Computers and Electrical Engineering 73 (2019) 1–8

Contents lists available at ScienceDirect

Computers and Electrical Engineering

journal homepage: www.elsevier.com/locate/compeleceng

Pulsar candidate recognition with deep learning

Haoyuan Zhang[a,b,*], Zhen Zhao[a], Tao An[a,c], Baoqiang Lao[a], Xiao Chen[a]

[a] Shanghai Astronomical Observatory, Chinese Academy of Sciences, 200030 Shanghai, China
[b] University of Chinese Academy of Sciences, 100049 Beijing, China
[c] Key Laboratory of Radio Astronomy, Chinese Academy of Sciences, 210008 Nanjing, China

ARTICLE INFO

Article history:
Received 29 August 2018
Revised 25 October 2018
Accepted 25 October 2018

Keywords:
Pulsar candidate classification
Radio astronomy
Machine learning
Methods and techniques
Convolutional neural network
Square kilometer array

ABSTRACT

In this paper, we present a deep learning-based recognition algorithm to identify pulsars by observing data containing millions of candidates including radio frequency interference and noise sources. The dataset is obtained from the High Time Resolution Universe survey created and updated by the Parkes telescope. We investigate several effective single and combined features via simple logistic regression. To deal with the imbalanced dataset, we oversimplify the original dataset at different sampling rates, which is also one of the learning parameters. After training the pre-processed dataset via a convolutional neural network, we provide a cross-validated evaluation of all candidates. Results show that the deep-learning based recognition algorithm can identify the pulsar and radio frequency interference signals with high accuracy. The precision and recall of radio frequency interference are both 100%, and those of pulsars are 91% and 94%, respectively.

1. Introduction

Large amounts of pulsar data are typically required by astrophysicists to find statistically-significant relationships needed to find pulsars. The pulsar candidate selection problem is important and meaningful because it is an important step to find new pulsars.

Recently, machine learning methods have been widely used for pulsar candidate selection problems [1–5]. However, with the advent of the Square Kilometer Array (SKA) radio telescope, the data volume has become extremely high. On the one hand, large-volume data provides a great opportunity to find more pulsars, but on the other hand, processing big data sets can become a daunting task rather quickly. The simple reason for this is that traditional machine learning methods cannot meet the SKA data challenge. Traditional machine learning methods find patterns from features extracted from the data [6,7]. This pattern recognition step does not work effectively for pulsar data. Unlike traditional machine learning methods, deep learning methods are used to learn directly from data. The development of an accelerator technique, e.g., graphics processing units (GPU), significantly expands the capacity of deep learning methods to deal with big data. Hinton applied deep neural networks (DNN) to classification problems and obtained highly accurate results [8]. In addition to highly accurate results, processing speed is also an important factor to consider. To increase the training speed, we adopt convolutional neural networks (CNN) in pulsar identification, which have fewer parameters and are thus faster than the DNNs. In this work, we effectively use data architecture to implement learning methods directly to raw data to reduce the system error and obtain highly accurate results. Additionally, by combining the L2 regularization step with a dropout step, we ensure that our model

* Corresponding author at: Shanghai Astronomical Observatory, Chinese Academy of Sciences, 200030 Shanghai, China.
E-mail address: zhy@shao.ac.cn (H. Zhang).

