCNN	FAST (1835)	0.9173	0.2227	0.9877	0.3634	4
H-CCNN+V-CCNN	FAST (1835)	0.9476	0.3110	0.9816	0.4723	6

Notes. The first two models were trained by Guo et al. (2019). The final model, ‘H-CCNN+V-CCNN’, is the embedding model of H-CCNN and V-CCNN. The boldface digits indicate the best performance.

from H-CCNN and high recall from V-CCNN. Therefore, the results show that all the metrics of H-CCNN+V-CCNN are higher than that of PICS, and are equal to or higher than that of PICS-ResNet. In summary, CCNN outperforms the PICS and the PICS-ResNet on the FAST data in general. It is worth noting that CCNN is trained only on FAST data while the PICS and PICS-ResNet are trained in three data sets, including PALFA, HTRU, and FAST. In theory, the more training samples for a deep neural network, the better the performance of the network. Therefore, it is likely to improve the performance of the CCNN if the other two data sets are available for training.

4.3 The analysis of the missing pulsars

Despite the prominent performance of CCNN on the FAST data, the missing pulsars of it are still required to be concerned about. In general, CCNN (V-CCNN) mistakenly classified four pulsars as the RFI. The diagnostic subplots of them are presented in Fig. 8, which guides us to analysing the characteristics of the missing pulsars in detail.

In summary, there are relatively obvious ‘pulsar-like’ patterns in the diagnostic subplots of all the missing pulsars. However, there is different confusing information in the subplots at the same time. They are summarized as follows:

(i) The presence of the RFI: for these missing pulsars, the RFI can mainly be observed in the time versus phase plot and frequency

versus phase plot. The strong and persistent one (e.g. the time versus phase plot in Fig. 8b) almost overshadows the pulsar signals. The periodic interference causes the oblique lines in the time versus phase plot (Fig. 8a) and the oblique lines in the frequency versus phase plot are caused by the RFI with zero DM (Fig. 8c). They are all likely to make some negative effects for CCNN to extract the pulsar signals.

(ii) Intensity variation: the signal intensity varies over time, and in some normalized subplots, the signals look like to be disappeared sometimes (e.g. the time versus phase plot in Figs 8c and d). This phenomenon is caused by the rotation of the beam pattern during the observation.

5 CONCLUSIONS

A novel model for pulsar candidate selection, CCNN, is proposed in this paper. Different from the existing ML models for the pulsar candidate selection, CCNN extracts the features from four diagnostic subplots all utilizing the CNN rather than traditional ML algorithms. Besides, the concatenate layer in CCNN has the ability to automatically learn the combination of the features extracted from the subplots. The introduction of this layer avoids labelling every subplot manually, which significantly reduces labor and expense. Finally, experiments of the CCNN on the FAST data are compared to state-of-the-art models, and CCNN outperforms them in summary.

Although CCNN has achieved outstanding results on the FAST data, several improvements and extensions still can be made in

