EPJ Web of Conferences **343**, 05010 (2025) https://doi.org/10.1051/epjconf/202534305010
AIMACE-2025

Machine Learning in Astrophysics: A Multi-Algorithm Comparison for Pulsar Prediction

Shania Soares[1*], *Archana* Pandita[2]

[1,2]Amity University Dubai, UAE

Abstract. Pulsar stars were discovered over 50 years ago. Known for their radiation pulses observed at regular intervals from planet Earth, they are a type of neutron star that possesses extremely powerful magnetic fields which are a result of the collapsed cores of the stars. Prediction of pulsar stars can be beneficial to scientists as they continue to uncover the mysteries relating to these exemplary celestial bodies. The purpose of this study is to compare the performance of various machine learning models to determine which one of them is the most capable of predicting pulsar stars. The models were built on the algorithms of Support Vector Machine (SVM), Random Forest and Naïve Bayes. The dataset used in this project is the widely known HTRU_2 dataset. The assessment of the models was done using evaluation metrics like the F1-score, recall, classification accuracy and precision. The metrics provided a comprehensive understanding of each model's performance. The training and prediction times were also investigated to evaluate the scalability of the models. Of the models developed, Random Forest soared as the highest performer in terms of accuracy with 98.32%. Following closely behind was SVM with 97.56%, a result achieved when combined with the RBF kernel. The Naïve Bayes model, though it was faster than the other two models, attained an accuracy of 95.42%, as the classification performance was slightly on the lower side. The concluding study aims to encapsulate the performance of each model in terms of its strengths and weaknesses, thereby determining its potential suitability for pulsar prediction.

1 Introduction

In astrophysics, some of the most studied concepts are those related to pulsar stars. These fascinating celestial bodies are the result of a supernova explosion that occurs when a neutron star meets its eventual end. The first sighting of its kind was encountered in 1967 by the Mullard Radio Astronomy Observatory [1] from a specific point in the sky. PSR B1919+21 was the name assigned to it; informally, it also went by LGM-1, which stood for "Little Green Men One" [2]. Over the years, there have been multiple pulsar star instances that were identified; however, it is believed that these are just a small percentage of the actual number of pulsar stars that exist in our galaxy. Although the pulsar signals tend to be highly regular, it becomes a challenge to detect them due to certain interferences from Earth-based radio signals and distortions that are caused by the interstellar medium. Delving into the topic of pulsar stars proves to be advantageous to furthering studies regarding general relativity, gravitational waves and exploring the interstellar medium. The rapid spinning of the stars, coupled with the stable rotational periods, equips them to be used as natural cosmic clocks that could be used for precise timekeeping. Scientists are able to dig deep into the theories about life and death in the cosmos. Detecting pulsar signals for classification tasks was

* Corresponding author: apandita@amityuniversity.ae

initially done using complex manual analysis of the signal's features. This was often a tedious and time-consuming task that was prone to errors. Recently, however, advancements have been made using various machine learning techniques to improve the accuracy of tracking the pulsar signals. This study will aim to showcase how machine learning algorithms such as Support Vector Machine (SVM), Random Forest and Naïve Bayes can assist with the task of classifying pulsar candidates from enormous datasets. Thereby contributing to the broad field of astrophysics by demonstrating the ability of modern technology to simplify tasks that were earlier deemed strenuous. Whether or not these models are feasible is determined by assessing the models based on their evaluation metrics. These can provide a deeper understanding of each model's performance. The dataset used for the models also plays a role in the outcomes achieved. In this case, the HTRU_2 dataset was implemented with all three models along with various data-preprocessing steps that were applied to it. Further advancements in pulsar classification can lead to modifications in fine-tuning models of stellar evolution as well as detecting ultra-high-energy cosmic rays, which are of particular interest to scientists. Overall, it intends to enhance deep-space exploration technologies by utilizing modern machine learning technologies.

2 Literature Review

The study of pulsar stars has been an area of significant interest in astrophysics, with research spanning decades to improve detection and classification methods. Multiple methods have been evaluated to classify pulsar signals from the array of signals captured on the radio telescope. Initially, manual methods were used to classify them based on the feature characteristics. However, ever since the evolution of technology, various machine-learning techniques have been implemented to perform this task. Random Forest [3] has proven effective in handling complex patterns in pulsar data, as it can differentiate between real pulsars and noise-induced signals. Lately, various deep learning approaches such as Convolutional Neural Networks (CNNs) have been explored for feature extraction and classification, outperforming traditional methods in accuracy [4]. Despite these advancements, pulsar classification still faces challenges such as class imbalance, where pulsars are significantly outnumbered by non-pulsar signals. Techniques like Synthetic Minority Over-sampling (SMOTE) [5] and anomaly detection models have been proposed to address this issue. Additionally, integrating ML models with real-time data processing in modern radio telescopes, like the Square Kilometre Array (SKA), is a key area of ongoing research [6]. Table 1 depicts a comprehensive list of studies that have been made based on Pulsar Star predictions and classifications using various Machine Learning Algorithms. It highlights the methodologies, limitations, shortcomings and conclusions from the research papers published that applied algorithms like SVM, Naïve Bayes, Random Forest, Logistic Regression and other machine learning models. The common limitations among them include challenges relating to imbalanced datasets, noisy data and computational complexity. Regardless of these challenges, the studies indicate that the use of various machine learning models has shown substantial outcomes in pulsar prediction in various cases. The research collectively emphasizes the importance of careful feature selection, hyperparameter tuning and data preprocessing which is imperative to improving the model's overall performance. While the majority of the models show significant results, they also come across several trade-offs and require fine-tuning to achieve optimal results.