https://doi.org/10.1016/j.compeleceng.2018.10.016
0045-7906/

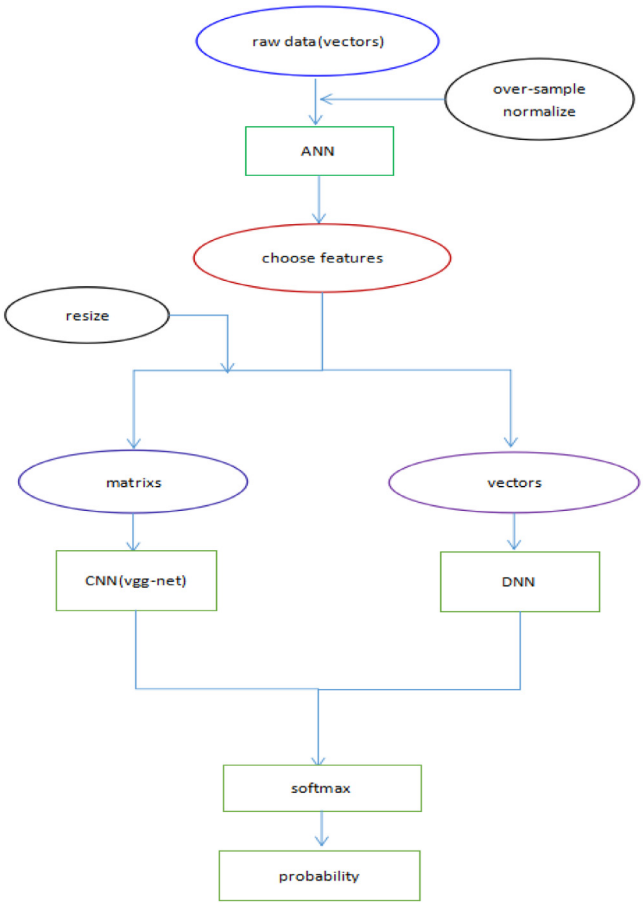

Fig. 1. The overall learning flow.

is more robust. Thus, the main contribution of this paper is the introduction of a new pulsar candidate selection method that implements deep learning methods to the system architecture to increase selection accuracy.

This paper is organized as follows: Section 2 describes the experimental setup, the learning model, and the data structure. Section 3 presents detailed data pre-processing procedures. Section 4 presents the test performance procedures and results. We analyze the results in Section 5. Section 6 summarizes the original results.

2. Methodology

Although there many examples of recent research on using machine learning and pattern recognition on space data, this work either applies a shallow artificial neural network (no more than 3 layers) or transitional statistical learning methods, such as the Naive Bayes algorithm or Ada-boost. In our work, we first pre-process High Time Resolution Universe data and then apply a deep convolutional neural network to train the recognition system. As shown in Fig. 1, the overall learning diagram mainly includes three stages: pre-processing, feature selection, and implementation of a VGG-net [9].

2.1. Learning model

One of the main advantages of CNN is that the input provided to the network can be raw data. Thus, we do not need to manually design feature vectors. The CNN can directly learn from the data by repeatedly applying convolutions to the input [7,10]. CNN has been applied to several areas, such as image classification, speech processing, and audio.

The architecture of a typical CNN is structured as an organized series of stages. Specifically, many neurons are interconnected to multiple layers of neurons via the activation function. As shown in Fig. 2, each neuron consists of multiple input variables x_i, the output variable y, the weights w_i corresponding to the inputs, and the activation function f. Neural networks typically use sigmoid or tanh as the activation function, but the Rectified Linear Unit (ReLU) can learn faster in networks that use many layers. The many layers are important because they ensure that we can build more powerful networks while maintaining the in-sample error close to zero. The first few stages are composed of two types of layers: convolutional

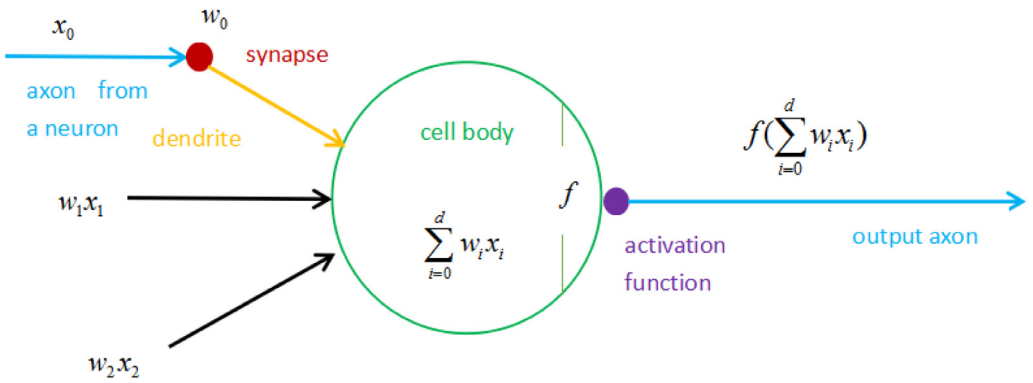

Fig. 2. The structure of single neuron.

