Analysis and Application of Automated Methods for Detecting Pulsars in the Green Bank Telescope 350MHz Drift-Scan Survey

by

David Paul Smithbauer

Thesis submitted to the
Benjamin M. Statler College of Engineering and Mineral Resources
at West Virginia University
in partial fulfillment of the requirements
for the degree of

Master of Science
in
Computer Science

Duncan R. Lorimer, Ph.D.
Maura A. McLaughlin, Ph.D.
Ramana A. Reddy, Ph.D.
Arun A. Ross, Ph.D., Chair

Lane Department of Computer Science and Electrical Engineering

Morgantown, West Virginia
2013

Keywords: pulsars, automation, machine learning, pattern recognition

Abstract

Analysis and Application of Automated Methods for Detecting Pulsars in the Green Bank Telescope 350MHz Drift-Scan Survey

by

David Paul Smithbauer
Master of Science in Computer Science

West Virginia University

Arun A. Ross, Ph.D., Chair

A significant portion of the process of detecting pulsars from radio sky surveys remains a largely manual task. The visual inspection of data in order to detect and validate potential pulsar candidates is by far the most time consuming portion of the overall process. Coupled with the fact that well over a Petabyte of pulsar survey data has been archived, the task of identifying these valuable phenomena is tedious and time consuming.

Using data from a survey performed with the National Radio Astronomy Observatory's (NRAO's) Green Bank Telescope (GBT) in 2007, this thesis explores the application of machine learning techniques to mitigate the manual efforts involved in pulsar candidate detection. The performance of three different classifiers is explored - Naive Bayes, C4.5 (J48) Decision Tree, and Support Vector Machine. Preprocessing and feature extraction methods are described and a framework for applying the classifiers to the survey data is presented. Multiple features were extracted from the survey data and used to train the classifiers. Cross-validation results of the various feature sets and classifiers are documented. Experiments suggest the potential of the proposed framework in rapidly detecting pulsars from large amounts of survey data.

Acknowledgements

My heartfelt thanks goes out to Dr. Arun A. Ross, my committee chair, for the numerous hours spent assisting, guiding, and mentoring me throughout this research and the pursuit of my degree. Even through his most recent transition to another university, he has readily made himself available to me and continued to be actively involved in my studies. His courses on pattern recognition and machine learning were some of the most intriguing and challenging I have ever had the privilege to experience.

This effort would not have been possible without the cooperation and partnership of the WVU Eberly College of Arts and Sciences Department of Physics and its faculty. Drs. Maura A. McLaughlin and Duncan R. Lorimer have provided their knowledge, time, and guidance to me at every opportunity. Lorimer's book was invaluable throughout this research and McLaughlin's patience and adeptness at educating me on the astrophysics of pulsars and the nuances of the GBT350 drift survey left me feeling like a student of both departments.

I am thankful to have had the opportunity to enroll in several courses instructed by Dr. Ramana A. Reddy. Early in my graduate career, I took Advanced Artificial Intelligence with him, and it was my first true introduction to many of the underlying concepts utilized in this research.

I cannot thank my family and friends enough for their support through all of this; they have encouraged and, at times, prodded me along. Jasmine, thank you for motivating me to complete my degree and for understanding the long hours and many late nights it required. Mom and Dad, you have always inspired me to be the best person I could be in all areas of life, and you have never wavered in your commitment to education.

Finally, none of this research would be possible without the data obtained by the National Radio Astronomy Observatory (NRAO). The NRAO is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

Contents

List of Figures

List of Tables

Notation

We use the following notation and symbols throughout this thesis:

$\mathbf{w}, \mathbf{D}$	:	Bold letters denote vectors
$p(\mathbf{w}\|\mathbf{D})$	:	Conditional probability function - probability of $\mathbf{w}$ given $\mathbf{D}$
$(\mathbf{x_i} \cdot \mathbf{x_j})$	:	Dot product
$\dot{P}$	:	First derivative - first derivative of P

Chapter 1

Introduction

In 2008 I was introduced to Dr. Maura McLaughlin and the process of identifying pulsars at a presentation provided at the West Virginia High Technology Consortium (WVHTC) Foundation in Fairmont, WV. I was fascinated by the vast amounts of data being recorded by these studies and the largely manual processes for identifying and classifying pulsars. McLaughlin presented the preliminary findings of the GBT350 drift survey, as it was currently in process at that time and would require years to scour through the immense number of candidate pointings recorded and post-processed from this survey.

After the presentation, I spoke with McLaughlin regarding the application of pattern recognition to the pulsar domain. She was excited to explore the possibility and we exchanged information. Shortly thereafter, Dr. Arun Ross and I were meeting with McLaughlin to outline a research plan between the WVU Lane Department of Computer Science and Electrical Engineering and the WVU Physics Department. In the early stages of this research both departments, along with the WVHTC Foundation, submitted a proposal to the National Science Foundation's (NSF) Cyber-Enabled Discovery and Innovation (CDI) solicitation.[1] Unfortunately, we were unsuccessful in obtaining the CDI funding, but the research continued over several years as part of my masters research.

This thesis, which is submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer Science, details the research performed in applying standard pattern recognition techniques to the GBT350 drift survey. Three different classifiers (J48

[1]NSF CDI - `http://www.nsf.gov/crssprgm/cdi/`

Decision Tree, Support Vector Machine - Binary Sequential Minimal Optimization, and Naive Bayes) were compared over various training samples of the survey and the training results were carefully recorded. Feature vectors were created from the pulsar candidate data, or candidate diagnostic plots, and each of the feature vectors were run through the same battery of 10-fold cross-validation training tests with each of the aforementioned classifiers. The results from the cross-validation tests were compared and the best performing classifier and feature vector were used to mine the entire GBT350 drift survey for pulsars.

1.1 Outline

This thesis is organized into chapters and provides both a brief introduction into the science and behavior of pulsars as well as a detailed account of the pattern recognition research performed on the GBT350 drift survey data and artifacts developed to facilitate this research. Chapter 1 discusses the background and importance of pulsar research and what has been discovered to date. It also presents introductory information regarding the approach to signal processing used when collecting pulsar survey data. It concludes with an overview of how the processed pulsar data are reviewed and candidate pulsar pointings are determined. Chapter 2 provides a small sampling of current research related to the application of machine learning and pattern recognition methods to astrophysics. Diagnostic plots are discussed in great detail in Chapter 3 - including a description of each graph, the data each represents, and an explanation of the statistical measurements and pointing information in the candidate diagnostic plot labels. The majority of the thesis is devoted to Chapter 4 and the methodology applied to this research. It covers the preprocessing and manipulation of the pulsar candidate data, feature generation and selection, and the classifiers used during the research. It describes the training and testing processes and provides all the compiled results of the cross-validation and testing, which show the performance statistics of each combination of classifier, feature, and data set. Finally, Chapter 5 presents the results and conclusions of this research. It discusses the limitations and the areas that could be further explored in subsequent work.

1.2 Description

Pulsars are born in supernova explosions that create highly magnetized, rapidly rotating neutron stars. The process of detecting these astronomical objects from radio sky surveys remains a largely manual task. The visual inspection of data in order to detect and validate potential pulsar candidates is by far the most time consuming portion of the overall process. Coupled with the fact that well over a Petabyte of pulsar survey data has been archived, the task of identifying these valuable phenomena is tedious and time consuming and becoming increasingly insurmountable as new surveys proliferate the amount of data gathered.[2] However, with modern computing power and machine learning techniques, it still remains possible to find these "needles in the haystack" and drastically reduce the manual constraints involved in pulsar candidate identification. It is the premise of this thesis that with minimal preprocessing and training, pulsar data sets can be mined for pulsars with a high degree of success, moving this process from the largely tedious and manual methods currently in use to the mostly automated and more rapid classification methods of the day.

In 2007, a drift-scan survey was performed using the National Radio Astronomy Observatory's (NRAO's)[3] Green Bank Telescope (GBT) at a radio frequency of 350 MHz that used 81.92 microsecond sampling and 2048 frequency channels.[4] The survey was performed while the GBT was undergoing track refurbishing and, therefore, the dish remained stationary to record the sky as it passed overhead. The area of the sky covered by the scan was 10,347 square degrees between declinations of -21 and 26 degrees and it included 459 known pulsars [2].

Each pointing recorded was a continuous block of data, approximately 140 seconds in duration and overlapped with the preceding pointing by 70 seconds, meaning each segment of data was processed in two different pointings [3]. Data are recorded at a rate of 25 MB/s, or 90 GB/hr, for 1,491 hours. Over 134 Terabytes of data were recorded in total from the GBT350 drift survey. After processing this data through multiple science software packages,

[2]Automated Detection Algorithms for Pulsars and Transients - ADAPT - NSF CDI Proposal 2008

[3]The National Radio Astronomy Observatory is a facility of the National Science Foundation operated under cooperative agreement by Associated Universities, Inc.

[4]GBT 350-MGz Driftscan Survey Processing - `http://www.as.wvu.edu/~pulsar/GBTdrift350/`

the eventual result was *2.5 million Candidate Diagnostic Plots (CDPs)*. Each CDP generated would require manual, visual inspection to determine whether the candidate was actually a pulsar. In two recent papers, published 6 years after the original survey was performed, the results of the GBT350 drift survey are presented [2, 3]. The survey uncovered 31 new pulsars, 10 of which are recycled pulsars - meaning their spin period has been increased by the angular momentum transferred via matter absorbed from a closely orbiting star.

The time required for the visual inspection of potential pulsar candidates is by far the most time consuming portion of the overall process. Eatough et al. support this premise by espousing the results of their study on the Parkes Multi-beam Survey which produced a vast number of CDPs requiring visual inspection - approximately 8 million. Eatough and colleagues postulated that the average time required to inspect a candidate pulsar, which can vary depending on credibility, is between 1 and 300 seconds. They go on to say that a database of 1 million candidate pulsars could take up to 10 years of continuous analysis to identify all potential pulsars [4]. This research is further examined in Chapter 2.

Throughout this thesis, the author will examine the application of machine learning and pattern recognition methods to the GBT350 drift survey in an attempt to partially mitigate the long period of time required by the manual review and inspection process of the CDPs.

1.3 Background

1.3.1 History of Pulsar Detection

Jocelyn Bell was a graduate student under the advisement of Anthony Hewish at Cambridge in 1967 when they discovered the first pulsar using a radio telescope and pen chart recorder [4]. During visual review of the power output from a radio telescope, they stumbled upon the regular telltale pattern of a pulsar. This was quite a find as only a small number of the known pulsars today are strong enough to be observed by their individual pulses alone [5]. Most require very sensitive telescopes and the application of advanced signal processing techniques in order to be detected.

The first pulsar detected was named “LGM-1” for “Little Green Men” as it was originally

thought to be evidence of extraterrestrial communication. The signal was extremely regular and produced sharp pulses every 1.3 seconds. No natural sources in the universe capable of producing such a signal were known at that time. After discovering multiple sources in the sky that produced pulsed signals, they published their findings without having determined the nature of the source. It was not until the end of 1968 that Thomas Gold demonstrated that pulsars were actually neutron stars, which had been predicted as early as 1933 but never observed until pulsars were discovered.[5] Currently there are over 1,800 known pulsars.[6]

1.3.2 Importance and Value of Pulsar Surveys

Pulsars are of great importance in studying the galaxy. Applications in solid-state physics, general relativity, galactic astronomy, astrometry, planetary physics, and cosmology have been made as a direct result of studying these objects. A subclass of pulsar, millisecond pulsars, have periods which can, at times, be measured to better than one part in 10^{15}. This clock-like precision and stability makes them invaluable for measuring other universal phenomena such as relativity and signal propagation effects in the galaxy due to ionized gas and magnetic fields. Millisecond pulsars can also be used to help directly detect gravitational waves [5].

1.4 Obtaining Pulsar Survey Data

1.4.1 Recording and Processing the Raw Data

The standard process for sifting through the recorded data in search of pulsars is a multi-step process that has been improved upon since the discovery of pulsars in 1967. To better explain the selection process, it is first necessary to discuss the data collection and recording processes.

To capture radiation from a pulsar, a large antenna is used to focus the signal to a feed horn that contains sensors for recording the signal and generating voltages. A Low-Noise

[5]APS Physics "This Month in Physics History" - `http://www.aps.org/publications/apsnews/200602/history.cfm`

[6]ATNF Pulsar Catalogue - `http://www.atnf.csiro.au/people/pulsar/psrcat/`

Amplifier (LNA) is used to strengthen the signal before sending it through to the rest of the data acquisition system. The signal then undergoes a myriad of mixing, filtering, and amplification which will not be discussed in detail here [1]. The GBT350 drift survey used the now-retired system called "Spigot" for data acquisition. Spigot was a custom auto-correlation spectrometer and digital signal processor. It has since been replaced with the Green Bank Ultimate Pulsar Processing Instrument (GUPPI) [2].

