Monthly Notices
of the
ROYAL ASTRONOMICAL SOCIETY

MNRAS **459,** 1104–1123 (2016)
Advance Access publication 2016 April 15

doi:10.1093/mnras/stw656

Fifty years of pulsar candidate selection: from simple filters to a new principled real-time classification approach

R. J. Lyon,[1]★ B. W. Stappers,[2] S. Cooper,[2] J. M. Brooke[1] and J. D. Knowles[1,3]

[1]*School of Computer Science, The University of Manchester, Manchester M13 9PL, UK*
[2]*Jodrell Bank Centre for Astrophysics, School of Physics and Astronomy, The University of Manchester, Manchester M13 9PL, UK*
[3]*School of Computer Science, University of Birmingham, Edgbaston, Birmingham B15 2TT, UK*

Accepted 2016 March 16. Received 2016 March 14; in original form 2015 June 24

ABSTRACT

Improving survey specifications are causing an exponential rise in pulsar candidate numbers and data volumes. We study the candidate filters used to mitigate these problems during the past 50 years. We find that some existing methods such as applying constraints on the total number of candidates collected per observation, may have detrimental effects on the success of pulsar searches. Those methods immune to such effects are found to be ill-equipped to deal with the problems associated with increasing data volumes and candidate numbers, motivating the development of new approaches. We therefore present a new method designed for online operation. It selects promising candidates using a purpose-built tree-based machine learning classifier, the Gaussian Hellinger Very Fast Decision Tree, and a new set of features for describing candidates. The features have been chosen so as to (i) maximize the separation between candidates arising from noise and those of probable astrophysical origin, and (ii) be as survey-independent as possible. Using these features our new approach can process millions of candidates in seconds (∼1 million every 15 s), with high levels of pulsar recall (90 per cent+). This technique is therefore applicable to the large volumes of data expected to be produced by the Square Kilometre Array. Use of this approach has assisted in the discovery of 20 new pulsars in data obtained during the Low-Frequency Array Tied-Array All-Sky Survey.

Key words: methods: data analysis – methods: statistical – pulsars: general.

1 INTRODUCTION

The search techniques used to isolate the radio emission of pulsars, are designed to find periodic broad-band signals exhibiting signs of dispersion caused by travel through the interstellar medium (ISM). Signals meeting these criteria are recorded as a collection of diagnostic plots and summary statistics, in preparation for analysis. Together these plots and statistics are referred to as a pulsar 'candidate', a possible detection of a new pulsar. Each candidate must be inspected by either an automated method, or a human expert, to determine their authenticity. Those of likely pulsar origin are highlighted for further analysis, and possibly allocated telescope time for confirmation observations. The remainder are typically ignored. The process of deciding which candidates are worthwhile investigating has become known as candidate 'selection'. It is an important step in the search for pulsars since it allows telescope time to be prioritized upon those detections likely to yield a discovery. Until recently (early 2000s) candidate selection was a predominately manual task. However advances in telescope receiver design, and the capabilities of supporting computational infrastructures, significantly increased the number of candidates produced by modern pulsar surveys (Stovall, Lorimer & Lynch 2013). Manual approaches therefore became impractical, introducing what has become known as the 'candidate selection problem'. In response, numerous graphical and automated selection methods were developed (Johnston et al. 1992; Edwards et al. 2001; Manchester et al. 2001; Keith et al. 2009; Navarro, Anderson & Freire 2003), designed to filter candidates in bulk. The filtering procedure used ranged in complexity from a simple signal-to-noise ratio (S/N) cut, through to more complex functions (Lee et al. 2013). In either case, automated approaches enabled large numbers of candidates to be selected at speed in a reproducible way.

Despite these advances the increasing number of candidates produced by contemporary pulsar surveys, tends to necessitate a pass of manual selection upon the candidates selected by software. Many have therefore turned to machine learning (ML) methods to build 'intelligent' filters (Eatough et al. 2010; Bates et al. 2012; Morello

*E-mail: robert.lyon@manchester.ac.uk

et al. 2014; Zhu et al. 2014), capable of reducing the dependence on human input. This has achieved some success. However these methods are often developed for a specific pulsar survey search pipeline, making them unsuitable for use with other surveys without modification. As a consequence, new selection mechanisms are often designed and implemented per survey. As more methods continue to emerge, it becomes increasingly unclear which of these best address the candidate selection problem, and under what circumstances. It is also unclear which are best equipped to cope with the trend for increasing candidate numbers, the overwhelming majority of which arise from noise. Existing approaches are not explicitly designed to mitigate noise, rather they are designed to isolate periodic detections. This does not achieve the same effect as explicitly mitigating noise. For example, isolating periodic candidates as potential pulsars, does not necessarily mitigate the impact of periodic noise. Thus, it is possible that these techniques will become less effective over time, as noise becomes responsible for an increasing proportion of all candidates detected.

Existing 'intelligent' approaches are also ill-equipped to deal with the data processing paradigm shift, soon to be brought about by next-generation radio telescopes. These instruments will produce more data than can be stored, thus survey data processing, including candidate selection, will have to be done online in real-time (or close to). In the real-time scenario, it is prohibitively expensive to retain all data collected (see Section 4.3.1). It therefore becomes important to identify and prioritize data potentially containing discoveries for storage. Otherwise such data could be discarded and discoveries missed. Thus, new techniques are required (Keane et al. 2014) to ensure preparedness for this processing challenge.

In this paper we describe a new candidate selection approach designed for online operation, that mitigates the impact of increasing candidate numbers arising from noise. We develop our arguments for producing such a technique in progressive stages. In Section 2, we describe the candidate generation process. We show that improvements in pulsar survey technical specifications have led to increased candidate output, and infer a trend for exponential growth in candidate numbers which we show to be dominated by noise. We also demonstrate why restricting candidate output based on simple S/N cuts, runs the risk of omitting legitimate pulsar signals. The trend in candidate numbers and the ineffectiveness of S/N filters, allows us to identify what we describe as a 'crisis' in candidate selection. In Section 3, we review the different candidate selection mechanisms employed during the past 50 years, to look for potential solutions to the issues raised in Section 2. Based on this review, in Section 4, we discuss these methods. We identify how all will be challenged by the transition to online processing required by telescopes such as the Square Kilometre Array (SKA), motivating the development of new approaches. In addition we critique the existing features used to describe pulsar candidates, fed as inputs to the ML methods employed by many to automate the selection process. In Section 5, we present our own set of eight candidate features, which overcome some of these deficiencies. Derived from statistical considerations and information theory, these features were chosen to maximize the separation between noise and non-noise arising candidates. In Section 6, we describe our new data stream classification algorithm for online candidate selection which uses these features. Section 6 also presents classification results that demonstrate the utility of the new approach, and its high level of pulsar recall. Finally, in Section 7 we summarize the paper, and comment on how the use of our method has helped to find 20 new pulsars during the Low-Frequency Array (LOFAR) Tied-Array All-Sky Survey (LOTAAS), though discovery details will be published elsewhere.

2 CANDIDATE GENERATION

Since the adoption of the fast Fourier transform (FFT) (Burns & Clark 1969; Taylor, Dura & Huguenin 1969; Hulse & Taylor 1974), the general pulsar search procedure has remained relatively unchanged. Signals focused at the receiver of a radio telescope observing at a central frequency f_c (MHz), with bandwidth B (MHz), are sampled and recorded at a pre-determined rate at intervals of $t_{\rm samp}$ (μs), chosen to maximize sensitivity to the class of signals being searched for. The data are subsequently split in to $n_{\rm chans}$ frequency channels, each of width $\Delta\upsilon$ (kHz). An individual channel contains $s_{\rm tot}$ samples of the signal taken at the interval $t_{\rm samp}$, over an observational period of length $t_{\rm obs}$ seconds, such that $s_{\rm tot} = \frac{t_{\rm obs}}{t_{\rm samp}}$. Each unique observation is therefore representable as an $n_{\rm chans} \times s_{\rm tot}$ matrix $\mathbf{M}$.

A pulsar search involves a number of procedural steps applied to the data in $\mathbf{M}$. The principal steps are similar for all searches, however the order in which these are undertaken can vary, as too can their precise implementation. In general, the first step involves radio frequency interference (RFI) excision, via the removal of channels (rows of the matrix) corresponding to known interference frequencies (Keith et al. 2010). Subsequently 'Clipping' (Hogden et al. 2012) may be applied to the data, which aims to reduce the impact of strong interference. This is achieved by setting to zero (or to the local mean) those samples which exhibit intensities higher than some pre-determined threshold in a given column in $\mathbf{M}$ (e.g. an intensity 2σ above the mean). Once these initial steps are complete, processing enters a computationally expensive phase known as de-dispersion.

Dispersion by free electrons in the ISM causes a frequency-dependent delay in radio emission as it propagates through the ISM. This delay temporally smears legitimate pulsar emission (Lorimer & Kramer 2006) reducing the S/N of their pulses. The amount of dispersive smearing a signal receives is proportional to a quantity called the dispersion measure (DM; Lorimer & Kramer 2006). This represents the free electron column density between an observer and a pulsar, integrated along the line of sight. The degree to which a signal is dispersed for an unknown pulsar cannot be known a priori (e.g. Keith et al. 2010; Levin 2012), thus several dispersion measure tests or 'DM trials' must be conducted to determine this value. This can be used to mitigate the dispersive smearing, thereby increasing the S/N of a signal (Lorimer & Kramer 2006). For a single trial, each frequency channel (row in $\mathbf{M}$) is shifted by an appropriate delay before each time bin is integrated in frequency. This produces 1 de-dispersed time series for each DM trial value.

Periodic signals in de-dispersed time series data, can be found using a Fourier analysis. This is known as a periodicity search (Lorimer & Kramer 2006). The first step after performing the FFT of a periodicity search usually involves filtering the data to remove strong spectral features known as 'birdies' (Manchester et al. 2001; Hessels et al. 2007). These may be caused by periodic or quasi-periodic interference. Summing techniques are subsequently applied, which add the amplitudes of harmonically related frequencies to their corresponding fundamentals. This step is necessary as in the Fourier domain, the power from a narrow pulse is distributed between its fundamental frequency and its harmonics (Lorimer & Kramer 2006). Thus for weaker pulsars the fundamental may not rise above the detection threshold, but the harmonic sum generally will. Periodic detections with large Fourier amplitudes post summing (above the noise background or a threshold level), are then considered to be 'suspect' periods.

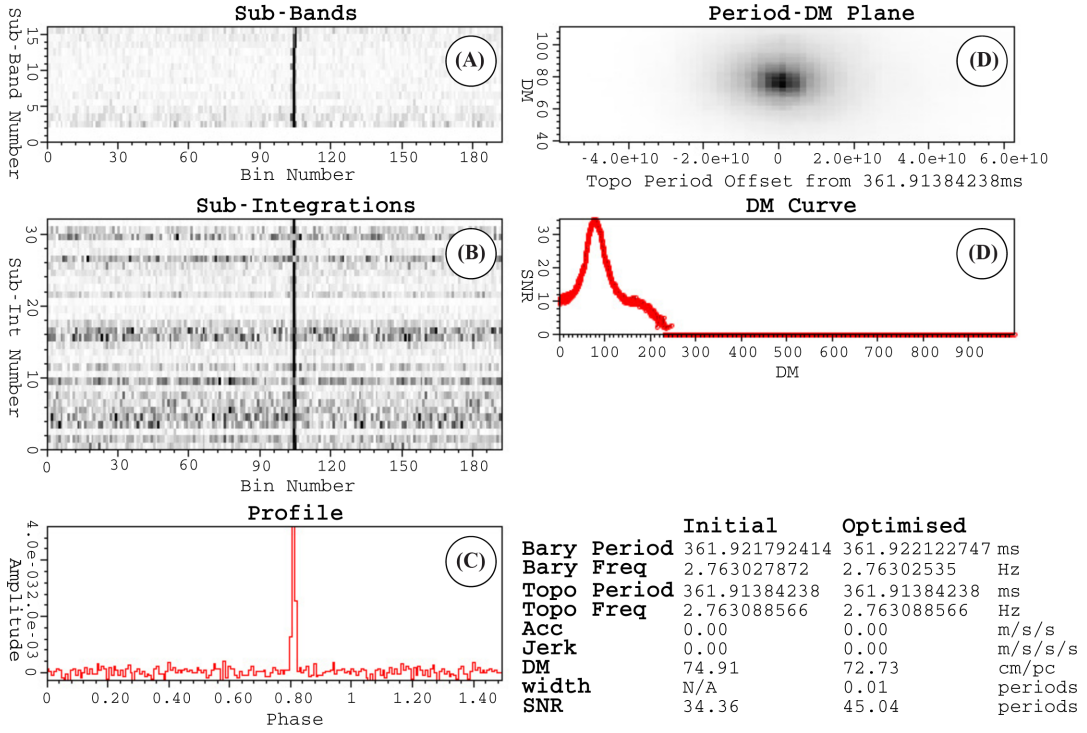

Figure 1. An annotated example candidate summarizing the detection of PSR J1706−6118. The candidate was obtained during processing of HTRU data by Thornton (2013).

A further process known as sifting (e.g. Stovall et al. 2013) is then applied to the collected suspects, which removes duplicate detections of the same signal at slightly different DMs, along with their related harmonics. A large number of suspects survive the sifting process. Diagnostic plots and summary statistics are computed for each of these remaining suspects forming candidates, which are stored for further analysis. The basic candidate consists of a small collection of characteristic variables. These include the S/N, DM, period, pulse width, and the integrated pulse profile. The latter is an array of continuous variables that describe a longitude-resolved version of the signal that has been averaged in both time and frequency. More detailed candidates also contain data describing how the signal persists throughout the time and frequency domains (Eatough et al. 2010). This can be seen in plots (A) and (B) in Fig. 1. Here persistence in frequency (A) is represented by a two-dimensional matrix showing pulse profiles integrated in time, for a set of averaged frequency channels (i.e. not full frequency resolution). Persistence through time (B), is represented by a two-dimensional matrix showing the pulse profile integrated across similarly averaged frequency channels as a function of time.

2.1 Modelling candidate numbers

Candidate numbers are anecdotally understood to be increasing steadily over time. Here we provide historical evidence supporting this view, obtained by reviewing most of the large-scale pulsar surveys conducted since the initial pulsar discovery by Hewish et al. (1968). The surveys studied are listed in Tables 2 and 3. This information has also been made available via an interactive online resource found at www.jb.man.ac.uk/pulsar/surveys.html.

Candidate numbers reported in the literature are summarized in Table 1, providing empirical evidence for rising candidate numbers. The rise is understood to be the result of expanding survey technical specifications (Stovall et al. 2013) occurring during the period depicted in Tables 2 and 3. Finer frequency resolution, longer dwell times, and acceleration searches (Eatough et al. 2013), have significantly increased the candidate yield (Lyon 2015). However, at present there is no accepted method for quantifying the effects of improving survey specifications on candidate numbers. It is therefore difficult to understand precisely how candidate numbers are changing, and what the S/N distribution of candidates should look like in practice. Such knowledge is needed if we are to design candidate selection approaches robust to error, and accurately plan survey storage requirements. Although it is difficult to capture all the steps involved in pulsar data analysis, we describe a model here that can be used as a proxy for estimating candidate numbers, linked to the number of dispersion trials undertaken per observation.