layers, which closely follow a ReLU, and pooling layers which performs a down-sampling operation. Additionally, we use the soft-max layer as the final layer, which extends the linear prediction values to class probabilities.

In this work, we adopt a VGG-net method to classify pulsars and non-pulsars from raw candidate data. The method increases the depth by using an architecture with very small convolution filters. The raw data was organized to fit a fixed-size matrix to account for the image geometry. The fixed-size matrix (minimum 3×3) can be used as the input of convolutional networks. The input matrix is passed through a stack of convolutional layers. Five max-pooling layers follow the convolutional layers. After that, there are three fully-connected layers that follow. The last soft-max layer is connected and used as the ultimate output layer, which is used to obtain the probability that a value belongs to a pulsar.

2.2. Algorithm and regularization

$$L(w) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right], \tag{1}$$

Our error measure is calculated from the cross-entropy error via Eq. (1). The term w denotes the weight vector, and N denotes the sample number. The terms y_n and $\hat{y}_n$ represent the real and expected outputs, respectively. The learning objective is to essentially minimize the learning error. The stochastic gradient descent method is adopted to find the optimal parameters. In addition, to ensure the error is close to the in-sample error, i.e., to make our model more robust, we need to add a regularization step to avoid over-fitting. First, we add a weight decay term, also called an L2-regularizer, that adjusts the L2 penalty multiplier 10^{-4} at one time within the range from 2×10^{-4} to 10^{-3} (we find the 5×10^{-4} is the best). Furthermore, we also adopt the dropout mechanism for fully-connected layers, setting the dropout rate from 0.5 to 0.15 (we find the 0.15 is the best). The augmented error model is calculated as

$$\hat{L}(w) = -\frac{1}{N}\sum_{n=1}^{N}\left[y_n \log \hat{y}_n + (1-y_n)\log(1-\hat{y}_n)\right] + \frac{\lambda}{2}w^2, \tag{2}$$

where λ is the penalty multiplier. The Adams Optimizer is adopted to find an improved optimum by introducing momentum mechanisms and adaptive learning rates. The back-propagation is used to calculate the new gradient.

3. Data and feature descriptions

In this section, we describe the datasets we have used, the data pre-processing technique designed to solve the data imbalance issue and the effective features we have selected for training.

3.1. Data pre-processing and generation

We use a random forest algorithm to deal with any outliers. The number of positive samples is 1196 while the number of negative samples grows to 89,996. Therefore, we over-sampled the positive sample initially to avoid an imbalance of our sample. Then, we obtain a different ratio of pulsars to non-pulsars (within the range 1:1 to 1:5) to choose the best ratio (we find that a 1:2 ratio is the best). We train each network for 50 epochs with a batch size of 512. Furthermore, we normalize our raw data, which can improve the accuracy and speed of the optimal solution from the stochastic gradient descent data. Then, we use an artificial neural network (ANN) to perform a simple classification step and showcase the distributions with good features (e.g., features that can be used to combine CNN and DNN classifications). Since the VGG-net requires the

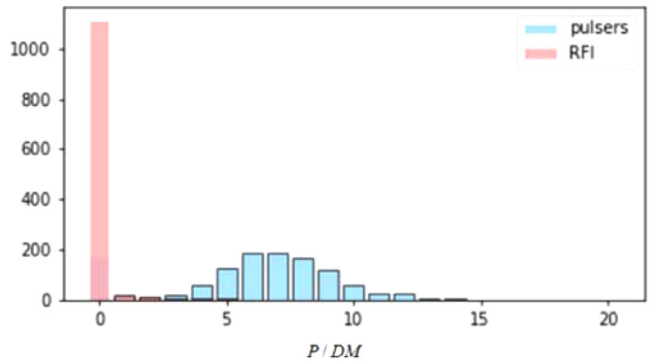

Fig. 3. Barycentric period and dispersion measure ratio.

