Research in Astronomy and Astrophysics, 24:035022 (14pp), 2024 March https://doi.org/10.1088/1674-4527/ad0c28

Applying Hybrid Clustering in Pulsar Candidate Sifting with Multi-modality for FAST Survey

Zi-Yi You[1,2], Yun-Rong Pan[1], Zhi Ma[1], Li Zhang[3], Shuo Xiao[1,2], Dan-Dan Zhang[1], Shi-Jun Dang[1,2], Ru-Shuang Zhao[1,4], Pei Wang[4], Ai-Jun Dong[1,2], Jia-Tao Jiang[5], Ji-Bing Leng[6], Wei-An Li[6], and Si-Yao Li[7]
[1] School of Physics and Electronic Science, Guizhou Normal University, Guiyang 550025, China
[2] Guizhou Provincial Key Laboratory of Radio Astronomy and Data Processing, Guizhou Normal University, Guiyang 550025, China
[3] College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China; lizhang.science@gmail.com
[4] CAS Key Laboratory of FAST, National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101, China
[5] Key Laboratory of Information and Computing Guizhou Province, Guizhou Normal University, Guiyang 550001, China
[6] School of Big data and Computer Science, Guizhou Normal University, Guiyang 550025, China
[7] Guizhou Software Engineering Research Center, Guiyang 550000, China
Received 2023 September 8; revised 2023 October 27; accepted 2023 November 10; published 2024 March 6

Abstract

Pulsar search is always the basis of pulsar navigation, gravitational wave detection and other research topics. Currently, the volume of pulsar candidates collected by the Five-hundred-meter Aperture Spherical radio Telescope (FAST) shows an explosive growth rate that has brought challenges for its pulsar candidate filtering system. Particularly, the multi-view heterogeneous data and class imbalance between true pulsars and non-pulsar candidates have negative effects on traditional single-modal supervised classification methods. In this study, a multi-modal and semi-supervised learning based on a pulsar candidate sifting algorithm is presented, which adopts a hybrid ensemble clustering scheme of density-based and partition-based methods combined with a feature-level fusion strategy for input data and a data partition strategy for parallelization. Experiments on both High Time Resolution Universe Survey II (HTRU2) and actual FAST observation data demonstrate that the proposed algorithm could excellently identify pulsars: On HTRU2, the precision and recall rates of its parallel mode reach 0.981 and 0.988 respectively. On FAST data, those of its parallel mode reach 0.891 and 0.961, meanwhile, the running time also significantly decreases with the increment of parallel nodes within limits. Thus, we can conclude that our algorithm could be a feasible idea for large scale pulsar candidate sifting for FAST drift scan observation.

Key words: methods: data analysis – surveys – methods: numerical

1. Introduction

So far, a lot of radio pulsars have been discovered by modern pulsar surveys, including the High Time Resolution Universe (HTRU, Burke-Spolaor et al. 2011) Parkes survey, Low-Frequency Array (LOFAR) Tied-Array All-Sky Survey (LOTASS, Coenen et al. 2014), Commensal Radio Astronomy FasT Survey (CRAFTS, Jiang et al. 2019; Wang et al. 2021), Galactic Plane Pulsar Snapshot (GPPS, Han et al. 2021), etc. Compared to previous ones, modern surveys tend to use more sensitive detection techniques, greater survey areas, improved data analysis techniques and collaborative efforts, such as CRAFTS. They often produce a large number of potential pulsar candidates, but only a very small proportion of these candidates (nearly one in ten thousand) are identified as real pulsars for the reason of large amounts of interference signals. Thus, it is critical to reduce the retention of a large number of non-pulsar signals without loss of pulsar-like samples. At present, this issue can be alleviated on two research points of pulsar search pipelines. (i) Signal processing: remove Radio Frequency Interference (RFI) signals from a large amount of observational data as much as possible (Morello et al. 2014; Yang et al. 2020) and optimize some important parameters such as signal-to-noise ratio (SNR) detections and so on. (ii) Candidate selection: minimize the labor of further observations, which focuses on filtering the pulsar-like samples among large numbers of candidates automatically and accurately by advanced artificial intelligence techniques. This paper involves the latter.

Generally, existing pulsar candidate selection methods based on artificial intelligence could be classified into three categories, according to the principles of these methods. The traditional scoring methods are regarded as the first category, e.g., Pulsar Evaluation Algorithm for Candidate Extraction (PEACE, Lee et al. 2013). The second category refers to Machine Learning (ML) based classifiers that usually perform better than the first category, e.g., Morello et al. (2014), Lyon et al. (2016), Tan et al. (2018), etc. More recently, Burdwan (2019) applied the eight features designed by Lyon et al. (2016) to test the performance of the Random Forest (RF) algorithm, K-Nearest Neighbors (KNN) algorithm and Logistic Regression (LR)

algorithm. Xiao et al. (2020) designed a reliable KNN based model named Pseudo-Nearest Centroid Neighbor (PNCN) classifier for pulsar survey data streams, which can effectively deal with the class imbalance problem. In these methods, the features used often depend heavily on human experience, whose classification performance may be adversely affected. For example, some classifiers only extract features from the Dispersion Measure (DM) and pulse profile curve, which lead to some RFIs being identified as pulsars incorrectly. As the radio environment is becoming more complex, it is more difficult to effectively distinguish pulsar candidates and non-pulsar candidates only by statistical features. In practice, the pulsars can be successfully identified just by human experts observing the corresponding diagnostic plots. Based on this inspiration, the third category utilizes diagnostic plots as the inputs of image recognition models or multi-method ensemble models including image recognition, so that the "pulsar-like" mode can be learned from the diagnostic sub-graph automatically by training Deep Learning (DL) based models, e.g., Wang et al. (2019), Guo et al. (2019), Zeng et al. (2020), Zhang et al. (2021), etc. Compared with the ML based models in the second category, these methods have better generalization ability. Among them, the methods described in Wang et al. (2019) and Zeng et al. (2020) are mainly used in the pulsar search pipeline of the Five-hundred-meter Aperture Spherical radio Telescope (FAST) survey. Wang et al. (2019) presented a new ensemble classification system on FAST for pulsar candidate selection that is composed of five classifiers. Furthermore, it was incorperated in the development of the Pulsar Image-based Classification System (PICS) (Zhu et al. 2014). Zeng et al. (2020) designed an end-to-end online learning model, namely Concat Convolutional Neural Network (CCNN), to identify the candidates without any intermediate labels which are processed from FAST data. The aforementioned methods are mostly based on a single mode. At present, there is little literature about multi-modal methods applied to astronomical data mining, especially pulsar identification. Zhang et al. (2021) proposed an early fusion based pulsar image identification framework with smart under-sampling, which was evaluated on the HTRU Medlat data set. In this work, a semi-supervised learning and Feature-level Multi-modal Fusion based Hybrid Clustering (FMFHC) scheme is designed for large scale candidate sifting through FAST pulsar search pipelines, in terms of Wang et al. (2019) and Ma et al. (2022).

A typical pulsar search pipeline for FAST drift scan observed data roughly includes the following several steps: (i) eliminating the obvious interference signals from the original data; (ii) dedispersing the data into time series with distinct DM values; (iii) performing a fast Fourier transform on each time series so as to further search for periodic signals; (iv) sifting these periodic signals and outputting candidates obtained to files (e.g., suffix pfd); (v) folding the data in a periodic manner and then outputting candidate images. In practice, there are still a lot of non-pulsar candidates in step (iv), including RFI. Our algorithm is introduced into the sifting stage of step (iv), aiming to further address the following issues in the pulsar candidate selection area:

(i) For some aforementioned methods (e.g., the supervised ML based candidate signal classifiers and DL based diagnostic subplots recognition models), the cost of obtaining a large amount of labeled data (in which the proportion between the real pulsar and the non-pulsar sample is extremely imbalanced) and periodic training (to avoid overfitting and underfitting) is too high, and these methods are all based on binary classification.

(ii) In the process of pulsar search, there are usually multi-view heterogeneous candidate data, which contain various types and attributes. In practical applications, it could be difficult to further mine the deep features hidden in these data through a single modal candidate selection algorithm.

The rest of this paper is organized as follows: Section 2 describes the pulsar candidate features and similarity measure involved. Section 3 presents the components of the overall algorithm in detail. In Section 4, the experimental data sets, data pre-processing methods and results are illustrated. The discussion is in Section 5. Finally, in Section 6, we present the conclusion and future work.

2. Pulsar Candidate Features and Similarity Measure

2.1. Pulsar Candidate Features

The extraction of candidate features is very important to maximize the separation between non-pulsar and pulsar candidates. We assume the pulsar candidates were processed by the software pipelines based on PulsaR Exploration and Search TOolkit (PRESTO, Ransom 2011; Yue et al. 2013), which implemented similar search steps for advanced telescope systems such as FAST.

