

Radio frequency interference mitigation using pseudoinverse learning autoencoders

Hong-Feng Wang^{1,2,3,5}, Mao Yuan^{6,2}, Qian Yin¹, Ping Guo^{4*}, Wei-Wei Zhu^{2*}, Di Li^{2,6,8*} and Si-Bo Feng⁷

¹ Image Processing and Pattern Recognition Laboratory, School of Artificial Intelligence, Beijing Normal University, Beijing 100875, China

² CAS Key Laboratory of FAST, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China; zhuww@nao.cas.cn

³ School of Information Management, Dezhou University, Dezhou 253023, China

⁴ Image Processing and Pattern Recognition Laboratory, School of System Science, Beijing Normal University, Beijing 100875, China; pguo@ieee.org

⁵ Institute for Astronomical Science, Dezhou University, Dezhou 253023, China

⁶ University of Chinese Academy of Sciences, Beijing 100049, China; dili@nao.cas.cn

⁷ Hanvon Technology Co.,Ltd, Beijing 100193, China

⁸ NAOC-UKZN Computational Astrophysics Centre, University of KwaZulu-Natal, Durban 4000, South Africa

Received 2019 September 19; accepted 2020 March 10

Abstract Radio frequency interference (RFI) is an important challenge in radio astronomy. RFI comes from various sources and increasingly impacts astronomical observation as telescopes become more sensitive. In this study, we propose a fast and effective method for removing RFI in pulsar data. We use pseudo-inverse learning to train a single hidden layer auto-encoder (AE). We demonstrate that the AE can quickly learn the RFI signatures and then remove them from fast-sampled spectra, leaving real pulsar signals. This method has the advantage over traditional threshold-based filter method in that it does not completely remove contaminated channels, which could also contain useful astronomical information.

Key words: pulsars: general — methods: numerical — methods: data analysis

1 INTRODUCTION

The impact of radio frequency interference (RFI) in radio astronomy is becoming more significant as human activity flourishes and radio telescopes become more sensitive. As one of the most sensitive single-dish radio telescopes, the Five-hundred-meter Aperture Spherical radio Telescope (FAST) is particularly susceptible to RFI, which comes from many different sources, such as terrestrial signals, cellphone stations, airplanes, and radar (Fridman & Baan 2001). Man-made RFI usually exists in a narrow stationary frequency range or in the form of short impulsive signals. Sometimes, satellites will generate RFI that changes over time due to Doppler shifting. Impulsive time-domain RFI can be identified by taking running-statistics on the signal time series.

Many techniques have been proposed to excise RFI from astronomical data, such as reference anten-

nas for RFI signal subtraction (Barnbaum & Bradley 1998; Briggs et al. 2000; Finger et al. 2018), spatial filtering techniques (Leshem et al. 2000; Ellingson & Hampson 2002; Smolders & Hampson 2002; Boonstra et al. 2002; Kocz et al. 2010; Keane et al. 2018), threshold-based signal filter method (Baan et al. 2004; Offringa et al. 2010; Nita & Gary 2010; Peck & Fenech 2013), machine-learning or deep-learning methods (Burd et al. 2018; Czech et al. 2018; Yang et al. 2020). Eatough et al. (2009) introduced a zero-DM filter that utilized a signature of wide-band impulsive RFI and removed them effectively in pulsar data. Pen et al. (2009) proposed the singular value decomposition (SVD) RFI mitigation method. An RFI signal is usually stronger than an astronomical signal. As a result, the largest SVD eigen values and vectors often represent RFI and the persistent structure in the data, such as the bandpass. By setting these eigenvalues to zero when reconstructing the data, we can possibly remove strong RFIs

* Corresponding author

and bandpass. A recent review by An et al. (2017) introduced all kinds of RFI mitigation strategies.

The FAST (Nan et al. 2011) is currently being commissioned (Jiang et al. 2019) and the 19-beams receiver has been installed (Li et al. 2018). Searching for new pulsars is one of the main scientific objectives of FAST, at present, it has discovered dozens of new pulsars (Qian et al. 2019; Zhang et al. 2019).

The FAST pulsar data includes two significant components: (1) the bandpass of the data; and (2) RFI, both wide-band and narrow-band. These two components are always or frequently reoccurring in the data. Conversely, real pulsar signals are often weaker than RFI and uncommon in the data. RFI excision is very important for pulsar search. However, there are two problems in the traditional methods of RFI mitigation, one is the low efficiency and the other is incomplete pulsar signal. When using threshold-based method to mitigate RFI, the pulsar signal will be influenced. Therefore, we could use machine learning techniques to model the signature of bandpass and RFI from a large segment of data, and then apply these models to fit and reconstruct data. Because these models are constructed from common reoccurring signatures, they could potentially catch and clean common RFI. Following these ideas, we experimented to remove RFI from pulsar data using the unsupervised machine learning method (i.e. pseudo-inverse learning auto-encoder). Our method is in theory similar to the SVD method. We use auto-encoder to learn the primary components of data. Because the most significant components of the raw data are the spectral bandpass and RFI, the recomposed data mainly contains these signals. When we subtract the recomposed data from the raw input data, we leave what is uncommon, (i.e., astronomical signals).

