Applying Hybrid Clustering in Multi-modal Pulsar Candidate Sifting for FAST Survey (Postprint)

Authors: Zi-Yi You, Yun-Rong Pan, Zhi Ma, Li Zhang, Shuo Xiao, Dan-Dan Zhang, Shi-Jun Dang, Ru-Shuang Zhao, Pei Wang, Ai-Jun Dong et al

Date: 2024-03-29T00:00:00+00:00

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.

Full Text

Preamble

Applying Hybrid Clustering in Pulsar Candidate Sifting with Multimodality for FAST Survey

Zi-Yi You[12], Yun-Rong Pan[1], Zhi Ma[1], Li Zhang[3], Shuo Xiao[12], Dan-Dan Zhang[1], Shi-Jun Dang[12], Ru-Shuang Zhao[14], Pei Wang[4], Ai-Jun Dong[12], 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] Guizhou Software Engineering Research Center, Guiyang 550000, China
[7] Guizhou Provincial Science and Technology Department, Guiyang 550001, China

Received 2023 September 8; revised 2023 October 27; accepted 2023 November 10; published 2024 March 6

Abstract

Pulsar search serves as the foundation for pulsar navigation, gravitational wave detection, and other research areas. Currently, the volume of pulsar candidates collected by the Five-hundred-meter Aperture Spherical radio Telescope (FAST) is growing explosively, posing significant challenges for its pulsar candidate filtering system. In particular, multi-view heterogeneous data and the severe class imbalance between true pulsars and non-pulsar candidates negatively impact traditional single-modal supervised classification methods. This study presents a multi-modal, semi-supervised learning-based pulsar candidate sifting algorithm that adopts a hybrid ensemble clustering scheme combining density-based and partition-based methods, integrated with a feature-level fusion strategy for input data and a data partition strategy for parallelization. Experiments on both the High Time Resolution Universe Survey II (HTRU2) and actual FAST observation data demonstrate that the proposed algorithm can excellently identify pulsars. On HTRU2, the precision and recall rates of its parallel mode reach 0.981 and 0.988, respectively. On FAST data, these metrics reach 0.891 and 0.961, respectively, while the running time also decreases significantly with the increment of parallel nodes within limits. Thus, we conclude that our algorithm offers a feasible approach for large-scale pulsar candidate sifting in FAST drift scan observations.

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

1. Introduction

Numerous 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), and others. Compared to previous surveys, modern ones tend to employ more sensitive detection techniques, larger survey areas, improved data analysis methods, and collaborative efforts such as CRAFTS. These surveys often produce a large number of potential pulsar candidates, yet only a very small proportion (nearly one in ten thousand) are identified as real pulsars due to substantial interference signals. Therefore, it is critical to reduce the retention of numerous non-pulsar signals without losing pulsar-like samples. This issue can be addressed at two stages of pulsar search pipelines: (i) Signal processing: removing Radio Frequency Interference (RFI) signals from observational data as much as possible (Morello et al. 2014; Yang et al. 2020) and optimizing important parameters such as signal-to-noise ratio (SNR) detections; and (ii) Candidate selection: minimizing the labor of further observations by automatically and accurately filtering pulsar-like samples among large numbers of candidates using advanced artificial intelligence techniques. This paper focuses on the latter.

