SPOTLIGHT: A realtime detection pipeline for FRBs and pulsars at uGMRT

A Thesis

submitted to
Indian Institute of Science Education and Research Pune
in partial fulfillment of the requirements for the
BS-MS Dual Degree Programme

by

Ritavash Debnath

Indian Institute of Science Education and Research Pune
Dr. Homi Bhabha Road,
Pashan, Pune 411008, INDIA.

May, 2025

Supervisor: Prof. Jayanta Roy
Co-supervisor: Prof. Wes Armour

Certificate

This is to certify that this dissertation entitled "SPOTLIGHT: A realtime detection pipeline for FRBs and pulsars at uGMRT" towards the partial fulfilment of the BS-MS dual degree programme at the Indian Institute of Science Education and Research, Pune represents study/work carried out by Ritavash Debnath at the National Centre for Radio Astrophysics, Pune under the supervision of Prof. Jayanta Roy, Associate Professor, and at the Oxford E-Research Centre, Oxford under the co-supervision of Prof. Wes Armour, Professor, during the academic year 2024-2025.

Prof. Jayanta Roy

Committee:

Prof. Jayanta Roy 

Prof. Wes Armour

Prof. Prasad Subhramanian 

This thesis is dedicated to my brother, mother, and father

Declaration

I hereby declare that the matter embodied in the report entitled SPOTLIGHT: A realtime detection pipeline for FRBs and pulsars at uGMRT are the results of the work carried out by me at the National Centre for Radio Astrophysics,Pune, under the supervision of Prof. Jayanta Roy and at the Oxford E-Research Centre, Oxford under the co-supervision of Prof. Wes Armour and the same has not been submitted elsewhere for any other degree.

Ritavash Debnath

Acknowledgments

I would like to express my sincere gratitude to my supervisor Prof. Jayanta Roy, and my co-supervisor Prof. Wes Armour, for their invaluable guidance, feedback, and support throughout my research. Their extensive extensive knowledge and experience were instrumental in the completion of this dissertation. Their insightful suggestion and comments pushed me to sharpen my thinking and brought greater depth to this work. I would also like to acknowledge my thesis expert Dr. Prasad Subhramanian for his valuable feedback and suggestions on the mid-year report and presentation, which helped me build my final thesis better.

Further, I would like to thank the SPOTLIGHT project members and GMRT engineers for their invaluable help and essential collaboration in teaching me the various parts of time-domain astronomy.

In addition, I wish to acknowledge the scholarship grant from "DAE-STFC Technology and Skills Programme - Building Indo-UK collaboration towards the Square Kilometre Array (SKA)" for my visit to the University of Oxford.

My appreciation also goes out to the library staff for their assistance in accessing various resources, and the IT support team at IISER, NCRA, and OeRC for their help with the technological aspects of my project. Their collective efforts enabled me to conduct my research efficiently.

Abstract

My thesis focuses on time-domain astronomy, a rapidly evolving field dedicated to the study of rapidly varying celestial sources. In particular, I explore the nature of transient radio sources, with an emphasis on pulsars and Fast Radio Bursts (FRBs). Pulsars, which are rapidly rotating, strongly magnetized neutron stars, serve as precise cosmic clocks and offer insights into extreme states of matter and gravity. FRBs, on the other hand, are millisecond-duration radio flashes observable at cosmological distances, making them powerful tools for probing highly energetic events and the ionized intergalactic medium (IGM).

The core of my research is linked to the SPOTLIGHT project — a commensal, GPU-powered, real-time survey instrument designed to enhance the discovery potential of the upgraded Giant Metrewave Radio Telescope (uGMRT). SPOTLIGHT aims to detect and localize FRBs and pulsars across the 300–1460 MHz radio spectrum, with the goal of discovering hundreds of these transients and associating FRBs with their host galaxies.

Working under the guidance of Prof. Wes Armour (University of Oxford) and Prof. Jayanta Roy (NCRA-TIFR), I contributed to the commissioning of SPOTLIGHT by working on building its end-to-end detection pipeline and preparing it for the early science phase starting in late 2024. This project is part of the "Building Indo-UK Collaborations Towards the Square Kilometre Array" initiative, funded by the DAE-STFC Technology and Skills Programme 2023. Through this work, I aim to push the boundaries of time-domain astronomy and support the discovery of new pulsars and FRBs using the uGMRT.

