Research in Astronomy and Astrophysics, 25:055007 (14pp), 2025 May

https://doi.org/10.1088/1674-4527/adc85b
CSTR: 32081.14.RAA.adc85b

Improving Pulsar Candidate Identification with Grid Group Uniform Sampling

Yi-Ning Song[1,2], Mao-Zheng Chen[1,3], and Zhi-Yong Liu[1,3]
[1] Xinjiang Astronomical Observatory, Chinese Academy of Sciences, Urumqi 830011, China; chen@xao.ac.cn, liuzhy@xao.ac.cn
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] Key Laboratory of Radio Astronomy and Technology (Chinese Academy of Sciences), Beijing 100101, China
Received 2024 September 18; revised 2025 January 10; accepted 2025 January 20; published 2025 May 8

Abstract

Pulsar candidate identification is an indispensable task in pulsar science. Based on the characteristics of imbalanced and diverse pulsar data sets, and the lack of a unified processing framework, we first used dimensionality reduction and visualization to analyze potential deficiencies caused by the incompleteness of current data set extraction methods. We found that the limited use of non-pulsar data may lead to bias in the result, which may limit the generalization ability. Based on the dimensionality reduction results, we propose a Grid Group Uniform Sampling (GGUS) method. This data preprocessing method improves the performance of Random Forest, Support Vector Machine, Convolutional Neural Network, and ResNet50 models on Lyon's features, diagnostic plots, and period-dispersion measure (period-DM) plots in the HTRU1 data set. The average recall increased by approximately 0.5%, precision by nearly 2%, and F_1 score by around 1.2% for all models and in all data sets. In the period-DM plots testing, the high-performance ResNet50 algorithm achieved over 98% F_1 using random sampling. GGUS demonstrated further improvements in this test, enhancing the average F_1 score, precision, and recall by approximately 0.07%, 0.1%, and 0.03%, respectively.

Key words: (stars:) pulsars: general – methods: data analysis – methods: statistical

1. Introduction

In recent years, the rise of artificial intelligence has led to the application of machine learning methods across a wide range of research and industries, aiming to solve frontier problems and drive innovation in both projects and technology (Camastra & Vinciarelli 2015; Zantalis et al. 2019). In the field of astronomy, for example, machine learning has been widely used in pulsar candidate identification, (e.g., Wang et al. 2018). Given the significance of pulsar research in detecting gravitational waves (e.g., Abbott et al. 2017; Xu et al. 2023), navigating in deep space (e.g., Deng et al. 2013), and providing an exceedingly accurate timing system (e.g., Hobbs et al. 2020), discovering new pulsars is also one of major focuses (e.g., Nan et al. 2006; Wang et al. 2023). Identifying new pulsars requires filtering genuine pulsar signals from large numbers of candidates from observational data processing results, making the application of machine learning algorithms to assist in candidate identification.

Different from typical computer vision tasks, pulsar candidate data sets are characterized by class imbalance, with pulsars representing only a very small fraction of the data, while the majority were consisted of background noises and radio frequency interferences (RFI). Special algorithm designs are needed to find pulsar signals and make the algorithms more robust and transferable. Recent advancements in image recognition algorithms have led to notable successes in pulsar candidate identification. For instance, Yin et al. (2022) achieved 100% accuracy on the High Time Resolution Universe (HTRU) survey data set using Generative Adversarial Networks (GANs) and Residual Neural Networks (ResNet). Liu et al. (2024b) employed a semi-supervised method to reduce manual labeling efforts, which achieved over 90% accuracy on both the FAST and HTRU data sets with 100 labeled samples. Liu et al. (2024a) addressed the imbalance in pulsar data by using ResNet, achieving over 97.5% performance across various metrics on both FAST and HTRU data sets. These studies show the challenges of data imbalance and potential labeling issues in pulsar searches.

Consequently, many researchers have explored data augmentation techniques to expand pulsar candidate data sets to counteract data set imbalance problem. The augmentation ratios vary widely, from several times (e.g., Liu et al. 2023) to dozens of times (e.g., Agarwal et al. 2020; Liu et al. 2024a), and even up to 50 times (e.g., Wang et al. 2019). While there is no precise metric to determine when data augmentation may lead to overfitting, extreme levels of augmentation may be hazardous. Excessive data augmentation may result in models learning limited characteristics of the specific data set, resulting in a lack of generalizability (Shorten & Khoshgoftaar 2019). Analyzing existing data sets can give us an understanding of the distribution of pulsars and non-pulsars, as well as the impact of non-pulsar signal selection on algorithmic outcomes.

Table 1
Features Designed by Lyon et al. (2016)

No.	Description	Feature
1	Mean of the integrated profile	P_μ
2	Standard deviation of the integrated profile	P_σ
3	Excess kurtosis of the integrated profile	P_k
4	Skewness of the integrated profile	P_s
5	Mean of the DM-SNR curve	DM_μ
6	Standard deviation of the DM-SNR curve	DM_σ
7	Excess kurtosis of the DM-SNR curve	DM_k
8	Skewness of the DM-SNR curve	DM_s

Moreover, when dealing with large-scale candidate data sets, it is necessary to evaluate whether the training data set is sufficient to ensure the algorithm's robustness and generalization capability.

To help algorithms use the pulsar and non-pulsar data sets comprehensively, we proposed a method called Grid Group Uniform Sampling (GGUS) for data extraction. Section 2 introduces the characteristics of pulsar candidate data set. Section 3 explains the rationale for dimensionality reduction and presents a comparison of the results in the HTRU1 data set. Section 4 proposes and refines the GGUS method for training and testing data extraction. Section 5 presents the experimental results, and Section 6 discusses our results, summarizes our findings and future research directions.

2. Pulsar Dataset

In the field of pulsar candidate classification, data sets like HTRU1 (Morello et al. 2014), HTRU2 (Thornton 2013), and FAST (Wang et al. 2019) are two-class classification data sets with pulsars and non-pulsars. In this paper, we selected the HTRU1 data set, one of the most commonly used data sets, for analysis. The HTRU1 data set comprises a subset of preprocessed data from the HTRU survey, including 1,196 pulsar instances coming from 521 pulsars and their harmonic data, along with 89,996 non-pulsar instances.

Based on Lyon et al. (2016)'s review of pulsar candidate identification and recent studies in the field (e.g., Yin et al. 2022; Liu et al. 2024b), pulsar data sets for machine learning classification can be divided into two types: numerical data based on pulsar features and visual image data sets.

The feature-based numerical data are derived from parameters that have been designed by researchers (e.g., Lee et al. 2013; Morello et al. 2014; Tan et al. 2018). Parameters such as the signal-to-noise ratio (S/N), pulse profile, dispersion measure (DM), pulsar period, and acceleration search are commonly selected as features for analysis. Among the manually designed pulsar features, we selected the features proposed by Lyon et al. (2016) as one of the test data sets in this study. Details of Lyon's features are presented in Table 1.

Pulsar candidate samples in image format provide data details, with each feature value can be learned by algorithms. In this article, we selected the pulsar diagnostic plots and the period-dispersion measure (period-DM) plots for analysis and testing.

The pulsar diagnostic plot consists of four sub-plots. Figure 1 illustrates diagnostic plot examples of a pulsar signal, a typical RFI, and a background noise. As one of the commonly used data set formats, researchers such as Zhu et al. (2014), Wang et al. (2019), Guo et al. (2019) selected and extracted specific features from these diagnostic plots for training and testing in their algorithms.

The period-DM plots, shown in Figure 2, reflect how the S/N of a signal varies with different periods of folding and different dispersion values. In period-DM plots, the most notable distinction between RFI and pulsar signals is that RFI usually does not exhibit significant dispersion, and its S/N often peaks at zero dispersion. In contrast, noise typically shows relatively low S/N values and lacks distinctive periods and dispersion characteristics.

This paper will apply Lyon's features, diagnostic plots, and period-DM plots obtained from HTRU1 for processing and testing.

3. Data Preprocessing and Visualization

Wang et al. (2018) argued that although manually designed features are compact and concise, their reliance on human design may lead to bias. In contrast, image-based data sets, though large and challenging to train, tend to yield more precise results. We also found that, with the increasing computational power, image recognition algorithms are advancing, and most recent approaches are result-oriented (e.g., Lyon et al. 2016; Wang et al. 2019). However, few researchers (e.g., Wang et al. 2019) have conducted in-depth analyses of the intrinsic features of pulsar data.

Hence, the first objective of this paper is to analyze the features of pulsar data sets from a macro perspective. We aim to provide a comprehensive and quantitative understanding of the overall characteristics of pulsar data.

