Astronomy and Computing 32 (2020) 100404

Contents lists available at ScienceDirect

Astronomy and Computing

journal homepage: www.elsevier.com/locate/ascom

Full length article

Predicting pulsar stars using a random tree boosting voting classifier (RTB-VC)

F. Rustam [a], A. Mehmood [b,*], S. Ullah [a], M. Ahmad [a], D. Muhammad Khan [b], G.S. Choi [c,*], B.-W. On [d]

[a] Department of Computer Science, Khwaja Fareed University of Engineering and Information Technology, Rahim Yar Khan, Punjab 64200, Pakistan
[b] Department of Computer Science & Information Technology, The Islamia University of Bahawalpur, Punjab 63100, Pakistan
[c] Department of Information & Communication Engineering, Yeungnam University, Gyeongbuk 38541, Republic of Korea
[d] Department of Software Convergence Engineering, Kunsan National University, Gunsan 54150, Republic of Korea

ARTICLE INFO

Article history:
Received 27 September 2019
Accepted 30 June 2020
Available online 12 August 2020

Keywords:
Pulsar stars
Astronomy
Machine learning
Ensemble learning
Tree-based algorithm
SMOTE

ABSTRACT

The recent exponential growth in the data volume and number of identified pulsar stars is due to pulsar candidate search experiments and surveys. In this study, we investigated the existing methods and techniques used for pulsar prediction, such as applying filters based on pulsar observations, which can adversely affect the success of accurate pulsar prediction. Some of the existing methods are not capable of dealing with large volumes of data and others fail to accurately select the best candidates from pulsar observations. Thus, we developed a new approach based on the traditional supervised machine learning algorithm, which yields faster and more accurate results. In this study, we present our hybrid machine learning classifier called the random trees boosting voting classifier (RTB-VC) for predicting pulsar stars. RTB-VC combines tree-based classifiers and it employs the High Time Resolution Universe 2 (HTRU2) data set comprising a set of eight features related to pulsars and non-pulsars. The HTRU2 data set is imbalanced and we solve this problem by using the synthetic minority oversampling technique to generate artificial data and obtain a balanced data set. A feature set is used to separate pulsar and non-pulsar candidates because the different distributions of variables in the data set are helpful for training models. In the proposed approach, the prediction stage of RTB-VC is based on a combination of soft voting, hard voting, and weighted voting to obtain highly accurate and relevant criteria for finally predicting pulsars or non-pulsars. The ensemble-based structure of RTB-VC yields accurate results based on pulsar observations with a high F_1 score for pulsars (98.3%). We evaluated the learning algorithm in terms of its accuracy, precision, recall, and F_1 score.

1. Introduction

Neutron stars are difficult to detect because they do not emit sufficient readily detectable radiation, but they can be detected under certain conditions when they detonate as supernovae and emit X-rays. Most neutron stars are pulsar stars that rotate rapidly with a very high magnetic field. These rapidly rotating astronomical objects emit radiation at rates of 100,000 km/s to 150,000 km/s with regular patterns and intervals. Pulsar stars emit electromagnetic radiation and these emissions gradually slow down to become quiet within ten to a hundred million years. The first pulsar star was discovered in 1968 (Hewish et al., 1968) by the Mullard Radio Astronomy Observatory, where this instrument was used to investigate radio signals emitted from a fixed point in the sky. The first observed pulsar was labeled as PSR B1919+21 and nicknamed as LGM-1, or "little green men one" (Bethapudi and Desai, 2018). Subsequently, many pulsar candidates and confirmed pulsar stars were discovered.

According to the Australia Telescope National Facility catalog, 2801 pulsars have been discovered (Anon, 2019a; Manchester et al., 2005), and it has also been estimated that there are 20,000 to 100,000 pulsars in our galaxy, and thus about 90% of these pulsars remain undiscovered (Johnston and Karastergiou, 2017). It is very difficult to detect pulsar stars because each emits a slightly different emission pattern, while detecting pulsars in noisy time series is also challenging because it is hard to predict whether signals belong to a pulsar or non-pulsar.

The methods employed to detect the radio emissions from pulsars aim to find occasional broadband signals with indications of

* Corresponding authors.
E-mail addresses: furqan.rustam1@gmail.com (F. Rustam), arifnhmp@gmail.com (A. Mehmood), saleem.ullah@kfueit.edu.pk (S. Ullah), maqsoodzee@gmail.com (M. Ahmad), khan.dostkhan@iub.edu.pk (D. Muhammad Khan), castchoi@ynu.ac.kr (G.S. Choi), bwon@kunsan.ac.kr (B.-W. On).

https://doi.org/10.1016/j.ascom.2020.100404
2213-1337/

scattering caused by movement through the interstellar medium. The signals that satisfy the criteria for pulsars are referred to as "candidates" and they may be confirmed as new pulsars. These candidates are examined using various automated techniques or by human experts in order to check their legitimacy. The candidates determined as probable pulsars are selected for further deep analysis whereas the remainder are ignored. This candidate investigation process is known as "candidate selection" (Lyon et al., 2016). The techniques employed for pulsar searches typically involve the manual (until 2000s) selection of candidates, where inspecting each observation generally requires about 1–300 s (Eatough et al., 2010). Therefore, up to 80,000 person hours are required to visually inspect about one million candidates. Thus, the typical manual visual classification process for pulsar candidates is an intractable problem. However, various novel graphical and automated techniques have been developed to select pulsar candidates, although these pulsar search techniques are very computationally expensive (but many studies have been conducted to improve the speed and sensitivity of these algorithms) (Keith et al., 2009). Continual improvements to the algorithms used in these new techniques have aimed to reduce the ratio of noise in pulsar signals, and thus the signal to noise ratio (SNR) is an important feature used in pulsar detection. In pulsar astronomy, the dispersion measure (DM) is another key feature of a pulsar (McLaughlin et al., 2006) because the pulse is delayed when an astronomer observes a radio pulsar and this delay depends on the radio frequency and the DM. The radio frequency and DM are essential data (and low in number) and they are not associated with any specific hypothesis. In addition, they have no inherent biases with respect to the number of profile/DM curve bins used to describe a pulsar, except possibly in terms of their resolution.

Machine learning techniques are employed to solve the problem of automatically discriminating true signals from noisy data. In the present study, we used various supervised machine learning algorithms and tools to predict pulsar stars. The High Time Resolution Universe 2 (HTRU2) data set was developed by Lyon et al. (2016) based on the data provided by Morello et al. (2014), and we employed a set of new features from the HTRU2 data set in order to obtain a feature set. This feature set contains eight advanced features related to pulsars with two target classes (pulsar and non-pulsar). Four of the features are statistics obtained from the integrated pulse profile (folded profile) and the other four were determined from the DM–SNR curve (Lyon et al., 2016). We used these features to train a machine learning model for predicting pulsars. The data set containing the target class ratios was imbalanced, so we solved the imbalanced data problem by using the synthetic minority oversampling technique (SMOTE) to generate artificial data and obtain equal ratios in the target classes (see Section 3.1). Machine learning models comprising random forest (RF), gradient boosting classifier (GBC), extra tree classifier (ETC), and our random trees boosting voting classifier (RTB-VC) were used in this study. RF, GBC, and ETC were employed as individual learners in our experiment, but we also used these models by combining them under the RTB-VC criteria. We employed the most commonly used combination schemes comprising hard (majority) voting, weighted voting, and soft (average) voting for classification. The final prediction is based on the most votes for a class in majority voting, while the voting weights should fluctuate among the different output classes in each classifier in weighted voting (Zhang et al., 2014), and the final class label is derived from the labels with the highest average probability in average voting. Details of the RTB-VC architecture and voting criteria are presented in Section 4.2. The models were trained with 70% of the data from the balanced data set (data set obtained after applying SMOTE) and the remaining 30% of the data were used for testing purposes. After training the models, a testing procedure was performed to evaluate the models based on their correctness in terms of the accuracy, precision, recall, and F_1 score. The SMOTE technique increases the volume of data and the results are subsequently obtained highly efficiently by RTB-VC, so this method is applicable to large volumes of data such as the upcoming surveys by the Square Kilometer Array project.

The remainder of this paper organized as follows. In Section 2, we describe pulsar research over the last 50 years. In Section 3, we describe the data set and its features. In Section 4, we provide details of the implementations of the algorithms and their features. We present our results and some discussion in Section 5. In Section 6, we give our conclusions.