2.1.1 Approximate model of candidate numbers

Selection begins in the spectral S/N regime as described in Section 2. Here each suspect period associated with a spectral S/N, is found through a Fourier analysis of a de-dispersed time series. However, we have incomplete knowledge of the S/N distribution of spectral suspects, which arise from either (i) variations in Galactic background noise, (ii) RFI, (iii) instrument noise, or (iv) legitimate phenomena. To overcome this, we model only the most significant contributor of candidates, Gaussian distributed background noise. Empirical evidence suggests most candidates originate from background noise. Our analysis of High Time Resolution Universe Survey (HTRU) data (Thornton 2013) supports this view, held by

Table 1. Reported folded candidate numbers.

Survey	Year	Candidates	deg^{-2}
2nd Molonglo Survey(Manchester et al. 1978)	1977	2500	∼0.1
Phase II survey (Stokes et al. 1986)	1983	5405	∼1
Parkes 20 cm survey (Johnston et al. 1992)	1988	∼150 000	∼188
Parkes Southern Pulsar Survey (Manchester et al. 1996)	1991	40 000	∼2
Parkes Multibeam Pulsar Survey (Manchester et al. 2001)	1997	8000 000	∼5161
Swinburne Int. Lat. Survey (Edwards et al. 2001)	1998	>200 000	∼168[a]
Arecibo P-Alfa all configurations (Cordes et al. 2006; Lazarus 2012; P-Alfa Consortium 2015)	2004	>5000 000	∼16 361[a]
6.5 GHz Multibeam Survey (Bates et al. 2011; Bates 2011)	2006	3500 000	∼77 778 [b]
GBNCC survey (Stovall et al. 2014)	2009	>1, 200 000	∼89[a]
Southern HTRU (Keith et al. 2010)	2010	55 434 300	∼1705
Northern HTRU (Barr et al. 2013; Ng 2012)	2010	>80 000 000	∼2890[a]
LOTAAS (Cooper, private communication)	2013	39 000 000	∼2000

Notes. [a]a lower bound on the number of candidates per square degree, calculated from incomplete candidate numbers;
[b]very long integration times, with further details supplied in Tables 2 and 3.

others (Lee et al. 2013; Morello et al. 2014). It is also logically consistent, since if most candidates arose from legitimate phenomena discovery would be trivial. Whilst if most arose from RFI, this would be concerning, as telescopes used for surveys are situated in low RFI environments. It thus appears sensible to conclude that candidates are noise dominated.

By modelling candidates arising only from background noise, we can estimate the approximate number of candidates a survey will yield. To achieve this, we assume there is a 1:1 mapping from spectral suspects to folded candidates.[1] We can then model the folded S/N distribution of noise-originating candidates only, from there onwards. By assuming at least 1 folded candidate is generated per dispersion trial, which also subsequently survives sifting, it is possible to calculate indicative candidate numbers. As folded candidate S/Ns are empirically well approximated by a Gaussian distribution,[2] we can also estimate the folded S/N distribution using a simple Gaussian model. The number of candidates arising from noise with a folded S/N of $n\sigma$ (i.e. $1\sigma, \ldots, n\sigma$), is estimated as follows using a Gaussian probability density function

$$f(d, \lambda, \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} \mathrm{e}^{-\frac{1}{2}(\frac{\lambda-\mu}{\sigma})^2} \times d, \tag{1}$$

where λ is the folded S/N, μ is the mean of the noise distribution, σ its standard deviation, and d the total number of dispersion trials. This model considers each dispersion trial to be a single draw from the noise distribution. Thus candidate numbers here are determined by d, and not a top C candidate cut, as is often used to limit candidate numbers (e.g. Thornton 2013). However since cuts are used in practice to remove weak candidates (arising from noise), we will incorporate them into our model. This is achievable whilst retaining knowledge of the resulting folded S/N distribution for a cut $C \in (0, \infty]$. First, we compute the total number of candidates arising from Gaussian distributed noise, with a folded S/N $> n_\sigma$ using

$$k(d, \mu, \sigma, n_\sigma) = \int_{n_\sigma}^{\infty} f(d, \lambda, \mu, \sigma)\mathrm{d}\lambda. \tag{2}$$

In practice Gaussian noise possessing an S/N in excess of 30σ is rare. Thus we can replace the upper limit of ∞ with $n_\sigma^{\max} = 30$, beyond which a detection is almost certainly not noise. Here n_σ is the cut off S/N that is n standard deviations from the mean, and we do not count candidates with S/Ns below this. Equation (2) is related to the cumulative distribution function (CDF), of the probability distribution in equation (1), where $k = 1 - \mathrm{CDF}$ as shown in Fig. 2. From this we can compute the number of n_σ candidates surviving a top C cut, using $h(f, C - k)$. Here $C - k$ gives the number of remaining candidate places in a top C cut, and h is defined by

$$h(f, s) = \begin{cases} 0, & s \leq 0 \\ f, & f - s \leq 0 \\ s, & f - s > 0, \end{cases} \tag{3}$$

where 0 is returned if there are no spaces in a top C cut, f is returned if all n_σ candidates make the cut, and finally s is returned if some n_σ candidates miss the cut. Now the total number of candidates returned during a survey using a single telescope, with an n_{beam} receiver making p pointings, can be estimated by

$$p \times (n_{\mathrm{beams}} \times \max(k, C)), \tag{4}$$

where k is given by equation (2) and C is the numerical cut-off per beam (e.g. $C = 100$). This allows us to identify what we describe as a 'crisis' in candidate selection. Since the functions f and k are linearly dependent on d, and since we can see empirically from Tables 2 and 3 that d is increasing, this means that even if n_σ is fixed, the number of noise-originating candidates to be evaluated will increase with d. Indeed, equation (4) implies the existence of a discernible trend in candidate numbers. Much like the exponential rise in data volumes described by Bates (2011), this model shows candidate numbers to be increasing exponentially as a function of d. This is shown more clearly in Fig. 3. This illustrates how candidate numbers change as d and the number of survey pointings increase. The plot is colour coded according to total pointings, with dashed lines indicating candidate numbers when $C = 100$, and the corresponding solid lines showing candidate numbers when C is discarded. We note here that we have concentrated on utilizing d as the number of dispersion trials to show a rise in candidate numbers. This should not be seen simply as relating to the maximum DM being searched. As the sampling time is increased and more channels are used, either to preserve high time resolution at higher DMs, or as the bandwidth increases, or both, then the number of 'dispersion' trials increases. Therefore d is also a good proxy for survey sensitivity. Of course the higher the time resolution, the greater s_{tot} increases, longer observations also increase s_{tot} considerably. In both cases this increases the likelihood of detecting more than one candidate

[1] A candidate obtained by folding a de-dispersed time series at a specific suspect period.
[2] Empirically observed in HTRU survey data.

Table 2. Technical specifications of pulsar surveys conducted between 1968–1999. Here F_c (MHz) is the central observing frequency, B (MHz) is the bandwidth, Δv (kHz) is the channel width (to 3.d.p), n_{chans} indicates the number of frequency channels, t_{samp}(μs) is the sample frequency (to 3.d.p), and t_{obs}(s) the length of the observation (to 1.d.p). Values that could not be found in the literature are indicated with '?'. The omission of a survey should be treated as an oversight as opposed to a judgement on its significance.

Survey	Year	F_c (MHz)	B (MHz)	Δv (kHz)	n_{chans}	t_{samp}(μs)	t_{obs}(s)	DM trials
1st Molonglo Survey (Large et al. 1968)	1968	408	4	2000	2	5000	15	?
Search at low Galactic Lat. (Davies, Large & Pickwick 1970)	1969	408	4	?	?	50 000	819	?
Arecibo Survey 1 (Hulse & Taylor 1974)	197?	430	8	250	32	5600	198	64
Jodrell Survey A (Davies, Lyne & Seiradakis 1977)	1972	408	4	2000	2	40 000	660	?
2nd Molonglo Survey (Manchester et al. 1978)	1977	408	4	800	4	20 000	44.7	?
Green Bank Northern hemisphere Survey (Damashek et al. 1978, 1982)	1977	400	16	2000	8	16 700	144	8
Princeton-NRAO Survey (Dewey et al. 1985)	1982–83	390	16	2000	8	5556	138	8
Green Bank short-period (Stokes et al. 1985)	1983	390	8	250	32	2000	132	?
Jodrell Survey B (Clifton & Lyne 1986)	1983–84	1400	40	5000	8	2000	540	?
Arecibo survey 2 (a) – Phase II Princeton-NRAO (Stokes et al. 1986)	1983	390	8	250	32	2000	132	?
Arecibo survey 2 (b) (Stokes et al. 1986)	1984–85	430	0.96	60	16	300	39	?
Jodrell Survey C Biggs & Lyne (1992)	1985–87	610/925/928/1420	4/8/32	125/500/1000	32	300	79	39
Parkes Globular Cluster Survey (20 cm) (Manchester et al. 1990a,b)[a]	1989–90	1491	80/320	1000/5000	80/64	300	3000	100
Parkes Globular Cluster Survey (50 cm) (Manchester et al. 1990a,b)[a]	1988–90	640	32	250	128	300	3000/4500	100
Arecibo Survey 3 (Nice, Fruchter & Taylor 1995)	198?	430	10	78.125	128	516.625	67.7	256
Parkes 20-cm Survey (I) (Johnston et al. 1992)	1988	1434	800	1000	80	300	78.6	100
Parkes 20-cm Survey (II) (Johnston et al. 1992)	1988	1520	320	5000	64	1200	157.3	100
Arecibo 430 MHz Intermediate Galactic Latitude Survey (Navarro et al. 2003)	1989–91	430	10	78.125	128	506.625	66.4	163
High Galactic Latitude Pulsar Survey of the Arecibo Sky (H1) (Foster et al. 1995)	1990	430	10	250	128	506	40	64
High Galactic Latitude Pulsar Survey of the Arecibo Sky (H2) (Foster et al. 1995)	1991	430	8	250	32	250	40	64
High Galactic Latitude Pulsar Survey of the Arecibo Sky (H3) (Foster et al. 1995)	1992	430	8	250	32	250	40	64
High Galactic Latitude Pulsar Survey of the Arecibo Sky (H4) (Foster et al. 1995)	1993	430	8	250	32	250	40	64
High Galactic Latitude Pulsar Survey of the Arecibo Sky (H5) (Foster et al. 1995)	1994–95	430	8	250	32	250	40	64
Arecibo Survey 4 Phase I (Nice, Taylor & Fruchter 1993)	1991	430	10	78.125	128	516.625	67.7	?
Arecibo Survey 4 Phase II (Camilo, Nice & Taylor 1993)	1992	429	8	250	64	250	40	192
Parkes Southern (Manchester et al. 1996)	1991–93	436	32	1250	256	300	157.3	738
Green Bank fast pulsar survey (Sayer, Nice & Taylor 1997)	1994–96	370	40	78.125	512	256	134	512
PMPS (Manchester et al. 2001)	1997	1374	288	3000	96	250	2100	325
Swinburne Int. Lat. survey (Edwards et al. 2001)	1998–99	1374	288	3000	96	125	265	375

Note. [a]more than one configuration used during the survey.

per DM trial. For simplicity this is not modelled here, and so what we present can be considered a lower limit.

There are two strategies available for dealing with the implied rise of noisy candidates. The first is to increase the lower S/N limit n_σ in equation (2). This effectively implements an S/N cut-off, used by many to filter in the spectral domain (Foster et al. 1995; Hessels et al. 2007; Burgay et al. 2013; Thornton 2013), and the folded domain (Damashek, Taylor & Hulse 1978; Manchester et al. 1978; Stokes et al. 1986; Manchester et al. 2001; Burgay et al. 2013). However in practice this cut-off would become high enough to reject weaker detections of interest (i.e. weaker pulsars, see Section 4.2.1) if it is to reduce candidate numbers. The second option is to impose a smaller constant cut-off C to the candidates collected per observation or beam, also done by many (Edwards et al. 2001; Jacoby et al. 2009; Bates et al. 2012; Thornton 2013) and accounted for in our model. Fig. 2 shows these two methods to be fundamentally the same. Imposing a fixed limit C on the output of equation (2), can only be achieved by increasing the lower value of n_σ in the integral, since the integrand is fixed by equation (1). This corresponds to setting a high S/N cut-off. Using either of these approaches impacts our ability to detect legitimate pulsar signals. This is particularly true of a top C cut, as it would appear that noise alone can fill up a top C cut, without even taking into consideration the influence of RFI, or legitimate phenomena. Taking d to the limit increases the certainty that noise will dominate a candidate cut, and reduces the likelihood of weak legitimate signals making it through to analysis. We now turn our attention to determining how to deal with these issues.

Table 3. Technical specifications of pulsar surveys conducted between 2000 and present, and projected specifications for instruments under development. X-ray pulsar searches undertaken during this period (Abdo et al. 2009; Ransom et al. 2011) are omitted. Here F_c (MHz) is the central observing frequency, B (MHz) is the bandwidth, Δv (kHz) is the channel width (to 3.d.p), n_{chans} indicates the number of frequency channels, $t_{samp}(\mu s)$ is the sample frequency (to 3.d.p), and $t_{obs}(s)$ the length of the observation (to 1.d.p). Values that could not be found in the literature are indicated with '?'. The omission of a survey should be treated as an oversight as opposed to a judgement on its significance.