Table 1. List of Studies on Pulsar Prediction using Machine Learning Algorithms

Methodology	Limitations	Shortcomings	Conclusion	References
SVM, Naive Bayes, Adaptive Boosting, Decision Tree, KNN, Random Forest	May struggle with imbalanced datasets without additional techniques like SMOTE.	Careful tuning of parameters is required for all algorithms.	Comparative analysis revealed that XGBoosting performed best, but all models showed strengths in different areas of classification.	[7]
Decision Tree, SVM, Random Forest, KNN	Limited by dataset quality, with noise and missing data posing challenges for feature extraction.	Performance degrades with highly imbalanced datasets; difficulty in feature selection.	Machine learning algorithms showed significant promise for detecting pulsar stars, though more data preprocessing is needed.	[8]
Real-Time Statistical, Data Analysis	Needs real-time data, which can be challenging due to data delays; computationally demanding on large datasets.	Algorithm complexity increases with the number of pulsar candidates; may require high-performance computing infrastructure.	New principled real-time approach offered improved accuracy over older methods, particularly for large data streams.	[9]
Naïve Bayes, Logistic Regression, SVM, Decision Tree, Perceptron	Can be computationally expensive; prone to overfitting without proper feature selection and regularization.	It requires careful balancing of preprocessing and model selection steps.	Provides a framework for applying machine learning models to pulsar star classification with potential for scalability.	[10]
Random Forest. Naïve Bayes, Multi-layer Perceptron, SVM	May not work well with high-dimensional data unless proper feature selection is applied.	Overfitting may occur if the number of trees is too high or too low.	Random Forest can handle spectral data well, though tuning of parameters is essential for optimal results.	[11]
SVM, Naïve Bayes, Random Forest, and	SVM and Naïve Bayes may perform sub optimally on time-series or	Each model has strengths, but their performance is highly	Random Forest and SVM performed best in terms of	[12]

LSTM Models	sequential data like pulsar data unless modified. Random Forest can be computationally expensive.	dependent on the feature engineering and hyperparameter tuning.	predictive power, but other algorithms like Naïve Bayes and LSTM also had significant use cases.	

3 Materials and Methods

The dataset for this project was retrieved from the University of California, Irvine (UCI) Machine Learning Repository. Named 'HTRU_2', this dataset has been widely used for tasks related to pulsar star classification. The features present in the dataset are extracted from the observations of radio signals collected by the High Time Resolution Universe Survey (South) (HTRU). These focus on the specific characteristics of the signals, such as signal-to-noise ratio, skewness, and kurtosis, which are considered essential in distinguishing pulsar signals from non-pulsar signals. Based on a binary classification concept, it includes a mix of positive (pulsar) and negative (non-pulsar) samples, where the main aim is to predict whether the features of the signal classify a particular signal as that from a pulsar star or not. The positive samples represent an actual pulsar, while the negative sample is identified as a non-pulsar. Based on this, the class labels are 0 for non-pulsar and 1 for pulsar signals. The attributes of the signals are classified on the following – Excess kurtosis of the integrated profile; Excess kurtosis of the DM-SNR curve; Standard deviation of the DM-SNR curve; Standard deviation of the integrated profile; Mean of the DM-SNR curve; Mean of the integrated profile; Skewness of the integrated profile; Skewness of the DM-SNR curve [10]. Table 2 shows a statistical description of the dataset.

Table 2. Statistical Description of the Dataset

	Mean of the integrated profile	**Mean of the DM-SNR curve**	**Excess kurtosis of the integrated profile**	**Excess kurtosis of the DM-SNR curve**	**Skewness of the integrated profile**	**Skewness of the DM-SNR curve**	**Standard deviation of the integrated profile**	**Standard deviation of the DM-SNR curve**	**target_class**
count	17898	17898	17898	17898	17898	17898	17898	17898	17898
mean	111.0800	12.614	0.478	8.303	1.770	104.858	46.549	26.326	0.091
std	25.653	29.473	1.064	4.506	6.168	106.515	6.843	19.470	0.288
min	5.812	0.213	-1.876	-3.140	-1.792	-1.977	24.772	7.370	0
25%	100.930	1.923	0.027	5.782	-0.189	34.961	42.376	14.437	0
50%	115.078	2.802	0.223	8.434	0.199	83.065	46.947	18.461	0
75%	127.0860	5.464	0.473	10.703	0.928	139.309	51.023	28.428	0
max	192.617	223.392	8.069	34.540	68.102	1191.000	98.779	110.642	1

