Proceedings of the 2023 International Conference on Mathematical Physics and Computational Simulation
DOI: 10.54254/2753-8818/12/20230466

Potential pulsars prediction based on machine learning

Yuexiang Wu
No.2 High School of East China Normal University, Shanghai, China, 201203

wuyuexiang@hsefz.cn

Abstract. The search for potential pulsars is a difficult job because of the complex nature of the signals and the vast amounts of data involved. In the last few years, a lot of researchers have tried to use machine learning to deal with complex data. This essay examines how machine learning could help to identify potential pulsars, exploring the various types of algorithms and the challenges and limitations associated with this approach. The essay mainly explored three themes: the training of 5 algorithms for the identification of pulsars, the improvement of 2 algorithms by adjusting parameters, and the simplification of the data to improve the processing speed and performance of the algorithms on prediction. All 5 algorithms reached great accuracy after adjustment and the simplification of the input data can help to boost the prediction time and accuracy for future research about pulsars. The essay highlights the need for further research in this area, as machine learning has demonstrated strong potential for pulsar prediction. By analyzing the results of several previous studies, this essay underscores the importance of machine learning as an approach for predicting potential pulsars and made improvements to the performance of current algorithms by adjusting parameters and simplifying the data.

Keywords: pulsars, machine learning, interdisciplinary astronomy (804).

1. Introduction

Pulsars are special objects in the universe which release beams of radiation. These beams are often observed as regular pulses of radiation, hence the name ‘pulsar’. Pulsars have been studied extensively by astrophysicists due to their unique properties and their potential to provide insights into the fundamental laws of physics. The study of pulsars is significant to astrophysics research because they provide a unique laboratory for nuclear physics and the performance of matter under extreme conditions. In the past, pulsars were detected by analyzing large amounts of data collected from radio telescopes. This process was time-consuming and required a significant amount of human effort. However, recent advances in machine learning have made it possible to automate this process and predict possible pulsars with high accuracy. Logistic Regression [1], K Neighbors Classifier [2], Decision Tree Classifier [3], Gaussian Na¨ıve Bayes, and Gradient Boosting Classifier are some of the algorithms supported by machine learning that could be used as classifiers. These algorithms can classify the complex data of pulsar observations and clarify features that are indicative of pulsar activity. By training these algorithms on known pulsar data, they can accurately predict the presence of pulsars in new observations. The paper used data from High Time Resolution Universe Survey 2 [4] which contains 17898 stars among which 1,639 are pulsars. This is half the number of all the pulsars that had been discovered so far. With the given eight continuous variables, and a single class variable for each star in the data set, the paper analyzes the key variables and the best ways to predict pulsars based on different scenarios. For example, some algorithms may be better suited for detecting pulsars in little amounts of data, while others may be better at identifying pulsars in huge amounts of data. By understanding the strengths and weaknesses of different machine learning algorithms, astrophysicists can optimize their pulsar detection methods and improve our understanding of these fascinating objects. In conclusion, the study of pulsars is crucial to our understanding of the universe, and recent advances in machine learning have made it possible to predict possible pulsars with high accuracy. By analyzing large data sets of pulsar observations, machine learning algorithms can identify patterns that are indicative of pulsar activity and help astrophysicists to optimize their pulsar detection methods. The paper provides a comprehensive analysis of the key traits and best ways to predict pulsars based on different scenarios, which contribute to our understanding of these fascinating objects.

2. Related work

In research on pulsar prediction methods and techniques, the author notes that there have been past works discussing the possible approach to predict potential pulsars based on machine learning. Rustam investigated the limitations of existing methods, such as the inability of some methods to handle large volumes of data [5]. To address these limitations, they came up with a new way based on a tree classifier called the random trees boosting voting classifier (RTB-VC). This classifier employs the High Time Resolution Universe 2 (HTRU2) data set, which is imbalanced, to predict pulsar stars. The authors sought to get artificial data with a synthetic minority oversampling technique for a balanced data set. The proposed approach achieved highly precise results of a high F1 score (98.3). Wang [6] set a convolutional neural network (CNN) of 11 layers to classify the pulsar data. With a similar data imbalance problem to Rustam, the author used synthetic minority samples based on the characteristics of pulsars. The model achieved a recall of 0.962 and a precision of 0.963 in experiments on the HTRU 1 dataset. Both studies address the challenge of accurately predicting pulsar stars from large volumes of data. The first study proposes a hybrid machine learning classifier that combines tree-based classifiers and employs the HTRU2 data set to predict pulsar stars. The second study proposes a CNN with 11 layers and a data augmentation method using synthetic minority samples. Both studies achieved highly accurate results and give great examples of applying machine learning to pulsar prediction. Based on that impressive achievement, the author sought to investigate the features of different machine learning algorithms in predicting the potential pulsars. In the process of enhancing the score of the algorithm, the author noticed some important correlating variables that decide the prediction outcome more significantly than others, and the author sought to find a way to save calculating time.