For algorithm selection, dimensionality reduction (Garzon et al. 2022) is chosen as the basis for visualization analysis. Compared to the original data set, dimensionality reduction offers advantages such as improving data set usability, reducing computational overhead, removing noise, and making results easier to interpret and visualize (Garzon et al. 2022). Common dimensionality reduction algorithms include Principal Component Analysis (PCA; Abdi & Williams 2010), Linear Discriminant Analysis (LDA. Tharwat et al. 2017), Local Linear Embedding (Roweis & Saul 2000), Multidimensional Scaling (Torgerson 1952), t-distributed Stochastic Neighbor Embedding (Van der Maaten & Hinton 2008), and

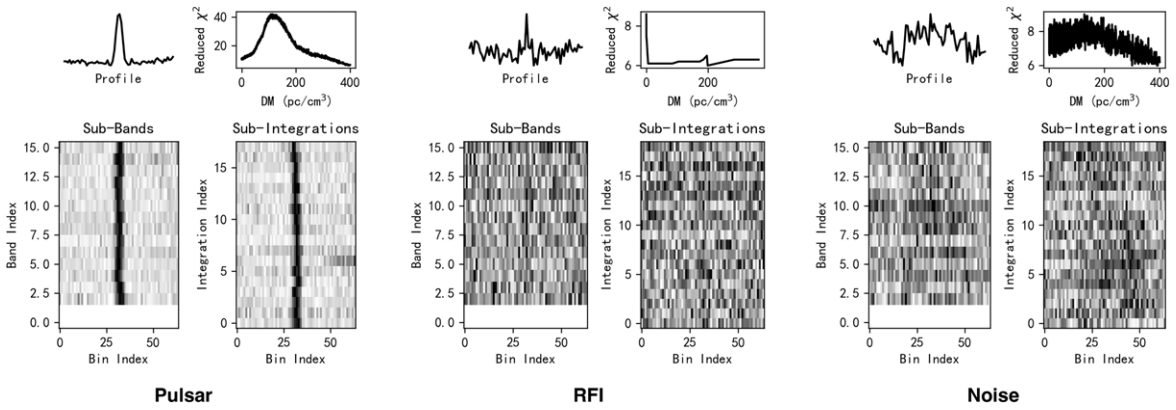

Figure 1. Pulsar diagnostic plot examples with four sub-plots. The plots from left to right are for signals of a pulsar, an RFI, and a background noise. In each plot, the four sub-plots from up to bottom and left to right are Profile, DM-S/N, Sub-Bands, and Sub-Integrations.

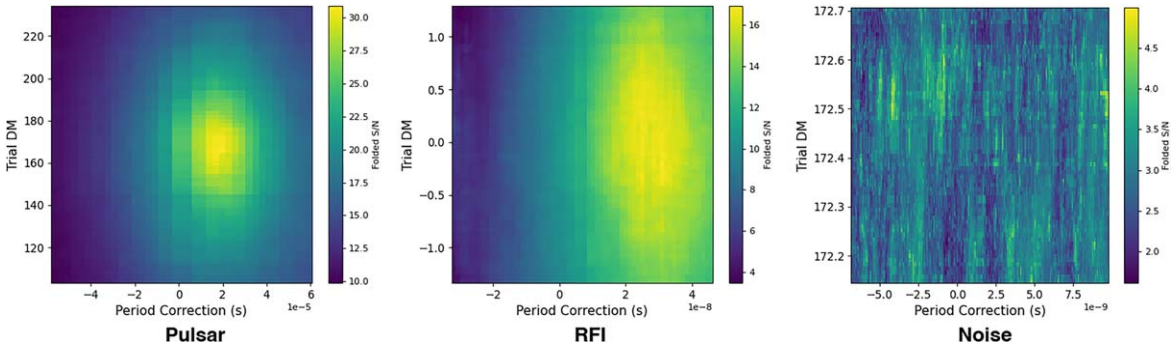

Figure 2. Pulsar period-DM plot examples. The plots from left to right are for signals of a pulsar, an RFI, and a background noise.

Isomap (Tenenbaum et al. 2000), which cover various methods including linear, nonlinear, and manifold learning approaches.

When dealing with large data sets, nonlinear and manifold learning algorithms experience exponential increases in computational complexity as the data set size grows. This phenomenon may lead to the "curse of dimensionality." Therefore, efficient and interpretable linear dimensionality reduction algorithms are preferred. Among linear algorithms, we selected the PCA as the method for dimensionality reduction and visualization. PCA is an unsupervised method, which can reduce data dimensions while preserving essential information, facilitating subsequent visualization and analysis.

3.1. Principal Component Analysis (PCA)

PCA is one of the most well-known linear dimensionality reduction algorithms. It projects high-dimensional data onto a lower-dimensional subspace while retaining most of the variance in the original data. The core idea of PCA is to identify the directions in which the variance of the data is maximized and project the data onto these directions, which are known as principal components (Abdi & Williams 2010).

Let the original data that needs to be reduced be represented as an $m \times n$ matrix $A = \begin{pmatrix} a_{11} & \cdots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \cdots & a_{mn} \end{pmatrix}$, where m represents the dimensions of each data point, and n is the number of data points. The goal of the PCA algorithm is to reduce the m-dimensional data to k-dimensions ($m > k$).

The PCA algorithm (Abdi & Williams 2010) can be divided into the following steps:

1. Perform decentering to make the data set's mean zero:

$$B = A - \bar{A} = \begin{pmatrix} a_{11} - \bar{a}_1 & \cdots & a_{1n} - \bar{a}_1 \\ \vdots & \ddots & \vdots \\ a_{m1} - \bar{a}_m & \cdots & a_{mn} - \bar{a}_m \end{pmatrix} = \begin{pmatrix} a'_{11} & \cdots & a'_{1n} \\ \vdots & \ddots & \vdots \\ a'_{m1} & \cdots & a'_{mn} \end{pmatrix}, \quad (1)$$

where $\bar{a}_i = \frac{1}{n}\sum_{j=1}^{n} a_{ij}$ is the mean of the ith row of the data.

2. Calculate the covariance matrix C:

$$C = \frac{1}{n}BB^T = \begin{pmatrix} \frac{1}{n}\sum_{i=1}^{n} a'^2_{1i} & \cdots & \frac{1}{n}\sum_{i=1}^{n} a'_{1i}a'_{mi} \\ \vdots & \ddots & \vdots \\ \frac{1}{n}\sum_{i=1}^{n} a'_{mi}a'_{1i} & \cdots & \frac{1}{n}\sum_{i=1}^{n} a'^2_{mi} \end{pmatrix}. \quad (2)$$

3. Find the eigenvectors $E = (\boldsymbol{e}_1, \boldsymbol{e}_2,\ldots,\boldsymbol{e}_m)$ and their corresponding eigenvalues $\Lambda = (\lambda_1, \lambda_2, \ldots, \lambda_m)$ of the covariance matrix C, which satisfy the following relation:

$$E^T CE = \Lambda = \begin{pmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_m \end{pmatrix}. \quad (3)$$

4. Select the top k eigenvectors corresponding to the largest k eigenvalues to form a new matrix $P' = (\lambda_1^{'}, \lambda_2^{'},\ldots,\lambda_k^{'})^T$.

5. Project the original data set onto the new space formed by the eigenvector matrix P' to obtain the reduced-dimensional matrix:

$$Y = P'A. \quad (4)$$

PCA is widely applied in machine learning, including for data visualization, feature extraction, and data compression. It is particularly effective for linear data and performs well on high-dimensional data sets. By reducing the dimensionality of the data, PCA helps improve the performance and interpretability of machine learning models (Jolliffe & Cadima 2016).

3.2. Dimensionality Reduction Results

As of now, no researchers have detailed whether using a subset of the non-pulsar data set instead of the entire data set affects the same algorithm's performance. To compare the potential impact of using a partial non-pulsar data set, we compared the dimensionality reduction and visualization results of the two data sets: 1. the pulsar data set (1196 samples) with the first 1196 non-pulsar samples; 2. the entire data set.

The analysis was performed using three types of pulsar data, generated in Section 2. The results of the dimensionality reduction visualizations are shown in Figures 3–5.

3.2.1. Dimensionality Reduction of Lyon's Features

Figure 3 displays the dimensionality reduction results for pulsar data. The results indicate that the overall design of the features is reasonable. The dimensionality reduction results for the small subset of non-pulsar data (left panel) resemble the structure observed with the full data set, suggesting that these features can effectively capture the characteristics of both pulsar and non-pulsar data. Notably, about 10% of the pulsar and non-pulsar data points form the edges of a triangular structure in the reduced-dimension space. This is due to repeated and incomplete dispersion experiments in some of the HTRU1 original data. Incomplete dispersion experiments resulted in missing values for some DM calculations. During visualization, missing values were replaced with zeros for analysis.