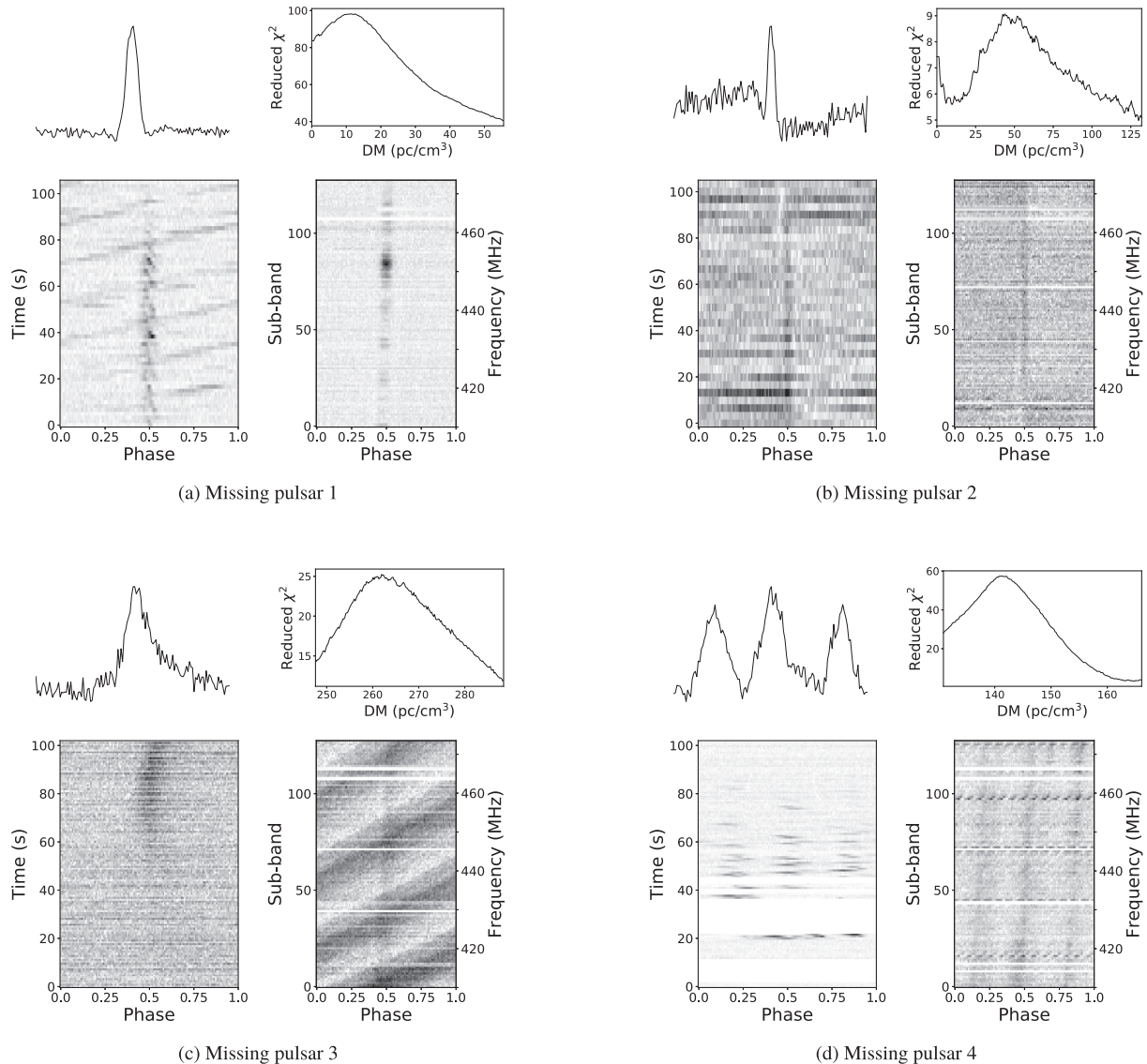


Figure 8. The diagnostic subplots of the missing pulsars. There are four diagnostic subplots for each pulsar: the upper left-hand plot is the profile, the upper right-hand plot is DM curve, the bottom left-hand plot is the time versus phase plot, and the bottom right-hand plot is the frequency versus phase plot.

future work in theory, to make the model more useful and powerful, as follows:

(i) Data collection and labelling: all the classifiers mentioned in this paper tend to mistakenly identify some non-pulsars as the pulsar signal so that they all suffer from a relatively low precision that increases the labor and expense for the further observations. Collecting additional labelled data for training is an effective way to improve the performance of CCNN for pulsar candidate selection on the FAST data.

(ii) Abandon resizing the input data: data resizing includes interpolation for the small sized data and scrunching for the large sized data. As a result, the former operation is likely to bring about the useless or even wrong information into the original diagnostic subplots while the latter may discard some important patterns within the subplots, leading to inaccurate identification for the pulsar candidates. Actually, the global pooling makes the CCNN have the ability to accept the input data with arbitrary size in theory. This

characteristic of CCNN will be detailedly discussed in our next work.

(iii) Robust learning: from the analysis of the missing pulsar, we can conclude that the ‘pulsar-like’ patterns all appear in the diagnostic subplots but are overshadowed by the interference to varying degrees. In ML, this type of data is referred to as noisy data (Zhu & Wu 2004). Robust learning is an effective way to address this problem since it has the ability to mitigate the negative effects coming from the noise. Therefore, robust learning methods can be added to the data preprocessing or the construction of models.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China (grant Nos 11973022, U1811464), the Natural Science Foundation of Guangdong Province (2020A1515010710), and the Joint Research Fund in Astronomy (U1531242) under cooperative

agreement between the National Natural Science Foundation of China (NSFC) and Chinese Academy of Sciences (CAS).

