USING MACHINE LEARNING TO PREDICT A PULSAR STAR

SridharaRao1, Prashanth Yarram2, Sai Vamshi3, Jahnavi4

[1sridhararao.it@jbiet.edu.in](mailto:1sridhararao.it@jbiet.edu.in), [2prashanthyarram@gmail.com](mailto:2prashanthyarram@gmail.com), [3saivamshi519@gmail.com](mailto:3saivamshi519@gmail.com)

,[4dlakshmijahnavi@gmail.com](mailto:4dlakshmijahnavi@gmail.com)

Department of Information Technology, JB Institute of Engineering and Technology, Hyderabad.

Abstract— **A highly magnetised revolving compact star known as a pulsar produces radiation beams from its magnetic poles. The study of pulsar stars has several useful applications in the astronomy sciences. The observation of pulsars in a binary neutron star system can be used to indirectly prove applications like the existence of gravitational radiation. As a result, identifying pulsars is essential for research into gravitational waves and general relativity. The discovery of pulsars in the cosmos can aid astrophysics study. There are currently millions of pulsar candidates available for search. From such a wide pool of candidates, pulsar detection can be aided by machine learning techniques. The article discusses nine widely used classification algorithms for the prediction of pulsar stars, and compares the results of each algorithm's performance using a variety of classification metrics, including classification accuracy, precision, and recall value, ROC score, and f-score on both balanced and unbalanced data.** **For improved outcomes, the data are balanced using the SMOTE approach. The XG Boosting method outperformed the other nine algorithms in terms of performance. The prospects of machine learning for pulsar identification in the realm of astronomy are discussed in the paper's conclusion.**

Keywords— **Accuracy, Algorithms, Classification, Machine Learning, Pulsars, SMOTE**.

1. Introduction

Pulsars are revolving neutron stars with a very strong magnetic field and extremely regular radiation bursts that typically last between a few milliseconds and several seconds [1]. Strong beams of light are produced by this accelerated light. They are widely used in astronomy. The importance of their successful detection in the field of astrophysics is due to the wide range of crucial applications. The task of pulsar detection has been accomplished using a variety of astronomical techniques. The radiation from pulsars is captured in large part by astronomical instruments.

For the purpose of pulsar detection, machine learning algorithms can be a very helpful alternative. The universe contains millions of pulsars, and classification methods can be used to conduct accurate detection. The k-nearest neighbour technique, logistic regression, support vector machines, decision trees, and ensembles of decision trees are only a few of the common classification algorithms covered in this work.

These techniques have been applied to a dataset of pulsars that includes around 18,000 samples of potential pulsars, of which roughly 1,600 are confirmed pulsars. The paper presents the algorithms' implementation details in depth. The accuracy, f-score, specificity, sensitivity, and ROC value of the algorithms discussed here have all been compared to see which one performs the best.

The comparisons' findings are detailed in the study. It is described how an artificial neural network was implemented on the dataset and how well it performed in comparison to other classification algorithms. Since this is a complete machine learning project, information on how the algorithms were deployed is also provided.

The performance of the best classifier is discussed in the paper's conclusion. At the conclusion of the paper, potential uses and future directions for machine learning algorithms for pulsar candidate detection are discussed.

1. Literature survey

In the past, a number of authors and experts have made predictions about pulsars in space. There have been numerous investigations, including those using machine learning techniques, to find these pulsars.

In his paper, N. Obody [2] discussed how well support vector machines (SVM) and artificial neural networks (ANN) performed in predicting pulsars from the same dataset as this paper. He gave a thorough analysis of the two approaches. He came to the conclusion that both SVM neural networks with linear kernels had a 98% accuracy rate.

Nevertheless, he came to the conclusion that none of the approaches stood out as a superior choice.

In their article, Zhen Hong Shang et al.[3] provided three classification algorithms and discussed how well they performed when it came to pulsar detection. Techniques for categorization based on decision trees, support vector machines, and neural networks were all presented.

In their article, P. Mounika et al. [4] examined the efficacy of four machine learning models for pulsar classification.