3.2.2. Dimensionality Reduction of Pulsar Diagnostic Plots

Figure 4 presents the dimensionality reduction results for pulsar diagnostic plots. Compared to the features designed by Lyon, the dimensionality reduction of image-based pulsar candidate data sets shows significant changes. The left panel exhibits a distinct linear structure, with non-pulsar data clustering in the lower-left corner. The right panel reveals substantial variations, with non-pulsar data forming a vertical bar-like structure around the x-axis range of approximately 0.4–0.6. Meanwhile, pulsar data is concentrated in the central-right area of the plot. This comparison indicates that training with a subset of non-pulsar data, like the first 1196 samples, may introduce biases into the results.

3.2.3. Dimensionality Reduction of Period-DM Plots

Figure 5 shows the dimensionality reduction results for the period-DM plot. Similar to the findings for the pulsar diagnostic plot, the dimensionality reduction results for pulsars with a subset of non-pulsar data differ from those for pulsars with the full non-pulsar data set. In the visualization of the small subset (left panel), pulsar and non-pulsar data are clustered in the lower-left and upper-right regions, respectively, with some overlap in the middle. The pulsar signals can be roughly clustered into two sub-groups, both showing a linear trend from the bottom-left to the upper-right. Compared to the left panel, the right panel shows a feature distribution skewed toward non-pulsar data, with non-pulsar samples dominating the feature space. While pulsar data still clusters in the lower-left corner, it follows a distinct linear trend from up-left to bottom-right.

Overall, for the pulsar candidate data set, particularly for diagnostic and period-DM plots, a reasonable and comprehensive selection of data from the entire data set may help algorithms learn a more extensive range of features distinguishing pulsars from non-pulsars, thereby improving their performance and robustness. Simply selecting a subset of the data set for training and testing, in contrast, may introduce biases.

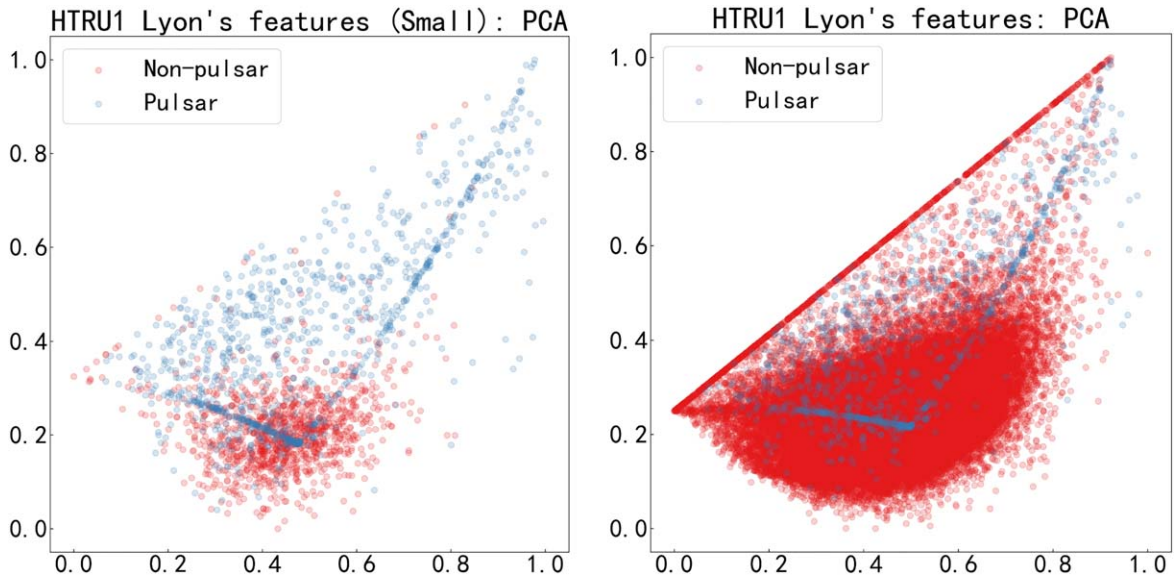

Figure 3. The dimensionality reduction results of the HTRU1 Lyon's features. From left to right, the plots show the pulsars with the first 1196 non-pulsars, and the pulsars with all non-pulsars. The normalized 2D dimensionality reduction results (range 0–1) were displayed on the x and y axes in figures.

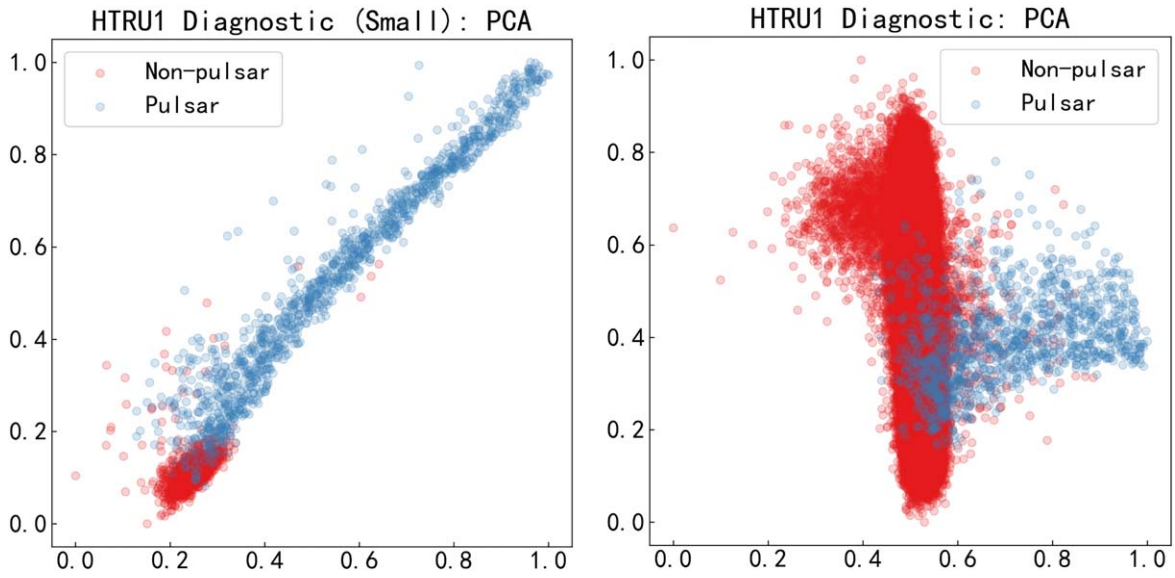

Figure 4. The dimensionality reduction results of the HTRU1 diagnostic plots. From left to right, the plots show the pulsars with the first 1196 non-pulsars, and the whole data set. The results of dimensionality reduction were normalized to a range of 0–1 and plotted on the x and y axes in figures.

4. Grid Group Uniform Sampling Method (GGUS)

The dimensionality reduction results show that data points occupy distinct positions in visualization plots based on their characteristics. This indicates the need to capture a comprehensive range of data types in training set design, which may affect model generalization. To make more comprehensive use of the data set and mitigate the impact of data imbalance, we designed an algorithm named the Grid Group Uniform Sampling (GGUS) method.[4] GGUS ensures that diverse data samples are uniformly selected, thereby enhancing model performance and robustness, as illustrated in Figure 6.

Figure 6 shows the flowchart of the GGUS method. Initially, performing data set extraction to obtain Lyon's features, diagnostic plots, and period-DM plots. PCA is subsequently

[4] https://github.com/Kassisong/GGUS-Grid-Group-Uniform-Sampling

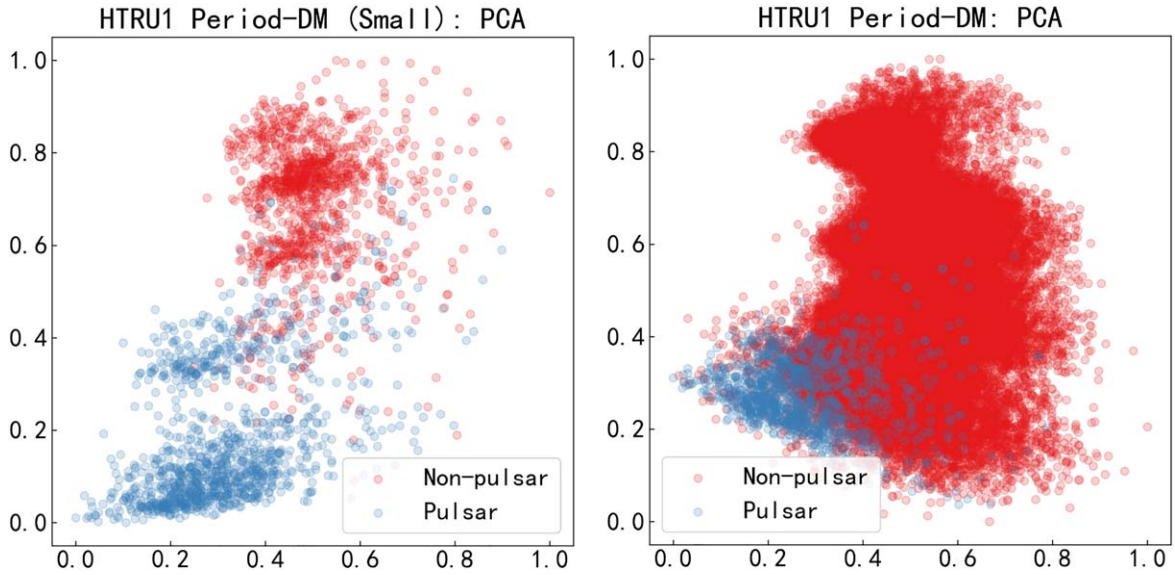

Figure 5. The dimensionality reduction results of the HTRU1 period-DM plots. The left panel shows the result for pulsars with the same number of non-pulsars, and the right panel shows the result for the entire data set. The results of dimensionality reduction were normalized to a range of 0–1 and plotted in figures.