In terms of statistical features, there are eight new features extracted from the pfd files by the feature extraction program (Lyon et al. 2016), including the mean value, excess kurtosis, standard deviation and skewness of the pulse profile, and the mean value, excess kurtosis, standard deviation and skewness of the DM-SNR curve. Note that the first four statistics correspond to the integrated pulse profile, and the remaining four correspond to the DM-SNR curve. These features were chosen to maximize the separation of various candidate classes when used together with an ML classifier. Furthermore, the High Time Resolution Universe II (HTRU2) data set was used during the work Lyon et al. (2016). In terms of diagnostic plots, most features of a candidate signal can be visualized through different diagnostic subplots, including a folded profile plot, sub-integrations plot, sub-bands plot, DM-SNR curve, etc. The ensemble model based on PICS (Wang et al. 2019) can also extract four main feature plots of

a candidate from the pfd files, which are one-dimensional (1D) data array summed profile and DM curve, two-dimensional (2D) data array time versus phase (TVP) and frequency versus phase (FVP). Note that the size of 1D feature plots is 64×1, while the size of 2D feature plots is 64×64. The experiments implemented on the FAST pulsar survey data (Wang et al. 2019) demonstrate that PICS-ResNet can achieve a higher recall rate of 98% than PICS (which is 95%).

2.2. Similarity Measure

A candidate believed to be a real pulsar must have very similar statistical features (e.g., standard deviation and skewness of pulse profile) and diagnostic subplots (e.g., 2D FVP) with some other known pulsars. That is the reason we plan to design a multi-modal clustering algorithm for large amounts of pulsar candidate data. Figure 1 displays an example of the features and plots. It can be seen that known pulsars J0358 +5413 and J1915+1606 are very similar in the integrated pulse profile and FVP diagram. Similarly, interference signal I and interference signal II are also very similar in the integrated pulse profile and FVP diagram. The detailed information on these plots is described in Table 1. The feature similarity between candidates will be further validated in the experimental results of Section 4.2, as shown in Figure 8.

3. The Method

3.1. Feature Fusion Strategy

Through fusing different features of multiple modalities extracted from a single candidate, feature fusion methods can further refine the features with higher discrimination. Among them, Discriminant Correlation Analysis (DCA) is a linear method which maximizes the pair-wise correlation between two feature sets and possesses very low computational complexity at the same time.

Depending on the DCA algorithm, our objective is preprocessing candidate data extracted from pfd files by fusing features from different modalities of the same object before feeding them to a hybrid clustering scheme as input data. According to Section 2.1, these modalities include a 1D data array (statistical feature format on HTRU2) and 2D arrays (FVP and TVP formats on PICS), as illustrated in Figure 2. The DCA algorithm is able to establish the correlation criterion between the two groups of feature vectors, to extract their canonical correlation features. In this work, it is assumed that N training samples are collected from two classes, which are {0: non-pulsar, 1: pulsar}. For each sample, two feature vectors with 8 and 64 dimensions are extracted from two modalities, which are 1D statistical features extracted from the feature extraction program of HTRU2 and 2D feature plots extracted from PICS. Then, $X = R^{8\times N}$ and $Y = R^{64\times N}$ denote the data matrices containing the two feature sets. The X template is composed of N 1D vectors (8×1) which have the same format as that of HTRU2. Furthermore, the Y template is generated by extracting N feature plots (TVP or FVP or fusion of both, 64×64) using the unsupervised Convolutional Auto-Encoder (CAE) network as depicted in Figure 3. Note that each feature plot is dimensionally reduced to 8×8 through CAE and then resized to 64×1 by the *reshape* method in the Python toolkit. So, the dimension of Y is defined as $64 \times N$.

The fine tuned CAE architecture we used is shown in Figure 3. Then, after model training (where Epochs=50, Batch_size=128, the optimizer is Adadelta and the loss function is Binary_crossentropy) and testing, the loss rate was maintained at around 35%, which can make the model retain most of the main features of TVP or FVP while ensuring runtime efficiency. By using this CAE, the entire clustering algorithm performed well in the subsequent experiments in Sections 4.1 and 4.2. Meanwhile, the process of the DCA method is described as follows:

(i) The N columns of both data matrix are divided into c separate groups. Let S_{bx} be the between-class scatter matrix defined in Equations (1) and (2):

$$S_{bx_{(8\times 8)}} = \sum_{i=1}^{c} n_i(\bar{x_i} - \bar{x})(\bar{x_i} - \bar{x})^T = \Phi_{bx_{(8\times c)}}\Phi^T_{bx_{(c\times 8)}}, \quad (1)$$

$$\Phi_{bx_{(8\times C)}} = [(\bar{x_1} - \bar{x}), \sqrt{n_2}(\bar{x_i} - \bar{x}),\ldots,\sqrt{n_c}(\bar{x_c} - \bar{x})], \quad (2)$$

where $x_{ij} \in X$ denotes the p dimensional feature vector of the jth sample in the ith class, $\bar{x_i}$ represents the mean of the ith class and $\bar{x}$ signifies the whole feature set mean.

(ii) Find transformation matrix W_{bx} to transform the corresponding feature matrix $X_{8\times N}$ to $X'_{r\times N}$ as follows:

$$X'_{(r\times n)} = W^T_{bx(r\times 8)}X_{(8\times n)}, \quad (3)$$

$$S'_{bx} = W^T_{bx}S_{bx}W_{bx} = \Phi'_{bx}(\Phi'_{bx})^T = I, \quad (4)$$

where $\Phi'_{bx}(\Phi'_{bx})^T$ is a strictly diagonally dominant matrix, which represents the correlation between the ith class and the jth class. Similar to $X_{8\times N}$, find transformation matrix W_{by} to unitize the between-class scatter matrix S_{by} , which transforms $Y_{64\times N}$ to $Y'_{r\times N}$.

(iii) Maximize the pair-wise correlation across the feature sets X' and Y'. This requires the between-set covariance matrix ($S'_{xy} = X'Y'^T$) to be diagonal through singular value decomposition (SVD) as follows:

$$S'_{xy_{(r\times r)}} = U\Sigma V^T \Rightarrow (U\Sigma^{-\frac{1}{2}})^T S'_{xy}(V\Sigma^{-\frac{1}{2}})T = I. \quad (5)$$

The transformation matrices for X' and Y' are:

