IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

A Machine Learning Based Approach for the Identification and Estimation of Pulsars upon Statistical Analysis

Dr. Shivaprasad. K. M
Professor, Department of ECE
R L Jalappa Institute of Technology, Doddaballapura, Bengaluru, India

Abstract: *The life cycle of stars has been an extensive topic of study in the field of astrophysics as it gives us a better understanding of the formation of planets and life. Stars shine by burning their hydrogen fuel in a nuclear fusion reaction. The energy released by this reaction keeps a star from collapsing onto itself due to its own gravity. But, towards the end of its life, a star depletes all of its fuel and there is nothing to help it from collapsing onto its own gravity. A star that weighs about 3 to 4 solar masses forms either a pulsar neutron star or a magnetron neutron star. Study of these stars is very important as it gives better insight into the formation and destruction of stars and planets. This study aims to develop a precise and effective machine learning model for detecting pulsar stars. Efforts have been put into evaluating the performance of a number of well-known machine-learning algorithms including K-Nearest Neighbors, Multiple Adaptive Regression Spline (MARS), neural network classifiers, etc., for identifying Pulsars using R programming language*

Keywords: pulsar stars, multiple adaptive regression spline, k nearest neighbors, neural networks, machine learning, R programming.

I. INTRODUCTION

One of the most intriguing areas of science is astrophysics. And a wealth of data from several surveys, like the Sloan Digital Sky Survey [1] and the High Time Resolution Universe Survey, is what fuels an ever-increasing understanding of our Universe [2]. Astronomers are searching for scalable and intelligent data management as a result of the previous ten years' worth of technical improvement. Various machine learning techniques have found applications in this domain. The word "Pulsar" is a combination of the words "Star" and "Pulse," and it refers to a particular class of neutron star.

Every proton and electron in the core of a huge star that has run out of fuel crashes to the ground. A neutron star is created if the neutron-only core that results from supernova explosion weighs between one and three solar masses. Pulsars and magnetron stars are two types of neutron stars. The most prevalent kind of neutron star that periodically emits radiation pulses is a pulsar star. Pulsar stars emit light beams that may be seen from Earth when they pass through our line of sight. For a number of reasons, pulsars have been called superb nature laboratories. The characteristics of gravity, the makeup of the interstellar medium (ISM), and many other topics are all studied through the use of pulsars. Machine learning techniques for pulsar star detection is a relatively young field of study. This procedure was carried out manually until recently. Lately, deep learning techniques such as artificial neural networks have also found application in detection of pulsar stars. [3]

The research content can be summarized as follows: Section 2 delves into existing methods and relevant studies in the field. Methodological details, including data pre-processing and the deployment of various models, are covered in Section 3. Section 4 is dedicated to the exploratory data analysis, identification of important features, and model evaluation for pulsar star classification using the three algorithms. Section 5 provides an insight into the performance and evaluation of three models (MARS, K Nearest Model, and Neural Network Model) for pulsar star classification, highlighting accuracy, evaluation metrics, and specific requirements for selecting the most suitable model. The research

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

contributions are discussed in Section 6. The paper concludes with Section 7, where a summary and implications of the findings are articulated.

II. LITERATURE REVIEW

Detecting pulsars is a crucial task for understanding various nuclear physics events. In this context, the topic of automatic pulsar detection from the acquired data is of paramount importance. The HTRU2 dataset's imbalance makes it so that the prediction accuracy is below par. [4] in their paper suggest a method called concatenated resampling for balancing data and a strategy to use the suggested method for very accurate prediction of pulsars. When using deep learning models, long short-term memory and Gated Recurrent Unit neural networks offer higher F1 scores than deep neural networks. When compared in terms of performance to cutting-edge approaches, the proposed approach performs better and achieves higher accuracy. By improving a number of well-known ML models, this study makes use of the supervised technique.

In a recent work done at Cornell university, [5] looked into the possibility of identifying pulsar candidates using probabilistic learning. Two deep learning models are used, a deep belief network based on Gaussian process mappings and Deep Kernel Learning. Overall, the performance of the models is extremely encouraging, attaining a pretty strong probability of discriminating between a positive and negative class. This study has shown how deep probabilistic learning can be used to identify pulsars using their characteristic properties, which include 2D dispersion measurements, sub-bands, and sub-integrations.

X-ray binaries consist of a compact entity that absorbs materials from an orbital companion star (XRBs). Creating effective, reliable methods to categorize tiny objects becomes more crucial as new X-ray sources are found using various X-ray observatories. To identify compact objects in XRB systems as either black holes or neutron stars, three ML techniques (BGP, KNN, and SVM) are examined by [6]. The average prediction accuracy for all three is a respectably high 81.8% to 84.1%. The KNN model performs better than the SVM and BGP in terms of predicting accuracy. Compared to 81.8% for Support vector machines and Bayesian Gaussian Processes, the K-Nearest Neighbors model predicted the categorization of 84.1% of the sources accurately. These discrepancies in prediction ability are mostly caused by variations in BH misclassifications or uncertain classifications across approaches.

To categorize unbalanced pulsar candidates in the study by [7], XGBoost, Random Forest, and a Hybrid Ensemble method were implemented. These models based on trees were utilized to choose features from the various (about 30) features of pulsar candidates assisting these methods. In XGBoost, the skewness of the integrated pulse profile is one of the most important factors, but in Random Forest, the chi-squared value for sine-squared fitted to the modified pattern and the value of optimal S/N play important roles. By comparing their relative scores, more than 20 features were chosen, and three approximate methods were used to apply them. Easy Ensemble, Random Forest, and XGBoost are merged in the Hybrid Ensemble approach.

In a study conducted by [4], many pulsar prediction methodologies now in use are looked into such as adding filters premised on observations of pulsars, which can reduce the accuracy of prediction of pulsars. Few current techniques are unable to handle large amounts of data, and others are unable to precisely choose the most qualified candidates from pulsar observations. As a result, a novel machine learning strategy was created that produces better outcomes. The study presents a hybrid classifier predicting pulsar stars. It is called the random trees boosting voting classifier (RTB-VC). RTB-VC, which merges tree-based classifiers, uses the High Time Resolution Universe 2 (HTRU2) data collection, which consists of eight features associated with pulsars as well as non-pulsars. Fake data is created using an oversampling technique and get a balanced dataset in order to address the imbalance in the HTRU2 data set. Candidates for pulsars and non-pulsars are differentiated using a feature set since the data set's variable distributions are valuable for training models. The ensemble-based structure of random trees boosting voting classifier generates credible conclusions based on given data. It has a high F1 score for identifying pulsars (98.3%), which indicates the reliability of these findings.