applied for dimensionality reduction. The "Remove Outlier Data" step ensures the elimination of anomalous data points to enhance the reliability of downstream analyses. We selected four machine learning models: Random Forest (RF), Support Vector Machine (SVM), Convolutional Neural Network (CNN), and ResNet50, as described in Section 4.2.2. These algorithms are selected for testing, and evaluating their respective performances in different data set formats. The arrows in the flowchart represent preprocessing to classification and optimization step.

4.1. The Grid Group Uniform Sampling Algorithm

GGUS can divide data with different characteristics into various groups based on dimensionality reduction results, and then uniformly sample data from each group according to the required amount of data. The grouping strategy is illustrated in Figure 7.

Figure 7 exhibits a 5 × 5 grid partitioning case. We also assign numbers to each partition to better illustrate the GGUS method. The detail of the grid number selection is discussed in Section 4.3. From the grouping results, we can see that some groups, such as 0-4, contain no data. Thus, GGUS ignores these no data groups during the data extraction process. Conversely, groups with substantial amounts of pulsar and non-pulsar data, such as 1–0 and 2–3, have data randomly sampled in proportion for training and validation sets. For groups with limited pulsar and non-pulsar data (e.g., 0–2 and 4–4) GGUS ensures proportional data extraction to include samples with similar characteristics in the training set. Details of GGUS are provided in the pseudocode of Algorithm 1 and Algorithm 2 in Appendix A.

4.2. Experimental Design

4.2.1. Dataset Partitioning

In the data set partitioning process, the following conditions were considered:

1. 80% of the pulsar samples were selected for the training set, while 20% of the data was allocated to the validation set.
2. To mimic the phenomenon of imbalance data set in survey, the validation set was constructed by extracting 20 times the number of pulsar samples from all non-pulsar samples. To evaluate the performance and robustness of the algorithm, the training set was created with varying ratios of non-pulsar to pulsar data, ranging from 1:1 to 10:1. Specific quantities for training and testing data are detailed in Table B1 in Appendix B.

4.2.2. Training Algorithms and Evaluation Metrics

To evaluate the improvements achieved by GGUS, we selected four representative models for testing and analysis: RF (Breiman 2001), SVM (Noble 2006), CNN (LeCun et al. 1998), and Residual Neural Network (ResNet) (He et al. 2016). RF is an ensemble learning algorithm widely applied in pulsar candidate identification (e.g., Lyon et al. 2016; Wang et al. 2019c). SVM, as a classical maximum-margin classification algorithm, is also frequently used in related

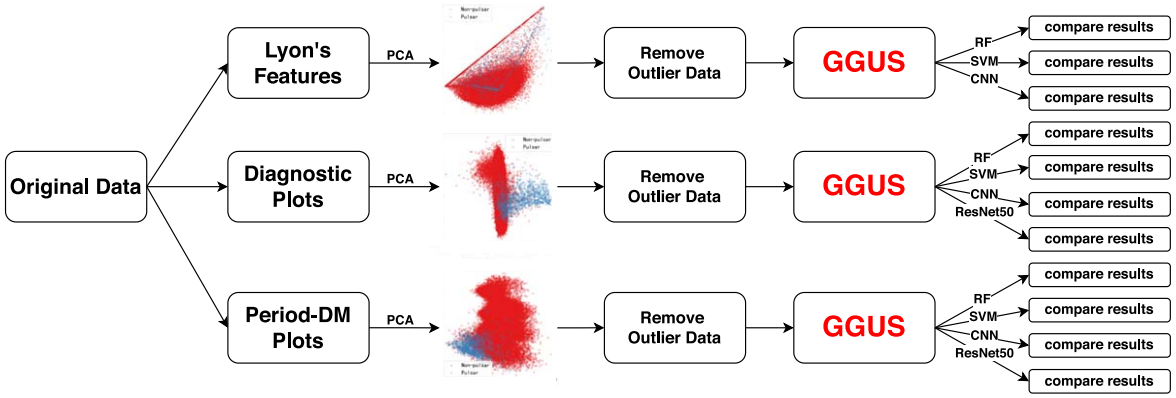

Figure 6. The flowchart of the GGUS method. "Remove Outlier Data" refers to the process of eliminating anomalous data points identified following dimensionality reduction. RF, SVM, CNN, and ResNet50 represent four machine learning algorithms: Random Forest, Support Vector Machine, Convolutional Neural Network, and ResNet50, respectively.

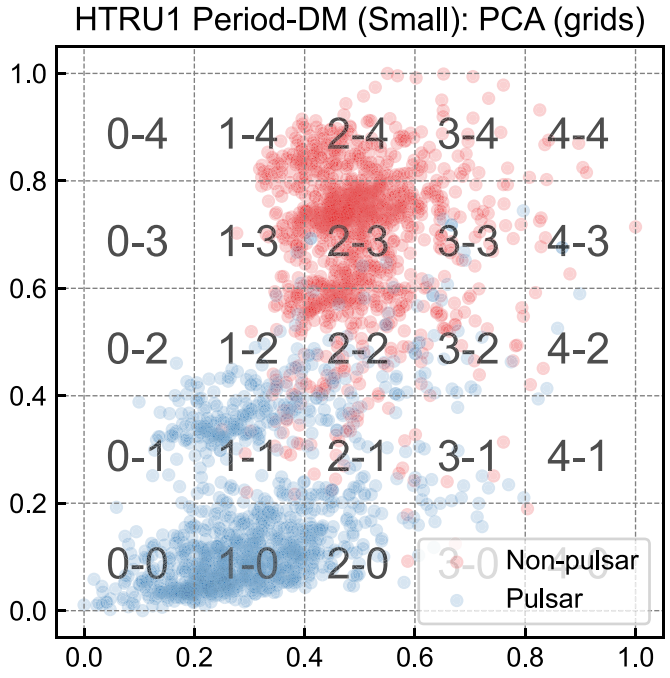

Figure 7. Diagram of GGUS's group strategy. The dimensionality reduction figure in this example is based on the left panel of Figure 5. The x and y axis respectively represent the normalized dimensionality reduction range of the data. The numbers in the diagram represent the "names" of individual grids resulting from the partitioning process.

studies (e.g., Guo et al. 2017; Devine et al. 2016). CNN serves as the foundational architecture for convolution-based deep learning networks. If GGUS proves effective with CNN, it suggests that other convolution-based deep learning algorithms may similarly benefit from performance improvements. ResNet is a typical representative of deep networks, famous for its robustness, which is commonly used in pulsar candidate identification (e.g., Xiao-Fei et al. 2021; Liu et al. 2024a). We selected ResNet50 for testing, which is renowned for its strong generalization and robustness. Details of algorithms can be found in Appendix C.

For algorithm evaluation, we used recall, precision, and F_1 score (Lyon et al. 2016) as the metrics to calculate the performance of RF, SVM, CNN, and ResNet50 in pulsar candidate identification.

4.2.3. Training and Testing Methods

For each experiment, we randomly extracted the required training and testing data from the HTRU1 data set as described in Section 4.2.1, and used the selected algorithm to perform training and testing. Afterward, we applied the GGUS method to sample the data set, followed by training and testing using the same algorithms. We repeat every experiment 10 times under the same conditions obtaining the average performance and error margins. The results were then used to generate comparative performance charts, which are shown in Figures 9, and 11–13.

To control variables, we employ a uniform 5×5 grid in the training and testing process, dividing the data into 25 groups for group sampling.