Existing pulsar candidate selection methods based on artificial intelligence can be classified into three categories according to their principles. Traditional scoring methods constitute the first category, exemplified by the Pulsar Evaluation Algorithm for Candidate Extraction (PEACE, Lee et al. 2013). The second category comprises Machine Learning (ML) based classifiers that typically outperform the first category (e.g., Morello et al. 2014; Lyon et al. 2016; Tan et al. 2018). 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). 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 handle the class imbalance problem. However, the features used in these methods often depend heavily on human experience, and their classification performance may be adversely affected. For instance, some classifiers extract features only from the Dispersion Measure (DM) and pulse profile curve, which can lead to incorrect identification of some RFIs as pulsars. As the radio environment becomes more complex, it is increasingly difficult to effectively distinguish pulsar candidates from non-pulsar candidates using only statistical features. In practice, pulsars can be successfully identified by human experts observing the corresponding diagnostic plots. Inspired by this, the third category utilizes diagnostic plots as inputs for image recognition models or multi-method ensemble models including image recognition, enabling "pulsar-like" patterns to be learned automatically from diagnostic sub-graphs 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). These methods demonstrate better generalization ability compared to ML-based models in the second category. Among them, the methods described in Wang et al. (2019) and Zeng et al. (2020) are primarily 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 composed of five classifiers, which was incorporated into 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 candidates without any intermediate labels processed from FAST data. However, these methods are mostly based on a single mode, and there is currently little literature on multi-modal methods applied to astronomical data mining, particularly pulsar identification. Zhang et al. (2021) proposed an early fusion-based pulsar image identification framework with smart undersampling, evaluated on the HTRU Medlat data set. In this work, we design a semi-supervised learning and Feature-level Multi-modal Fusion based Hybrid Clustering (FMFHC) scheme for large-scale candidate sifting through FAST pulsar search pipelines, building upon Wang et al. (2019) and Ma et al. (2022).

A typical pulsar search pipeline for FAST drift scan observed data roughly includes the following steps: (i) eliminating 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 to further search for periodic signals; (iv) sifting these periodic signals and outputting candidates to files (e.g., with suffix pfd); and (v) folding the data periodically and outputting candidate images. In practice, numerous non-pulsar candidates remain in step (iv), including RFI. Our algorithm is introduced at the sifting stage of step (iv), aiming to address the following issues in pulsar candidate selection: (i) For some aforementioned methods (e.g., supervised ML-based candidate signal classifiers and DL-based diagnostic subplot recognition models), the cost of obtaining large amounts of labeled data (with an extremely imbalanced proportion between real pulsar and non-pulsar samples) and periodic training (to avoid overfitting and underfitting) is too high, and these methods are all based on binary classification; and (ii) In the pulsar search process, there are usually multi-view heterogeneous candidate data containing various types and attributes, making it difficult to mine 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 measures involved. Section 3 presents the components of the overall algorithm in detail. Section 4 illustrates the experimental data sets, data pre-processing methods, and results. Section 5 provides discussion. Finally, Section 6 presents conclusions and future work.

2.1. Pulsar Candidate Features

The extraction of candidate features is crucial for maximizing the separation between non-pulsar and pulsar candidates. We assume the pulsar candidates were processed by software pipelines based on the PulsaR Exploration and Search TOolkit (PRESTO, Ransom 2011; Yue et al. 2013), which implements similar search steps for advanced telescope systems such as FAST. In terms of statistical features, eight new features are 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 with an ML classifier. Furthermore, the High Time Resolution Universe II (HTRU2) data set was used in the work of 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: one-dimensional (1D) data array summed profile and DM curve, and 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. Experiments implemented on FAST pulsar survey data (Wang et al. 2019) demonstrate that PICS-ResNet can achieve a higher recall rate of 98% compared to PICS (which achieves 95%).

2.2. Similarity Measure

A candidate believed to be a real pulsar must have statistical features (e.g., standard deviation and skewness of pulse profile) and diagnostic subplots (e.g., 2D FVP) very similar to those of other known pulsars. This motivates our design of a multi-modal clustering algorithm for large amounts of pulsar candidate data. Figure 1 [Figure 1: see original paper] 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. 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 [Figure 8: see original paper].

3.1. Feature Fusion Strategy

By fusing different features of multiple modalities extracted from a single candidate, feature fusion methods can further refine features with higher discrim-