Another study shows that pulsar identification may very well be successfully accomplished utilizing ensemble classification strategies which are particularly tailored for dealing with imbalanced classification issues. The researched ensembles outperform approaches that do not take into account class balance, according to test findings relying on HTRU2 data, and their usage for other astrophysical applications is suggested [8].

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

A study discovered that the test results of three approaches on the HTRU2 dataset are excellent considering the under-sampling approach. It performs better than 96% and conforms to 10-fold repeated cross validation. Random Forest produced results with a higher level of accuracy when compared to bagged Classification and Regression Tree and Adaboost classification trees. Thus, Random Forest offers a solid defense against under-fitting, especially when dealing with an unbalanced data set like HTRU2. The test performance of the three approaches is also strong in the situation of over-sampling; the accuracy measure performs better than 96% and is nearly as excellent as repeated cross validation performance. In both cases of undersampling and oversampling, the Adaboost classification tree algorithm and the Random Forest algorithm perform significantly better than the bagged Classification and Regression Tree classifier. Both of these results can be attributed to their efficient operation. They come to the conclusion that the Adaboost and Random Forest Algorithms work successfully to adjust the train set distribution. This is especially true in the case of an unbalanced data set such as HTRU2 [9].

It was found that the use of ensemble learning techniques helps minimize mistakes brought on by the employment of certain algorithms. The accuracy of ensemble learning can be increased by the various separate algorithms working together to make decisions. In this study, pulsar candidates are found using the Bagging approach in ensemble learning, with the Radial Basis Function (RBF) neural network algorithm used as the sub-classifier. Through the use of supervised learning, the sub-classifiers were taught. The experiment demonstrates that this approach not only ensures training speed but also enhances pulsar candidate detection performance. The algorithm can also cut down on the expense of manual participation. In this research, a pulsar candidate detection approach based on bagging was developed to address the issue of pulsar candidate detection. The signal feature was extracted using the observation signal's cumulative contour. In the paper, nine characteristics were extracted and the sub-classifier in the Bagging detection technique was RBF. The most recent detection results were compiled using a vote system. Simulation tests validate this algorithm's performance in terms of detection. The simulation results demonstrate that the approach described in this research outperforms both the SVM algorithm and the ANN algorithm in terms of detection performance [10].

An essential requirement for pulsar navigation is the quick and precise detection of X-ray pulsar emissions. The majority of the existing approaches for identification include extracting the cumulative profile characteristics and then comparing them to aspects of the standard profile. However, a high signal-to-noise ratio profile necessitates prolonged observation, which has a significant influence on real-time identification. The X-ray pulsar signal is transformed into time interval sequences in this study, and one-dimensional convolution neural networks are used to finish the feature extraction and identification. A convolutional block attention module (CBAM) and suggest using the CBAM-Inception module to build the network in terms of network architectural design. The benefits of Inception and CBAM are combined in this structure, which also leverages channel and spatial attention techniques to improve the Inception network's ability to extract features. The suggested technique by [11] may significantly reduce the necessary observation time while ensuring high-accuracy X-ray pulsar signal detection, according to experimental results. Additionally, the CBAM-Inception block can significantly increase network recognition capabilities, as shown by a comparison of convolution blocks.

Authors of a another paper [12] have carried out a research study based on the distinctive aspects of the dataset in great detail and documented it for the star classification as well as for the research of pulsars. There has been construction of decision models that are based on multiple classifiers, and these models have been compared to other published results. The results of applying machine learning methods to the datasets of stellar and pulsar objects are encouraging to say the least. In contrast to the imbalanced pulsar data, the stellar data presents a balanced classification challenge. The trials have produced good accuracy ratings for both data sets after being averaged over multiple train and test split runs. Heatmaps are used to show the correlation of the qualities, showing how strongly the attributes are related to the target class.

In a study by [13], it was concluded that higher F1- Scores are possible by using sampling techniques such stratified K-fold cross-validation, which significantly reduce the model's misclassifications. The best classifiers for oversampling and under-sampling, respectively, in pulsar classification are Random Forest & Logistic Regression. It would appear that selecting features to include in the model does not improve its accuracy because all of the characteristics provide information that is required to categorise candidates as pulsars. Because of the considerable imbalance that exists

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

between the target classes in the HTRU-2 dataset, sampling and resampling techniques are constantly required to suppress this imbalance.

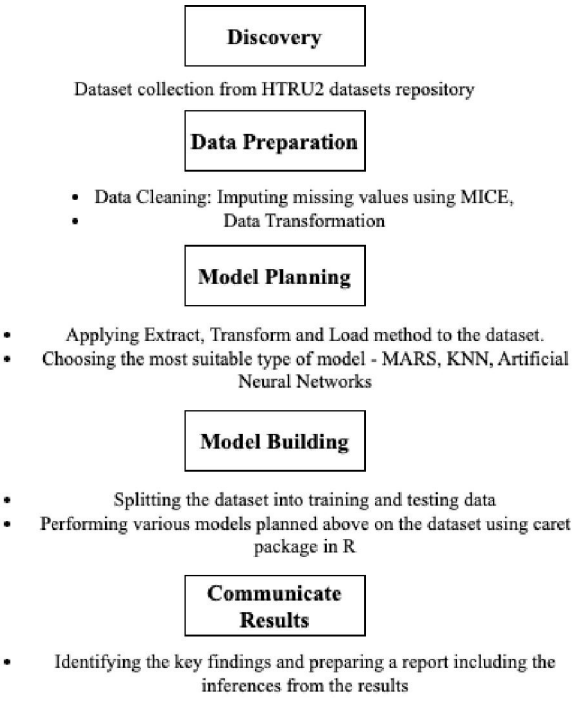
Pulsar detecting telescopes are required to classify radio wave data as it is broadcast. For traditional machine learning algorithms to provide useful results, a lot of processing power, memory, and time would be required. (Recent surveys gather information at a pace of 0.5 to 1 terabyte per second). The aforesaid restrictions are explicitly addressed by stream classification algorithms. Extremely Fast Decision Tree is one of the strategies for stream categorization that has the potential to learn progressively as it comes into contact with fresh data. The data coming from the pulsar detection data streams, on the other hand, are quite imbalanced (there are less examples of pulsars in the data than non-pulsar objects). It would be detrimental to the model's accuracy to learn progressively from such a data stream (of detecting pulsars). Hence, an enhanced Extremely Fast Decision Tree is provided in a study by [14] that can recognize unbalanced data streams.