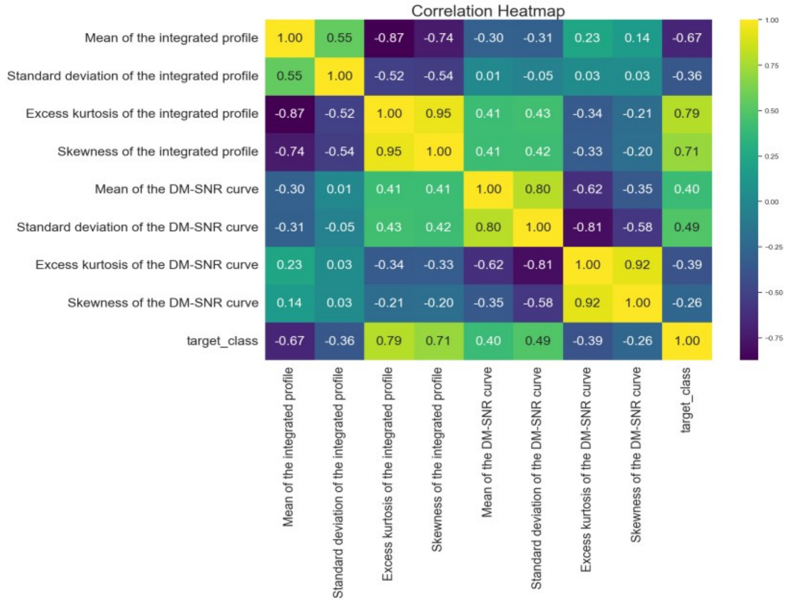

Fig. 1. Heatmap showing the Correlation between the Features of the Pulsar Dataset

Figure 1 visually depicts the correlation between the features using a heatmap. There are quite a few high correlations both negative and positive. However, since there are only 8 features in the entire dataset, all of them have been used. The highest positive correlation is 0.95 between the features - Excess kurtosis of the integrated profile and Skewness of the integrated profile. Meanwhile, the highest negative correlation is between - Mean of the integrated profile and the Excess kurtosis of the integrated profile features with a value of -0.87.

4 Data Preprocessing

The dataset had no null or missing values. However, it contained a considerable number of outliers, as seen in Figure 2. This was dealt with by using the soft-margin variant of SVM. Particularly in this case, a few outliers can be classified incorrectly or be classified with a margin less than 1. This tradeoff is controlled using the C parameter with a Low C value of 10 which is much more lenient and prioritizes finding a more general decision boundary. Although this results in a slightly higher training error, this is beneficial as it helps the model to prevent overfitting and adapt better to unseen data.

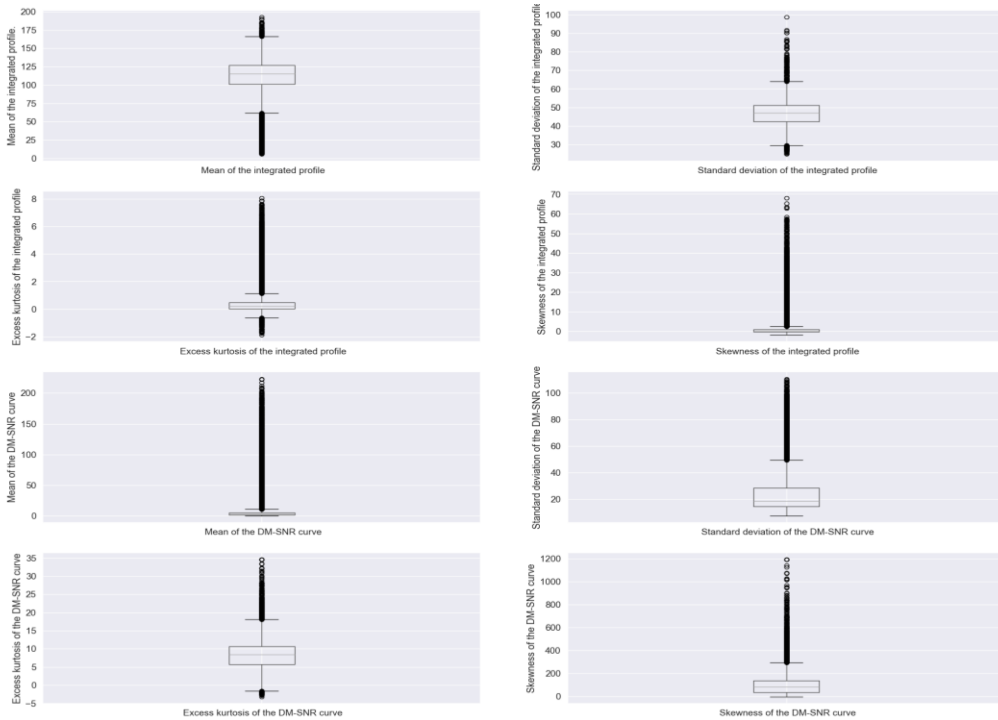

Fig. 2. Boxplot charts showing the outliers in the dataset

From the dataset, the column labelled ‘target_class’ was extracted as the output label which the model will predict, while the remaining features were used to train the model. The dataset was then split into testing and training sets, with 20% of the data being allocated for testing and the remaining 80% for training. Thus, out of 17,898 records, 14,318 were for training and 3,580 were for testing. There was a very apparent class imbalance that was noticed. The number of positive pulsar samples were way less when compared to the instances of the negative pulsar samples. This imbalance was addressed in the training set samples by using the Synthetic Minority Oversampling Technique (SMOTE). Here, new synthetic samples are created within the existing samples in the minority sample class (positive pulsars). This method picks up a point between an existing sample in the minority class and their corresponding K-Nearest Neighbour which is found using a distance metric such as the Euclidean distance metric. These act as reference points for the new data. A new data point is then created at this specific point which interpolates the values of the features of the two reference points [13]. Therefore, before the number of samples in the negative pulsar class was 12,953 and in the positive pulsar class was 1,365. After applying the SMOTE resampling technique both classes now have the same number of samples which is 12,953. In this manner, the problem of class imbalance is solved. This method was used for the SVM model. The features of the pulsar dataset had to be similarly scaled as they all had varying scales. Hence, feature scaling was done using the Standardization method to shrink the range of the features so that they are closer together, thereby ensuring that the model does not have a bias towards bigger numbers of less importance during the training process [14]. At this stage, the data was ready to be trained on the SVM, Random Forest and Naïve Bayes Models. The programming language used for this project is Python. Several Python libraries were also imported to support the tasks that were implemented. Pandas and NumPy were used for data analysis, Matplotlib for data visualization, time for tracking the execution time of the models and the sklearn library mainly handled all tasks related to machine learning.

