Searching for pulsars using image pattern recognition

W. W. Zhu[1], A. Berndsen[1], E. C. Madsen[1], M. Tan[1], I. H. Stairs[1], A. Brazier [2], P. Lazarus[3], R. Lynch[4], P. Scholz[4], K. Stovall[5], S. M. Ransom[6], S. Banaszak[7], C. M. Biwer[7,19], S. Cohen[5], L. P. Dartez[5], J. Flanigan[7], G. Lunsford[5], J. G. Martinez[5], A. Mata[5], M. Rohr[7], A. Walker[7], B. Allen[8,7,17], N. D. R. Bhat[9,10], S. Bogdanov[11], F. Camilo[11,13], S. Chatterjee[2], J. M. Cordes[2], F. Crawford[12], J. S. Deneva [20], G. Desvignes[3], R. D. Ferdman[4,14], P. C. C. Freire[3], J. W. T. Hessels[15,16], F. A. Jenet[5], D. L. Kaplan[7], V. M. Kaspi[4], B. Knispel[8,17], K. J. Lee[3], J. van Leeuwen[15,16], A. G. Lyne[14], M. A. McLaughlin[18], X. Siemens[7], L. G. Spitler[3], A. Venkataraman[13]

ABSTRACT

In the modern era of big data, many fields of astronomy are generating huge volumes, the analysis of which can sometimes be the limiting factor in research. Fortunately, powerful data-mining techniques have been developed by computer scientists, ready to be applied to various fields. In this paper, we present a novel artificial intelligence (AI) program that identifies pulsars from recent surveys using image pattern recognition with deep neural nets—the PICS (Pulsar Image-

[1] Department of Physics and Astronomy, 6224 Agricultural Road, University of British Columbia, Vancouver, BC, V6T 1Z1, Canada; zhuww@phas.ubc.ca, berndsen@phas.ubc.ca

[2] Astronomy Department, Cornell University, Ithaca, NY 14853, USA

[3] Max-Planck-Institut für Radioastronomie, Auf dem Hügel 69, D-53121 Bonn, Gemnay

[4] Department of Physics, McGill University, Montreal, QC H3A 2T8, Canada

[5] Center for Advanced Radio Astronomy, University of Texas at Brownsville, Brownsville, TX 78520, USA

[6] NRAO, Charlottesville, VA 22903, USA

[7] Center for Gravitation, Cosmology and Astrophysics. University of Wisconsin Milwaukee, Milwaukee, WI 53211 USA

[8] Max-Planck-Institut für Gravitationsphysik, D-30176 Hanover, Germany

[9] International Centre for Radio Astronomy Research, Curtin University, Bentley, WA 6102, Australia

[10] Centre for Astrophysics & Supercomputing, Swinburne University, Hawthorn, Victoria 3122, Australia

[11] Columbia Astrophysics Laboratory, Columbia University, New York, NY 10027, USA

[12] Department of Physics and Astronomy, Franklin and Marshall College, PO Box 3003, Lancaster, PA 17604-3003, USA

[13] Arecibo Observatory, HC3 Box 53995, Arecibo, PR 00612, USA

[14] University of Manchester, Jodrell Bank Observatory, Macclesfield, Cheshire SK11 9DL, UK

[15] ASTRON, the Netherlands Institute for Radio Astronomy, Postbus 2, NL-7900 AA, Dwingeloo, the Netherlands

[16] Astronomical Institute 'Anton Pannekoek,' University of Amsterdam, Science Park 904, NL-1098 XH Amsterdam, the Netherlands

[17] Leibniz Universität Hannover, D-30167 Hannover, Germany

[18] Department of Physics, West Virginia University Morgantown, WV 26506, USA

[19] Department of Physics, Syracuse University, NY 13244 USA

[20] Naval Research Laboratory, 4555 Overlook Ave SW, Washington, DC 20375

based Classification System) AI. The AI mimics human experts and distinguishes pulsars from noise and interferences by looking for patterns from candidate plots. Different from other pulsar selection programs which use pre-designed patterns, the PICS AI teaches itself the salient features of different pulsars from a set of human-labeled candidates through machine learning. The training candidates are collected from the Pulsar Arecibo L-band Feed Array Survey. The information from each pulsar candidate is synthesized in four diagnostic plots, which consist of up to thousands pixel of image data. The AI takes these data from each candidate as its input and uses thousands of such candidates to train its ∼9000 neurons. The deep neural networks in this AI system grant it superior ability in recognizing various types of pulsars as well as their harmonic signals. The trained AI's performance has been validated with a large set of candidates from a different pulsar survey, the Green Bank North Celestial Cap survey. In this completely independent test, PICS ranked 264 out of 277 pulsar-related candidates, including all 56 previously known pulsars, to the top 961 (1%) of 90008 test candidates, missing only 13 harmonics. The first non-pulsar candidate appears at rank 187, following 45 pulsars and 141 harmonics. Performance of this system can be improved over time as more training data are accumulated. This AI system has been integrated into the PALFA survey pipeline and has discovered three new pulsars to date.

Subject headings: pulsars — stars: neutron — methods: data analysis, machine learning — techniques: image processing

1. Introduction

Recent pulsar surveys such as the Pulsar Arecibo L-band Feed Array (PALFA; Cordes et al. 2006; Kaspi 2012; Lazarus 2013) survey and the Green Bank North Celestial Cap (GBNCC; Lynch et al. 2013, K. Stovall et al. in prep.) survey are expected to find—or are already finding—hundreds of new pulsars among many millions of pulsar candidates. However, the surveys are polluted by radio frequency interference (RFI) that makes it hard to pick out the pulsars from the candidates produced using simple metrics such as the signal-to-noise ratio (SNR). Human experts can look at diagnostic plots of the candidates and identify the pulsars more successfully, but it is impractical to inspect millions of candidates that way. In this paper, we present an artificial intelligence (AI) system that emulates human experts and classifies pulsar candidates using patterns from four standard diagnostic plots—the pulse profile, time-versus-phase plot, frequency-versus-phase plot, and dispersion-measure (DM) curve (see Section 2.1 for details). This system was trained with PALFA candidates classified by human experts, its parameters were tuned using a cross-validation set of candidates, and its final performance was compared against a large set of manually identified candidates from the GBNCC survey. Both the PALFA and GBNCC candidates are generated using the *PRESTO*[1] search pipeline (Ransom 2001; Ransom et al. 2002), but other pulsar-searching pipelines, such as the one in the Einstein@Home project (Allen et al. 2013; Knispel et al. 2013), can produce these same diagnostic plots, so it is possible to apply our AI system to most pulsar surveys with little modification.

In the past, several successful candidate-sorting schemes have been developed for different surveys. Some involve graphical interfaces which allow for the interactive selection of pulsar candidates based on the pulse period and SNR (Faulkner et al. 2004). Some apply heuristic scoring algorithms to the candidate diagnostic plots, using statistical tests, curve fitting, and a graphical interface to visually inspect the distribution of pulsar scores in the scores' parameter space (Keith et al. 2009). A very effective sorting scheme was constructed (Lee 2009; Lee et al. 2013) using a combination of six carefully designed heuristic scores. One particular score compares the candidate's pulse frequency against the frequency distribution drawn from a large sample of the survey candidates; recognizing that the majority of candidates are RFI, this single histogram removes a large fraction of the repeatedly observed RFI, especially harmonics of 60 Hz. Eatough et al. (2010) improved the method of Keith et al. (2009) by applying machine learning (ML) on the heuristic scores. Instead of inspecting the score distribution by eye, they fed the scores into an artificial neural network and trained the network to classify candidates. Bates et al. (2012) expanded the number of scores used by fitting the candidate's features with different model curves, and also used a neural network to combine these scores. Another comprehensive score-based system (Kaspi 2012; Lazarus 2013) was developed and has been used to find many pulsars for the PALFA survey. Most recently, Knispel et al. (2013) designed algorithms that check for outstanding signals in some of the diagnostic plots by binning the plots with pre-designed patterns in the shape of vertical lines or area patches, and applied them in the Einstein@Home project.