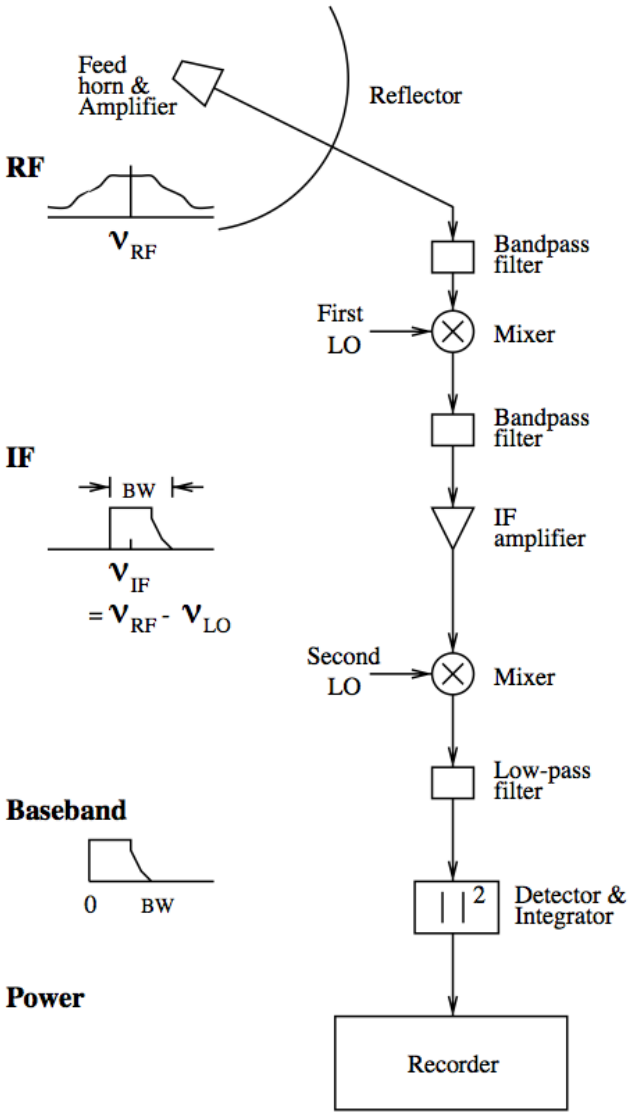

Figure 1.1: Schematic of Radio Telescope Receiver from [1]

Once the data have been converted to an electronic signal, it must undergo processing to correct for observational bias, noise, gaps in collection over time, and various other criteria.

The following sections will detail the processing that must occur prior to being able to utilize the recorded data in search of pulsars.

1.4.2 Effects of the Interstellar Medium and Dedispersion

A key step in this processing is to correct for how the signal of a pulsar is received at its destination. This process is known as dedispersion. As the signal from a pulsar travels from its origin toward Earth, it passes through areas of ionized gas, or plasma. When the signal passes through this interstellar medium, its signal characteristics are modified, causing the signal to differ from that of the same signal at its origin. The most noticeable effect of this signal modification results in lower-band radio frequencies arriving later than their higher-frequency counterparts, even though a pulsar emits its pulse across a wide radio frequency band simultaneously [1].

There are multiple methods available for correcting this phenomenon [6]. One technique that can be used is known as 'incoherent dedispersion'. This method requires splitting the full bandwidth of the signal into sub-bands and recording each of these sub-bands separately. Once the observation is complete, each of the subands can be shifted by an appropriate amount of time delay to allow the sub-bands to align properly. However, due to the discretized nature of this method, it does not correct for the dispersion delay inherent inside each of the sub-bands.

An alternate technique was developed in 1971 by Hankins. It is called 'coherent dedispersion' as this process accounts for the signal delay before it is passed to a detector. It also does not require splitting the band into smaller sub-bands, and as such, is not susceptible to the inherent limitations of the incoherent dedispersion method. Coherent dedispersion uses a combination of Fourier and Inverse Fourier transforms on the obtained time series to obtain the corrected voltage time series. The final product is now corrected of any in-band dispersion and can be passed to a detector for recording. It does, however, require very high sampling rates and is computationally more expensive [1, p. 120-122] [7].

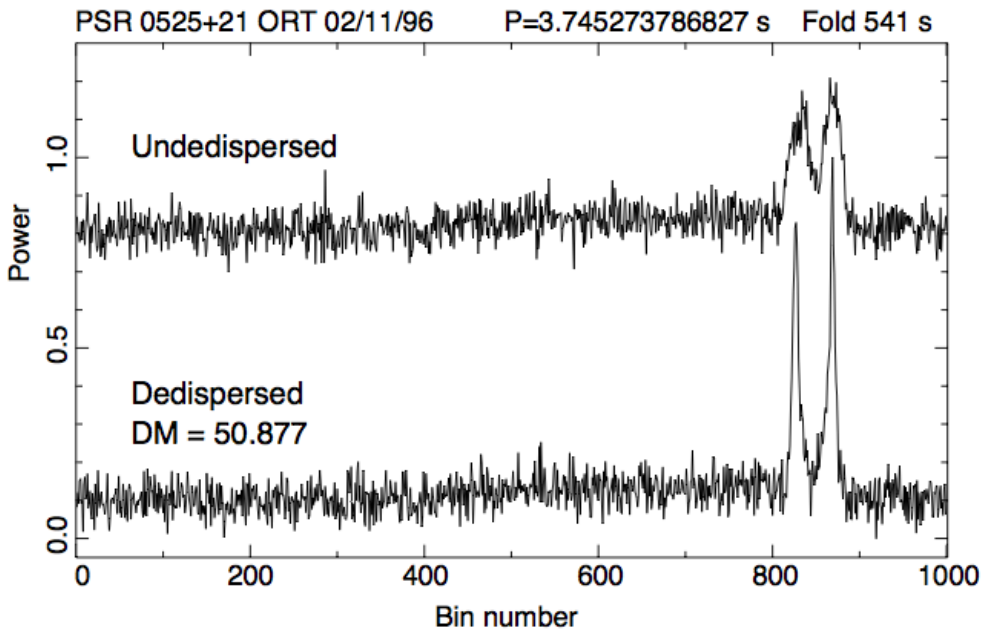

Figure 1.2: Example of Undedispersed vs. Dedispersed Signal from [1]

Simple dedispersion

The simplest form of dedispersion considers the raw data as a two-dimensional array of time samples and frequency channels where R_{jl} represents the j^{th} time sample and the l^{th} frequency channel.

$$T_j = \sum_{l=1}^{n_{chans}} R_{j+k(l)}, l, \tag{1.1}$$

where $k(l)$ is the nearest integer number of time samples corresponding to the dispersion delay of the l^{th} frequency channel relative to some reference frequency[5, p.127-130].

$$t_{DM} \simeq 4.15 \times 10^3 \text{ s} \times DM \times [(\frac{v_1}{\text{MHz}})^{-2} - (\frac{v_2}{\text{MHz}})^{-2}] \tag{1.2}$$

$k(l)$ can also be written as the magnitude of the delay between two frequencies, v_1 and v_2, using Equation 1.2, where DM is the dispersion measure in units of parsec/cm^3 [3].

Tree dedispersion

Due to the expensive nature of the simple dedispersion technique which performs in $O(n^2)$ where n is the number of channels, another algorithm was developed. The Tree dedispersion algorithm operates on the premise that each tree can be built from smaller pieces (or trees)

which begin with simple two-channel 'branches'. This algorithm performs in $O(\log_2 n)$, where n is the number of channels. The Tree dedispersion method does assume the dispersion delay across the frequency band is linear. This is an oversimplification, but in many cases this is adequate [5, pp.130-131].

1.4.3 Barycentric Correction

Once the data are dedispersed to a number of time series over trial DM values, it may be necessary to correct for the change in frame of reference from the observatory and the target if the data were recorded over a long period of time (> 30 minutes). This correction process is called a Barycentric Correction. For each pointing in the GBT350 drift survey, a transformation to the solar system barycenter was made. This was done using the DE200 ephemeris which is in the J2000 system used by the PRESTO software [3].

1.4.4 Detecting Periodic Signals

Next, a type of Fourier transform is applied to these time series to illuminate any significant features. The k^{th} Fourier component of the Discrete Fourier Transform can be computed by the equation:

$$F_k = \sum_{j=0}^{N-1} T_j \times e^{(-2\pi ijk/N)}, \tag{1.3}$$

where $i = \sqrt{-1}$ and N is the number of elements in time series T_j [5, p.133].

1.4.5 Correcting for Noise

Many other steps can be applied to the data to further remove noise and potentially improve the resultant analysis. Some of these include attempting to increase the sensitivity to narrow pulses, reducing possible false positives, and correcting for discontinuities in the time series of the recorded data.

There are two different kinds of noise inherent to radio telescopes that must be addressed during the search for pulsars. The first kind of noise is known as 'receiver noise', T_{rec}. This is the noise inherent to the electronic components used in the telescope itself. The second is

'sky noise', T_{sky}, which is radio noise observed in the beam. This kind of noise is generated by emission sources within the galaxy itself. The total temperature of the sample can be computed as $T_{rec} + T_{sky}$ [5, p.263].

A common technique for ensuring the response to noise is as uniform as possible is to 'whiten the spectrum'. This involves breaking the spectrum up into a number of contiguous pieces. For each piece, the mean and root mean square value can be calculated. Normalizing the local root mean square and subtracting a running median results in the whitened spectrum having a S/N ratio of any spectral feature that is simply its amplitude [5, pp.136-137].

1.5 Selecting Pulsar Candidates

Once the signal has been dedispersed and Fourier transform applied as described above, all that remains is to examine the candidate periods and signal-to-noise (S/N) ratios for each of the computed DMs. It is not uncommon for the same pulsar to appear many times at different S/N ratios [5, p.142]. Each of the computed candidates is fed into the PRESTO Prepfold program, discussed in greater detail in Chapters 3 and 4, and CDPs are generated for each.

It is here where the long, manual process of reviewing the CDPs begins. Each CDP is visually inspected on three main criteria:

1. a distinct peak in the shape of the signal at DMs > 0 parsec/cm^3,
2. broadband emission, and
3. reasonably consistent emission over time.

Any promising candidates are scheduled for follow-up observations which include regular timing observations [3].

Chapter 2

Current Methods

2.1 Analogous Research and Publications

This Chapter details some of the methods and techniques used by other researchers that are similar in nature to those used in this research. It is not meant to be exhaustive or representative of all related research; instead, it is intended as a brief look at other parallel research.

2.1.1 Support Vector Machines and Kd-tree for Identifying Quasars

Support vector machines and kd-tree for separating quasars from large survey data bases by Dan Gao et al. (2007) describes a study that showed Support Vector Machines (SVMs) and k-dimensional tree (kd-tree) algorithms are "effective automated algorithms to classify point sources" [8]. The study focused on distinguishing quasars from normal stars in the Sloan Digital Sky Survey (SDSS) and the Two-Micron All Sky Survey (2MASS).

Gao et al. acknowledge that astronomy, like many other fields, has entered the 'data avalanche era' and that data mining tools must be relied upon for discovering, classifying, clustering, and even defining object types within large astrophysics data sets. Gao and colleagues provided numerous other sources emphasizing just how much research has been occurring in this area for sightly more than a decade. For brevity, the list of authors and a summary of their research as outlined by Gao et al. is listed in Table 2.1.