Contents

List of Figures

List of Tables

Introduction

Thousands of light years away from us, the universe is much more violent and extreme, consisting of massive star explosions (supernovae), neutron star mergers, supermassive black holes, etc. These cosmic objects emit bursts of ultra-high energy radiation [35]($\sim 10^{40}$-10^{45}Joules) which can be detected at large distances. Particles accelerated in these regions reach immense energies which are impossible to recreate in laboratories. But the study of such environments is crucial to understand the laws of physics operating in extraordinary regimes. During the 21st century, these environments have been studied over a large range of wavelengths. Apart from high-energy emissions in the X-ray and gamma-ray bands, these regions also exhibit strong radio emission. Powerful magnetic fields and shock waves — generated as matter interacts with the surrounding medium — produce radio bursts from electrons (and positrons) spiraling around and occasionally channeled along magnetic field lines. X-ray and gamma-ray studies have revealed a universe teeming with such extreme sources, often missed by traditional optical telescopes due to their narrow field of view and the obscuring effects of intervening dust. These events being so rapid, rare, and have an almost uniform distribution in the sky require a rapid all sky survey with high fields of view. One of the key advantages of radio monitors over traditional X-ray and gamma-ray monitors is their ability to detect and localize events immediately with arc-sec precision. A wide variety of these radio sources have been detected by prominent radio facilities around the world like - Low-Frequency Array (LOFAR), Giant Meter Radio Telescope (GMRT), South African Karoo Array Telescope (MeerKAT), Australian SKA Pathfinder (ASKAP), Canadian Hydrogen Intensity Mapping Experiment (CHIME), Five-hundred-meter Aperture Spherical radio Telescope (FAST), etc. These radio objects can be classified into two broad types based on their temporal variability in the sky - faint radio sources like radio galaxies, Active Galactic Nuclei (AGN) populations, etc, which have almost constant observed intensity throughout our lifetime are known as the persistent radio sources. Brighter radio sources

like neutron stars, radio jets, magnetars, etc, whose intensity varies on short timescales (from few milliseconds to months)[57] are known as radio transients. The transients can be classified into two types based on their source emission mechanism-

- Synchrotron sources emit incoherent emissions. These explosive events create shocks which accelerate particles to extremely high energies (10^{19} eV) , and also amplify the ambient magnetic fields. These ultra-relativistic particles spiral around the magnetic field lines, emitting polarized radiation as they lose energy [34]. Examples of such sources include Type I supernovae caused by core collapse of massive stars [59], relativistic jet outflows of kinetic energy and matter from accelerating black holes and neutron star systems [16], etc.

- Coherent bursts are short timescale bursts (often less than a second, going down to a few milliseconds). These are some of the highest energy density events in the universe. While, synchrotron emission has an upper limit of $\sim 10^{12}$ K for brightness temperature [35], extreme coherent bursts can reach up to brightness temperatures of $\sim 10^{35}$K. The incoherent sources are also referred to as 'slow' transients, while the coherent ones as 'fast' [10].

Pulsar emission is the earliest detected form of such coherent burst, which proved the existence of neutron stars [25], laid precise tests of general relativity [27]. Pulsars are rapidly rotating magnetized neutron stars, which are highly periodic in nature with diverse pulse periods . Pulsars with pulse periods of a few milliseconds are called millisecond pulsars, while the ones with pulse periods of a few seconds are known as normal pulsars. The periodicity of the pulsar is correlated to the rotation speed of the neutron star. They are weak radio sources with intensities varying from 5μJy to 1μJy (1Jy $\equiv 10^{-26}$W m^{-2} Hz^{-1})[40]. Similar to the radio pulsars, such periodic coherent radio emission is also observed in Rotating Radio transients (RRATs) and intermittent pulsars [45].

Quantitative estimate of location of the pulsar can be made by measuring the dispersion of the pulses - the delay in pulse arrival times across various frequencies. This delay occurs due to the differential effects of the intervening ionized medium on the group velocity of the pulse across frequencies. So, the pulse emitted at lower frequency travels slower through the intervening medium than those emitted at higher frequencies. Hence, higher the value of dispersion measure, further away is the pulse traveling from. Most of the pulsars detected

till date are galactic with low values of dispersion measure [26].