Notably, these previous candidate-sifting systems use heuristically-designed functions which characterize patterns in the diagnostic plots into a set of scores. Such score-based systems have some advantages. They make good use of the candidate's properties like period, DM and computed information like the significance of the periodicity and the width/height of the summed profile. However, such systems often rely on matching candidates with some pre-designed patterns, such as a Gaussian-like peak. Some of these designed patterns do not

[1] http://www.cv.nrao.edu/ sransom/presto/

match pulsars with multiple pulse peaks well. When these designed patterns do match the candidate, they tend to average out the details and small features in the diagnostic plot. These small features can sometime be very useful in distinguishing pulsars from RFI. Some score-based system selects pulsar candidates by drawing the score distributions from known pulsars. Such system may bias against rare type of pulsars.

In contrast, the PICS AI applies image pattern recognition directly to the original diagnostic plots, and determines what patterns to match through machine learning. Using the original diagnostic plots allows the AI to utilize the detailed information in the plots. Using machine learning allows us to train the AI with a wide variety of pulsar candidates. A significant fraction of our training candidates are pulsars or their harmonics with weak, braod, or mult-peak pulse profile. As a result, the PICS AI is not only sensitive to pulsar candidates but also to their harmonics. In fact, one of the new pulsars discovered by PICS, PSR J1914+08 (Figure 2), was identified from one of its harmonics. In this case, the fundamental frequency was missed by the PALFA pipeline due to RFI. Finally, an important feature of the PICS AI is that it neglects information such as the candidate's period, DM and sigma value. Instead, it focuses on only the details in the normalized diagnostic plots. This feature makes it a good complement to the score-based systems.

This paper introduces a new approach to the candidate-sifting problem using supervised ML based on image patterns. In Section 2, we describe the candidate plots which contain the physical features distinguishing them as pulsars, and we detail the data-preparation process required to adapt these plots as inputs to the ML system. We also describe the structure of the PICS AI system. In Section 3, we detail the AI's test performance and results from classifying GBNCC data. Finally in Section 4, we discuss the results and the AI system's unique strengths and features.

2. Implementation

The diagnostic plot that a human expert relies on to identity a pulsar contains several important subplots. This is why the PICS AI is constructed with a two layer hierarchy (Figure 1). The first layer consists of a group of ML classifiers trained to look at different subplots, providing a pool of experts capable of recognizing pulsar patterns. Each classifier rates how pulsar-like a candidate is with a number between 0 (not a pulsar) and 1 (a pulsar), giving a prediction matrix that is the output of the first layer in the PICS AI. These votes are fed into a second-layer classifier which learns to properly weight these votes and forms a final consensus on how pulsar-like a candidate is.

In this section we introduce the detail implementation of the PICS AI, starting from the most important subplots that the AI uses.

2.1. Four features

Our goal is to train an AI program that mimics human experts. Here we first introduce how the PALFA pipeline finds pulsar candidates and then discuss which features human experts look for when identifying promising candidates.

The PALFA survey uses the 7-beam L-band (1.4 GHz) receiver at the Arecibo observatory. The survey takes 5-minute snapshots of the Galactic plane. In recent observations, data were taken using the Mock spectrometer, which has a 65.5 μs sampling time and 960 frequency channels covering 322.6 MHz of bandwidth. The data from each beam are analyzed using a *PRESTO* based pulsar-searching pipeline.

Pulsar radio emission is typically a broadband signal originating from kpc distances. Therefore, the signal is dispersed by the ionized interstellar medium. This causes a low-frequency signal to arrive later than one at a higher frequency. The delay between the two frequencies ν_1 and ν_2 is proportional to $\mathrm{DM}(\nu_1^{-2} - \nu_2^{-2})$ where DM, the dispersion measure, is the column density of free electrons along the line of sight. This is non-zero and remains nearly constant for a pulsar. Accordingly, *PRESTO* first searches for narrow band or non-dispersed periodic signals in the raw channeled data and removes them, since they are likely terrestrial. Then it generates time series for an array of DM values by adding time appropriate delays to each frequency channel. The range of DM searched is $0 < \mathrm{DM} < 5000\,\mathrm{pc}\cdot\mathrm{cm}^{-3}$, which easily encompasses the expected Galactic interstellar dispersion for all lines of sight in the PALFA survey (Cordes & Lazio 2001) and also maintains sensitivity to any possible highly-dispersed extragalactic radio sources (e.g. Lorimer et al. 2007; Thornton et al. 2013, L. Spitler et al. in prep.). After that it searches for periodic signals in the "de-dispersed" time series using a Fourier Transform and picks out the significant periodicity peaks in the power spectrum using harmonic summing techniques (see Ransom 2001 for details). For each candidate periodicity, *PRESTO* folds the de-dispersed time series into a 3D data cube (time interval, phase and channel frequency) using the period, and stores the folded data in a *pfd* format file together with data-descriptors such as the date and coordinates of the observation. These *pfd* files can later be converted to candidate diagnostic plots using routines in the *PRESTO* software suite.

A direct inspection of the folded 3D data array is inconvenient, so it is usually projected into several lower-dimensional plots. The *PRESTO* routine *show_pfd* is designed to display the *pfd* file as a candidate plot that contains several of these projections. Figure 2 is an example pulsar candidate plot, with the most important subplots highlighted.

1. **Summed profile:** One can sum all frequency channels and time intervals to create a summed intensity-versus-phase pulse profile. Pulse profiles of real pulsars are usually composed of one or several very narrow peaks, though there are some known pulsars with pulse profiles which are broad and/or contain multiple peaks.

2. **Time-versus-phase plot:** This plot is obtained by summing the data over the different frequency channels, leaving the pulse profile as averaged over subintervals of the observation. One or more vertical stripes in this plot indicates that a pulsed signal was observed for the duration of the scan.

3. **Frequency-versus-phase plot:** Summing the data cube over the different time intervals leaves the frequency-versus-phase plot. The presence of one or more persistent vertical lines in this subplot, as in the example, indicates a broadband signal during the pulsed emission, as expected for a pulsar candidate.

4. **DM curve:** The plotting program searches over a range of DMs around the best reported value. For each DM trial, it de-disperses the data cube accordingly and calculates the χ^2 of the de-dispersed pulse profile against a horizontal line fit. The DM curve is a plot of the trial DMs against their corresponding χ^2 values. A large χ^2 value indicates the presence of a strong periodic signal. The DM curve of a real pulsar will always peak at a non-zero value.

These are the four most important features that human experts look at when classifying candidates, and they are the inputs to our AI system.

2.2. Data preparation

The application of pattern recognition to pulsar candidates is not a trivial task. The candidate plots have different sizes and some are 2D images with thousands of pixels. For ML to work, we need to carefully prepare the data.