4.3. Algorithm Optimization

Analyzing the results after dimension reduction, we can find that more than 95% non-pulsar data points on the right panel of Figure 4 fall within a narrow range of x-axis (0.4–0.6), and those data points' x-axis value small than 0.4 potentially being considered outliers. We compare the results before and after removing outliers in Figure 8. The left panel shows the dimensionality reduction result from the right panel of

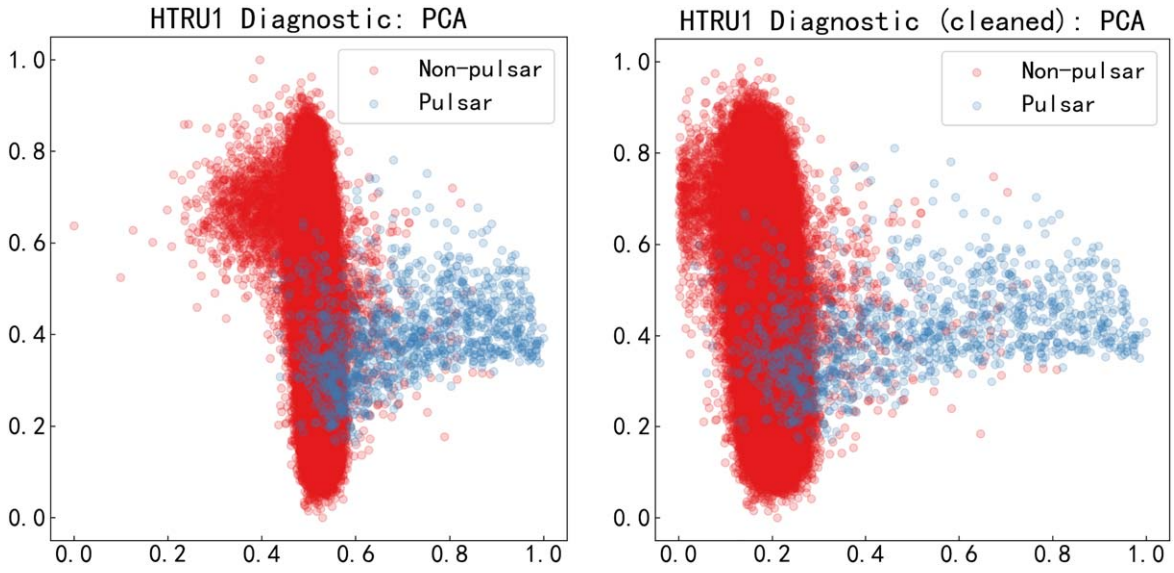

Figure 8. Comparison result of dimensionality reduction of diagnostic plots before and after removing outliers. Plots from left to right show the results before and after removing outliers, respectively. The x and y axes respectively represent the values of dimensionality reduction on two principal components after feature normalization, which makes the subsequent grouping step more convenient.

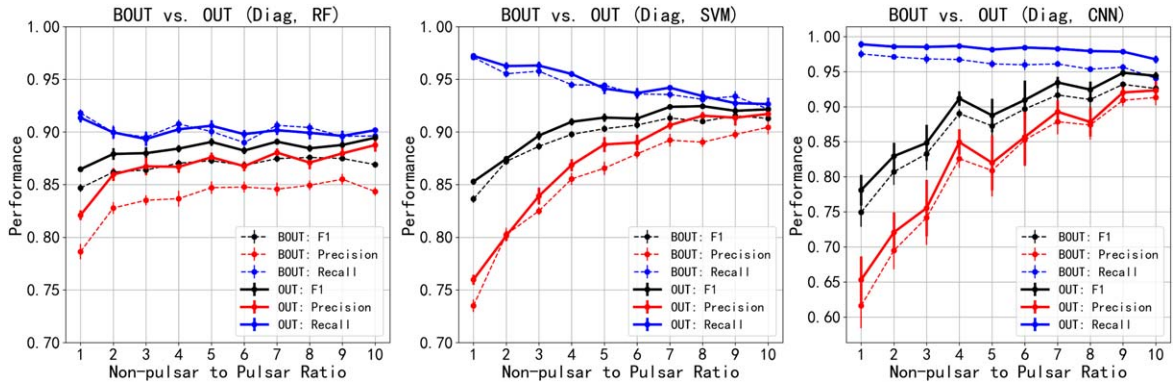

Figure 9. Results before and after removing outliers from the pulsar diagnostic plots. In each sub-figure titles, "BOUT" refers to the results before outlier removal, and "OUT" refers to the results after outlier removal. "Diag" indicates diagnostic plots, followed by the tested algorithm names.

Figure 4, while the right panel shows the result after removing outliers.

Comparing the results before and after removing outliers in Figure 8, it can be observed that the distribution of the dimensionality reduction result after removing outliers is relatively more uniform. To evaluate whether the outlier removal affects the algorithm's performance, we tested the results before and after outlier removal using GGUS (Figure 9).

From the RF testing results in the left panel of Figure 9, we can see that although the recall at some ratios decreased after outlier removal, the average recall changed by less than 0.1%. However, both the precision and F_1 score showed a noticeable improvement after outlier removal, with an increase of 3.0% and 1.6%, respectively. In the SVM method in the middle panel, all metrics experience a rise after outlier removal, with an average increase of 0.3% in recall, 1.5% in precision, and 1.0% in F_1. Similarly, in the CNN method (right panel), all performance metrics exhibited improvements after removing outliers (recall: 2.1%, precision: 1.6%, and F_1: 1.9%). Based on these analyses, it can be concluded that dimensionality reduction followed by outlier removal can enhance algorithm performance.

The performance of the algorithms may also be affected by the choice of grid division numbers. To examine the relationship between the number of grid divisions (ranging from 2×2 to 9×9) and classification effectiveness, we tested various grid

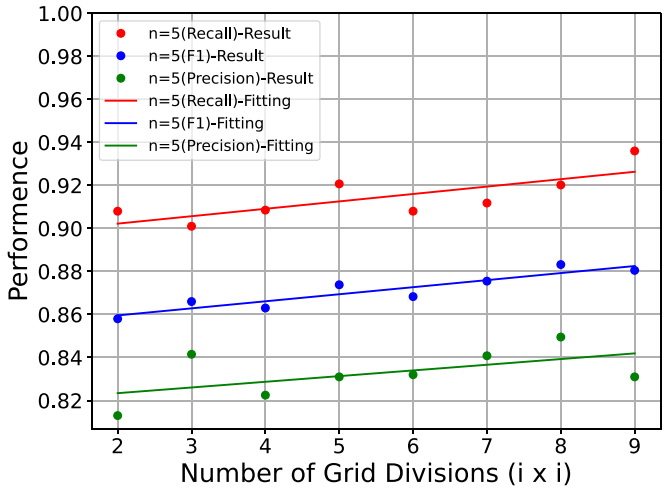

Figure 10. Relationship between the number of grid divisions and algorithm performance. The points in the figure represent the result of different grid division strategies, while the lines indicate the linear fitting result.

division strategies. Figure 10 illustrates the testing results obtained using RF on the period-DM plots. The ratio of non-pulsar to pulsar data in the training set was 5:1.

Linear fitting results shown in Figure 10 indicate that as the number of grid divisions increases, the algorithm's performance exhibits a slow, fluctuating increase. This fluctuation is primarily associated with the robustness of the algorithm and the random selection of the data set. In practical applications, we should balance the computing cost and performance. In practical applications, the strategies for grid division may be influenced by factors such as data set characteristics, dimensionality reduction techniques, and the choice of algorithms. Hence, the grid division strategy could be flexibly adjusted based on the structure of the data set.

In the following experiments, we selected a 5×5 grid division as an example, primarily to demonstrate the GGUS's applicability and generality. However, in actual algorithm design, the grid division strategy should fit with the specific application scenario.

5. Data Analysis Results

In this section, we evaluated the effectiveness of the GGUS method by using the data sets, algorithms, and test methods mentioned in Sections 2, 4.2.2, and 4.2.3. For comparison, a control group is established using a "random sampling from all data" approach to evaluate the improvement of using GGUS. As shown by the results in Section 3.2 and comparison experiments, sequential extraction performs worse than random sampling; therefore, we selected global random sampling as the control group to demonstrate the performance improvements achieved by GGUS.

5.1. Test Results of Lyon's Features

On the one hand, tabular data like Lyon's features are well-suited for traditional machine learning algorithms (Zhong et al. 2016). On the other hand, the standard ResNet50 model is designed for image data and lacks the architecture to handle tabular data directly. Therefore, we did not test ResNet50 on Lyon's features. Instead, we used only RF, SVM, and CNN in the comparative experiments.