[15] primarily focus on the development of a more efficient algorithm for outlier detection in data mining, a field crucial for various applications such as fraud detection, activity monitoring, and network intrusion detection. They argue that existing outlier detection methods fall short in handling complex patterns, largely due to their inability to accurately estimate density and extract proper neighbor information. Their research introduces a neighborhood weighted-based outlier detection (NWOD) algorithm, which tackles the above issues and can yield reliable detection results in various situations. The novel algorithm, NWOD, is shown to outperform its counterparts through experiments conducted on synthetic and real-world datasets such as the HTRU2.

III. PROPOSED METHODOLOGY

A. Workflow

As the problem statement required analysis of data, the workflow followed the general data analysis life cycle as shown in Figure 1. The first stage of the workflow involved discovering the data. The Pulsar dataset used in the work is obtained from the HTRU2 datasets repository. The second stage of the workflow involved preparation of the data by cleaning it and then transforming it. A library called MICE was used to clean and impute the data. The third stage of the workflow involved planning of the model.



Data Analysis life cycle

Algorithms like KNN, MARS, and Neural Networks were chosen as the problem statement fell under classification category. The fourth stage of the workflow involved performing training of the models after splitting the dataset into

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

training data and testing data. The package called CARET was used to train the KNN model and the MARS model, while the NeuralNet package was used for training the Neural Network model. The fifth and the final stage of the workflow involved identification of the outcomes and preparation of a report

B. Materials

i. System requirements: The platform should support RStudio and the required packages. RStudio is typically compatible with Windows, macOS, and Linux operating systems.
ii. Software requirements:
RStudio: Version 4.2.2 or higher is required to run the analysis and code smoothly.
R: RStudio requires R to be installed on the system. Version 4.2.2 or higher is recommended as specified.
iii. Package requirements:
caret: Used for training and evaluating machine learning models.
mice: Utilized for handling missing data through multiple imputation.
skimr: Used for summarizing data and displaying missing value information.
corrplot: Employed for creating correlation plots to visualize attribute relationships.
tidyverse: A collection of R packages used for data manipulation and visualization.
ggplot2: A powerful data visualization package in R.
neuralnet: Used for building artificial neural network models.
nnet: Another package used for neural network modeling.

C. Methods

Splitting ratio of the dataset: 80% training data and 20% testing data

1. Multivariate Adaptive Regression Splines model

MARS is a flexible non-parametric regression technique used for modeling complex relationships between multiple input variables, denoted by $\boldsymbol{x} \in \mathbb{R}^D$, and a single output variable, denoted by $y \in \mathbb{R}$. The foundation of MARS lies in the idea of piecewise linear functions and the use of basis functions to construct the model. The algorithm builds a model in a step-wise manner, starting from a constant value and iteratively adding basis functions (piecewise linear functions) to capture non-linear relationships between the predictors and the target variable. [16]
The MARS model can be represented mathematically as follows:
Start with a constant model:

$$f(\boldsymbol{x}) = c \qquad (1)$$

Add basis functions to the model in a step-wise manner:

$$f_k(\boldsymbol{x}) = c_k + \sum_{j=1}^{J} \beta_{kj} h(\boldsymbol{x}, \boldsymbol{\gamma}_{kj}) \qquad (2)$$

where $(f_k(\boldsymbol{x}))$ is the (k)-th model term, (c_k) is a constant coefficient for the (k)-th term, (β_{kj}) is the regression coefficient associated with the basis function $(h(\boldsymbol{x}, \boldsymbol{\gamma}_{kj}))$ in Equation (2).
$(h(\boldsymbol{x}, \boldsymbol{\gamma}_{kj}))$ is a basis function that is a product of one-dimensional basis functions:

$$h(\boldsymbol{x}, \boldsymbol{\gamma}_{kj}) = \prod_{d=1}^{D} h_{jd}(x_d, \gamma_{kjd}) \qquad (3)$$

where (x_d) is the value of the (d)-th predictor variable, (D) is the number of predictors, (h_{jd}) is a univariate basis function, and (γ_{kjd}) are tuning parameters that control the location and width of the basis function in Equation 3.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

The basis functions are chosen through a forward-backward algorithm that starts with a single basis function and adds more basis functions while optimizing the model's performance using a criterion like the Sum of Squared Residuals (SSR) or other goodness-of-fit metrics.

The algorithm continues to add or remove basis functions until a stopping criterion is met (e.g., a predefined number of basis functions or a maximum level of interaction complexity).

The MARS algorithm effectively handles non-linear relationships, interactions, and variable selection. It automatically determines the number of basis functions and their locations, making it a powerful tool for modeling complex data without assuming a specific functional form for the relationship between the predictors and the target variable. Since MARS uses piecewise linear functions, it is computationally efficient and well-suited for handling high-dimensional data with complex interactions. It has found applications in various fields, including data mining, statistics, machine learning, and pattern recognition. MARS is particularly useful when dealing with data that exhibit non-linear and interactive effects among the predictor variables.

The helper packages for this model in the R programming language are dplyr for data wrangling and ggplot2 for plotting. For fitting MARS models, we use earth package and for automating the tuning process, caret package is used. Multivariate adaptive regression splines (MARS) offer a practical method for capturing nonlinear relationships in the data by evaluating cut points (knots) in a manner comparable to step functions. This technique assesses each data item for each predictor as a knot, and it produces a linear regression model with the candidate feature as its dependent variable (s). The target class, training data and earth method were passed to the train function of caret package to train the model. The 'degree' tuning parameter was kept at a constant value of 1. Accuracy was employed in order to pick the best model out of those that had the highest value. The model was trained with the following final parameters: nprune = 14, and degree = 1.

2. K-Nearest Neighbors model

K-Nearest Neighbors (KNN) is a popular and intuitive non-parametric supervised learning algorithm used for both classification and regression tasks. The foundation of KNN lies in the idea that similar data points tend to have similar labels or target values. It makes predictions based on the majority class or average value of the k-nearest data points in the feature space.

KNN is based on the assumption that data points belonging to the same class or having similar target values tend to be close to each other in the feature space. It utilizes the concept of distance metrics, such as Euclidean distance, to measure the similarity between data points. The key idea is to find the k-nearest neighbors to a given data point and use their labels (in classification) or target values (in regression) to make predictions for that data point.

For a classification task with a training dataset of labeled instances $(\{(x_1, y_1), (x_2, y_2)$, where (x_i) represents the feature vector of the (i)-th instance, and (y_i) is the corresponding class label (categorical variable), for a new data point (x_{new}), the KNN algorithm proceeds as follows:

Compute the distance between (x_{new}) and all training data points $(\{x_1, x_2, \ldots, x_n\})$ using a distance metric (e.g., Euclidean distance):

$$[\text{distance}(x_{\text{new}}, x_i) = \sqrt{\sum_{j=1}^{D}(x_{\text{new},j} - \qquad (4)$$

where (D) is the number of features, and $(x_{\text{new},j})$ and $(x_{i,j})$ are the (j)-th features of the new data point and the (i)-th training data point, respectively in Equation 4.