Survey	Year	F_c (MHz)	B (MHz)	Δv (kHz)	n_{chans}	$t_{samp}(\mu s)$	$t_{obs}(s)$	DM trials
Parkes high-lat multibeam (Burgay et al. 2006)	2000–03	1374	288	3000	96	125	265	?
Survey of the Magellanic Clouds (Manchester et al. 2006)	2000–01	1374	288	3000	96	1000	8400	228
1.4 GHz Arecibo Survey (DM < 100) (Hessels et al. 2007)	2001–02	1175	100	390.625	256	64	7200	?
1.4 GHz Arecibo Survey (DM > 100) (Hessels et al. 2007)	2001–02	1475	100	195.313	512	128	7200	?
Large Area Survey for Radio Pulsars (Jacoby et al. 2009)	2001–02	1374	288	3000	96	125	256	375
EGRET 56 Pulsar survey (Crawford et al. 2006)	2002–03	1374	288	3000	96	125	2100	150
EGRET error box survey (Champion, McLaughlin & Lorimer 2005)	2003	327	25	48.828	512	125	260	392
A0327 Pilot (Deneva et al. 2013)	2003	327	25	48.828	512	256	60	6358
The Perseus Arm Pulsar Survey (Burgay et al. 2013)[a]	2004–09	1374	288	3000	96	125	2100	183/325
The 8gr8 Cygnus Survey (Rubio-Herrera et al. 2007; Janssen et al. 2009)	2004	328	10	19.531	512	819.2	6872	488
Parkes deep northern Galactic Plane (Lorimer, Camilo & McLaughlin 2013)	2004–05	1374	288	3000	96	125	4200	496
P-ALFA Survey (intial) (WAPP) (Cordes et al. 2006; Deneva et al. 2009)	2004	1420	100	390.625	256	64	134	96
P-ALFA Survey (anticipated) (WAPP) (Cordes et al. 2006; Deneva et al. 2009)[a]	2004–10	1420	300	390.625	1024	64	134	96/1272
6.5 Ghz Multibeam Pulsar Survey (Bates et al. 2011)	2006–07	6591	576	3000	192	125	1055	286
Green Bank 350 MHz Drift Scan (Boyles et al. 2013)	2007	350	50	24.414	2048	81.92	140	?
GBT350 (Spigot) (Deneva et al. 2013)	2007	350	50	24.414	2048	82	140	?
P-ALFA Survey (MOCK) (Spitler et al. 2014; Deneva et al. 2009; Lazarus 2012)[a]	2009–14	1375	322.6	336.042	960	65.5	120/300	5016
GBNCC (GUPPI) (Deneva et al. 2013; Stovall et al. 2014)[a]	2009–14	350	100	24.414	4096	82	120	17 352/26 532
Southern HTRU (LOW) (Keith et al. 2010)	2010–12	1352	340	390.625	870	64	4300	?
Southern HTRU (MED) (Keith et al. 2010)	2010–12	1352	340	390.625	870	64	540	1436
Southern HTRU (HIGH) (Keith et al. 2010)	2010–12	1352	340	390.625	870	64	270	8000
A0327 (MOCK) (Deneva et al. 2013)	2010	327	57	55.664	1024	125	60	6358
LPPS (Coenen et al. 2014)	2010	142	6.8	12.143	560	655	3420	3487
LOTAS (Coenen et al. 2014)[a]	2010–11	135	48	12.295	3904	1300	1020	16 845/18 100
Northern HTRU (LOW) (Barr et al. 2013; Ng 2012)[a]	2010–14	1360	240	585.9	410	54.61	1500	406/3240
Northern HTRU (MED) (Barr et al. 2013; Ng 2012)[a]	2010–14	1360	240	585.9	410	54.61	180	406/3240
Northern HTRU (HIGH) (Barr et al. 2013; Ng 2012)[a]	2010–14	1360	240	585.9	410	54.61	90	406/3240
SPAN512 (Desvignes et al. 2012)	2012	1486	512	500	1024	64	1080	?
LOTAAS (Lofar Working Group 2013; Cooper 2014)	2013	135	95	12.207	2592	491.52	3600	7000
A0327 (PUPPI) (Deneva et al. 2013)	2014	327	69	24.503	2816	82	60	6358
SUPERB (Barr 2014; Keane et al., in preparation)	2014	1374	340	332.031	1024	32	540	1448
GMRT High Resolution Southern Sky Survey (MID) (Bhattachatyya 2014; Bhattachatyya et al. 2016)	2014	322	32	15.625	2048	60	1200	6000
GMRT High Resolution Southern Sky Survey (HIGH) (Bhattachatyya 2014; Bhattachatyya et al. 2016)	2014	322	32	31.25	1024	30	720	6000
FAST[b] (Smits et al. 2009b)	2016	1315	400	42.105	9500	100	600	?
SKA[b] (Configuration A) (Smits et al. 2009a)	2020–22	1250	500	50	9500	64	1800	?
SKA[b] (Configuration B) (Smits et al. 2009a)	2020–22	650	300	50	9500	64	1800	?

Note. [a]more than one configuration used during the survey.
[b]Projected future survey with configuration specifics subject to change.

3 CANDIDATE SELECTION METHODS

3.1 Manual selection

During the earliest surveys, manual selection involved the inspection of analogue pen chart records for periodic signals (Large, Vaughan & Wielebinski 1968; Manchester et al. 1978). This process was subsequently replaced by digital data inspection, with the adoption of early computer systems. From then on, manual selection involved the inspection 'by eye' of digitally produced diagnostic plots describing each candidate. Those found exhibiting pulsar-like characteristics were recorded for analysis, whilst the remainder were ignored (though retained on disc for possible re-analysis).

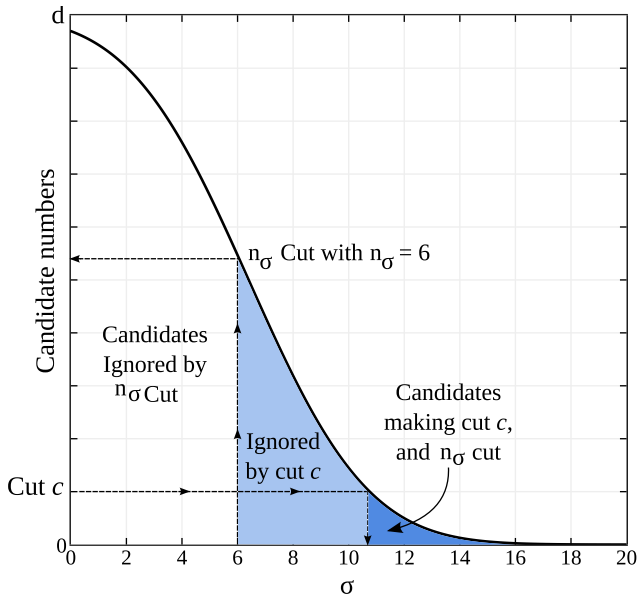

Figure 2. Diagram of 1 − CDF of equation (1), showing the relationship between $n\sigma$ and constant cuts. This illustrates their impact on the number of noise candidates making it through to the candidate selection stage.

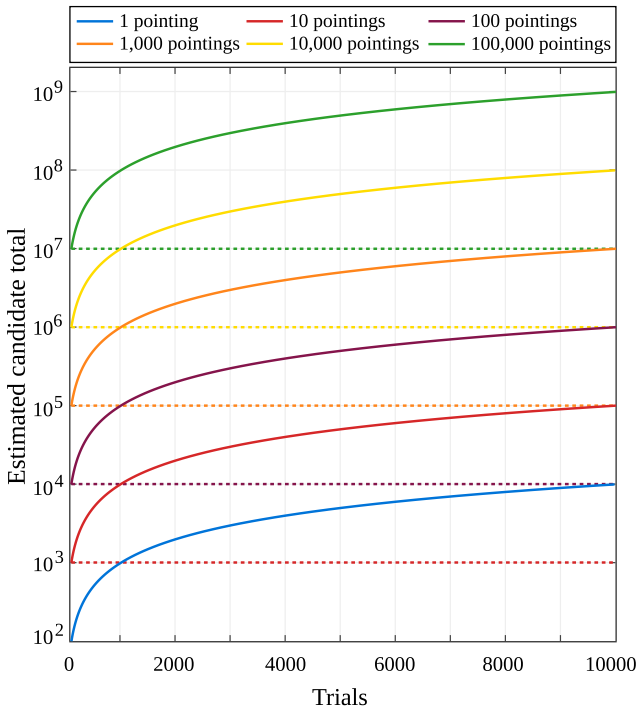

Figure 3. Candidate numbers predicted by equation (4) (using $n_\sigma = 7$ and $n_\sigma^{\max} = 100$), varied according to the total number of survey pointings for a single beam receiver. Coloured dashed lines indicate the total number of candidates returned when using a conservative $C = 100$ cut. The corresponding solid colour lines indicate the total number of candidates returned when the cut is discarded. The solid lines are truncated such that they begin where $C = 100$ to avoid overlapping lines complicating the plot.

During the initial period of digitization, pulsar surveys produced very few candidates with respect to modern searches. The second Molonglo survey conducted during the 1970s, produced only 2500 candidates in total (Manchester et al. 1978). These yielded 224 pulsar detections (Manchester et al. 2005), a hit rate of almost 9 per cent.[3] Thus during this period manual selection was entirely practical. Soon after however, increasing candidate numbers began to cause problems. The first mention of this within the literature (to the best of our knowledge) was made by Clifton & Lyne (1986) regarding Jodrell Survey B. The number of candidates produced during this survey necessitated extensive manual selection on the basis of pulse profile appearance and S/N. Although such heuristic judgements were not new, their explicit mention with respect to candidate selection indicated that a shift in procedure had occurred. Whereas before it was possible to evaluate most, if not all candidates by eye, here it became necessary to expedite the process using heuristics. Contemporary surveys reacting to similar issues imposed high S/N cut-offs to limit candidate numbers directly. The Arecibo Phase II survey used an 8σ S/N cut, thus only ∼5405 candidates required manual inspection (Stokes et al. 1986).

The use of heuristics and S/N cuts proved insufficient to deal with candidate number problems. Additional processing steps such as improved sifting were applied in response, and these became increasingly important during this period. However as these measures apply high up the processing pipeline (close to the final data products), their capacity to reduce candidate numbers was limited. Consequently attempts were made to automatically remove spurious candidates lower down the pipeline, with the aim of preventing them ever reaching human eyes. During the Parkes 20-cm survey, two software tools were devised by Johnston et al. (1992) to achieve this. Together these encapsulated and optimized the general search procedure discussed in Section 2. The software ('MSPFIND' and another unnamed tool) was explicitly designed to reduce the quantity of spurious candidates, while maintaining sensitivity to millisecond pulsars (MSPs). Only candidates with an S/N >8 were allowed through the pipeline to manual inspection. It is unclear how many candidates required manual inspection, though the number was less than 150 000 (Johnston et al. 1992). During the same period, a similar software tool known as the Caltech Pulsar Package (Deich 1994), was developed for the Arecibo 430 MHz Intermediate Galactic Latitude Survey (Navarro et al. 2003). These represent some of the earliest efforts to systematise the search process in a reproducible way.

3.2 Summary interfaces

The success achieved via low-level filtering and sifting, continued to be undermined by ever-increasing candidate numbers brought about by technological advances. By the late 1990s, manual selection was therefore becoming increasingly infeasible. This spawned many graphical tools, designed to summarize and filter candidates for speedy and concise evaluation. The first of these, RUNVIEW (Burgay et al. 2006), was created to analyse data output by the Parkes Multibeam Survey (PMPS; Manchester et al. 2001). During the Swinburne Intermediate-latitude survey, Edwards et al. (2001) devised a similar graphical tool that included distributional information of candidate parameters. A later reprocessing of PMPS data

[3] The hit rate of the recent southern HTRU medium latitude search was much lower, at around 0.01 per cent (Lyon 2015).

for binary and MSPs, spawned the development of a more sophisticated graphical tool for candidate viewing called REAPER. REAPER used a dynamic customizable plot (Faulkner et al. 2004) that enabled heuristic judgements of candidate origin to be made using multiple variables. The use of REAPER led to the discovery of 128 unidentified pulsars in PMPS data. This corresponds to ∼15.4 per cent of the known pulsars in PMPS data, given that 833 have now been identified (Lorimer et al. 2015).

Following the success of REAPER, an updated version of the tool called JREAPER was developed by Keith et al. (2009). It incorporated algorithms which assigned numerical scores to candidates based on their parameters, permitting candidate rankings. By ignoring those candidates achieving low rankings, the amount of visual inspection required was reduced. When applied to data gathered during the PMPS, use of JREAPER led to the discovery of a further 28 new pulsars (Keith et al. 2009), corresponding to ∼3.4 per cent of known PMPS pulsars. Thus by 2009, summary interfaces had helped find ∼18.7 per cent of all PMPS pulsars illustrating the usefulness of graphical approaches. More recently, web-based candidate viewing systems incorporating similar scoring mechanisms have appeared (Cordes et al. 2006; Deneva et al. 2009, 2013). One such tool, The Pulsar Search Collaboratory (Rosen et al. 2010),[4] also incorporates human scoring via the input of high school students. Students taking part in the programme have discovered several new pulsars (Rosen et al. 2013). This includes PSR J1930−1852, a pulsar in a double neutron star system (Swiggum et al. 2015).

3.3 Semi-automated ranking approaches

Semi-automated selection approaches have recently begun to emerge. Amongst the most popular are those employing ranking mechanisms to prioritize promising candidates for human attention. The most notable of these is the PEACE system developed by Lee et al. (2013). PEACE describes each candidate via six numerical features, combined linearly to form a candidate score. Ranked candidates are then analysed via graphical viewing tools by students in the Arecibo Remote Command Centre Programme. To date PEACE has been used during the Greenbank Northern Celestial Cap Survey (GBNCC; Stovall et al. 2014) and the Northern High Time Resolution Universe Survey (HTRU north, Ng 2012; Barr et al. 2013). Periodic and single-pulse candidates obtained during the A0327 survey (Deneva et al. 2013), were similarly ranked using an algorithm based on PEACE. Over 50 participants (of varying expertise) from four universities, were then invited to view the A0327 candidates via a web-based interface.

3.4 Automated 'Intelligent' selection

Intelligent selection techniques are gaining widespread adoption. The nature of the intelligence arises from the domain of statistical learning theory, more generally known as ML. In particular, from a branch of ML known as *statistical classification*. The aim of classification is to build functions that accurately map a set of input data points, to a set of class labels. For pulsar search this means mapping each candidate to its correct label (pulsar or non-pulsar). This is known as candidate *classification*, a form of supervised learning (Mitchell 1997; Duda, Hart & Stork 2000; Bishop 2006). If $S = \{X_1, \ldots, X_n\}$ represents the set of all candidate data, then X_i is an individual candidate represented by variables known as *features*. Features describe the characteristics of the candidate such that $X_i = \{X_i^j, \ldots, X_i^m\}$, where each feature $X_i^j \in \mathbb{R}$ for $j = 1, \ldots, m$. The label y associated with each candidate, may have multiple possible values such that $y \in Y = \{y_1, \ldots, y_k\}$ (e.g. MSP, RFI, noise etc.). However since the goal here is to separate pulsar and non-pulsar candidates, we consider the binary labels $y \in Y = \{-1, 1\}$, where $y_1 = -1$ equates to non-pulsar (synonymous with negative) and $y_2 = 1$ to pulsar (synonymous with positive).

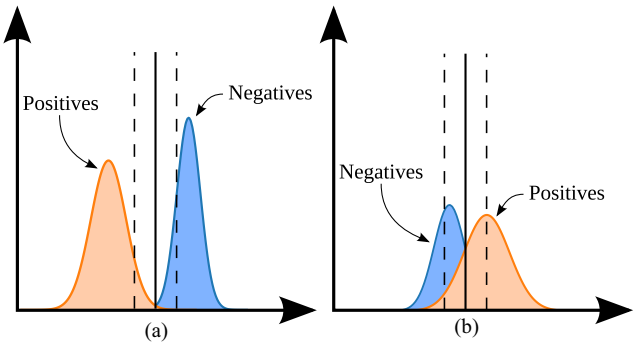

Figure 4. Example of the varying separability of features from highly separable in (a), to poorly separable in (b).