2. Related work

It is very difficult to identify true pulsar signals based on observations because data noise is a common problem in the pulsar search domain, which requires signal processing. Various techniques have been introduced to deal with these difficulties, including general solutions and those that are highly specific to particular domains. In this section, we discuss the existing techniques employed for pulsar search and classification, as well as explaining why new machine learning techniques are required to solve these problems.

A recent study (Bethapudi and Desai, 2018) used supervised machine learning algorithms to separate pulsar signals from radio frequency interference (RFI) and noise. They used an artificial neural network multi-layer perceptron, Adaboost, GBC, and XGBoost to separate pulsar signals from noise. Their search for pulsars increased the number of pulsar candidates and they also showed that the SNR of the folded profile is the best feature for pulsar classification. Based on the study by Bethapudi and Desai (2018), we also used a machine learning classifier (GBC) in our experiments to predict pulsars and non-pulsars. In addition, previous studies conducted image-based pulsar classification by using machine learning algorithms. In particular, an artificial intelligence system called the Pulsar Image-based Classification System (PICS) was proposed for identifying pulsars (Zhu et al., 2014), which comprises two layers. The first layer comprises a combination of artificial neural networks (ANNs), a convolutional neural network, and support vector machine (SVM) used to examine and score candidate subplots. In the second layer, a logistic regression model is employed to combine the candidate subplot scores to reduce the classification errors in the training data. PICS was integrated with the PALFA survey (a large-scale survey of radio pulsars conducted at 1.4 GHz using the Arecibo 305-meter telescope and the ALFA multi-beam receivers (Lazarus, 2012) to discover six pulsars.

A system for pulsar classification was also proposed called straightforward pulsar identification using neural networks (SPINN) (Morello et al., 2014). ANNs were used in the SPINN system for training based on HTRU observations and the SPINN system successfully discovered four pulsars. Another study (Keith et al., 2009) conducted statistical analyses of data to generate diagnostic plots and over 28 more pulsars were found in a data set that had been previously mined by other researchers. They obtained output scores by performing analyses based on the sub-band, DM curve, and pulse profile (folded profile), and then combined these output scores to determine whether a candidate was a strong.

In our proposed method, we use a voting classifier (ensemble model) with three base learners comprising RF (a bagging method), GBC (a boosting method), and ETC. Voting methods have been used for classification in many studies of pulsar classification and prediction. Explanations of why bagging and boosting algorithms are highly effective for improving the accuracy of artificial and realworld data sets were provided in the study by Bauer and Kohavi (1999) who compared the results obtained by voting methods and non-voting methods, and showed that voting methods can reduce the mean-squared error. Voting classifiers make final predictions based on the predictions generated by sub-models. No previous studies have used voting classifiers for pulsars, but many have applied ensemble models to classification and prediction problems. The classifiers used in ensemble learning are weighted based on their probable classification accuracy with test data in a time-evolving environment (Wang et al., 2003). Ensemble learning improves both the efficiency and accuracy of learning models for performing classification tasks (Wang et al., 2003). Weighted voting classification techniques were proposed for classifying different types of data by Zhang et al. (2014), where the results obtained by each classifier were combined according to the weighted voting classification rules, and the results of weighted voting classifier experiments showed that this approach increased the classification accuracy as well as having the capacities of robust simplification and universality.

The issue of ensemble learning was addressed by Kuncheva (2014) who proposed the combination of several classifiers in order to achieve high performance. In the present study, we also combined several tree-based classifiers because it is considered that tree-based algorithms can learn any type of data rapidly and accurately. The tree-based classification and regression tree model is the most widely used technique employed in data mining problems and classification problems (Chang and Chen, 2005). The predictor space is divided by tree-based algorithms for simplicity and to improve the clarity of the resulting predictive model. Tree-based algorithms such as the RF algorithm (Liaw et al., 2002), GBC algorithm (Mason et al., 2000), and ETC algorithm (Chen et al., 2016) perform best with categorical data. A previous study by Bethapudi and Desai (2018) used three tree-based algorithms comprising GBC, Adaboost, and eXtremeGradientBoosting for separating pulsars signals from noise, where they collectively applied 500 trees in eXtremeGradientBoosting to achieve the desired results. Similarly, another study by Devine et al. (2016a) used six machine learning models comprising ANN (Bishop, 2006), direct ruler learner (Cohen, 1995), hybrid rule and tree learner (Frank et al., 1998), SVM (Cortes and Vapnik, 1995), standard tree ruler learner (Salzberg, 1994), and RF (Breiman, 2001) to classify dispersed pulsar group. Most of these six algorithms are tree based and the results obtained demonstrated the consistent performance of these models.

3. Data set and SMOTE

In Sections 2 and 3, we stated that pulsars are rare types of neutron stars that emit radio signals, which can be detected on the Earth. Many pulsars have been identified using statistical methods. In the present study, we used a data set obtained from Kaggle (a well-known online community of data scientists and machine learners) (Anon, 2019b). This data set contains 17,898 examples of pulsars and non-pulsars, and eight features identified by Lyon (Lyon et al., 2016). The data set includes 16,259 misclassified examples caused by RFI/noise and 1639 real examples of pulsars. All of these examples were checked by human annotators. This data set is known as HTRU2 and it was collected in the High Time Resolution Universe Survey (Bethapudi and Desai, 2018; Lyon et al., 2016). HTRU2 was obtained by analyzing HTRU medium latitude data (Thornton et al., 2013). No positional information related to the pulsars or other astronomical details are included in the data set. PulsarFeatureLab tools was used to extract the feature data for pulsars based on the candidate files (Anon, 2019c).

Table 1
Data set of pulsar candidates.

Data set	Total examples	Non-pulsar	Pulsar
HTRU2	17,898	16,259	1639

Table 2
Numbers of examples before and after applying SMOTE.

Example	Non-SMOTE case	SMOTE case
Total Examples	17,898	32,518
Non-Pulsar Examples	16,259	16,259
Pulsar Examples	1639	16,259

As shown in Table 1, the data set with a total of 17,898 examples contains 16,259 non-pulsar examples and only 1639 pulsars examples, which comprise $\approx$ 10% of data, and thus this unequal ratio means that the data set is imbalanced. These imbalanced data are problematic for classification procedures. Thus, we solved this imbalanced data problem by using the SMOTE technique.

3.1. SMOTE

SMOTE is a well-known technique used to solve imbalanced data problems (Chawla et al., 2002). Imbalanced data problems arise when the class distribution is skewed toward a specific class. According to He and Garcia (2009), a data set is considered imbalanced if the ratio of classes is 1:10 and highly imbalanced if the ratio is 1:100. According to HTRU2 data statistics, the data set only contains 1639 pulsar observations but 16,259 non-pulsar observations, and thus the ratio of 1:10 indicates an imbalanced data set. The SMOTE technique generates mock data to obtain a balance between the minority and majority class populations to resolve the imbalanced data problem. An argument is passed to the SMOTE algorithm to set a threshold value for mock data in order to balance the minority and majority classes. SMOTE selects comparative records and alter the records in one column at a time by adding a random value according to the difference in the adjacent records. An oversampling technique is applied to balance the minority class ratio rather than undersampling because it is necessary to oversample the minority class when the majority/minority ratio is very high according to Barandela et al. (2004). The ratio was 1:10 in our data set so we applied an oversampling technique as suggested by Barandela et al. (2004). We obtained a ratio of 1:1 for the pulsar and non-pulsar examples by using the SMOTE technique, as shown in Table 2.

A recent study by Bethapudi and Desai (2018) used the SMOTE technique to obtain a balanced data set for separating pulsar signals from noise. Another study by Devine et al. (2016b) also used SMOTE in pulsar search experiments. To implement SMOTE, we used an open-source Python toolbox called "imbalanced-learn", which is widely used to solve the problem of imbalanced data sets in machine learning (Lemaıtre et al., 2016). The "imbalanced-learn" tool is based on the numpy, scipy, and scikit-learn Python modules.

3.2. Feature descriptions

The feature set used in the present study was derived by Lyon et al. (2016) and evaluated according to its potential use in separating pulsar and non-pulsar candidates. We used these features in the present study because of the following two reasons. First, these features were selected without any distinct notions regarding their suitability or expressiveness, thereby avoiding potential

Table 3
Descriptions of the features used for pulsar search (Lyon et al., 2016).