Our proposed method has two advantages: (1) the influence of pulsar signal is reduced by adjusting the number of neurons in the hidden layer and regularization parameter; and (2) by training the auto-encoder with pseudo-inverse learning, the algorithm can run efficiently.

The rest of this paper is organized as follows. In Section 2, we introduce pseudo-inverse learning auto-encoder (PILAE), and we then use the method to remove RFI. In Section 3, the ultra-wide-band data of FAST is processed through PILAE, and the experiment results are analyzed. Our discussion and conclusions are described in Section 4.

2 METHOD

In this section, we introduce pseudo-inverse learning auto-encoder (PILAE). We then describe the procedure of mitigating RFI by using PILAE according to the characteristic of data.

2.1 Pseudo-inverse Learning Auto-encoder

Pseudo-inverse learning (PIL) was proposed by Guo & Lyu (2001) and Guo & Lyu (2004). Wang et al. (2017) used pseudo-inverse learning for training stacked auto-encoder to classify astronomical spectrum and recover defective spectra. The training speed of this method is fast because there is only feed-forward propagation in PILAE. Because of the heavy burden of FAST data storage, it is particularly important to propose rapid computational method to eliminate RFI. Therefore, we have made some improvements to the traditional PILAE method. It can rapidly remove RFI while retaining the celestial signal. The training set is $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N] \subseteq \mathbb{R}^{N \times n}$, the i th vector can be described as: $\mathbf{x}_i = [x_1, x_2, \dots, x_n]^T \subseteq \mathbb{R}^{n \times 1}$. $\mathbf{X} \subseteq \mathbb{R}^{N \times n}$ is a column vector. So, the steps of pseudo inverse learning to train the auto-encoder can be concluded as follows: Step 1: the number of hidden layer neurons setting. The rank of the input matrix is used to decide the number of hidden layer neurons. A singular value decomposition is applied to the input matrix:

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T, \quad (1)$$

where matrices $\mathbf{U} \subseteq \mathbb{R}^{N \times N}$ and $\mathbf{V}^T \subseteq \mathbb{R}^{r \times r}$ are orthogonal matrices, $\mathbf{\Sigma} \subseteq \mathbb{R}^{N \times r}$ is a diagonal matrix. The diagonal elements are the eigenvalues of the input matrix \mathbf{X} . The number of non-zero elements is the rank r of the $\mathbf{\Sigma}$ matrix:

$$r = \text{rank}(\mathbf{X}) = \text{the number of nonzero}(\mathbf{\Sigma}). \quad (2)$$

The number of hidden layer neurons p is set to be less than the dimension n of the input matrix for learning the features of the training data. It also can avoid identity mapping and complex calculation of high dimensions. If the number of hidden layer neurons is too small, then this will lead to large reconstruction errors and the original data feature missing. The parameter p is related to the rank of the input matrix r and the dimension of the input matrix n . At the same time, we should be aware of the feature learning of the original data and the model reconstruction error. So, the value of p is set to be between the number of samples and the rank:

$$p = r + \alpha(n - r), \alpha \in (0, 1], \quad (3)$$

where the parameter α is an empirical parameter to set the number of hidden neurons. If the auto-encoder input matrix is full rank, then the rank r of the input matrix is equal to the dimension n of the input data. We then utilize the dimension reduction to set the value of the number of hidden layer neurons p (β is empirical parameter) to

$$p = \beta r, \beta \in (0, 1]. \quad (4)$$

When removing RFI, we can set the number of hidden layer elements based on experimental experience, which can improve the efficiency of the algorithm without calculating the rank of the input matrix.

Step 2: encoder weight initialization (\mathbf{W}_e). Utilizing random value (zero median and unit variance) to initialize the encoder weight. Therefore, the input matrix can map n dimension data to p dimension through encoder weight, and the output of the hidden layer is

$$\mathbf{H} = \mathbf{X}\mathbf{W}_e. \quad (5)$$

Step 3: decoder weight calculation (\mathbf{W}_d). The loss function of the auto-encoder is

$$\mathbf{E} = \frac{1}{2} \|\mathbf{H}\mathbf{W}_d - \mathbf{X}\|^2, \quad (6)$$

where symbol \mathbf{W}_d is the decoder weight of the auto-encoder. To avoid overfitting, L_2 norm is added to the loss function, which thus can be written as

$$\mathbf{E} = \frac{1}{2} \|\mathbf{H}\mathbf{W}_d - \mathbf{X}\|^2 + \frac{\lambda}{2} \|\mathbf{W}_d\|^2. \quad (7)$$