5 Results and Discussion

5.1 Support Vector Machine (SVM)

They are supervised machine learning algorithms. In this case, a hyperplane is drawn while taking all the data points into consideration. The hyperplane is a line that can segregate the data points into separate classes, as seen in Figure 3. The distance of the line to the nearest element of each class should be the largest. In cases where the data is not linearly separable, the data is plotted into higher dimensions so that they can be easily separated. SVM has four kernel functions that can be employed, namely – Linear, Polynomial, Radial Basis Function (RBF) and Sigmoid kernels. The linear kernel creates a straight decision boundary, whereas the polynomial kernel allows for non-linear decision boundaries by projecting data into a space that is of higher dimension. The RBF kernel is widely used for its ability to handle data that is non-linearly separable by mapping the feature space into a higher dimension, while the sigmoid kernel, based on a hyperbolic tangent function, is less commonly used but still valuable in certain scenarios [11].

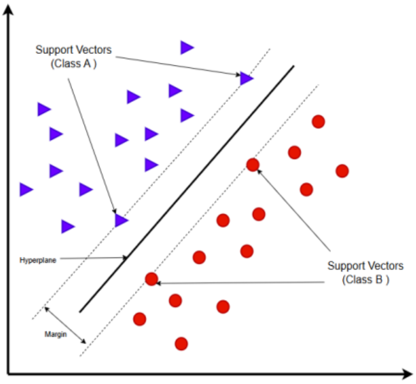

Fig. 3. Architecture of SVM

These different kernels provide flexibility in learning from the pulsar dataset, as they can adapt to various types of data distributions. The Support Vector Machine (SVM) model achieved quite a high accuracy. All four kernel functions were applied to the data to make a comparison as seen in Table 3. Among these, the Radial Basis Function (RBF) performed the best, with an accuracy of 97.65%, thereby proving to be the most effective kernel as it can handle complex decision boundaries.

Table 3. Accuracy Results of SVM Kernel Functions

Kernel Type	Accuracy
Linear	0.9749
Polynomial	0.9729
RBF (Radial Basis Function)	0.9765
Sigmoid	0.8763

Following closely behind were the Linear and Polynomial kernels, with an accuracy of 97.49% and 97.29% respectively, which is slightly lower than RBF but still shows competitive performance. The Sigmoid kernel depicted the lowest accuracy at 87.63%, indicating that it struggles to model the relationships within the data in an effective manner when compared to the other kernels. Using the RBF kernel, the training time taken was 2.7633 seconds and the prediction time was 0.6667 seconds. Since RBF performed the best, it was chosen for the final comparison between all the algorithms.

Table 4. Confusion Matrix for SVM Model using RBF

Actual/Predicted	**Non-Pulsar (Class 0)**	**Pulsar (Class 1)**
Non-Pulsar (Class 0)	3244 (TN)	62 (FP)
Pulsar (Class 1)	22 (FN)	252 (TP)

Table 4 shows the confusion matrix for the SVM model using the Radial Basis Function (RBF) kernel. True Negatives (TN) are 3244, meaning the model correctly identified 3244 non-pulsars. False Positives (FP) are 62, where non-pulsars were misclassified as pulsars. False Negatives (FN) are 22, where pulsars were misclassified as non-pulsars. True Positives (TP) are 252, indicating the model correctly identified pulsars.

5.2 Random Forest

Random Forest is a supervised machine learning algorithm that relies on multiple decision trees with the intention to achieve a final prediction. A random subset of the data is used to train each tree, and the outputs of all the trees are then combined to generate the final prediction.. It is capable of handling class imbalance quite well, particularly when combined with techniques such as assigning class weights or oversampling the minority class. The output of all trees is then processed in the majority voting system to derive the final decision. This is considered to be a robust system because apart from using a random subset of the data, it also uses a random selection of features for splitting the nodes. This randomness ensures that the trees are less correlated with each other, thereby reducing the risk of overfitting and helping the model perform better [15]. There were specific parameters that were tuned for the Random Forest model. The number of decision trees used was 200, in order to maintain a balance between performance and speed. Each tree's depth was limited to 15 to prevent overfitting and capturing the noise in the data. Because we are dealing with an imbalanced dataset, the class weight was set to 'balanced' to prohibit the model from ignoring the minority classes. For considering the number of features to be used for the best split, the square root of the total number of features was used as this works well for high-dimensional data.