Feature	Description	Definition
$Prof_{(mean)}$	Mean of the integrated profile Pr.	$\frac{1}{n}\sum_{i=1}^{n} prof_i$
$Prof_{(standard\ deviation)}$	Standard deviation of the integrated profile Pr.	$\sqrt{\frac{\sum_{i=1}^{n}(prof_i-\overline{Pr})^2}{n-1}}$
$Prof_{(excess\ kurtosis)}$	Excess kurtosis of the integrated profile Pr.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(prof_i-\overline{Pr})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(prof_i-\overline{Pr})^2))^2}-3$
$Prof_{(skewness)}$	Skewness of the integrated profile Pr.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(prof_i-\overline{Pr})^3)}{(\sqrt{\frac{1}{n}(\sum_{i=1}^{n}(prof_i-\overline{Pr})^2})^3}$
$DM_{(mean)}$	Mean of the DM–SNR curve D_c.	$\frac{1}{n}\sum_{i=1}^{n} d_i$
$DM_{(standard\ deviation)}$	Standard deviation of the DM–SNR curve D_c.	$\sqrt{\frac{\sum_{i=1}^{n}(d_i-\overline{D_c})^2}{n-1}}$
$DM_{(excess\ kurtosis)}$	Excess kurtosis of the DM–SNR curve D_c.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(d_i-\overline{D_c})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(d_i-\overline{D_c})^2))^2}-3$
$DM_{(skewness)}$	Skewness of the DM–SNR curve D_c.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(d_i-\overline{D_c})^3)}{(\sqrt{\frac{1}{n}(\sum_{i=1}^{n}(d_i-\overline{D_c})^2})^3}$

bias. Second, these features obtained the best separation and classification results when applied in machine learning algorithms according to Lyon et al. (2016).

Table 3 shows that four of the eight features were obtained from the integrated pulse profile, $P = prof_1, prof_2 ... prof_n$, and the other four were derived from the DM–SNR curve, $D_c = d_1, d_2 ... d_n$. For both P and D_c, all p_i and $d_i \in$ N for i = 1... n. The SNR of the folded profile is the most descriptive feature when considering all of the feature rankings (for further details of these features, see Lyon et al. (2016). Figs. 1 and 2 show some of the characteristics of the features in the data set, where Fig. 1 also illustrates the distributions of all the features in the data set in terms of the mean and standard deviation.

Fig. 1 illustrates the distributions of all the feature variables in the data set in terms of the mean and standard deviation. Fig. 2 shows the ratio of the pulsar and non-pulsar examples in scatter plots with respect to the skewness and kurtosis. Fig. 2(a) shows that the data points are scattered in the skewness and kurtosis profiles. Fig. 2(a) (b) shows that the skewness of the DM–SNR curve and kurtosis of DM–SNR curve are very close to each other for the target classes.

4. Proposed method

In this study, we applied four tree-based machine learning algorithms (RF, GBC, ETC, and RTB-VC) and SMOTE to address the pulsar prediction problem. We used eight pulsar features to train the tree-based algorithms for predicting pulsars.

4.1. Problem area

In the pulsar search domain, various problems have been addressed with different methods. We also investigated some of these problems in our analysis of pulsar search. As discussed briefly in Section 1, the first problem is that it is very difficult to search for pulsars because they have different emission patterns and the noise in these signals makes the detection problem more challenging. The second problem is that the pulsar candidate data set (HTRU2) used in our study is an imbalanced data set and the imbalanced data problem hinders classification or prediction. A further problem is that the application of machine learning is new in the domain of pulsar astronomy, and thus little research has been published for guidance (Bethapudi and Desai, 2018). Thus, we aimed to solve these problems related to pulsar prediction by using machine learning techniques.

Table 4
Implementation details for the machine learning algorithms.

Machine learning algorithm	Parameters
Random Forest	no. of estimators = 200 maximum tree depth = 8 random state = 2
Gradient Boosting Classifier	no. of estimators = 100 learning rate = 0.1 random state = 2 maximum tree depth = 10
Extra Trees Classifiers	no. of estimators = 200 maximum tree depth = 8 random state = 2
Random Trees Boosting Voting Classifier	base learner = RF, GBC & ETC Voting = hard, soft & weighted

4.2. Supervised machine learning models

In this section, we discuss the machine learning algorithms and their implementation in detail. We implemented the machine learning algorithms in Python using the "scikit-learn" module (Pedregosa et al., 2011). We used the following four learning algorithms.

1. RF
2. GBC
3. ETC
4. RTB-VC

We set the parameters based on a trial and error method to obtain the best results with the learning models, and these settings are shown in Table 4.

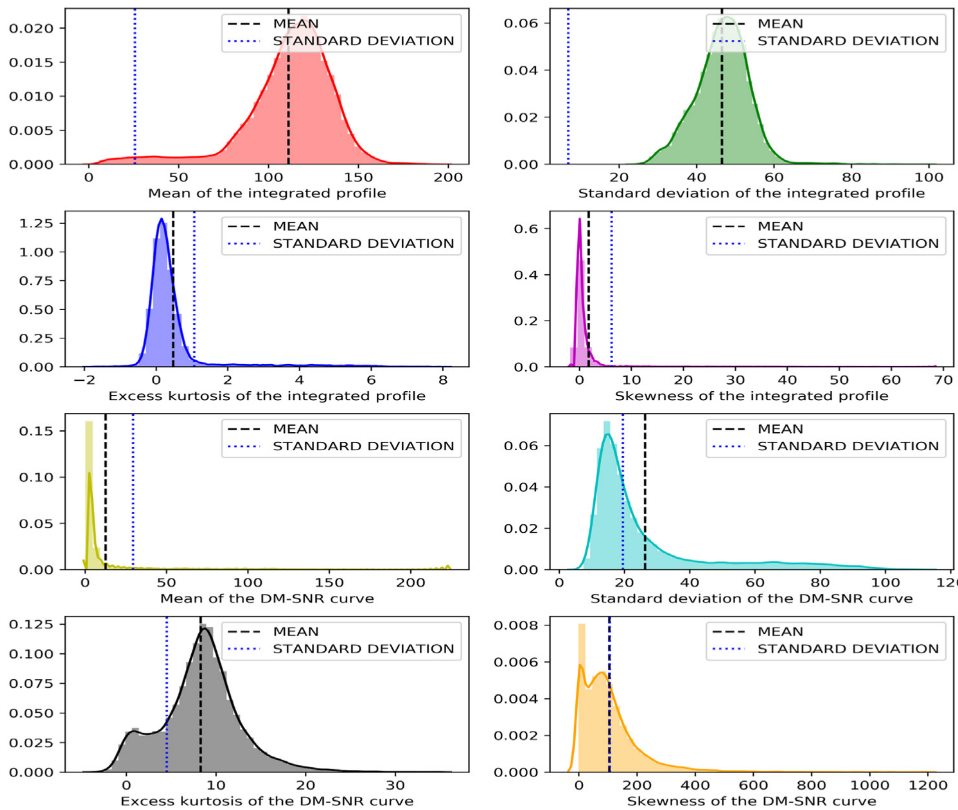

Fig. 1. Distribution of variables in the feature data set.

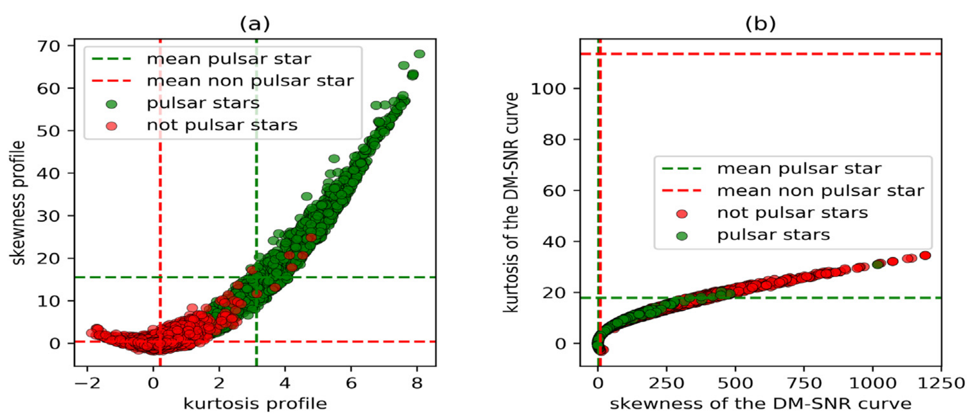

Fig. 2. (a) Scatter plots of skewness and kurtosis for the target classes. (b) Scatter plots of skewness and kurtosis for the DM–SNR curve of the target classes.