The SDSS and GBT350 drift survey were both drift-scans, meaning the telescope re-

Table 2.1: Related Research Applications of Machine Learning Algorithms to Astrophysics from Gao et al. (2007)

Authors	**Summary**	**Publication Year**
Hatzim-inaoglou, Mathez and Pelló	Automated distinguishment between quasars and starts/galaxies by photometry	2000
Wolf et al.	Photometric method for identifying stars, galaxies and quasars in multicolor surveys	2001
McGlynn et al.	Decision trees for automated classification of X-ray sources	2004
Carballo, Cofiño and González-Serrano et al.	Neural networks to select quasar candidates from combined radio and optical surveys	2004
Suchkov, Hanisch and Margon	Oblique decision tree classifier optimized for astronomical classification and redshift esitmation	2005
Ball et al.	Classifed stars and galaxies using decision trees	2006
Woźniak et al.; Woźniak, Williams and Gupta	SVMs successfully applied for classification of variable stars	2001, 2004
Humphreys et al.	Galaxy morphology classification	2001
Qu et al.	Solar-flare detection	2003
Zhang and Zhao	Classification of multiwavelength data	2003, 2004
Wadadekar; Wang et al.	Estimation of photometric redshifts of galaxies	2005, 2007
Rohde et al.	Classification of different object catalogues in astrophysics	2005, 2006
Wang et al.	SVMs and kernel regression for photometric redshift estimation	2007
Hsieh, Yee, and Lin	Kd-tree algorithm to improve redshift accuracy of galaxies	2005
Kubica et al.	Kd-tree for intra- and inter-night linking of asteroid detections	2007

mained stationary allowing the sky to pass overhead. SDSS used a dedicated, wide field, 2.5 meter telescope at Apache Point Observatory, NM. Imaging was recorded using a 142 megapixel camera in five broad bands. The 2MASS survey was performed using highly automated 1.3 meter telescopes. The first is located at Mt. Hopkins, AZ, and the second at the Cerro Tololo Inter-American Observatory (CTIO) in Chile. Both telescopes recorded data using a three-channel camera, where each channel was comprised of a 256 x 256 array of HgCdTe (Mercury Cadmium Telluride) detectors.

Photometric data from both surveys were used to train and perform quasar classification as both data sets contained matching records for nearly all objects observed during the surveys. Principal component analysis (PCA) was used to determine the minimum number of uncorrelated variables and showed vastly diminishing return after three independent variables. Both the kd-tree and SVM classifiers were trained and tested using a 10-fold cross-validation scheme. Performance metrics were captured from the cross-fold validation. These metrics included overall classification accuracy, true negative rate, true positive rate, weighted accuracy, G-mean, precision, recall, and F-measure. All of the aforementioned metrics are functions of the confusion matrix generated for each cross-validation run.

Gao et al. assert that the accuracy of the two classifiers, given the best photometric feature vectors, was greater than 97% and very capable of distinguishing quasars from stars. They also report that the kd-tree algorithm was much faster than the SVM algorithm, but SVMs showed slightly better performance in G-mean, F-mean, and weighted accuracy. They note, however, that the tradeoff for accuracy versus the speed of the kd-tree algorithm was not significant enough to warrant the sole use of SVMs. In their conclusion, they maintain that based on the metrics they gathered it was inconclusive which classifier was superior. In addition, they found that accuracy did not always increase as more features were added - the “Curse of Dimensionality”.

2.1.2 Neural Networks for Identification of Quasars

Another recent study by Carballo et al. used neural networks for identifying quasi-stellar objects (QSOs) from the Faint Images of the Radio Sky at Twenty cm survey (FIRST) and

SDSS Data Release 5 (DR5) photometric survey [9]. In particular, they were attempting to classify QSOs with a redshift $z \geq 3.6$ for the purpose of studying the theory of accretion of matter on to supermassive black holes that are in the center of galaxies.

They approached the problem as a two-class, supervised learning problem where they trained and employed a Neural Network (NN) using labeled data from the FIRST-DR5 survey to train and eventually identify QSOs in the matching unlabeled data from the SDSS-DR5 survey. Between the two surveys there were 8,665 photometric matches - 4,250 sources with DR5 spectra (labeled) and 4,415 with DR5 spectra (unlabeled). The NN was applied to the 4,415 sources without DR5 spectra which yielded 58 high-z QSO candidates. These results were then cross referenced against the NASA/IPAC Extragalactic Database (NED), SDSS Data Release 6 (DR6), and follow-up spectroscopy with the William Herschel Telescope.

The 4,248 objects with spectra were used to train a feed-forward NN with a single layer for the input data and an output layer yielding 1 for high-z QSOs or 0 otherwise. The output function is defined in Equation 2.1, where y^i is the discrete value in the range (0, 1) for the ith output as computed by the sigmoid function $f(a^i)$ and a^i is a linear function of the inputs $(x_1, x_2, \ldots, x_d)$. w_0 is the bias and $(w_1, w_2, \ldots, w_d)$ are the weights applied to the various connections in NN, which occur during training.

$$y^i = f(a^i) = \frac{1}{1 + e^{-a^i}}, \tag{2.1}$$

with

$$a^i = w_0 + \sum_{j=1}^{d} w_j x_j^i \tag{2.2}$$

Carballo et al. used the NN model known as "logistic linear discriminant" and the following error function, which is the mean of the squared errors of the outputs.

$$\frac{1}{m} \sum_{i=1}^{m} (y^i - t^i)^2 \tag{2.3}$$

While not much is divulged in their research as to how the NN is structured (i.e., how many levels, number of nodes in each level, etc.) they go into thorough detail as to the results of the application of the trained NN on the remaining data. According to Carballo and team, their approach was able to separate high-z QSOs from the remaining classes with

96% completeness and 62% efficiency. They defined efficiency as the fraction of sources that actually are high-z QSO candidates with $y \geq y_c$, y_c as the threshold level for classifying the candidate as a high-z QSO candidate, and completeness as the inverse fraction.

Similar to our application of machine learning to pulsar data in the GBT350 drift survey, Carballo et al. also suffered from a data set that was vastly disproportionate in the number of true high-z candidates versus non high-z candidates - 52 to 4415 respectively. This lead them to apply the "leave-one-out" cross-validation technique for partitioning the training and test samples such that the classifier could be empirically evaluated such that the candidates used for learning were not re-used for testing. This approach was also used in the training methodology applied to our pulsar classifiers.

Ultimately, Carballo and colleagues concluded from their research that an estimated 11 FIRST high-z QSOs were missed by SDSS (7 QSOs and 4 candidates).

2.1.3 Neural Networks for Identification of Pulsars

Eatough et al. published their findings in 2010 regarding the application of Artificial Neural Networks (ANNs) for discovering pulsars in the Parkes Multi-beam Pulsar Survey (PMPS) [4]. Similarly to that of the GBT350 drift survey, the PMPS produced a vast number of candidate diagnostic plots requiring visual inspection - approximately numbering 8 million. Eatough and colleagues postulate that the average time required to inspect a candidate pulsar, which can vary depending on credibility, is between 1 and 300 seconds. They go on to say that a single person evaluating a database of 1 million candidate pulsars would need anywhere from 12 days to 10 years of continuous analysis to identify all potential pulsars. They also cite fatigue over this tedious, manual process as a risk that could increase the potential for human error.

In lieu of human review, Eatough et al. explored the possibility of using ANNs and were successful in identifying a new pulsar in the PMPS data set that was previously unknown. They state that ANNs have been used in other astronomy applications for some time, specifically the morphological classification of galaxies.

Their paper addresses the difficulty that arises when applying machine learning to dis-

Table 2.2: Top Pulsar Features Used by Eatough et al. in their Creation and Tuning of their ANNs

1	Pulse profile Signal to Noise Ratio (SNR)
2	Pulse profile width
3	χ^2 of fit to theoretical DM-SNR curve
4	No. of DM trials with SNR >10
5	χ^2 of fit to optimized theoretical DM-SNR curve
6	χ^2 of fit to theoretical acceleration-SNR curve.
7	No. of acceleration trials with SNR >10
8	χ^2 of fit to optimized theoretical acceleration-SNR curve
9	Root mean square (RMS) scatter in subband maxima
10	Linear correlation across subbands
11	RMS scatter in subintegration maxima
12	Linear correlation across subintegrations

tinguishing RFI from viable pulsar candidates. In order to create the ANNs and to properly train them, they enumerated a list of common features of genuine pulsar candidates, selected various subsets of these features, trained the ANNs, and examined their performance. The subset of features they settled upon for training the ANNs is listed in Table 2.2.

They also experimented with two different ANN architectures. The first was an 8:8:2 - meaning eight input nodes, eight nodes in layer two, and two output nodes. The input node criteria corresponded to features 1 - 8 in Table 2.2. Likewise, a 12:12:2 was later created to add features 9 - 12. Eatough et al. note that while the second architecture provides more adjustable weight parameters and can, in theory, represent more complex scenarios, it is also computationally more complex. The larger ANN takes more time to both train and classify the data.

The ANNs were trained using 259 pulsars, with varying characteristics, and 1,625 non-pulsars. The 8:8:2 ANN positively classified approximately 13,000 candidates which included 92% of the true pulsars included within the input data. The 12:12:2 ANN improved that result slightly by recovering 93%. The use of the 8:8:2 ANN did lead to the discovery of a new pulsar - PSR J1926+0739. In their summary, however, they caution that pulsar candidates could have been missed due to "poor training of the ANNs..., abnormal candidate plots generated by [their] search software, or unbalanced training sets". Furthermore they caution that ANNs are not yet a true replacement for human, visual inspection [4].

Chapter 3

Candidate Diagnostic Plots

The output of interest produced by the science software packages mentioned in Chapter 1 is the Candidate Diagnostic Plot, or CDP (see Figure 3.1). For the GBT350 drift survey, these plots were generated using the Prepfold program, a part of the PRESTO suite of pulsar search and analysis software.[1] PRESTO and its use in this research will be explained in greater detail in Chapter 4.

Each potential pointing has a corresponding CDP which is output in Postscript format and contains detailed information about the pointing and the radio signal at that location. For the GBT350 drift survey, over 2.5 million CDPs were generated. Each of these CDPs must, at present, be manually reviewed by visual inspection to determine whether the candidate is or is not a pulsar. This is a very tedious process that relies on domain knowledge that must be communicated to each new reviewer and that can be prone to error or misinterpretation.

3.1 Overview

There are several sections into which the data within the CDP are divided. The main two categories are the 'Search Information' section at the top-right which consists of textual data and the graphs which present various information about the structure of the pulse.

The graph data are broken down into four subcategories. These subcategories and their

[1]PRESTO - `http://www.cv.nrao.edu/~sransom/presto/`

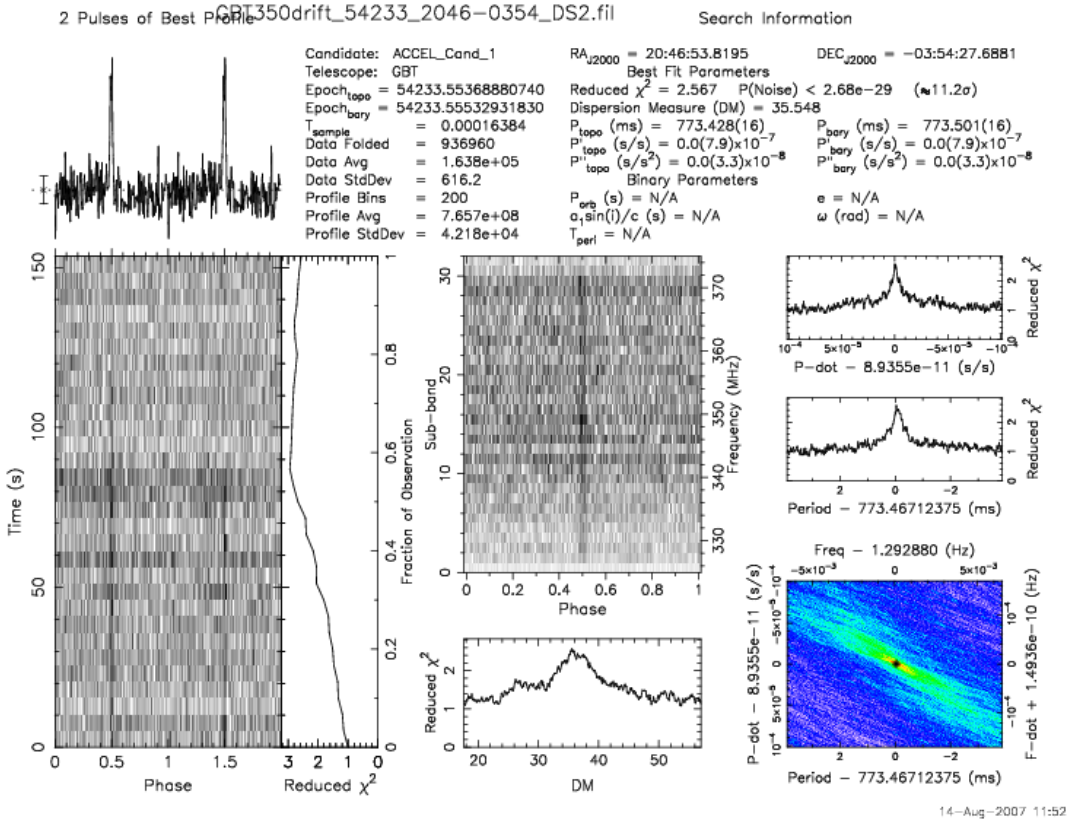

Figure 3.1: Example Prepfold Candidate Diagnostic Plot of a Pulsar

graphs will be explained in more detail in the following subsections.[2] The subcategories are the 1) Subintegration and Pulse Profile , 2) Frequency and Sub-bands, 3) Dispersion Measure, and the 4) Period and Period Derivative ($\dot{P}$) categories. Each subcategory consists of one or more graphs and each serves to provide an 'at-a-glance' meaning to the reviewer of the CDP in order for him or her determine its viability as a potential pulse profile of an actual pulsar.