ination. Among them, Discriminant Correlation Analysis (DCA) is a linear method that maximizes the pair-wise correlation between two feature sets while possessing very low computational complexity. Based on the DCA algorithm, our objective is to pre-process candidate data extracted from pfd files by fusing features from different modalities of the same object before feeding them into 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 [Figure 2: see original paper]. The DCA algorithm can 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: {0: non-pulsar, 1: pulsar}. For each sample, two feature vectors with 8 and 64 dimensions are extracted from two modalities: 1D statistical features from the feature extraction program of HTRU2 and 2D feature plots from PICS. Then, X = ${}^{8}\times$ and Y = ${}^{64}\times$ denote the data matrices containing the two feature sets. The X template is composed of N 1D vectors (8×1) with the same format as 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 depicted in Figure 3 [Figure 3: see original paper]. 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. Thus, the dimension of Y is defined as $64 \times$ N. The fine-tuned CAE architecture we used is shown in Figure 3. After model training (where Epochs=50, Batch_{size}=128, optimizer is Adadelta, and loss function is Binary_{crossentropy}) and testing, the loss rate was maintained at around 35%, which allows the model to retain most of the main features of TVP or FVP while ensuring runtime efficiency. Using this CAE, the entire clustering algorithm performed well in subsequent experiments in Sections 4.1 and 4.2. Meanwhile, the DCA method proceeds as follows: (i) The N columns of both data matrices are divided into c separate groups. Let S_{bx} be the between-class scatter matrix defined in Equations (1) and (2). (ii) Find transformation matrix W_{by} to unitize the between-class scatter matrix S_{by}, which transforms $Y_{64}\times$ to Y' . (iii) Maximize the pair-wise correlation across the feature sets. This requires the between-set covariance matrix to be diagonalized through singular value decomposition (SVD). (iv) Early feature-level fusion is performed by summing the final transformed feature sets, that is Z = X* + Y*, where Z is called the Canonical Correlation Discriminant Features (CCDFs). In this way, data samples with fused features can be input into the hybrid clustering model in the next step.

3.2. Hybrid Clustering

Aiming to classify data into various homogeneous clusters based on their similarities, clustering methods are divided into different categories including density-based methods, partition-based methods, and others. As a representative partition-based clustering algorithm, K-Means is widely applied (Krishna

& Narasimha Murty 1999). Due to its drawbacks (it is only applicable to distinguishing clusters with a hyperspherical data distribution and the clustering results are usually sensitive to parameter settings), extended versions have been presented such as Arthur & Vassilvitski (2007) and Nguyen (2018). In addition, the Density Peaks Clustering (DPC) algorithm adopts density peaks as features to quickly discover potential cluster centers without any prior knowledge by 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 of Clustering by Fast Search and Find of Density Peaks (CFSFDP, Rodriguez & Laio 2014) that it is easy to divide the same cluster into multiple microclusters corresponding to its multiple high-density points. 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 [Figure 4: see original paper], with the main steps 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). The mixed kernel function is defined as:

$$K_{RBF\_Poly}(x_i, x_j) = \lambda \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right) + (1-\lambda)(x_i \cdot x_j + 1)^q$$

where $K_{RBF\_Poly}(x_i, x_j)$ 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 data distribution if the λ value is reasonable. The values of these three parameters are determined ($\sigma = 1$, $q = 2$, $\lambda = 0.95$) based on past experience since there is currently no analytical method for selecting fusion coefficients, as mentioned in Wang & Xu (2017) and Huang et al. (2013).

The local density ρ_i of data point x_i is defined as Equation (12):

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

where ρ_i denotes local density of data point x_i and $KNN(x_i)$ refers to the K-nearest neighbors of x_i.

Further, parameter δ_i is defined as the minimum distance between x_i and any other point with higher density:

$$\delta_i = \min_{j:\rho_j > \rho_i} d(x_i, x_j)$$

where $d(x_i, x_j)$ signifies the Mahalanobis distance between x_i and x_j.

In addition, the density threshold $\rho_{outlier}$ is used for extracting outliers from the entire data set, some of which may be special pulsars that need to be investigated. An improved K-dist graph method is employed to determine $\rho_{outlier}$ (Wang et al. 2020b). In the K-dist graph, outliers are often 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 outliers, $\rho_{outlier}$ is determined from Equation (13):

$$\rho_{outlier} = \max\{\rho_i | \forall x_i \in PL, \forall x_j \in \xi, r_i \leq 5r_j\}$$

where PL_j denotes a data point in PL, and ξ represents the set of outliers determined in an autonomous manner.