(1) RF
The RF classifier is an ensemble model for classification based on the bagging method. RF involves a combination of classifiers (decision trees) where each classifier (decision tree) contributes a single vote to the assignment of the most frequent class for the input vector (w):

$$\widehat{C}^B_{rf} = majorityvote\{\hat{C}_b(w)\}_1^B, \tag{1}$$

where $\hat{C}_b(w)$ is the class prediction for the bth RF tree. The final prediction is made by voting between the RF trees based on the majority class predictions. The combination of many decision trees means that RF has some exceptional characteristic, and thus RF differs significantly from traditional tree-based algorithms (Rodriguez-Galiano et al., 2012). We can also define RF as the simplest case of majority voting and ($\widehat{pc}$) is the predicted class obtained by majority voting for each tree in the RF:

$$\widehat{pc} = mode\{T_1(x), T_2(x), T_3(x)...T_n(x)\} \tag{2}$$

$$\widehat{pc} = mode\{\sum_{i=1}^{n} T_i(x)\}, \tag{3}$$

where $T_1(x)$, $T_2(x)$... are the RF trees and n is the number of trees in the RF. To understand the majority voting criteria, if we assume that RF tree T_1 predicts 0, T_2 predicts 0, and T_3 predicts 1, then the equation gives the following.

$$\widehat{pc} = mode\{0, 0, 1\} = 0 \tag{4}$$

In this case, the final prediction is 0 because the majority of the trees (T_1&T_2) predict the class as 0. These characteristics of RF have attracted much interest because RF is highly accurate and robust to noisy data (Dietterich, 2000b) (for further details of the RF technique, see study (Breiman, 2001).

(2) GBC

GBC uses a boosting method for classification tasks. GBC fits a number of weak learners (decision trees) and they are sequentially trained based on the classifier errors of the predecessors. The first decision tree classifier is fitted based on the data set, before the following decision tree is trained based on the errors of the first classifier and added to the first classifier, and so on. This sequential coupling of classifiers is useful for reducing errors and boosting the classification accuracy. Several previous studies used GBC in astronomy research, e.g., for separating pulsar signals from noise (Bethapudi and Desai, 2018), for the photometric classification of supernovae in astrophysics (Möller et al., 2016), and for the automatic detection and classification of galaxies using the Galaxy Zoo catalog (Tramacere et al., 2016) (for further details of GBC, see Friedman (2001). GBC defines a mean square error (MSE) and minimizes it as follows:

$$L = MSE = \sum (v_i - v_i^p)^2, \tag{5}$$

where L is the loss, v_i is the ith target value, v_i^p is the ith prediction, and L (v_i, v_i^p) is the loss function. GBC updates the predictions based on the learning rate and determines the values where the MSE is minimized. Mathematically, the MSE is minimized as follows:

$$v_i^p = \frac{v_i^p + r * \delta \sum (v_i + v_i^p)^2}{\delta V_i^p} \tag{6}$$

$$v_i^p = v_i^p - r * 2 * \sum (v_i + v_i^p), \tag{7}$$

where r is the learning rate (the learning rate determines how strongly a classifier should be fused with its predecessor (Bethapudi and Desai, 2018)) and $\sum(v_i + v_i^p)$ is the sum of the residual value. Thus, GBC basically updates the predictions where the sum of the residuals is close to 0 or the minimum, and the predicted values are adequately close to the actual values. We used four parameters in the GBC algorithm, as shown in Table 4, and implemented GCB with the "scikit-learn" module.

(3) ETC

ETC is a meta-estimator that is also known as "extremely randomized trees", which uses a number of extra decision trees and fits them based on various subsamples from the data set, before employing an averaging technique to improve the accuracy and control model overfitting. The tree size is controlled by its default parameters but it can also be controlled by setting these parameter values manually. Predictions based on an individual decision tree are not very accurate when new data are considered, so the extra trees used in this method work together to obtain a strong final prediction. ETC operates in a similar manner to RF by randomizing certain decisions, minimizing learning from data, and control overfitting, but RF is more "squeezed" than ETC because training ETC is less computationally expensive. ETC has never been applied previously to core pulsar searches or for separating pulsar candidates from noise. However, a recent study in 2017 by Trovato et al. (2017) used ETC for detecting neutrinos from point-like sources in a collaboration with the Cubic Kilometer Neutrino Telescope KM3NeT).

(4) RTB-VC

In our novel approach, we used RTB-VC to search for pulsars. RTB-VC is a machine learning model used to solve problems related to classification or prediction. The use of voting for predictive models is based on the premise that a set of models can perform more accurately than a single individual model (Rustam et al., 2019). A study by Dietterich (2000a) used a voting technique to evaluate ensemble learning models and they showed that ensembles of different classifiers generally achieved better results than any individual classifier. Similarly, RTB-VC operates with several other classifiers by combining them in a single model and the resultant model is stronger than any of the individual models. In our approach, we used RF, GBC, and ETC as sub-estimators, and obtained the results by combining hard, soft, and weighted voting methods. The new voting criteria can be generalized as follows:

$$\widehat{rf} = mode(\sum_{i=1}^{n} RFdt_i) \tag{8}$$

$$\widehat{gbc} = mode(\sum_{i=1}^{n} GBCdt_i) \tag{9}$$

$$\widehat{etcl} = mode(\sum_{i=1}^{n} ETCdt_i), \tag{10}$$

where ($\widehat{rf}$) is the prediction made by RF, $\widehat{gbc}$ is the prediction made by GBC, and ($\widehat{etcl}$) is the prediction made by ETC. Further evaluations are conducted as follows:

$$\widehat{v1} = mode\{\sum_{i=1}^{n} rf_i, \sum_{i=1}^{n} gbc_i, \sum_{i=1}^{n} etcl_i\} \tag{11}$$

$$\widehat{v2} = argmax\{\sum_{i=1}^{n} rf_i, \sum_{i=1}^{n} gbc_i, \sum_{i=1}^{n} etcl_i\}, \tag{12}$$

where ($\widehat{v1}$) is the prediction made by the three classifiers (RF, GBC, and ETC) based on majority voting, and ($\widehat{v2}$) is the prediction made by the three classifiers based on the probabilities of the outcomes. The final prediction ($\widehat{P_f}$) is computed by associating weights w_1 and w_2 with predictions ($\widehat{v1}$) and ($\widehat{v2}$), respectively:

$$\widehat{p_f} = \underset{i}{\text{argmax}} (\sum_{i=1}^{n} w_1 * v1_i, \sum_{i=1}^{n} w_2 * v2_i) \tag{13}$$

where ($\widehat{p_f}$) is the final prediction computed by the RTB-VC. The operation of the RTB-VC can be illustrated by the following example. We pass a random example from the data set through RTB-VC to obtain a prediction (pulsar or non-pulsar) based on the test example. In the first phase, the test example is passed through hard voting criteria and RTB-VC makes a prediction v1 by majority (plurality) voting between the predictions made by all of the classifiers. The predictions made by the base learners are as follows.

RF prediction class = 1 (Pulsar)
GBC prediction class = 1 (Pulsar)
ETC prediction class = 1 (Pulsar)
$v1 = mode\{1, 1, 1\} = 1$

In this scenario, the majority of the base learners predict that the class is pulsar, so prediction v1 in this phase of RTB-VC is the pulsar class. In the second phase, RTB-VC computes a probability score for each class by using the base learners. The probability scores obtained for each class by RF are as follows.

Non-Pulsar Class = 0.00599553
Pulsar Class = 0.99400447
The GBC probability scores against each class are as follows.
Non-Pulsar Class = 0.0
Pulsar Class = 1.0

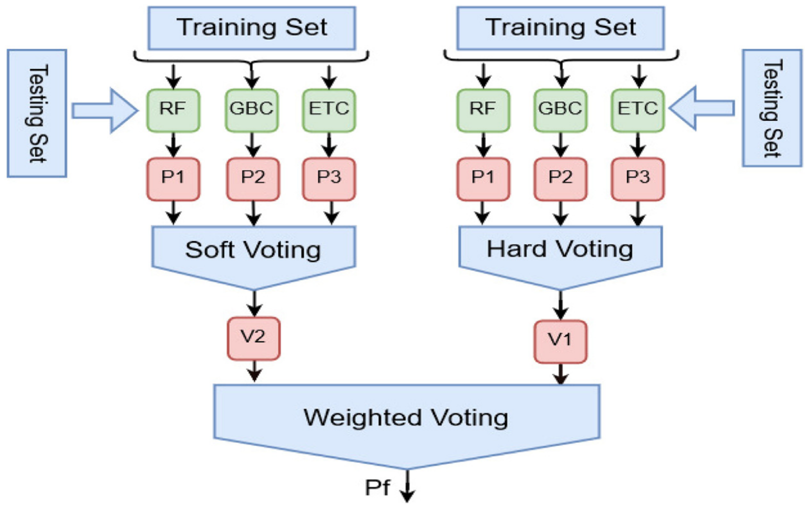

Fig. 3. Architecture of RTB-VC.