$$W_{cx} = U\Sigma^{-\frac{1}{2}}, \quad (6)$$

$$W_{cy} = V\Sigma^{-\frac{1}{2}}. \quad (7)$$

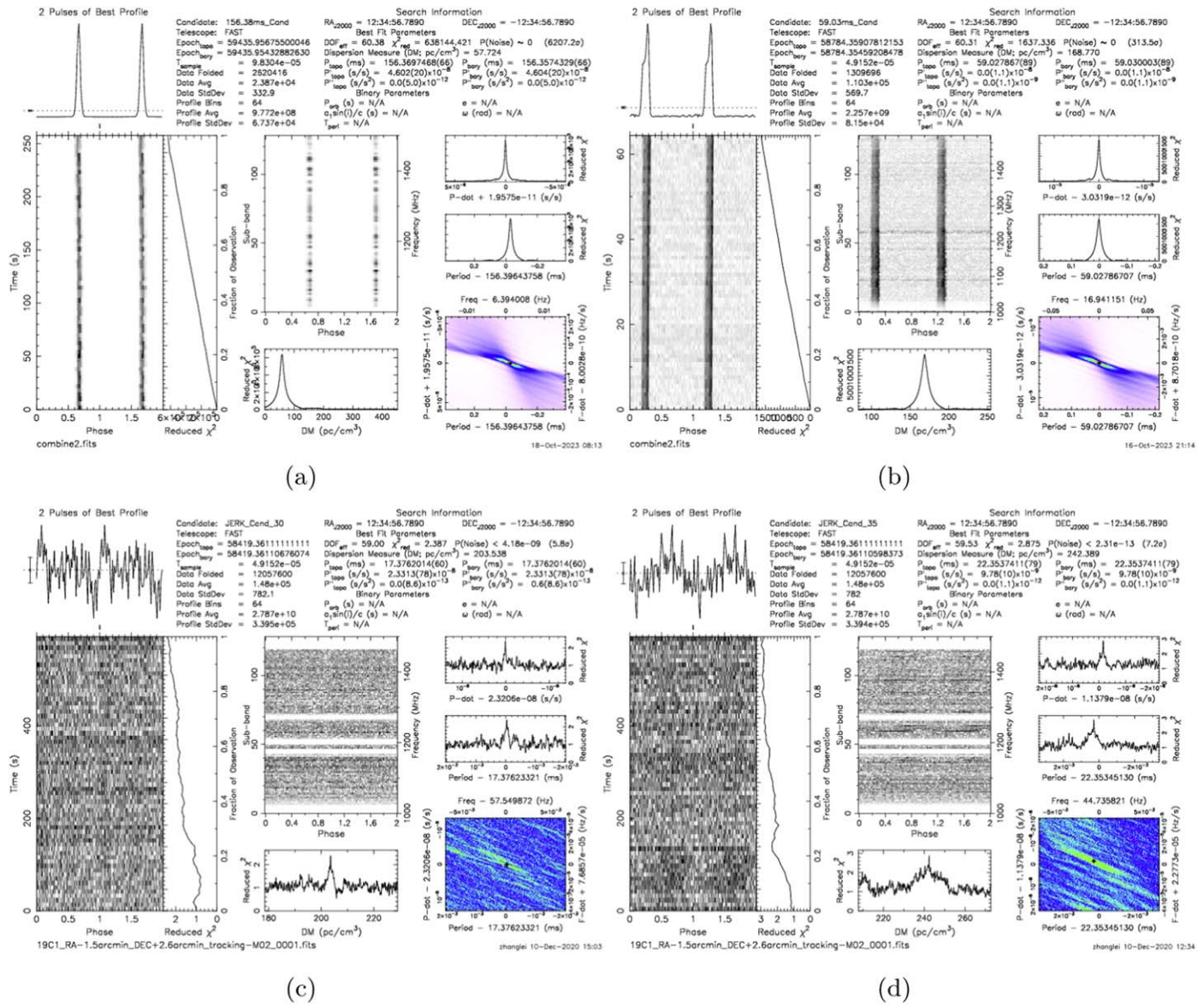

Figure 1. An example of the similarity between candidates. Note that (a) is the diagnostic plots of known pulsar J1906+0746 and (b) is those of known pulsar J1807-0847; (c) is the diagnostic plots of interference signal I and (d) is those of interference signal II.

Consequently, the final transformed feature sets are:

$$X^* = W_{cx}^T W_{bx}^T X, \tag{8}$$

$$Y^* = W_{cy}^T W_{by}^T Y. \tag{9}$$

(iv) The early feature-level fusion is performed by summing the final transformed feature sets, that is $Z = X^* + Y^*$, where Z called the Canonical Correlation Discriminant Features (CCDFs).

In this way, the data samples with fused features can be inputted into the hybrid clustering model in the next step.

3.2. Hybrid Clustering

Aiming to classify data into various homogeneous clusters in terms of their similarities, clustering methods are divided into different categories including density-based methods, partition-based methods and so on. As a representative partition-based clustering, K-Means is widely applied (Krishna & Narasimha Murty 1999). Due to its drawbacks (it is only applicable to distinguishing clusters with a hyper spherical data distribution and the clustering results are usually sensitive to the parameter settings), extended versions have been presented such as Arthur & Vassilvitski (2007) and Nguyen (2018). In addition, the

Table 1
The Detailed Information on the Plots in Figure 1

Parameter / Name	J0358+5413	J1915+1606	Interference Signal I	Interference Signal II
Telescope	FAST	FAST	FAST	FAST
Candidate	156.38 ms_cand	59.03 ms_cand	JERK _ cand 30	JERK _ cand 35
$\text{Epoch}_{\text{topo}}$	59435.95677812153	58784.35905500046	58419.36111111111	58419.36111111111
$\text{Epoch}_{\text{bary}}$	59435.95439208478	58784.35452882630	58419.36110676074	58419.36110598373
Data folded	2620416	1309696	12057600	12057600
Data Avg	2.387e+04	1.103e+05	1.48e+05	1.48e+05
Data StdDev	332.9	569.7	782.1	782
Profile Bins	64	64	64	64
Profile Avg	9.772e+08	2.257e+09	2.787e+10	2.787e+10
Profile StdDev	6.737e+04	8.15e+04	3.395e+05	3.394e+05
DOFeff	59.53	60.31	59.00	59.53
DM(pc/cm^3)	57.727	168.770	203.538	242.389
Ptopo(ms)	156.3697468	59.027867	17.3762014	22.3537411
P'topo(s/s)	$4.602(20) \times 10^{-8}$	$0.0(1.0) \times 10^{-8}$	$2.3313(78) \times 10^{-8}$	$9.78(10) \times 10^{-9}$
P''topo(s/s^2)	$0.0(5.0) \times 10^{-12}$	$0.0(1.0) \times 10^{-9}$	$0.0(8.6) \times 10^{-13}$	$0.0(1.1) \times 10^{-12}$
P_{Noise}	$0(6207.2\sigma)$	$0(313.5\sigma)$	$<4.18e-09(5.8\sigma)$	$<2.31e-13(7.2\sigma)$
P_{bary}(ms)	156.3274369(66)	59.030003(89)	17.3762014(60)	22.3537411(79)
Ptopo(ms)	156.3697468	59.027867	17.3762014	22.3537411
P'topo(s/s)	$4.604(20) \times 10^{-8}$	$0.0(1.0) \times 10^{-8}$	$2.3313(78) \times 10^{-8}$	$9.78(10) \times 10^{-9}$
P''topo(s/s^2)	$0.0(5.0) \times 10^{-12}$	$0.0(1.0) \times 10^{-9}$	$0.0(8.6) \times 10^{-13}$	$0.0(1.1) \times 10^{-12}$
$\text{R.A.}_{\text{J2000}}$	12:34:56.7890	12:34:56.7890	12:34:56.7890	12:34:56.7890
$\text{Decl.}_{\text{J2000}}$	−12:34:56.7890	−12:34:56.7890	−12:34:56.7890	−12:34:56.7890
Receiver Name	19 beam receiver	19 beam receiver	19 beam receiver	19 beam receiver
Antenna Gain	16.1k Jy-1	16.1k Jy-1	16.1k Jy-1	16.1k Jy-1
Bandwidth	500 MHz	500 MHz	500 MHz	500 MHz
Temperature	20 K	20 K	20 K	20 K
Terminal Name	psr	psr	psr	psr
Sampling Rate	49.152 μs	49.152 μs	49.152 μs	49.152 μs
Polarization channels	4	4	4	4
Number of channels	4096	4096	4096	4096

Density Peaks Clustering (DPC) algorithm adopted density peaks as features to quickly discover the potential cluster centers without any prior knowledge through drawing a 2D decision graph. More recently, another density-based algorithm using global and local consistency adjustable manifold distance (McDPC) was proposed by Wang et al. (2020b) to address the drawback (that it is easy to divide the same cluster into multiple microclusters corresponding to its multiple high-density points) of Clustering by Fast Search and Find of Density Peaks (CFSFDP, Rodriguez & Laio 2014). FMFHC is developed as a fusion of clustering methods based on DPC and K-Means.

The clustering process of FMFHC is summarized in Figure 4, in which the main steps are summarized as follows:

(i) To better adapt to different data structures, the k-nearest neighbor based mixed kernel function of a Radial Basis Function (RBF) and Polynomial (RBF_Poly) is used to calculate the density values of fused input data (mentioned in Section 3.1), as expressed in Equations (10) and (11).

$$K_{\text{RBF\_Poly}}(x_i, y_j) = \lambda \exp\left[-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right] + (1 - \lambda)(x_i \cdot y_j)((x_i \cdot y_j) + 1)^{q-1}, \quad q \geqslant 1,\ \lambda \in [0, 1] \tag{10}$$

where $K_{\text{RBF\_Poly}}(x_i, y_j)$ represents the mixed kernel function of data points x_i and x_j, λ signifies the weight of the RBF function, σ is the width of the RBF function and q is the order of the polynomial function. Although σ, q and λ cannot significantly improve the clustering effect, they can make the mixed kernel distance based similarity calculation more stable for multiple shapes of the data distribution if the λ value is reasonable. The values of these three parameters are determined ($\sigma = 1$, $q = 2$, $\lambda = 0.95$) by past experiences since there is no analytical method for the selection of fusion coefficients currently, as mentioned in Wang & Xu (2017) and