Select the (k) data points with the smallest distances to (x_{new}).

For classification, determine the majority class among the (k) neighbors and assign that class label to (x_{new}).

For a regression task, the process is similar, but instead of taking the majority vote, the average (or weighted average) of the target values of the (k) nearest neighbors is used as the predicted target value for $(x_{\text{nev}}$.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

The choice of the hyperparameter (k) is critical in KNN. A small value of (k) may lead to noise influencing the prediction, while a large value of (k) may smooth out important local patterns in the data. Typically, the value of (k) is chosen through hyperparameter tuning, such as cross-validation, to find the optimal balance between bias and variance in the model.

KNN is a simple and intuitive algorithm, but it can be computationally expensive, especially for large datasets. It also requires careful preprocessing and scaling of features to avoid undue influence from features with larger scales. Nonetheless, KNN is widely used for its simplicity, effectiveness in handling non-linear relationships, and ease of implementation.

For fitting K-NN models, caret package is used. K-NN is a memory-based algorithm which makes predictions about each observation based on its "similarity" to other observations. This suggests that training samples are required at runtime and that predictions are created based only on the associations between the samples themselves. This supervised learning model is implemented using caret library in which the trainControl() method is used to set the three parameters with repeated cross-validation resampling technique. Ten resampling iterations with three complete sets of folds are computed. Dataset is preprocessed to center and scale the data. After pre-processing, training data is converted with the mean value set to approximately 0 and the standard deviation set to 1. The algorithm is tuned using 10 as the tune length. Accuracy was employed in order to pick the best model out of those that had the highest value. The model was run with k equal to 15, which was the final value. [17][18]

3. Neural Networks

Artificial Deep Neural Networks (DNN), often referred to simply as deep neural networks or deep learning models, are a class of machine learning models inspired by the structure and function of the human brain. These networks consist of multiple layers of interconnected artificial neurons, and they are particularly powerful in solving complex problems across various domains, including computer vision, natural language processing, and speech recognition.

The foundation of DNN lies in the concept of mimicking the human brain's neural connections and learning process. The basic building blocks of DNN are artificial neurons, also known as nodes or units, which are organized into layers. The information flows through these neurons, where each neuron applies an activation function to the weighted sum of its inputs. Deep learning models can learn to automatically extract hierarchical features from the input data, enabling them to represent complex patterns and relationships.

A typical feedforward DNN consists of an input layer, one or more hidden layers, and an output layer. Each layer has a set of neurons, and connections between neurons in adjacent layers are associated with weights.

Let's represent a DNN with (L) layers as follows:

Input layer: The input layer consists of (N) neurons, where each neuron represents a feature of the input data. The input vector is denoted as $(\boldsymbol{x} \in \mathbb{R}^N)$.

Hidden layers: There can be multiple hidden layers (commonly denoted by (H)) in a deep neural network. Each hidden layer contains a certain number of neurons, and the output of one hidden layer serves as the input to the next layer. Let $(\boldsymbol{h}^{(l)} \in \mathbb{R}^{M_l})$ represent the output vector of the (l)-th hidden layer, where (M_l) is the number of neurons in the (l)-th hidden layer.

Output layer: The output layer typically consists of (K) neurons, where (K) is the number of classes for classification tasks or the number of output units for regression tasks. The output vector is denoted as $(\boldsymbol{y} \in$.

The forward propagation in a DNN can be described as follows:

For the first hidden layer $((l = 1))$:

$$[\boldsymbol{h}^{(1)} = \phi^{(1)}(\boldsymbol{W}^{(1)}\boldsymbol{x} + \boldsymbol{b}^{(1)})] \quad (5)$$

For the subsequent hidden layers $((l = 2, 3, \ldots,$:

$$[\boldsymbol{h}^{(l)} = \phi^{(l)}(\boldsymbol{W}^{(l)}\boldsymbol{h}^{(l-1)} + \boldsymbol{b}^{(l)})] \quad (6)$$

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

For the output layer $((l = H + 1))$:

$$[\boldsymbol{y} = \phi^{(H+1)}(\boldsymbol{W}^{(H+1)}\boldsymbol{h}^{(H)} + \boldsymbol{b}^{(I \qquad (7)$$

Where $(\boldsymbol{W}^{(l)})$ is the weight matrix for the (l)-th layer. $(\boldsymbol{b}^{(l)})$ is the bias vector for the (l)-th layer. $(\phi^{(l)})$ is the activation function applied element-wise to the output of each neuron in the (l)-th layer in Equations 5, 6, 7.

The activation function $(\phi^{(l)})$ introduces non-linearity into the model, allowing it to learn complex relationships in the data. Common activation functions include the Rectified Linear Unit (ReLU), sigmoid, and hyperbolic tangent (tanh).

The DNN learns from data using an optimization algorithm like stochastic gradient descent (SGD) or its variants (e.g., Adam, RMSprop). The learning process involves minimizing a loss function (also known as a cost function) that measures the difference between the predicted output and the actual target values.

The model parameters (weights and biases) are updated iteratively during the training process to minimize the loss function. This process is known as backpropagation, where gradients of the loss with respect to the model parameters are computed and used to update the weights and biases in the opposite direction of the gradient to minimize the loss.

An Artificial Neural Network model is implemented using the neuralnet and nnet libraries. The neural network performs computations through a learning process, consisting of interconnected input/output units with adjusted weights. These weights are updated during the learning phase to accurately predict the class labels of the inputs. To control the training parameters for the algorithm, the trainControl() function is utilized, applying repeated cross-validation as the resampling technique. The resampling involves 10 folds or iterations, computing three complete sets of folds.

The dataset in consideration had 9 attributes, thus a model which is capable of finding a separation between datapoints of higer dimensions is required. These algorithms were chosen as they are capable of working well with high dimensional data.

IV. RESULTS

4.1 Exploratory Data Analysis

The exploration of the data revealed some of its characteristics inferred from the summary shown in Figure 2. They are as follows:

In total, there are 12528 observations in the dataset, and 3538 missing cells. Thus, about 3.1% of the cells in the dataset have missing values. As this percentage is small, these cells could be replaced with the median value of the attribute, or they could be dropped without losing much of the data.

All the 9 attributes are numeric.

There are no duplicate rows or columns in the dataset.

Many of the data set attributes are highly correlated with one another. Some of these attributes could be dropped without much change in the accuracy of the ML model, thus making the model more efficient during the training phase.