To build accurate classification systems, it is desirable to utilize features that separate the classes under consideration. This is illustrated in Fig. 4. An ML function 'learns' to separate candidates described using features, from a labelled input vector known as the training set T. It contains pairs such that $T = \{(X_1, y_1), \ldots, (X_n, y_n)\}$. The goal of classification is to induce a mapping function between candidates and labels based on the data in T, that minimizes generalization error on test examples (Kohavi & John 1997). The derived function can then be used to label new unseen candidates.

The first application of ML approaches to candidate selection was accomplished by Eatough et al. (2010). In this work each candidate was reduced to a set of 12 numerical feature values inspired by the scoring system first adopted in JREAPER. A predictive model based on a multilayered perceptron (MLP), a form of artificial neural network (Haykin 1999; Bishop 2006), was then constructed. Using this model, a re-analysis of a sample of PMPS data was completed and a new pulsar discovered (Eatough 2009). Neural network classifiers based on the MLP architecture were also developed to run on data gathered during the HTRU survey. Bates et al. (2012) modified the earlier approach by describing candidates using 10 further numerical features (22 in total). The same features were used to train neural network classifiers applied to HTRU medium latitude data by Thornton (2013). More recently the SPINN system developed by Morello et al. (2014), utilized developments from the field of computer science to optimize neural network performance on a set of six features. SPINN is currently being applied as part of the Survey for Pulsars and Extragalactic Radio Bursts (SUPERB; Barr 2014; Keane et al., in preparation).

Convolutional neural networks (CNN; Bengio 2009), which achieved prominence due to their high accuracy on difficult learning problems such as speech and image recognition, have been adapted for candidate selection. The Pulsar Image-based Classification System (PICS) developed by Zhu et al. (2014), uses the CNN and other types of ML classifier to perform image classification on candidate plots. PICS is technically the most sophisticated approach available, and it appears to possess high accuracy. However this comes at the expense of high computational costs. Particularly with respect to runtime complexity.

[4] http://pulsarsearchcollaboratory.com

4 DISCUSSION

4.1 Critique of manual selection

Manual selection has retained a vital role in pulsar search (Keith et al. 2010), as demonstrated by its use during recent surveys (Bates et al. 2011; Boyles et al. 2013; Coenen et al. 2014). The strongest argument in favour of manual selection is its presumed accuracy, i.e. by Eatough (2009) and Morello et al. (2014). However, to the best of our knowledge, no study of the accuracy of expert selection has been conducted. Although intuitively one would expect manual accuracy to be high, studies in other domains indicate otherwise. Most famously studies in medicine and finance (Meehl 1954; Barber & Odean 2000) suggest that expert decision making is flawed due to unconscious biases. Indeed manual selection is already known to be a subjective and error prone process (Eatough 2009; Eatough et al. 2010). In any case, it is infeasible to continue using manual approaches given the rise in candidate numbers predicted in Section 2.1, also anticipated by others (Keane et al. 2014). Thus irrespective of the true accuracy of manual selection, it must be supplanted to keep pace with increasing data capture rates and candidate numbers.

4.2 Critique of automated approaches

ML approaches are becoming increasingly important for automating decision making processes in finance (Chandola, Banerjee & Kumar 2009), medicine (Markou & Singh 2003; Chandola et al. 2009), safety critical systems (Markou & Singh 2003; Hodge & Austin 2004; Chandola et al. 2009) and astronomy (Ball & Brunner 2009; Borne. 2009; Way et al. 2012). Given the widespread adoption of ML, the continued application of manual selection raises a fundamental question: why has a transition to completely automated selection not yet occurred? Specific barriers to adoption may be responsible, such as the expertise required to implement and use ML methods effectively. Where this barrier is overcome, approaches emerge that are typically survey and search specific.

A further problem is the limited public availability of pulsar specific code and data. Thus to adopt ML approaches new systems generally need to be built from scratch. ML approaches also have to be 'trained' upon data acquired by the same pipeline they will be deployed upon[5]. If training data are not shared, it has to be collected before a survey begins. The cost of doing so may be a further barrier to adoption. Perhaps more simply, existing automated approaches may not yet be accurate enough to be trusted completely. If this is the case, it is unlikely to be caused by the choice of ML system (e.g. neural network, probabilistic classifier, or any other). Those methods described in Section 3.4 employ well-studied ML techniques, proven to be effective for a variety of problems. Drops in performance are more likely to be due to deficiencies in (i) the features describing candidates, and (ii) the data used to train learning algorithms. In the following section, we present evidence suggesting that existing candidate features may well be sub-optimal.

4.2.1 *Sub-optimal candidate features*

Candidate features can be categorized as being either fundamental to, or as being derived from candidate data. The latter derive new information on the assumption that it will possess some utility, whilst the former do not. For instance the S/N or period of a candidate, can be considered fundamental. A good example of a derived feature is the χ^2 value of a *sine* curve fit to the pulse profile as used by Bates et al. (2012). Using curve fittings in this manner expresses an underlying hypothesis. In this case Bates et al. (2012) suppose a good χ^2 fit to be indicative of sinusoidal RFI. Whilst the reasoning is sound, such a feature represents an untested hypothesis which may or may not hold true.

The majority of existing features are derived (see Eatough et al. 2010; Bates et al. 2012; Thornton 2013; Morello et al. 2014), and are based upon the heuristics used when selecting candidates manually. As manual selection is imperfect, we cannot rule out the possibility of having designed features, and thereby automated methods, which make the same mistakes as ourselves. Some features in use have been found to introduce unwanted and unexpected biases against particular types of pulsar candidate (Bates et al. 2012; Morello et al. 2014). Fundamental features are not necessarily better. For example the folded or spectral S/N, is often used as a primitive filter and as a feature for learning. As noise candidates possessing folded S/Ns of 6σ are common (Nice et al. 1995), using an S/N cut at this level allows large numbers of likely noise-originating candidates to be rejected. However as noted by Bates et al. (2012), such cuts are helpful only if one assumes all low-S/N candidates are attributable to noise. In practice, the application of cuts has prevented the detection of weaker pulsar signals as warned in Section 2.1. PSR J0812−3910 went unseen in High Latitude survey data (Burgay et al. 2006), as its spectral S/N was below the survey's threshold for folding. Similarly PSR J0818−3049 went undetected during the same survey, as its folded S/N was below the cut applied prior to manual selection. What is more, there is no agreed upon S/N cut level for any stage in the search pipeline. Domain experience usually plays a role in determining the level, but this is often not specified and difficult to quantify. Levels used include 6σ (Damashek et al. 1978; Thornton 2013), 6.3σ (Manchester et al. 1978), 7σ (Foster et al. 1995; Hessels et al. 2007), 7.5σ (Manchester et al. 1996), 8σ (Stokes et al. 1986; Johnston et al. 1992; Edwards et al. 2001; Manchester et al. 2001; Burgay et al. 2006, 2013), 8.5σ (Nice et al. 1995), 9σ (Jacoby et al. 2009; Bates et al. 2011), and finally 9.5σ (Jacoby et al. 2009).

A further problem with many existing features is that they are implementation dependent. They are described using concepts that can be expressed in various ways mathematically (S/N used by Bates et al. 2011; Thornton 2013; Lee et al. 2013; Morello et al. 2014), are subject to interpretation without precise definition (pulse width used by Bates et al. 2011; Lee et al. 2013; Thornton 2013; Morello et al. 2014), or implicitly use external algorithms which go undefined (e.g. curve fitting employed by Bates et al. 2011; Thornton 2013). It is therefore difficult to build upon the work of others, as features and reported results are not reproducible. Thus direct comparisons between features are rare (Morello et al. 2014) and impractical.

4.2.2 *Feature evaluation issues*

The techniques most often used to evaluate features are inadequate for determining how well they separate pulsar and non-pulsar candidates. The most common form of evaluation is undertaken in two steps. The first determines the presence of linear correlations between features and class labels (Bates et al. 2011), the second compares the performance of different classifiers built using the features (Bates et al. 2011; Lee et al. 2013; Morello et al.

[5] The data an algorithm 'learns' from must possess the same distribution as the data it will be applied to, otherwise its performance will be poor.

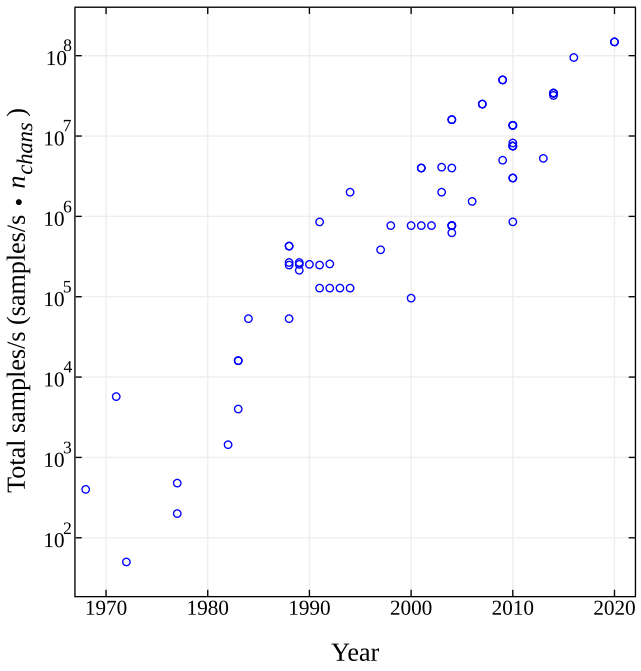

Figure 5. Scatter plot showing the total number of samples per second recorded by all pulsar surveys listed in Tables 2 and 3, as a function of time.

2014) – the standard 'wrapper' method (Kohavi & John 1997; Guyon & Elisseeff 2003). This two-step evaluation considers strong linear correlations and accurate classification performance, characteristic of 'good' feature sets. However this fails to consider the presence of useful non-linear correlations in the data. Finally using classifier outputs to assess feature performance is known to give misleading results (Brown et al. 2012), as performance will vary according to the classifier used.

In order to build robust shareable features tolerant to bias, it is necessary to adopt standard procedures that facilitate reproducibility and independent evaluation within the pulsar search community. Morello et al. (2014) began this process via the sharing of a fully labelled data set, and by providing a clear set of design principles used when creating their features. Here we make similar recommendations, closely followed when designing and evaluating the new feature set described in Section 5. It is recommended that features,

(i) minimize biases and selection effects (Morello et al. 2014),
(ii) be survey independent for data interoperability,
(iii) be implementation independent, with concise mathematical definitions allowing for reproducibility,
(iv) be evaluated using a statistical framework that enables comparison and reproducibility,
(v) guard against high dimensionality (Morello et al. 2014),
(vi) be accompanied by public feature generation code, to facilitate co-operation and feature improvement,
(vii) be supplied in a standard data format,
(viii) be evaluated on multiple data sets to ensure robustness.

4.3 Future processing challenges

The number of samples per second recorded by pulsar surveys has been increasing steadily over time, as shown in Fig. 5. This measure serves as a useful proxy for estimating raw data throughput per second

$$\text{bits/s} = \left(\frac{10^6}{t_{\text{samp}}}\right) \times n_{\text{chans}} \times n_{\text{pol}} \times n_{\text{beams}} \times n_{\text{bits}}, \quad (5)$$

where n_{pol} is the number of polarizations, n_{bits} the number of bits used to store an individual sample, and t_{samp} the sampling rate expressed in microseconds. Finer frequency resolution, faster sampling rates and longer observations, increase the data capture rate and thereby the total volume of data generated during a survey. These have been increasing over time as shown in Tables 2 and 3, a trend likely to continue (Smits et al. 2009a,b; Keane et al. 2014). If it does continue, it will become infeasible to store all raw observational data permanently. It will similarly become impractical to store all candidate data. This is perhaps best illustrated via an example SKA scenario. Suppose for a single observation there are 1500 beams and 4000 DM trials. If just one candidate is above the S/N selection threshold per DM-acceleration combination, this leads to 4000 candidates produced per beam and 6×10^6 per observation. If each candidate is 50 kB in size[6] then 0.3 TB of candidate data will be generated per observation. For a hypothetical survey lasting 50 d, where there are 120 observations per day, this equates to 3.6×10^{10} individual candidates, and 1.8 PB of candidate data alone (raw data storage requirements are much greater). In the absence of sufficient archiving capacity, here it becomes important to find and prioritize candidates of scientific value for storage. To achieve this, the processing of observational data will have to be done in real time, from candidate generation to candidate selection. Given the real-time constraint it is impossible to incorporate human decision making into the candidate selection process, thus automated approaches will have to be trusted to accurately determine which data to retain, and which to discard. This will need to be done at high levels of data throughput, with a strict execution time constraint (i.e. before more data arrives). The ML methods currently used for candidate filtering as described in Section 3, are not optimized for real-time operation. Rather they are designed for high accuracy, and as such their learning models are not designed to be resource efficient. Their memory and runtime requirements typically grow linearly with the number of candidates observed, whilst quadratic growth or worse is also common. In environments with high data rates, these filters can quickly become processing bottlenecks as their runtime increases. Increasing data rates therefore present two distinct problems for candidate selection: they make it implausible to store all observational data reducing the feasibility of offline analysis, and restrict our use of candidate selection approaches to those that can operate within strict real-time constraints.

The shift to online processing has already occurred in other domains in response to similar data pressures (ATLAS Collaboration 2008). Indeed closer to home, some pulsar/fast transient searches are already being undertaken with real-time processing pipelines (Thompson et al. 2011; Ait-Allal et al. 2012; Barr 2014; van Heerden et al. 2014). Real-time searches for fast radio bursts (Lorimer et al. 2007; Keane et al. 2012; Thornton et al. 2013) are also becoming increasingly common (Karastergiou et al. 2015; Law et al. 2015; Petroff et al. 2015). These concerns are returned to in Section 6.

[6] Existing surveys already produce candidates larger than this (Barr 2014; Cooper 2014; Bhattachatyya et al. 2016).

Table 4. The eight features derived from the integrated pulse profile $P = \{p_1, \ldots, p_n\}$, and the DM-SNR curve $D = \{d_1, \ldots, d_n\}$. For both P and D, all p_i and $d_i \in \mathbb{N}$ for $i = 1, \ldots, n$.