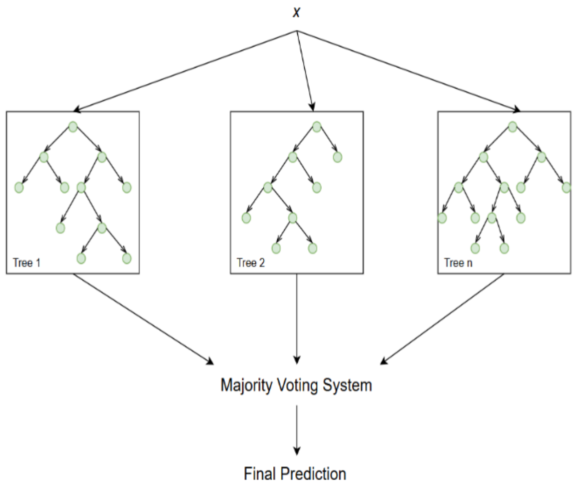

Fig. 4. Architecture of Random Forest

The Random Forest Algorithm received an accuracy of 98.32% which is fairly better than what was achieved with the SVM model. This substantial level of accuracy highlights the algorithm's strength in handling binary classification tasks. The model took 4.3113 seconds for training and 0.0481 seconds for prediction.

Table 5. Confusion Matrix for Random Forest Model

Actual/Predicted	**Non-Pulsar (Class 0)**	**Pulsar (Class 1)**
Non-Pulsar (Class 0)	3280 (TN)	26 (FP)
Pulsar (Class 1)	34 (FN)	240 (TP)

The confusion matrix further clarifies these metrics, as shown in Table 5, 3280 True Negatives (TN) were correctly identified as non-pulsars. For False Positives (FP), 26 non-pulsars were incorrectly classified as pulsars. False Negatives (FN) are 34, indicating that 34 pulsars of them were misclassified as non-pulsars. True Positives (TP) are 240, meaning the model successfully classified them as pulsars. These low misclassification rates highlight the robustness of the Random Forest model.

5.3 Naïve Bayes

Naïve Bayes is another supervised machine learning technique whose foundation is derived from the Bayes' theorem as seen in Figure 5. The assumption here is that the features are independent of each other, thereby giving it the 'naïve' name. It is often used in problems like text classification, where the features are conditionally independent [11].

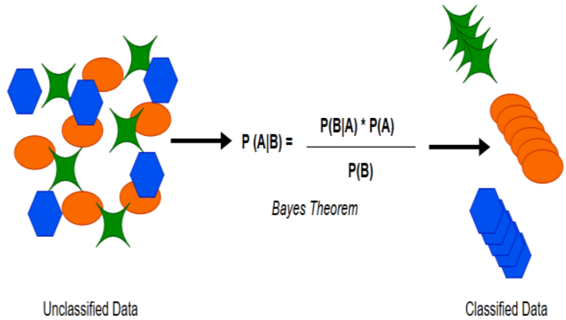

Fig. 5. Architecture of Naïve Bayes

Since the pulsar stars dataset has continuous features, some are assumed to follow a normal distribution. Therefore, a Gaussian Naïve Bayes (GNB) classifier has been used. The 'var_smoothing' parameter was applied to add a small value to the variance during computation to prevent division by zero or extremely small variances that could lead to numerical instability. The Naive Bayes classifier yielded an accuracy of 95.42%. This result demonstrates its capability to classify pulsar and non-pulsar stars but falls short when compared to more complex models like Random Forest and SVM.

Table 6. Confusion Matrix for Naïve Bayes Model

Actual/Predicted	Non-Pulsar (Class 0)	Pulsar (Class 1)
Non-Pulsar (Class 0)	3178 (TN)	128 (FP)
Pulsar (Class 1)	36 (FN)	238 (TP)

The confusion matrix for the Naïve Bayes model, as seen in Table 6, indicates that the model correctly classified 3178 instances as non-pulsars (True Negatives) and 238 instances as pulsars (True Positives). However, it misclassified 128 non-pulsars as pulsars (False Positives) and 36 pulsars as non-pulsars (False Negatives). This demonstrates that the model has a strong ability to correctly identify non-pulsars while maintaining a reasonable accuracy ratio, although some misclassifications are apparent, particularly in distinguishing the true pulsars. Regarding computational performance, the training time was extremely fast at 0.0050 seconds, while the prediction time was at 0.0020 seconds, demonstrating the efficiency of Naive Bayes in real-time predictions.
In Figure 6, the graph illustrates the performance comparison of the three machine learning models SVM, Random Forest, and Naive Bayes based on their classification accuracy. Random Forest achieved the highest accuracy at 98.32%, closely followed by SVM, with the RBF kernel at 97.65%, while Naive Bayes had the lowest accuracy among them at 95.42%. Table 7 displays the comparison between the three machine learning algorithms - Support Vector Machine (SVM), Random Forest, and Naive Bayes based on their performance on the HTRU_2 dataset using various evaluation metrics and several key observations emerge. The Random Forest model slightly outperforms SVM in accuracy (98.32% vs 97.65%), while the Naïve Bayes model lags behind at 95.42%.