λ is the regularization parameter, which can reduce the influence on the celestial signal when removing the RFI. Taking the derivative of Equation (7) and the equation can be expressed as

$$(\mathbf{H}\mathbf{W}_d - \mathbf{X})\mathbf{H}^T + \lambda\mathbf{W}_d = 0. \quad (8)$$

Therefore, the decoder weight can be expressed as

$$\mathbf{W}_d = (\mathbf{H}^T\mathbf{H} + \lambda\mathbf{I})^{-1}\mathbf{H}^T\mathbf{X}. \quad (9)$$

Step 4: the raw data reconstruction. After this calculation, we can obtain the reconstruction data. \mathbf{H} is the output of the hidden layer, and \mathbf{W}_d is the decoder weight. \mathbf{O} is the recomposed data, which mainly contain the spectral bandpass and radio interference signals as

$$\mathbf{O} = \mathbf{H}\mathbf{W}_d. \quad (10)$$

2.2 PILAE-based RFI Mitigation

We utilize one basic auto-encoder (AE) to mitigate the RFI. The network contains an input layer, a hidden layer, and an output layer. Although the input data and output data tend to be consistent in AE, there are still reconstruction errors because of data dimensionality reduction through the hidden layer. In general, the radio interference signal is much stronger than the pulsar signal in the data. Therefore, the output layer mainly outputs RFI and bandpass signals. We trained the model by pseudo-inverse learning. The model can run efficiently because it does not need back propagation. Our algorithm contains the following four steps (the algorithm flow chart is shown in Fig. 1):

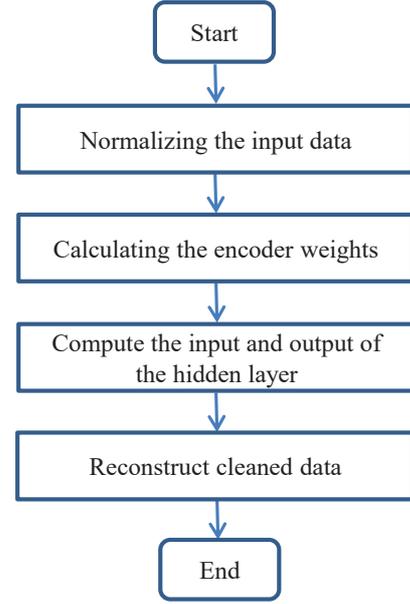


Fig. 1 The algorithm flow chart of PILAE RFI removal.

Step 1: normalizing the input data. The input data is a two-dimensional matrix (4096×4096), where each row represents different time data of the same frequency band, and each column represents different frequency data of the same time band. We normalize the input data value to between 0 and 1.

Step 2: calculating the encoder weights. We randomly initialize the encoder weight, and the weight value satisfies the Gaussian distribution with a mean value 0 and variance 1. In the traditional PIL algorithm, the pseudo-inverse of the input matrix is used as the weight of the encoder. We randomly initialize the weight of encoder to extract random features from input data, which can also be regarded as adding noise to data and improve the efficiency of RFI mitigation.

Step 3: compute the input and output of the hidden layer. The input of the hidden layer is calculated through the input matrix and the weight of the encoder (Eq. (5)), and computed the output of the hidden layer using activation function (sigmoid function).

Step 4: reconstruct cleaned data. At first, we calculate the weight of the decoder according to Equation (9). Then, according to Equation (10), we can obtain the output matrix (\mathbf{O}), which is the reconstruction of the input data. The recomposed data mainly contains RFI data. Thus, we use Equation (11) to remove RFI from raw data as

$$\mathbf{R} = \mathbf{X} - \mathbf{O}, \quad (11)$$

where \mathbf{R} is the cleaned data after bandpass and RFI has been removed, \mathbf{X} is the original data.

3 EXPERIMENTS AND RESULTS

In this section, we present how the PILAE model is trained, and we experiment with the model’s performance and robustness.

3.1 Experiment Data

We use FAST ultra-wide-band (UWB) receiver data to test the validity of the model. The FAST ultra-wide-band receiver was installed on FAST from 2016 to 2018 for testing purposes. During that time, it was used for pulsar searches in drifting and tracking mode and discovered dozens of new pulsars. The UWB receiver operates in 270–1800 MHz frequency range with an effective system temperature in the range of 60–120 K. We use it to sample spectrum in 200 μ s intervals. In each interval, we collect two spectra, one with 4096 channels in 0–1 GHz frequency range, one with 4096 channels in 1–2 GHz frequency range. Not all frequency channels in the data contain useful astronomical data, but some of them are occasionally contaminated by broad-band time-domain RFI and narrow-band frequency-domain RFI, which reduced their effectiveness in detecting weak pulsars. In this experiment, we use four 0–1 GHz FAST UWB pulsar data for testing. The four pulsar data contain real pulsars, which we use to demonstrate how the PILAE method removes bandpass and narrow-band RFI without subtracting the pulsar signal. The pulsars are J2112+4059, J2113+4644, J0659+1414 and J2006+4101. Pulsar J2112+4059, J0659+1414 and J2006+4101 have strong interference signals, while pulsar J2113+4644 has strong pulsar signals and weak interference signals.