REFERENCES

- Abadi M. et al., 2015, TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems, Available at: <https://www.tensorflow.org/about/bib>
- Backer D. C., Kulkarni S. R., Heiles C., Davis M., Goss W., 1982, *Nature*, 300, 615
- Bates S. et al., 2012, *MNRAS*, 427, 1052
- Chen Y.-Y., Lin Y.-H., Kung C.-C., Chung M.-H., Yen I., 2019, *Sensors*, 19, 2047
- Chollet F. et al., 2015, Keras, Available at: <https://keras.io/getting-started/faq/>
- Coenen T. et al., 2014, *A&A*, 570, A60
- Demorest P. B. et al., 2012, *ApJ*, 762, 94
- Deneva J. et al., 2009, *ApJ*, 703, 2259
- Eatough R. P., Molkenthin N., Kramer M., Noutsos A., Keith M., Stappers B., Lyne A., 2010, *MNRAS*, 407, 2443
- Faucher-Giguere C.-A., Kaspi V. M., 2006, *ApJ*, 643, 332
- Guo P. et al., 2019, *MNRAS*, 490, 5424
- Haykin S., 1998, *Neural Networks: A Comprehensive Foundation*. Macmillan, New York
- Helfand D. J., Huang J.-H., 1987, *Proc. IAU Symp. 125, The Origin and Evolution of Neutron Stars*. Kluwer, Dordrecht, p. 273
- Hewish A., Bell S. J., Pilkington J. D., Scott P. F., Collins R. A., 1968, *Nature*, 217, 709
- Huang S., Tang J., Dai J., Wang Y., 2019, *Sensors*, 19, 2018
- Hubel D. H., Wiesel T. N., 1959, *J. Physiol.*, 148, 574
- Hulse R. A., Taylor J. H., 1975, *ApJ*, 195, L51
- Ioffe S., Szegedy C., 2015, preprint ([arXiv:1502.03167](https://arxiv.org/abs/1502.03167))
- James G., Witten D., Hastie T., Tibshirani R., 2013, *An Introduction to Statistical Learning*. Springer, New York
- Jeni L. A., Cohn J. F., De La Torre F., 2013, *Humaine Assoc. Conf. Affective Comput. Intell. Interact., Facing Imbalanced Data – Recommendations for the Use of Performance Metrics*, IEEE Computer Society, Washington DC, p. 245
- Kingma D. P., Ba J., 2014, preprint ([arXiv:1412.6980](https://arxiv.org/abs/1412.6980))
- Lee K. et al., 2013, *MNRAS*, 433, 688
- Leinweber D. J., 2007, *J. Investing*, 16, 15
- Li D. et al., 2018, *IEEE Microw. Mag.*, 19, 112
- Lin M., Chen Q., Yan S., 2014, preprint ([arXiv:1312.4400](https://arxiv.org/abs/1312.4400))
- Lyne A. G. et al., 2004, *Science*, 303, 1153
- Lyons R., Brooke J., Knowles J., Stappers B., 2013, *IEEE Int. Conf. Syst. Man Cybern., A Study on Classification in Imbalanced and Partially-Labelled Data Streams*, IEEE, Washington DC, p. 1506
- Lyons R. J., Stappers B., Cooper S., Brooke J., Knowles J., 2016, *MNRAS*, 459, 1104
- Manchester R. N. et al., 2001, *MNRAS*, 328, 17
- Manchester R. et al., 2013, *Publ. Astron. Soc. Aust.*, 30, 17
- Manchester R. N., Hobbs G. B., Teoh A., Hobbs M., 2005, *AJ*, 129, 1993
- Mingmei, 2020, *World's largest radio telescope starts formal operation*, Available at: http://www.xinhuanet.com/english/2020-01/11/c_138696939.htm
- Morello V., Barr E., Bailes M., Flynn C., Keane E., van Straten W., 2014, *MNRAS*, 443, 1651
- Nan R. et al., 2011, *Int. J. Mod. Phys. D*, 20, 989
- Pearson K. A., Palafox L., Griffith C. A., 2018, *MNRAS*, 474, 478
- Ransom S. M., 2001, *New Search Techniques for Binary Pulsars*. Harvard University, Harvard
- Ransom S. M., Eikenberry S. S., Middleditch J., 2002, *AJ*, 124, 1788
- Rokach L., 2010, *Artifi. Intell. Rev.*, 33, 1
- Rumelhart D. E., Hinton G. E., Williams R. J., 1986, *Nature*, 323, 533
- Sarle W. S., 1996, in Meyer M. M., Rosenberger J. L., eds, *Stopped training and other remedies for overfitting*, Proc. 27th Symp. on the Interface of Computer Science and Statistics, Interface Foundation of North America, Fairfax, Virginia, p. 352
- Smits R., Kramer M., Stappers B., Lorimer D., Cordes J., Faulkner A., 2009, *A&A*, 493, 1161
- Tetko I. V., Livingstone D. J., Luik A. I., 1995, *J. Chem. Inf. Comput. Sci.*, 35, 826
- van Haasteren R. et al., 2011, *MNRAS*, 414, 3117
- van Leeuwen J. et al., 2013, *Proc. IAU Symp. 291, Neutron Stars and Pulsars: Challenges and Opportunities After 80 Years*. Kluwer, Dordrecht
- Wang Y.-C., Li M.-T., Pan Z.-C., Zheng J.-H., 2019a, *Res. Astron. Astrophys.*, 19, 133
- Wang H. et al., 2019b, *Sci. China Phys. Mech. Astron.*, 62, 959507
- Yegnanarayana B., 1994, *Sadhana*, 19, 189
- Yue Y., Li D., Nan R., 2012, in Maria T. V. T., eds, *Proc. IAU Symp. 8, FAST low frequency pulsar survey*, Kluwer, Dordrecht, p. 577
- Zhang K. et al., 2019, *Sci. China Phys. Mech. Astron.*, 62, 959506
- Zhu W. et al., 2014, *ApJ*, 781, 117
- Zhu X., Wu X., 2004, *Artif. Intell. Rev.*, 22, 177

This paper has been typeset from a $\text{\TeX}/\text{\LaTeX}$ file prepared by the author.