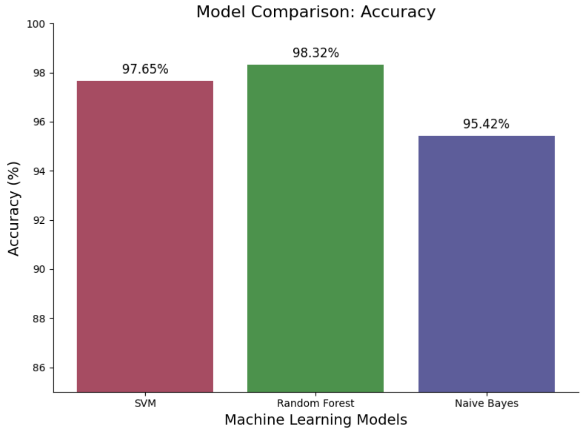

Fig. 6. Comparison of Accuracy achieved with each model

Table 7. Comparison across SVM, Random Forest and Naive Bayes based on Evaluation Metrics

Evaluation Metric	Class	SVM (RBF)	Random Forest	Naïve Bayes
Accuracy		97.65%	98.32%	95.42%
Training Time (secs)		2.7633	4.3113	0.0050
Prediction Time (secs)		0.6667	0.0481	0.0020
Precision	0	0.99	0.99	0.99
	1	0.80	0.90	0.65
Recall	0	0.98	0.99	0.96
	1	0.92	0.88	0.87
F1 - Score	0	0.99	0.99	0.97
	1	0.86	0.89	0.74

When it comes to training times, Naïve Bayes produced the fastest results requiring only 0.0050 seconds while Random Forest despite having a higher accuracy, took the longest time at 4.3113 seconds. As far as precision is concerned, Random Forest performs the best for pulsars (Class 1) at 0.90, on the other hand, Naïve Bayes seemed to struggle with a much lower precision of 0.65. For recall, SVM has a higher value for pulsars (0.92), although Random Forest and Naïve Bayes perform quite similarly in that aspect with values of 0.88 and 0.87 respectively. The F1-Score for non-pulsars (Class 0) is consistently high across all models with SVM and Random Forest achieving 0.99, while Naïve Bayes is a bit lower at 0.97. For pulsars (Class 1), however, Random Forest (0.89) and SVM (0.86) outperform Naïve Bayes (0.74) thereby indicating that Naïve Bayes struggles more with the minority class. The results of the Naïve Bayes model depict that it has a strong performance when dealing with non-pulsars (Class 0), however, it fails to maintain that performance when it comes to pulsars (Class 1). Nevertheless, its simplicity and speed make it a viable option for exploratory analysis or real-time applications where high precision is not a strict requirement.

6 Conclusion and Future Work

The Random Forest model proved to be highly competitive as it demonstrated the best performance among all the models in terms of accuracy and classification metrics, thereby making it the most suitable choice for this dataset. Its robustness and the ability to prevent overfitting contribute to its superior performance which ensures high reliability in identifying both classes. The model achieves strong precision and recall scores, particularly for the minority class, making it well-suited for the imbalanced datasets. The computational power and time for this model are on the higher side as it utilises multiple trees to combine their outputs [16]. It is a strong contender to be used in situations where the main goal is to maximise the overall accuracy as it excels at drawing clear decision boundaries between classes and can perform exceptionally well in situations where precision for a particular class like pulsars, in this case, is substantial. The time complexity for Random Forest is O(T.nlogn) where (T) is the number of trees and (n log n) is the complexity of each tree. Despite this, it is the best-performing model in this study as the overall classification performance of the Random Forest model outweighs the computational trade-offs.

The Support Vector Machine (SVM) model along with the RBF kernel also performed well but was slightly outperformed by the Random Forest model. The data preprocessing techniques such as scaling the features using Standardization and utilising the SMOTE method for resampling the dataset to address the class imbalance issue have hugely contributed to this result. The main drawback of using SVM with the RBF kernel is that it becomes computationally expensive for larger datasets due to its $O(n^3)$ complexity.[17] This issue could be partially addressed by using Principal Component Analysis (PCA) to minimise the dimensionality of the dataset, which might result in improvements in the training speed and efficiency.

Meanwhile, Naïve Bayes turned out to be efficient but not as accurate. Its main drawback in this case was the significantly lower precision for pulsars, indicating that it incorrectly classified a huge number of pulsar instances. In situations where speed is the topmost priority and there is not much emphasis placed on achieving the best precision for pulsar classification, it could definitely be beneficial. Although further tuning and enhancements would be required for the model. The time complexity for Naive Bayes is O(Np) wherein (N) represents the number of training samples and (p) is the number of features. [45] Therefore, even when the dataset is large, it remains efficient and provides fast training since the complexity is linear.

Future efforts could focus on building different models with suited parameters to achieve even better results. Some of these include Deep Neural Networks (DNN), Logistic Regression, Gradient Boosting Machines (GBM), etc. There has been use of Convolutional Neural Networks (CNN) to aid in predicting pulsar stars and they tend to outperform traditional machine learning models. As seen in [17] where the CNN model built to predict the pulsars has achieved accuracies higher between 97-98%. This was the result when tried with different layers and epochs. Impurities in the data can lead to poor model performance, hence it is essential to use appropriate data pre-processing procedures to enhance the models' efficacy. Various other methods and advanced feature engineering techniques can be tuned to further enhance predictive capabilities, especially in real-world scenarios with large-scale datasets.

References

1. Hewish, A., Bell, S. J., Pilkington, J. D. H., Scott, P. F., & Collins, R. A. (1968). Observation of a rapidly pulsating radio source. Nature, **217**, 709-713.