We extract the key feature plots from the *pfd* files. These plots consist of 1D data arrays (summed profile or DM curve) and 2D data (time/frequency-versus-phase plot). The sizes of the data arrays vary from candidate to candidate. For example, some candidates have 64 phase bins, while others may have only 32 bins; some candidates have 50 time steps, while others have 100. This is a result of the search pipeline, which uses fewer phase bins and more time steps for short period pulsars, and more phase bins with fewer time steps for long period pulsars—in either case, the number of pulses coherently added in each time interval is roughly the same. For ML to work on these plots, the features should have the same size and scale, so we down-sample or interpolate the data to a uniform size: 64 bins for the summed profile, 64×64 (or 48×48 depending on the classification algorithm, see Table 1 for details) bins for the time/frequency-versus-phase plot, and 60 bins for the DM curve. Piecewise linear interpolation was used for the 1D data and spline interpolation[2] for the 2D data. We also normalize the data to zero median and unit variance to remove the absolute scale of the plots. For the 2D image arrays, normalization is performed line-by-line along the phase axis, which removes instrumental variations over the course of the observation and across the observing band, but maintains the variance in signal across the phase—this should be dominated by the pulsar signal.

Once the plots are resized and normalized, we can use them to train a ML system. Depending on the AI algorithm, however, a large number of input parameters may slow down the training process. In these cases the number of internal parameters requiring training is proportional to the size of the input data, so to speed up the computation, the image features (time/frequency versus phase plot) are characterized by a principal component analysis[3] (PCA). The PCA algorithm uses a singular value decomposition to compute a limited set of basis images from the training data. An unknown candidate's data, then, has a new representation as the set of eigenvalues resulting from the projection onto these basis images. In this way, we can compress the image features from thousands of numbers per candidate to only 24 numbers[4], greatly reducing the number of parameters required in a fit. Figure 3 shows the original 2D images of a pulsar candidate and their reconstructions from the most significant 24 PCA components. One can see that the PCA reconstructions capture the important features in the images, especially the vertical stripes. By filtering out the weaker PCA components we also reduce the noise level in the reconstructed image. Some image classifiers trained much faster with PCA-compressed features and perform as well as those trained with full-sized images. However, we did not apply the PCA-compression for the deep-neural-net classifiers because they perform significantly better without it.

One last challenge was to prevent the AI from developing any phase-related bias. In principle, a candidate's pulse may appear at any phase, but in practice many candidates peak at phase 0 or 0.5. We found that an earlier version of the AI failed to detect some

[2] The 2D interpolation routine in scipy.ndimage.interpolation.map_coordinates was used.

[3] `http://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

[4] The number of singular vectors used in the decomposition is a free parameter in the AI system, but is fixed by optimizing the performance in cross-validation tests.

good candidates that peaked away from phase 0 or phase 0.5. We resolved this problem by shifting candidates' strongest peaks to phase 0.5 before feeding them to the AI. The resulting AI was tested with candidates of random phase and showed no sign of bias.

In summary, the data-preparation procedure for all candidates fed into the AI system involves rescaling the data to zero median and unit variance, shifting the peak intensity to a phase of 0.5, down-sampling or interpolating onto a standardized grid and, optionally, applying PCA.

2.3. Two-Layer AI: a committee of experts

The first layer of PICS (see Figure 1) uses a combination of two ML algorithms on each of the four subplots, giving eight ratings in total. The set of ML algorithms includes artificial neural networks (ANN; see Section 2.4 for detail), convolutional neural networks (CNN[5]; see Section 2.5 for detail) and support vector machine (SVM[6]; an algorithm that finds a direction in the parameter space on which the distance between the two classes of data points are maximized; see Chang & Lin (2011) and Pedregosa et al. (2011) for the implementation details). The choice of which algorithms to use was determined both by their individual performance and their benefit to the overall performance. It should be noted that the combinations were the result of extensive testing, including using other standard ML algorithms such as decision trees. In the end, PICS uses an ANN with one hidden layer of logistic units and a radial kernel SVM classifier on each of the 1D subplots (pulse profile, DM curve), and it uses a CNN and SVM on each of the 2D subplots (pulse interval versus phase, pulse frequency versus phase).

In the second layer, we combine the scores from the eight first-layer classifiers using another ML classifier (Figure 1). Several algorithms are appropriate for this purpose, and we tested logistic regression (LR), ANN and SVM algorithms. The best performance was from a simple LR with L2 penalty, and this option was chosen for the second layer of the PICS system. The LR algorithm assigns each of the first-layer scores (x_i) a weight (w_i), computes their weighted sum, and converts this sum to a probability using the logistic function

$$P = \frac{1}{1 + e^{-\sum_i w_i x_i}} \, . \tag{1}$$

The LR algorithm finds the best set of w_i that minimize the classification errors in the

[5] `https://github.com/aberndsen/NeuralNetwork`

[6] `http://scikit-learn.org/stable/modules/svm.html`

training data. With L2 penalty it also try to find a small $\sum_i wi^2$ and thus prevent the weights w_i from growing too large.

The classifiers in both layers of the system all have internal design parameters that need to be fixed. This is accomplished by splitting the labelled PALFA data into a training set and a test, or cross validation, set that the ML system has yet to see. In searching over the parameter space, we maximize the F_1 score on the cross validation set. The F_1 score is a performance metric commonly used in machine learning, defined as the harmonic average of precision p (the ratio of true pulsars to the total number of candidates ranked as a pulsar) and completeness c (the ratio of ranked true pulsars to the total number of pulsars in the training set)

$$F_1 = \frac{2pc}{p+c}\,. \tag{2}$$

The optimized parameters of the eight classifiers and their test performance on each feature plot is listed in Table 1. The performance is also depicted in Figure 1, which shows the overall structure of the PICS system.

Maximizing the precision alone would result in a cautious AI system which would miss a lot of true pulsars, while a system designed to maximize completeness would bring in a lot of false positives. When training the PICS, we preferred a balanced AI that maximizes F_1. This is because we want to improve the AI's completeness when we add more varieties of pulsars to the training data and we also want to improve its precision when we add more RFI candidates, to emphasize on one metric will hinder our ability to improve the other. When applying the AI in practice, we can adjust its completeness and precision to our needs by changing the cut on P, the AI probability score.

2.4. The neural networks

In the first layer of PICS, we used the ANN for the 1D subplots and the CNN for the 2D subplots. In this Section, we briefly introduce some terminology common to the understanding of both these neural networks.

Biologically, a neural network is a collection of neurons connected by synapses, where individual neurons respond, or "fire", to different inputs. An artificial neural network is the computer analog, modelled as a function of many inputs (synapses) to produce a single output. These functions are often called activation functions h, and typically have finite range for classification purposes. The most commonly used activation functions are the logistic (sigmoid) function $h(x) = 1/(1 + \exp(-x))$ and the hyperbolic tangent function $h(x) = \tanh(x)$. The former maps any float input to a number between 0 and 1, the latter maps to a number between -1 and 1.

In the ANN, these neurons are distributed among different layers. All neurons in a single layer l receive the same inputs from the previous layer $\mathbf{a}^{(l-1)}$, but weights the signals with a unique set of parameters ($\mathbf{w}_i^{(l)}$ for the i^{th} neuron in the l^{th} layer). Thus neurons in the same layer "fire" under different circumstances as determined by their weights $\mathbf{w}_i^{(l)}$ and bias b_i, $h(\mathbf{w}_i^{(l)} \cdot \mathbf{a}^{(l-1)} + b_i)$. The weights $\mathbf{w}^{(l)}$ connect the outputs of a previous layer in the ANN to the input of the next, serving the role of the synapses in the neural network. A given layer with $N_{(l)}$ neurons accepting $N_{(l-1)}$ inputs has $(N_{(l)} + 1) \times N_{(n-l)}$ weights, or synapses, which makes the number of synapses in a given network much larger than the number of neurons. A synapse, however, controls how much weight is given to a single input, so the collection of synapses forms the pattern which will produce an activation. Since there are so many synapses in an ANN, there are a lot of patterns that the system can respond to. In practice the patterns are not known, so the weights are initialized randomly and are determined through the training process.