Similarly, the ETC probability scores against each class are as follows.
Non-Pulsar Class = 0.0
Pulsar Class = 1.0

RTB-VC obtains prediction v2 by using the probability scores for RF, GBC, and ETC based on soft voting criteria, as follows.
Non-Pulsar Class = (0.00599553+0.0+0.0)/3 = 0.00199851
Pulsar Class = (0. 99400447+1.0+1.0)/3 = 0.99800149
$v2 = argmax\{0.00199851, 0.99800149\} = 1$

In this case, the pulsar class has a greater probability score than the non-pulsar class, so prediction v2 is the pulsar class. In the final prediction stage, v1 and v2 are passed through weighted voting criteria, where we assign weights to predictions v1 and v2 of {0.3, 0.2}, and we determine these weights by tuning the algorithms to obtain the best results based on the weights. The class with the highest weighted average is the final prediction as follows.
$p_f = argmax\{v1 * w1, v2 * w2\} = \{1 * 0.2, 1 * 0.3\} = 1$

RTB-VC predicts that test example is a pulsar, and thus it also belongs to the pulsar class in the data set. A tie may occur between the weighted average of the classes but we can avoid these ties and incorrect labeling by using a real-valued confidence value for each classification. The architecture of RTB-VC is illustrated in Fig. 3.

4.3. Implementation of proposed method

In our method, we used supervised machine learning techniques to predict pulsar stars. In the first step, we loaded the data and then preprocessed our data set to solve the imbalanced data problem. We used the SMOTE method to oversample the minority class (see Section 3.1). The data were then split into two sets comprising the training set and testing set after applying the SMOTE technique. The data were split at a ratio of 70:30, where 70% of the data were used for training purposes and 30% for testing purposes. The training set was used to train the machine learning models as defined in Section 4.2, where these learning models were trained based on the eight features of pulsars. After training the models, we evaluated their performance using the test data in terms of the accuracy, precision, recall, and F_1 score.

Fig. 4 shows the steps followed in this study comprising: **1-** loading the data; **2-** oversampling the minority class; **3-** splitting the data into training and testing sets; **4-** training the machine learning models; **5-** validating the performance of the models; and **6-** optimizing the parameters to maximize the prediction performance.

Table 5
Confusion matrix.

	Pulsar(1)	**Non-Pulsar(0)**
Pulsar(1)	True Positive (TP)	False Positive (FP)
Non-Pulsar(0)	False Negative (FN)	True Negative (TN)

4.4. Evaluation measures

We calculated various performance score for each machine learning algorithm to evaluate their performance. The evaluations were based on the scores for four parameters comprising the accuracy, precision, recall, and f1 score in order to compare the performance of the machine learning algorithms. We also calculated a confusion matrix for each algorithm. The confusion matrix is also known as the error matrix and its specific layout allowed the performance of each supervised machine learning algorithm to be visualized. Table 5 illustrates the confusion matrix used for binary classification, where the rows represent the ground truth labels and the columns represent the class labels predicted by the machine learning algorithms.

For a binary classification problem, each element of the confusion matrix corresponds to a true positive (TP), true negative (TN), false positive (FP), or false negative (FN), as defined in Table 5. In our binary classification problem, if the classifier predicts that a candidate is a pulsar, then it is placed in the positive class and assigned a label of 1. If the classifier predicts that a candidate is a non-pulsar, then it is placed in the negative class and assigned a label of 0. The confusion matrix contains the following four terms.

- **TP** is an outcome where the machine learning model correctly predicts the pulsar class.
- **TN** is an outcome where the machine learning model correctly predicts the non-pulsar class.
- **FP** is an outcome where the machine learning model incorrectly predicts the pulsar class but the actual class is the non-pulsar class.
- **FN** is an outcome where the machine learning model incorrectly predicts the non-pulsar class but the actual class is the pulsar class.

Next, we define the parameters used to evaluate the machine learning models.

(a) **Accuracy** scores were used to define the correctness of the machine learning models. The accuracy score is mostly

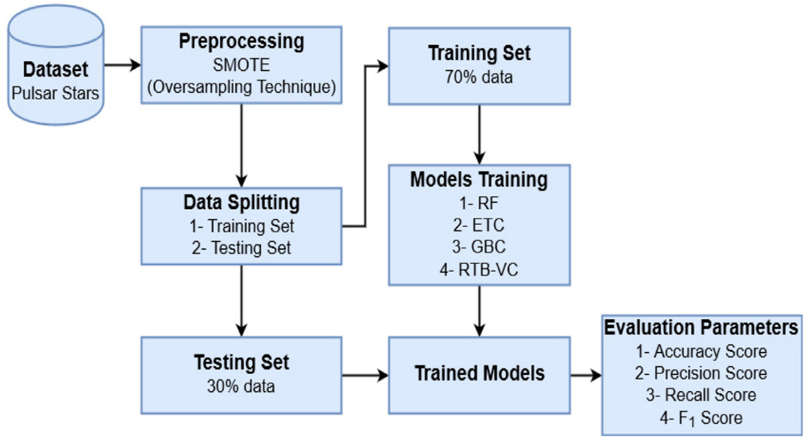

Fig. 4. Diagram illustrating the method employed in this study.

used in classification and regression problems where the classes are equally distributed (Bethapudi and Desai, 2018). The accuracy score is the ratio of the number of correct predictions relative to the total number of input samples, e.g., if the total number of input samples is 100 and the model correctly predicts 70 samples in the prediction phase, then the accuracy is 70%. The accuracy is defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

The best value for the accuracy score is 1 and the worst is 0. The accuracy can also be defined as follows.

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ number\ of\ predictions} \tag{15}$$

(b) **Precision** is also known as the positive predictive value, where it is defined as the fraction of the observations that are actually positive among all of the examples predicted as positive by the model.

$$Precision = \frac{TP}{TP + FP} \tag{16}$$

The best score for precision is 1 and the worst is 0. A high precision score denotes that an algorithm is better at determining observations in the true class. The precision can also be defined as follows.

$$Precision = \frac{Number\ of\ true\ positives}{Total\ number\ predicted\ as\ positive} \tag{17}$$

(c) **Recall** is also known as the sensitivity, which denotes the capability of correctly detecting pulsars. Recall is defined as follows.

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

Recall implies that the true class contains all positively classified observation, where the minimum recall score value is 0 and the maximum is 1. The score is greater when a Type II error (non-rejection of a false null hypothesis) has a higher cost than a Type I error (rejection of a true null hypothesis).

(d) $F_1 Score$ is also known as the F measure and it is the harmonic mean of the precision and recall. The maximum value of the F_1 score is 1 and the minimum is 0 in the same manner as the accuracy, precision, and recall. The F_1 score is defined as follows.

$$F_1 Score = 2 * \frac{Precision * Recall}{Precision + Recall} \tag{19}$$

Table 6
Evaluation results obtained by machine learning algorithms with the SMOTE technique.

Algorithm	Accuracy	Precision	Recall	F_1 Score
RF	0.976	0.982	0.970	0.976
ETC	0.976	0.977	0.973	0.976
GBC	0.966	0.979	0.952	0.965
RTB-VC	**0.983**	**0.986**	**0.978**	**0.982**

5. Results and discussion

In this study, we used the HTRU2 data set extracted from Kaggle to produce accurate models for predicting pulsar stars. The data set containing 16,259 non-pulsar examples and 1639 pulsar examples had an unequal ratio of the target classes, and thus it was affected by the imbalanced data problem. We used the SMOTE technique (by generating artificial mock data) to address the imbalanced data problem. After applying the SMOTE technique, we obtained target classes with an equal ratio and we used 70% of the examples for training the learning models. We used 30% of the data to test the learning algorithms and the performance of each algorithm was averaged to summarize the overall performance. The results showed that the classifier could achieve high accuracy by using the features described in Section 3.2.