3.1.1 Subintegration and Pulse Profile

The most important section of a CDP is the Subintegrations and Pulse Profile. This is a pair of graphs that provides a visual representation of the signal strength over time and phase. The Subintegrations plot (see Figure 3.2) is comprised of 'bins' which represent the strength of the signal at a point in time of a portion of the data within the measured pulse. Each bin is shaded with a grayscale value where white indicates no signal is present and darker

[2]PRESTO and Finding Pulsars - `http://www.astro.virginia.edu/~rsl4v/PSC/time.html`

bins indicate stronger signal. The X-axis is the phase of the pulsar, which encompasses two full rotations of the candidate pulsar. The Y-axis is time and measures the seconds from the start of the observation. Dark vertical lines in the time series that occur from the start of the observation through the end tend to represent pulsars as the signal remained strong at that phase throughout the observation.

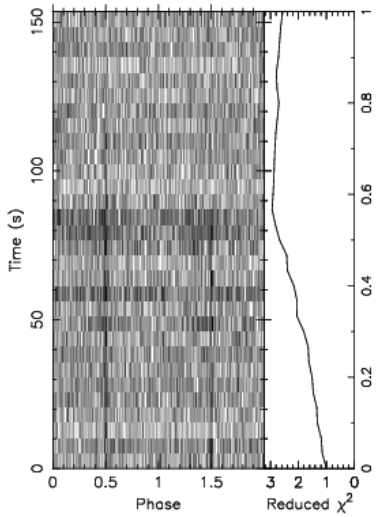

Figure 3.2: Example Prepfold Subintegrations Graph

Figure 3.3: Example Prepfold Pulse Profile Graph

The smaller graph above the Subintegrations plot is the Pulsar Profile (see Figure 3.3). This graph represents the strength of the pulse as a function of phase. In other words, it is the integration of the results of the sub-folds, or rows, within the Subintegrations plot. The spikes in the graph represent the portion of the period within the observed signal that could represent the pulsar beam pointing toward the telescope while the rest of the graph displays background noise.

3.1.2 Frequency and Sub-bands

The next category is the Frequency and Sub-bands plot (see Figure 3.4). Pulsars typically exhibit RFI across a broad spectrum, so this plot helps display the strength of the signal across the frequency. This graph is structured very similarly to that of the Subintegrations graph in that the data are still discretized and darker bins represent stronger signal strength. However, this graph's Y-axis has been replaced with Frequency and shows the observation frequency range over which the signal has been observed. The signal is also broken down into sub-bands as labeled along the left vertical axis. These observations typically utilize 32 or 64 sub-bands. The shaded bins represent the power collected in a single sub-band over the duration of the entire observation.

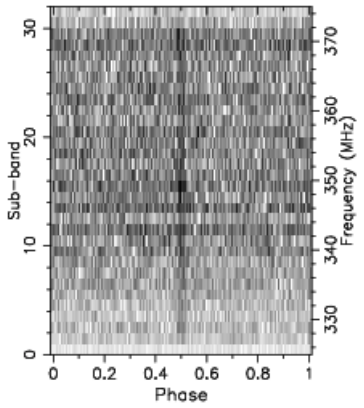

Figure 3.4: Example Prepfold Frequency vs. Sub-band Graph

One should be able to see an increase in signal strength at the same phase location as that displayed in the Time Domain plot. Again, this is represented by dark vertical lines in the graph.

3.1.3 Dispersion Measure

The Dispersion Measure (DM) portion of the CDP consists of one graph with an X-axis labeled DM and Y-axis of Reduced χ^2 as seen in Figure 3.5. DM is the integrated electron density along the line of sight between the pulsar and the Earth. As space is not empty, electrons encountered along the pulsar signal's path disperse the signal as it travels toward Earth. This is the reason for frequencies in the lower end of the spectrum arriving later than

those in the upper range.

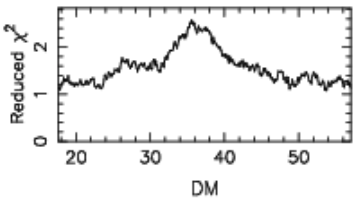

Figure 3.5: Example Prepfold DM vs. Reduced χ^2 Graph

The Reduced χ^2 value is a statistical measure of the fitness of of a model to a set of observations. When applying the Reduced χ^2 model to the pulsar data, a high value of Reduced χ^2 provides the reviewer with greater confidence that the signal has significance. When combined with DM, the distance of the pulsar to Earth can be estimated, helping to eliminate candidates that most likely occurred due to RFI generated by electrical systems here on the Earth.

3.1.4 Period and Period Derivative

The last subcategory of the CDP is that of the Period and Period Derivative ($\dot{P}$) graphs shown in Figure 3.6. These three graphs all provide information relative to the rotational period of the pulsar. The Period plot provides a measure of how well the period was measured while the $\dot{P}$ graph provides a measure of acceleration or deceleration of the pulsar period. The $\dot{P}$ graph should peak near 0 as pulsar's rotation is generally stable but is slowing down ever so slightly. It is possible, however, to see a $\dot{P}$ reading where the pulsar's period is increasing or decreasing in binary pulsar systems or pulsars with orbiting planets. True pulsars should still exhibit a sharp peak on the Reduced χ^2 vs. $\dot{P}$ plot.

The third graph is a visual combination of the previous two where color is used to represent the Reduced χ^2 value, with the red end of the spectrum representing higher values of Reduced χ^2. As seen in Figure 3.7, if the candidate is truly a potential pulsar, then only a single, well defined region of red should exist indicating good measurement of period and $\dot{P}$ instead of random RFI.

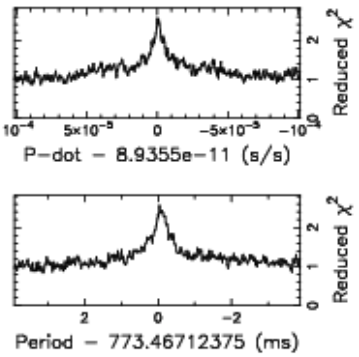

Figure 3.6: Example Prepfold Period and Period Derivative Graphs

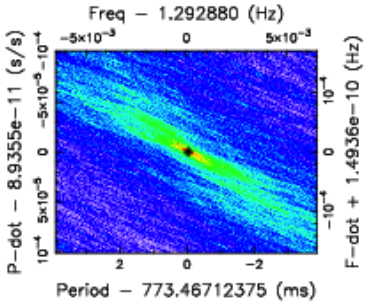

Figure 3.7: Example Prepfold Period Derivative and Reduced χ^2 Graph

Chapter 4

Methodology

4.1 Preprocessing the Data

Before it was possible to perform any type of data mining on the pulsar data, it was first necessary to understand the storage structure and layout of the GBT350 drift survey data. All data used in this research resided on the WVU Physics Department's Beowulf cluster.

The results of the PRESTO software package were all stored in a directory which contained 32 subdirectories at the time of this research. These subdirectories correspond to the whole number portion of the Epoch, or Modified Julian Date, value of the observation (e.g., 54233). Progressing further down into the directory structure, the next level is organized by the combination of right ascension and declination using the J2000 coordinate system.[1] Within each of the folders at this level, the acceleration and single pulse candidate information is stored. This is the PRESTO output for the candidates of best profile and includes the Postscript output used to generate the CDPs.

4.1.1 Extracting the Prepfold Files

While all of the pulsar data from the GBT350 drift survey had already been run through the PRESTO[2] program to determine optimal candidates and create the CDPs from those selected, it was necessary to rerun all of the data through a modified version of a subcom-

[1]PRESTO and Finding Pulsars - `http://www.astro.virginia.edu/~rsl4v/PSC/presto_glossary.html`

[2]PRESTO - `http://www.cv.nrao.edu/~sransom/presto/`

ponent of PRESTO. One of the C programs that is included with the PRESTO suite is 'export_pfd'. This application exports the Prepfold output to a CDP. However, the output of this program is, by default, a vectorized Postscript file that renders the CDP, including all of its textual information, as an image. This presented a challenge, as it would be necessary to first extract the actual data points. Because the data were vectorized, neither the text nor graphs contained on the CDPs were of any use in their current format. To solve this problem the 'export_pfd' program was modified to output the two-dimensional data for all graphs and the textual 'Search Information' to a text file as well as the original Postscript file.

This new version of 'export_pfd' was installed on the WVU Astrophysics Department's Beowulf cluster and a Python script 'exportPfdData.py' was written to recursively extract the Prepfold files and run the newly modified 'export_pfd' C program. For each Prepfold file throughout the GBT350 data new 'ascii_graph_output.txt', 'ascii_image_output.txt', and 'ascii_text_output.txt' files were generated.

The 'ascii_graph_output.txt' file contained the X and Y coordinate data for the 'Time vs. Reduced χ^2', 'Combined Best Profile', 'DM vs. Reduced χ^2', 'Period vs. Reduced χ^2', and 'P-dot vs. Reduced χ^2' graphs. The 'ascii_text_output.txt contained all of textual information under 'Search Information' heading, and the 'ascii_image_output.txt' contained the complex image data also available from the Prepfold output.

Once the modified version of 'export_pfd' was run and the textual data extracted from the Prepfold files, a 'splitGraphFiles.py' was created and executed to split the graph data into individual files for easier ingestion into Octave.[3] This preparatory step created separate graph files consisting of headers and X and Y coordinate data for each of the aforementioned graphs. The output files were named 'timeRedChi.txt', 'combBestProfile.txt', 'dmRedChi.txt', 'periodRedChi.txt', and 'pdotRedChi.txt' respectively. The original, combined graph file - 'ascii_graph_output.txt' was then removed to save space and prevent data duplication.

[3]Octave - `http://www.gnu.org/software/octave`

4.2 Feature Selection and Generation

All features were generated using the GNU Octave product. Octave is an open source software program that performs numerical computations and has its own high-level interpreted language. Octave's syntax is largely compatible with the commercial software product - MATLAB.[4] Two Octave files were created for the purpose of feature generation. The first 'generateFeatures_functionLibrary.m' was, as its name implies, a function library where various feature generation functions could be created and stored to be executed on the extracted Prepfold data. The functions in this library were used by the other Octave file 'generateFeatures.m' that recursively executed each of the feature functions against the extracted Prepfold candidates.

The feature functions created and executed on the pulsar data were:

1. redChiYValueFeatures

   This function calculated the vertical minimum, median, maximum, and mean of the 'dmRedChi.txt' graph Y column data as a four-feature input vector.
   {MinY, MedianY, MaxY, MeanY}

2. redChiYValue2Features

   This function calculated the vertical (maximum − minimum), mean, and standard deviation of the 'dmRedChi.txt' graph Y column data as a three-feature input vector.
   {MaxY − MinY, MeanY, StdDevY}

3. redChiXValueCurveFeatures

   This function calculated the X value at the peak and computed the ratio of the X value of the second order polynomial fit to the maximum Y value.
   {XValAtPeak, fitCurveMaxToDMChiMaxRatio}

   This was achieved by by calculating the approximate second order polynomial that fit the DM Reduced χ^2 function using the 'polyfit/polyval' functions of Octave and computing the ratio of the maximum height of the interpolated polynomial to the

[4]MATLAB - `http://www.mathworks.com/products/matlab/`

maximum Y value of the graph (see Figure 4.1). The X value at the peak and the computed ratio was stored as a two-feature input vector.