We train the neural net with manually labeled candidates through back-propagation. The goal of training is to minimize the difference between the neural net output and the human classifications y (0 for RFI and 1 for pulsar) of the candidates. This difference, the prediction error $\delta^{(l)} = a^{(l)} - y$, can be formed for each layer l, and the correction to the j^{th} weight in the i^{th} neuron $\Delta_{i,j}^{(l)}$ is determined as a function of $\mathbf{a}$ and $\delta_j^{(l+1)}$. The exact functional form depends on the choice of activation function and error function. In practice, $\Delta_{i,j}^{(l)}$ is often averaged over an ensemble of candidates, since training is done in batches for computational efficiency.

2.5. The convolutional neural network

The best individual classifier in the collection of experts is the convolutional neural network trained on the frequency versus phase plot. This deep, 5-layer network is similar to LeNet-5 (LeCun et al. 1998), a system proven to be very successful in recognizing handwritten digits. This CNN also represents the state of the art in machine learning and, as such, warrants a detailed description of its structure and an explanation of its superior performance in the candidate-selection process.

In the first layer of the CNN, the input image is fed to groups of neurons with shared weights. We scan the input image with a sliding window to get a set of sub-images. The neurons with shared weights each look at one of these sub-images and become active if certain feature as defined by the shared weights is detected from it. The different groups of neurons are used to detect different features. This process can also be viewed as convolving the input image with a set of small image kernels, or features, and then applying the hyperbolic tangent activation function to form a set of feature maps. Specifically, the k^{th} feature map is given by

$$h^k = \tanh\big((W^k * x) + b\big) \ , \tag{3}$$

where W^k is the feature kernel, x is the input image, and b is a bias term. The number of kernels, k, is a free parameter in the system, and it determines the richness in the representation of the data. The second layer of the CNN is a *max-pooling* process. This is a form of down-sampling which bins the 2D feature maps and chooses the maximum value within each bin. The main advantage is a reduction in computational complexity in subsequent layers, though pooling also has the advantage of introducing translation invariance of the feature W^k across the bin (Boureau et al. 2010). These two steps are illustrated in Figure 4.

The output of the max-pooling layer is fed into another convolution layer for feature detection and yet another max-pooling layer. This output is fed into the fifth and final layer, a traditional, fully-connected ANN. While the previous four layers function to locate small-scale features across the input images, the final layer combines this information to detect large-scale features and develop a global understanding of the original image.

The CNN configurations for both the time-versus-phase plot and frequency-versus-phase plot were the same. In the first step (Figure 4 left panel), both CNNs take input images down-sampled to 48×48 pixels, convolve them with 20 different 16×16 image kernels, producing 20 feature maps of size 33×33. The subsequent max-pooling layer (Figure 4 right panel) divides each map into 3×3 boxes and compresses the convolved image to a size 11×11 by taking the maximum in each box. The second convolution layer convolves 50 8×8 image kernels shared across the 20 feature maps, resulting in 50 different 4×4 feature maps. A second 2×2 max-pooling layer further compresses each feature map to an images of size 2×2. The result is an array of 50×4 numbers characterizing the local, kernel-sized features of the original image. The final layer is a traditional, fully-connected artificial neural network consisting of 500 hidden logistic units which take these 200 numbers to compute one final score.

There are 8820 artificial neurons distributed through the five layers of the CNN including the ones in the image kernels. All neurons in this network are hyperbolic tangent functions except the one that forms the final output, which is chosen to be a sigmoid function in order to map onto the classification labels 0 and 1. The training of the neural net is to let these synapses learn and store the patterns distinguishing pulsars from RFI. As an example, the last layer has 10^5 weights connecting the 200 outputs of the second last layer to the 500 hidden neurons in the last layer. Each hidden neuron takes, as input, a weighted sum of the 200 outputs of the previous layer. All of the connection parameters are initialized randomly, and then updated through the process of back-propagation.

The structure of the CNN is determined by the choice of image size, kernel size, number of kernels, pooling size, and neural network size, these parameters are often called neural net design or hyper-parameters. The best CNN design, as described above, is determined through cross-validation tests. The labelled data are randomly shuffled and split into a training set formed from 60% of the candidates and a validation set from the remaining 40%, and a large grid search is performed. For each choice of design parameters the CNN is trained and the performance is characterized by computing its F_1 score (see Section 2.3) on the validation set. The CNN we used gives $F_1 = 92\%$ when trained and tested on the time-versus-phase plots, and $F_1 = 94\%$ for the frequency-versus-phase plots (Table 1). After optimizing the CNN design, the final AI is trained with all training candidates in order to maximize its sensitivity. The final performance of the PICS system is tested with a set of completely independent GBNCC data (see Section 3).

This CNN performs well on the frequency-versus-phase plot because a large fraction of RFI is narrow-band emission. For continuous emission this RFI shows up as a horizontal line, while burst-like or periodic RFI appears as a small dot. This is opposed to the broadband pulsed signal of a pulsar, which shows up as a vertical stripe. Since the CNN excels at detecting small-scale features, it can easily detect this form of RFI. A second reason this classifier is the best discriminant of pulsars is simply because there is more information in the 2D plots than the 1D plots. For similar reasons, the algorithm can recognize and reject burst-like RFI or signals that drift around in phase using the time-versus-phase plot.

This deep neural net is implemented using *Theano* (Bergstra et al. 2010), a computation library for python which compiles numerical expressions to run efficiently on either CPU or GPU architectures.

3. Results

We trained the PICS AI system with 3756 labelled PALFA candidates, among them 1659 are pulsars and their harmonics and 2097 are non-pulsars. These are only few thousands of known pulsar/harmonic candidates in PALFA but millions of non-pulsars. We picked similar numbers of pulsars and non-pulsars from PALFA candidates classified by other human experts. The pulsar candidates include unique pulsars and also the same pulsar with different beam offsets. To ensure that the AI is sensitive to both pulsars and their harmonics, $\sim 40\%$ of the training pulsar candidates are harmonics of known pulsars. We also manually rated each of the four features for every candidate. This is because some RFI candidates may have a good pulse profile or DM curve, and we want to inform the AI that those are good features that resembles a pulsar even when the candidate itself is not pulsar.

To ensure that the trained classifiers can work on unseen data and to determine their best initial parameters, we performed the standard cross-validation tests (Section 2.5). We shuffled and split the 3756 candidates into two groups, 60% for training and 40% for validating. Furthermore, we repeated this shuffle, split, train and validate procedure 10 times to make sure their performance on validation data is reliable and repeatable. The individual first-layer classifiers score in the range 86–94% F_1 (with less than 1% RMS each) on the validation data (Table 1). The second layer of PICS, a LR algorithm that combines the scores from first-layer classifiers to form a consensus vote, scored an average of 96% F_1 (8% misses, 1% false positives) on the validation data.

To further measure the AI's performance and determine whether it can be generalized to other surveys, we applied it to a large set of GBNCC candidates that was never seen by the AI system during training. Like PALFA, the GBNCC survey uses the *PRESTO* search pipeline and generates candidates in *pfd* format, but this survey was conducted using the Green Bank Telescope (GBT) instead of Arecibo, and its receiver operates at 350 MHz instead of 1400 MHz, so the RFI environments of the two surveys are expected to be different. We applied the AI to 90008 manually labelled GBNCC candidates. An initial test showed that the AI can sort all 56 pulsars in this data set to the top 3.8%, and 68% of the pulsars to the top 0.16%. The first RFI appeared in rank 136, following 29 pulsars and 106 harmonics.