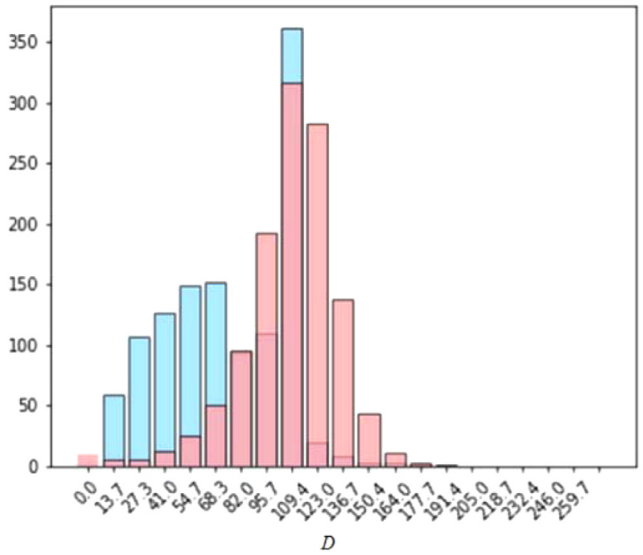

Fig. 4. Standard Deviation of the $DM-S/N$ curve.

input to be an image, after selecting the effective features, we resize the feature data into matrix-like images. The VGG-net subtracts the value used to train the data.

3.2. Selected features

We sorted these features after the ANN classification and then used good single features to construct the combined features. We used a feature subset to construct high-order features with good results. By carrying out a simple data classifier that tested the performance of various features, we considered the flow features. Fig. 3 shows the barycentric period and the dispersion measure ratio. Fig. 4 shows the standard deviation of the DM-S/N curve. Fig. 5 shows the signal and noise ratio. Fig. 6 shows the excess kurtosis of the integrated pulse file.

4. Deploying and testing

We then move our entire recognition learning system onto our own NVIDIA Tesla P100 GPU server and determine various performance metrics.

4.1. Error metrics

Due to the high-class imbalance of the original dataset, we cannot directly infer information from the classification alone. Therefore, we use confusion matrices to evaluate the effectiveness of our classifiers, as shown in Table 1.

Table 1 includes four kinds of values. True positives (*TP*) represent the number of pulsars that were correctly classified as pulsars; True negatives (*TN*) represent the number of non-pulsars that were incorrectly classified as non-pulsars; False

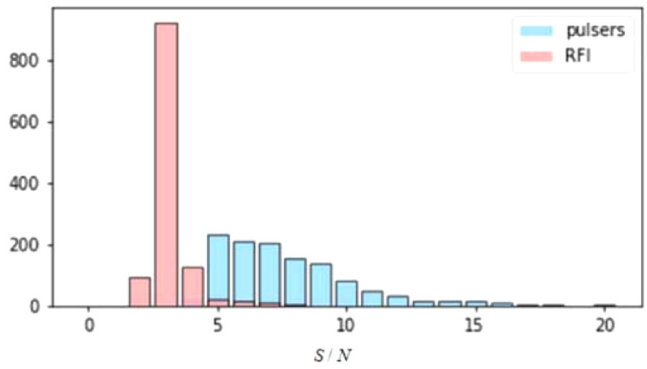

Fig. 5. Signal-to-noise ratio.