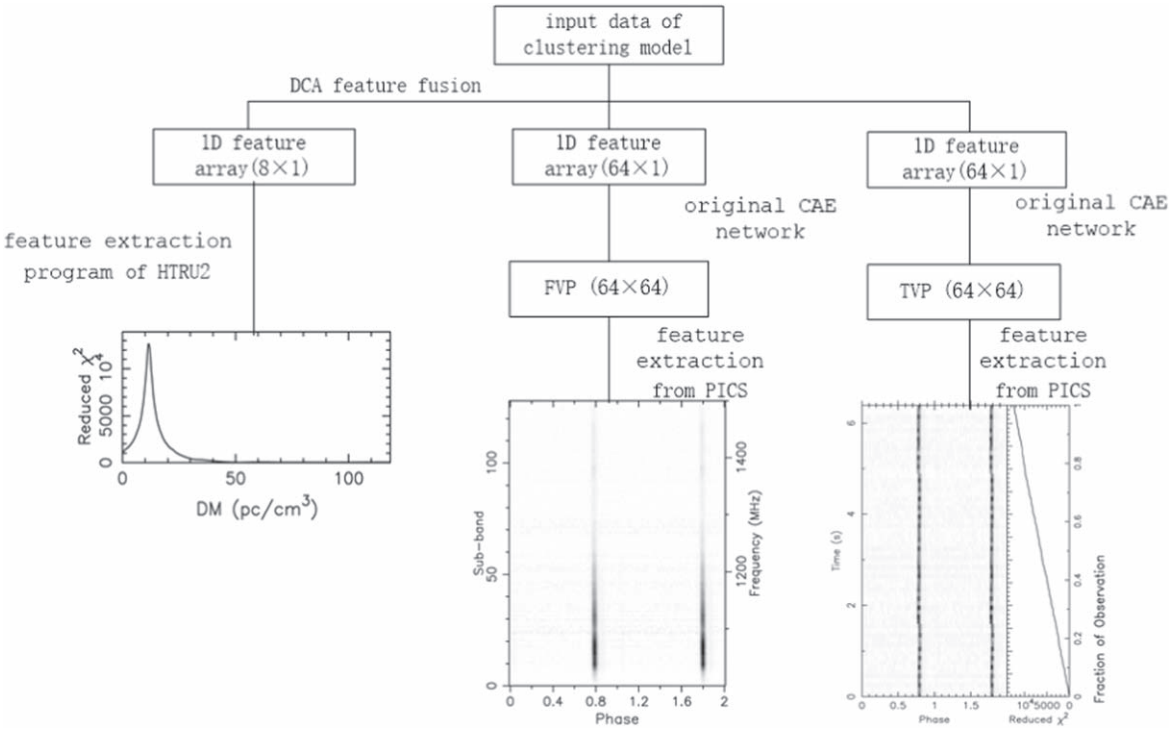

Figure 2. Diagram of the feature fusion model.

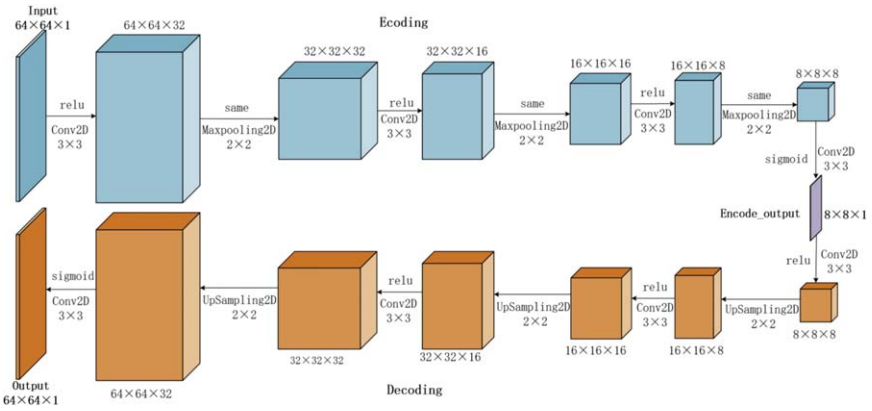

Figure 3. Network architecture of original CAE. Note that the encoding module of CAE has four hidden layers as follows. The first layer: one 2D convolution with $(3 \times 3) \times 32$ (output channels) and one 2D max-pooling with 2×2; the second layer: one 2D convolution with $(3 \times 3) \times 16$ and one 2D max-pooling with 2×2; the third layer: one 2D convolution with $(3 \times 3) \times 8$ and one 2D max-pooling with 2×2; the fourth layer: one 2D convolution with $(3 \times 3) \times 1$. The following value was used: the Input size, $64 \times 64 \times 1$; kernel, 3×3 pixels; padding, 1×1 pixels; the Encode_output size, $8 \times 8 \times 1$.

Huang et al. (2013).

$$\rho_i = \sum_{x_j \in \mathrm{KNN}_{(x_i)}} K_{\mathrm{RBF\_Poly}}(x_i, x_j), \tag{11}$$

where ρ_i denotes local density of data point x_i and $\mathrm{KNN}(x_i)$ refers to the KNN of x_i. Further, parameter δ_i of data point x_i is defined as Equation (12)

$$\delta_i = \min_{x_j : \rho_j \succ \rho_i} S_{\mathrm{Mah}}(x_i, x_j), \tag{12}$$

where $S_{\mathrm{Mah}}(x_i, x_j)$ signifies the Mahalanobis distance between x_i and x_j. In addition, the density threshold ρ_{outlier} is used for extracting outliers from the whole

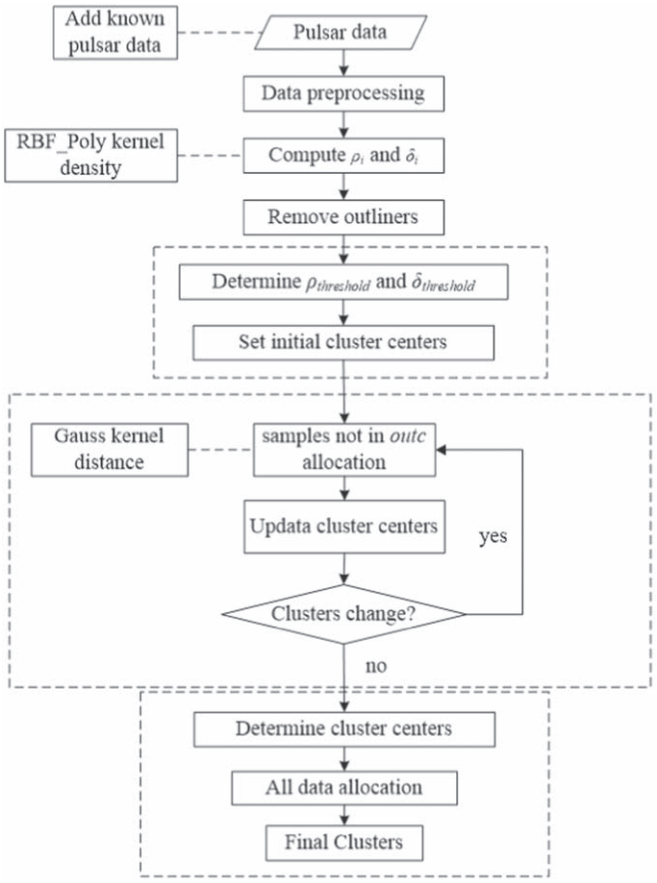

Figure 4. Flow chart of hybrid clustering scheme.

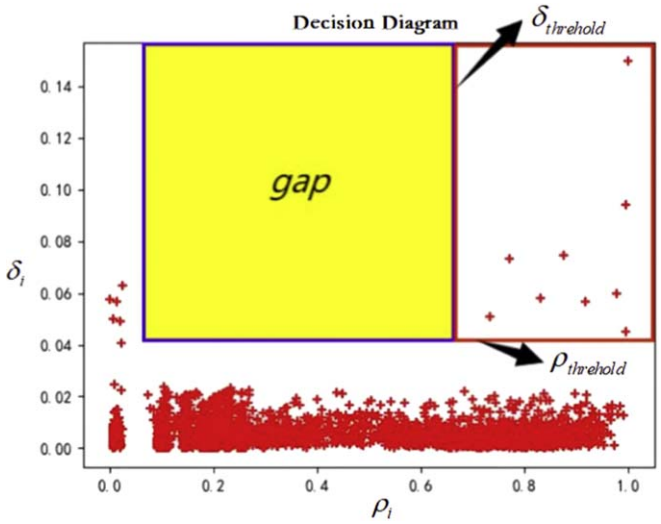

Figure 5. 2D decision graph based on ρ_i and δ_i.

data set, some of which may be special pulsars that need to be further investigated. An improved K-dist graph method is employed to determine ρ_{outlier} (Wang et al. 2020b). In the K-dist graph, outliers often are located at the leftmost local density level named *PL*, which comprises edge points of natural clusters with lower ρ and lower δ values and outliers with lower ρ and higher δ values. To further distinguish which data points in *PL* should be considered as outliers, ρ_{outlier} is determined from Equation (13)

$$PL \supseteq \begin{cases} \{\xi\text{: } PL_1,\ldots,PL_j\},\ \rho_{x_1},\ldots,\rho_{x_j} > \rho_{\text{outlier}} \\ \{outc\text{: } PL_{j+1},\ldots,PL_w\},\ \rho_{j+1},\ldots,\rho_n \leqslant \rho_{\text{outlier}} \end{cases}, \quad (13)$$

where PL_j denotes a data point in *PL*, *outc* means the set of outliers and *outc* is set in an autonomous manner, i.e., $\forall x_i \in outc$, $\forall x_j \in \xi$, $5\rho_{x_i} \leqslant \rho_{x_j}$.