2. F. Rustam et al., “Predicting pulsar stars using a random tree boosting voting classifier (RTB-VC),” Astronomy and Computing, vol. **32**, p. 100404, (Jul. 2020), doi: https://doi.org/10.1016/j.ascom.2020.100404.
3. Breiman, L. (2001). Random forests. Machine Learning, 45, 5-32.
4. Zhu, W. W., Zhang, L., & Wang, P. (2019). Pulsar candidate classification based on convolutional neural networks. The Astrophysical Journal, 881(1), 20.
5. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research, 16, 321-357.
6. Keane, E. F., Bhattacharyya, B., Flynn, C., & Kramer, M. (2015). The future of pulsar astronomy with the Square Kilometre Array. Proceedings of the International Astronomical Union, **291**, 249-253.
7. A. Sadhu, “Pulsar Star Detection: A Comparative Analysis of Classification Algorithms using SMOTE,” International Journal of Computer and Information Technology(2279-0764), vol. **11**, no. 1, Mar. 2022, doi: https://doi.org/10.24203/ijcit.v11i1.193
8. P. Mounika, P. Siva Spandana, M. Anuja, N. Krishna, S. Prada, and U. Scholar, “DETECTION OF PULSAR STARS USING MACHINE LEARNING ALGORITHMS,” (Jun. 2020).
9. R. J. Lyon, B. W. Stappers, S. Cooper, J. M. Brooke, and J. D. Knowles, “Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach,” Monthly Notices of the Royal Astronomical Society, vol. **459**, no. 1, pp. 1104–1123, Apr. 2016, doi: https://doi.org/10.1093/mnras/stw656.
10. A. Ylnner, V. Sanchez, D. Carolina, Saire, Josimar Edinson Chire, and P. B. Franco, “Machine Learning Pipeline for Pulsar Star Dataset,” arXiv.org, (2020). https://arxiv.org/abs/2005.01208
11. José-Luis Solorio-Ramírez, Raúl Jiménez-Cruz, Yenny Villuendas-Rey, and Cornelio Yáñez-Márquez, “Random forest Algorithm for the Classification of Spectral Data of Astronomical Objects,” Algorithms, vol. **16**, no. 6, pp. 293–293, Jun. 2023, doi: https://doi.org/10.3390/a16060293
12. S. Pathak, S. J. Quraishi, A. Singh, M. Singh, K. Arora and D. Ather, "A Comparative Analysis of Machine Learning Models: SVM, Naïve Bayes, Random Forest, and LSTM in Predictive Analytics," 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS), Tashkent, Uzbekistan, (2023), pp. 790-795, doi: https://doi.org/10.1109/ICTACS59847.2023.10390255
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, “SMOTE: Synthetic Minority Over-sampling Technique,” Journal of Artificial Intelligence Research, vol. **16**, pp. 321–357, (Jun. 2002), doi: https://doi.org/10.1613/jair.953.
14. A. Ambarwari, Q. Jafar Adrian, and Y. Herdiyeni, “Analysis of the Effect of Data Scaling on the Performance of the Machine Learning Algorithm for Plant Identification,” Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. **4**, no. 1, pp. 117–122, (Feb. 2020), doi: https://doi.org/10.29207/resti.v4i1.1517
15. Ankhita Ravishankar Sathanur, “Random Forest Identification of Pulsars,” International journal of high school research, vol. **5**, no. 4, pp. 14–22, (Aug. 2023), doi: https://doi.org/10.36838/v5i4.3.
16. Akshay Gole, Sankalp Singh, Prathmesh Kanherkar, P.R.Abhishek, and Prof . Pallavi Wankhede, “Comparative Analysis of Machine Learning Algorithms : Random Forest algorithm, Naive Bayes Classifier and KNN - A survey”, JRPS, vol. **13**, no. 3, pp. 194–197, (Apr. 2022).
17. J. R. Song, “The effectiveness of different machine learning algorithms in classifying pulsar stars and the impact of data preparation,” Journal of Physics Conference Series, vol. **2428**, no. 1, pp. 012046–012046, (Feb. 2023), doi: https://doi.org/10.1088/1742-6596/2428/1/012046.
18. E. Lee, F. Rustam, W. Aljedaani, A. Ishaq, V. Rupapara, and I. Ashraf, “Predicting Pulsars from Imbalanced Dataset with Hybrid Resampling Approach,” Advances in Astronomy, vol. **2021**, pp. 1–13, (Dec. 2021), doi: https://doi.org/10.1155/2021/4916494.
19. Juie Shetye, E. Verwichte, M. Stangalini, and J. G. Doyle, “Chandra Observations of Candidate Sub-Parsec Binary Supermassive Black Holes,” The Astrophysical Journal, vol. **866**:144 (5pp), (Jan. 2020), doi: https://doi.org/10.3847/1538-4357.
20. J. Holewik and G. Schaefer, “Ensemble Classifiers for Pulsar Detection,” Emergence, Complexity and Computation, pp. 153–166, 2021, doi: https://doi.org/10.1007/978-3-030-65867-0_7.