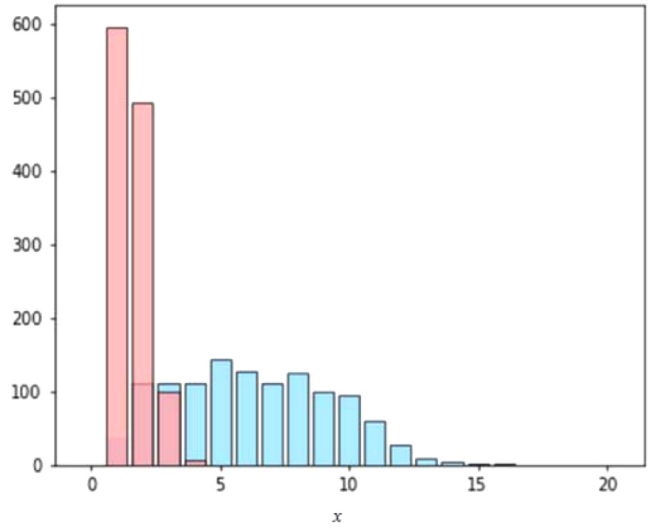

Fig. 6. Excess kurtosis of the integrated pulse file.

Table 1
Confusion matrices.

	Positive	Negative
TRUE	*TP*	*FN*
FALSE	*FP*	*TN*

negatives (*FN*) represent the number of pulsars that were classified as non-pulsars; False positives (*FP*) represent the number of non-pulsars that were not classified as pulsars. Based on these values, we evaluate the following performance metrics,

- *Recall*: This term represents the probability that our classifier correctly classifies the pulsars. The term is characterized by

$$Recall = \frac{TP}{TP + FN}.$$

- False negative rate (*FNR*): This metric represents the likelihood that our classifier incorrectly classifies the pulsars as non-pulsars. This metric is characterized by

$$FNR = \frac{FN}{FN + TP}.$$

- *Precision*: This term represents the probability that the result is correctly identified as a pulsar. The term is characterized by

$$Precision = \frac{TP}{TP + FP}.$$

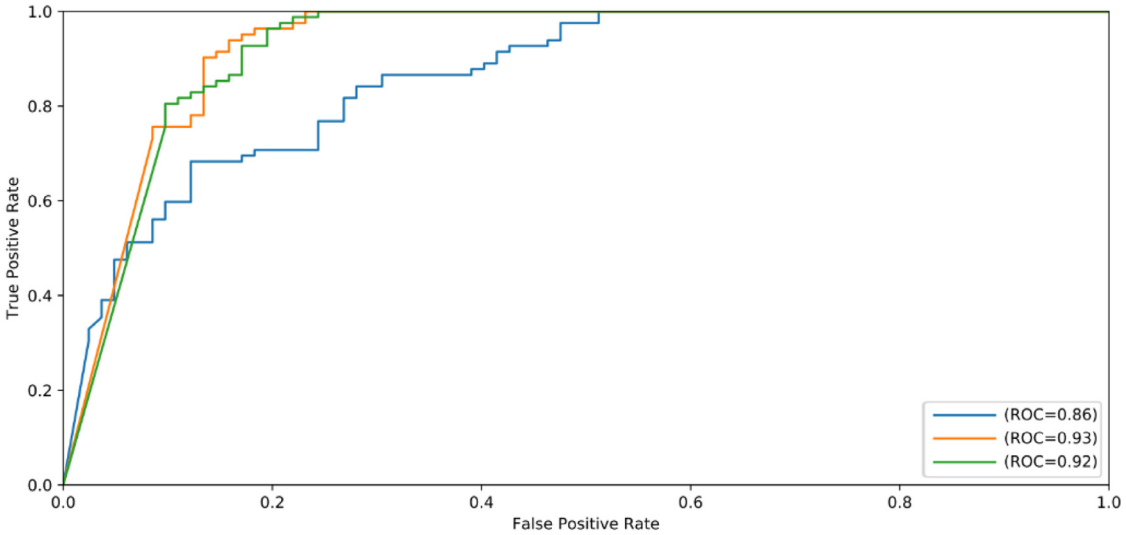

Fig. 7. Receiver operating characteristic curve. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 2
Classification report.