An inspection of the false positive candidates revealed many to be harmonics of the 60 Hz power signal (see Figure 6 for an example). Table 2 lists the 6 most-populated frequency bins in a frequency histogram of all candidates. Although these frequencies comprise <1% of the frequency domain searched by *PRESTO*, 24519 (roughly 27%) of the candidates fall into these frequency bins. It is standard practice to remove harmonics of 60 Hz for North American surveys; in spite of these efforts, a significant amount of such RFI remains. Using the image patterns in the diagnostic plots, our AI was able to reject the vast majority of this RFI, with the remaining false positives strongly resembling pulsars (see Figure 6 for an example).

In light of this observation, we adjusted the scores of all candidates using a Bayesian prior on the pulse frequency f to reduce the score for the false-positive candidates in contaminated frequency bins:

$$P(p|f) = \frac{P(p)}{P(p) + [P(f|r)/P(f|p)]\,P(r)}\,. \tag{4}$$

Here $P(p)$ is the probability of being a pulsar as previously scored by PICS, $P(r) = 1 - P(p)$, while $P(f|r)$ and $P(f|p)$ are the probability density functions, or likelihoods, of RFI and pulsars in frequency. $P(f|r)$ can be well-approximated by sampling the frequency distribution of all test candidates, which is dominated by non-pulsar signals, especially 60 Hz and its harmonics (Table 2). We approximate the prior distribution $P(f|p)$ by median-filtering $P(f|r)$, which removes spikes caused by RFI and leaves a distribution that reflects the survey sensitivity. At the harmonics of 60 Hz $P(f|r) \gg P(f|p)$, such that $P(p|f) < P(p)$, and the AI score is reduced. Away from the few affected frequency bins (Table 2), $P(f|r) \simeq P(f|p)$ and the AI score is lightly affected, with $P(p|f) \simeq P(p)$.

The frequencies of the test candidates are binned from 0 Hz to 2000 Hz in steps of 0.5 Hz. Due to the limited bin size, all slow pulsars (f <1 Hz) are binned to the lowest two bins, along with a lot of RFI. Since the underlying structure of the prior distributions is not properly resolved in this domain, we did not apply the Bayesian rules to candidates with f <1 Hz.

Figure 5 shows the distribution of GBNCC test candidates when sorted by $P(p|f)$, and summarizes the performance of PICS. After applying the Bayesian rule (Equation (4)), all 56 pulsars and 208 of the 221 harmonics were sorted and placed into the top 961 (1%) of the 90008 candidates by the PICS system. The first RFI candidate appeared rank 187, following 45 pulsars and 141 harmonics.

A blind cross-validation test was performed using pulsars and RFI candidates collected from the PALFA survey but never seen by the AI. We found that the same Bayesian rule (Equation (4)) that worked for the GBNCC test also clearly reduced the false positive rate in this cross-validation test. This indicates that the Bayesian rule can be generalized to surveys other than GBNCC. Caution must be advised, though, since a weak pulsar candidate falling in an RFI-contaminated frequency bin (Table 2) will be down-weighted by the Bayesian prior and could be rejected. Fortunately, the chance of this happening is very small, since only a few frequency bins ($\sim 0.7\%$, Table 2) are affected significantly by the prior. In light of this, the Bayesian-rule RFI rejection is included only as an option in PICS and can be turned off.

Despite the size and complexity of PICS, candidate classification is a fast process because most of the individual classifiers simply apply dot products. The hard part of the computation was already done during the training phase. It took $\sim$ 45 minutes to classify the 90008 GBNCC candidates using a cluster of 24 2.7GHz CPUs, $\sim$ 0.7 CPU second per candidate, though most of the time was spent on disk I/O and not all CPUs were used at 100% capacity. Using the same computer cluster, the AI would be able to classify a million candidates in several hours.

The PICS AI has been integrated into the PALFA pipeline, such that we can query and sort candidates using the AI rating on the *cyberska.org* web platform. With the help of the

AI, we have found many promising candidates over several weeks, three of which have been confirmed as new PALFA discoveries.

- J1914+08 (Figure 2) is a 147 ms pulsar with a DM of 289.2 pc/cm^3.
- J1938+20 (Figure 7) is 2.6 ms MSP with a DM of 237.1 pc/cm^3.
- J1901+02 (Figure 8) is a 885.24 ms pulsar with a DM of 403 pc/cm^3.

These pulsars are now subjects of on-going timing observations by Arecibo or Jodrell Bank Observotaries.

The AI source code and a trained classifier are accessible on github[7].

4. Discussion

PICS scores the candidates with a number between 0 and 1, a higher score corresponding to a more pulsar-like candidate. When tested with the GBNCC data, the AI placed 100% of the pulsars and 94% of the harmonics in the top 1% (Figure 5) of ranked candidates. By rejecting 99% of the candidates, PICS can improve the speed of the candidate classification process for a human expert by a factor of $\gtrsim$100. If we combine the AI with other ratings or scores such as the signal-to-noise ratio, the sorting efficiency can be further improved, making it possible for a few human experts to sift through millions of candidates. Such a system can be very useful for existing and future pulsar surveys.

In the GBNCC test, PICS AI showed very good performance. However, in the top 1% of the test candidates, there were still a few pulsars ranked below hundreds of non-pulsars. These were pulsars with broad pulses and low signal-to-noise ratios, and were surpassed by some strong RFI signals that also have broad features. It seems that the PICS AI could benefit from training with more pulsar/RFI training candidates that have broad pulse profiles.

Despite being limited by the quantity and diversity of the training data, the use of image-pattern-based machine learning in PICS is a novel idea and has some advantages:

The AI uses a deep neural network composed of many neurons and hidden weights which have the capability to recognize subtle or complex features. By gathering more

[7] `https://github.com/zhuww/ubc_AI`

human-identified candidates from surveys, we will be able to further improve the AI system. Specifically, if a survey encounters a new kind of RFI that our system or a method based on analytical heuristics fails to reject, we can improve our AI by incorporating examples of these RFI into our training data and retrain the system. Future improvements to PICS include expanding the training set to candidates from both PALFA and GBNCC surveys, and providing input capability for survey products from other search software. Rather broadly, by increasing the pool of training candidates, we can improve the accuracy of the PICS AI.

The PICS AI makes a classification based solely on image patterns in the diagnostic plots. A caveat/feature is that, because these plots are re-binned and normalized to unit variance, the AI ignores the DM, period and pulse amplitude differences between different candidates. Although those lost information may be useful in some rare cases (e.g. ruling out millisecond candidates with high DM based on DM smear timescale), this procedure forces PICS to learn the universal features of pulsars (broadband, pulsed and dispersed emission) and helps it adapt to other surveys in which sensitivity in candidate's DM, frequency, and signal strength may be different. PICS is also forced to rely less on information that other score-based systems rely on, making it a good complement to them.

Being trained with many harmonic candidates with weak or multi-peak pulse profiles, the PICS AI is also very good at finding these weak or harmonic candidates. PICS has discovered three new pulsars since being integrated into the PALFA survey pipeline. PSR J1938+20 was discovered as a very weak candidate (5.0σ according to the version of *PRESTO* in the PALFA pipeline at the time of discovery; Figure 7). This makes it the least significant pulsar candidate confirmed by the PALFA survey. Such a candidate probably would not have been found by candidate sorting method that rely heavily on the significance of the signal. PSR J1914+08 was discovered as two of its multi-peaked harmonics, not as a candidate at its actual frequency. Other candidate selection methods that rely on fitting the pulse profile with a single Gaussian curve would likely down-rate these harmonics. The last pulsar, J1901+02, was discovered by the AI as one of the few top candidates.