3. Data set

The data set is made up of 17,898 examples with 1,639 positive ones (which are pulsars) and 16,259 negative ones (which are not pulsars). Figure 1 illustrates that the distribution of positive and negative samples is very uneven. This may result in the post-training model only for most classes and not for a few. The author applies oversampling here by using SMOTE [7]. It helps to balance the data set by generating synthetic samples of the minor examples. Specifically, SMOTE selects one of the positive samples and obtains its k nearest neighbors. It then creates a new sample from the selected one and its k nearest neighbors. This process is repeated until the balance is achieved.

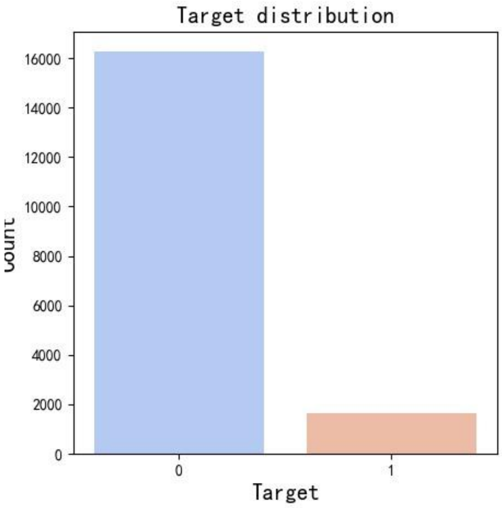

Figure 1. The distribution of positive data (which are pulsars) and negative data (which are not pulsars).

4. Methods

4.1. Variables and correlation

This paper first generates the statistical magnitude of each variable, as shown in Table 1, and noticed that the square deviation is high. To specify the determinacy of each variable in the prediction, then a correlation matrix is generated (Figure 2). This helps to identify which variables are strongly related to each other and which ones are not. The correlation coefficient lies in the interval between -1 to +1. -1 indicates a direct negative correlation and +1 indicates a direct positive correlation, which means 0 indicates no correlation. The diagonal of the table shows the correlation of each variable with itself, which is always 1.

Table 1. The statistical magnitude of each variable.

Stat	mean_ int	std_ int	Ex_ int	Sk_ int	Mean_ DM	Std_ DM	Ex_ DM	Sk_ DM
Mean	111.08	46.55	0.48	1.77	12.61	26.33	8.30	104.86
Std	25.65	6.84	1.06	6.17	29.47	19.47	4.51	106.51

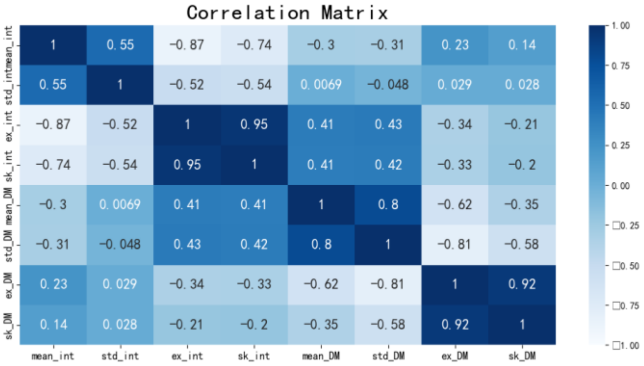

Figure 2. The table shows the correlation coefficients between variables in the dataset.

Note that mean int and ex int, mean int and sk int, ex DM and std DM, sk int and ex int, sk DM and ex DM, std DM and mean DM all show a strong correlation, but because there are few features in this case, feature selection won’t be applied to eradicate the unimportant features. The author would try to eliminate some features that are unimportant at the end of the essay, and compare the advantages and disadvantages of the algorithm with the one without feature elimination.