Label	*Precision*	*Recall*	*F*1-score	Support
0	1.00	1.00	1.00	89,996
1	0.91	0.94	0.92	1996

- False positive rate (*FPR*): This term represents the probability of incorrectly classifying non-pulsars. The term can be calculated via the expression

$$FPR = \frac{FP}{FP + TN}.$$

- *F*-score: The *F*1-score is the harmonic average of the precision and recall terms. The *F*-score can ensure that our model will not ignore certain pulsar types and will make fewer mistake by considering both recall and precision. For a good classification, the *F*1-score is close to 1. Since finding a pulsar candidate requires high computational states, we should try our best to avoid missing any pulsars. The expression for an *F*1-score is characterized by

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall}.$$

4.2. Validation

We used a ten-fold cross-validation to get 10 results with 5 different dropout values and 5 different sample ratios to evaluate our model. We randomly divided all labeled data into ten subsets. We then used nine of the subsets to train our model. We used the remaining subset to validate our model. After ten cycles of validation, we got the average value. This procedure used to make sure the validation data set is clean and connect validation error with the error out of the sample. Furthermore, we repeated the validation procedure 5 times to ensure the performance is reliable. This can make sure our model can work well on new data. Therefore, we can choose the best model by using best validation error.

5. Result and discussion

We show our results according to the following ratios: 1:1 (blue curve), 1:2 (orange curve) and 1:4 (green curve). The positive-to-negative ratio receiver operating characteristic (ROC) curves is shown in Fig. 7. The 1:2 positive-to-negative ratio is shown in Fig. 8. Fig. 8 has the best performance, lowest expected error rates and highest threshold curves. We evaluate our model via a confusion matrix as shown in Table 2.

Table 2 shows the classification performance measured during the cross-validation. Label 0 represents the non-pulsar, which has the very high precision because we have enough negative samples. Label 1 represents the pulsar that has high precision but is worse than label 0 because it is over-sampled in fewer positive samples. Our resulting increase in the positive sample suggests that our precision will improve. Therefore, our deep learning technique works well in classification.