They discussed k-nearest neighbour, support vector machines, random forests, and decision tree classifier (KNN).

They came to the conclusion that the KNN performed better than the other models.

The EPN pulsar dataset was used by Amitesh Singh et al. [5] in their research, and machine learning regression algorithms like decision tree regressor, k-nearest neighbour regressor, and support vector

Pulsar prediction using the vector regressor and other methods. To discover the most effective algorithm for the job, they compared the regressors' performances. In order to cut down on the number of periods, they adopted FFA procedures.

For the aforementioned job, a variety of alternative machine learning techniques can be applied. In order to analyse their performance and determine which algorithm will work best for classifying pulsars, this project integrates all machine learning classification algorithms under one roof.

1. System architecture

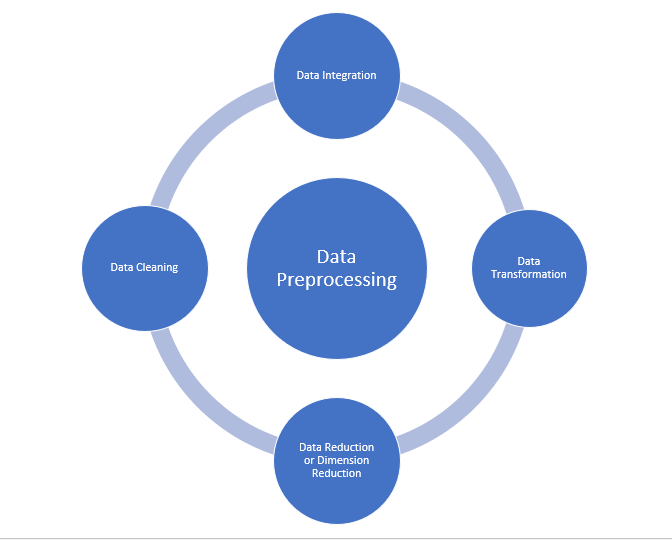


Fig. 1 Data Pre-processing

For any model, we must first provide the dataset as the information, pre-process it, then select the model and evaluate the accuracy of our model. The optimum computation for our model will be one that provides a high degree of precision.

*Data Pre-processing:*

We will handle the null values and missing values in the dataset pre-processing step. Information Pre-processing is an information mining technique that entails arranging unorganised data logically. Real data frequently contains many misunderstandings since it is sometimes fragmented, in conflict, and weak in explicit practises or floats. Pre-processing of information is a proven method for resolving such problems. Data preparation is required when data is lacking, erratic, or noisy. There are numerous ways to handle raw data:

Data Cleaning: the practise of utilising the mean, median, mode, or any other formula to fill in the missing values in data.

Data reduction: The dataset is changed in this phase to ensure that any model's outputs will be the same, but unnecessary dataset values are also eliminated.

Data Integration: If necessary, data from different sources is combined in this step, and redundant information is also eliminated.

Data Transformation: In this step, operations like normalization is performed.

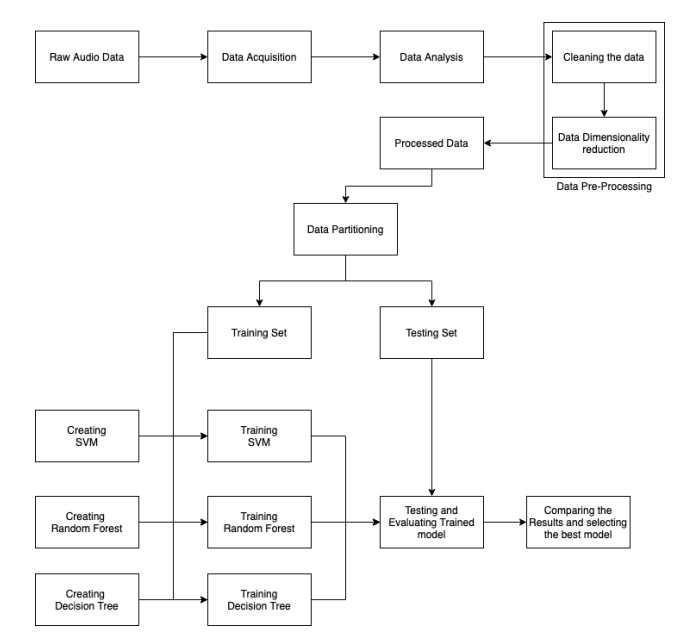


Fig. 2 System architecture of Pulsar star prediction system

Data Acquisition:

Data acquisition is the process of importing or loading the required data into a Python workspace.

transforming common tabular data, such as csv files, etc., into data that Python can understand, like a "ndarray" object or "numpy" object.

Data Analysis:

Data analysis entails comprehending the fundamentals of the loaded data. to understand how many rows and columns there are, what kind of data each column contains, and how to create graphs using those statistics.so that we may quickly complete the step of data pre-processing.

Data Pre-processing:

Data pre-processing entails preparing the data for input into the algorithm by cleaning and preparing it.

1. Cleaning: managing or deleting the dataset's empty values.

2. Dimensionality reduction: removing columns and features from the dataset that aren't necessary for solving the classification problem in order to lessen the computational load on the hardware.

Data Splitting:

Data splitting entails the development of the training set and testing set in order to train the algorithm and assess the model's performance.

Creating Algorithms:

starting the training by creating numerous algorithms that can receive input and produce output and giving them the train data.

Training:

making a certain algorithm comprehend the training data and develop concept intelligence.

Testing:

Process used to forecast the results for the test set's inputs. also comprehend the Trained Model's performance.

1. METHODOLOGY
2. *The Dataset:*

17,898 pulsar candidates are present in the dataset utilised for pulsar detection[6]. The dataset can be found at [7]. 16,259 of these 17,898 possibilities are phoney samples that were created by other people. The remaining 1,639 are confirmed pulsar samples, while the rest were obscured by noise or radio frequency interference (RFI). Eight continuous characteristics describe each sample in the collection. The integrated profile of pulse provided the statistics for the first four features. This continuous sequence of variables describes a signal that has been time- and frequency-averaged and has been resolved for longitude. Similar results are found for the remaining four feature variables from the DM-SNR (Dispersion Measure- Signal to Weight Ratio) curve[8]. SNR, or signal-to-noise ratio, is a measurement of signal strength inversely related to background noise. The 8 features taken are:

•Mean of the integrated profile.

• Standard deviation of the integrated profile.

• Excess kurtosis of the integrated profile.

• Skewness of the integrated profile.

• Mean of the DM-SNR curve.

• Standard deviation of the DM-SNR curve.

• Excess kurtosis of the DM-SNR curve.

• Skewness of the DM-SNR curve.

1. *Data Pre-Processing*

90% of the total samples, or 16,108 observations, are in the training set used to put machine learning models into action. The test set includes the remaining 10% of samples. The training set is a class-imbalanced dataset, with the non-pulsar class accounting for the majority of the observations. There is no need for extra pre-processing, such as normalisation and standardisation, because the data are regularly distributed. The k-fold cross-validation approach is used to compare the created models. This page

1. *Balancing the dataset using SMOTE*

Any classification method will perform badly on the minority class while learning to construct a decision boundary due to the imbalanced nature of the dataset employed. By oversampling the data in the dataset's minority class, this issue can be resolved. This is accomplished by copying the minority class's data before the model is trained and fitted. This type of data augmentation exists. The newly synthesised data don't give the model any new information. Synthetic Minority Oversampling Technique, generally known as SMOTE[11], is the oversampling method that is most frequently utilised.

SMOTE chooses data examples that are near the feature space in order to function. The model works well because it makes sense and because the data points are near to the feature space.

1. *Model Implementation*

The Python computer language's Scikit package is used to implement the machine learning models for pulsar prediction[9]. The aforementioned algorithms are put into practise and evaluated utilising the 10-fold cross validation technique. The models' optimal hyperparameters are tweaked for the optimum performance.