4.2. Classifier algorithm

The author then applies Logistic Regression, K Neighbors Classifier (KNC), Decision Tree Classifier (DTC), Gaussian Na¨ıve Bayes (GNB), and Gradient Boosting Classifier (GBC) to the training of the prediction of the data. The output results include recall rate, precise rate, F1 value, accuracy rate, run time, AUC value, confusion matrix, and ROC curve. The recall rate is the proportion of correctly identified out of all actual positives. It measures the ability of the model to identify all positive instances. The precision rate is the proportion of true positives out of all predicted positives. The F1 value is the harmonic mean of precision and recall. The Accuracy rate is the proportion of correct predictions out of all predictions. Run time is the time taken by the model to make predictions on a given dataset. AUC value is the area under the receiver operating characteristic (ROC) curve. It measures the ability of the model to distinguish between positive and negative instances. A confusion matrix is a table that demonstrates the parameters of true positives, false positives, true negatives, and false negatives for a given model. The ROC curve is a plot of the true positive rate (recall) against the false positive rate for different threshold values. Table 2 illustrates the results.

Table 2. The statistics value for the performance of each algorithm.

Model	Recall	Precise	Accuracy	AUC	Time
Log Reg	0.7662	0.9064	0.9676	0.9778	0.20
KNC	0.5575	0.8872	0.9285	0.9378	0.18
DTC	0.6519	0.8766	0.9482	0.9158	0.39
GNB	0.6235	0.8702	0.9426	0.9662	0.02
GBC	0.7649	0.9000	0.9670	0.9766	2.11

For the problem of pulsar detection, due to its great scientific research value, the primary goal is to detect all potential pulsars, that is, pay attention to the proportion of pulsars predicted correctly in the number of real pulsars, which corresponds to the Recall rate. It is noted that the recall rate of logistic regression and gradient lifting trees is relatively high. However, after the classification results are obtained by the machine learning algorithm, it will also take a large amount of manpower and material resources for scientists to further verify the data predicted as pulsars. Therefore, it is expected that the positive prediction results can be as accurate as possible, that is, the proportion of correctly predicted pulsars in all the positive predictions should be tracked, which corresponds to a precision rate. Note that logistic regression and gradient lifting trees are also highly accurate. In this case, the recall rate and the accuracy rate need to be combined. Generally speaking, F1 values and AUC values can describe the combined effect of the model. It is found that in the F1 value, logistic regression is the best performance, followed by the gradient lifting tree; In terms of AUC value, logistic regression performs the best, followed by GBC, KNC and DTC poorly. Therefore, the author believes that if the model is evaluated from the perspective of F1 value and AUC value, logistic regression has the best performance, followed by GBC. Therefore, the author will also evaluate the model mainly based on F1 values and AUC values, to be more specific, the ROC curve and the AUC values.

4.3. Parameter adjustment

The author then adjusts the parameters in different algorithms by using GridSearchCV (). GridSearchCV () is a function in the scikit-learn [8] that is used to tune the hyperparameters of a machine-learning model. The function works by exhaustively searching over a specified parameter grid to find the best combination of hyperparameters for the model. The best combination of hyperparameters is determined based on the performance metric specified by the user (e.g., accuracy, F1 score, etc.). Once the most competent hyperparameters are obtained, the model could be retrained to make predictions on new data. Figure 3 shows the performance of KNC before the adjustment and Figure 4 shows the performance afterwards. Figure 5 shows the performance of DTC before the adjustment and Figure 6 shows the performance afterwards. Table 3 concludes the changes in the performance of KNC and DTC, which is also demonstrated in Figure 7.

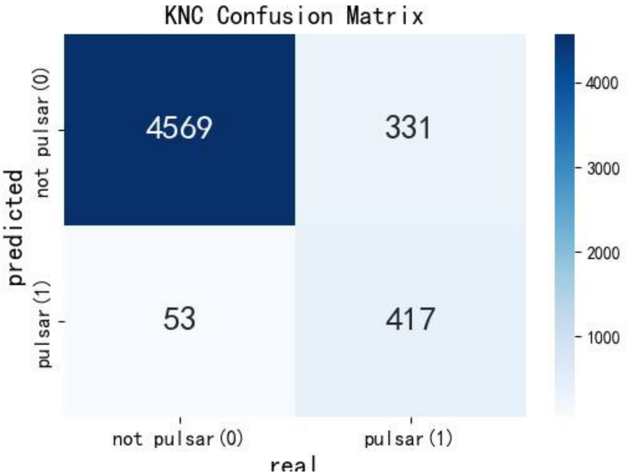

Figure 3. The confusion matrix heat map shows how much data have been predicted correctly or not in the KNC before the adjustment of the parameter.

Table 3. The improvement of KNC and DTC after the adjustment of the parameter.