Table 6 shows that the RTB-VC algorithm consistently performed better than RF, GBC, and ETC in terms of all the evaluation scores. This is an extremely desirable result for an ensemble classifier because assigning positive labels excessively frequently will return an intractable number of candidates. The classifiers achieved high accuracy, precision, recall, and F_1 scores because they did not always predict non-pulsars.

The TP rate was higher than the FP rate with all of the machine learning algorithms. RTB-VC achieved the highest precision score of 0.986 and it also consistently obtained high scores in terms of all the evaluation parameters. The scores in bold type in Table 6 indicate the best performance results obtained (see Fig. 5).

Fig. 6 shows the binary classification results (pulsars and non-pulsars) in a confusion matrix. The RTB-VC learning model obtained more accurate predictions than the other methods. The TP and TN rates were very high, thereby indicating the accuracy of the predictions for both pulsars and non-pulsars. According to the confusion matrix, RTB-VC predicted 9590 of 9756 examples correctly (TP + TN) and only 166 were incorrect predictions (FP + FN), which was a lower amount than that with any of the other algorithms. The good performance of the algorithms can be explained by the tree-based algorithms being intrinsically

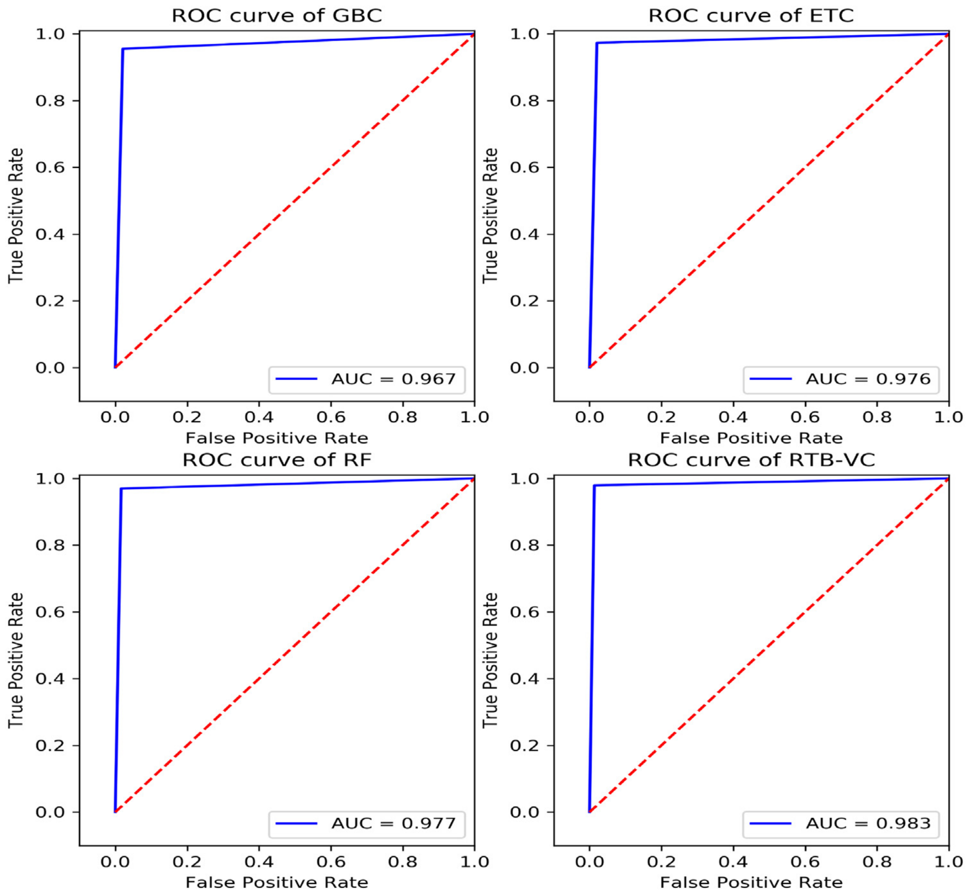

Fig. 5. True positive rate vs false positive rate for all of the machine learning algorithms.

Table 7
Results obtained by the machine learning algorithms based on the raw data set.

Algorithms	Accuracy	Precision	Recall	F_1 Score
RF	0.980	0.925	**0.859**	0.887
ETC	0.979	0.931	0.834	0.880
GBC	0.978	0.912	0.851	0.881
RTB-VC	**0.981**	**0.938**	0.846	**0.890**

Table 8
Experimental results obtained based on the HTRU2 data set by Lyon et al. (2016).

Algorithm	Accuracy	Precision	Recall	F_1 Score
C4.5	0.946	0.635	0.904	0.740
Multi-Layered Perceptron	0.947	0.650	**0.913**	0.752
Naive Bayes	0.937	0.579	0.863	0.692
Support Vector Machine	0.961	0.723	0.871	0.789
GH-VFDT	**0.978**	**0.899**	0.829	**0.862**

more robust to the features used in our study because tree-based algorithms automatically find important features, and thus they can offer many insights into the data.

We also performed an experiment based on the raw data (imbalanced data) with the machine learning algorithms and the results are shown in Table 7. According to Table 7, RTB-VC also performed better with imbalanced data set but the overall performance of the learning models was not as good compared with the results obtained after applying the SMOTE technique. The best classifier should correctly classify both pulsars and non-pulsars. Thus, it should misidentify zero pulsars and zero non-pulsar candidates to obtain the highest possible accuracy. However, high accuracy with an imbalanced data set could be misleading in different cases, such as if we apply a classifier to a data set containing 10 pulsar and 90 non-pulsar examples where the classifier may predict all of the non-pulsars with an accuracy of 0.90, which is misleading. Table 7 shows that the accuracy was 0.981 with RTB-VC without SMOTE but the F_1 score was only 0.890. In the case of imbalanced data, the F_1 score may provide a better evaluation of a classifier's performance, which explains why we compared the F_1 scores for all of the approaches (Lu et al., 2019).

Good results were also achieved by Lyon et al. (2016) with a tree-based algorithm called the Gaussian Hellinger very fast decision tree (GH-VFDT). Our novel approach improved the HTRU2 classification results compared with those obtained by Lyon et al. (2016), and the individual algorithms used in our experiments also performed well in comparison. The experimental results obtained by Lyon et al. (2016) based on the HTRU2 data set are shown in Table 8, and our results are presented in Tables 6 and 7 with and without the SMOTE technique, respectively. The results obtained by RTB-VC and GH-VFDT are compared in Fig. 7.

Table 8 shows the results obtained based on the HTRU2 data set by Lyon et al. (2016). The scores in bold type indicate the best results obtained. According to Table 7, GH-VFDT obtained the highest scores for three evaluation parameters comprising the accuracy, precision, and F_1 score, and the multi-layered perceptron performed best in terms of the recall score.

Using the machine learning algorithms, we observed that the accuracy and F_1 scores obtained with RF, GBC, and ETC varied only after the third significant digit, but RTB-VC performed significantly better in terms of all the evaluation parameters. The superior performance of RTB-VC may be attributed to the tree-based algorithm, as discussed above, but also to the ensemble