3.2 PILAE Hyper Parameter Tuning

We select the data of pulsar J2112+4059 to optimize the hyper parameters of our model, and use the S/N of the single pulse to evaluate the performance. The J2112+4059 original data are shown in Figure 2. The horizontal axis is time and the vertical axis is frequency. The pulsar signal is hard to visualize in Figure 2, because of the existence of strong RFI and varying signal baseline (bandpass). Especially in the low-frequency part, pulsar signal due to the raising baseline. To show the pulsar signal after RFI removal, we apply PILAE to the data in Figure 2, we use 400 neurons in the hidden layer, and $\lambda = 0.1$ regularization. After RFI mitigation, the pulsar signal is clearly visible in Figure 3. The pulsar signal in Figure 3 is dispersed by interstellar medium, causing low-frequency signals to arrive later than higher-frequency signals. We can remove this dispersion effect through an operation called de-dispersion. We can fold the de-dispersed data by using

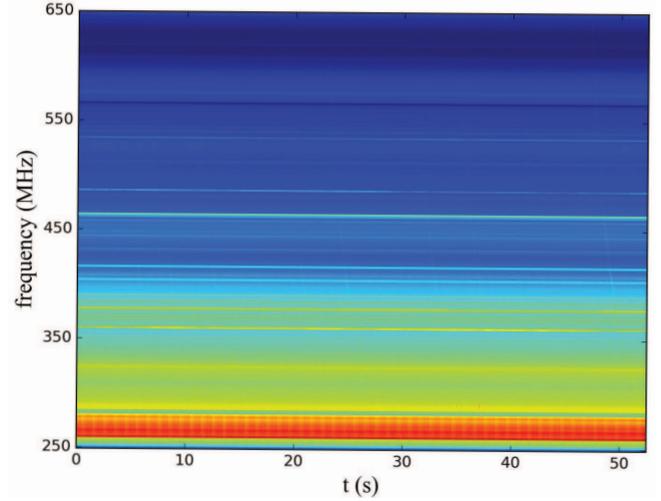


Fig. 2 The original frequency versus time pulsar data, down-sampled to 4096×4096 . There are strong narrow-band RFI and a pulsar’s signal in the data.

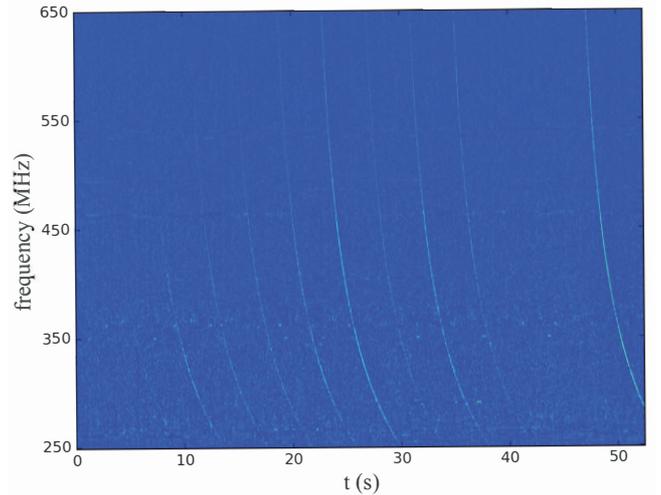


Fig. 3 The data in Fig. 2 cleaned with PILAE, showing pulsar signal without strong RFI.

the period of the pulsar and summed all the frequencies into a folded pulse profile, and we then calculate the S/N of the profile¹. This S/N is an good indicator of the signal strength and we used it to evaluate our model in subsequent analysis.

To optimize the hyper parameters of our model, we perform two experiments to 1. we experiment different levels of regularization with a fixed-number of hidden layer neurons; and 2. we changed the number of neurons in the hidden layer and find the setting that maximize the resulting pulsar S/N.

In the first experiment, we use J2112+4059 pulsar data to test and optimize for the regularization parameters λ .

¹ Refer to formula 7.1 of handbook of pulsar astronomy, the authors are D. R. Lorimer & M. Kramer.