"n neighbour" = 7 is used to implement the K-nearest neighbour algorithm. This suggests that the categorization process takes into account 7 neighbours. Equation (2), where p=2, is used to get the distance between the observations, or Euclidean distance. Implementation of the support vector classifier uses C=1.0. The model is trained using the linear kernel because it produced superior results when compared to other kernels. Entropy criteria and a decision tree classifier with a maximum depth of 6 are used in its implementation.

The "max feature" setting in the random forest algorithm, which determined the number of trees utilised to build the random forest, is set to 100. Additionally, "max depth"=6 is set for the model's training, which determines the tree's depth. Overfitting of the model is avoided by using "max depth"=6. The decision tree is used as the basis estimator and the adaptive boosting algorithm is constructed with "n estimate"=100, a learning rate of 1.0, and these parameters. Implementation of the gradient descent boosting classifier uses "n estimate"=100 and "max depth"=1.0. When training the model, a "learning \_rate" of 1.0 is employed. "mlogloss" is used to implement the XG boosting technique.

1. Conclusions

The most extensively researched natural phenomenon in radio astronomy is pulsars. The outcomes of this project demonstratehat the finding of pulsar stars has enormous potential thanks to machine learning systems. Its great accuracy and minimal prediction time might be credited for this.Using five classification measures, including classification accuracy, f-score, precision, recall, and ROC value, this research built the nine most popular classification algorithms and assessed their performance on 10-fold cross-validation. e. The algorithms for categorizing` data were applied to both unbalanced and SMOTE-balanced data. The algorithms perform better on SMOTE-balanced data than they do on unbalanced data. The XGBoosting approach outperformed the other eight algorithms on both types of training data, with the maximum accuracy of approximately 98%, while the other models were only slightly worse than the XGBoost model. Additionally, XGBoosting performed better in terms of ROC values, precision, recall, and f-score. Additionally, a nine-layered artificial neural network with minimal overfitting demonstrated accuracy of nearly 98%. As a result, it can be said that neither of them stands out as a superior choice than the other. The greatest outcome for pulsar detection will come from implementing any one of them[10].

The findings in this work can aid in future investigations into radio astronomy for pulsar detection. Machine learning will be used more effectively while also accelerating the process. The outcomes reported here will show to be quite beneficial if machine learning models are heavily applied to the work.

References

1. <https://imagine.gsfc.nasa.gov/science/objects/neutron_stars1.html>
2. https://d1b10bmlvqabco.cloudfront.net/attach/jz8smbptoj35ra/i8xgc5x4 yhoyo/k0fny104qt72/project.pdf
3. Cheng Jun Zhang, Zhen Hong Shang, Wan Min Chen, Liu Xie, Xiang Hua Miao. (2020). A Review of Research on Pulsar Candidate Recognition Based on Machine Learning. Procedia Computer Science, Volume 166, 2020, Pages 534-538, ISSN 1877-0509, <https://doi.org/10.1016/j.procs.2020.02.050>.
4. <https://www.irjet.net/archives/V7/i6/IRJET-V7I61195.pdf>
5. Singh, Amitesh & Pathak, Kamlesh. (2020). A machine learning-based approach towards the improvement of SNR of pulsar signals.
6. R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, J. D. Knowles, Fifty Years of Pulsar Candidate Selection: From simple filters to a new principled real-time classification approach MNRAS, 2016.
7. https://archive.ics.uci.edu/ml/datasets/HTRU2
8. R. J. Lyon, "Why Are Pulsars Hard To Find?", PhD Thesis, University of Manchester, 2015.
9. https://scikitlearn.org/stable/supervised\_learning.html#supervisedlearning
10. <https://github.com/apratimsadhu01/Pulsar-Star-Detection>
11. Chawla, Nitesh & Bowyer, Kevin & Hall, Lawrence & Kegelmeyer, W.. (2002). SMOTE: Synthetic Minority Over-sampling Technique. J. Artif. Intell. Res. (JAIR). 16. 321-357. 10.1613/jair.953.