For the RF algorithm, the average recall rate for different ratios increased by 0.13% (left panel of Figure 11). The GGUS's precision and F_1 scores are higher than those from random sampling, with an average improvement of 1.6% and 1.0%, respectively. For SVM, GGUS results in an average recall improvement of 0.5% over random sampling. Precision and F_1 scores also improve considerably, with average increases of 3.2% and 1.9%, respectively. Certain ratios show notable improvements. For instance, when $n = 5$, F_1, precision, and recall of GGUS increased by 2.9%, 4.2%, and 1.4%, respectively. In CNN, GGUS demonstrates a slight improvement in recall compared to random sampling, with an average increase of 0.2%. Precision and F_1 scores show more significant improvements across various training set ratios, with increases ranging from 1% to 8%, and average improvements of 3.8% and 2.7%, respectively.

5.2. Test Results of Diagnostic Plots

Figure 12 shows the results of GGUS comparing randomly sampled data used on the data set of diagnostic plots.

For the RF algorithm, GGUS improves the F_1 score on average by 2.3% compared to random sampling across different ratios. The precision score increased by 4.4%, with the greatest improvement of 5.7% at $n = 8$. However, the recall rate showed no obvious improvement, with average differences less than 0.1%. In SVM, GGUS shows slightly lower recall compared to random sampling, with a 0.1% average decrease. However, precision and F_1 scores still show improvements of 1.2% and 0.6%, respectively. In CNN testing, GGUS improved various performance metrics compared to random sampling. The F_1 score, precision, and recall increased on average by 1.3%, 1.1%, and 1.4%, respectively. Testing with ResNet50 shows that GGUS improves the algorithm performance, with the most notable improvement in recall, averaging 2.9%. As the ratio of imbalanced samples increases, recall performance improves further, with a 4.6% increase at $n = 10$. Precision also shows modest improvements across ratios, with an average increase of 1.1%.

5.3. Test Results of Period-DM Plots

Overall, the results shown in Figure 13 demonstrate that performance improvement with the period-DM plots is slightly less pronounced than with the previous data sets but still shows enhancements across all evaluation metrics.

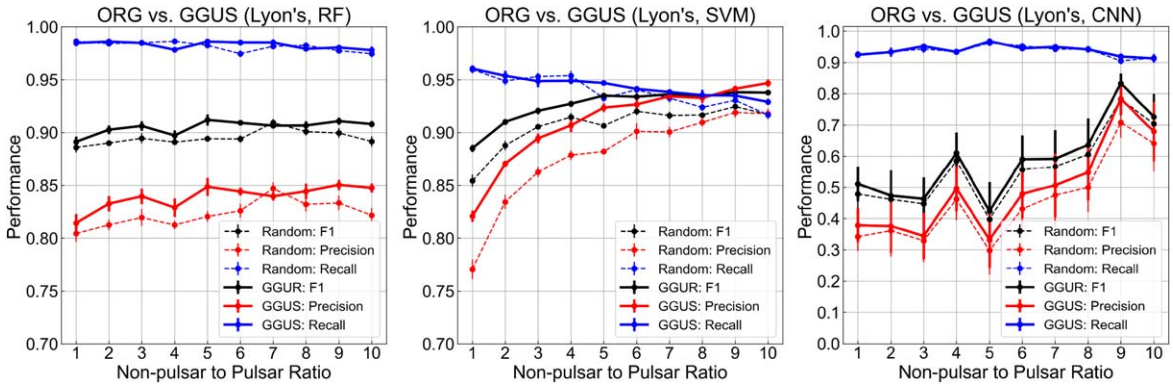

Figure 11. Performance comparison of random sampling and GGUS methods on the pulsar Lyon's features data set. In the sub-figure titles, "ORG" denotes random sampling, while "Lyon's" indicates Lyon's features, followed by the tested algorithm names. The dashed line represents the test results of random sampling, while the bold solid line represents the test results of GGUS. Error bars in figures represent the standard deviation for each group test.

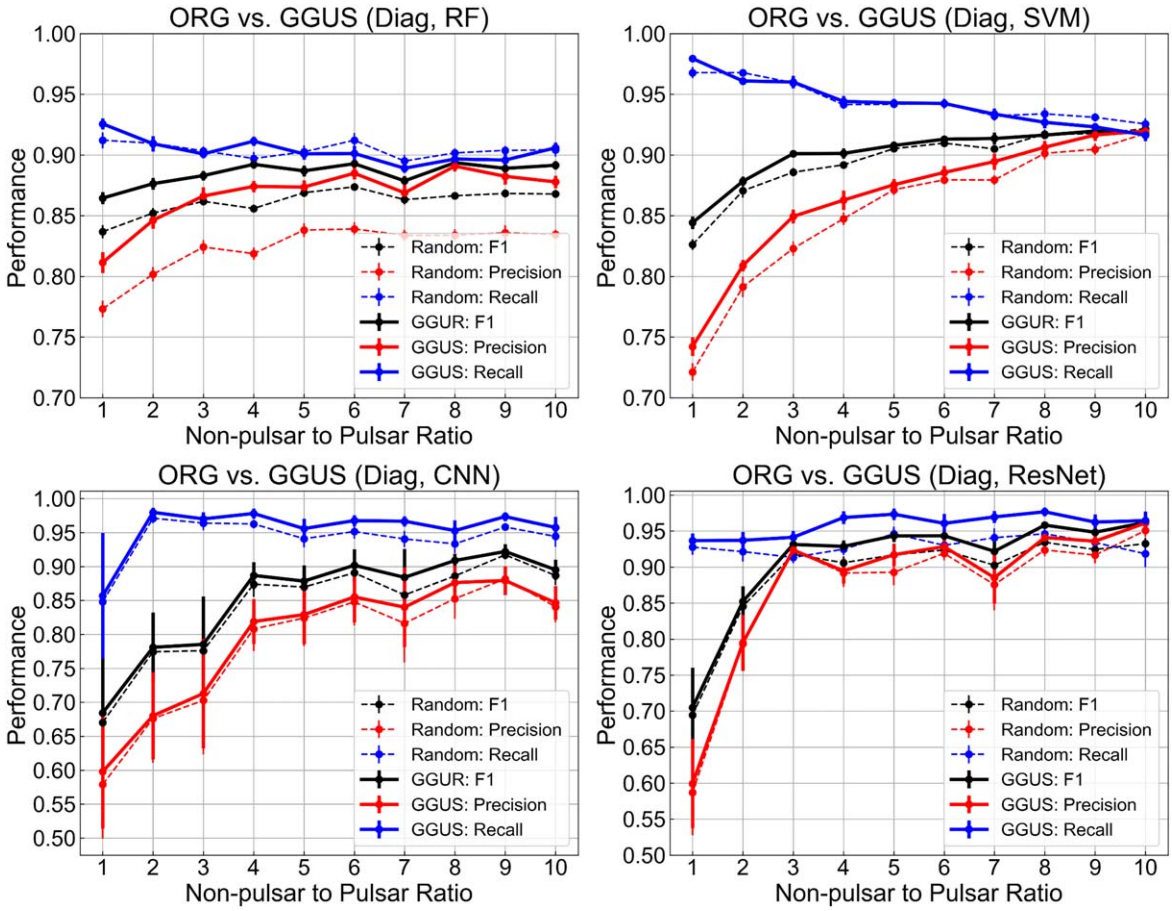

Figure 12. Performance comparison of random sampling and GGUS across algorithms on the pulsar diagnostic plots data set. In the sub-figure titles, "ORG" denotes random sampling, while "Diag" indicates diagnostic plots, followed by the tested algorithm names. The dashed line and solid line respectively represent the test results of random sampling and the test results of GGUS. Error bars in plots represent the standard deviation within each test group across different training ratios.