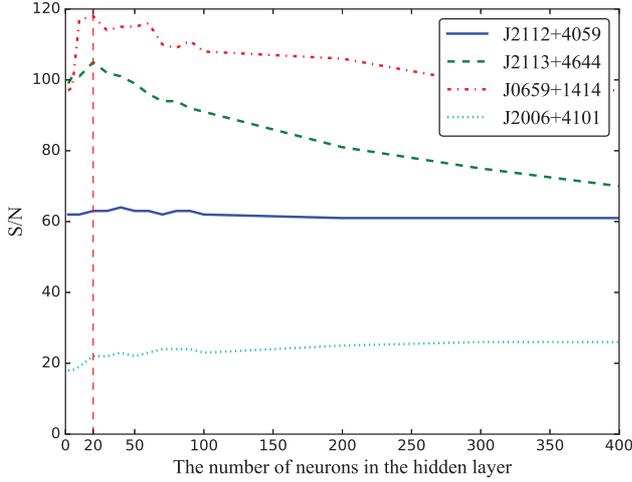


Fig. 4 The number of neurons in the hidden layer and the S/N of four pulsars relation curve.

We set the number of hidden layer neurons to 20, and grid searched λ between 0 and 1. The resulting S/N of the pulsar is listed in Table 1. The experimental results show that the different values of λ have very limited impact to the S/N. Therefore, we choose a medium value of 0.5 for λ in our future experiments.

In the second experiment, we use pulsar J2112+4059, J2113+4644, J0659+1414, and J2006+4101 to fine tune the number of neurons in the hidden layer. We set $\lambda = 0.5$, and experimented with a range of numbers between 2 and 400 for the hidden-layer neurons, and calculate the resulting S/Ns after RFI mitigation. Figure 4 shows that for strong pulsars like J2112+4059, J2113+4644, J0659+1414, the optimal number of neurons is possibly around 20. The S/N of relatively weak pulsars like J2006+4101 seems to increase slowly with number of neurons. According to this experiment, we choose to use 20 neurons in the hidden layer for later experiments.

3.3 PILAE Model Performance and Robustness

In this section, we use UWB FAST data to test and compare the performance of PILAE and SVD models. We determine the PILAE hyper parameters through the aforementioned experiments and set the number of the hidden layer neurons to 20, and λ to 0.5, and use this model to remove RFI from pulsars J2112+4059, J2113+4644, J0659+1414, and J2006+4101. For the SVD method, we optimize for the best rank of eigenvectors to remove as RFI based on the resulting S/N, and choose 2 of the largest SVD eigenvectors and flagged them as RFI. We present the S/Ns of the pulsar signals before and after applying PILAE and SVD to the data of pulsars J2112+4059, J2113+4644, J0659+1414 and J2006+4101 (Table 2). For PSR J2113+4644, J0659+1414, both models end up im-

prove the S/N of the pulsars. But for PSR J2112+4059, because of the presence of a strong pulsar signal, both PILAE and SVD models seem to have absorbed a small portion of the pulsar signals into its reconstructed RFI and produce reduced S/N. For all four pulsars, the RFI mitigation results of the PILAE method are better than or equal to those of the SVD method. For PSR J2006+4101, we can increase the number of hidden layer neurons in the PILAE method. When the number of hidden layer neurons is set to 200, the S/N of PSR J2006+4101 can reach 25 after RFI removal. We show the pulse profiles and the frequency-phase plot of the four pulsars before and after RFI removal in Figures 5, 6, 7 and 8. The pulse profiles of the strong pulsars J2112+4059 and J2113+4644 show some degree of baseline distortion after RFI treatments, which suggests that PILAE and SVD methods may affect the off pulse profile baseline. The frequency-phase plots of these pulsars show that most of the bright narrow-band RFI features in the original data are no longer visible after the RFI mitigation. However, some weaker narrow-band RFI remains. These remaining RFIs seem to be varying in a short time scale. This suggests that PILAE and SVD-based RFI mitigation methods are best for removing persistent RFIs; while the fast-varying RFIs need to be treated with other techniques.

We compared the results of RFI cleaning from the two models and also evaluated the running time of the two methods. FAST pulsar data is a two-dimensional matrix 4096×262144 . Our method takes a short computing time, it takes ~ 0.1 s to remove the RFI from a single set of pulsar data (4096×4096 2D data) using a 24-core computer. A fits data process cost about ~ 6 s. SVD method² will take about 16 s to finish processing 4096×4096 2D data, it takes ~ 17 min to process one fits data. The PILAE method exhibits a faster performance than that exhibited by the SVD method.

Finally, we test whether the randomly initialized PILAE model converges and is robust. The weight of the encoder is randomly initialized to satisfy the Gaussian distribution with mean value 0 and variance 1, the number of hidden layer neurons is set to 20, and the regularization λ is set to 0.5. After removing RFI from J2112+4059 pulsar data, the S/N can be calculated. We perform 10 independent experiments and obtain the mean and standard deviation of the S/N is $62.7(\pm 0.7)$. In comparison, we use the bootstrap method to estimate the nature random fluctuation in S/N due to sampling. We select one dataset and randomly selected half of its channels to calculate the S/N. The resulting S/N is $58.7(\pm 1.1)$ in 10 independent experiments. The experimental results show that the error caused by random initialization of the encoder weight is compa-

² The method used `numpy.linalg.svd` function.