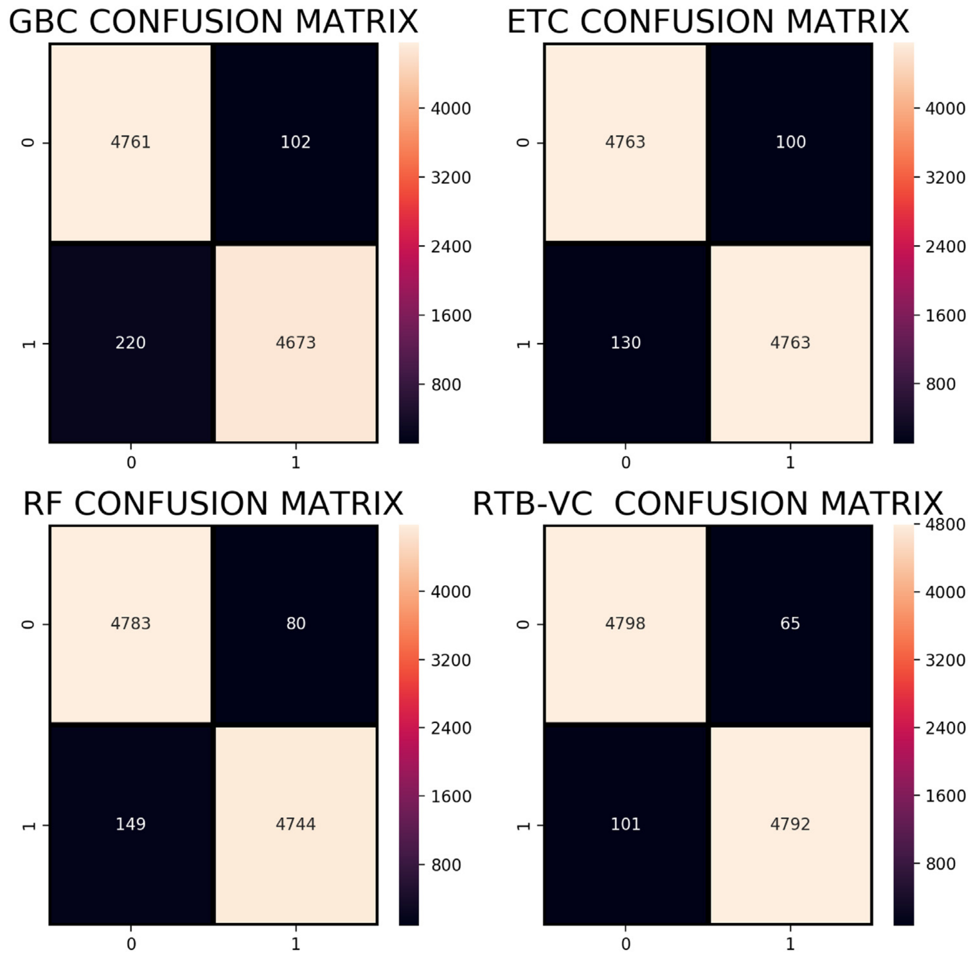

Fig. 6. Confusion matrix obtained for all of the learning algorithms, where 1 represents the pulsar class and 0 represents the non-pulsar class (see Table 5).

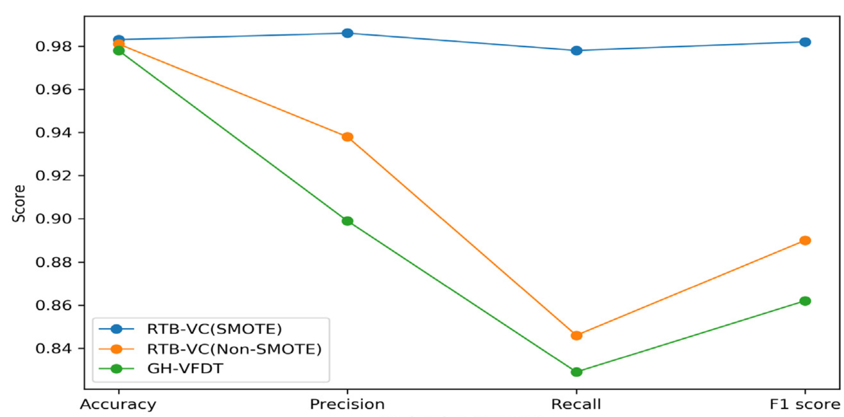

Fig. 7. Comparison of best performance obtained in the study by Lyon et al. (2016) using GH-VFDT and with RTB-VC (SMOTE and without SMOTE) in the present study.

architecture of RTB-VC. Thus, the tree-based algorithm and ensemble architecture allowed RTB-VC to outperform all of the other learning models.

The other models used in our experiments also performed well because of their significant characteristics. For example, GBC sequentially trains the base learner trees based on the errors in the predecessor tree in an additive manner (Mason et al., 2000). The most important hyper-parameter is the learning rate for the performance of GBC, but GBC also has a critical drawback because the sequential algorithm makes it ineffective with massive volumes of data. In addition, RF is a bagging algorithm rather than a boosting algorithm and it generates trees autonomously using a random sample of the data. Thus, parallel execution is possible with RF, which makes it more powerful than GBC. ETC also works in the same manner as RF but this meta-estimator considers more optima by fitting a number of randomized decision trees based on various sub-samples, and an averaging technique is then used to improve the accuracy and avoid overfitting (Netolický et al., 2017). We also compared the performance of RTB-VC with

Table 9
Comparison between RTB-VC and other standard algorithms in terms of the F_1*Score*.

Algorithms	SMOTE	Non-SMOTE
NB	0.959	0.741
DT	0.971	0.835
LR	0.976	0.881
SVM	0.969	0.874
RTB-VC	0.982	0.890

standard algorithms comprising naïve Bayes (NB), decision tree (DT), logistic regression (LR), and SVM with and without the SMOTE technique in terms of the F_1*Score*. Table 9 shows that RTB-VC performed better than all of the other standard and non-ensemble algorithms.

6. Conclusion

In this study, we investigated the pulsar star prediction problem based on the pulsar candidate selection process. We employed multiple techniques to solve this problem by using eight pulsar features that are useful for predicting pulsar and non-pulsar candidates. We developed a candidate classification algorithm (RTB-VC) for pulsar data processing. We also implemented other supervised machine learning algorithms and compared their efficiency and that of our algorithm at predicting pulsar stars. Most of the well-known learning algorithms (RF, GBC, and ETC) can be applied with a generic change of the hyperparameter. We used the HTRU2 data set in this study, which is an imbalanced data set. The high imbalance problem was solved using the SMOTE technique, which also reduced the likelihood of overfitting by the learning algorithms. We found that the tree-based algorithms tested in this study could automatically yield important feature to provide many insights into the data. In addition, we demonstrated that these tree-based algorithms are suitable for rapid training and testing, and the training process is not complex. We also showed that RTB-VC obtained the highest accuracy at target class prediction. A combination of hard, soft, and weighted voting could produce more accurate results than the individual methods because the combination of majority voting and probability-based prediction yielded stronger prediction criteria.

Moreover, we demonstrated that balancing the ratio of the target classes produced better classification results and that the imbalanced data problem could result in overfitting of the learning models. According to the confusion matrix shown in Fig. 6, the equal TP and TN rates indicate that the learning models used in our experiment avoided overfitting. RTB-VC predicted more accurate labels/target classes than any other ensemble classifier, and thus an ensemble of algorithms could achieve better classification results. This explains why the TP rate was much higher while the FP rate was less than half a percent with all of the learning algorithms, which are important results.

Finally, it should be noted that our results and conclusions are based on a single data set (HTRU2) and the feature set derived by Lyon et al. (2016). The algorithms used in this study have not been tested on other data set sets and their performance could be different with other data sets. In future research, we plan to implement neural network algorithms to solve the pulsar prediction problem and we will also conduct tests with other pulsar search data sets (e.g., see (Lyon et al., 2016) using these machine learning algorithms. In future research, we will also address the problem of incorrect pulsar prediction by developing a new method.

CRediT authorship contribution statement

F. Rustam: Conception and design of study, Acquisition of data, Methodology, Writing - original draft, Writing - review & editing, Analysis and/or interpretation of data. **A. Mehmood:** Conception and design of study, Writing - review & editing. **S. Ullah:** Acquisition of data. **M. Ahmad:** Acquisition of data. **D. Muhammad Khan:** Writing - review & editing. **G.S. Choi:** Analysis and/or interpretation of data. **B.-W. On:** Analysis and/or interpretation of data.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

All authors approved the version of the manuscript to be published (F. Rustam, A. Mehmood, S. Ullah, M. Ahmed, D. Muhammad Khan, G.S. Choi, B.-W. On).