(ii) The improved cluster center selection scheme of DPC is used with automatic determination of the number of clusters and their center points. All the values of ρ_i and δ_i ($x_i \notin outc$) are used for generating the 2D decision graph which helps to select the initial cluster centers automatically. In the derived decision graph as shown in Figure 5, the parameters ρ_{threhold} and δ_{threhold} are reasonably set as the truncation threshold to form a rectangle with a red border, in which all data points are selected as representative points (i.e., initial cluster centers). Moreover, the gap area with the yellow background separates these representative points from other points. Note that the representative points are also the multi-density center points, and the rest of the data points will be allocated to these center points to form intermediate microclusters. This representative point selection scheme is similar to that in the McDPC algorithm, which can divide the remaining samples into different density levels. However, it has better generalization performance and can identify more multi-density data sets compared to McDPC.

(iii) After the number of clusters k and the initial cluster centers are determined, the improved iterative optimization scheme of cluster centers of K-Means is used for all data point regroupings and final convergence. The distance between any point x_i ($x_i \notin outc$) and each cluster center in the current iteration is calculated based on an RBF kernel function. Note that RBF can improve the similarity measure between two points by mapping from measured distances to a high-dimensional space. Moreover, starting from the 2nd iteration, a weighted distance optimization is adopted for similarity measure as shown is Equations (14) and (15)

$$new\_\rho_{center_j} = \frac{Max\_\rho - Min\_\rho}{Max\_\rho - \rho_{center_j}}. \quad (14)$$

Here $new\_\rho_{center_j}$ denotes the weight value of cluster center $center_j$ used for distance optimization, and Max_ρ and Min_ρ signify the maximum and minimum density

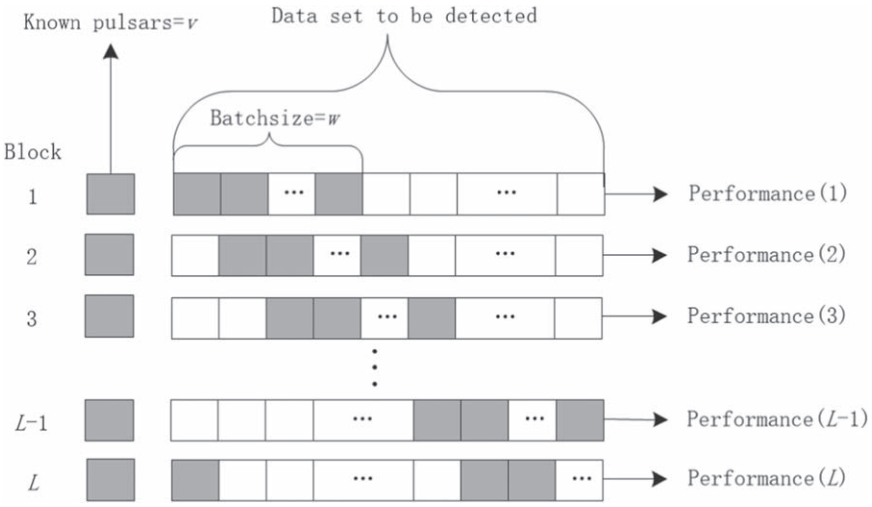

Figure 6. Data partition based on sliding window.

of data points not in *outc* respectively.

$$S_{new}(x_i, center_j) = \sqrt{2\left(1 - \exp\left(-\frac{\| x_i - center_j \|^2}{\sigma}\right)\right)} \times (new\_\rho_{center_j})^2, \quad (15)$$

where $S_{new}(x_i, center_j)$ signifies the weighted distance between $x_i (x_i \notin outc)$ and $center_j$. As a result, all the clusters in the current iteration and corresponding cluster centers are updated. The weighted distance makes data points move closer to cluster centers with relatively smaller density nearby, which is conducive to determining cluster boundaries.

For each iteration, the Sum of Squares of Errors (SSE) for all the points are updated. Then, the K-Means iterative process will enter the next update cycle of cluster centers until the SSE value has little change compared with the previous round. Finally, all the data points except *outc* are assigned to the k clusters.

The Parallel Hybrid Clustering Analyzer (PHCAL, Ma et al. 2022) combines the advantages of the McDPC and K-Means algorithms to ensure the stability and depth of data mining for pulsar candidates. Compared to PHCAL, the clustering process of FMFHC can improve the flexibility of determining initial cluster centers on data sets with an irregular shape distribution and get a more stable clustering and outliers.

3.3. Data Partition Strategy

Statistics show that the FAST 19 beam receiver can provide more than a million candidates per night, as mentioned in Liu et al. (2021) and Yin et al. (2022). To improve the time performance of FMFHC, it is essential to examine the parallel implementation of FMFHC based on the models, e.g., Message Passing Interface (MPI), SparkCore, etc. For this reason, the sliding window based data partition strategy for candidate data streams (Ma et al. 2022) is adopted on the basis of data structure. As can be seen in Figure 6, the window size of each round is fixed to Batchsize = w. Then, a relatively complete set is formed by selecting appropriate samples from the actual pulsars of various types, which will be added to the block to be detected (shadow areas) at a specific ratio (v: w) in each round. According to the clustering results in every block, the clusters whose pulsar sample proportion is greater than a certain threshold (e.g., 50%) will be regarded as pulsar data areas and entered into a unified list for further validation. In addition, it should be determined if the outliers screened out before clustering are special pulsars. The sliding window mode enables each data sample to appear in two or more blocks with multiple data distributions, so it becomes possible for some data points classified incorrectly in some blocks to be identified correctly in other blocks. Note that the specific parallelization schemes are not discussed in this work.

3.4. Time Complexity Analysis

As a combination of clustering methodology of density-based and partition-based methods, the serial clustering process of FMFHC is more complex than the K-Means and DPC algorithms. However, this deficiency can be made up as much as possible by using a reasonable parallelization method. Table 2 shows the time complexity of a serial clustering process of FMFHC and compares it with three other common serial algorithms, i.e., K-Means++ (Arthur & Vassilvitski 2007), McDPC (Wang et al. 2020b) and KNN (Peterson 2009).

Let the total number of samples of a data set be n, then: (i) The time complexity of K-Means++ is $O(nkTM)$, which can

Table 2
Time Complexity Statistics of FMFHC and Other Serial Algorithms

Algorithm	Parallel Mode of FMFHC	Serial Mode of FMFHC	K-Means++	McDpc	KNN
Time Complexity	$\lim_{G(p)\to 1} O((G(p)m)^2)$	$O(n^2)$, $if \sum_{i=1}^{4} C_i < \text{threshold}_1$	$O(nKTM)$	$O(n^2)$	$O(nM + nD^a)$

Note. T: the number of iterations; M: the characteristic number of elements; m: the number of samples of Block(i); k: the number of cluster centers; D: the nearest neighbor parameter.

be simplified to $O(n)$ as k, T and M are considered as constants. (ii) The time complexity of McDPC based on different density levels is $O(n^2)$ since the computing complexity of parameters ρ and δ is $O(n^2)$. (iii) The complexity of KNN is $O(nM + nD)$ as calculated from the worst case. (iv) The serial time complexity of FMFHC without applying the data partitioning strategy in Section 3.3 is $O(\sum_{i=1}^{4} nC_iL + n^2 + nKTM)$, where $O(\sum_{i=1}^{4} nC_iL)$ denotes the time complexity of the original CAE architecture which has four convolution layers, $O(n^2)$ signifies the time complexity of the multi-density center selection scheme and $O(nkTM)$ represents the time complexity of the improved cluster center iterative optimization of K-Means. Note that C_i is the time complexity of a single convolution layer $i\,(1 \leqslant i \leqslant 4)$ of a sample (that is $C_i = O(V_i^2 \times W_i^2 \times Cin_i \times Cout_i)$, where V_i denotes the size of output feature map of convolution layer i, W_i signifies the size of convolution kernel of convolution layer i, Cin_i represents the number of input channels, $Cout_i$ corresponds to the number of output channels), and L is the number of training iterations. In addition, the time complexity of the feature fusion process between the 1D feature arrays (8×1) and (64×1) is close to $O(n)$ according to Section 3.1, which could be neglected.