21. M. A. B. Mahmoud and P. Guo, “Learning from imbalanced pulsar data by combine DCGAN and PILAE algorithm,” New Astronomy, vol. **85**, pp. 101561–101561, (Dec. 2020), doi: https://doi.org/10.1016/j.newast.2020.101561.
22. S. Bethapudi and S. Desai, “Separation of pulsar signals from noise using supervised machine learning algorithms,” Astronomy and Computing, vol. **23**, pp. 15–26, (Apr. 2018), doi: https://doi.org/10.1016/j.ascom.2018.02.002.
23. G. Biau and E. Scornet, “A Random Forest Guided Tour,” TEST, vol. **25**, no. 2, pp. 197–227, (Apr. 2016), doi: https://doi.org/10.1007/s11749-016-0481-7.
24. Deepanshu Beniwal, A. Roy, H. Yadav, and A. Chauhan, “Detection of Pulsars by Classical Machine Learning Algorithms,” 2021 2nd International Conference for Emerging Technology (INCET), pp. 1–7, (May 2021), doi: https://doi.org/10.1109/INCET51464.2021.9456250.
25. Z.-X. Liang and Y. Liang, “Magnetic-field oscillation model of pulsar radio emission: prediction of an observable effect,” Monthly Notices of the Royal Astronomical Society, vol. **530**, no. 1, pp. 203–209, (Mar. 2024), doi: https://doi.org/10.1093/mnras/stae909.
26. C. J. Zhang, Z. H. Shang, W. M. Chen, L. Xie, and X. H. Miao, “A Review of Research on Pulsar Candidate Recognition Based on Machine Learning,” Procedia Computer Science, vol. **166**, pp. 534–538, (2020), doi: https://doi.org/10.1016/j.procs.2020.02.050.
27. H. Lin, X. Li, and Z. Luo, “Pulsars detection by machine learning with very few features,” Monthly Notices of the Royal Astronomical Society, vol. **493**, no. 2, pp. 1842–1854, (Jan. 2020), doi: https://doi.org/10.1093/mnras/staa218.
28. S. Thanu and D. V. Subha, “Emerging Trends In Pulsar Star Studies: A Synthesis Of Machine Learning Techniques In Pulsar Star Research,” Annals of Computer Science and Information Systems, (Feb. 2022), doi: https://doi.org/10.15439/2023r66.
29. V Priyanka, B. S. Anil, and B. R. Dinakar, “A Comparative Study of Binary Classification Methods for Pulsar Detection,” 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), (Dec. 2018), doi: https://doi.org/10.1109/iceeccot43722.2018.9001517.
30. A. Seth, A. Monga, U. Yadav, and A. S. Rao, “Finding Accuracies of Various Machine Learning Algorithms by Classification of Pulsar Stars,” Lecture notes in mechanical engineering, pp. 51–59, (Jan. 2021), doi: https://doi.org/10.1007/978-981-15-8542-5_5.
31. D. Petrusevich, “INVESTIGATION OF PULSAR STARS ASTRONOMICAL DATASET BY MEANS OF MACHINE LEARNING ALGORITHMS,” SGEM International Multidisciplinary Scientific GeoConference EXPO Proceedings, vol. **20**, pp. 199–206, (Sep. 2020), doi: https://doi.org/10.5593/sgem2020/2.1/s07.026.
32. J. Wolfson et al., “A Naive Bayes machine learning approach to risk prediction using censored, time-to-event data,” Statistics in Medicine, vol. **34**, no. 21, pp. 2941–2957, (May 2015), doi: https://doi.org/10.1002/sim.6526.
33. A. Patle and D. S. Chouhan, “SVM kernel functions for classification,” IEEE Xplore, (Jan. 01, 2013). https://ieeexplore.ieee.org/abstract/document/6524743
34. gulsahdemiryurek, “Pulsar Star Classification,” Kaggle.com, (Apr. 14, 2019). https://www.kaggle.com/code/gulsahdemiryurek/pulsar-star-classification.
35. prashant111, “SVM Classifier Tutorial,” Kaggle.com, (Mar. 13, 2020). https://www.kaggle.com/code/prashant111/svm-classifier-tutorial#3.1-Linear-kernel
36. P. Guo et al., “Pulsar candidate classification using generative adversary networks,” Monthly Notices of the Royal Astronomical Society, vol. **490**, no. 4, pp. 5424–5439, (Oct. 2019), doi: https://doi.org/10.1093/mnras/stz2975.
37. R. Lyon. "HTRU2," UCI Machine Learning Repository, 2015, doi: https://doi.org/10.24432/C5DK6R.
38. Manchester, R. N., Lyne, A. G., Camilo, F., Bell, J. F., Kaspi, V. M., D'Amico, N., ... & Crawford, F. (2001). The Parkes multibeam pulsar survey-I. Observing and data analysis systems, discovery and timing of 100 pulsars. Monthly Notices of the Royal Astronomical Society, 328(1), 17-35.
39. Keith, M. J., Jameson, A., van Straten, W., Bailes, M., Johnston, S., Kramer, M., ... & Levin, L. (2010). The High Time Resolution Universe Pulsar Survey-I. System configuration and initial discoveries. Monthly Notices of the Royal Astronomical Society, 409(2), 619-627.
40. Lorimer, D. R., & Kramer, M. (2012). Handbook of pulsar astronomy. Cambridge University Press

41. Bates, S. D., Lorimer, D. R., & Rane, A. (2012). The application of artificial intelligence techniques to pulsar candidate selection. Monthly Notices of the Royal Astronomical Society, 427(2), 1052-1062.
42. Morello, V., Barr, E. D., Bailes, M., Flynn, C., Keane, E. F., van Straten, W., & Caleb, M. (2014). SPINN: A straight-forward machine learning method for pulsar candidate selection. Monthly Notices of the Royal Astronomical Society, 443(2), 1651-1662.
43. Lyon, R. J., Brooke, J. M., Knowles, J. D., Stappers, B. W., & Lyon, R. J. (2016). Pulsar candidate classification using machine learning. Monthly Notices of the Royal Astronomical Society, 459(1), 1104-1123.
44. C. Fleizach and S. Fukushima, “A naive Bayes classifier on 1998 KDD Cup.” Available: https://cseweb.ucsd.edu/~cfleizac/cse250b/project1.pdf