Feature	Description	Definition
Prof.μ	Mean of the integrated profile P.	$\frac{1}{n}\sum_{i=1}^{n} p_i$
Prof.σ	Standard deviation of the integrated profile P.	$\sqrt{\frac{\sum_{i=1}^{n}(p_i - \bar{P})^2}{n-1}}$
Prof.k	Excess kurtosis of the integrated profile P.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(p_i - \bar{P})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(p_i - \bar{P})^2))^2} - 3$
Prof.s	Skewness of the integrated profile P.	$\frac{\frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{P})^3}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - \bar{P})^2}\right)^3}$
DM$_\mu$	Mean of the DM-SNR curve D.	$\frac{1}{n}\sum_{i=1}^{n} d_i$
DM$_\sigma$	Standard deviation of the DM-SNR curve D.	$\sqrt{\frac{\sum_{i=1}^{n}(d_i - \bar{D})^2}{n-1}}$
DM$_k$	Excess kurtosis of the DM-SNR curve D.	$\frac{\frac{1}{n}(\sum_{i=1}^{n}(d_i - \bar{D})^4)}{(\frac{1}{n}(\sum_{i=1}^{n}(d_i - \bar{D})^2))^2} - 3$
DM$_s$	Skewness of the DM-SNR curve D.	$\frac{\frac{1}{n}\sum_{i=1}^{n}(d_i - \bar{D})^3}{\left(\sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \bar{D})^2}\right)^3}$

5 NEW CANDIDATE FEATURES

The model introduced in Section 2.1 implies that candidate numbers are rising exponentially, and increasingly dominated by noise. We aim to address these problems by finding candidate features that maximize the separation between noise and non-noise candidates, reducing the impact of the largest contributor to high candidate numbers. We also seek to minimize the number of features we use, so as to avoid the problems associated with the 'curse of dimensionality' (Hughes 1968), which reduces classification performance. In total, we extracted eight new features for this purpose from two components of the typical pulsar candidate following the recommendations of Section 4.2.1. These features are defined in full in Table 4.

The first four are simple statistics obtained from the integrated pulse profile (folded profile). The remaining four similarly obtained from the DM-SNR curve shown in plot (E) in Fig. 1. These features are fundamental to the data, are dissociated with any specific hypothesis, and are few in number. Likewise they possess no intrinsic biases, except perhaps resolution, with respect to the number of profile/DM curve bins used to describe a candidate. The chosen features are also survey/implementation-independent, provided integrated profile and DM-SNR curve data have the same numerical range, and the same 'natural' DM window[7] for candidates output by different surveys.

'This is defined as the range of DMs around the DM that gives the highest spectral detection significance for the candidate. The limits of this range are defined by the change in DM that corresponds to a time delay across the frequency band equivalent to the candidates intial detection period'. These features were selected by returning to first principles with respect to feature design. By incorporating knowledge of the increasing trend in candidate numbers predicted in Section 2.1, potential features were evaluated according to how well they each separated noise and non-noise candidates. Starting with simple lower order statistics as possible features (mean, mode, median etc.), the ability of each to reject noise was considered statistically via a three-stage process. Higher order statistics and derived features described by Thornton (2013) were then added to the pool of possible features, and evaluated similarly. Those achieving the best separation, and the best classification results when used together with ML classifiers (see Section 6.3), were then selected for use. Thus these features were chosen with no preconceived notions of their suitability or expressiveness. Rather features were chosen on a statistical basis to avoid introducing bias.

5.1 Feature evaluation

There are three primary considerations when evaluating new features. A feature must (i) be useful for discriminating between the various classes of candidate, (ii) maximize the separation between them, and (iii) perform well in practice when used in conjunction with a classification system. Three separate evaluation procedures have therefore been applied to the features listed in Table 4. The first two forms of evaluation are presented in the section that follows, whilst classification performance is described in Section 6.3, to allow for a comparison between standard classifiers and our stream algorithm described in Section 6. As features in themselves are without meaning unless obtained from data, we first describe the data sets used during our analysis, before presenting details of the evaluation.

5.1.1 Data

Three separate data sets were used to test the discriminating capabilities of our features. These are summarized in Table 5. The first data set (HTRU 1) was produced by Morello et al. (2014). It is the

[7] This is defined as the range of DMs around the DM that yields the highest spectral detection for a candidate. The limits of this range are defined by a change in the DM that corresponds to a time delay across the frequency band equivalent to the initial detection period of a candidate.

Table 5. The pulsar candidate data sets used.

Data set	Examples	Non-pulsars	Pulsars
HTRU 1	91 192	89 995	1196
HTRU 2	17 898	16 259	1639
LOTAAS 1	5053	4987	66

first labelled[8] candidate data set made publicly available. It consists of 1196 pulsar and 89 995 non-pulsar candidates, in pulsar hunter xml files (.phcx files). These candidates were generated from a re-processing of HTRU Medium Latitude data, using the GPU-based search pipeline PEASOUP (Barr et al., in preparation). The pipeline searched for pulsar signals with DMs from 0 to 400 $cm^{-3}pc$, and also performed an acceleration search between -50 and $+50$ m s^{-2}. The HTRU 1 candidate sample possesses varied spin periods, duty cycles, and S/Ns.

In addition two further data sets were used during this work. The first (HTRU 2), is made available for analysis.[9] It comprises 1639 pulsar and 16 259 non-pulsar candidates. These were obtained during an analysis of HTRU Medium Latitude data by Thornton (2013), using a search pipeline that searched DMs between 0 and 2000 cm^{-3} pc. The pipeline produced over 11 million candidates in total. Of these 1610 pulsar and 2592 non-pulsar candidates were manually labelled by Bates et al. (2012) and Thornton (2013). These were combined with an additional 13 696 candidates, sampled uniformly from the same data set according to observational session and month. These additional candidates were manually inspected and assigned their correct labels. Together the two sets of labelled candidates form HTRU 2. It contains 725 of the known 1108 pulsars in the survey region (Levin 2012), along with re-detections and harmonics. HTRU 2 also contains noise, along with strong and weak forms of RFI. The third and final candidate data set (LOTAAS 1), was obtained during the LOTAAS survey (Lofar Working Group 2013; Cooper 2014) and is currently private. The data set consists of 66 pulsar and 4987 non-pulsar candidates. Feature data were extracted from these data sets using a new custom written PYTHON tool, the PULSAR FEATURE LAB. This tool is made available for use.[10]

5.1.2 General separability

The discriminating capabilities of the new features when applied to HTRU 1, are summarized in Fig. 6 via standard box and whisker plots. For each feature there are two distinct box plots. A coloured box plot representing the feature distribution of known pulsars, and a plain black box plot showing the feature distribution of non-pulsars. As the features have numerical ranges which differ significantly, feature data were scaled to within the range [0, 1] prior to plotting. This enables a separability comparison on the same scale. For each individual feature, the median value of the negative distribution was also subtracted. Thus the plots are centred around the non-pulsar median, allowing differences between pulsar and non-pulsar distributions to be seen more clearly.

The visualization shows there to be a reasonable amount of separation between the pulsar and non-pulsar feature distributions. This is initial evidence for the usefulness of these features[11] but only on a visual level. Thus we applied a two-tailed students t-test to feature data, in order to determine if the means of the pulsar and non-pulsar distributions were significantly different. A rejection of the null hypothesis (no significant difference) would provide statistical evidence for the separability indicated in the box plots. For all data sets, there was a statistically significant difference between the pulsar and non-pulsar distributions at $\alpha = 0.01$. A non-parametric Wilcoxon signed-rank test (Wilcoxon 1945), was also undertaken with no difference in results. This suggested the features to be worthy of further, more rigorous investigation. The next step involved determining the extent of any linear correlation between the features and the target class variable.

5.1.3 Correlation tests

The point–biserial correlation coefficient r_{pb} (Das Gupta 1960), measures the linear correlation between variables, when the target variable is dichotomous. It is equivalent to the Pearson product moment (Pearson 1895; Guyon & Elisseeff 2003), though it is better suited to candidate data, as it naturally assumes a discrete target label $y \in Y$ as described previously. The value of r_{pb} for a data sample is given by

$$r_{pb} = \frac{\bar{x}_1 - \bar{x}_2}{\sigma} \cdot \sqrt{\frac{n_2 \cdot n_1}{n \cdot (n-1)}}, \tag{6}$$

where n is the total number of samples, $\bar{x}_1$ and $\bar{x}_2$ the mean value of groups one and two, respectively, and σ the sample standard deviation. Much like Pearson's product moment, the coefficient obtains a value in the range $[-1, 1]$. A positive correlation implies that moving from group one to group two, is associated with an increase in the output variable (high values tend to co-occur with group two). A negative correlation implies that moving from group one to group two, is associated with a decrease in the output variable. Table 6 shows the correlation between the eight features and the target class variable, for the three sample data sets. The average (mean) correlation has also been computed. Since r_{pb} is non-additive, this average had to be determined using Fisher's Z transformation (Fisher 1921)

$$z = \frac{1}{2} \ln \left(\frac{1 + r_{pb}}{1 - r_{pb}} \right). \tag{7}$$

Using equation (7) the corresponding correlations of each feature on the three data sets were transformed into additive z values, summed, and the mean obtained. The mean z value was then transformed back into a meaningful correlation using the inverse of the Fisher-Z,

$$r_{pb} = \frac{e^{2 \cdot z} - 1}{e^{2 \cdot z} + 1}. \tag{8}$$

The data in Table 6 shows there to be three features that on average, exhibit strong correlations ($>|0.5|$). These include the mean, excess kurtosis, and skew of the integrated pulse profile. All features exhibit a weak correlation on at least one data set, which is stronger on others. The lowest correlation witnessed on HTRU 1, between the standard deviation of the DM-SNR curve and the target variable, performed much better on HTRU 2. This is probably due to differences between the DM ranges of candidates in each data set (0–400 $cm^{-3}pc$ for HTRU 1 and 0–2000 $cm^{-3}pc$ for HTRU 2). Irrespective of this no features are completely uncorrelated. Whilst there is variation in the effective linear separability of features across all data sets, it is surprising that such simple measures possess discriminatory ability at all. However, caution must be used when judging features based upon their linear correlations. Those features which possess linear correlations close to zero, may possess

[8] Containing correctly labelled pulsar and non-pulsar candidates.
[9] https://dx.doi.org/10.6084/m9.figshare.3080389.v1
[10] http://dx.doi.org/10.6084/m9.figshare.1536472
[11] Similar levels of separability were observed when the same plot was produced for both the HTRU 2 and LOTAAS 1 data sets.

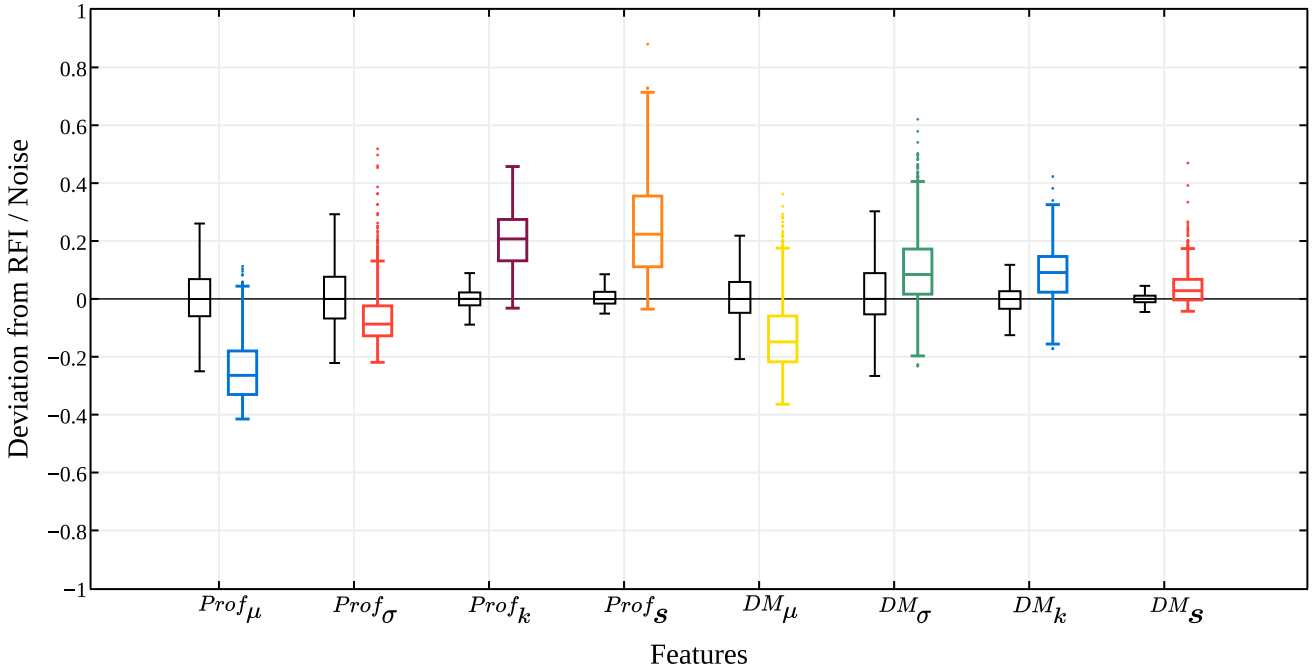

Figure 6. Box plots (median and IQR) showing the linear separability of our new features. Feature data were extracted from 90 000 labelled pulsar candidates produced by Morello et al. (2014), via the PULSAR FEATURE LAB. There are two box plots per feature. The coloured boxes describe the feature distribution for known pulsars, where corresponding coloured dots represent extreme outliers. Those box plots in black describe the RFI/noise distribution. Note that the data of each feature was scaled to the interval [0, 1], before the median of the RFI/noise distribution was subtracted to centre the non-pulsar plots on zero.

Table 6. The point–biserial correlation coefficient for each feature on the three test data sets.

Feature	Dataset			Avg. r_{pb}
	HTRU 1	HTRU 2	LOTAAS 1	
Prof.μ	−0.310	−0.673	−0.508	−0.512
Prof.σ	−0.084	−0.364	−0.337	−0.266
Prof.k	0.545	0.792	0.774	0.719
Prof.s	0.601	0.710	0.762	0.697
DM$_{\mu}$	−0.174	0.401	0.275	0.175
DM$_{\sigma}$	0.059	0.492	0.282	0.287
DM$_{k}$	0.178	−0.391	0.426	0.074
DM$_{s}$	0.190	−0.230	−0.211	−0.096

useful non-linear correlations which are harder to discern. Thus we turn to the tools of information theory (MacKay 2002; Guyon & Elisseeff 2003; Brown 2009) to look for such relationships.

5.1.4 *Information theoretic analysis*

Information theory uses the standard rules of probability to learn more about features and their interactions. Features which at first appear information-poor, may when combined with one or more other features, impart new and meaningful knowledge (Guyon & Elisseeff 2003). Applying this theory to candidate features enables their comparison, evaluation, and selection within an established framework for the first time.

Information theory describes each feature X^j in terms of *entropy*. Entropy is a fundamental unit of information borrowed from thermodynamics by (Shannon & Weaver 1949), that quantifies the uncertainty present in the distribution of X^j.