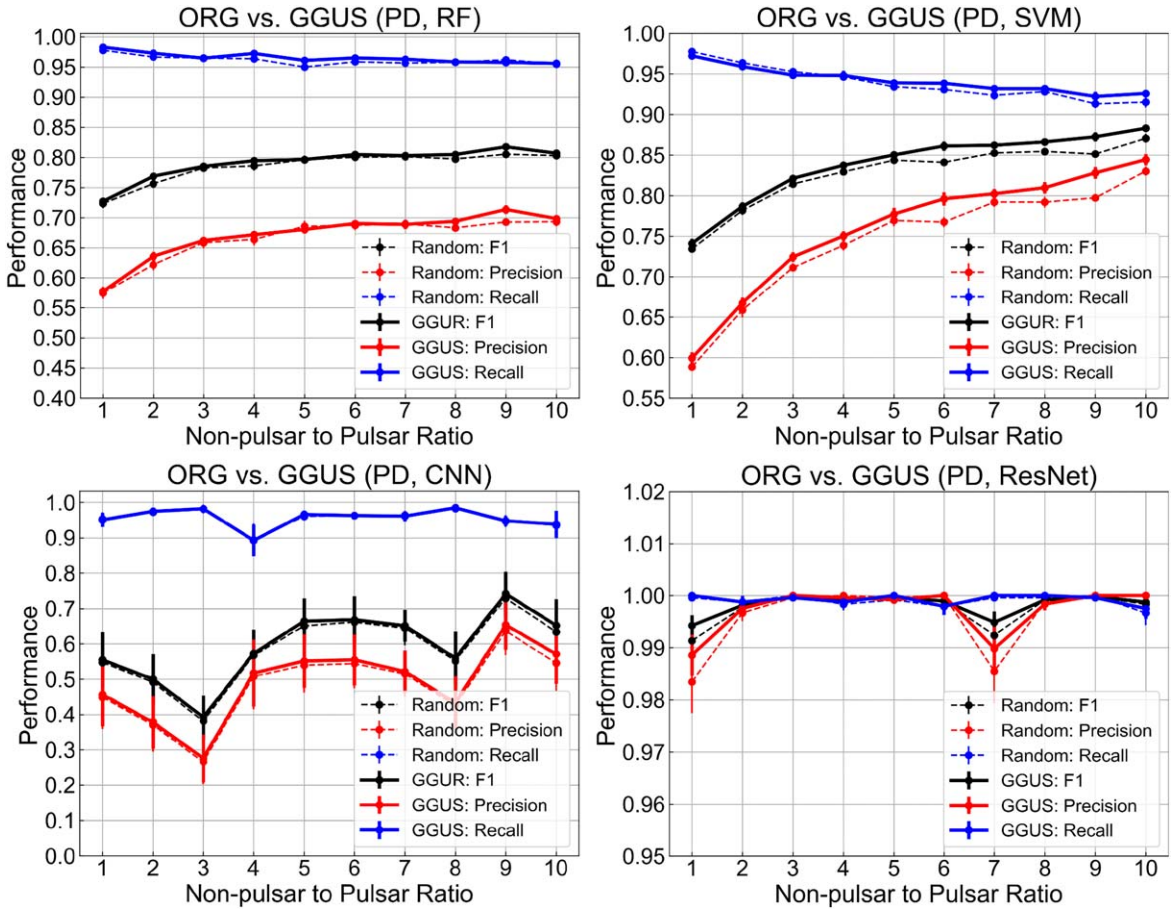

Figure 13. Performance comparison of random sampling and GGUS across algorithms on the pulsar period-DM plots data set. In the sub-figure titles, "ORG" denotes random sampling, while "PD" indicates period-DM plots, followed by the tested algorithm names. The dashed line represents the random sampling test results, and the bold solid line represents the GGUS results. The plots use error bars to show the standard deviation of performance across different training ratios for each test group.

For RF testing, recall rates improved most notably at $n = 4$, 5, 6, and an average increase of 0.4% across all ratios. Precision showed the most significant increase at $n = 9$, with a 2.1% improvement, though there was a slight decrease in recall from 96.2% to 95.8%. On average, GGUS-extracted data improved both precision and F_1 scores by 0.6%. For SVM, GGUS increased precision and F_1 scores across all ratios. Average improvements include F_1 by 1.1%, precision by 1.5%, and recall by 0.31%. For CNN, on average, GGUS improved the F_1 score by 1.0%, precision by 1.4%, and recall by 0.3%, respectively. Although ResNet50 achieved over 98% performance on the validation set, GGUS still shows overall performance benefits. For instance, at $n = 1$, GGUS improved the average recall rate to 100% from almost 100% (99.95% to 100%), with precision increasing by 0.5%. At $n = 7$, GGUS maintained a 100% recall rate with a 0.4% increase in precision. Over all, GGUS improved the F_1 scores of the ResNet50 algorithm by 0% to 0.2%.

6. Discussion and Conclusion

This section discusses results, limitations, and future directions, followed by a summary of our work.

6.1. Discussion

Although GGUS improved the average F_1 score from about 0.1% to 2.9% in RF, SVM, CNN, and ResNet50 for pulsar candidate identification, its performance on semi-supervised and unsupervised algorithms remains untested and could be examined in future work. In addition, practical applications should dynamically adjust grid partitioning as needed. For instance, the bar-like structures depicted in Figure 8 may use a one-dimensional horizontal data division method. This approach can simplify algorithm's complexity and may still yield performance improvements. There are numerous possibilities for further exploration regarding grid partitioning patterns, and we aspire to conduct a more detailed investigation

in the future, aiming to derive quantifiable relationships, including relevant computational formulas.

Returning to the results obtained using the basic version of the GGUS method, an analysis of the test results reveals that the recall across various algorithms is generally higher than the precision, primarily due to the larger number of non-pulsar samples compared to pulsar samples in the test data sets.

GGUS shows notable improvements across various data sets for the RF algorithm, particularly in the diagnostic plot data set. For the SVM algorithm, recall and precision converge as the pulsar to non-pulsar ratio increases. It is a probability because of the model's lack of weight adjustments for minority samples. While GGUS provides modest recall improvements for SVM, it enhances precision and maintains recall comparable to random sampling, resulting in an overall improvement in the F_1 score. CNN's performance remains relatively modest, mainly due to the algorithm's simple architecture. However, GGUS boosts recall, precision, and F_1 scores across data sets for the CNN algorithm. For ResNet50, GGUS improves recall by nearly 3% on average in the diagnostic plot data set, with a nearly 5% improvement at $n = 10$ as sample imbalance increases. In the period-DM plot data set, random sampling with ResNet50 has achieved an almost 100% F_1 score; however, GGUS still shows overall benefits, demonstrating its generalizability and stability in different scenarios.

In the results of 11 tests presented in this paper, except for a slight decrease in recall for one algorithm (SVM on diagnostic plots), the average performance of other algorithms improved from about 0.1% to 5%. Overall, GGUS achieved an improvement in precision while maintaining or increasing recall. This demonstrates that GGUS can help the algorithms learn the characteristics of the data set more accurately and comprehensively, and enhances overall classification performance. Test groups where the average precision increased by more than 3% include SVM and CNN on Lyon's features data set and RF on the diagnostic plots data set. For other algorithms, the average precision improvement ranged between 0.1% and 2%. Overall, precision in all tests showed an average performance increase of nearly 2%. As for recall, the initial performance under random sampling was relatively high compared to precision. Therefore, the improvement in recall with GGUS was less obvious but still enhanced recall for most algorithms by 0.1%–1%. The most significant improvement was observed in ResNet50 on the diagnostic plots data set, with an average increase of 2.9%. In all tests, recall improved by approximately 0.5% on average. F1 combines recall and precision to evaluate overall algorithm performance and showed an average improvement of about 1.2% across all tests with GGUS.

Additionally, most machine learning algorithms used on images data set are data-driven, which performance is highly reliant on data set quality. Accordingly, a general, high-quality, standardized data set is the key of robust algorithm. However, due to differences in telescope parameters and environment factors, such as sampling rates, sub-band numbers, frequency ranges, and variations in the RFI environment, pulsar surveys produce data sets with varying characteristics. Consequently, there is no standardized procedure for processing pulsar candidate data sets, which limits their general applicability. We hope to collaborate with pulsar survey teams in further research of the current data set, aiming to design a standardized and universally applicable data set to support pulsar candidate selection across various projects.

6.2. Summary

In this paper, we first analyzed the data set distribution through dimensionality reduction and visualization. From the results, we concluded that non-pulsar data plays a significant role in the training set extraction process. Then we created a data preprocessing method, GGUS, to enhance the algorithm's generalization capability. Through comparative experiments, the necessity of removing outliers from the non-pulsar data was verified. We also found that increasing the number of grid divisions can enhance the effectiveness of GGUS. GGUS can extract well-balanced training sets for various machine learning algorithms to learn the features of pulsar candidates. Our results show that the GGUS method can consistently enhance the performance of various algorithms across different data sets, demonstrating generalizability and adaptability. Due to the impact of imbalanced data sets, the GGUS method shows the most significant improvement in precision, with a relatively smaller increase in recall. Specifically, the average recall increases by approximately 0.5%, precision by about 2%, and the F1 score by around 1.2%. These results emphasize the role of optimized data extraction methods in improving the performance of different algorithms for pulsar candidate identification.

Acknowledgments

This work was supported by the National Key Research and Development Program of China under grant No. 2018YFA0404603. The research work is also partly supported by the Operation, Maintenance and Upgrading Fund for Astronomical Telescopes and Facility Instruments, budgeted from the Ministry of Finance of China (MOF) and administered by the Chinese Academy of Sciences (CAS).