If $\sum_{i=1}^{c} C_i$ are small enough (i.e., $\sum_{i=1}^{4} C_i \leqslant threshold_1$) and n is large enough (i.e., $n > threshold_2$), the serial time complexity of FMFHC can be simplified to $O(n^2)$, where $\sum_{i=1}^{4} C_i$, k, T, M and L are considered as constants. Obviously, this is an idealized state and the complexity value is close to McDPC but higher than KNN and K-Means++. In terms of this premise, the time complexity of the parallel mode of FMFHC can be further discussed when using the sliding window based data partition strategy. As a result, the parallel complexity of FMFHC is $O((G(p)m)^2)$ in light of the Sun-Ni theorem (Sun & Ni 2002), where $G(p)$ denotes the factor and m signifies the number of samples in Block(i) with $m \ll n$. When the number of parallel nodes p tends to a certain threshold (close to the total number of divided blocks) and the communication delay tends to be ignored, the complexity value is simplified to $O(m^2)\,(G(p) \to 1)$. Therefore, if the communication overhead is very low or even negligible, the time complexity of the parallel mode of FMFHC is significantly lower than that of its serial version in theory, which will be verified in practice later. In addition, the speedup (S_p) and parallel efficiency (E_p) for the parallel version of FMFHC are defined as follows.

Table 3
Basic Information on Both Data Sets used in This Study

Data Set	Samples	Pulsars	Non-pulsars	R^a
HTRU2	17 898	1639	16 259	9.92:1
FAST data	157 616	78	157538	2019.71:1

Note. R: the class imbalance ratio of non-pulsars to pulsars.

$$S_p = \frac{T_s}{T_p} = \left(\frac{O(n^2)}{\lim_{G(p)\to 1} O((G(p)m)^2)} \right)^{,m \ll n, \sum_{i=1}^{4} C_i < \text{threshold}_1}, \tag{16}$$

$$E_p = \frac{S_p}{p}, \tag{17}$$

where T_s is the serial running time of FMFHC, and T_p is the running time of the parallel mode of FMFHC under p parallel nodes. In theory, the performance of the parallel mode of FMFHC will remain consistent for different data sizes and hardware resources, depending on the following two conditions: (i) The p value is close to a sufficiently big threshold; (ii) The ratio of communication delay in total running time is small enough to be neglected.

4. Experiments and Results

4.1. Datasets and Evaluation Metrics

Our algorithm was tested on both HTRU2 and FAST data sets. HTRU2 is an open telescope data set describing a sample of pulsar candidates collected during the HTRU Survey, which consists of 16 259 non-pulsar samples and 1639 pulsar samples. It is widely adopted to evaluate the performance of ML based classification algorithms. Lyon et al. (2016) made the HTRU2 data set available, which has been uploaded on the website.[8] The class imbalance ratio of HTRU2 is 9.92:1. It is widely adopted to evaluate the performance of ML based classification algorithms. Another FAST data set is obtained

[8] https://figshare.com/articles/data set/HTRU2/3080389/1

Table 4
Confusion Matrix

Actual Category Predicted Results	Positive	Negative
True data	True positive (TP)	False Negative (FN)
False data	False positive (FP)	True Negative (TN)

from the actual observation data of FAST (CRAFTS). The CRAFTS database is uploaded on the website.[9] In the FAST data set, 157 616 candidates with pfd files were collected from the survey, among which 78 were pulsar samples and 157 538 RFI samples. The class imbalance ratio of the FAST data set is 2019.71:1. Table 3 shows the basic information on both experimental data sets.

The evaluation metrics adopted for performing candidate classification usually are Precision, Recall and F1-Score. Table 4 shows the confusion matrix of the classification. Precision means the proportion of actual pulsar samples properly classified in all the candidates which are classified as positive, and Recall is the proportion of actual pulsars correctly classified. Precision and Recall are often inversely proportional (when Precision is high, Recall is usually low), so F1-Score can be used to reconcile this pair of metrics. Combined with the data partition strategy in Section 3.3, the overall performance metrics, i.e., $\text{Precision}_{\text{overall}}$, $\text{Recall}_{\text{overall}}$ and $\text{F1}-\text{Score}_{\text{overall}}$, are defined as follows:

$$\text{Precision}_{\text{overall}} = \frac{1}{L}\left(\sum_{l=1}^{L}\frac{\text{TP}_l}{(\text{TP}_l + FP_l)}\right), \tag{18}$$

$$\text{Recall}_l = \frac{\text{TP}_l}{\text{TP}_l + FN_l}(1 \leqslant l \leqslant L), \tag{19}$$

$$\text{F1}-\text{Score}_{\text{overall}} = \frac{1}{L}\left(\sum_{l=1}^{L}\frac{2 \times \text{Precision}_l \times \text{Recall}_l}{\text{Precision}_l + \text{Recall}_l}\right), \tag{20}$$

where TP_l, FP_l and FN_l respectively denote the number of True Positive, False Positive and False Negative cases in *Block*(l), and L is the total number of divided data blocks.

$$\text{Recall}_{\text{overall}} = \frac{\text{UTP}}{\text{TP} + \text{FN}}, \tag{21}$$

where $\text{UTP} = \text{TP}_1 \cup \text{TP}_2 \cup \text{TP}_3 \cdots \text{TP}_L$ means the union of identified pulsar samples in each data block, and TP and FN respectively denote the total number of True Positive and False Negative cases in entire data set. Note that, once a pulsar sample has been correctly identified in a *Block*(l), it will be counted to TP.

[9] http://groups.bao.ac.cn/ism/CRAFTS/202203/t20220310_683697.html

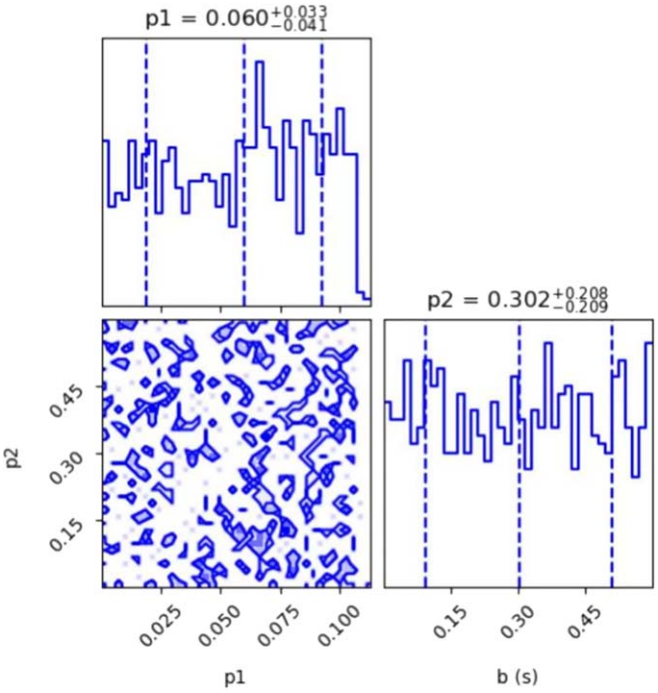

Figure 7. The ρ_{threhold} and δ_{threhold} triangle plot. Note that $p1$ denotes ρ_{threhold} and $p2$ denotes δ_{threhold}.

4.2. Clustering Effect Test using the HTRU2 Data Set

To validate the clustering result of our model, it has been experimented on with HTRU2 and compared with other single-modal candidate signal classifiers, including supervised and unsupervised learning algorithms implemented on HTRU2 in the mentioned literature. The data pre-processing was carried out first. According to the multi-modal fusion strategy in Section 3.1, all the new candidates in HTRU2 were formed by the fusion of the original 1D feature arrays and related 2D TVP arrays. Note that the 2D TVP arrays were extracted from other selected pfd files with very similar characteristics to corresponding HTRU2 samples. Moreover, 800 pulsar samples and 4259 non-pulsar samples were used for DCA algorithm training. Next, 1600 of the 1639 real pulsar samples of HTRU2 were randomly selected as a pulsar set s, while the remaining 39 pulsar samples were randomly dispersed to the non-pulsar samples of HTRU2 to form the data set to be detected. In terms of the data partition strategy in Section 3.3, the sliding window size was set to Batchsize=2 where the unit-size was 1161, then the entire data set to be detected was divided into (t_1, t_2, $\cdots$, t_{13}, t_{14}) based on unit-size, where t_1, t_2,…,$t_{13} = 1161$ but $t_{14} = 1205$. Consequently, the experimental data set consists of 14 data blocks, including {*Block*(1): [s, t_1, t_2], *Block*(2): [s, t_2, t_3], …, *Block*(13): [s, t_{13}, t_{14}], *Block*(14): [s, t_{14}, t_1]}. Each *Block*(i) was clustered separately, and the clusters with higher pulsar proportions than a certain value (e.g., $\geqslant$ 50%) in the

Table 5
Classification Results with Different Methods on HTRU2

Classfication	Method	Precision	Recall	F1-Score
Supervised	SVM	0.723	0.901	0.789
	PNCN	0.923	0.831	0.874
	RF	0.958	0.891	0.921
	KNN	0.952	0.875	0.909
Unsupervised/Semi-supervised	K-Means ++(k=35)	0.926	0.747	0.827
	McDpc	0.592	0.288	0.388
	PHCAL	0.946	0.905	0.881
	The parallel mode of FMFHC	**0.981**	**0.988**	**0.974**

clustering results were selected into the pulsar-like candidate list.