Given in the summary of dataset (Figure 2), out of the total 3538 missing values, there are 1735 Excess kurtosis of the integrated profile values, 1175 Standard deviation of the Dispersion Measure/Signal-to-Noise Ratio curve values and 625 Skewness of the Dispersion Measure/Signal-to-Noise Ratio curve values. Other important details about the attributes can be observed such as their complete data rate, mean, standard deviation, quartiles, as well as the histogram distribution.

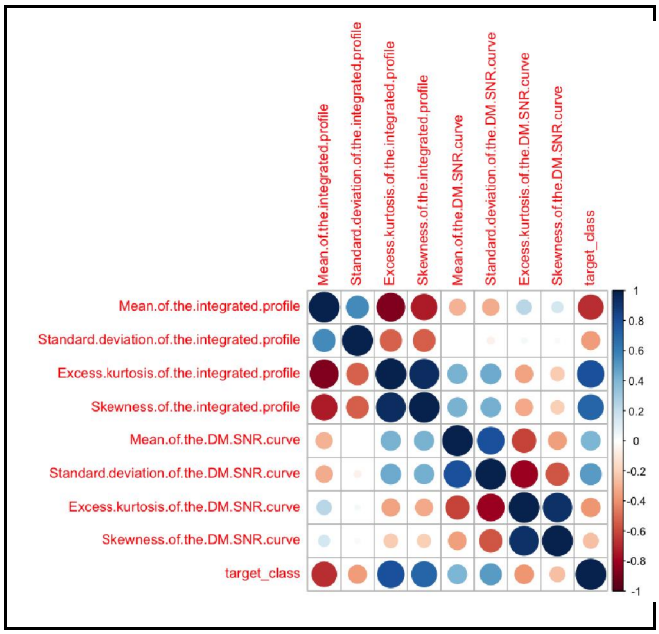
skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
Mean.of.the.integrated.profile	0	1.00	111.04	25.67	5.81	100.87	115.18	127.11	189.73	▁▁▁▃▇
Standard.deviation.of.the.integrated.profile	0	1.00	46.52	6.80	24.77	42.36	46.93	50.98	91.81	▁▃▇▂▁
Excess.kurtosis.of.the.integrated.profile	1735	0.86	0.48	1.06	-1.74	0.02	0.22	0.47	8.07	▇▁▁▁▁
Skewness.of.the.integrated.profile	0	1.00	1.78	6.21	-1.79	-0.19	0.20	0.93	68.10	▇▁▁▁▁
Mean.of.the.DM.SNR.curve	0	1.00	12.67	29.61	0.21	1.91	2.79	5.41	222.42	▇▁▁▁▁
Standard.deviation.of.the.DM.SNR.curve	1178	0.91	26.35	19.61	7.37	14.40	18.41	28.34	110.64	▇▃▁▁▁
Excess.kurtosis.of.the.DM.SNR.curve	0	1.00	8.33	4.54	-3.14	5.80	8.45	10.73	34.54	▁▇▅▁▁
Skewness.of.the.DM.SNR.curve	625	0.95	105.53	107.40	-1.98	35.20	83.13	140.00	1191.00	▇▂▁▁▁
target_class	0	1.00	0.09	0.29	0.00	0.00	0.00	0.00	1.00	▇▁▁▁▁

Summary of the dataset with histogram

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

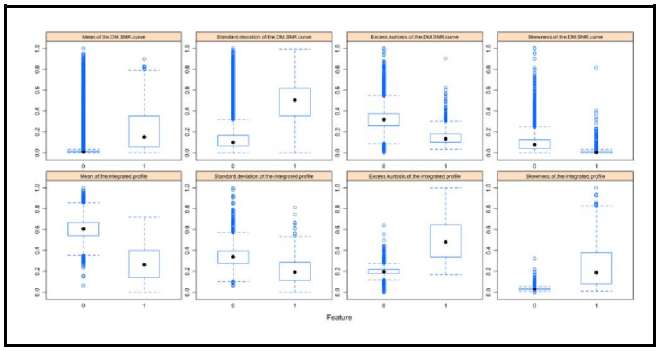
Identification of Important Features

On plotting a heatmap (Figure 3) for all the attributes, it is found that certain attributes have a high correlation. Some of these attributes are Skewness of the Integrated Profile and Excess Kurtosis of the Integrated Profile, Standard Deviation of the DM-SNR Curve and Mean of the DM-SNR Curve, Skewness of the DM-SNR Curve and Excess Kurtosis of the DM-SNR Curve. Excess Kurtosis of the Integrated Profile and Skewness of the Integrated Profile are found to have a high degree of correlation with the Target Class.



Correlation between different attributes using heatmap

On plotting a box plot (Figure 4) for all the attributes against the target class, it is found that certain attributes have a huge variation in their mean for different classes in the Target Class. These attributes possibly have the most effect on the output. Some of these attributes are Standard Deviation of the DM SNR, Excess Kurtosis of the Integrated Profile, and Mean of the Integrated Profile.

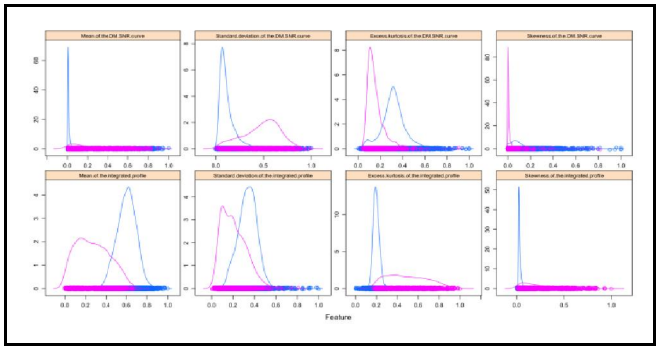


Box plot for all attributes against the target class

On plotting a density plot (Figure 5) for all the attributes against the target class, certain attributes have greatly different skewness and kurtosis of the density for different classes in the Target Class. These attributes could have a high impact on the Target Class.