Appendix A
Pseudocode

Algorithm 1. GGUS Grouping

Ensure:
DeData: Dimensionality-Reduced and normalized Data, $DeData \in [0-1]$
N: Number of groups.
Require:
GridClass: Grouping of all data points
$Step = \frac{1}{N}$
for i=0 **to** DeData.length **do**
  $GridX = DeData[i][0]/step$ {Extract the first dimension data}
  $GridY = DeData[i][1]/step$ {Extract the second dimension data}
  $GridClass[i] = (GridX, GridY)$
end for

Algorithm 2. GGUS Data Extraction Algorithm

Ensure:
Data: Original training data
GridClass: Data groupings
n: Number of data samples to extract, $n \leqslant Data.length$
per: Proportion of the data set to be used for training
Require:
TrainData: Training data set uniformly extracted based on groupings
ValData: validate data set uniformly extracted based on groupings
{In the process of selecting pulsar candidate data, it is necessary to design the extraction ratio of pulsar to non-pulsar data.}
NumData = data.length
for i=0 **to** GridClass.max **do**
for j=0 **to** GridClass.max **do**
  GroupIndices = indices contain [*i*, *j*] in *GridClass*
  $NumSamples = GroupIndices.length$
  $TrainSize = (n * per) * (NumSamples/n)$
  $ValSize = n * (1 - per) * (NumSamples/n)$
  TrainIndices ← select *TrainSize* random samples from *GroupIndices* as training indices
  ValIndices ← select *ValSize* random samples from remaining *GroupIndices* as validation indices
end for
end for
  $TrainData = Data[TrainIndices]$
  $ValData = Data[ValIndices]$

Appendix B
Dataset Parameter

Table B1
Number of Samples for Algorithm Testing

Assumed Ratio of P[a] to NP[b] Samples	Number of P versus NP in the Training Set	Number of P versus NP in the Testing Set
1:1	957:957	239: 4782
1:2	956:1914	239: 4782
1:3	956:2869	239: 4782
1:4	956:3826	239: 4782
1:5	956:4783	239: 4782
1:6	956:5742	239: 4782
1:7	956:6698	239: 4782
1:8	956:7854	239: 4782
1:9	956:8611	239: 4782
1:10	956:9568	239: 4782

Notes.
[a] P: Pulsar.
[b] NP: Non-pulsar.

Appendix C
Details of Algorithms

For the RF classifier, 100 trees were used, each with a maximum depth of 5. To address the imbalance in the data set, class weights for pulsar and non-pulsar samples were calculated during classification to enable the model to handle the data set's imbalance effectively.

The SVM classifier employed a simple linear kernel function for classification, with a penalty parameter set to 1.0 and a polynomial kernel function of degree.

We employed a simple CNN structure, consisting of two convolutional layers with ReLU activation, followed by two pooling layers and two Dropout layers. A Sigmoid function is applied after the fully connected layer. The model is compiled using the RMSprop optimizer and binary cross-entropy loss function. Specific model parameters are provided in Table C1.

Table C1
CNN Network Architecture

Layer (Type)	Output Shape	Parameter
Conv2D(3,3)	(None, 62, 62, 32)	320
MaxPooling2D(2,2)	(None, 31, 31, 32)	0
Dropout(0.25)	(None, 31, 31, 32)	0
Conv2D(3,3)	(None, 29, 29, 64)	18496
MaxPooling2D(2,2)	(None, 14, 14, 64)	0
Dropout(0.25)	(None, 14, 14, 64)	0
Flatten	(None, 12544)	0
Dense (128)	(None, 128)	1605760
Dense(1)	(None, 1)	129

Total params: 1624705 (6.20 MB)
Trainable params: 1624705 (6.20 MB)
Non-trainable params: 0 (0.00 Byte)

High-performance algorithms like ResNet50 are easy to overfit with small data input sizes. To reduce this effect in the ResNet50 structure, we use a Flatten layer to reshape the data and a Dropout layer (Dropout = 0.5). Next, a Dense layer with a Sigmoid activation function was used for classification. The model was compiled with the Adam optimizer (Learning Rate = 1e-4) and binary cross-entropy loss. The batch size was 32. To further reduce overfitting, the number of training epochs was limited to 3 (simply for comparative testing).

ORCID iDs

Yi-Ning Song https://orcid.org/0000-0002-1274-4766

References

Abbott, B. P., Abbott, R., Abbott, T., et al. 2017, PhRvL, 119, 161101
Abdi, H., & Williams, L. J. 2010, WIREs Comp. Stat., 2, 433
Agarwal, D., Aggarwal, K., Burke-Spolaor, S., Lorimer, D. R., & Garver-Daniels, N. 2020, MNRAS, 497, 1661
Breiman, L. 2001, Machine Learning, 45, 5
Camastra, F., & Vinciarelli, A. 2015, Machine Learning for Audio, Image and Video Analysis: Theory and Applications (Berlin: Springer)
Deng, X., Hobbs, G., You, X., et al. 2013, AdSpR, 52, 1602
Devine, T. R., Goseva-Popstojanova, K., & McLaughlin, M. 2016, MNRAS, 459, 1519
Garzon, M., Yang, C.-C., Venugopal, D., et al. 2022, Dimensionality Reduction in Data Science, Vol. 766 (Berlin: Springer)
Guo, P., Duan, F., Wang, P., et al. 2017, arXiv:1711.10339
Guo, P., Duan, F., Wang, P., et al. 2019, MNRAS, 490, 5424
He, K., Zhang, X., Ren, S., & Sun, J. 2016, in Proc. IEEE Conf. on Computer Vision and Pattern Recognition (Las Vegas, NV: IEEE), 770
Hobbs, G., Guo, L., Caballero, R., et al. 2020, MNRAS, 491, 5951
Jolliffe, I. T., & Cadima, J. 2016, RSPTA, 374, 20150202
LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, IEEEP, 86, 2278
Lee, K., Stovall, K., Jenet, F., et al. 2013, MNRAS, 433, 688
Liu, Y., Jin, J., & Zhao, H. 2024a, NewA, 106, 102125
Liu, Y., Jin, J., Zhao, H., He, X., & Guo, Y. 2023, ApJ, 954, 86
Liu, Y., Jin, J., Zhao, H., & Wang, Z. 2024b, ApJ, 967, 155
Lyon, R. J., Stappers, B., Cooper, S., Brooke, J. M., & Knowles, J. D. 2016, MNRAS, 459, 1104
Morello, V., Barr, E., Bailes, M., et al. 2014, MNRAS, 443, 1651
Nan, R.-D., Wang, Q.-M., Zhu, L.-C., et al. 2006, ChJAA, 6, 304
Noble, W. S. 2006, NatBi, 24, 1565
Roweis, S. T., & Saul, L. K. 2000, Sci, 290, 2323
Shorten, C., & Khoshgoftaar, T. M. 2019, Journal of Big Data, 6, 1
Tan, C. M., Lyon, R., Stappers, B., et al. 2018, MNRAS, 474, 4571
Tenenbaum, J. B., Silva, V. D., & Langford, J. C. 2000, Sci, 290, 2319
Tharwat, A., Gaber, T., Ibrahim, A., & Hassanien, A. E. 2017, AI Communications, 30, 169
Thornton, D. 2013, The High Time Resolution Radio Sky, PhD thesis, Univ. Manchester
Torgerson, W. S. 1952, Psychometrika, 17, 401
Van der Maaten, L., & Hinton, G. 2008, Journal of Machine Learning Research, 9, 2579
Wang, H., Zhu, W., Guo, P., et al. 2019, SCPMA, 62, 1
Wang, N., Xu, Q., Ma, J., et al. 2023, SCPMA, 66, 289512
Wang, Y., Pan, Z., Zheng, J., Qian, L., & Li, M. 2019c, Ap&SS, 364, 1
Wang, Y., Zheng, J., Pan, Z., & Mingtao, L. 2018, Journal of Deep Space Exploration, 5, 203
Wang, Y.-C., Li, M.-T., Pan, Z.-C., & Zheng, J.-H. 2019, RAA, 19, 133
Xiao-Fei, L., Bao-Qiang, L., Tao, A., Zhi-Jun, X., & Zhong-Li, Z. 2021, ChA&A, 45, 364
Xu, H., Chen, S., Guo, Y., et al. 2023, RAA, 23, 075024
Yin, Q., Li, Y., Li, J., Zheng, X., & Guo, P. 2022, ApJS, 264, 2
Zantalis, F., Koulouras, G., Karabetsos, S., & Kandris, D. 2019, Future Internet, 11, 94
Zhong, G., Wang, L.-N., Ling, X., & Dong, J. 2016, The Journal of Finance and Data Science, 2, 265
Zhu, W., Berndsen, A., Madsen, E., et al. 2014, ApJ, 781, 117