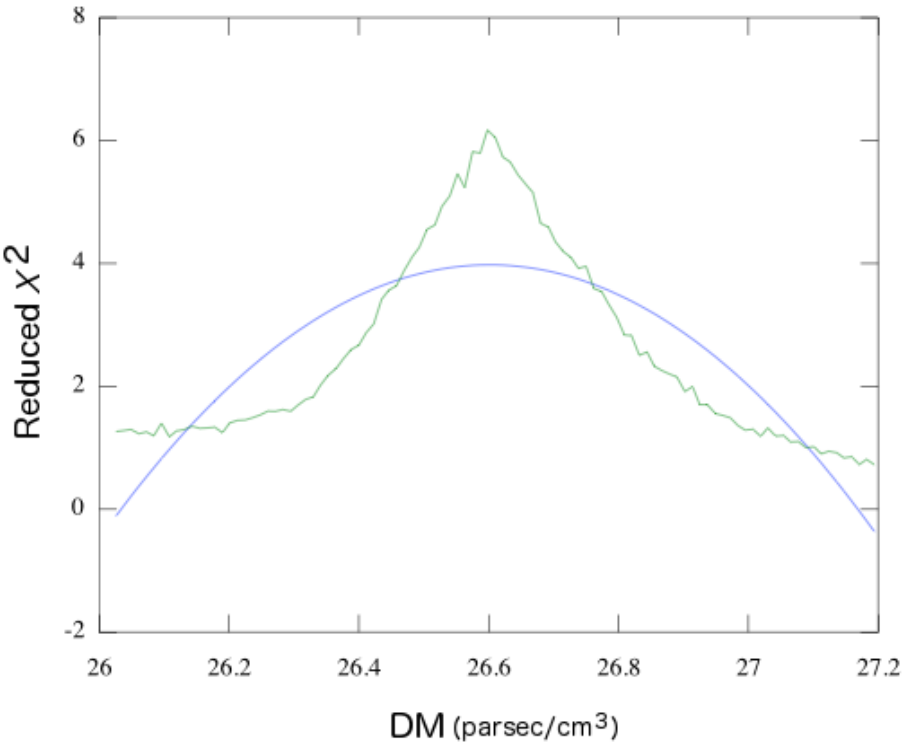

Figure 4.1: Interpolated Polynomial Fit Function Over DM vs. Reduced χ^2 Graph

At this time, it is important to note that feature selection was not examined to better determine the features that most likely model the two-class pulsar problem. Both Gao et al. (2008) and Eatough et al. (2010) discussed in Chapter 2 performed feature analysis to determine a suboptimal feature set for training and classification. Gao et al. even went as far as using PCA to determine the minimum number of uncorrelated features available for selection. No sophisticated methods, such as PCA or automated feature selection, were used during this research. A simplistic approach was employed that could, and most likely should, be expanded in future research to assess the efficiency and performance of alternate features.

All feature vectors described above and used throughout this research were generated on the 'DM vs. Reduced χ^2' graph, as outlined in Chapter 3. While the 'DM vs. Reduced χ^2' graph is a key component in determining whether a CDP represents a pulsar or not, it is not

the only factor. The 'DM vs. Reduced χ^2' graph was selected because of its simplicity as well as its importance in isolating whether the candidate signal was a possible astrophysical source (it exists within a reasonable distance to the Earth) or whether it should be eliminated from consideration entirely because its pattern suggests that its origin is most likely terrestrial RFI.

Features were then generated for all Presto candidate pointings and consolidated using the 'consolidateFeatures.py' script. This script recursively traversed the candidate pointings and created a master file for the provided input feature vector, supplied as an input parameter to the script. The consolidated files had the header information and the input feature vectors of a single kind for all Prepfold candidate pointings. This file was then downloaded and used with the local Java classification code.

4.3 Classifier Selection

This section will discuss the classification algorithms selected to discover patterns within the pulsar data, the selection process, training of the classifiers, test criteria utilized, and the application developed to exercise the classifiers on the extracted features.

Three different classifiers were applied to the feature sets extracted from GBT350 drift survey. They were the Naive Bayes, C4.5 (J48) Decision Tree, and Support Vector Machine - Binary Sequential Minimal Optimization (SMO) classification algorithms. Selection of these algorithms was determined purely from academic experience with the aforementioned classifiers.

4.3.1 Naive Bayes

According to Mitchell, "Bayesian learning algorithms that calculate explicit probabilities for hypotheses, such as the naive Bayes classifier, are among the most practical approaches to certain types of learning problems [10, p.154]." He goes on to cite a study performed by Michie et al. (1994) in which the researchers state that the Naive Bayes classifier is on par with other popular classification algorithms like neural networks and decision tree algorithms and, in some cases, can outperform them [11]. Unlike the Bayes classifier (see Equation 4.1),

where the posterior probability $p(\mathbf{w}|\mathbf{D})$ must be calculated for each parameter of $\mathbf{w}$.

$$p(\mathbf{w}|\mathbf{D}) = \frac{\mathrm{p}(\mathbf{D}|\mathbf{w})\mathrm{p}(\mathbf{w})}{\mathrm{p}(\mathbf{D})} \tag{4.1}$$

The Naive Bayes classifier, however, works by estimating the prior and likelihood probabilities based upon their frequencies over the training data and this value is then used to classify the target instance. These estimated probabilities become the learned hypothesis which is used to classify new instances (see Equation 4.2). The Naive Bayes classifier makes the assumption that the probability of observing the conjunction of all data $\mathbf{D}$ is simply the product of the probabilities of each parameter in $\mathbf{w}$. Let $p(\mathbf{w})$ be the prior probability of $\mathbf{w}$ and the observed data be $\mathbf{D} = \{\mathrm{t}_1, \ldots, \mathrm{t}_\mathrm{N}\}$. This yields the posterior probability $p(\mathbf{w}|\mathbf{D})$ or the uncertainty in $\mathbf{w}$ after the observed data set $\mathbf{D}$. $p(\mathbf{D}|\mathbf{w})$ is called the 'likelihood' function which defines how probable the observed data set $\mathbf{D}$ is for various values of the parameter vector $\mathbf{w}$ [12, p.22].

$$p(\mathbf{w}|\mathbf{D}_\mathrm{i}) = \mathrm{p}(\mathbf{w}) \prod_\mathrm{i} \mathrm{p}(\mathbf{D}_\mathrm{i}|\mathbf{w}) \tag{4.2}$$

4.3.2 C4.5 Decision Tree

This statistical classifier was developed by Ross Quinlan as an extension to the ID3 algorithm [13]. It is part of a larger set of classifiers known as tree-based classifiers. This genre of classifiers deals with dividing the solution space into finite sections that can be easily quantified by a simple model for any given point in space. If one views the entire solution space as a combination of these models, all that is necessary to determine the appropriate model for any given point χ in space is to traverse a binary tree where each node evaluates the input according to its decision criteria and further reduces the available solution space until a terminal node, or classification, is reached [12, p. 663-664].

The most interesting part of decision trees is how to construct them given a training set of data. The structure of the tree must be determined by the training set - meaning an input variable and a threshold must be chosen for each node that will ultimately lead to the best classification of the input data. Most decision tree algorithms employ some variation of a

top-down greedy search through the solution space of possible decision trees [10, p.55]. They begin by attempting to discover the best candidate root node by evaluating each attribute using a statistical test to measure performance.

In classification problems the most common statistical methods for growing a decision tree are based on performance. Oftentimes this is measured as a factor of entropy, or a measure of the expected value of the information contained in the distribution. Two commonly used entropy measurements are cross-entropy and Gini index [12, p.666]. In both methods let $p_{\tau k}$ be the proportion of data points in region R_τ of the solution space assigned to class k, where k = 1, ..., K.

Cross Entropy

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k} \ln p_{\tau k} \tag{4.3}$$

Gini Index

$$Q_\tau(T) = \sum_{k=1}^{K} p_{\tau k}(1 - p_{\tau k}) \tag{4.4}$$

4.3.3 Support Vector Machine - Binary Sequential Minimal Optimization

While single-layer NNs can be easily and efficiently trained, they are limited to linear decision boundaries of the input space. Increasing the number of layers can allow for the ability to handle non-linear boundaries, but has the negative effect of increasing training complexity due to local minima and high dimensionality of the weight space. Support Vector Machines (SVMs) are a different class of learning algorithm, generically known as kernel machines, that can retain their training efficiency even given complex, nonlinear configurations [14, p.749].

SVMs are decision machines and do not generate posterior probabilities like the aforementioned Bayesian statistical classifier. SVMs also take a differing approach to classification from that of NNs. Whereas NNs use a fixed number of vertexes, or perceptrons, to determine the solution, SVMs can have varying numbers of basis functions to determine the appropriate classification of an input vector. While each basis function of the SVM has a single function to determine its output, in contrast the NN, while having a fixed number, can adapt the

functions during training. An important aspect of SVMs is that "the determination of the model parameters corresponds to a convex optimization problem, and so any local solution is also a global optimum [12, p.325]."

To find the decision boundary to separate two classes with an SVM is essentially a quadratic programming optimization problem. By effectively projecting a lower-dimensional problem into a higher-dimensional space, the input space can still be divided using a linear decision boundary in the higher dimensional space. The danger is in overfitting the data if the dimensionality approaches the number of inputs. To mitigate overfitting, SVMs work by attempting to find the optimal linear boundary with the largest margin between positive samples on one side and negative samples on the other [14, p.749].

Sequential Minimal Optimization (SMO) is a newer SVM algorithm that is gaining popularity. It works by breaking apart the quadratic programming problem into subproblems and solving the smallest possible optimization problem at each step. The SMO algorithm accomplishes this by solving for two Lagrange multipliers at each step. Because of this approach, the memory required when using SMO is linear with respect to the training set size. Large matrix computation is therefore unnecessary, allowing SMO to perform between linear and quadratic instead of cubic like many other SVM algorithms [15].

Let $\vec{x_i}$ be a set of training inputs with classification $y_i = \pm 1$. To find the optimal separator in the input space then one must solve the quadratic programming problem to find values of the parameters α_i that maximize the following expression:

$$\sum_i \alpha_i - \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j y_i y_j (\mathbf{x_i} \cdot \mathbf{x_j}) \tag{4.5}$$

while $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$. The equation of the decision boundary itself can be written as follows, once the optimal α_is have been discovered.

$$h(\mathbf{x}) = \text{sign}(\sum_{\text{i}} \alpha_{\text{i}} \text{y}_{\text{i}} (\mathbf{x} \cdot \mathbf{x_i})) \tag{4.6}$$

As its name implies, the "support vectors" are those whose weight α_i for each data point are not equal to 0 [14, p.749].

4.4 Training Preparation

Once the features had been extracted and generated and classifiers had been selected, training of each classifier over a given feature set could occur. This section outlines the training preparation process including obtaining a semi-randomized distribution of suitable training data and converting the collected training data to a compatible format. The next section covers the cross-validation, training, and testing that occurred after the training data were prepared. Section 4.5 also summarizes the results and measurements from the training and test runs.

4.4.1 Obtaining the Training Data

For each of the classifiers defined in the previous section, a common set of sample features was used to train each learning algorithm. All training samples for both pulsars and non-pulsars were taken from the GBT350 drift survey data. As this is a two-class learning problem, 111 positively identified pulsars were used for the positive pulsar (1) class and 100 random non-pulsars were used for the negative pulsar (2) class.

To obtain this data two Python scripts were created. First, a 'retrieveNegatives.py' script was developed. As its name implies, its purpose was to download a subset of the non-pulsars to be used for training the classifiers. The WVU Astrophysics Department has a MySQL database in which they keep information about the CDPs that have already been analyzed. The script worked by retrieving a specified number of non-pulsar pointing names by first querying the database for a list of candidate names and their relative path location. It then proceeded to download the compressed Prepfold file for each pointing from the server and extract it. A randomize function was added to the SQL query in an an attempt to gather a true random sampling of non-pulsars and prevent introducing any bias into the training of the negative pulsar classification.

Next, a 'retrievePositives.py' script was developed to retrieve the pulsar candidate pointings that had already been positively identified as pulsars by the researchers at the WVU Astrophysics Department. Unlike when the negative pulsar pointings were retrieved, the positive pointings could not be queried in the MySQL database due to incompleteness of

the classification data contained at that time. Instead, they were retrieved from a directory containing the current list of positively identified pulsars. The researchers kept a running list of all positive pulsars and copied the candidate pointing Postscript file to a positive results directory. The 'retrievePositives.py' script worked by running down the list of positive pointing names (e.g. GBT350drift_54233_0420-0354_DM48.81_Z0_ACCEL_Cand_1.pfd.ps) and retrieving the compressed Prepfold file from amongst all the other (non-pulsar or not yet examined) candidate pointings on the WVU Astrophysics cluster. The compressed file was then downloaded and extracted. This was performed for all positively identified pulsars that had been discovered to date in the GBT350 drift survey data.

As previously discussed, the Postscript files do not contain the necessary text and graph data on which to extract feature sets. Therefore the modified 'exportPfd' application and Python scripts discussed in sections 4.1.1 and 4.2 were run on the extracted Prepfold data to generate the same features sets generated on the WVU Astrophysics cluster.