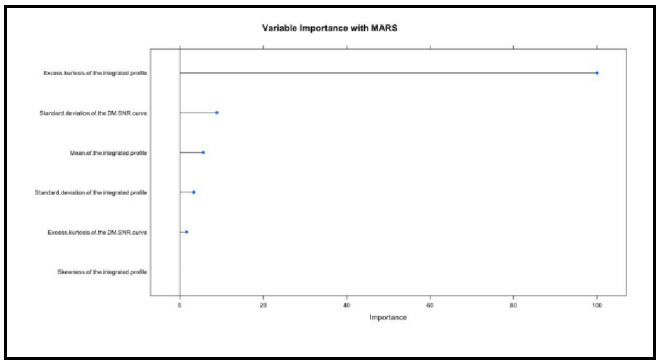
However, these attributes might not have any significant impact on the Target Class value while using certain algorithms. This is because the working of each algorithm differs, so the attributes that have a big impact on one model might not have any significant impact on another model.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023



Density plot for all attributes against the target class

Model Evaluation



Important attributes of the model trained using MARS algorithm

It is inferred from Figure 6 that the variable Excess kurtosis of the integrated profile shows the most significance while Skewness of the integrated profile does not show any importance with the Multivariate Adaptive Regression Splines model.

V. CONFUSION MATRIX OF THE MARS MODEL

	Reference		
Prediction		0	1
	0	2263	43
	1	12	187

The model produces a total of 2263 true negatives with a value of 0, 187 true positives with a value of 1, and a total of 43 false negatives and 12 false positives as shown in Table 1.

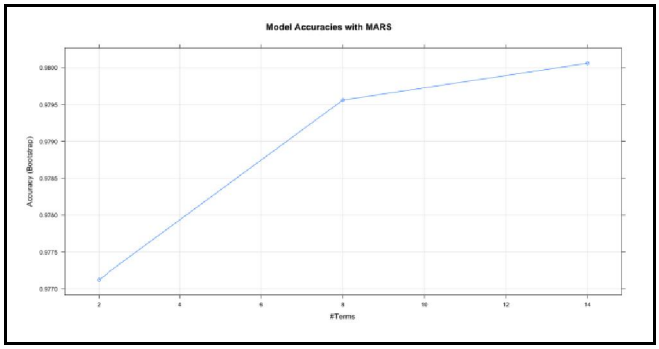
STATISTICS OF THE MARS MODEL

Accuracy	0.978
Precision	0.9813
Recall	0.9947
F1 Score	0.9879
95% CI	(0.9715, 0.9834)
No Information Rate	0.9082

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

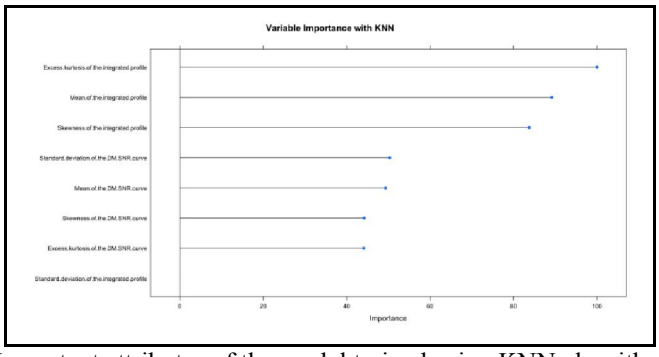
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.8599
Mcnemar's Test P-Value	5.228E-05
Sensitivity	0.9947
Specificity	0.8130
Pos Pred Value	0.9814
Neg Pred Value	0.9397
Prevalence	0.9082
Detection Rate	0.9034
Detection Prevalence	0.9206
Balanced Accuracy	0.9039
'Positive' Class	0

Upon evaluating the MARS model, the model accuracy on the testing data is found to be 97.8% as given in Table II.



Accuracies of the model trained using MARS algorithm

As given in Figure 7, the no. of terms increase from 2 to 8 the accuracy of MARS model increases sharply and then there is a slow increase from 8 to 14 terms. Highest accuracy was found when the no. of terms to retain the model i.e., nprune value = 14.



Important attributes of the model trained using KNN algorithm

It is inferred from Figure 8 that the variable Excess kurtosis of the integrated profile shows the most significance followed by the Mean of the integrated profile with the Multivariate Adaptive Regression Splines model. Mean and

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

Skewness of the integrated profile showed similar importance. Standard deviation of the integrated profile doesn't show any contribution to the model's performance.

CONFUSION MATRIX OF THE K-NN MODEL

	Reference		
Prediction		0	1
	0	2266	48
	1	9	182

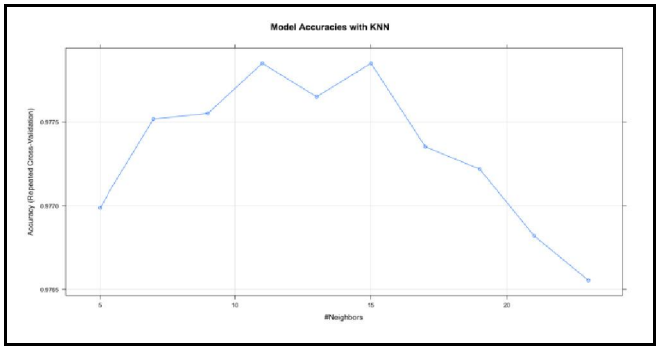
The model produces a total of 2266 true negatives with a value of 0, 182 true positives with a value of 1, and a total of 48 false negatives and 9 false positives as given in Table III.

STATISTICS OF THE KNN MODEL

Accuracy	0.9772
Precision	0.9792
Recall	0.9960
F1 Score	0.9875
95% CI	(0.9706, 0.9827)
No Information Rate	0.9082
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.8523
Mcnemar's Test P-Value	4.823E-07
Sensitivity	0.9960
Specificity	0.7913
Pos Pred Value	0.9793
Neg Pred Value	0.9529
Prevalence	0.9082
Detection Rate	0.9046
Detection Prevalence	0.9238
Balanced Accuracy	0.8937
'Positive' Class	0

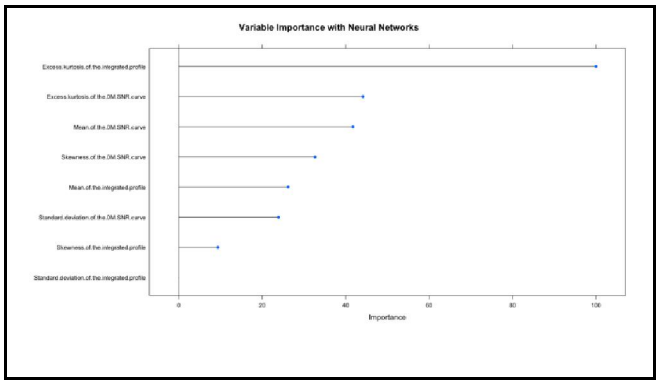
Upon evaluating the K-NN model, the model accuracy on the testing data is found to be 97.72% as shown in Table IV.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023



Accuracies of the model trained using KNN algorithm

Using KNN algorithm, accuracy of the models vary as the number of neighbors increases as shown in Figure 9. There is no steady increase or decrease in the accuracy of the model. Highest accuracy is found when the no. of neighbors = 15.