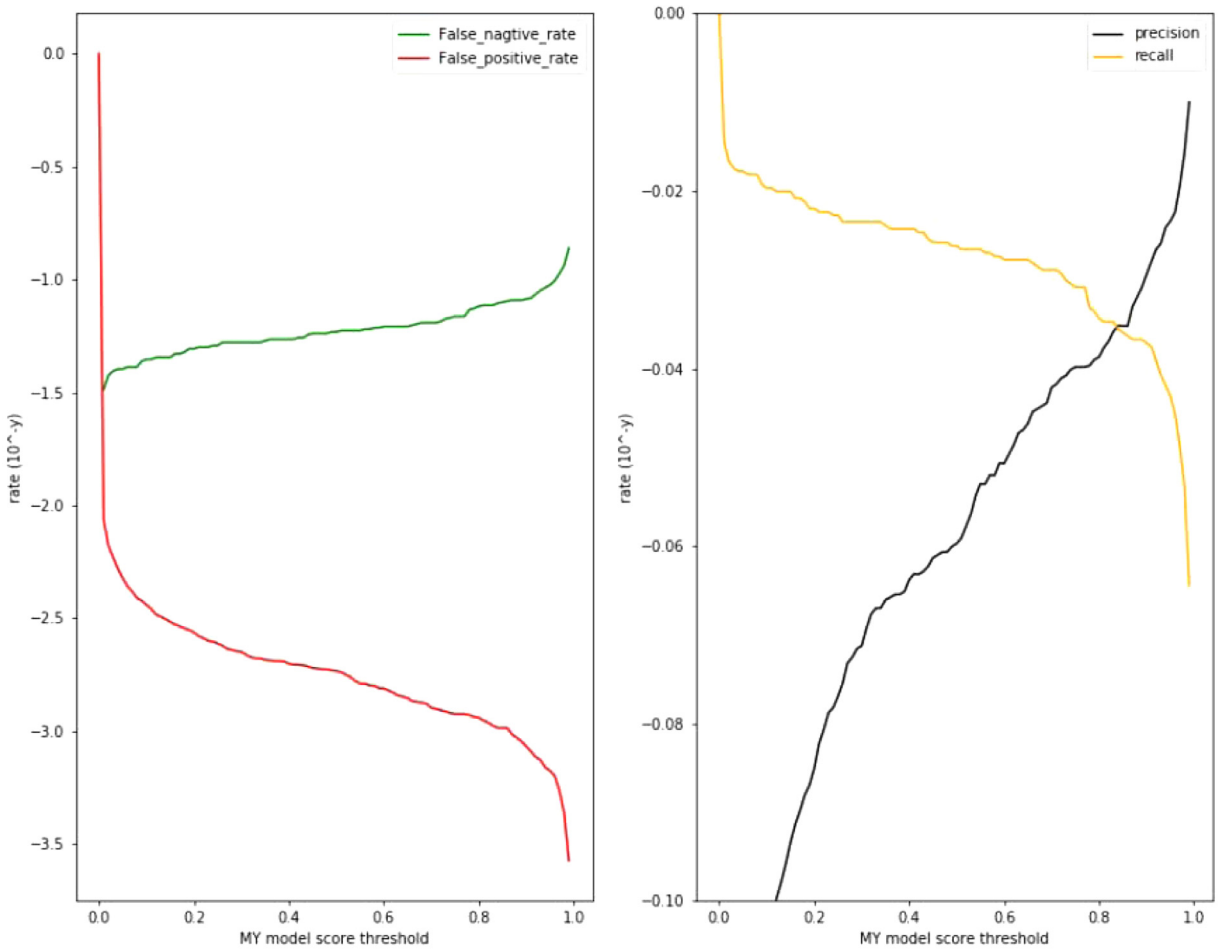

Fig. 8. The performance evaluation that falls under a 1: 2 over-sampled rate. (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Table 3
*F*1-score results.

C4.5	MLP	NB	SVM	GH-VFDT	CNN
0.74	0.75	0.69	0.79	0.86	0.92

We use an *F*1-score to compare state-of-the-art traditional machine learning models, as shown in Table 3, because these models combine precision and recall. The *F*1-score data comes from fifty years of pulsar candidate selection [11]. Thus, our model is better than any existing models.

We use the CNN because it can train the model faster than the DNN. The CNN is applied to images because adjacent pixels in an image with relevance. We achieve high precision but we do not know our data's internal relevance. We will study the internal relevance in future work.

6. Conclusion

In this paper, we presented a VGG-net-based learning algorithm for pulsar recognition. In the data pre-processing step, we used a random forest algorithm to remove outliers. We also used data normalization and standardization techniques to improve the accuracy and speed of the optimal solution via a stochastic gradient descent method. In the feature engineering step, we used an ANN to help us choose good features by performing a simple classification. We sorted these features according to their accuracy and then used these good single features to construct a set of combined features. The feature subset is selected by calculating the mutual information coefficient order. The validity of these features is verified by linear regression. Furthermore, we used the feature subset to construct high-order features with good results. While training via the High Time Resolution Universe dataset, we adopted over-sampling techniques to deal with imbalance issues of the

original data. We manually adjusted an L2-penalty multiplier and the drop-out rate based on existing operator experience. We normalized each layer and adopted early termination methods to avoid over-fitting, which made the model more generalizable. We also investigated the goodness of various extracted features, and well-trained the VGG-net learning model. We adopted ten-fold cross-validation to evaluate the performance. A variety of ROC curves were created for various oversampling ratios to find the best oversampling ratio for our model. Ultimately, we reported our best cross-validated results after we adjusted several different parameters in terms of recall, precision and ROC. Results have shown that deep learning-based recognition algorithms could achieve highly accurate precision and recall.