The values of ρ_{threhold} and δ_{threhold} have a significant influence on the F1-Score, so they have an important effect on the clustering. During the execution of our algorithm on HTRU2, the values of ρ_{threhold} and δ_{threhold} were randomly changed over 2000 rounds. Then, the 800 pairs of ρ_{threhold} and δ_{threhold} (the corresponding values of F1-Score were not less than 0.8) in the 2000 rounds were selected to make an actual two parameter triangle plot, as shown in Figure 7. The fitting result shows that there is no interaction between them. Moreover, both ρ_{threhold} and δ_{threhold} can be fine-tuned by using the heuristic methods (Wang et al. 2020a). On HTRU2, by analyzing the K-dist graph, we plot the respective inflection points of ρ_i and $\delta_i(x_i \notin outc)$ which are easily found. Consequently, ρ_{threhold} may be taken from a range of values ($\rho_i \in [0.2, 0.5]$) to obtain the best results, and it is the same for δ_{threhold} ($\delta_i \in [0.02, 0.075]$), and then the values were fine-tuned heuristically. For data sets with a single local density level, reasonable thresholds for ρ_i and δ_i will result in the inclusion of all obvious cluster centers, based on the DPC's assumption that data points with higher ρ and higher δ values should be selected as cluster centers. By such a heuristic method, it is not difficult to find a reasonable ρ_{threhold} value ($\rho_{\text{threhold}} = 0.3$) and δ value ($\delta_{\text{threhold}} = 0.024$). In addition, ρ_{outlier} is set to 0.00051 according to Section 3.2.

Table 5 shows the classification performance of FMFHC on HTRU2, compared with other unsupervised/semi-supervised algorithms (including K-Means++, Arthur & Vassilvitski 2007), McDPC (Wang et al. 2020b) and PHCAL (Ma et al. 2022) and supervised algorithms (including the RF in Burdwan 2019), and the KNN, SVM and PNCN in Xiao et al. (2020) implemented on HTRU2. Among all these unsupervised/semi-supervised and supervised algorithms, the parallel mode of FMFHC has the highest Precision (reaching 98.1%), Recall (reaching 98.8%) and F1-Score (reaching 97.4%). Moreover, upon executing several rounds of the control test in which 39 pulsar samples were randomly selected to form the data set to be detected, all the 39 pulsar samples could be detected (i.e., 100%) in each round by the parallel mode of FMFHC.

4.3. Robustness Test Using the FAST Data

The FAST data set was used to further verify the robustness and efficiency of FMFHC. Note that the data pre-processing on FAST data is very similar to that on HTRU2. At first, to change the class imbalance ratio between pulsar and non-pulsar samples in a single block, 1600 known pulsar samples from multiple types were prepared as a known sample set s' and added to this data set. Some of the pulsar samples in s' are known pulsars searched by CRAFTS during synchronization testing, and they can be found in the linked star catalog.[10] Moreover, the DM range of these pulsar samples is often set to [2, 1000] (cm^{-3}pc). Those remaining in s' were collected from international surveys such as HTRU. All of these pulsar samples are normal pulsars. According to Section 3.1, all the new candidates in FAST data (containing the above known sample set s') were formed by the DCA fusion of the 1D feature arrays and related 2D TVP arrays, which were extracted from corresponding pfd files by the feature extraction programs of HTRU2 and PICS. In addition, the known sample set s' (1600 real pulsar samples) and 10 000 non-pulsar samples were used for DCA algorithm training. Next, the sliding window size was also set to Batchsize=2 and the unit-size was 1251, then the original data set was divided into (g_1, g_2, $\cdots$, g_{125}, g_{126}), where g_1, g_2, $\cdots$, $g_{125} = 1251$ but $g_{126} = 1241$. As a result, the experimental data set consisted of 126 data blocks, i.e., Block(1):[s', g_1, g_2], Block(2):[s', g_2, g_3],..., Block(125): [s', g_{125}, g_{126}], Block(126):[s', g_{126}, g_1]. Note that the original 78 pulsar samples from the FAST data were randomly distributed in (g_1, g_2, $\cdots$, g_{125}, g_{126}). The subsequent clustering process is the same as that on HTRU2, refer to Section 3.2. The specific parameters $\rho_{\text{threhold}} = 0.3$, $\delta_{\text{threhold}} = 0.023$ and $\rho_{\text{outlier}} = 0.00051$.

Our algorithm has been experimented on by using a Linux cluster environment with seven physical computing nodes with seven Intel 6230 Xeon @ 2.1 GHz CPUs with 480 CPU cores (four Nvidia-GeForce-RTX-2080Ti, 5.3TB of total RAM, 3.6PB of total disk space). The system is running Linux 3.10.0-862.el7.x86_64, Python 3.8, Tensorflow 2, and MPI4py. The experimental results demonstrate that the highest number of pulsars identified in a round by the parallel mode of FMFHC achieves 76 of 78, with an average of 75 (Recall of 96.1%, Precision of 89.1% and F1-Score of 92.7%), compared with the PICS (Recall of 95%) and

[10] http://groups.bao.ac.cn/ism/CRAFTS/CRAFTS/

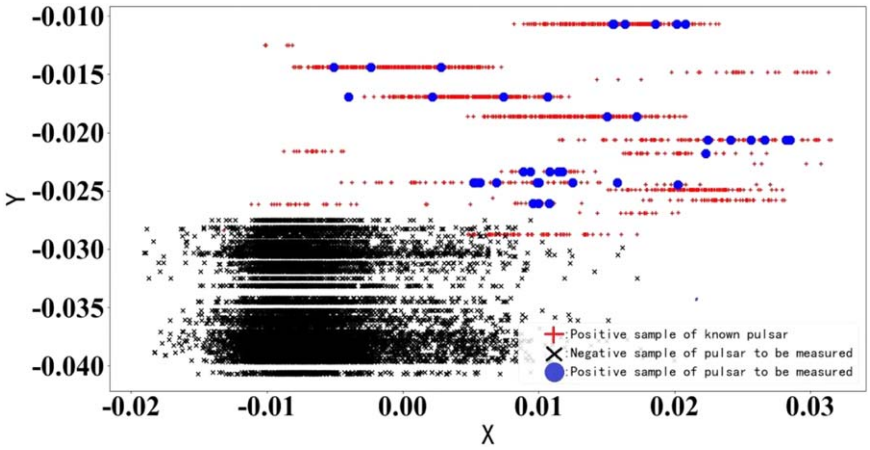

Figure 8. The clustering effect of FMFHC on FAST data. Note that the input data after multi-modal feature fusion are transformed into the arrays with two features *X* and *Y*.

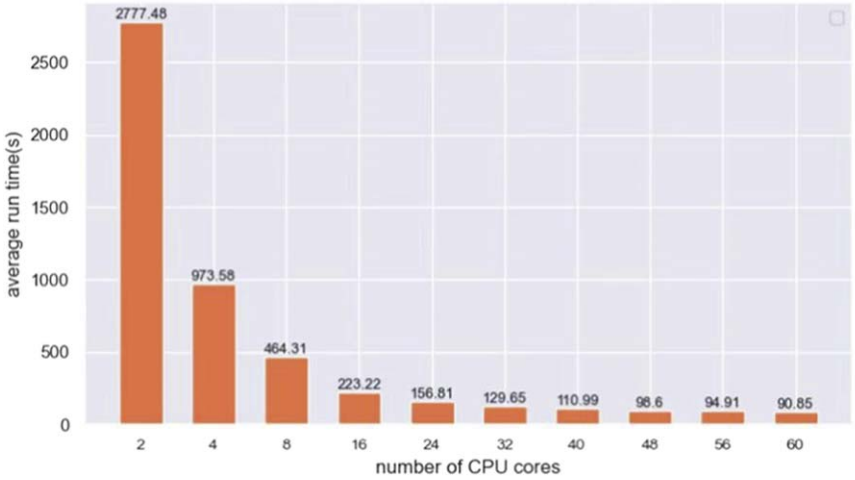

Figure 9. Average running time of parallel FMFHC based on MPI.