But, in 2007, Lorimer et al. [37] discovered a single burst coming from an estimated distance of 500 Mpc (much higher than the size of milky way). These single pulses are called Fast Radio Bursts (FRBs). FRBs can be characterized by their short duration (sub-second), broad band pulses with very high dispersion measure values [11]. The high dispersion measure values are consistent with their extragalactic origins. Since their discovery, there have been global efforts to study FRBs, which has led to an increase in the number of known FRBs. The current published sample exceeds 700 unique sources. About $\sim 4\%$ of this population are repeaters, meaning the detected pulse has been observed before. But many of these repeaters have been seen twice. Unlike pulsars, there hasn't been any clear periodicity observed for repeating FRBs, suggesting that the one-off sources may produce more pulses, given enough follow-up. By studying the large sample of known FRBs [3], people are starting to find various trends, suggesting subpopulations based on burst morphology and spectra. But most of the results are still not robust, and need further work. Some initially considered one-off FRBs to be just less active repeating FRBs, but in 2021, Pleunis et al. [50] analyzed bursts from the first FRB catalog [3] and found repeaters to have longer durations in time and narrower bandwidths compared to one-offs. This suggests that there are likely at least two distinct types of FRBs, rather than one-offs being simply less active repeaters. But, it is still unknown if this demonstrates different sources for repeaters and one-offs, or just different emission mechanisms from the same type of sources. As the number of known FRBs will increase, further trends and sub-populations will emerge, making it easier to validate these theories. Periodicity of the sub-bursts is also an important factor to estimate the emission mechanism for repeater FRBs. Till now, only FRB 20191221A [3] has shown a strict periodicity of 216.8 ms between sub-bursts, making periodicity seem to be a rare event and raising the suspicion of different origin compared to other known FRBs. The large number of sub-bursts per unit time means that the central energy reservoir is not exhausted easily. Studying the individual burst energies and average spectra of repeaters, it is noticed that their activity and properties are change with time, raising the need for more follow-ups. Most results have been contradictory to previous observations. It has been seen that repeaters can produce multiple types of bursts, some of which appear to be more similar to one-off FRBs [33]. Identification of new sources and following up already known sources is key step in answering these questions. Overall, having larger number of known sources, and burst characteristics for individual sources allows for meaningful population studies and robust results [28]. Apart from this, having larger samples aid in applying FRBs

to broader problems in astrophysics and cosmology. By localizing FRBs to (sub-)arcsecond precision and associating them with some host galaxy, we can gain information about the local conditions of the source. For example- some precisely localized FRBs have been seen to be present in star-forming regions of the galaxy (example- Marcote et al. 2020[43]). But, there have been contradictory situations where FRB source has been detected $\sim 200 - 250$ pc from the nearest star forming regions [56]. Yet again, some non-repeaters have been found in regions of really low star formation rate and outskirts of galaxies [23][42]. But, none of these observations are robust as the number of observations will increase, new trends may start to appear. But currently, from the present data, it is hypothesized that FRBs maybe magnetars formed via a variety of channels such as binary mergers, core collapse supernova, accretion-induced collapse, etc [44]. Magnetars are a type of neutron star with extremely powerful magnetic field ($\sim 10^{13} - 10^{15}$G)[29]. Apart from this, by acquisition of the full polarization data for the FRBs, we can gain more information regarding the magnetic fields at the source, intervening plasma. Cho et al. (2020)[9] found a one-off FRB to have complicated polarimetric properties changing in time. Day et al. (2020) [12] studied the polarimetric properties of five ASKAP detections and found repeaters to completely linearly polarized, whereas one-off sources to be more heterogeneous in their polarization properties. But, there have been contrasting results of repeaters having swings in polarization angles [39]. Even though, some of the observed circular polarization may have been converted from linear due to propagation effects, it seems that the dominant factor is the emission mechanism itself. As seen above, most of theories regarding FRBs are not robust and require analysis of bigger datasets to figure out proper trends and generalized results. FRB signals have been uniformly distributed across the sky. However, their arrival times are inherently unpredictable, making them challenging to detect from a few hours of observation for a random patch of sky. This raises the requirement for a real-time search of FRB over a certain field of view. Other than all this degeneracy, FRB signals are narrow-band (detected over shorter frequency range)[24] but they have been detected over large range of frequencies (110 MHz to 8 Ghz). This raises the requirement for searching over broad bandwidth.