(ii) The improved cluster center selection scheme of DPC is used with automatic determination of the number of clusters and their center points. All values of ρ_i and δ_i ($x_i \notin outc$) are used to generate the 2D decision graph which helps select the initial cluster centers automatically. In the derived decision graph shown in Figure 5 [Figure 5: see original paper], the parameters $\rho_{threshold}$ and $\delta_{threshold}$ 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 in Equations (14) and (15):

$$\text{new\_}\rho_{center_j} = \frac{\rho_{center_j} - r_{Min}}{r_{Max} - r_{Min}}$$

where $\text{new\_}\rho_{center_j}$ denotes the weight value of cluster center j used for distance optimization, and r_{Max} and r_{Min} signify the maximum and minimum density of data points not in $outc$, respectively.

The weighted distance is calculated as:

$$S_{new}(x_i, center_j) = \|x_i - center_j\| \cdot (1 - \text{new\_}\rho_{center_j})$$

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 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 points is updated. Then, the K-Means iterative process enters the next update cycle of cluster centers until the SSE value shows little change compared with the previous round. Finally, all 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 irregular shape distributions and achieve more stable clustering and outlier detection.

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 models such as Message Passing Interface (MPI) and SparkCore. For this reason, the sliding window based data partition strategy for candidate data streams (Ma et al. 2022) is adopted based on data structure. As shown in Figure 6 [Figure 6: see original paper], the window size of each round is fixed to Batchsize = w. Then, a relatively complete set is formed by selecting appropriate samples from actual pulsars of various types, which are added to the block to be detected (shadow areas) at a specific ratio (v:w) in each round. According to the clustering results in each block, clusters whose pulsar sample proportion exceeds a certain threshold (e.g., 50%) are regarded as pulsar data areas and entered into a unified list for further validation. In addition, it should be determined whether 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, making it possible for some data points classified incorrectly in some blocks to be identified correctly in other blocks. Note that specific parallelization schemes are not discussed in this work.

3.4. Time Complexity Analysis

As a combination of density-based and partition-based clustering methodologies, the serial clustering process of FMFHC is more complex than the K-Means

and DPC algorithms. However, this deficiency can be compensated as much as possible by using a reasonable parallelization method. Table 2 shows the time complexity of the serial clustering process of FMFHC and compares it with three other common serial algorithms: K-Means++ (Arthur & Vassilvitski 2007), McDPC (Wang et al. 2020b), and KNN (Peterson 2009). Let the total number of samples in a data set be n. Then: (i) The time complexity of K-Means++ is $O(nkTM)$, which can be simplified to $O(n)$ when k, T, and M are considered 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 $O(n^2 + nkTM + \sum_{i=1}^{L} C_i V_i W_i^2 C_{in_i} C_{out_i})$, where $\sum_{i=1}^{L} C_i V_i W_i^2 C_{in_i} C_{out_i}$ 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 iterative optimization of K-Means. Note that C_i is the time complexity of a single convolution layer, where V_i denotes the size of the output feature map of convolution layer i, W_i signifies the size of the convolution kernel of convolution layer i, C_{in_i} represents the number of input channels, and C_{out_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 $(8\$\times 1) and (64\times\$1)$ is close to $O(n)$ according to Section 3.1, which could be neglected. If $\sum_{i=1}^{L} C_i$ are small enough and n is large enough, the serial complexity of FMFHC can be simplified to $O(n^2)$, where $\sum_{i=1}^{L} C_i$, k, T, M, and L are considered 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 the time complexity of the parallel mode of this premise, FMFHC can be further discussed when using the sliding window based data partition strategy. As a result, the parallel complexity of FMFHC is $O\left(\frac{n^2}{p} + G(p)\right)$ according to Sun & Ni (2002), where $G(p)$ denotes the communication factor and p signifies the number of parallel nodes. 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 $O(m^2)$ where $m = n/p$ is the number of samples per block. Therefore, if the communication overhead is very low or even negligible, 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:

$$S_p = \frac{T_s}{T_p}, \quad E_p = \frac{S_p}{p}$$

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 two conditions: (i) The p value is close to a sufficiently large threshold; and (ii) The ratio of communication delay in total running time is small enough to be neglected.