4.4.2 Converting Features to ARFF

To train the classifiers, the Weka Toolkit Java API was used. First, a small Java application was developed to convert the feature data collected by the 'consolidateFeatures.py' script from a Comma Separated Values (CSV) file to an Attribute-Relation File Format (ARFF) that was more compatible with Weka. The ARFF format is an ASCII text file format used for enumerating a list of instances with a common set of attributes. The Department of Computer Science of the University of Waikato developed ARFF files for use on the Machine Learning Project.[5]

ARFF files are divided into two sections - header and data. The header section describes the relation and defines each of the attributes and their types. See 4.2 for a sample pulsar header using a four-feature training set.

The data section consists of a listing of all the input vectors, or instance data, for training or classification. Each input vector must order the attributes as defined by the header.

Once the training set of known pulsars and known non-pulsars was converted to this

[5]Attribute-Relation File Format (ARFF) - `http://www.cs.waikato.ac.nz/ml/weka/arff.html`

```
@relation pulsar - dispersion measure vs. red. chi^2

@attribute maxY–MinY    numeric
@attribute meanY        numeric
@attribute stdDevY      numeric
@attribute class        {1,2}
```

Figure 4.2: Example ARFF Header Section

```
@data
3.115,1.653107,0.705869,1
2.408,1.240327,0.426928,1
3.874,2.726855,1.153033,1
3.612,2.682682,0.78183,1
1.628,1.133579,0.282525,1
2.168,1.563149,0.622684,2
1.5,0.883188,0.249613,2
0.891,0.572782,0.186254,2
0.075,0.074861,0.019963,2
1.375,0.701918,0.411993,2
```

Figure 4.3: Example ARFF Data Section

format, training of the classifiers consisted of cross-validation on the 211 instances. Using Weka's built in cross-validation API, the Java application exercised each of the classifiers over 10 folds of the training data. Cross-validation is a method by which a smaller training set can be used, in its entirety, to better train the classifiers. This is accomplished by breaking the data into N groups of equal size, where N is the number of folds. Then each of the groups from 1..N are classified using the remaining N - 1 groups to train the classifiers. This process is repeated for all N groups and the results averaged together to present the performance of the cross-validation training method for each classifier. This form of cross-validation is also known as the 'leave-one-out' technique [12].

Table 4.1: Weka Statistics and Definitions

Weka Statistics	
Correctly Classified Instances	The number of samples that were correctly classified
Incorrectly Classified Instances	The number of samples that were incorrectly classified as a class other than that which they actually are
Kappa Statistic	A measure of the agreement of prediction with the correct class (1.0 is complete agreement)
Mean Absolute Error	Statistical measurement of the closeness of a prediction to the actual outcome
Root Mean Squared Error	Statistical measurement of error between the pairwise differences between the predicted model and actual model
Relative Absolute Error	The normalized total absolute error
Root Relative Squared Error	Reduces relative squared error to the same dimensions as the prediction by applying the suare root
Total Number of Instances	Total number of all samples of all classes
TP Rate	*True Positive Rate* - The ratio of samples that were classified as class x to the number samples that actually have class x
FP Rate	*False Positive Rate* - The ratio of samples that were classified as class x, but belong to a different class, to all samples which are not class x
Precision	The ratio of samples that actually have class x to all those that were classified as class x
Recall	Equivalent to True Positive Rate
F-Measure	A combined measure of Precision and Recall computed as $(2 * Recall * Precision)/(Recall + Precision)$

4.5 Cross-validation

The Weka Toolkit API provides a wealth of statistics to measure the performance of the classifier with the training and test data sets. For each of the following results, the Weka calculations have been consolidated into tables that combine the feature vector used to train the classifier, classifier used, and Weka statistical results.

The Weka statistics calculated for each 10-fold cross-validation are listed in Table 4.1. These are the default statistics produced by Weka, and they were used to determine the performance of each of the classifiers given the provided feature vector as input.

The first 10-fold cross-validation test that was performed was upon the $\{MaxY - MinY,$

MeanY, StdDevY} feature vector using the negative pulsar training set that was non-randomized. This non-randomized negative pulsar training set contained data purely from the 'GBT350drift_54287' folder, or the same region of the observed sky. Because it was assumed that using a localized region of training data may introduce a significant bias in the training of the classifiers, later tests were adapted to use a second training set that included negative pulsars randomly sampled over the entire GBT350 drift survey.

The details of the cross-validation process will be outlined using this first test run. However, for the remainder of this section only the combined Weka statistical output will be provided. The process utilized remained the same, with the exception of the use of the randomized negative pulsar data set. Each consolidated set of feature data was converted from CSV to ARFF format and loaded into the 'Classifiers.java' application which instantiated each of the three aforementioned classifiers and performed 10-fold cross-validation. The output was recorded to a results folder for analysis.

The output from the 'Classifiers.java' program and Weka Toolkit generated the following results. For the J48 classifier, Weka created a decision tree representing the best separation of the training space and that yielded the 93.3% correct classification of the training instances. The decision tree is displayed in Figure 4.4.

```
meanY <= 1.194113
|   meanY <= 1.092356: 2 (98.0/5.0)
|   meanY > 1.092356
|   |   maxY–minY <= 0.912: 2 (2.0)
|   |   maxY–minY > 0.912
|   |   |   maxY–minY <= 1.667: 1 (4.0)
|   |   |   maxY–minY > 1.667: 2 (2.0)
meanY > 1.194113: 1 (105.0/3.0)
Number of Leaves  :     5
Size of the tree :      9
```

Figure 4.4: Weka Generated J48 Pruned Tree

The output for the Support Vector Machine using Binary Sequential Minimal Optimiza-

tion, as generated by Weka, shows the normalized weights for each feature attribute. This output is displayed in Figure 4.5.

```
Kernel used:
Linear Kernel: K(x,y) = <x,y>
Classifier for classes: 1, 2
BinarySMO

Machine linear: showing attribute weights,
not support vectors.
        -0.3214 * (normalized) maxY-minY
 +      -0.4616 * (normalized) meanY
 +      -0.3481 * (normalized) stdDevY
 -       0.9738

Number of kernel evaluations: 654 (45.771% cached)
```

Figure 4.5: Weka Generated Kernel Parameters

The third and final classifier used for each run, the Naive Bayes classifier, generated statistical weights for each of the attributes over the entire set of training samples. This yielded the probability weights displayed in Figure 4.6.

Of all the 10-fold cross-validation training runs, the *{MaxY − MinY, MeanY, StdDevY}* vector with the non-randomized pulsar training samples performed with the lowest error rate and, hence, performed the best with regard to correctly classifying the training data. This test run resulted in a 93.3% correct classification rate or 197 (correct classifications) / 211 (total instances).

The confusion matrix for each of the classifiers and the *{MaxY − MinY, MeanY, StdDevY}* feature vector is displayed in Table 4.4. One immediately observable fact from this data is that the Support Vector Machine classified all training samples as pulsars with no instances classified as non-pulsars. It is possible the SVM was misconfigured. However, the J48 classifier correctly classified 103 pulsars as 'pulsars' and 94 non-pulsars as 'non-pulsars'

Attribute	Class 1 (0.53)	2 (0.47)
maxY–minY		
mean	139.5698	0
std. dev.	532.8657	2.6401
weight sum	111	100
precision	15.8407	15.8407
meanY		
mean	48.0494	0
std. dev.	177.9714	0.8827
weight sum	111	100
precision	5.2964	5.2964
stdDevY		
mean	36.3835	0
std. dev.	136.1275	0.6612
weight sum	111	100
precision	3.9672	3.9672

Figure 4.6: Weka Generated Baysian Statistics

with only 14 misclassifications.

The remaining 10-fold cross-validation test runs were generated on the randomized non-pulsar data set. The exact process was repeated for each of the feature vectors - *{MinY, MedianY, MaxY, MeanY}*, *{MaxY − MinY, MeanY, StdDevY}*, *{XValAtPeak, fitCurve-MaxToDMChiMaxRatio}* which were generated on the Dispersion Measure vs. Reduced χ^2 graph.

Table 4.2: Cross-validation for {MaxY − MinY, MeanY, StdDevY} Calculated on DM Graph with Non-randomized Negative Pulsar Samples

10-fold Cross-validation Performance Metrics			
Feature Vector - {MaxY − MinY, MeanY, StdDevY}			
Classifier	**J-48**	**SVM**	**Bayes**
Correctly Classified Instances	197 (93.3649%)	111 (52.6066%)	136 (64.455%)
Incorrectly Classified Instances	14 (6.6351%)	100 (47.3934%)	75 (35.545%)
Kappa statistic	0.8671	0	0.3127
Mean absolute error	0.1046	0.4739	0.3597
Root mean squared error	0.2508	0.6884	0.5977
Relative absolute error	20.9828%	95.0403%	72.1319%
Root relative squared error	50.2189%	137.8702%	119.6982%
Total Number of Instances	211	211	211

Table 4.3: Detailed Cross-validation Metrics by Classifier and Class for {MaxY − MinY, MeanY, StdDevY} Calculated on DM Graph with Non-randomized Negative Pulsar Samples

Detailed 10-fold Cross-validation Performance Metrics by Class									
Feature Vector - {MaxY − MinY, MeanY, StdDevY}									
Classifier	**J-48**			**SVM**			**Bayes**		
	Class		*W. Avg.*	*Class*		*W. Avg.*	*Class*		*W. Avg.*
	1	2	-	1	2	-	1	2	-
TP Rate	0.928	0.94	0.934	1	0	0.526	0.324	1	0.645
FP Rate	0.06	0.072	0.066	1	0	0.526	0	0.676	0.32
Precision	0.945	0.922	0.934	0.526	0	0.277	1	0.571	0.797
Recall	0.928	0.94	0.934	1	0	0.526	0.324	1	0.645
F-Measure	0.936	0.931	0.934	0.689	0	0.363	0.49	0.727	0.602
ROC Area	0.922	0.922	0.922	0.5	0.5	0.5	0.683	0.683	0.683

Table 4.4: 10-fold Cross-validation Confusion Matrices Calculated on DM Graph with Non-randomized Negative Pulsar Samples

Cross-validation Confusion Matrices						
Feature Vector - {MaxY − MinY, MeanY, StdDevY}						
Classifier	**J48**		**SVM**		**Bayes**	
Class	*a*	*b*	*a*	*b*	*a*	*b*
$a = 1$	103	8	111	0	36	75
$b = 2$	6	94	100	0	0	100

Table 4.5: Cross-validation for {MinY, MedianY, MaxY, MeanY} Calculated on DM Graph with Randomized Negative Pulsar Samples

10-fold Cross-validation Performance Metrics			
Feature Vector - {MinY, MedianY, MaxY, MeanY}			
Classifier	**J-48**	**SVM**	**Bayes**
Correctly Classified Instances	192 (90.9953%)	111 (52.6066%)	122 (57.8199%)
Incorrectly Classified Instances	19 (9.0047%)	100 (47.3934%)	89 (42.1801%)
Kappa statistic	0.8191	0	0.1891
Mean absolute error	0.1405	0.4739	0.4198
Root mean squared error	0.2864	0.6884	0.6466
Relative absolute error	28.1663%	95.0403%	84.1862%
Root relative squared error	57.362%	137.8702%	129.4905%
Total Number of Instances	211	211	211

Table 4.6: Detailed Cross-validation Metrics by Classifier and Class for {MinY, MedianY, MaxY, MeanY} Calculated on DM Graph with Randomized Negative Pulsar Samples

Detailed 10-fold Cross-validation Performance Metrics by Class									
Feature Vector - {MinY, MedianY, MaxY, MeanY}									
Classifier	**J-48**			**SVM**			**Bayes**		
	Class		*W. Avg.*	*Class*		*W. Avg.*	*Class*		*W. Avg.*
	1	2	-	1	2	-	1	2	-
TP Rate	0.928	0.89	0.91	1	0	0.526	0.207	0.99	0.578
FP Rate	0.11	0.072	0.092	1	0	0.526	0.01	0.793	0.381
Precision	0.904	0.918	0.91	0.526	0	0.277	0.958	0.529	0.755
Recall	0.928	0.89	0.91	1	0	0.526	0.207	0.99	0.578
F-Measure	0.916	0.904	0.91	0.689	0	0.363	0.341	0.69	0.506
ROC Area	0.906	0.906	0.906	0.5	0.5	0.5	0.782	0.782	0.782

Table 4.7: 10-fold Cross-validation Confusion Matrices Calculated on DM Graph with Randomized Negative Pulsar Samples