Table 1 The Relation between Regularization Parameters and S/N

λ	0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
S/N	63	64	63	63	63	63	63	62	62	63

Table 2 PILAE and SVD RFI Mitigation Results

Pulsar Name	Original S/N	PILAE S/N	SVD S/N
J2112+4059	89	63	62
J2113+4644	98	105	98
J0659+1414	82	114	103
J2006+4101	23	22	22

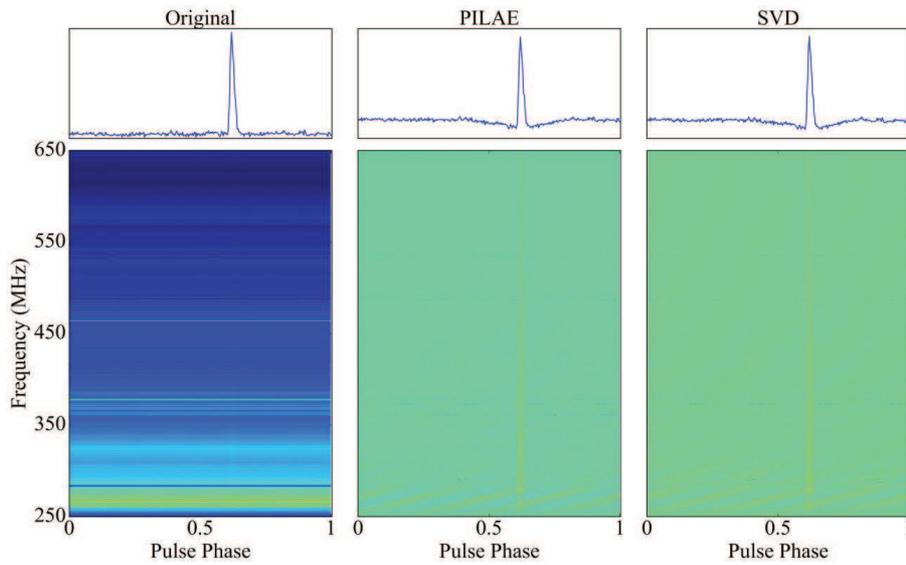


Fig. 5 The pulse profile and the folded frequency-phase plot of PSR J2112+4059 before and after RFI mitigation processes. The left panels show the original data. The middle panels show the result of the PILAE method. The right panels show the result of the SVD method.

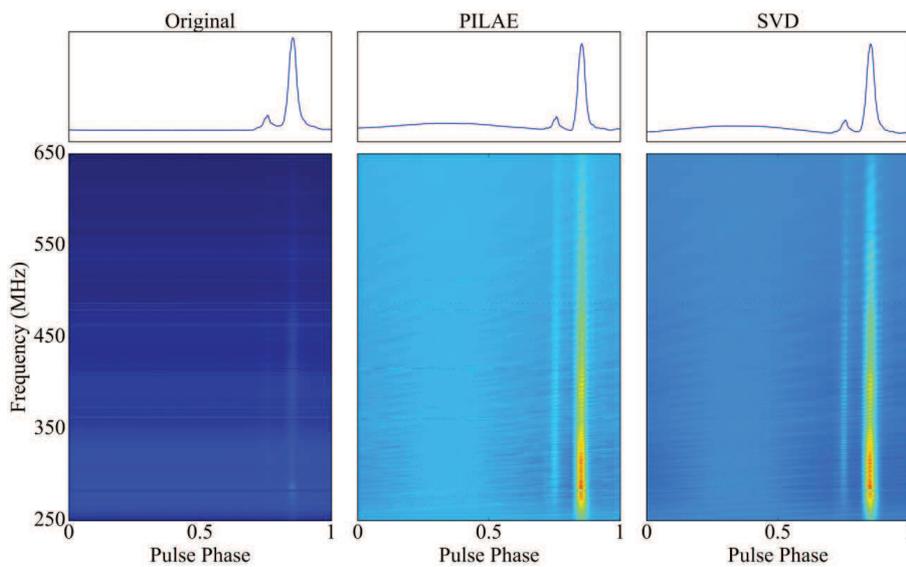


Fig. 6 The pulse profile and the folded frequency-phase plot of J2113+4644 presented in the same order as in Fig. 5.