The entropy of X^j is defined as

$$H(X^j) = -\sum_{x \in X^j} P(x)\log_2 P(x), \tag{9}$$

where x corresponds to each value that X^j can take, and $P(x)$ the probability of x occurring. If a given value of x occurs with a high probability, then the entropy of X^j is low. Conceptually this can be understood to mean that there is little uncertainty over the likely value of X^j. Likewise if all possible values of a feature are equally likely, then there is maximum uncertainty and therefore maximum entropy.[12] Whilst entropy can provide an indication of the uncertainty associated with a feature variable, its main usefulness arises when conditioned on the target variable (true class label) Y. The conditional entropy of X^j given Y is

$$H(X^j|Y) = -\sum_{y \in Y} p(y) \sum_{x \in X^j} P(x|y)\log_2 P(x|y), \tag{10}$$

where $P(x|y)$ is the probability of x given y such that

$$P(x|y) = \frac{P(x \cap y)}{P(y)}. \tag{11}$$

This quantifies the amount of uncertainty in X^j once the value of Y is known. Using equations (9)–(11) it is possible to define the mutual information (MI; Brown et al. 2012)[13] between the feature X^j, and the class label Y. This can be considered another method of measuring the correlation between a feature and the target variable which detects non-linearities. MI is defined as

$$I(X^j; Y) = H(X^j) - H(X^j|Y). \tag{12}$$

[12] Max entropy for a feature with n possible values is given by $\log_2(n)$.
[13] Also known as information gain, or a specific case of the Kullback–Leibler divergence (MacKay 2002).

Table 7. The entropy $H(X^j)$, and mutual information $I(X^j; Y)$ of each feature. Features are ranked according to their mutual information content with respect to the class label Y. Higher mutual information is desirable.

Feature	Dataset						Avg.	
	HTRU 1		HTRU 2		LOTAAS 1			
	$H(X^j)$	$I(X^j; Y)$	$H(X^j)$	$I(X^j; Y)$	$H(X^j)$	$I(X^j; Y)$	$H(X^j)$	$I(X^j; Y)$
Prof.k	1.062	0.073	1.549	0.311	0.948	0.088	1.186	0.157
Prof.μ	1.993	0.065	2.338	0.269	1.986	0.085	2.106	0.139
Prof.s	0.545	0.063	0.523	0.245	0.114	0.074	0.394	0.127
DM$_k$	1.293	0.021	2.295	0.146	1.842	0.083	1.810	0.083
Prof.σ	2.011	0.007	1.972	0.115	2.354	0.061	2.112	0.061
DM$_\sigma$	2.231	0.004	2.205	0.171	0.013	0.006	1.483	0.060
DM$_\mu$	1.950	0.028	0.835	0.114	0.015	0.008	0.933	0.050
DM$_s$	0.138	0.013	1.320	0.041	2.243	0.045	1.233	0.033

The MI expresses the amount of uncertainty in X^j removed by knowing Y. If $I(X^j|Y) = 0$ then X^j and Y are independent. Whereas if $I(X^j|Y) > 0$, then knowing Y helps to better understand X^j. As MI is symmetric, knowing X^j equivalently helps to better understand Y. Thus MI is often described as the amount of information that one variable provides about another (Brown et al. 2012). It is desirable for features to possess high MI with respect to pulsar/non-pulsar labelling.

The MI metric helps identify relevant features, by enabling them to be ranked according to those that result in the greatest reduction of uncertainty. It is one of the most common filter methods (Kohavi & John 1997; Guyon & Elisseeff 2003; Brown et al. 2012) used for feature selection (Brown 2009). The entropy and MI of our features are listed in Table 7, ranked according to their mean MI content, where higher MI is desirable. To produce this table feature data were discretized, for reasons set out by Guyon & Elisseeff (2003), enabling use with the information-theoretic FEAST[14] and MITOOLBOX[15] toolkits developed by Brown et al. (2012). The data were discretized using 10 equal-width bins using the filters within the WEKA data mining tool.[16] Simple binning was chosen ahead of more advanced minimum description length based discretization procedures (Fayyad & Irani 1993), to simplify feature comparisons.

The four features extracted from the integrated profile contain the largest amounts of MI. These are the most relevant features. The MI content of features extracted from the DM-SNR is much lower. It is tempting therefore to write off these low-scoring features since their linear correlation coefficients were also shown to be low in Section 5.1.2. However whilst MI indicates which features are relevant, it is entirely possible for these to contain redundant information (Guyon & Elisseeff 2003). Thus choosing the most relevant features may not produce optimal feature subsets (Kohavi & John 1997), since these could contain the same information. The joint mutual information criterion (JMI; Yang & Moody 1999) can detect and minimize such redundancy (Guyon & Elisseeff 2003; Brown et al. 2012). Given a set of features the JMI selects those with complementary information, starting with the feature possessing the most MI X^1. In 'forward selection' (Kohavi & John 1997; Guyon & Elisseeff 2003), a common method of feature selection, a greedy iterative process is used to decide which additional features are most complementary to X^1, using the notion of the JMI score

$$\mathrm{JMI}(X^j) = \sum_{X^k \in F} I(X^j X^k; Y), \tag{13}$$

where $X^j X^k$ can be understood as a joint probability, and F is the set of features. The iterative process continues until a desired number of features are selected. This produces a feature set that minimises redundancy. Alternatively, if the desired number of features to select equals the total number of those available, features are ranked according to the JMI. Using the JMI in this manner, our features have been ranked such that a lower rank is preferable. Upon applying this criterion poor features are revealed to be useful. This is shown in Table 8 which demonstrates that features extracted from the DM-SNR curve impart complementary information, and are therefore ranked higher than profile features which possess greater MI. The standard deviation of the DM-SNR curve in particular, is ranked as the second 'best' feature on two of the three test data sets. Likewise the excess kurtosis and skewness of the DM-SNR curve, are the second and fourth 'best' features for LOTAAS data, respectively. In the next section we describe a new data stream classification algorithm, which takes advantage of these features.

Table 8. The JMI rank of each feature. Features are ranked according to their average JMI across the three test data sets, where a lower rank is better.

Feature	Dataset			Avg. rank
	HTRU 1	HTRU 2	LOTAAS 1	
Prof.k	1	1	1	1
Prof.μ	3	3	3	3
DM$_\sigma$	2	2	8	4
Prof.s	4	4	6	4.7
DM$_k$	6	6	2	4.7
Prof.σ	7	5	5	5.7
DM$_\mu$	5	7	7	6.4
DM$_s$	8	8	4	6.7

6 STREAM CLASSIFICATION

Data streams are quasi-infinite sequences of information, which are temporally ordered and indeterminable in size (Gaber, Zaslavsky & Krishnaswamy 2005; Lyon et al. 2013, 2014). Data streams are produced by many modern computer systems (Gaber et al. 2005) and are likely to arise from the increasing volumes of data output by modern radio telescopes, especially the SKA. However many of the

[14] http://www.cs.man.ac.uk/~gbrown/fstoolbox
[15] http://www.cs.man.ac.uk/~pococka4/MIToolbox.html
[16] http://www.cs.waikato.ac.nz/ml/weka

effective supervised ML techniques used for candidate selection do not work with streams (Lyon et al. 2014). Adapting existing methods for use with streams is challenging, it remains an active goal of data mining research (Yang & Wu 2006; Gaber, Zaslavsky & Krishnaswamy 2007). Until that goal is realized, new stream-ready selection approaches are required.

6.1 Unsuitability of existing approaches

Supervised ML methods induce classification models from labelled training sets (Mitchell 1997; Bishop 2006). Provided these are large, representative of rare and majority class examples, and independent and identically distributed (i.i.d.) to the data being classified (Bishop 2006) good classification performance can be expected to result. However the notion of a training set does not exist within a data stream. There are instead two general processing models used for learning.

(i) Batch processing model: at time step i, a batch b of n unlabelled instances arrives, and is classified using some model trained on batches b_1 to b_{i-1}. At time $i+1$ labels arrive for batch b_i, along with a new batch of unlabelled instances b_{i+1} to be classified.

(ii) Incremental processing model: a single data instance arrives at time step i defined as X_i, and is classified using some model trained on instances X_1 to X_{i-1}. At time $i+1$ a label arrives for X_i, along with a new unlabelled instance X_{i+1} to be classified.

In both models learning proceeds continually, as labelled data becomes available. This allows for adaptive learning. Standard supervised classifiers simply cannot be trained in this way. Even if they could, the CPU and memory costs of their training phases make them impractical for streams (Gaber 2012). This was recognized by Zhu et al. (2014) with respect to their PICS system.[17]

Given these problems how should candidate selection be addressed in streams? One may consider training an existing supervised candidate classifier offline, which could then be applied to a candidate stream. This is a plausible approach, provided the classifier processes each example before the next one arrives. For this to be viable, the classifier must also be trained with data that is i.i.d. with respect to the data in the stream. However data streams are known to exhibit distributional shifts over varying time periods. For example, a changing RFI environment can exhibit shifts over both short (minutes/hours), and/or long (days/weeks/years) time-scales. In either case the shifts cause violations of the i.i.d. assumption, a phenomena known as 'concept drift' (Widmer & Kubat 1996; Gaber et al. 2005). To mitigate the impact of drift, adaptive algorithms able to learn from distributional changes are required, as pre-existing training data no longer characterises the post-drift data distribution (Lyon 2015). Such algorithms must be capable of completely reconstructing their internal learning models in an efficient manner per each significant distributional shift. Standard supervised learning models are 'static', i.e. they remain unchanged once learned. A static classifier applied to streaming data subject to drifts, will exhibit a significant deterioration in classification performance over time (Aggarwal et al. 2004). This makes standard supervised learning unsuitable for data streams. In the next section we describe our new 'intelligent' data stream classifier, which overcomes these deficiencies.

[17] Zhu et al. (2014) indicated efforts are under way to rectify this.

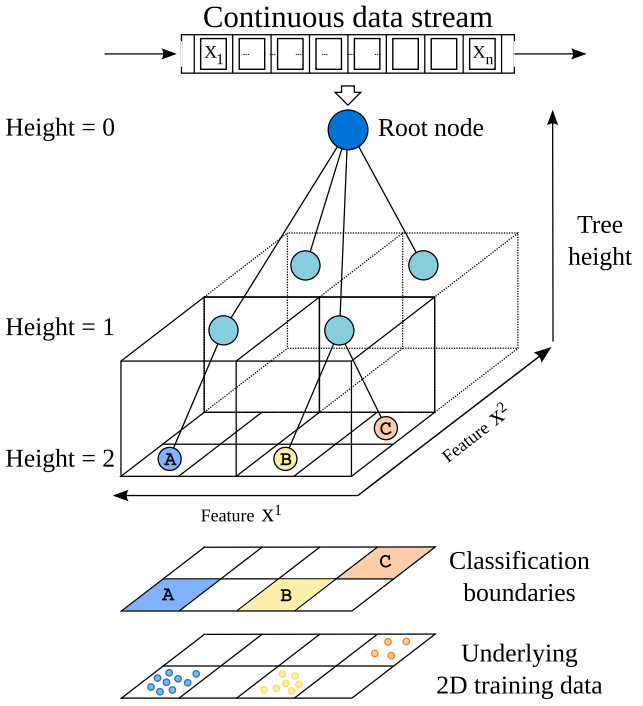

Figure 7. An overview of how a streaming decision tree partitions the data space to derive a classification. Each candidate is passed down the tree, and tested at each node it reaches including the root. Each node test outcome determines which branch the candidate continues down, until it reaches a leaf at the bottom of the tree. The tree shown here assigns the class labels A, B, and C to examples reaching the leaf nodes.

6.2 Gaussian Hellinger Very Fast Decision Tree

The Gaussian Hellinger Very Fast Decision Tree (GH-VFDT) is an incremental stream classifier, developed specifically for the candidate selection problem (Lyon et al. 2014). It is a tree-based algorithm based on the Very Fast Decision tree (VFDT) developed by Hulten, Spence & Domingos (2001). It is designed to maximize classification performance on candidate data streams, which are heavily imbalanced in favour of the non-pulsar class. It is the first candidate selection algorithm designed to mitigate the imbalanced learning problem (He & Garcia 2009; Lyon et al. 2013, 2014), known to reduce classification accuracy when one class of examples (i.e. non-pulsar) dominates the other. The algorithm uses tree learning (Mitchell 1997) to achieve this, whereby the data are partitioned using feature split point tests (see Figs 7 and 8) that aim to maximize the separation of pulsar and non-pulsar candidates. This involves first choosing the variable that acts as the best class separator, and then finding a numerical threshold 'test point' for that variable that maximises class separability.

The tree is 'grown' with labelled data to determine optimal splits, using the Hoeffding bound (Hoeffding 1963). The bound is used to choose statistically with high probability, those split points that would have been selected, if given access to all training data in advance (as in the traditional learning scenario). By calculating the observed mean $\bar{X}^j$ of a feature, the bound is able to determine with confidence $1-\delta$ (where δ is user supplied), that the true mean of the feature is at least $\bar{X}^j - \epsilon$ where

$$\epsilon = \sqrt{\frac{R^2 \ln(1/\delta)}{2n}}, \tag{14}$$

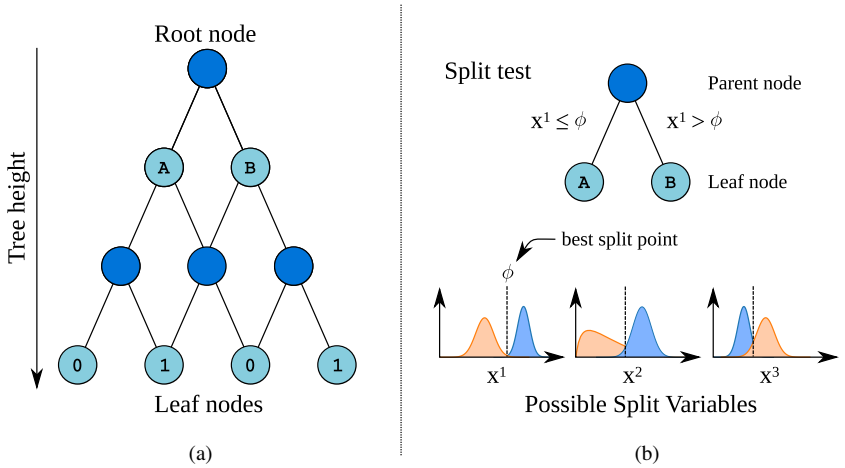

Figure 8. An overview of how a decision tree partitions the data space using binary split point 'tests' at each node. The best feature variable at each node is first determined, then an optimal numerical split point threshold chosen. Candidates with feature values below the threshold are passed down the left-hand branch of the tree, and possibly subjected to further split tests. Similarly for candidates with feature values above the threshold, except these are passed down the right-hand branch. Eventually candidates reach the leaf nodes, where they are assigned class labels.

and R^2 is the feature range squared. This ensures that the statistically optimal split is always chosen. A split is not made until enough examples in the stream have been seen, i.e. until there is enough evidence to advocate its use. The quality of the splits, and therefore the accuracy of the approach, improve over time. This is because the model of the underlying data distributions improves as more examples are observed. The performance of the algorithm approaches that of a non-streamed classifier as the number of examples observed approaches infinity (Hulten et al. 2001). The tree is also able to adapt to change (Lyon 2015) by updating the data distributions with each observed labelled example. Once there is evidence to suggest an alternative split point is better than one in use, the tree replaces the sub-optimal split. This is achieved by pruning the branch of the tree containing the sub-optimal split, and replacing it with a new branch which begins to 'grow' from the new split point.