My project SPOTLIGHT revolves around developing such a real-time FRB detection pipeline for uGMRT. SPOTLIGHT is a time-domain survey instrument to perform a real-time commensal search for Fast Radio Bursts (FRBs) and Pulsars with a PetaFlop system installed at the GMRT funded under the National Supercomputing Mission (NSM). This system is capable of executing real-time High Performance Computing and AI applications to ensure simultaneous time-domain detection and arc-second imaging. It uses the full GMRT

bandwidth of 300 MHz - 1460 MHz to carry out detections. This detection system searches for millisecond bursts and periodic signals up to a dispersion measure (DM) limit of 2000 pc cm^{-3} . To ensure search over a large field of view, SPOTLIGHT uses a multi-beam (2000 beams) search spanning a FoV of around ~ 2 degrees. The SPOTLIGHT system in its full usage is estimated to detect ~ 270 FRBs over 3 years (calculated based on CHIME's FRB event rate and GMRT sky coverage). Using the total GMRT sky coverage, the SPOTLIGHT system will be able to cover a patch of the southern sky, which no other telescopes are able to cover.

Since, this is a large scale project, I have worked on a few aspects of it focusing on-

- Developing a real-time Radio Frequency Interference removal algorithm to filter the data from anomalous radio signals from diverse terrestrial and cosmic sources. Filtering out actual signals from RFI is taking out needle from a hay stake, and requires robust statistical algorithms and high computation speed.
- Developing a pipeline to calculate the flux density of candidates using the radiometric equation (fine-tuned for GMRT). This is useful in case the actual data is corrupt. The data from the antenna undergo a lot of processing, and the detection pipeline for SPOTLIGHT takes in real intensity values. Upon detection of a candidate, the required data gets dumped to disk and calculation of important burst properties like flux density take place. In case, the dumped data is corrupted, these burst properties can be estimated using the radiometer equation. This makes this pipeline useful for both real-time and offline searches.
- Developing a clustering code for removal of unwanted and redundant candidates from actual list of detected candidates.
- Developing a quasi-realtime pipeline for detection of FRBs.
- Developing a real-time code for candidate detection including clustering, candidate feature extraction, and candidate classification.
- Developing a coincidence/anti-coincidence filtering algorithm to remove RFI across multiple beams and also detect the same candidate observed in nearby beams.

All these aspects form significant components of the broader SPOTLIGHT project. To create a more streamlined and connected narrative, I have dedicated the first chapter to providing

a brief introduction to the various sub-parts of SPOTLIGHT. The subsequent chapters delve deeper into each topic, outlining the work I have undertaken in detail.

Chapter 1

An Introduction to uGMRT and SPOTLIGHT

1.1 Upgraded Giant Metrewave Radio Telescope

The whole project is centered around building a detection pipeline for GMRT. So, it is only fair to start with a short introduction to GMRT. The Giant Metrewave Radio Telescope is one of the most largest and most sensitive low frequency radio interferometric arrays in the world. The array consists of 30 antennas (each of 45 m diameter) and spanning over 25 Km. GMRT can operate in three modes:

- **Regular earth-rotation Aperture Synthesis mode** In this mode, GMRT operates as a radio interferometer using its 30 antennas to create a large effective aperture. Each antenna collects raw voltage data. Signals from each antenna pair undergo cross-correlation in the GMRT software correlator to generate visibility data. This visibility data is later converted into images using Fourier techniques. The large collection area improves sensitivity, which allow for deeper observations. This mode is generally used for imaging galaxies, AGNs, 21 cm observations, etc.
- **Incoherent Array Beamforming (IA mode) and Phased Array Beamforming (PA mode)** Due to the variation in the positions of the antennas, the same signal is different antennas of the same array with certain delay. Normally, signals from an

n-element phased array are combined by summing the voltage signals from individual antennas, applying appropriate delay and phase compensation.. Considering the elements to be identical, this phased array provides a sensitivity which is n times the sensitivity of a single antenna. The beam produced by a phased array is narrower than that of a single element. This narrowing occurs because the array combines voltage signals from different elements, each with a specific phase adjustment, forming a more focused beam pattern. In Incoherent array, the voltage signals from each antenna are added without any phase correction. The resulting telescope beam retains the shape of a single element, as the phase information from individual elements is lost during the detection process. As a result, the beam width (hence the Field of View) is much more for IA, as compared to PA mode. The sensitivity to a point source for incoherent array is $\sqrt{n}$ times than that for a single antenna. In transient search, where detection of new pulsars and FRBs don't require higher sensitivies and rather need wide field of view, incoherent phased array beamforming is a better choice. But for following up on known pulsars and FRBs (whose locations are already known), it is optimal to run phased array beamforming with higher sensitivity.