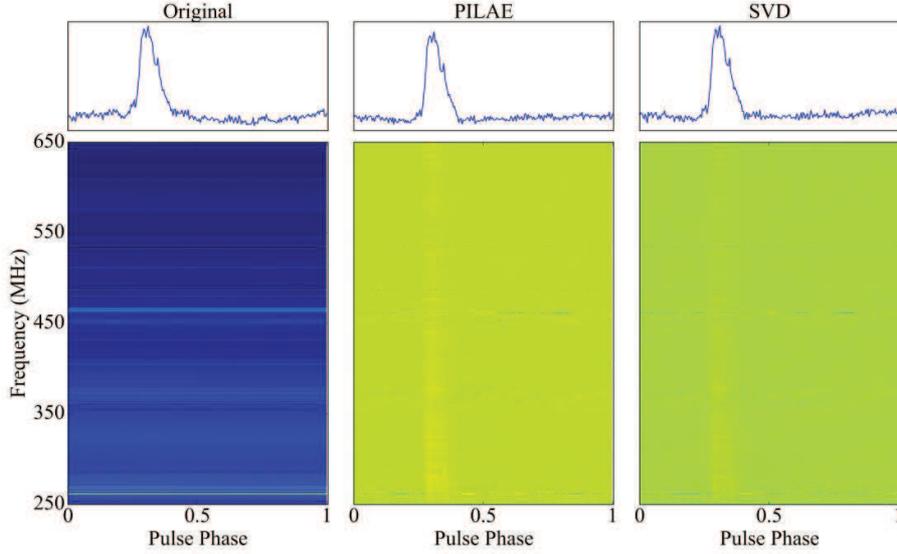


Fig. 7 The pulse profile and the folded frequency-phase plot of J0659+1414 presented in the same order as in Fig. 5.

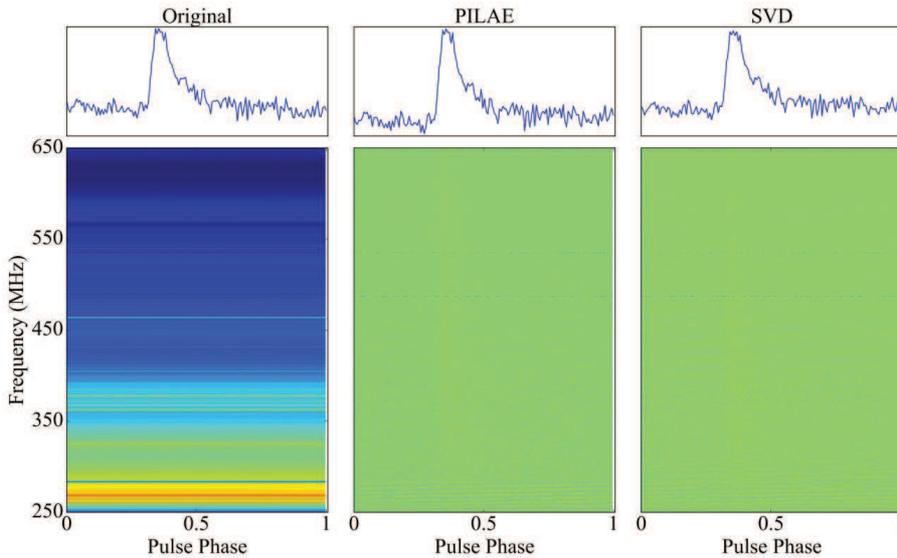


Fig. 8 The pulse profile and the folded frequency-phase plot of J2006+4101 presented in the same order as in Fig. 5.

rable to the nature fluctuation of S/N itself and does not cause substantial extra divergence, indicating that the performance of the PILAE model is robust against random initialization.

4 DISCUSSION AND CONCLUSIONS

In this paper, we introduce a new RFI mitigation algorithm – PILAE. This algorithm can remove both persistent narrow-band RFI and the bandpass from pulsar data. Traditional threshold-based filter method often completely removes all data in bad channels or contaminated spectral samples, causes the pulsar signals in those channels and samples to be lost. We demonstrate in Section 3

Table 2 that PILAE could cleanly remove both types of RFI yet retain majority of the pulsar signals. This is a highly desirable feature for pulsar searching as well as for a slew of other exploratory scientific objectives (Li et al. 2019) including radio exoplanets, gravitational wave, and so on. We also demonstrate that our method is slightly better than the state-of-the-art SVD-based RFI removal technique in recovering pulsar signals (Table 2), as well as taking less computing cycles to complete (Sect. 3.3). Although better than threshold-based method in principle, the PILAE method and the SVD method share the same caveat that they sometimes still remove a small portion of the pulsar signal, especially when the pulsar is substan-

tially stronger than the RFIs. In these cases (for example for PSR J2122+4059), the PILAE and SVD methods still could clearly enhance the visibility of the signal to eyes, and they are unlikely to hinder the detection of the pulsar signal because they are already very strong. However, it is not advised to use these methods to precisely measure the pulse profiles and polarizations because they could introduce small distortion to the baseline.