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, consisting of 16,259 non-pulsar samples and 1,639 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. The class imbalance ratio of HTRU2 is 9.92:1. Another FAST data set was obtained from actual observation data of FAST (CRAFTS). The CRAFTS database is uploaded on the website. 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 were 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 candidate classification are typically Precision, Recall, and F1-Score. Table 4 shows the confusion matrix of the classification. Precision means the proportion of actual pulsar samples properly classified among all candidates 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 these metrics. Combined with the data partition strategy in Section 3.3, the overall performance metrics—$\text{Precision}_{overall}$, $\text{Recall}_{overall}$, and $\text{Score}_{overall}$—are defined as follows:

$$\text{Precision}_{overall} = \frac{\sum_{l=1}^{L} TP_l}{\sum_{l=1}^{L} (TP_l + FP_l)}$$

$$\text{Recall}_{overall} = \frac{|UTP|}{|TP| + |FN|}$$

$$\text{F1-Score}_{overall} = 2 \cdot \frac{\text{Precision}_{overall} \cdot \text{Recall}_{overall}}{\text{Precision}_{overall} + \text{Recall}_{overall}}$$

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. $UTP = TP_1 \cup TP_2 \cup TP_3 \cdots TP_L$ 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 the entire

data set. Note that once a pulsar sample has been correctly identified in a Block(l), it will be counted in TP.

4.2. Clustering Effect Test using the HTRU2 Data Set

To validate the clustering result of our model, we experimented on HTRU2 and compared it with other single-modal candidate signal classifiers, including supervised and unsupervised learning algorithms implemented on HTRU2 in the mentioned literature. Data pre-processing was carried out first. According to the multi-modal fusion strategy in Section 3.1, all new candidates in HTRU2 were formed by fusing 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 4,259 non-pulsar samples were used for DCA algorithm training. Next, 1,600 of the 1,639 real pulsar samples of HTRU2 were randomly selected as a pulsar set s, while the remaining 39 pulsar samples were randomly dispersed among 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 (t1, t2, $\cdots$, t13 = 1161 but t14 = 1205. Consequently, the experimental data set consists of 14 data blocks, including {Block(1): [s, t1, t2], Block(2): [s, t2, t3], Block(13):[s, t13, t14], Block(14):[s, t14, t1]}. Each Block(i) was clustered separately, and the clusters with higher pulsar proportions than a certain value (e.g., 50%) were selected. The values of $\rho_{threshold}$ and $\delta_{threshold}$ 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 $\rho_{threshold}$ and $\delta_{threshold}$ were randomly changed over 2000 rounds. Then, the 800 pairs of $\rho_{threshold}$ and $\delta_{threshold}$ (with corresponding F1-Score values not less than 0.8) in the 2000 rounds were selected to create an actual two-parameter triangle plot, as shown in Figure 7 [Figure 7: see original paper]. The fitting result shows that there is no interaction between them. Moreover, both $\rho_{threshold}$ and $\delta_{threshold}$ can be fine-tuned using heuristic methods (Wang et al. 2020a). On HTRU2, by analyzing the K-dist graph, we plot the respective inflection points of ρ_i and δ_i ($x_i \notin outc$) which are easily found. Consequently, $\rho_{threshold}$ may be taken from a range of values ($\rho_i \in [0.2, 0.5]$) to obtain the best results, and the same applies to $\delta_{threshold}$ ($\delta_i \in [0.02, 0.075]$), with values 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 DPC' s assumption that data points with higher ρ and higher δ values should be selected as cluster centers. Using such a heuristic method, it is not difficult to find a reasonable $\rho_{threshold}$ value ($\rho_{threshold} = 0.3$) and $\delta_{threshold}$ value ($\delta_{threshold} = 0.024$). In addition, $\rho_{outlier} = 0.00051$ 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 RF in Burdwan 2019), and KNN, SVM, and PNCN in Xiao et al. (2020). Among all these unsupervised/semi-supervised and supervised algorithms implemented on HTRU2, 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 control tests in which 39 pulsar samples were randomly selected to form the data set to be detected, all 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 data pre-processing on FAST data is very similar to that on HTRU2. First, to change the class imbalance ratio between pulsar and non-pulsar samples in a single block, 1,600 known pulsar samples from s'and multiple types were prepared as a known sample set added to this data set. Some pulsar samples in s' are known pulsars searched by CRAFTS during synchronization testing, and they can be found in the linked star catalog. 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 these pulsar samples are normal pulsars. According to Section 3.1, all new candidates in FAST data (containing the above known s') were formed by the DCA fusion of the 1D sample set feature arrays and related 2D TVP arrays, extracted from corresponding pfd files by the feature extraction programs of HTRU2 and PICS. In addition, the known sample set (1,600 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 (g1, g2, $\cdots$, g125, g126), where g1, g2, $\cdots$, g125 = 1251 but g126 = 1241. As a result, the experimental data set consisted of 126 data blocks, i.e., Block(1):[s' , g1, g2], Block(125):[s' , g125, g126], Block(126):[s' , g126, g1]. Note that the original 78 pulsar samples from the FAST data were randomly distributed in (g1, g2, $\cdots$, g125, g126). The subsequent clustering process is the same as that on HTRU2, with parameters specific to $\rho_{threshold} = 0.3$, $\delta_{threshold} = 0.023$, and $\rho_{outlier} = 0.00051$.