Ultimately—and with enough resources—we envision a system that adaptively trains the AI and re-scores the survey candidates on the fly while human experts are classifying them. Such an "online" system will ensure portability of the AI when the RFI environment of the telescope slowly changes over time. Remarkably, owing to the nature of the ML algorithms employed, this does not require coding new heuristics to characterize any emerging forms of RFI. One of challenges in implementing an online learning system is that the training data will be very imbalanced as pulsar surveys generate a lot more RFI than pulsars. Lyon et al. (2013) has explored the performance of some existing online ML algorithms in dealing with imbalanced data stream from pulsar survey and found promise in these methods. Whether PICS could be adapted into a successful online learning system will be explored in future work.

Pulsar research at UBC is supported by NSERC Discovery and Special Research Opportunity grants and a Discovery Accelerator Supplement, by CANARIE and by the Canada Foundation for Innovation. Pulsar research at McGill is supported by an NSERC Discovery grant and Accelerator Supplement, by the Canada Research Chair program, by the Centre de Recherche Astrophysique du Québec, by the Canadian Insitute for Advanced Research, and by the Lorne Trottier Chair in Astrophysics and Cosmology. This work was also supported by US National Science Foundation (NSF) grants 1104902, 1104617, and 1105572. LGS gratefully acknowledges the financial support by the European Research Council for the ERC Starting Grant BEACON under contract no. 279702.

REFERENCES

Allen, B., et al. 2013, ApJ, 773, 91

Bates, S. D., et al. 2012, MNRAS, 427, 1052

Bergstra, J., et al. 2010, in Proceedings of the Python for Scientific Computing Conference (SciPy), oral Presentation

Boureau, Y.-L., Ponce, J., & LeCun, Y. 2010, in International Conference on Machine Learning, 111–118

Chang, C.-C., & Lin, C.-J. 2011, ACM Transactions on Intelligent Systems and Technology, 2, 27:1, software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`

Cordes, J. M., & Lazio, T. J. W. 2001, ApJ, 549, 997

Cordes, J. M., et al. 2006, ApJ, 637, 446

Eatough, R. P., Molkenthin, N., Kramer, M., Noutsos, A., Keith, M. J., Stappers, B. W., & Lyne, A. G. 2010, MNRAS, 407, 2443

Faulkner, A. J., et al. 2004, MNRAS, 355, 147

Kaspi, V. M. 2012, in American Astronomical Society Meeting Abstracts, Vol. 219, American Astronomical Society Meeting Abstracts #219, 237.12

Keith, M. J., Eatough, R. P., Lyne, A. G., Kramer, M., Possenti, A., Camilo, F., & Manchester, R. N. 2009, MNRAS, 395, 837

Knispel, B., et al. 2013, astro-ph.GA:1302.0467

Lazarus, P. 2013, in IAU Symposium, Vol. 291, IAU Symposium, 35–40

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. 1998, Proceedings of the IEEE,, 86(11), 2278

Lee, K. J. 2009, PhD thesis, Peking University

Lee, K. J., et al. 2013, MNRAS, 433, 688

Lorimer, D. R., Bailes, M., McLaughlin, M. A., Narkevic, D. J., & Crawford, F. 2007, Science, 318, 777

Lynch et al. 2013, in IAU Symposium, Vol. 291, IAU Symposium, 41–46

Lyon, R. J., Brooke, J. M., Knowles, J. D., & Stappers, B. W. 2013, astro-ph.IM:1307.8012

Pedregosa, F., et al. 2011, Journal of Machine Learning Research, 12, 2825

Ransom, S. M. 2001, PhD thesis, Harvard University

Ransom, S. M., Eikenberry, S. S., & Middleditch, J. 2002, Astron. J., 124, 1788

Thornton, D., et al. 2013, Science, 341, 53

This preprint was prepared with the AAS LaTeX macros v5.2.

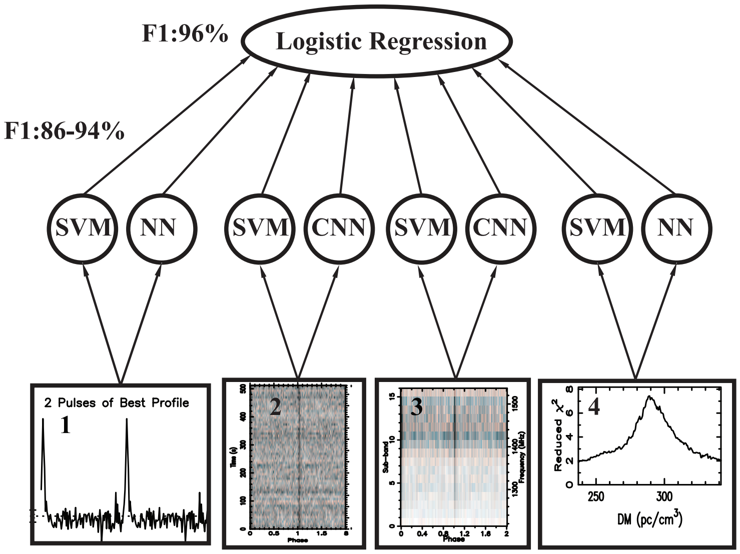
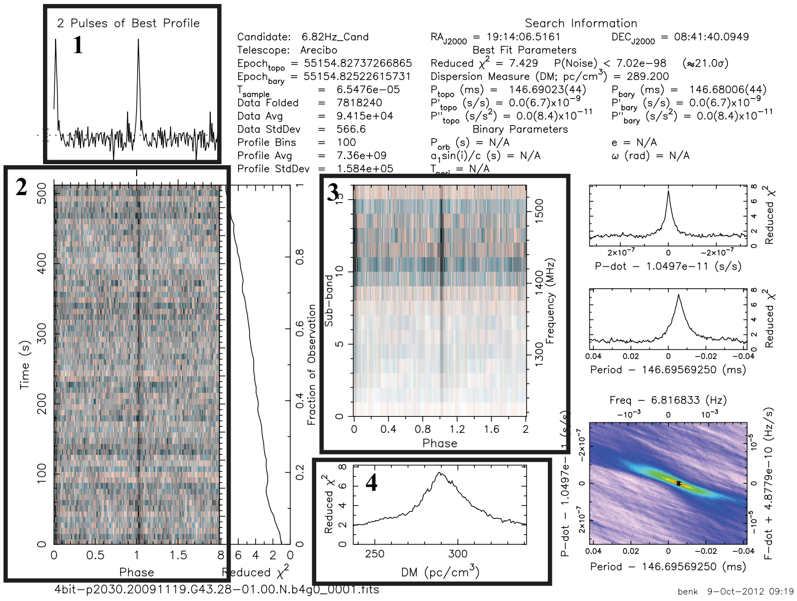


Fig. 1.— The first layer of classifiers learns how to rate each feature to ~90% F_1, and the second layer learns how to classify candidates based on the output of the first layer. (SVM: support vector machine, NN: Neural Network, LR: logistic regression and adaboost are machine learning algorithms. See Section 2.3 for the definition of F_1.)



```
                                   Search Information
Candidate:  6.82Hz_Cand            RA_J2000 = 19:14:06.5161          DEC_J2000 = 08:41:40.0949
Telescope:  Arecibo                         Best Fit Parameters
Epoch_topo = 55154.82737266865     Reduced χ^2 = 7.429    P(Noise) < 7.02e−98    (≈21.0σ)
Epoch_bary = 55154.82522615731     Dispersion Measure (DM; pc/cm^3) = 289.200
T_sample        =  6.5476e−05      P_topo (ms) =  146.69023(44)      P_bary (ms) =  146.68006(44)
Data Folded     =  7818240         P'_topo (s/s) = 0.0(6.7)x10^−9     P'_bary (s/s) = 0.0(6.7)x10^−9
Data Avg        =  9.415e+04       P''_topo (s/s^2) = 0.0(8.4)x10^−11 P''_bary (s/s^2) = 0.0(8.4)x10^−11
Data StdDev     =  566.6                    Binary Parameters
Profile Bins    =  100             P_orb (s) = N/A                   e = N/A
Profile Avg     =  7.36e+09        a_1sin(i)/c (s) = N/A             ω (rad) = N/A
Profile StdDev  =  1.584e+05       T_peri = N/A
```

Fig. 2.— *prepfold* diagnostic plot for PICS-discovered pulsar PSR J1914+08. The four key subplots of the PICS AI system are highlighted: 1. summed pulse profile 2. time-versus-phase plot 3. frequency-versus-phase plot 4. DM curve. For subplot 1, 2 and 3 the pulse phase are wrapped around twice to show two duplicated pulses.