Cross-validation Confusion Matrices						
Feature Vector - {MinY, MedianY, MaxY, MeanY}						
Classifier	**J48**		**SVM**		**Bayes**	
Class	*a*	*b*	*a*	*b*	*a*	*b*
$a = 1$	103	8	111	0	23	88
$b = 2$	11	89	100	0	1	99

Table 4.8: Cross-validation for {MaxY − MinY, MeanY, StdDevY} Calculated on DM Graph with Randomized Negative Pulsar Samples

10-fold Cross-validation Performance Metrics			
Feature Vector - {MaxY − MinY, MeanY, StdDevY}			
Classifier	**J-48**	**SVM**	**Bayes**
Correctly Classified Instances	194 (91.9431%)	111 (52.6066%)	125 (59.2417%)
Incorrectly Classified Instances	17 (8.0569%)	100 (47.3934%)	86 (40.7583%)
Kappa statistic	0.8383	0	0.2153
Mean absolute error	0.1357	0.4739	0.406
Root mean squared error	0.2734	0.6884	0.6361
Relative absolute error	27.2063%	95.0403%	81.4142%
Root relative squared error	54.7632%	137.8702%	127.3957%
Total Number of Instances	211	211	211

Table 4.9: Detailed Cross-validation Metrics by Classifier and Class for {MaxY − MinY, MeanY, StdDevY} Calculated on DM Graph with Randomized Negative Pulsar Samples

Detailed 10-fold Cross-validation Performance Metrics by Class									
Feature Vector - {MaxY − MinY, MeanY, StdDevY}									
Classifier	**J-48**			**SVM**			**Bayes**		
	Class		*W. Avg.*	*Class*		*W. Avg.*	*Class*		*W. Avg.*
	1	2	-	1	2	-	1	2	-
TP Rate	0.928	0.91	0.919	1	0	0.526	0.234	0.99	0.592
FP Rate	0.09	0.072	0.082	1	0	0.526	0.01	0.766	0.368
Precision	0.92	0.919	0.919	0.526	0	0.277	0.963	0.538	0.762
Recall	0.928	0.91	0.919	1	0	0.526	0.234	0.99	0.592
F-Measure	0.924	0.915	0.919	0.689	0	0.363	0.377	0.697	0.529
ROC Area	0.906	0.906	0.906	0.5	0.5	0.5	0.689	0.688	0.688

Table 4.10: 10-fold Cross-validation Confusion Matrices Calculated on DM Graph with Randomized Negative Pulsar Samples

Cross-validation Confusion Matrices						
Feature Vector - {MaxY − MinY, MeanY, StdDevY}						
Classifier	**J48**		**SVM**		**Bayes**	
Class	*a*	*b*	*a*	*b*	*a*	*b*
$a = 1$	103	8	111	0	26	85
$b = 2$	9	91	100	0	1	99

Table 4.11: Cross-validation for {XValAtPeak, fitCurveMaxToDMChiMaxRatio} Calculated on DM Graph

10-fold Cross-validation Performance Metrics			
Feature Vector - {XValAtPeak, fitCurveMaxToDMChiMaxRatio}			
Classifier	**J-48**	**SVM**	**Bayes**
Correctly Classified Instances	122 (57.8199%)	125 (59.2417%)	120 (56.872%)
Incorrectly Classified Instances	89 (42.1801%)	86 (40.7583%)	91 (43.128%)
Kappa statistic	0.1313	0.1462	0.1081
Mean absolute error	0.4623	0.4076	0.4386
Root mean squared error	0.5159	0.6384	0.5171
Relative absolute error	92.7029%	81.7347%	87.9493%
Root relative squared error	103.3166%	127.8556%	103.5644%
Total Number of Instances	211	211	211

Table 4.12: Detailed Cross-validation Metrics by Classifier and Class for {XValAtPeak, fitCurveMaxToDMChiMaxRatio} Calculated on DM Graph

Detailed 10-fold Cross-validation Performance Metrics by Class									
Feature Vector - {XValAtPeak, fitCurveMaxToDMChiMaxRatio}									
Classifier	**J-48**			**SVM**			**Bayes**		
	Class		*W. Avg.*	*Class*		*W. Avg.*	*Class*		*W. Avg.*
	1	2	-	1	2	-	1	2	-
TP Rate	0.838	0.29	0.578	1	0.14	0.592	0.865	0.24	0.569
FP Rate	0.71	0.162	0.45	0.86	0	0.452	0.76	0.135	0.464
Precision	0.567	0.617	0.591	0.563	1	0.77	0.558	0.615	0.585
Recall	0.838	0.29	0.578	1	0.14	0.592	0.865	0.24	0.569
F-Measure	0.676	0.395	0.543	0.721	0.246	0.496	0.678	0.345	0.521
ROC Area	0.559	0.559	0.559	0.57	0.57	0.57	0.586	0.586	0.586

Table 4.13: 10-fold Cross-validation Confusion Matrices Calculated on DM Graph with Randomized Negative Pulsar Samples

Cross-validation Confusion Matrices						
Feature Vector - {XValAtPeak, fitCurveMaxToDMChiMaxRatio}						
Classifier	**J48**		**SVM**		**Bayes**	
Class	*a*	*b*	*a*	*b*	*a*	*b*
$a = 1$	93	18	111	0	96	15
$b = 2$	71	29	86	14	76	24

Chapter 5

Conclusion

5.1 Results

5.1.1 Full Test Run

After examining the output from the cross-validation test runs, the best classifier, J48, and feature vector, *{MaxY − MinY, MeanY, StdDevY}* calculated on the DM vs. Reduced χ^2 graph, were used to run a full test. The J48 classifier was again retrained on the same 211 pointings used for cross-validation and the consolidated list of potential pointings was fed through the trained classifier. This list included 441,062 pointings and only contained one pointing that was also within the positive pulsar training set. The format of this AARF input file was much like the format of the AARF training file only it had a placeholder character '?' for the actual classification and an additional, unprocessed attribute string 'location'. The 'location' attribute was used to hold the full path to the pointing, so that the name, position, DM, and CDP were all retrievable for validation of the results.

The entire training and classification process took only minutes running within a virtual machine on a laptop. The output was then split into the two respective classes. The trained J48 classifier identified 47,281 pointings (11%) as pulsars out of the total 441,062 pointings. Given the cross-validation test runs yielded an optimum performance of 92% on the training data, this seemed a reasonable proportion for the classification of the test results.

To validate the success of the test run, a script searched for each of the manually identified

pulsars in the positively classified pulsar output. Each pulsar was searched by its 'Pointing Location'. Since the PRESTO software package was configured to output 30 of the best profiles at different potential DMs for the same pointing, oftentimes multiple positively classified candidates occurred for the same pulsar.

Only one pulsar out of the 34 new pulsars discovered from the GBT350 drift survey was actually used in the training set - J1327−0745. It was recovered in the full test run as would be expected of training data. The more interesting discovery is how many new pulsars were recovered, positively identified as pulsars, within the test data and were not part of the training data. Also of great interest was how close the identified candidate pointings' DMs were to the actual DM of the pulsar. Table 5.1 lists the results of the search for each new pulsar in the classifier output. Entries listed as "Not in test" were not available in the sample set of data retrieved from the WVU Astrophysics server. Entries in italics were candidates at nearly the exact DM of the pulsar for that location.

The trained J48 Classifier was able to recover 11 of the 34 pulsars, or 32%. However, due time constraints and limited access to the data, the test was only executed on approximately 1/5th of the overall candidates generated by the GBT350 drift survey. Since only a portion of the survey was covered by the test, one more validation step needed to occur. Each of the manually discovered pulsars that was not found in the positive classifications needed further examination to determine if it was actually contained within the 441,062 pointings used for this test. Interestingly, the 23 pulsars not positively identified as pulsars by the trained classifier were not present in the test set. With this new knowledge, it was now possible to say that all 11 known pulsars contained within the data had been recovered, or 100%.

Further analysis was performed on the 47,281 positively classified pointings to determine if new, undiscovered pulsars might be contained within the set. First, these were pared down by eliminating CDPs with a DM value < 3, a Reduced χ^2 value < 5, and a period < 1 ms. This resulted in 11,988 pointings that had to be manually inspected. Of those remaining, 910 were selected via manual review as potential pulsars. Over 860 CDPs were confirmed to belong to known pulsars, while some were discarded as RFI. Unfortunately, at this time, no new pulsars have been discovered from this set.

Table 5.1: New Pulsars Recovered by Automated Analysis of the GBT350 Drift Survey - Using J48 and {MaxY − MinY, MeanY, StdDevY} Calculated on DM

Pulsar Name	**Positive Matches**
J1023+00	Not in test
J2256−10	Not in test
J1327−07	*GBT350drift_54290_1327-0745_DM27.90_Z0_ACCEL_Cand_1.pfd*
	GBT350drift_54290_1327−0745_DM99.00_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54290_1327−0745_DM108.36_Z0_ACCEL_Cand_2.pfd
	GBT350drift_54290_1327−0745_DM27.90_Z50_ACCEL_Cand_1.pfd
J0336+17	Not in test
J1923+25	Not in test
J1738−08	*GBT350drift_54289_1737−0817_DM55.35_Z50_ACCEL_Cand_1.pfd*
	GBT350drift_54289_1737−0817_DM55.50_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54289_1737−0817_DM1.74_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54289_1737−0817_DM55.35_Z0_ACCEL_Cand_1.pfd
J0931−19	Not in test
J2221−01	*GBT350drift_54267_2221−0131_DM3.24_Z0_ACCEL_Cand_1.pfd*
	GBT350drift_54267_2221−0131_DM3.24_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM104.49_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM88.47_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM100.20_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM2.91_Z0_ACCEL_Cand_2.pfd
	GBT350drift_54267_2221−0131_DM2.91_Z50_ACCEL_Cand_2.pfd
	GBT350drift_54267_2221−0131_DM88.47_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM104.49_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54267_2221−0131_DM100.20_Z0_ACCEL_Cand_1.pfd
J0343+04	Not in test
J1643−10	Not in test
J1444+18	Not in test
J1941+01	Not in test
J1543−07	GBT350drift_54255_1543−0705_DM64.56_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54255_1543−0705_DM31.44_Z50_ACCEL_Cand_1.pfd

	GBT350drift_54255_1543−0705_DM49.83_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54255_1543−0705_DM29.70_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54255_1543−0705_DM71.28_Z50_ACCEL_Cand_1.pfd
J1911+22	Not in test
J1613+20	Not in test
J1502+00	Not in test
J1134+24	Not in test
J2013−20	Not in test
J2013−06	*GBT350drift_54285_2013−0650_DM64.32_Z0_ACCEL_Cand_1.pfd*
	GBT350drift_54285_2013−0650_DM64.20_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54285_2013−0650_DM86.58_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54285_2013−0650_DM1.11_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54285_2013−0650_DM84.36_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54285_2013−0650_DM1.11_Z0_ACCEL_Cand_1.pfd
J1930−01	Not in test
J1745−01	GBT350drift_54274_1745−0104_DM77.94_Z0_ACCEL_Cand_1.pfd
J1736−02	*GBT350drift_54233_1735−0240_DM55.65_Z0_ACCEL_Cand_1.pfd*
	GBT350drift_54233_1735−0240_DM55.65_Z50_ACCEL_Cand_1.pfd
J1519−06	GBT350drift_54253_1518−0618_DM108.81_Z50_ACCEL_Cand_1.pfd
J1918−10	Not in test
J1902−08	GBT350drift_54288_1903−0848_DM47.22_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54288_1903−0848_DM56.97_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54288_1903−0848_DM67.23_Z0_ACCEL_Cand_2.pfd
	GBT350drift_54288_1903−0848_DM65.04_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54288_1903−0848_DM67.47_Z0_ACCEL_Cand_1.pfd
J1633−20	Not in test
J1556−05	Not in test
J1853−06	GBT350drift_54285_1852−0649_DM39.09_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54285_1852−0649_DM43.92_Z0_ACCEL_Cand_1.pfd
J2033−19	Not in test
J1547−09	Not in test

J0459−05	Not in test
J2033+00	Not in test
J1758−10	Not in test
J2111+21	GBT350drift_54287_2111+2114_DM62.49_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54287_2111+2114_DM64.56_Z0_ACCEL_Cand_1.pfd
	GBT350drift_54287_2111+2114_DM59.73_Z0_ACCEL_Cand_2.pfd
	GBT350drift_54287_2111+2114_DM57.51_Z0_ACCEL_Cand_4.pfd
	GBT350drift_54287_2111+2114_DM92.79_Z0_ACCEL_Cand_2.pfd
	GBT350drift_54287_2111+2114_DM62.55_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54287_2111+2114_DM57.18_Z50_ACCEL_Cand_2.pfd
	GBT350drift_54287_2111+2114_DM65.79_Z50_ACCEL_Cand_1.pfd
	GBT350drift_54287_2111+2114_DM58.86_Z50_ACCEL_Cand_3.pfd

5.2 Continuing Research

Ideally, it would have been directly pertinent to this research to obtain the remaining pointings from the GBT350 drift survey to process and determine if the 23 pulsars not contained within this test data set were recovered from the full data set. However, given the time constraints imposed, this was not possible. At some future time, this data will be processed to determine the true performance of this machine learning framework on the entire survey.