Our algorithm was tested using a Linux cluster environment with seven physical computing nodes, each 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 runs Linux 3.10.0-862.el7.x86_{64}, Python 3.8, Tensorflow 2, and MPI4py. The highest number of pulsars identified in a round by the parallel mode of FMFHC achieved 76 of 78, with an average of 75 (Recall of 96.1%, Precision of 89.1%, and F1-Score of 92.7%), compared with PICS (Recall of 95%) and PICS-ResNet (Recall of 98%) in the mentioned literature (Wang et al. 2019). Figure 8 [Figure 8: see original paper] shows the clustering effect of FMFHC on FAST data.

In addition, running time data were collected under different numbers 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 for 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 [Figure 9: see original paper] depicts the average running time of parallel FMFHC with different numbers of CPU cores. As shown in 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 shows failure modes and how they fail. After analysis, it is concluded that non-pulsar based transients in FAST data can be roughly classified into other cosmic source signals such as Fast Radio Bursts (FRBs), RFI, low DM or narrowband signals, and very weak signals, which can be identified by signal shapes and features. Most non-pulsar signals incorrectly identified as pulsars here are broadband RFI. Moreover, the failure modes of clustering methods include 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 classification performance testing and running time evaluation. On HTRU2, FMFHC ensures excellent clustering effect (the highest Precision, Recall, and F1-Score), appearing better than other single-modal pulsar candidate classification methods in the mentioned literature. On FAST experimental data collected from CRAFTS, it still performs well (Precision and Recall) compared with PICS and PICS-ResNet, and its parallel mode significantly reduces execution time while ensuring classification performance. It should be noted 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 data distribution than HTRU2 data. Consequently, the performance of FMFHC on the CRAFTS database of FAST does not appear as excellent as that on HTRU2. Nevertheless, we can conclude 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 through more meaningful classifications; and (iii) It could still be improved with optimization of the data partition strategy and relevant parameters. In brief, our algorithm provides 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, with contributions summarized as follows: (i) A feature-level fusion scheme based on the DCA algorithm is applied to maximize 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 an extended cluster center iterative optimization scheme of K-Means is adopted to improve clustering effect for data distributions with multiple shapes and screen out outliers that could be special pulsars; and (iii) The semi-supervised learning mode without a large number of training samples and the sliding window based data partition strategy are adopted to enhance the efficiency of the overall algorithm and reduce execution time.

FMFHC is proven feasible, but still has shortcomings: (i) Enough real data are still needed to further validate our algorithm; and (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 testing and improvement. The program codes of FMFHC were uploaded to the website.

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, the Science and Technology Foundation of Guizhou Province (Key Program, 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 data from FAST (Five-hundred-meter Aperture Spherical radio Telescope). FAST is a Chinese national mega-science facility operated by the National Astronomical Observatories, Chinese Academy of Sciences. We 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, which 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 (Vel-

lore, 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
Huang, H. J., Ding, S. F., Zhu, H., & Xu, X. Z. 2013, Journal of Computers,
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,
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

Note: Figure translations are in progress. See original paper for figures.