At present, FAST produces over two hundred terabytes of data per day. During one night of drift-scan, the telescope produces around 10 000 fits files in pulsar search project. Therefore, we need an efficient way to process data in order to keep up with the data stream. The training of PIL is easier than other deep-learning methods because the training process does not require back-propagation. It takes ~ 6 s to process one fits data using a 24-core computer without considering I/O time. We can process ~ 14 000 fits data per day using this method. Therefore, this method can be implemented efficiently and it can meet the requirements of real-time processing of FAST data.

We could use this method to quickly clean the data and then process the results by eye or other searching techniques to determine whether or not they contain pulsar signals. In the future, machine learning methods can be used to identify these images directly, which will greatly improve the efficiency of the pulsar search.

In this work, we find the optimal hyper parameters through grid search. In future work, the hyper parameters could be determined automatically through the method of auto machine leaning (Yao et al. 2018).

Acknowledgements The research work is supported by the National Natural Science Foundation of China (NSFC, Grant Nos. 11988101, 61472043, 11743002, 11873067, 11690024, 11673005 and 11725313) and the Outstanding Youth Fund Project of Natural Science Fund of Shandong Province (Grant No. ZR2019YQ03). This project is also supported by the Joint Research Fund in Astronomy (U1531242) under cooperative agreement between the NSFC and the Chinese Academy of Sciences (CAS). WWZ is supported by the Chinese Academy of Science Pioneer Hundred Talents Program, the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB23000000).

References

- An, T., Chen, X., Mohan, P., & Lao, B. Q. 2017, *Acta Astronomica Sinica*, 58, 43
- Baan, W. A., Fridman, P. A., & Millenaar, R. P. 2004, *AJ*, 128, 933
- Barnbaum, C., & Bradley, R. F. 1998, *AJ*, 116, 2598
- Boonstra, A., van der Veen, A., & Raza, J. 2002, in 2002 IEEE International Conference on Acoustics, Speech, and Signal Processing, 3, III
- Briggs, F. H., Bell, J. F., & Kesteven, M. J. 2000, *AJ*, 120, 3351
- Burd, P. R., Mannheim, K., März, T., et al. 2018, *Astronomische Nachrichten*, 339, 358
- Czech, D., Mishra, A., & Inggs, M. 2018, *Astronomy and Computing*, 25, 52
- Eatough, R. P., Keane, E. F., & Lyne, A. G. 2009, *MNRAS*, 395, 410
- Ellingson, S. W., & Hampson, G. A. 2002, *IEEE Transactions on Antennas and Propagation*, 50, 25
- Finger, R., Curotto, F., Fuentes, R., et al. 2018, *PASP*, 130, 025002
- Fridman, P. A., & Baan, W. A. 2001, *A&A*, 378, 327
- Guo, P., & Lyu, M. 2001, *Advances in Neural Networks and Applications*, 321, <https://www.researchgate.net/publication/293477570>
- Guo, P., & Lyu, M. R. 2004, *Neurocomputing*, 56, 101
- Jiang, P., Yue, Y., Gan, H., et al. 2019, *Science China Physics, Mechanics, and Astronomy*, 62, 959502
- Keane, E. F., Barr, E. D., Jameson, A., et al. 2018, *MNRAS*, 473, 116
- Kocz, J., Briggs, F. H., & Reynolds, J. 2010, *AJ*, 140, 2086
- Leshem, A., van der Veen, A.-J., & Boonstra, A.-J. 2000, *ApJS*, 131, 355
- Li, D., Dickey, J. M., & Liu, S. 2019, *RAA (Research in Astronomy and Astrophysics)*, 19, 016
- Li, D., Wang, P., Qian, L., et al. 2018, *IEEE Microwave Magazine*, 19, 112
- Nan, R., Li, D., Jin, C., et al. 2011, *International Journal of Modern Physics D*, 20, 989
- Nita, G. M., & Gary, D. E. 2010, *MNRAS*, 406, L60
- Offringa, A. R., de Bruyn, A. G., Biehl, M., et al. 2010, *MNRAS*, 405, 155
- Peck, L. W., & Fenech, D. M. 2013, *Astronomy and Computing*, 2, 54
- Pen, U.-L., Chang, T.-C., Hirata, C. M., et al. 2009, *MNRAS*, 399, 181
- Qian, L., Pan, Z., Li, D., et al. 2019, *Science China Physics, Mechanics, and Astronomy*, 62, 959508
- Smolders, B., & Hampson, G. 2002, *IEEE Antennas and Propagation Magazine*, 44, 13
- Wang, K., Guo, P., & Luo, A. L. 2017, *MNRAS*, 465, 4311
- Yang, Z., Yu, C., Xiao, J., & Zhang, B. 2020, *MNRAS*, 492, 1421
- Yao, Q., Wang, M., Chen, Y., et al. 2018, *arXiv e-prints*, arXiv:1810.13306
- Zhang, L., Li, D., Hobbs, G., et al. 2019, *ApJ*, 877, 55