Important attributes of the model trained using Neural networks algorithm

Excess kurtosis of the integrated profile shows the most significance with the Neural networks model as shown in Figure 10. Excess kurtosis and Mean of the Dispersion Measure/Signal-to-Noise Ratio curve showed similar importance. Standard deviation of the integrated profile doesn't show any significant contribution to the model's performance.

CONFUSION MATRIX OF THE NEURAL NETWORKS MODEL

	Reference		
Prediction		0	1
	0	2261	36
	1	14	194

Table V depicts that the model produces a total of 2261 true negatives with a value of 0, 194 true positives with a value of 1, and a total of 36 false negatives and 14 false positives.

STATISTICS OF THE NEURAL NETWORKS MODEL

Accuracy	0.98
Precision	0.9843
Recall	0.9938
F1 Score	0.9890

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

95% CI	(0.9738, 0.9851)
No Information Rate	0.9082
P-Value [Acc > NIR]	< 2.2e-16
Kappa	0.8749
Mcnemar's Test P-Value	0.002979
Sensitivity	0.9938
Specificity	0.8435
Pos Pred Value	0.9843
Neg Pred Value	0.9327
Prevalence	0.9082
Detection Rate	0.9026
Detection Prevalence	0.9170
Balanced Accuracy	0.9187
'Positive' Class	0

Upon evaluating the Neural networks model, the model accuracy on the testing data is found to be 98% as depicted in Table VI.

Comparison with state-of-the-art models

The best performing models proposed in [19] and [4] have been used as the state-of-the-art benchmark for the models proposed in this paper.

TABLE VII: COMPARISON WITH STATE-OF-THE-ART MODELS

Model	Accuracy	Precision	Recall
The proposed KNN Model	0.9772	0.9792	0.9960
The Proposed MARS Model	0.978	0.9813	0.9947
The proposed DNN model	0.98	0.9843	0.9938
SVM model proposed in [19]	0.9809	0.9482	0.8369
RF model proposed in [19]	0.9809	0.9406	0.8445
DNN model proposed in [19]	0.9803	0.9803	0.9872
Best CR resampling ETC in [4]	0.9810	0.8870	0.8940
Best SMOTE resampling ETC in [4]	0.9780	0.8770	0.8890
Best ADASYN resampling ETC in [4]	0.9720	0.8110	0.9060

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

Best CC resampling ETC in [4]	0.9590	0.720	0.9140
LSTM model in [4]	0.98	0.96	0,91
DNN model in [4]	0.98	0.96	0.90
GRU model in [4]	0.,98	0.95	0.92

Clearly, as depicted in Table VII, the proposed models achieve similar results as the current state-of-the-art models for the task of identifying pulsar stars

VI. DISCUSSIONS

Based on the results of the study, all the three models perform well in terms of accuracy. The MARS model has an accuracy of 0.978, the K Nearest Model has an accuracy of 0.9772, and the Neural Network Model has an accuracy of 0.98. These values are all above the No Information Rate, which is 0.9082, and have a p-value less than 2.2e-16, indicating that the models have a significant accuracy compared to the No Information Rate.

In terms of the evaluation metrics, the Neural Network Model has the best results. It has the highest Kappa value (0.8749) and balanced accuracy (0.9187), which indicates that it has the best overall performance in terms of both accuracy and class imbalance. It also has the highest sensitivity (0.9938) and specificity (0.8435), which shows that it is able to accurately identify both pulsar and non-pulsar stars.

The MARS model has a slightly lower balanced accuracy (0.9039) and specificity (0.8130) compared to the Neural Network Model, but it has a higher sensitivity (0.9947). The KNN Model has the lowest balanced accuracy (0.8937) and specificity (0.7913), but it has the highest sensitivity (0.9960).

It is to be noted that accuracy is not the only measure of a model's performance. Other factors, such as training time, prediction time, interpretability, ability to generalize to new data, overfitting, etc. also play an important role in determining the best model for a particular problem. For instance, if the data is complex and the goal is to build a model that can generalize well to unseen data, a more complex model like a neural network may perform better than a simpler model like KNN. On the other hand, if interpretability is important, then a simpler model like KNN may be a better choice.

In the case of pulsar star classification, the specific requirements could include:

Accuracy: Pulsar star classification is an important task in astrophysics, and high accuracy is usually desired to minimize the number of false positive or false negative classifications.

Computational Efficiency: The data size can be large, and the model must be able to handle this efficiently, both in terms of training time and prediction time.

Interpretability: Understanding how the model is making its predictions is important for domain experts, as they may need to understand the reasons for a particular classification.

Ability to handle non-linear relationships: Pulsar star data can have complex relationships between variables, and the model must be able to handle these relationships effectively.

Robustness to Outliers: Pulsar star data can contain outliers, and the model must be robust to these to avoid being influenced by noisy or irrelevant data.

Generalization: The model must generalize well to new unseen data, as the goal is to classify pulsar stars that have not been seen before.

Overfitting: Overfitting can lead to poor performance on unseen data, so the model must be designed to avoid overfitting.

These requirements can be used to guide the choice of a model for pulsar star classification, and different models may have varying strengths and weaknesses in satisfying these requirements.

VII. CONTRIBUTION

This research contributes to the field in the following ways:

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

1. Comparative Analysis: The research provides a comparative analysis of three different models - MARS, KNN, and Neural Network - in the context of pulsar star classification. It not only compares the accuracy of these models but also other important metrics like balanced accuracy, specificity, and sensitivity.
2. Interpretability and Model Selection: It highlights the importance of factors other than accuracy, such as interpretability, computational efficiency, and ability to generalize, in selecting the best model for a specific task. This underlines the fact that the best model depends on the specific requirements of the task.
3. Empirical Evidence: The research provides empirical evidence on the performance of these three models on a real-world task - pulsar star classification. The results can serve as a guideline for other researchers and practitioners working on similar tasks.
4. Feature Importance: The research identifies the feature - "Excess Kurtosis of the Integrated Profile" - as having the most impact on the classification task. This insight can be beneficial for future studies involving similar datasets or tasks.
5. Potential Application: Given the high accuracy of the three models, the research presents potential practical applications in astrophysics for the classification of pulsar stars, which could assist in future space exploration and understanding of celestial bodies.