References

Anon, 2019a. ATNF Pulsar catalogue. URL https://www.atnf.csiro.au/research/pulsar/psrcat/. (Accessed 28 November 2019).
Anon, 2019b. Predicting a Pulsar Star(HTRU2 dataset), https://www.kaggle.com/pavanraj159/predicting-a-pulsar-star/. (Accessed 28 November 2019).
Anon, 2019c. Pulsar feature lab tool. URL https://figshare.com/articles/Pulsar_Feature_Lab/1536472/1/. (Acessed 28 November 2019).
Barandela, R., Valdovinos, R.M., Sánchez, J.S., Ferri, F.J., 2004. The imbalanced training sample problem: Under or over sampling?. In: Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR). Springer, pp. 806–814.
Bauer, E., Kohavi, R., 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Mach. Learn. 36 (1–2), 105–139.
Bethapudi, S., Desai, S., 2018. Separation of pulsar signals from noise using supervised machine learning algorithms. Astron. Comput. 23, 15–26. doi:10.1016/j.ascom.2018.02.002, arXiv:1704.04659.
Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer Science+Business Media.
Breiman, L., 2001. Random forests. Mach. Learn. 45 (1), 5–32.
Chang, L.-Y., Chen, W.-C., 2005. Data mining of tree-based models to analyze freeway accident frequency. J. Saf. Res. 36 (4), 365–375.
Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P., 2002. SMOTE: synthetic minority over-sampling technique. J. Artif. Intell. Res. 16, 321–357.
Chen, D.L., Kumar, M., Motwani, V., Yeres, P., 2016. Is justice really blind? And is it also deaf. And is it Also Deaf.
Cohen, W.W., 1995. Fast effective rule induction. In: Machine Learning Proceedings 1995. Elsevier, pp. 115–123.
Cortes, C., Vapnik, V., 1995. Support-vector networks. Mach. Learn. 20 (3), 273–297.
Devine, T.R., Goseva-Popstojanova, K., McLaughlin, M., 2016a. Detection of dispersed radio pulses: a machine learning approach to candidate identification and classification. Mon. Not. R. Astron. Soc. 459 (2), 1519–1532.
Devine, T.R., Goseva-Popstojanova, K., McLaughlin, M., 2016b. Detection of dispersed radio pulses: a machine learning approach to candidate identification and classification. Mon. Not. R. Astron. Soc. (ISSN: 0035-8711) 459 (2), 1519–1532. doi:10.1093/mnras/stw655.
Dietterich, T.G., 2000a. Ensemble methods in machine learning. In: International Workshop on Multiple Classifier Systems. Springer, pp. 1–15.
Dietterich, T.G., 2000b. An experimental comparison of three methods for constructing ensembles of decision trees: Bagging, boosting, and randomization. Mach. Learn. 40 (2), 139–157.
Eatough, R.P., Molkenthin, N., Kramer, M., Noutsos, A., Keith, M., Stappers, B., Lyne, A., 2010. Selection of radio pulsar candidates using artificial neural networks. Mon. Not. R. Astron. Soc. 407 (4), 2443–2450.
Frank, E., Wang, Y., Inglis, S., Holmes, G., Witten, I.H., 1998. Using model trees for classification. Mach. Learn. 32 (1), 63–76.
Friedman, J.H., 2001. Greedy function approximation: a gradient boosting machine. Ann. Stat. 1189–1232.
He, H., Garcia, E.A., 2009. Learning from imbalanced data. IEEE Trans. Knowl. Data Eng. 21 (9), 1263–1284.

Hewish, A., Bell, S.J., Pilkington, J.D., Scott, P.F., Collins, R.A., 1968. Observation of a rapidly pulsating radio source. Nature 217 (5130), 709.
Johnston, S., Karastergiou, A., 2017. Pulsar braking and the P–diagram. Mon. Not. R. Astron. Soc. 467 (3), 3493–3499.
Keith, M., Eatough, R., Lyne, A., Kramer, M., Possenti, A., Camilo, F., Manchester, R., 2009. Discovery of 28 pulsars using new techniques for sorting pulsar candidates. Mon. Not. R. Astron. Soc. 395 (2), 837–846.
Kuncheva, L.I., 2014. Combining Pattern Classifiers: Methods and Algorithms. John Wiley & Sons.
Lazarus, P., 2012. The PALFA survey: Going to great depths to find radio pulsars. Proc. Int. Astronom. Union 8 (S291), 35–40.
Lemaıtre, G., Nogueira, F., Aridas, C.K., 2016. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. CoRR abs/1609.06570.
Liaw, A., Wiener, M., et al., 2002. Classification and regression by randomForest. R news 2 (3), 18–22.
Lu, Y., Cheung, Y.-m., Tang, Y.Y., 2019. Bayes Imbalance impact index: A measure of class imbalanced dataset for classification problem. arXiv preprint arXiv:1901.10173.
Lyon, R.J., Stappers, B.W., Cooper, S., Brooke, J.M., Knowles, J.D., 2016. Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach. Mon. Not. R. Astron. Soc. (ISSN: 0035-8711) 459 (1), 1104–1123. doi:10.1093/mnras/stw656.
Manchester, R., Hobbs, G., Teoh, A., Hobbs, M., 2005. The Australia telescope national facility pulsar catalogue. Astron. J. 129 (4), 1993.
Mason, L., Baxter, J., Bartlett, P.L., Frean, M.R., 2000. Boosting algorithms as gradient descent. In: Advances in Neural Information Processing Systems. pp. 512–518.
McLaughlin, M.A., Lyne, A., Lorimer, D., Kramer, M., Faulkner, A., Manchester, R., Cordes, J., Camilo, F., Possenti, A., Stairs, I., et al., 2006. Transient radio bursts from rotating neutron stars. Nature 439 (7078), 817.
Möller, A., Ruhlmann-Kleider, V., Leloup, C., Neveu, J., Palanque-Delabrouille, N., Rich, J., Carlberg, R., Lidman, C., Pritchet, C., 2016. Photometric classification of type ia supernovae in the supernova legacy survey with supervised learning. J. Cosmol. Astropart. Phys. 2016 (12), 008.
Morello, V., Barr, E., Bailes, M., Flynn, C., Keane, E., van Straten, W., 2014. SPINN: a straightforward machine learning solution to the pulsar candidate selection problem. Mon. Not. R. Astron. Soc. 443 (2), 1651–1662.
Netolickỳ, P., Petrovskỳ, J., Dařena, F., Žižka, J., 2017. Text classification using time windows applied to stock exchange. Int. J. New Comput. Archit. Appl. 7 (2), 62–67.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12 (Oct), 2825–2830.
Rodriguez-Galiano, V.F., Ghimire, B., Rogan, J., Chica-Olmo, M., Rigol-Sanchez, J.P., 2012. An assessment of the effectiveness of a random forest classifier for land-cover classification. ISPRS J. Photogramm. Remote Sens. 67, 93–104.
Rustam, F., Ashraf, I., Mehmood, A., Ullah, S., Choi, G.S., 2019. Tweets classification on the base of sentiments for US airline companies. Entropy 21 (11), 1078.
Salzberg, S.L., 1994. C4. 5: Programs for Machine Learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1994.
Thornton, D.e.a., Stappers, B., Bailes, M., Barsdell, B., Bates, S., Bhat, N., Burgay, M., Burke-Spolaor, S., Champion, D., Coster, P., et al., 2013. A population of fast radio bursts at cosmological distances. Science 341 (6141), 53–56.
Tramacere, A., Paraficz, D., Dubath, P., Kneib, J.-P., Courbin, F., 2016. ASTErIsM: application of topometric clustering algorithms in automatic galaxy detection and classification. Mon. Not. R. Astron. Soc. 463 (3), 2939–2957.
Trovato, A., Coniglione, R., Sapienza, P., Barrios-Martí, J., 2017. Expectations for detection of neutrinos from point-like sources with KM3NeT/ARCA. PoS 999.
Wang, H., Fan, W., Yu, P.S., Han, J., 2003. Mining concept-drifting data streams using ensemble classifiers. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. AcM, pp. 226–235.
Zhang, Y., Zhang, H., Cai, J., Yang, B., 2014. A weighted voting classifier based on differential evolution. In: Abstract and Applied Analysis, vol. 2014. Hindawi.
Zhu, W.W., Berndsen, A., Madsen, E.C., Tan, M., Stairs, I.H., Brazier, A., Lazarus, P., Lynch, R., Scholz, P., Stovall, K., Ransom, S.M., Banaszak, S., Biwer, C.M., Cohen, S., Dartez, L.P., Flanigan, J., Lunsford, G., Martinez, J.G., Mata, A., Rohr, M., Walker, A., Allen, B., Bhat, N.D.R., Bogdanov, S., Camilo, F., Chatterjee, S., Cordes, J.M., Crawford, F., Deneva, J.S., Desvignes, G., Ferdman, R.D., Freire, P.C.C., Hessels, J.W.T., Jenet, F.A., Kaplan, D.L., Kaspi, V.M., Knispel, B., Lee, K.J., van Leeuwen, J., Lyne, A.G., McLaughlin, M.A., Siemens, X., Spitler, L.G., Venkataraman, A., 2014. Searching for pulsars using image pattern recognition. Astrophys. J. 781 (2), 117. doi:10.1088/0004-637x/781/2/117.