Model	Recall	Precise	F1	Accuracy	AUC
KNC	0.5575	0.8872	0.6847	0.9285	0.9378
DTC	0.6519	0.8766	0.7477	0.9482	0.9158
KNC(ADJ)	0.5689	0.8957	0.6959	0.9497	0.9497
DTC(ADJ)	0.8142	0.9043	0.8569	0.9222	0.9222

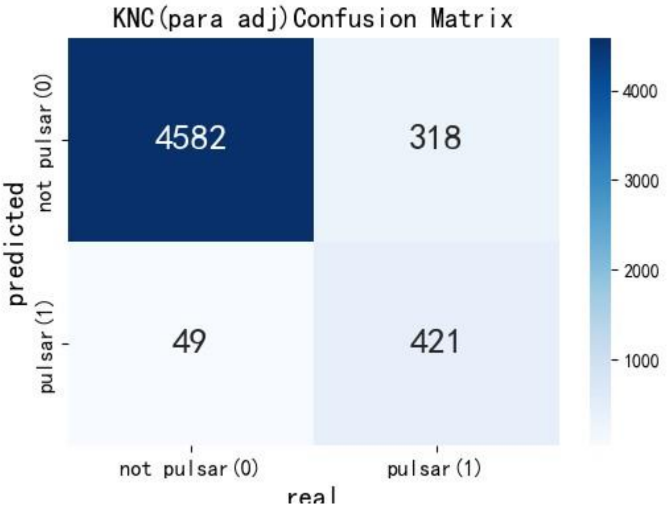

Figure 4. The confusion matrix heat map shows how much data have been predicted correctly or not in the KNC after the adjustment of the parameter.

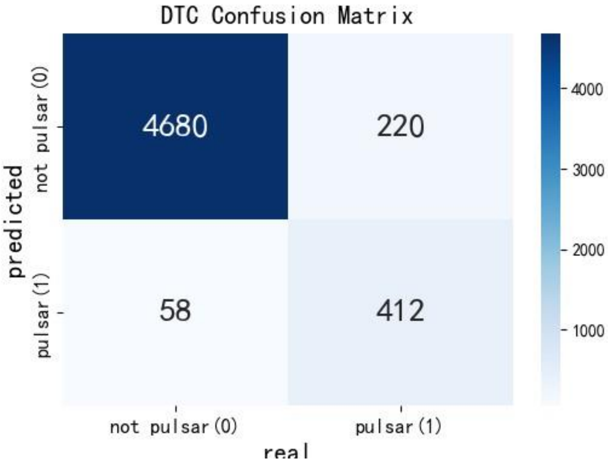

Figure 5. The confusion matrix heat map shows how much data have been predicted correctly or not in the DTC before the adjustment of the parameter.

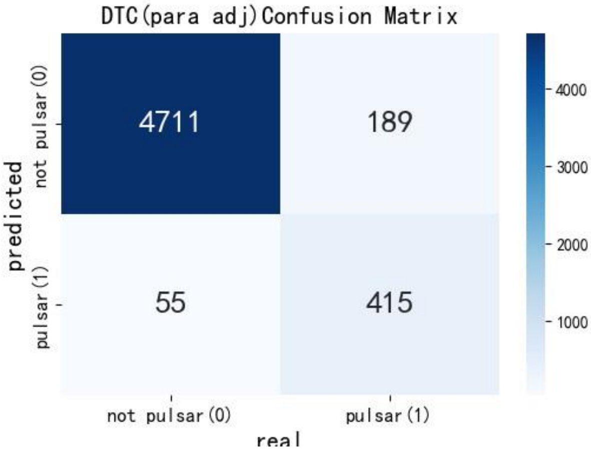

Figure 6. The confusion matrix heat map shows how much data have been predicted correctly or not in the DTC after the parameter adjustment.

It can be seen from the ROC curve that the performance of KNC and DTC is improved, and the improvement of KNC is more obvious than that of DTC. Table 4 compares the algorithm after parameter adjustment with other algorithms, Figure 8 is the comparison with other algorithms' ROC curves.