Acknowledgments

This simulation was performed on the China SKA Data Processor Prototype at Shanghai Astronomical Observatory with funding from the Ministry of Science and Technology of China (Grant nos. 2016YFE0100300 and SQ2018YFA040022) and the Chinese Academy of Sciences (CAS, grant no. 114231KYSB20170003). Tao An thanks to the youth innovation promotion association of the CAS.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.compeleceng.2018.10.016.

References

[1] Eatough RP, Molkenthin N, Kramer M, Noutsos A, Keith MJ, Stappers BW, Lyne AG. Selection of radio pulsar candidates using artificial neural networks. Mon Not R Astron Soc 2010;407(4):2443–50. https://doi.org/10.1111/j.1365-2966.2010.17082.x.
[2] Bates SD, Lorimer DR, Verbiest JP. The pulsar spectral index distribution. Mon Not R Astron Soc 2013;431(2):1352–8. https://doi.org/10.1093/mnras/stt257.
[3] Lee KJ, Stovall K, Jenet FA, Martinez J, Dartez LP, Mata A, Flanigan J. PEACE: pulsar evaluation algorithm for candidate extraction–a software package for post-analysis processing of pulsar survey candidates. Mon Not R Astron Soc 2013;433(1):688–94. https://doi.org/10.1093/mnras/stt758.
[4] Morello V, Barr ED, Bailes M, Flynn CM, Keane EF, van Straten W. SPINN: a straightforward machine learning solution to the pulsar candidate selection problem. Mon Not R Astron Soc 2014;443(2):1651–62. https://doi.org/10.1093/mnras/stu1188.
[5] Wagstaff KL, Tang B, Thompson DR, Khudikyan S, Wyngaard J, Deller AT, Wayth RB. A machine learning classifier for fast radio burst detection at the VLBA. Publ Astron Soc Pac 2016;128(966):084503. http://doi.org/10.1088/1538-3873/128/966/084503.
[6] Kotsiantis SB, Zaharakis I, Pintelas P. Supervised machine learning: a review of classification techniques. Emerg Artif Intell Appl Comput Eng 2007;160:3–24.
[7] Bengio Y. Learning deep architectures for AI. Foundations and Trends® in Machine Learning 2009;2(1):1–127. http://dx.doi.org/10.1561/2200000006.
[8] Hinton GE, Salakhutdinov RR. Reducing the dimensionality of data with neural networks. Science 2006;313(5786):504–7. doi:10.1126/science.1127647.
[9] Simonyan K, Zisserman A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.
[10] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Adv Neural Inf Process Syst 2012:1097–105. doi:10.1234/12345678.
[11] Lyon RJ, Stappers BW, Cooper S, Brooke JM, Knowles JD. Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. Mon Not R Astron Soc 2016;459(1):1104–23. https://doi.org/10.1093/mnras/stw656.

Haoyuan Zhang received the B.Sc. degree from Nanjing University of Posts and Telecommunications, Nanjing, China, in 2016, in electrical and computer engineering. He is currently a master student in University of Chinese Academy of Sciences. His research mainly focuses on pulsar astronomy and machine learning.

Zhen Zhao received the B.Sc. degree from Guilin University of Electronic Technology, Guilin, China, in 2013, and M.Sc. degree from University of Manitoba, Winnipeg, MB, Canada, in 2017, both in electrical engineering. He is currently an assistant engineer at Shanghai Astronomical Observatory. His research focuses on semi-supervised learning, big data analytics, and their applications to astronomical data.

Tao An is a research professor of Shanghai Astronomical Observatory, Chinese Academy of Sciences. Organization Committee member of Commission B4 Radio Astronomy, International Astronomical Union, Deputy Director of Youth Committee, Chinese Astronomical Union, Head of SKA group. Research interests include radio astronomy, astrophysics, and astronomical technique.

Baoqiang Lao is a software engineer of Shanghai Astronomical Observatory. His research focuses on wide-field imaging algorithms for radio interferometry and its acceleration algorithms.

Xiao Chen is an assistant engineer in Shanghai Astronomical Observatory, China. His research area mainly focuses on mitigation of Radio Frequency Interference and Very Long Baseline Interferometry technology.