The upgraded GMRT consists of four working bands - 120-250 MHz (Band 2), 250-500 MHz (Band 3), 550-850 MHz (Band 4), 1050-1450 MHz (Band 5). There are gaps in this frequency range which is due to presence of very strong radio frequency interference (RFI) such as TV channels, mobile communication bands, FM band, etc. [21]

1.2 The SPOTLIGHT system

The emergence of modern astronomical facilities such as the uGMRT, eVLA, LOFAR, MWA, MeerKAT, ASKAP, CHIME and the construction of the international mega-science project, the Square Kilometre Array (SKA), are driving the growth and technology of next-generation radio astronomical instrumentation. GMRT is recognized as a pathfinder facility for SKA and is currently one of the few interferometers with a commensal survey backend.

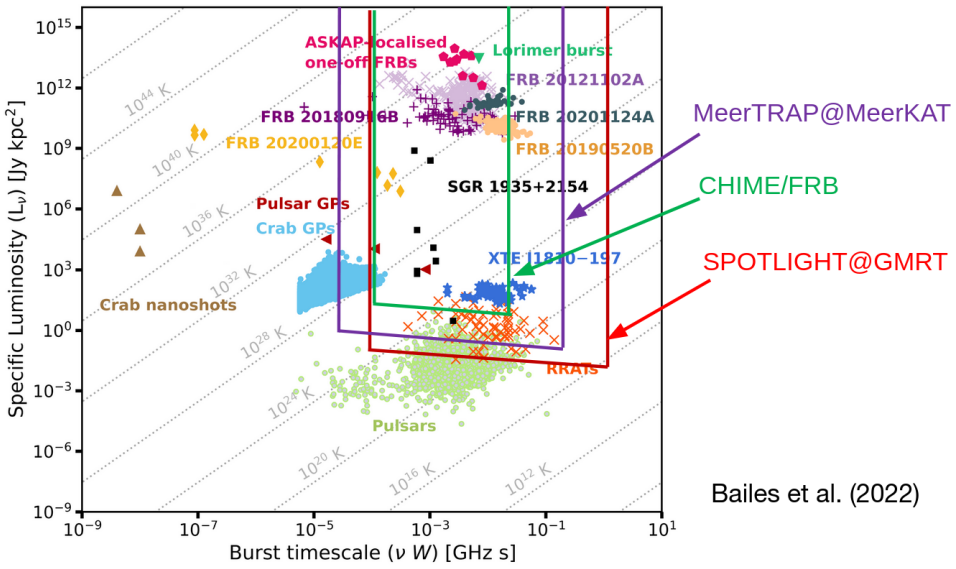

Figure 1.1: Specific luminosity variation across burst timescale for fast radio transient population in the coherent radio emission regime. The SPOTLIGHT survey coverage is shown in comparison to CHIME and MeerTRAP [5]

The above figure depicts the transient sky landscape, showcasing coherent emitters on a luminosity versus pulse timescale diagram. It can be seen that SPOTLIGHT covers a much wider and deeper search compared to the other large-scale surveys. But yet, it can be seen in the figure that there are many gaps in the parameter space. With the recent emergence of long period transients and ultra-long period galactic magnetars [6], it is a necessity to expand the parameter space in order to better estimate the FRB progenitor models. The SPOTLIGHT detection pipeline aims to populate the parameter space more uniformly, resulting in more robust science results.

The system has a PetaFlop computing capacity (hosting $\sim$ 90 A100 GPUs installed on 60 of C-DAC's indigenously developed Rudra servers, several 10s of TB of memory and 2 PB of storage) for carrying out real-time commensal search. The SPOTLIGHT pipeline will run parallel to regular GMRT observations and will search for FRBs and pulsars within the full-width-half-maximum (FWHM) of the field-of-view (FoV) of the primary beam. Whenever, the GMRT antennas are used for some observation, the SPOTLIGHT system piggybacks onto the ongoing observation and searches for transients in that location of the sky.

1.2.1 Pipeline components

The SPOTLIGHT pipeline consists of multiple sub-parts focusing on various aspects like beam-forming, candidate detection, candidate imaging, post-processing, etc. In order to understand the detection pipeline that I have worked on, it is better to have an idea of the functionalities of the overall cluster. The cluster consists of 60 Rudra servers. These servers are split into three sub-clusters.