VIII. CONCLUSION AND FUTURE SCOPE

The research assessed various methods of pulsar detection and star classification, revealing strengths and limitations. The handling of imbalanced datasets, like HTRU2, and difficulty in managing large volumes of observational data are common challenges. Of the models assessed, the Neural Network Model emerged as the most accurate (98%), closely followed by MARS (97.8%) and KNN Model (97.7%). Each has its strengths and selection should be guided by the requirements and priorities of the task, including sensitivity, specificity, interpretability, and computational efficiency. Future research should aim to address dataset imbalance. Optimizing oversampling techniques and synthetic data creation are promising avenues. Furthermore, advanced data processing techniques that can handle vast quantities of data from pulsar detecting telescopes need to be developed. Extremely Fast Decision Tree is one potential solution, but its handling of imbalanced data streams needs improvement. Lastly, deep learning, particularly convolutional neural networks, offer exciting potential for feature extraction and classification. Optimizing these models could significantly improve pulsar detection's speed and accuracy. In conclusion, while significant advancements have been made in pulsar detection and star classification, there remain numerous opportunities for further innovation and exploration. The field holds the promise of new and advanced machine learning models and techniques, which could revolutionize our cosmic understanding

REFERENCES

[1] D. G. York, ‘The Sloan Digital Sky Survey: Technical Summary’, Astron. J., vol. 120, no. 3, pp. 1579–1587, Sep. 2000, doi: 10.1086/301513.
[2] M. J. Keith et al., ‘The High Time Resolution Universe Pulsar Survey I: System configuration and initial discoveries’, Mon. Not. R. Astron. Soc., vol. 409, no. 2, pp. 619–627, Dec. 2010, doi: 10.1111/j.1365-2966.2010.17325.x.
[3] S. D. Bates et al., ‘The High Time Resolution Universe Pulsar Survey — VI. An artificial neural network and timing of 75 pulsars’, Mon. Not. R. Astron. Soc., vol. 427, no. 2, pp. 1052–1065, Dec. 2012, doi: 10.1111/j.1365-2966.2012.22042.x.
[4] E. Lee, F. Rustam, W. Aljedaani, A. Ishaq, V. Rupapara, and I. Ashraf, ‘Predicting Pulsars from Imbalanced Dataset with Hybrid Resampling Approach’, Adv. Astron., vol. 2021, p. e4916494, Dec. 2021, doi: 10.1155/2021/4916494.
[5] S. Andrianomena, ‘Probabilistic learning for pulsar classification’, J. Cosmol. Astropart. Phys., vol. 2022, no. 10, p. 016, Oct. 2022, doi: 10.1088/1475-7516/2022/10/016.
[6] Z. L. De Beurs, N. Islam, G. Gopalan, and S. D. Vrtilek, ‘A Comparative Study of Machine-learning Methods for X-Ray Binary Classification’, Astrophys. J., vol. 933, no. 1, p. 116, Jul. 2022, doi: 10.3847/1538-4357/ac6184.
[7] Y. Wang, Z. Pan, J. Zheng, L. Qian, and M. Li, ‘A Hybrid Ensemble method for Pulsar Candidate Classification’, Astrophys. Space Sci., vol. 364, no. 8, p. 139, Aug. 2019, doi: 10.1007/s10509-019-3602-4.

IJARSCT
ISSN (Online) 2581-9429
International Journal of Advanced Research in Science, Communication and Technology (IJARSCT)
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
Impact Factor: 7.301
Volume 3, Issue 2, August 2023

[8] J. Holewik, G. Schaefer, and I. Korovin, 'Imbalanced Ensemble Learning for Enhanced Pulsar Identification', in Advances in Swarm Intelligence, Y. Tan, Y. Shi, and M. Tuba, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2020, pp. 515–524. doi: 10.1007/978-3-030-53956-6_47.
[9] M. Azhari, A. Alaoui, A. Abarda, B. Ettaki, and J. Zerouaoui, 'Using Ensemble Methods to Solve the Problem of Pulsar Search', in Big Data and Networks Technologies, Y. Farhaoui, Ed., in Lecture Notes in Networks and Systems. Cham: Springer International Publishing, 2020, pp. 183–189. doi: 10.1007/978-3-030-23672-4_14.
[10] L. Wang, L. Luo, and S. Zhang, 'A New Method of Pulsar Candidate Detection Based on Bagging', in 2021 China Automation Congress (CAC), Oct. 2021, pp. 5959–5962. doi: 10.1109/CAC53003.2021.9728494.
[11] L. Xiang, Z. Zhou, L. Miao, and Q. Chen, 'Signal Recognition Method of X-ray Pulsar Based on CNN and Attention Module CBAM', in 2021 33rd Chinese Control and Decision Conference (CCDC), May 2021, pp. 5436–5441. doi: 10.1109/CCDC52312.2021.9602569.
[12] J. Talwar, P. Jain, C. Jain, and B. Kaur, 'Stellar and Pulsar Classification using Machine Learning', vol. 1, no. 4, 2021.
[13] D. Beniwal, A. Roy, H. Yadav, and A. Chauhan, 'Detection of Pulsars by Classical Machine Learning Algorithms', in 2021 2nd International Conference for Emerging Technology (INCET), May 2021, pp. 1–7. doi: 10.1109/INCET51464.2021.9456250.
[14] V. Gamage, M. Ayoob, and K. Jayakumar, 'Pulsar Candidate Selection Using Gaussian Hellinger Extremely Fast Decision Tree', in 2022 2nd International Conference on Image Processing and Robotics (ICIPRob), Mar. 2022, pp. 1–7. doi: 10.1109/ICIPRob54042.2022.9798721.
[15] Z.-Y. Xiong et al., 'A neighborhood weighted-based method for the detection of outliers', Appl. Intell., vol. 53, no. 9, pp. 9897–9915, May 2023, doi: 10.1007/s10489-022-03258-0.
[16] J. H. Friedman, 'Multivariate Adaptive Regression Splines', Ann. Stat., vol. 19, no. 1, pp. 1–67, Mar. 1991, doi: 10.1214/aos/1176347963.
[17] E. Fix and J. L. Hodges, 'Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties', Int. Stat. Rev. Rev. Int. Stat., vol. 57, no. 3, pp. 238–247, 1989, doi: 10.2307/1403797.
[18] T. Cover and P. Hart, 'Nearest neighbor pattern classification', IEEE Trans. Inf. Theory, vol. 13, no. 1, pp. 21–27, Jan. 1967, doi: 10.1109/TIT.1967.1053964.
[19] A. Punia, A. Sardana, and M. Subashini, 'Evaluating advanced machine learning techniques for pulsar detection from HTRU survey', in 2017 International Conference on Intelligent Sustainable Systems (ICISS), Dec. 2017, pp. 470–474. doi: 10.1109/ISS1.2017.8389455.

APPENDIX

GitHub repository: https://github.com/devika1402/Identification-of-Pulsar-star