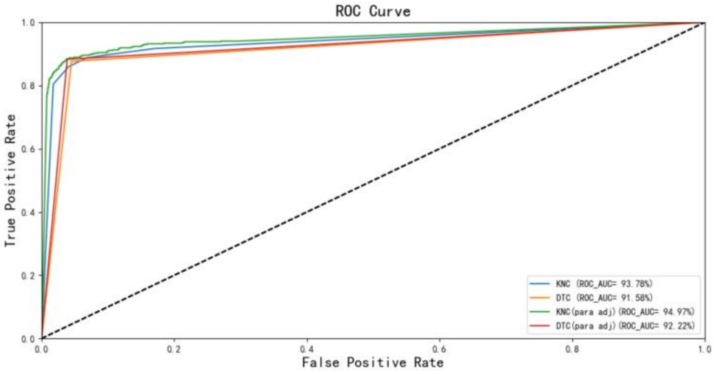

Figure 7. The ROC curve shows the change of AUC and ROC due to the adjustment of parameters. This is the ROC Curve for KNC and DTC before and after adjustments [9].

Table 4. The improvement of KNC and DTC after the adjustment of the parameter.

Model	Recall	Precise	F1	Accuracy	AUC
LOG REG	0.7662	0.9064	0.8304	0.9676	0.9778
GNC	0.6235	0.8702	0.7265	0.9426	0.9662
GBC	0.7649	0.9000	0.8270	0.9670	0.9766
KNC(ADJ)	0.5689	0.8957	0.6959	0.9497	0.9497
DTC(ADJ)	0.8142	0.9043	0.8569	0.9222	0.9222

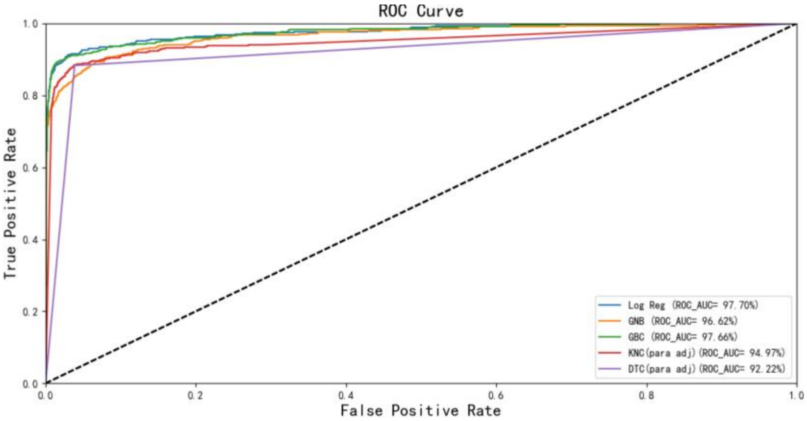

Figure 8. The ROC curve shows the change of AUC and ROC of all algorithms due to the adjustment of the parameter [10].

5. Results

It is noted that the F1 value of the decision tree after the adjustment has the best performance, and the AUC value is close to the logistic regression and gradient lifting tree. Since the running time of the decision tree is significantly lower than the latter two, the decision tree model can be applied after tuning to detect massive pulsar data. Considering the large correlation between multiple features, this paper selects the model with the largest F1 value – the decision tree model after parameter adjustment to output the importance of features in Table 4. Feature importance is calculated based on how much each feature reduces the impurity of the tree (e.g., Gini impurity or entropy) when it is used to split the data. Features that result in the largest reduction in impurity are considered the most important.

Table 5. The significance of different variables in the original data set.

variables	significance
Ex_int	0.911312
Std_DM	0.049091
Std_int	0.015791
Sk_DM	0.010620
Mean_int	0.005700
Mean_DM	0.003854
Sk_int	0.003632
Ex_DM	0.000000

Table 6. The improvement of each algorithm when trained with two figures instead of eight.

Model	Recall	Precise	F1	Accuracy	AUC
LOG REG	0.01	0.00	0.00	0.00	0.00
KNC	-0.08	-0.03	-0.07	-0.02	-0.02
DTC	0.12	-0.01	0.08	0.03	0.01
GNC	0.02	-0.02	0.01	0.000	0.000
GBC	0.04	-0.01	0.02	0.00	0.00
KNC(ADJ)	-0.11	-0.01	-0.08	0.00	-0.02
DTC(ADJ)	0.10	-0.01	0.05	0.00	0.00

The author is surprised to find that ex-int features were extremely important in the decision tree after adjustment, std DM features were less important, and other features had no effect, which indicated that ex-int had a great influence on the results in the decision tree after adjustment. std DM has a small effect, while other characteristics have no effect. Therefore, the author tries to select two features, ex int and std DM, to form a training set, apply the above classification algorithm, and compare and evaluate the advantages and disadvantages of algorithms based on two features and those based on full features. In order to make a more intuitive comparison, the author outputs the result by subtracting the index value of the algorithm with full features from the index value of the algorithm with two features. If it is regular, it means that the algorithm with the full feature is superior to the two features; if it is negative, it means that the algorithm with the full feature is inferior to the two features. The results are shown in Table 6.