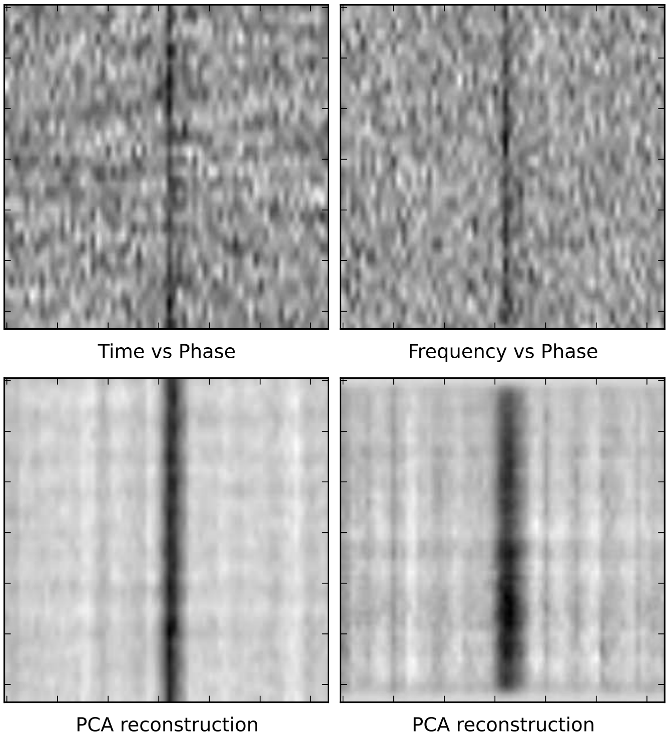


Fig. 3.— Top: The original time-versus-phase and frequency-versus-phase plots from a pulsar candidate. Bottom: The PCA-reconstructed plots from the top 24 PCA components.

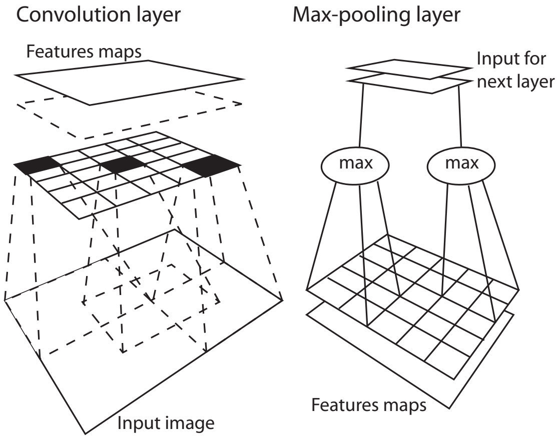


Fig. 4.— A schematic of the first two layers in the convolutional neural network. Left: convolution layer; from bottom up, the input image is convolved with a set of image kernels, forming feature maps that show the presence of certain features in different positions of the image. right: max-pooling layer; the feature maps are compressed to smaller size by taking in only the maximum values of adjacent pixels. The PICS CNN consists of two sets of alternating convolution and max-pooling layers, and a final artificial neural network layer.

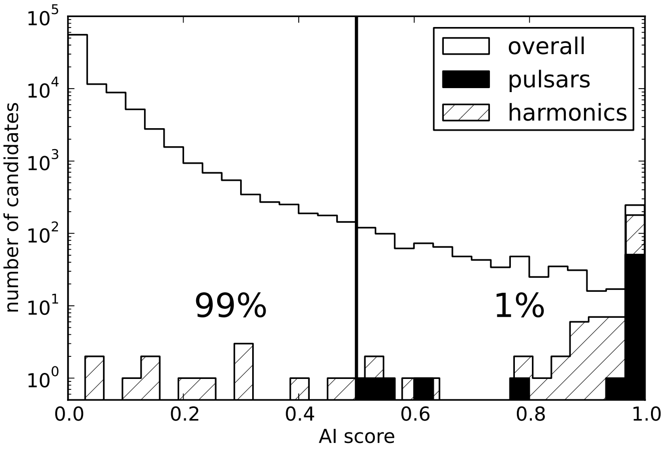
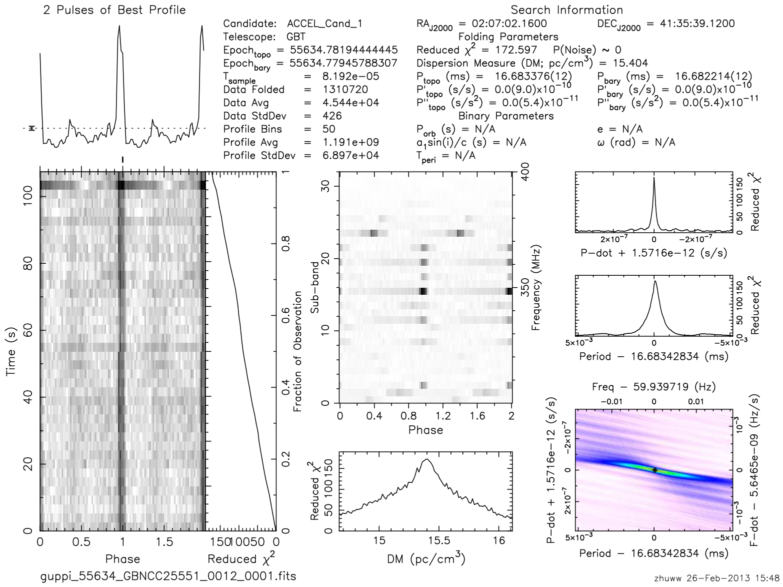
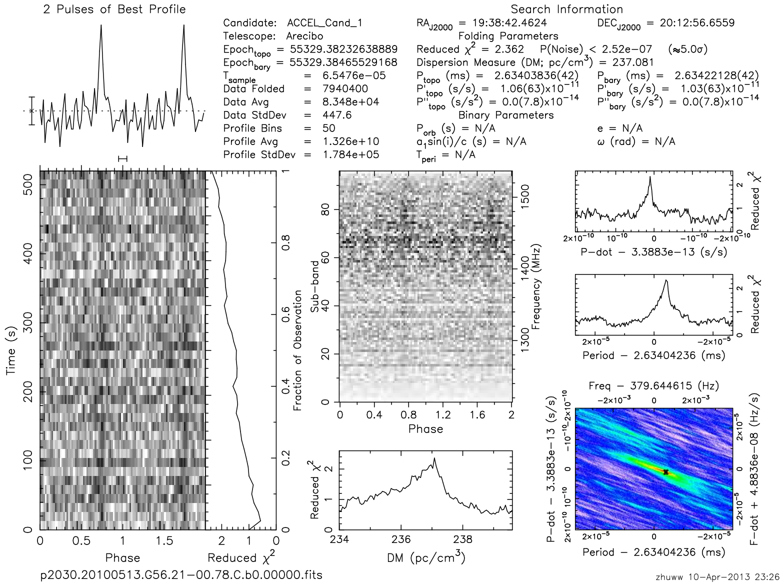


Fig. 5.— The unfilled histogram is the distribution of AI scores of 90008 GBNCC candidates. The filled histogram is the AI score distribution of known pulsars. The hatched histogram is that of the harmonics of known pulsars.