The key feature of the GH-VFDT, is its use of the skew-insensitive Hellinger distance measure (Hellinger 1909; Nikulin 2001) to evaluate split points during learning. This measure makes the classifier robust to the imbalanced learning problem, preventing the classifier from becoming biased towards the abundant non-pulsar class (Lyon 2015). By modelling each feature distribution as a Gaussian, the Hellinger distance between the pulsar and non-pulsar distributions can be measured. If Q and N are the pulsar and non-pulsar distributions, respectively, the distance for a single feature is given by

$$d_{\mathrm{H}}(Q, N) = \sqrt{1 - \sqrt{\frac{2\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2}} \mathrm{e}^{-\frac{1}{4}\frac{(\mu_1-\mu_2)^2}{\sigma_1^2+\sigma_2^2}}}, \tag{15}$$

where Q has mean μ_1, variance σ_1^2 and standard deviation σ_1, with N defined similarly. The goal of split evaluation is to choose the split which maximizes the Hellinger distance, maximizing pulsar and non-pulsar separation. This approach requires that only the mean and standard deviation of each feature be known. This significantly reduces the GH-VFDT memory overheads, as knowledge of the entire feature distribution(s) is not required for learning. Therefore the runtime and memory requirements of the algorithm are sub-linear with respect to the number of examples processed, and grow in only constant time for each new node added to the tree. This makes the algorithm suitable for use upon very high throughput data streams such as those described in Section 4.3.

Algorithm 1 Gaussian Hellinger Very Fast Decision Tree

Require: An input stream $S = \{..., (X_i, y_i), ...\}$, such that each X_i is a candidate, X_i^j its j-th feature and y_i its class label. The parameter $\delta \in (0, 1)$ is the confidence desired, and $\tau \in (0, 1)$ a parameter which if set, prevents split point ties.

1: **procedure** GH-VFDT(S, δ, τ)
2: Let DT be a decision tree with leaf l_1
3: **for** $i \leftarrow 1$ to $|S|$ **do** ▷ For each stream instance.
4: $l \leftarrow sort(X_\mathrm{i}, y_\mathrm{i})$ ▷ Sort instance X_i to leaf l.
5: $k \leftarrow y_\mathrm{i}$ ▷ Get class.
6: **for** $j \leftarrow 1$ to $|X_\mathrm{i}^\mathrm{j}|$ **do** ▷ For each feature.
7: update $\mu_{\mathrm{jk}}(l, X_\mathrm{i}^\mathrm{j})$ ▷ Update observed μ at leaf.
8: update $\sigma_{\mathrm{jk}}(l, X_\mathrm{i}^\mathrm{j})$ ▷ Update observed σ at leaf.
9: Label l with majority class of instances seen at l
10: **if** all X_i seen at l don't belong to same class **then**
11: $F_\mathrm{a} \leftarrow null$ ▷ Best feature.
12: $F_\mathrm{b} \leftarrow null$ ▷ 2nd best feature.
13: **for** $j \leftarrow 1$ to $|X_\mathrm{i}^\mathrm{j}|$ **do** ▷ For each feature.
14: $dist \leftarrow d_H(X_\mathrm{i}^\mathrm{j})$ ▷ From equation 15.
15: $F_\mathrm{a}, F_\mathrm{b} \leftarrow getBest(dist, X_\mathrm{i}^\mathrm{j})$
16: $\epsilon = \sqrt{\frac{R^2\ ln(1/\delta)}{2n}}$ ▷ Hoeffding bound.
17: **if** $d_\mathrm{H}(F_\mathrm{a}) - d_\mathrm{H}(F_\mathrm{b}) > \epsilon$ or $\epsilon < \tau$ **then**
18: Replace l with new leaf that splits on F_a
19: **for** each branch of split **do**
20: Add new leaf l_m
21: **for** $k \leftarrow 1$ to $|S|$ **do** ▷ For each class.
22: **for** $j \leftarrow 1$ to $|X_\mathrm{i}|$ **do** ▷ For each X_i^j.
23: $\mu_{\mathrm{ijk}}(l_\mathrm{m}) \leftarrow 0$
24: $\sigma_{\mathrm{ijk}}(l_\mathrm{m}) \leftarrow 0$
25: **return** DT

A complete outline of the GH-VFDT is given in Algorithm 1. On line 7 tree statistics used to compute the Hellinger distance are updated. In particular, the running mean and standard deviation maintained at each leaf, for feature j, and class k are updated. The call to $getBest(dist, X_i^j)$ returns the best and second best features

Table 9. Confusion matrix describing the outcomes of binary classification.

		Predicted	
		−	+
Actual	−	True negative (TN)	False positive (FP)
	+	False negative (FN)	True positive (TP)

found at a leaf. This is achieved by choosing those that maximize the Hellinger distance via an iterative process. On line 18 tree split points are first generated and evaluated. Here data are discretized using 10 equal-width bins, and a binary split point chosen.

This approach has already been shown to significantly improve recall rates for pulsar data, above the levels achieved by established stream classifiers. When applied to a data stream containing 10 000 non-pulsar candidates for every legitimate pulsar (HTRU data obtained by Thornton (2013)), it raised the recall rate from 30 to 86 per cent (Lyon et al. 2014). This was achieved using candidate data described using the features designed by Bates et al. (2012) and Thornton (2013). A full implementation of the algorithm can be found online for public use.[18]

6.3 Classification performance

Existing features and algorithms have been evaluated predominantly in terms of classification accuracy. Such an analysis considers candidate selection as a binary classification problem, whereby candidates arising from pulsars are considered positive (+), and those from non-pulsars negative (−). There are then four possible outcomes for an individual classification decision. These outcomes are summarized in Table 9 and are evaluated using standard metrics such as those outlined in Table 10. The goal of classification is to minimize the number of false positives, whilst maximizing the true positives. Features in this domain are most often chosen according to how well they maximize classifier recall (the fraction of legitimate pulsar candidates correctly classified) and specificity (fraction of non-pulsar candidates correctly classified).[19] Those classifiers with high recall and specificity exhibit high accuracy, often interpreted to mean that underlying features are good discriminators.

This form of evaluation enables approaches to be tested quickly, with readily interpretable results. However using classifier performance as a proxy to measure feature-separability tests the classification system used as much as the features under investigation (Brown et al. 2012). The choice of classifier can influence the outcome of the evaluation giving misleading results. Evaluation metrics themselves can also be misleading. Pulsar data sets are imbalanced with respect to the total number of pulsar and non-pulsar candidates within them (Lyon et al. 2013, 2014). Thus for data sets consisting of almost entirely non-pulsar examples, high accuracy can often be achieved by classifying all candidates as non-pulsar. In these situations it is an unhelpful metric.

To overcome these possible sources of inaccuracy when evaluating the GH-VFDT, we make use of the G-mean metric (He & Garcia 2009). This describes the ratio between positive and negative accuracy, a measure insensitive to the distribution of pulsar and non-pulsar examples in test data sets. Additionally we employ multiple classifiers in our evaluation which differ greatly in terms of their internal learning models. This allows for a more general view of feature performance in practice to be revealed. This is also useful for evaluating the performance of the GH-VFDT with respect to standard static supervised classifiers, which are at an advantage in such tests. Here we make use of four standard classifiers found in the WEKA tool. These include the decision tree algorithm C4.5 (Quinlan 1993), MLP neural network (Haykin 1999), a simple probabilistic classifier Naïve Bayes (NB; Bishop 2006), and the standard linear soft-margin support vector machine (SVM; Cortes & Vapnik 1995).

6.3.1 *GH-VFDT classification evaluation procedure*

Feature data were extracted from the data sets listed in Table 5, and then independently sampled 500 times. Each sample was split into test and training sets. For HTRU 1 and 2, sampled training sets consisted of 200 positive and 200 negative examples, with remaining examples making up the test sets. LOTAAS 1 training sets contained 33 positive examples and 200 negative, with remaining examples similarly making up the test sets. Each classifier (five in total) was then trained upon, and made to classify each independent sample, therefore there were $3 \times 500 \times 5 = 7500$ tests in total. The performance of each algorithm per data set was then averaged to summarize overall performance. To evaluate classifier performance results, one-factor analysis of variance tests were performed, where the algorithm used was the factor. Tukey's Honestly Significant Difference test (Tukey 1949), was then applied to determine if differences in results were statistically significant at $\alpha = 0.01$. The full results are shown in Table 11.

These results indicate that it is possible to achieve high levels of classifier performance using the features described in Section 5. What is more, the classification results are consistent across all three data sets. Recall rates on all three test data sets are high, with 98 per cent recall achieved by the MLP on HTRU 1 and LOTAAS 1 data. High levels of accuracy were observed throughout testing and G-mean scores on HTRU 1 were particularly high. The algorithms also exhibited high levels of specificity and generally low false positive rates. The exception being the 6 per cent false positive rate achieved by the NB classifier on HTRU 2 data. This outcome is unremarkable for NB, the simplest classifier tested, as the HTRU 2 data set is populated with noise and borderline candidates. Thus we suggest that these represent the first survey independent features developed for the candidate selection problem.

The results also show that the GH-VFDT algorithm consistently outperformed the static classifiers, in terms of both specificity and false positive return rate. This is a highly desirable outcome for a stream classifier, since assigning positive labels too often will return an unmanageable number of candidates. The classifier does not always predict 'non-pulsar' to give this result. It is precise, achieving the best precision on two out of the three data sets. G-mean and recall rates were also high for the GH-VFDT, the latter reaching 92.8 per cent on HTRU 1 data. The recall rates are lower on the remaining two data sets. However it is worth noting that these data sets are considerably smaller than HTRU 1. This is important, since the performance of the GH-VFDT (and of other stream algorithms) improves as more examples are observed. The lower levels of recall on HTRU 2 and LOTAAS 1 are therefore to be expected given the smaller data set size. In terms of the usefulness of this algorithm for SKA data streams, the GH-VFDT returns consistently less than 1 per cent of candidates as false positives. This greatly reduces the quantity of candidates to be analysed. The GH-VFDT also classifies candidates rapidly. It classified candidates at a rate of ∼70 000 per second using a single 2.2 GHz Quad Core mobile CPU (Intel Core

[18] https://github.com/scienceguyrob/GHVFDT
[19] The approaches in Section 3.4 evaluate in this manner.

Table 10. Standard evaluation metrics for classifier performance. True Positives (TP) are those candidates correctly classified as pulsars. True Negatives (TN) are those correctly classified as *not* pulsars. False Positives (FP) are those incorrectly classified as pulsars, False Negatives (FN) are those incorrectly classified as *not* pulsars. All metrics produce values in the range [0, 1].

Statistic	Description	Definition
Accuracy	Measure of overall classification accuracy.	$\frac{(TP+TN)}{(TP+FP+FN+TN)}$
False positive rate (FPR)	Fraction of negative instances incorrectly labelled positive.	$\frac{FP}{(FP+TN)}$
G-Mean	Imbalanced data metric describing the ratio between positive and negative accuracy.	$\sqrt{\frac{TP}{TP+FN} \times \frac{TN}{TN+FP}}$
Precision	Fraction of retrieved instances that are positive.	$\frac{TP}{(TP+FP)}$
Recall	Fraction of positive instances that are retrieved.	$\frac{TP}{(TP+FN)}$
F-Score	Measure of accuracy that considers both precision and recall.	$2 \times \frac{precision \times recall}{precision+recall}$
Specificity	Fraction of negatives correctly identified as such.	$\frac{TN}{(FP+TN)}$

Table 11. Results obtained on the three test data sets. Bold type indicates the best performance observed. Results with an asterisk indicate no statistically significant difference at the $\alpha = 0.01$ level.

Data set	Algorithm	G-Mean	F-Score	Recall	Precision	Specificity	FPR	Accuracy
HTRU 1	C4.5	0.962*	0.839*	0.961	0.748	0.962	0.038	0.962
	MLP	**0.976**	0.891	**0.976**	0.820	0.975	0.025*	0.975
	NB	0.925	0.837*	0.877	0.801	0.975	0.025*	0.965
	SVM	0.967	0.922	0.947	0.898	0.988	0.012	0.984
	GH-VFDT	0.961*	**0.941**	0.928	**0.955**	**0.995**	**0.005**	**0.988**
HTRU 2	C4.5	0.926	0.740	**0.904**	0.635*	0.949*	0.051*	0.946*
	MLP	**0.931**	0.752	0.913	0.650*	0.950*	0.050*	0.947*
	NB	0.902	0.692	0.863	0.579	0.943	0.057	0.937
	SVM	0.919	0.789	0.871	0.723	0.969	0.031	0.961
	GH-VFDT	0.907	**0.862**	0.829	**0.899**	**0.992**	**0.008**	**0.978**
LOTAAS 1	C4.5	0.969	0.623	0.948	0.494	0.991	0.009	0.990
	MLP	**0.988**	0.846*	**0.979**	0.753	0.998	0.002	0.997*
	NB	0.977	0.782	0.959	0.673	0.996	0.004	0.996
	SVM	0.949	**0.932**	0.901	**0.966**	**0.999***	**0.001***	**0.999**
	GH-VFDT	0.888	0.830*	0.789	0.875	**0.999***	**0.001***	0.998*

i7-2720QM Processor) when applied to a larger sample of HTRU 2 data consisting of 11 million examples. A discussion of the statistics of the pulsars incorrectly classified by the new methods will be discussed in a future paper.

7 SUMMARY

This paper has described the pulsar candidate selection process, and contextualized its almost 50 year history. During this time candidate selection procedures have been continually adapting to the demands of increased data capture rates and rising candidate numbers, which has proven to be difficult. We have contributed a new solution to these problems by demonstrating eight new features useful for separating pulsar and non-pulsar candidates, and by developing a candidate classification algorithm designed to meet the data processing challenges of the future. Together these enable a high fraction of legitimate pulsar candidates to be extracted from test data, with recall rates reaching almost 98 per cent. When applied to data streams, the combination of these features and our algorithm enable over 90 per cent of legitimate pulsar candidates to be recovered. The corresponding false positive return rate is less than half a per cent. Thus together these can be used to significantly reduce the problems associated with high candidate numbers which make pulsar discovery difficult, and go some way towards mitigating the selection problems posed by next-generation radio telescopes such as the SKA. The combination of these features and our classification algorithm has already proven useful, aiding in the discovery of 20 new pulsars in data collected during the LOTAAS (Cooper 2014). Details of these discoveries will be provided elsewhere, demonstrating the utility of our contributions in practice.

The features described in this paper are amongst the most rigorously tested in this domain. However whilst we advocate their use on statistical grounds, we do not demonstrate their superiority to other features. Future work will consider how these compare to those used previously, and determine if combining them with those already in use is worthwhile. Thus for the time being it is advisable to construct as large a set of features as possible, and use the tools described herein to select feature sets statistically.