PICS-ResNet (Recall of 98%) in mentioned literature (Wang et al. 2019). Figure 8 shows the clustering effect of FMFHC on FAST data.

In addition, the running time data were collected under a different number of parallel nodes to further verify the time complexity of FMFHC. Note that the entire running time of FMFHC refers to the sum of execution time of data import, CAE based feature extraction, DCA based feature fusion and hybrid clustering, excluding the training time for the CAE and DCA models. Figure 9 depicts the average running time of parallel FMFHC with a different number of CPU cores. As can be seen from the figure, the average running time decreases with the increment of parallel nodes within limits. When the number of cores reaches 36, it drops to 90.85 s.

Table 6
Failure Modes and Codes

Code	How They Fail	Failure Cause	Existence	
			Yes	No
301	Isolated points	Classification criteria		✓
302	Collinearity	Signal feature selection		✓
303	Data standardization	Dimension of indicators		✓
304	Pattern similarity measure	Data Volume and Variety	✓	
305	Small pulsar samples clustering	Class imbalance between true pulsars and noise candidates		✓
306	Fast clustering of large sample data	Cluster center selection and manual setting of cluster numbers	✓	

After analysis, it is concluded that the non-pulsar based transients in FAST data could be roughly classified into other cosmic source signals such as Fast Radio Bursts (FRBs), RFI, low DM or narrowband signals, and very weak signals, and they can be identified by the signal shapes and signal features. Most of the non-pulsar signals incorrectly identified as pulsars here are broadband RFI. Moreover, the failure modes of clustering methods include the following six categories, where pattern similarity measure and fast clustering of large sample data still exist for FMFHC, as shown in Table 6.

5. Discussion

The overall performance analysis of FMFHC includes two parts: classification performance testing and running time evaluation. On HTRU2, FMFHC can ensure the clustering effect (the highest Precision, Recall and F1-Score) which looks better than other single-modal pulsar candidate classification methods in mentioned literature. On the FAST experimental data collected from CRAFTS, it still performs well (Precision and Recall) compared with the PICS and PICS-ResNet, and its parallel mode significantly reduces the execution time while ensuring the classification performance. It should be explained that HTRU2 is a public data set specifically designed to test the performance of pulsar candidate classifiers, where each data sample is carefully selected. However, actual FAST observation data are more diverse and complex in terms of data distribution than HTRU2 data. So, the performance of FMFHC on the CRAFTS database of FAST does not appear as excellent as that of HTRU2. Consequently, we can believe that FMFHC is effective for high-volume pulsar candidate data streams in actual scenarios. (i) A pulsar candidate data stream, regardless of its capacity, can be divided into fixed size blocks for parallel processing. (ii) It will further promote the discovery of outliers by clustering more meaningful classifications. (iii) It still could be improved with the optimization of data partition strategy and relevant parameters. In brief, our algorithm has provided a good theoretical and practical reference for sifting large numbers of pulsar candidate signals obtained from the FAST survey.

6. Conclusion

A multi-modal hybrid clustering method named FMFHC is presented for large numbers of pulsar candidates in this paper, whose contributions are summarized as follows:

(i) A feature-level fusion scheme based on the DCA algorithm is applied to maximize the separation between pulsar and non-pulsar candidates for large amounts of pulsar candidate data.
(ii) A combination of the multi-density peak identification scheme using a mixed kernel function for density computation and extension of cluster center iterative optimization scheme of K-Means is adopted to improve the clustering effect for data distribution with multiple shapes and screen out the outliers which could be special pulsars.
(iii) The semi-supervised learning mode without a large number of training samples and sliding window based data partition strategy are adopted to enhance the efficiency of the overall algorithm and reduce the execution time.

FMFHC is proved to be feasible, but there are still the following shortcomings: (i) Enough real data are still needed to further validate our algorithm. (ii) Due to limitations in experimental conditions, the actual performance comparison between our algorithm and other advanced parallel algorithms in recent years has not been conducted in an MPI experimental environment. In the future, we plan to connect the proposed algorithm to the pulsar distributed search pipeline based on PRESTO for application test and improvement. The program codes of FMFHC were uploaded on website.[11]

Acknowledgments

This research is partially supported by the National Key R&D Program of China (No. 2022YFE0133700), the National Natural Science Foundation of China (NSFC, grant Nos. 12273008, 11963003, 12273007 and 62062025), the National SKA Program of China (No. 2020SKA0110300), the Guizhou Province Science and Technology Support Program (General Project), No. Qianhe Support [2023] General 333, Science and Technology Foundation of Guizhou Province (Key Program, No. [2019]1432), the Guizhou Provincial Science and Technology Projects (Nos. ZK[2022]143 and ZK[2022]304), and the Cultivation project of Guizhou University (No. [2020] 76). This work made use of the data from FAST (Five-hundred-meter Aperture Spherical radio Telescope). FAST is a Chinese national mega-science facility, operated by National Astronomical Observatories, Chinese Academy of Sciences. We would like to thank Dr. Lyon for providing the publicly available data set and feature extraction scripts, and Zhu W. for providing the feature extraction program of PICS. They were very helpful for our research.

References

Arthur, D., & Vassilvitski, S. 2007, in Proc. Eighteenth Annual ACM-SIAM Symp. on Discrete Algorithms, 1027
Burdwan 2019, Detection of Pulsars using Machine Learning Algorithms – A Study in Emerging Trends in Artificial Intelligence for Internet of Things (Vellore, Tamil Nadu: Vellore Institute of Technology), 210
Burke-Spolaor, S., Bailes, M., Bates, S. D., et al. 2011, MNRAS, 416, 2465
Coenen, T. J., Leeuwen, J. V., Hessels, J. W. T., et al. 2014, A&A, 570, 16
Guo, P., Duan, F. Q., Wang, P., et al. 2019, MNRAS, 490, 5424
Han, J. L., Wang, C., Wang, P. F., et al. 2021, RAA, 21, 107

[11] https://github.com/You-Ziyi/FMFHC

Huang, H. J., Ding, S. F., Zhu, H., & Xu, X. Z. 2013, Journal of Computers, 8, 8
Jiang, P., Yue, Y. L., Gan, H. Q., et al. 2019, SCPMA, 62, 5
Krishna, K., & Narasimha Murty, M. 1999, ITSMC, 29, 433
Lee, K. J., Stovall, K., Jenet, F. A., et al. 2013, MNRAS, 433, 688
Liu, G. R., Li, Y. F., Bao, Z. L., Yin, Q., & Guo, P. 2021, in Int. Conf. on Intelligent Control and Information Processing (Dali: IEEE), 188
Lyon, R. J., Stappers, B. W., Cooper, S., Brooke, J. M., & Knowles, J. D. 2016, MNRAS, 459, 1104
Ma, Z., You, Z. Y., Liu, Y., et al. 2022, Univ, 8, 461
Morello, V., Barr, E. D., Bailes, M., et al. 2014, MNRAS, 443, 1651
Nguyen 2018, Computers & Security, 78, 60
Peterson, L. E. 2009, SchpJ, 4, 1883
Ransom, S., 2011 PRESTO: PulsaR Exploration and Search TOolkit, Astrophysics Source Code Library, ascl:1107.017
Rodriguez, A., & Laio, A. 2014, Sci, 344, 1492
Sun, X. H., & Ni, L. M. 2002, JPDC, 19, 27
Tan, C. M., Lyon, R. J., Stappers, B. W., et al. 2018, MNRAS, 474, 4571
Wang, H. F., Zhu, W. W., Guo, P., et al. 2019, SCPMA, 62, 1
Wang, H. L., & Xu, D. X. 2017, J. Cont. Sci. Eng., 2017, 12
Wang, P., Li, D., Clark, C. J., et al. 2021, SCPMA, 62, 129562
Wang, Y. Z., Wang, D., Pang, W., et al. 2020a, Neurocomputing, 400, 352
Wang, Y. Z., Wang, D., Zhang, X. F., et al. 2020b, Neural Computing and Applications, 32, 13465
Xiao, J. P., Li, X. R., Lin, H. T., & Qiu, K. B. 2020, MNRAS, 492, 2119
Yang, Z. C., Yu, C., Xiao, C., & Zhang, B. 2020, MNRAS, 492, 1421
Yin, Q., Wang, Y., Zheng, X., & Zhang, J. K. 2022, Electronics, 11, 2216
Yue, Y. L., Li, D., & Nan, R. D. 2013, Proc. Int. Astron. Union, 8, 577
Zeng, Q. G., Li, X. R., & Lin, H. T. 2020, MNRAS, 494, 3110
Zhang, S. C., Kong, X. C., Zhou, Y. Y., et al. 2021, RAA, 21, 257
Zhu, W. W., Berndsen, A., Madsen, E. C., et al. 2014, ApJ, 781, 117