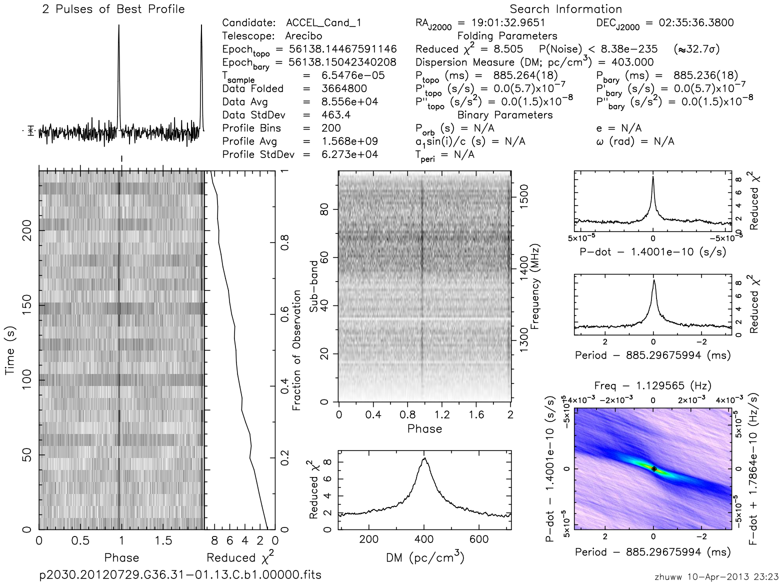
```
                                                              Search Information
Candidate:  ACCEL_Cand_1            RA_J2000 = 02:07:02.1600                DEC_J2000 = 41:35:39.1200
Telescope:  GBT                                 Folding Parameters
Epoch_topo = 55634.78194444445      Reduced χ^2 = 172.597    P(Noise) ~ 0
Epoch_bary = 55634.77945788307      Dispersion Measure (DM; pc/cm^3) = 15.404
T_sample        =  8.192e-05        P_topo (ms) =  16.683376(12)            P_bary (ms) =  16.682214(12)
Data Folded     =  1310720          P'_topo (s/s) = 0.0(9.0)x10^-10         P'_bary (s/s) = 0.0(9.0)x10^-10
Data Avg        =  4.544e+04        P''_topo (s/s^2) = 0.0(5.4)x10^-11      P''_bary (s/s^2) = 0.0(5.4)x10^-11
Data StdDev     =  426                          Binary Parameters
Profile Bins    =  50               P_orb (s) = N/A                         e = N/A
Profile Avg     =  1.191e+09        a_1sin(i)/c (s) = N/A                   ω (rad) = N/A
Profile StdDev  =  6.897e+04        T_peri = N/A
```

Fig. 6.— An example of a ∼60 Hz RFI signal that was ranked higher than the weakest four pulsars in the GBNCC test data.



```
                                                              Search Information
Candidate:  ACCEL_Cand_1            RA_J2000 = 19:38:42.4624                DEC_J2000 = 20:12:56.6559
Telescope:  Arecibo                             Folding Parameters
Epoch_topo = 55329.38232638889      Reduced χ^2 = 2.362    P(Noise) < 2.52e-07    (≈5.0σ)
Epoch_bary = 55329.38465529168      Dispersion Measure (DM; pc/cm^3) = 237.081
T_sample        =  6.5476e-05       P_topo (ms) =  2.63403836(42)           P_bary (ms) =  2.63422128(42)
Data Folded     =  7940400          P'_topo (s/s) =  1.06(63)x10^-11        P'_bary (s/s) =  1.03(63)x10^-11
Data Avg        =  8.348e+04        P''_topo (s/s^2) = 0.0(7.8)x10^-14      P''_bary (s/s^2) = 0.0(7.8)x10^-14
Data StdDev     =  447.6                        Binary Parameters
Profile Bins    =  50               P_orb (s) = N/A                         e = N/A
Profile Avg     =  1.326e+10        a_1sin(i)/c (s) = N/A                   ω (rad) = N/A
Profile StdDev  =  1.784e+05        T_peri = N/A
```

Fig. 7.— The *PRESTO* plot of PSR J1938+20.



Search Information

Candidate: ACCEL_Cand_1
Telescope: Arecibo
$\mathrm{Epoch_{topo}}$ = 56138.14467591146
$\mathrm{Epoch_{bary}}$ = 56138.15042340208
T_{sample} = 6.5476e−05
Data Folded = 3664800
Data Avg = 8.556e+04
Data StdDev = 463.4
Profile Bins = 200
Profile Avg = 1.568e+09
Profile StdDev = 6.273e+04

$\mathrm{RA_{J2000}}$ = 19:01:32.9651 $\mathrm{DEC_{J2000}}$ = 02:35:36.3800

Folding Parameters

Reduced χ^2 = 8.505 P(Noise) < 8.38e−235 (≈32.7σ)
Dispersion Measure (DM; pc/cm^3) = 403.000
P_{topo} (ms) = 885.264(18) P_{bary} (ms) = 885.236(18)
P'_{topo} (s/s) = 0.0(5.7)x10^{-7} P'_{bary} (s/s) = 0.0(5.7)x10^{-7}
P''_{topo} (s/s^2) = 0.0(1.5)x10^{-8} P''_{bary} (s/s^2) = 0.0(1.5)x10^{-8}

Binary Parameters

P_{orb} (s) = N/A e = N/A
$a_1\sin(i)/c$ (s) = N/A ω (rad) = N/A
T_{peri} = N/A

Fig. 8.— The *PRESTO* plot of PSR J1901+02.

Table 1. Classifier parameters and test performance.

classifier	feature (size[a])	nodes[b]	γ[c]	C[d]	F_1[e]
NN	summed profile (64)	25	–	2	0.86
SVM	summed profile (64)	–	0.08	1	0.93
CNN	time vs phase (48×48)	500	–	1	0.92
SVM	time vs phase (64×64)	–	0.01	1	0.88
CNN	frequency vs phase (48×48)	500	–	1	0.94
SVM	frequency vs phase (64×64)	–	0.001	24	0.88
NN	DM curve (60)	9	–	10	0.91
SVM	DM curve (60)	–	0.2	25	0.91
LR	layer-one pool (8)	–	–	0.1	0.96

[a] The input features were down-sampled or interpolated to a uniform size.

[b] Number of nodes in the hidden layer of the neural network (NN). For the CNNs, only the number of nodes of the last hidden layer is listed. See Section 2.5 for more details.

[c] γ is the radius parameter for the *rbf* kernel function used in the SVM.

[d] The C parameter controls the regulation in NN and SVM, where the regulation penalty is proportional to 1/C.

[e] The average F_1 scores of the classifiers from 10 trials. In each trial, we randomly shuffle our known candidates before splitting them into training and testing data. The standard deviations of the F_1 scores are <0.01.

Table 2. The 60 Hz RFI and its harmonics.

frequency range (Hz)[a]	Counts[b]	Percentage[c]
18−19	2101	2.3%
58−62	12941	14.4%
118−121	4480	5.0%
139−140	2157	2.4%
179−181	1744	1.9%
239−241	1096	1.2%

[a] Frequency ranges for the top 6 bins in a histogram of candidates.

[b] Number of candidates within the given frequency range.

[c] Percentage of candidates within the given frequency range.