It can be seen that the KNC algorithm with two features is better than the full-feature algorithm. Except for the F1 value of the DTC model, the index values of other algorithms have decreased, but not significantly. Therefore, the author can realize approximate mass pulsars detection based on the training set composed of ex-int and std DM solely. That is to say, in the face of massive data sets, the above two features can first be extracted for detection, so as to save computing time.

6. Conclusion

In this case, the author applied logistic regression, KNC, DTC, GNC and GBC to do a binary classification of pulsars. Based on the F1 value and AUC value evaluation model, logistic regression performed best for untuned algorithms, followed by GBC. KNC and DTC's performances are relatively poor. The author selected KNC and DTC parameters, and the performance of both improved after the parameters were adjusted. The F1 value of the K proximity algorithm increased from 0.6847 to 0.6959, the AUC value increased from 0.9378 to 0.9497, the F1 value of the decision tree increased from 0.7477 to 0.8569, the AUC value increased from 0.9158 to 0.9222. The improvement of DTC is relatively not obvious. The author selects the model with the largest F1 value – the DTC model after the regulation to explore the importance of features, and find that ex int has a decisive effect on the results, std DM has a small effect, and other features have little effect. Therefore, the author only extracted ex int and std DM as features to form a new training set. Logistic regression, KNC, DTC, GNC and GBC were used for training again. Compared with the model with full-feature training, the model with two-feature training can achieve the approximate classification effect. Therefore, this case can also play a reference role for the learning of feature selection which saves a lot of time for data collection and computing.

In the last few years, several machine learning algorithms have been developed for pulsar prediction, including decision trees, random forests, and neural networks. These algorithms have been shown to outperform traditional methods of pulsar detection, such as Fourier analysis and template matching. In particular, deep learning models, such as convolutional neural networks (CNNs), have shown great promise in identifying pulsars in noisy data.

Despite the success of machine learning in pulsar prediction, there are still many challenges to be addressed. One of the challenges is to improve the interpretability of machine learning models. While these models can achieve high accuracy in predicting pulsars, it can be difficult to understand how they are making their predictions. This is particularly important in the case of pulsar detection, where the discovery of new pulsars can lead to important insights into the nature of the universe. To address this challenge, researchers have developed techniques such as feature importance analysis and visualization tools.

In conclusion, machine learning has revolutionized the field of pulsar astronomy by enabling more accurate and efficient detection of these fascinating objects. As machine learning continues to evolve, it is likely that more exciting developments in the field of pulsar astronomy can be seen. However, it is important to address the challenges of imbalanced datasets and model interpretability to ensure that machine learning is used in a responsible and effective way.

References

[1] Allison, P. D. Logistic Regression Using the SAS System: Theory and Application. SAS Publishing (2001).

[2] Yu, X., X. Yu. The Research on an Adaptive k-Nearest Neighbors Classifier. IEEE International Conference on Cognitive Informatics IEEE, 2007.

[3] Ping, L., et al. Decision Tree Network Traffic Classifier Via Adaptive Hierarchical Clustering for Imperfect Training Dataset. International Conference on Wireless Communications IEEE, 2009.

[4] Burke-Spolaor, S., et al. The High Time Resolution Universe Survey - III. Single-pulse searches and preliminary analysis. Monthly Notices of the Royal Astronomical Society 4 (2011): 2465-2476.

[5] Rustam, F., et al. Predicting pulsar stars using a random tree-boosting voting classifier (RTB-VC). Astronomy and Computing 32 (2020):100404.

[6] Wang, Y. C., et al. Pulsar candidate classification with deep convolutional neural networks. Research in Astronomy and Astrophysics (2019).

[7] Chawla, N. V., et al. SMOTE: Synthetic Minority Over-sampling Technique. Journal of Artificial Intelligence Research 16.1(2002):321-357.Raschka, Sebastian et al. Python machine learning: machine learning and deep learning with Python, scikit-learn, and TensorFlow. (2017).

[8] Alom, Md Zahangir, et al. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. (2018).

[9] Tomonori, Totani. "Cosmological Fast Radio Bursts from Binary Neutron Star Mergers." Publications of the Astronomical Society of Japan 5 (2013):201-201.

[10] Buda, Mateusz, A. Maki, and M. A. Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. arXiv e-prints (2017).