It is regrettable that in the allotted time available for this research optimal feature selection was not examined to better determine the features that most likely model the two-class pulsar classification problem. Automatic feature selection would have enabled more optimum feature vectors to be created and would have hopefully increased the performance of the best classifier above the 92% performance rate currently experienced. With better performance, the number of false positives should diminish, meaning even less time should be required to review the identified candidates. It is highly unlikely that the feature vectors generated from the Reduced χ^2 graph were the best for training and classifying the pulsars

in this survey. This graph is only one of six graphs contained within the CDPs and would seem suboptimal in comparison to features that could be generated from the Pulse Profile and potentially the Subintegrations graphs. However, the Reduced χ^2 graph did present a simple starting point from which to experiment with the pulsar results.

New classifiers, such as Artificial Neural Networks (ANNs), could be experimented with to determine if there exists a better classifier for identifying pulsars than that of the J48 decision tree algorithm suggested by this research. Some of the current research examined in Chapter 2 points to the success of other classifiers with regard to this task.

Using a hierarchical or sequential classifier paradigm could also help to improve the performance of the framework. The classification could occur in such a manner that the output from one classifier would become the input to the next. Decisions could be made at each juncture, allowing for greater flexibility in handling different classes. Reinforcement learning could be used to allow the classifier to update itself over multiple training examples.

Another interesting approach to classifying this data would have been to extend it from a two-class problem to a three-class problem. This still would have involved training the classifiers with both pulsars and non-pulsars as well as a mix of other exotic, or unusual, pulsars. This third class could act like a 'maybe' category and could lend itself to better identifying pulsars, or other phenomena, that do not strictly conform the the two-class separation.

5.3 Summary

The preliminary results of this research show much promise that the CDP review process of the GBT350 drift survey data can benefit greatly from the application of machine learning and pattern recognition techniques. If Eatough et al. (2010) are correct in their prediction that the average time required to inspect a candidate pulsar is between 1 and 300 seconds, splitting the difference and applying that logic to the 2.5 million CDPs generated on the GBT350 drift survey still yields an average review period of over 11 years for a single researcher to complete [4]!

While this research was performed over the course of 5 years, no single execution of a

script or application took more than 1 to 2 days to complete. The largest tasks included running the modified Prepfold 'exportPfd' application over the 441,062 pointings and calculating features, in triplicate, for each pointing. However, both of these were completed in short order and with modest computing power. Now, that this generic framework has been created, the entire process could be reapplied to the entire GBT350 drift survey (2.5 million pointings) in an estimated time frame of 2 weeks.

The results presented by Boyles et al. and Lynch et al. (2013) highlight 31 of the 34 pulsars discovered by the current methodology (see Table 5.2). While each manually identified pulsar candidate was verified via follow-up observations and timing solutions were carefully depicted for each pulsar, the entire process took nearly 6 years to complete. Of those 6 years, approximately 3 were required to manually review the CDPs by a number of different reviewers and organizations simultaneously. If the performance of this framework test can be linearly extrapolated to the entire 2.5 million diagnostic plots, meaning the overall number of potential CDPs identified by the framework remains 11% of the total and pulsars continue to be accurately detected by the framework, then all relevant output from the framework could be reviewed in 118 days instead of 3 years! This, of course, assumes that the reviewing of the CDPs happens after PRESTO processing completes, but in reality some review occurs in parallel with the processing.

Given the success at applying three generic classifiers to three very elementary feature vectors with approximately 92% accuracy during training, it is the author's opinion that even higher rates of accuracy and automation could be achieved with further refinement - primarily through more feature and classifier experimentation. Any optimization gained would only help to further reduce the amount of manual inspection required. The author also hopes to apply this framework to other large pulsar surveys in an effort to better tune and measure performance, ensure a generic approach is adhered to by the framework, and ultimately identify new, undiscovered pulsars.

Table 5.2: New Pulsars Discovered by Manual Analysis of the GBT350 Drift Survey

Pulsar	**P(ms)**	**DM parsec/cm^2**	**Institution**	**Pointing Location**
J1023+00	1.69	14.32	McGill	54279_0009_0060
J2256−10	2.29	13.77	UBC	54281_2257−1018
J1327−07	2.68	27.92	WVU	54290_1327−0745
J0336+17	2.73	21.3	WVU	54224_0006_0240
J1923+25	3.79	18.85	McGill	54279_1924+2519
J1738−08	4.18	55.31	WVU	54289_1737−0817
J0931−19	4.64	41.48	UTB	54296_0930−1906
J2221−01	32.8	3.19	WVU	54267_2221−0131
J0343+04	39.1	40.56	NRAO	54324_0348+0428
J1643−10	62.8	76	UTB	54281_1643−1015
J1444+18	132	16.98	UTB	54228_0003_0371
J1941+01	217	51.04	NRAO	54317_1941+0129
J1543−07	242	30.37	UTB	54255_1543−0705
J1911+22	320	47.03	NRAO	54236_1912+2240
J1613+20	427	19.93	NRAO	54271_1612+2000
J1502+00	464	22.19	NRAO	54269_1502+0038
J1134+24	501	23.26	UTB	54227_0002_0227
J2013−20	544	38.85	UBC	54295_2013−2008
J2013−06	580	63.68	WVU	54285_2013−0650
J1930−01	594	35.95	UTB	54273_1930−0153
J1745−01	680	67.45	UTB	54274_1745−0104
J1736−02	783	54.75	WVU	54233_1735−0240
J1519−06	795	28.27	WVU	54253_1518−0618
J1918−10	799	62.44	NRAO	54316_1918−1059
J1902−08	887	66.79	WVU	54288_1903−0848
J1633−20	936	48.63	UBC	54295_1633−2006
J1556−05	975	23.93	McGill	54241_1555−0503
J1853−06	1048	43.25	WVU	54285_1852−0649
J2033−19	1282	23.81	UBC	54296_2033−1939
J1547−09	1577	37.62	UBC	54282_1547−0944
J0459−05	1883	47.64	McGill	54240_0459−0505
J2033+00	2506	37.62	McGill	54280_2033+0030
J1758−10	2513	119.74	UMW	54281_1758−1015
J2111+21	3952	58.85	WVU	54287_2111+2114

References

[1] D. Bhattacharya, "Detection of radio emission from pulsars," *NATO ASI*, pp. 103–128, 1998.

[2] J. Boyles, R. S. Lynch, S. M. Ransom, I. H. Stairs, D. R. Lorimer, M. A. McLaughlin, J. W. T. Hessels, V. M. Kaspi, V. I. Kondratiev, A. Archibald, A. Berndsen, R. F. Cardoso, A. Cherry, C. R. Epstein, C. Karako-Argaman, C. A. McPhee, T. Pennucci, M. S. E. Roberts, K. Stovall, and J. van Leeuwen, "The green bank telescope 350 mhz drift-scan survey. I. Survey observations and the discovery of 13 pulsars," *The Astrophysical Journal*, vol. 763, no. 2, pp. 80, 2013.

[3] Ryan S. Lynch, Jason Boyles, Scott M. Ransom, Ingrid H. Stairs, Duncan R. Lorimer, Maura A. McLaughlin, Jason W. T. Hessels, Victoria M. Kaspi, Vladislav I. Kondratiev, Anne M. Archibald, Aaron Berndsen, Rogerio F. Cardoso, Angus Cherry, Courtney R. Epstein, Chen Karako-Argaman, Christie A. McPhee, Tim Pennucci, Mallory S. E. Roberts, Kevin Stovall, and Joeri van Leeuwen, "The green bank telescope 350 mhz drift-scan survey II: Data analysis and the timing of 10 new pulsars, including a relativistic binary," *The Astrophysical Journal*, vol. 763, no. 2, pp. 81, 2013.

[4] R.P. Eatough, N. Molkenthin, M. Kramer, A. Noutsos, M.J. Keith, et al., "Selection of radio pulsar candidates using artificial neural networks," 2010.

[5] D.R. Lorimer, *Handbook of Pulsar Astronomy*, Cambridge Observing Handbooks for Research Astronomers. Cambridge University Press, 2004.

[6] W. R. Burns and B. G. Clark, "Pulsar Search Techniques," *Astronomy & Astrophysics*, vol. 2, pp. 280–287, July 1969.

[7] T. H. Hankins, "Microsecond Intensity Variations in the Radio Emissions from CP 0950," *The Astrophysical Journal*, vol. 169, pp. 487, Nov. 1971.

[8] Dan Gao, Yan-Xia Zhang, and Yong-Heng Zhao, "Support vector machines and kd-tree for separating quasars from large survey data bases," *Monthly Notices of the Royal Astronomical Society*, vol. 386, no. 3, pp. 1417–1425, May 2008.

[9] R. Carballo, J. I. González-Serrano, C. R. Benn, and F. Jiménez-Luján, "Use of neural networks for the identification of new $z \geq 3.6$ QSOs from FIRST-SDSS DR5," *Monthly Notices of the Royal Astronomical Society*, vol. 391, no. 1, pp. 369–382, 2008.

[10] Thomas M. Mitchell, *Machine Learning*, McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[11] D. Michie, D. J. Spiegelhalter, and C.C. Taylor, *Machine Learning, Neural and Statistical Classification*, 1994.

[12] Christopher M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.

[13] J. Ross Quinlan, *C4.5: programs for machine learning*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993.

[14] Stuart J. Russell and Peter Norvig, *Artificial Intelligence: A Modern Approach (2nd Edition)*, Prentice Hall, December 2002.

[15] John C. Platt, "Advances in kernel methods," chapter Fast training of support vector machines using sequential minimal optimization, pp. 185–208. MIT Press, Cambridge, MA, USA, 1999.

Appendix A

Glossary

Table A.1: Glossary

Glossary	
ANN	Artificial Neural Network
API	Application Programming Interface
ARFF	Attribute-Relation File Format
CDI	Cyber-enabled Discovery and Innovation
CDP	Candidate Diagnostic Plot
CSV	Comma Separated Values
CTIO	Cerro Tololo Inter-American Observatory
DM	Dispersion Measure
DR5	Data Release 5
FIRST	Faint Images of the Radio Sky at Twenty cm survey
PCA	Principal Component Analysis
GBT	Green Bank Telescope
GBT350 drift survey	Green Bank Telescope 350MHz Drift-Scan Radio Survey
GUPPI	Green Bank Ultimate Pulsar Processing Instrument
LNA	Low-Noise Amplifier
NN	Neural Network
NED	NASA/IPAC Extragalactic Database
NRAO	National Radio Astronomy Observatory
NSF	National Science Foundation

PMPS	Parkes Multi-beam Pulsar Survey
QSO	Quasi-Stellar Objects
RFI	Radio Frequency Interference
SDSS	Sloan Digital Sky Survey
SNR	Super Nova Remnant
SVM	Support Vector Machine
Correctly Classified Instances	The number of samples that were correctly classified
F-Measure	A combined measure of Precision and Recall computed as $(2 * Recall * Precision)/(Recall + Precision)$
FP Rate	*False Positive Rate* - The ratio of samples that were classified as class x, but belong to a different class, to all samples which are not class x
Incorrectly Classified Instances	The number of samples that were incorrectly classified as a class other than that which they actually are
Kappa Statistic	A measure of the agreement of prediction with the correct class (1.0 is complete agreement)
Mean Absolute Error	Statistical measurement of the closeness of a prediction to the actual outcome
Precision	The ratio of samples that actually have class x to all those that were classified as class x
Recall	Equivalent to True Positive Rate
Relative Absolute Error	The normalized total absolute error
RMS	Root Mean Squared (error) - Statistical measurement of error between the pairwise differences between the predicted model and actual model
Root Relative Squared Error	Reduces relative squared error to the same dimensions as the prediction by applying the suare root
Total Number of Instances	Total number of all samples of all classes

TP Rate	*True Positive Rate* - The ratio of samples that were classified as class x to the number samples that actually have class x