ACKNOWLEDGEMENTS

This work was supported by grant EP/I028099/1 from the UK Engineering and Physical Sciences Research Council (EPSRC). HTRU 2 data were obtained by the High Time Resolution Universe Collaboration using the Parkes Observatory, funded by the Commonwealth of Australia and managed by the CSIRO. LOFAR data were obtained with the help of the DRAGNET team, supported by ERC Starting Grant 337062 (PI Hessels). We would also like to thank Konstantinos Sechidis for some insightful discussions with respect to information theoretic feature selection, Dan Thornton for

initially processing HTRU 2 data, and our reviewer for their helpful feedback.

REFERENCES

Abdo A. A. et al., 2009, Science, 325, 840
Aggarwal C. et al., 2004, Proc. 10th Int. Conf. on Knowledge discovery and data mining. ACM, New York, NY, p. 503
Ait-Allal D., Weber R., Dumez-Viou C., Cognard I., Theureau G., 2012, C. R. Phys., 13, 80
ATLAS Collaboration, 2008, J. Instrum., 3, 04003
Ball N., Brunner R. J., 2009, Int. J. Mod. Phys. D, 19, 7
Barber B. M., Odean T., 2000, J. Finance, 55, 2
Barr E. D., 2014, Proc. Extreme-Astrophysics in an Ever-Changing Universe: Time-Domain Astronomy in the 21st Century. Available at: http://www3.mpifr-bonn.mpg.de/div/jhs/Program_files/EwanBarrCrete2014.pdf (accessed 2016 January 6)
Barr E. D. et al., 2013, MNRAS, 435, 2234
Bates S. D., 2011, PhD thesis, Univ. Manchester
Bates S. D. et al., 2011, MNRAS, 411, 1575
Bates S. D. et al., 2012, MNRAS, 427, 1052
Bengio J., 2009, Found. Trends Mach. Learn., 2, 1
Bhattachatyya B., 2014, Proc Transient Key science project meeting 2014. Available at: http://www.jb.man.ac.uk/meetings/transients2014/pdfs/Bhaswati.pdf (accessed 2016 January 6)
Bhattachatyya B. et al., 2016, ApJ, 817, 130
Biggs J. D., Lyne A. G., 1992, MNRAS, 254, 257
Bishop C. M., 2006, Pattern Recognition and Machine Learning. Springer-Verlag, New York
Borne K. D., 2009, Next Generation of Data Mining. CRC Press, Boca Raton, FL, p. 91
Boyles J. et al., 2013, ApJ, 763, 80
Brown G., 2009, Proc. 12th Int. Conf. Artif. Intell. Stat., p. 49
Brown G., Pocock A., Zhao Z., Luján M., 2012, J. Mach. Learn. Res., 13, 27
Burgay M. et al., 2006, MNRAS, 368, 283
Burgay M. et al., 2013, MNRAS, 429, 579
Burns W. R., Clark B. G., 1969, A&A, 2, 280
Camilo F., Nice D. J., Taylor J. H., 1993, ApJ, 412, L37
Champion D. J., McLaughlin M. A., Lorimer D. R., 2005, MNRAS, 364, 1011
Chandola V., Banerjee A., Kumar V., 2009, ACM Comput. Surv., 41, 3
Clifton T. R., Lyne A. G., 1986, Nature, 320, 43
Coenen T. et al., 2014, A&A, 570, A60
Cooper S., 2014, Proc. LOFAR Sci. Available at: http://www.astron.nl/lofarscience2014/Documents/Tuesday/Session (accessed 2016 January 6)
Cordes J. M. et al., 2006, ApJ, 637, 446
Cortes C., Vapnik V., 1995, Mach. Learn., 20, 273
Crawford F. et al., 2006, ApJ, 652, 1499
Damashek M., Taylor J. H., Hulse R. A., 1978, ApJ, 225, L31
Damashek M., Backus P. R., Taylor J. H., Burkhardt R. K., 1982, ApJ, 253, L57
Das Gupta S., 1960, Psychometrika, 25, 4
Davies J. G., Large M. I., Pickwick A. C., 1970, Nature, 227, 1123
Davies J. G., Lyne A. G., Seiradakis J. H., 1977, MNRAS, 179, 635
Deich W. T. S., 1994, PhD thesis, California Institute of Technology
Deneva J. S. et al., 2009, ApJ, 703, 2259
Deneva J. S., Stovall K., McLaughlin M. A., Bates S. D., Freire P. C. C., Martinez J. G., Jenet F., Bagchi M., 2013, ApJ, 775, 1
Desvignes G., Cognard I., Champion D., Lazarus P., Lespagnol P., Smith D. A., Theureau G., 2012, in van Leeuwen J., ed., Proc. IAU Symposium 291, Neutron Stars and Pulsars: Challenges and Opportunities After 80 Years. Cambridge Univ. Press, Cambridge, p. 375
Dewey R. J., Taylor J. H., Weisberg J. M., Stokes G. H., 1985, ApJ, 294, L25
Duda R. O., Hart P. E., Stork D. G., 2000, Pattern Classification, 2nd edn. Wiley, New York
Eatough R. P., 2009, PhD thesis, Univ. Manchester
Eatough R. P., Molkenthin N., Kramer M., Noutsos A., Keith M. J., Stappers B. W., Lyne A. G., 2010, MNRAS, 407, 2443
Eatough R. P., Kramer M., Lyne A. G., Keith M. J., 2013, MNRAS, 431, 292
Edwards R. T., Bailes M., van Straten W., Britton M. C., 2001, MNRAS, 326, 358
Faulkner A. J. et al., 2004, MNRAS, 355, 147
Fayyad U., Irani K., 1993, Proc. 13th Int. Joint Conf. on Artificial Intelligence (IJCAI), 1022
Fisher R. A., 1921, Metron 1, 3
Foster R. S., Cadwell B. J., Wolszczan A., Anderson S. B., 1995, ApJ, 454, 826
Gaber M. M., 2012, Data Mining and Knowledge Discovery. Springer-Verlag, Berlin, p. 79
Gaber M. M., Zaslavsky A., Krishnaswamy S., 2005, ACM SIGMOD Record, 34, 18
Gaber M. M., Zaslavsky A., Krishnaswamy S., 2007, Advances in Database Systems. Springer US, Boston, MA, p. 39
Guyon I., Elisseeff A., 2003, J. Mach. Learn. Res., 3, 1157
Haykin S., 1999, Neural Networks A Comprehensive Foundation, Prentice-Hall, Englewood Cliffs, NJ
He H., Garcia E. A., 2009, IEEE Trans. Knowl. Data Eng., 21, 1263
Hellinger E., 1909, J. reine Angew. Math., 136, 210
Hessels J. W. T., Ransom S. M., Stairs I. H., Kaspi V. M., Freire P. C. C., 2007, ApJ, 670, 363
Hewish A., Bell S. J., Pilkington J. D. H., Scott P. F., Collins R. A., 1968, Nature, 217, 5130
Hodge V. J., Austin J., 2004, Artif. Intell. Rev., 22, 85
Hoeffding W., 1963, J. Am. Stat. Assoc., 58, 13
Hogden J., Wiel S. V., Bower G. C., Michalak S., Siemion A., Werthimer D., 2012, ApJ, 747, 141
Hughes G., 1968, Inf. Theory, 14, 55
Hulse R. A., Taylor J. H., 1974, ApJ, 191, L59
Hulten G., Spencer L., Domingos P., 2001, Proc. of 7th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining
Jacoby B. A., Bailes M., Ord S. M., Edwards R. T., Kulkarni S. R., 2009, ApJ, 699, 2009
Janssen G. H., Stappers B. W., Braun R., van Straten W., Edwards R. T., Rubio-Herrera E., van Leeuwen J., Weltevrede P., 2009, A&A, 498, 223
Johnston S., Lyne A. G., Manchester R. N., Kniffen D. A., D'Amico N., Lim J., Ashworth M., 1992, MNRAS, 255, 401
Karastergiou A. et al., 2015, MNRAS, 452, 1254
Keane E. F., Stappers B. W., Kramer M., Lyne A. G., 2012, MNRAS, 425, L71
Keane E. F. et al., 2014, Proc. Sci., Advancing Astrophysics with the Square Kilometre Array (AASKA14). SISSA, Trieste, PoS#40
Keith M. J, Eatough R. P., Lyne A. G., Kramer M., Possenti A., Camilo F., Manchester R. N., 2009, MNRAS, 395, 837
Keith M. J. et al., 2010, MNRAS, 409, 619
Kohavi R., John G. H., 1997, Artif. Intell., 97, 273
Large M. I., Vaughan A. E., Wielebinski R., 1968, Nature, 220, 753
Law C. J. et al., 2015, ApJ, 807, 16
Lazarus P., 2012, in van Leeuwen J., ed., Proc. IAU Symp 291, Neutron Stars and Pulsars: Challenges and Opportunities After 80 Years. Cambridge Univ. Press, Cambridge, p. 35
Lee K. J. et al., 2013, MNRAS, 433, 688
Levin L., 2012, PhD thesis, Swinburne University
LOFAR Pulsar Working Group 2013, Proc. LOFAR Status Meeting. Available at: http://www.lofar.org/wiki/lib/exe/fetch.php?media=public:lsm_new:2013_03_06_hesself.pdf (accessed 2016 January 6)
Lorimer D., Kramer M., 2006, Handbook of Pulsar Astronomy. Cambridge Univ. Press, Cambridge
Lorimer D. R., Bailes M., McLaughlin M. A., Narkevic D. J., Crawford F., 2007, Science, 318, 777
Lorimer D. R., Camilo F., McLaughlin M. A., 2013, MNRAS, 434, 347

Lorimer D. R. et al., 2015, MNRAS, 450, 2185
Lyon R. J., 2015, PhD thesis, Univ. Manchester
Lyon R. J., Brooke J. M., Knowles J. D., Stappers B. W., 2013, IEEE Trans. Syst. Man Cybern., 1506
Lyon R. J., Brooke J. M., Knowles J. D., Stappers B. W., 2014, 22nd Int. Conf. on Pattern Recognition, p. 1969
MacKay D. J. C., 2002, Information Theory, Inference and Learning Algorithms. Cambridge Univ. Press, Cambridge
Manchester R. N., Lyne A. G., Taylor J. H., Durdin J. M., Large M. I., Little A. G., 1978, MNRAS, 185, 409
Manchester R. N., Lyne A. G., D'Amico N., Johnston S., Lim J., Kniffen D. A., 1990a, Nature, 345, 598
Manchester R. N., Lyne A. G., Robinson C., D'Amico N., Bailes M., Lim J., 1990b, Nature, 352, 219
Manchester R. N. et al., 1996, MNRAS, 279, 1235
Manchester R. N. et al., 2001, MNRAS, 328, 17
Manchester R. N., Hobbs G. B., Teoh A., Hobbs M., 2005, AJ, 129, 4
Manchester R. N., Fan G., Lyne A. G., Kaspi V. M., Crawford F., 2006, ApJ, 649, 235
Markou M., Singh S., 2003, Signal Process., 18, 2499
Meehl P. E., 1954, Clinical Versus Statistical Prediction: A Theoretical Analysis and a Review of the Evidence. Univ. Minnesota Press, MN
Mitchell T. M., 1997, Machine Learning, 1st edn. McGraw-Hill, New York City, NY
Morello V., Barr E. D., Bailes M., Flynn C. M., Keane E. F., van Straten W., 2014, MNRAS, 443, 1651
Navarro J., Anderson S. B., Freire P. C. C., 2003, ApJ, 594, 943
Ng C., 2012, in van Leeuwen J., ed., Proc. IAU Symp. 291, Neutron Stars and Pulsars: Challenges and Opportunities After 80 Years. Cambridge Univ. Press, Cambridge, p. 53
Nice D. J., Taylor J. H., Fruchter A. S., 1993, ApJ, 402, L49
Nice D. J., Fruchter A. S., Taylor J. H., 1995, ApJ, 449
Nikulin N. S. et al., 2001, Encyclopedia of Mathematics. Springer, Berlin
P-Alfa Consortium 2015, ALFA Pulsar Studies. Available at: http://www.naic.edu/alfa/pulsar/ (accessed 2016 January 6)
Pearson K., 1895, Proc. R. Soc. A, 58, 347
Petroff E. et al., 2015, MNRAS, 447, 246
Quinlan J. R., 1993, C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo, CA
Ransom S. M. et al., 2011, ApJ, 727, L16
Rosen R. et al., 2010, Astron. Educ. Rev., 9, 010106
Rosen R. et al., 2013, ApJ, 768, 85
Rubio-Herrera E., Braun R., Janssen G., van Leeuwen J., Stappers B. W., 2007, preprint (astro-ph/0701183)
Sayer R. W., Nice D. J., Taylor J. H., 1997, ApJ, 474, 1
Shannon C. E., Weaver W., 1949, The Mathematical Theory of Communication. Univ. of Illinois Press, Champaign, IL
Smits R., Kramer M., Stappers B., Lorimer D. R., Cordes J., Faulkner A., 2009a, A&A, 493, 1161
Smits R., Lorimer D. R., Kramer M., Manchester R., Stappers B., Jin C. J., Nan R. D., Li D., 2009b, A&A, 505, 919
Spitler L. G. et al., 2014, ApJ, 790, 101
Stokes G. H., Taylor J. H., Weisberg J. M., Dewey R. J., 1985, Nature, 317, 787
Stokes G. H., Segelstein D. J. Taylor J. H., Dewey R. J., 1986, ApJ, 311, 694
Stovall K., Lorimer D. R., Lynch R. S., 2013, Class. Quantum Gravity, 30, 22
Stovall K. et al., 2014, ApJ, 791, 22
Swiggum J. K. et al., 2015, ApJ, 805, 156
Taylor J. H., Jura M., Huguenin G. R., 1969, Nature, 223, 797
Thompson D. R., Majid W. A., Wagstaff K., Reed C., 2011, in Srivastava A. N., Chawla N. V., Perera A. S., eds, NASA Conf. Intelligent Data Understanding
Thornton D., 2013, PhD thesis, Univ. Manchester
Thornton D. et al., 2013, Science, 341, 53
Tukey J., 1949, Biometrics, 5, 99
van Heerden E., Karastergiou A., Roberts S. J., Smirnov O., 2014, General Assembly and Scientific Symposium XXXIth URSI
Way M. J., Scargle J. D., Ali K. M., Srivastava A. N., 2012, Advances in Machine Learning and Data Mining for Astronomy, 1st edn. Taylor and Francis, London
Widmer G., Kubat M., 1996, Mach. Learn., 23, 69
Wilcoxon F., 1945, Biometrics Bull., 1, 80
Yang H. H., Moody J., 1999, NIPS, 12, 687
Yang Q., Wu X., 2006, Int. J. Inf. Technol. Decis. Mak., 5, 597
Zhu W. W. et al., 2014, ApJ, 781, 117

This paper has been typeset from a TEX/LATEX file prepared by the author.