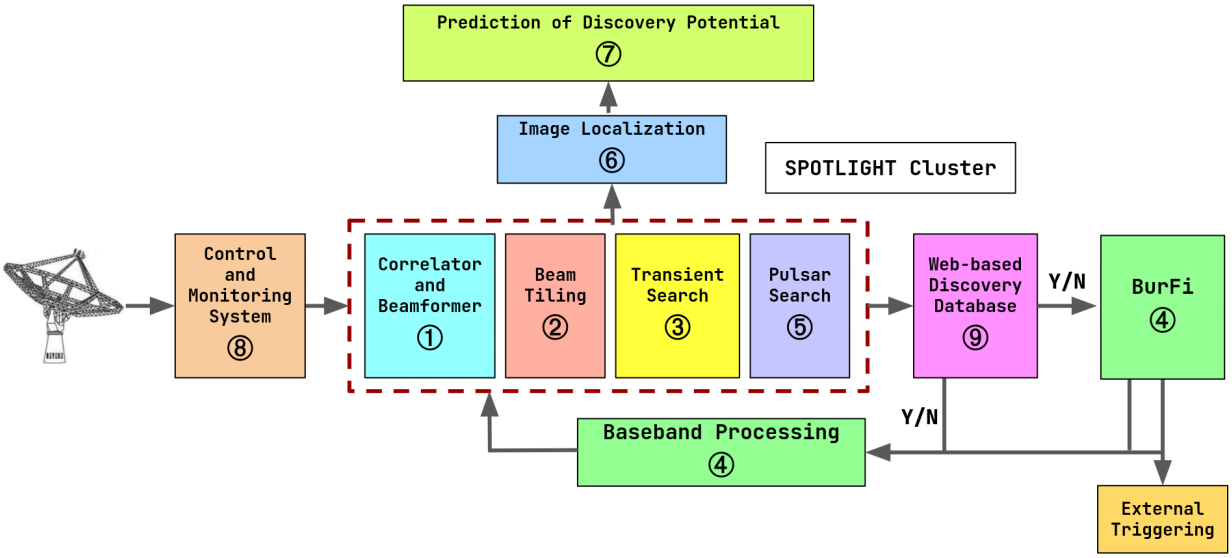

Figure 1.2: The SPOTLIGHT pipeline

The above figure depicts the various components of the SPOTLIGHT cluster, which are described in detail below-

1. **Correlation and Beamforming:** The first sub-cluster, which consists of 16 servers, performs correlation and beamforming of 2000 beams at 1.3 millisecond time-resolution. Correlation is an important step for interferometric imaging, during which signals from multiple antennas are combined to estimate spatial information about the sky. During this step, the time varying voltage data from every pair of antennas is cross-multiplied to compute visibility data. The visibility data represents Fourier components of the sky brightness distribution. By applying a Fourier transform, the visibility data is converted into sky intensity maps. So, the raw voltage data is finally converted to intensity data with integration times of 1.3 ms. After correlation, the data from multiple

antennas are summed together to form multiple beams using Phased Array (PA) or Incoherent Array (IA) mode as discussed in 1.2, enhancing the sensitivity and field of view. The fully functional beamformer of SPOTLIGHT will generate 2000 beams in real-time.

2. **Beam Tilling:** Beam tilling refers to the technique of arranging multiple beams in a structured patten to cover a larger field of view efficiently. In the SPOTLIGHT cluster, there are two different modes for beam tilling based on the observation type.

   - **Survey mode:** This covers a broad FoV compared to the primary beam-width. There is a 50 % overlap between the beams to maintain sensitivity.
   - **Targeted mode:** This mode has a smaller FoV, but the beam overlap is optimized to enhance sensitivity. Suitable for observing known targets.

   One crucial part of beam tilling is clustering together simulated beams in close proximity. This is really important for Coincidence filtering (for eliminating false positives), which will be discussed later. The beam tilling software effectively covers a larger fraction of the FoV while maintaining high and uniform sensitivity across the entire FoV. This ensures that any faint FRB events occurring across the field of view are not missed. [46]

3. **Transient Search Pipeline:** The data from the first sub-cluster is sent to the second sub-cluster, consisting of 24 servers, for the multi-beam transient search. the transient detection pipeline consists of eight crucial steps, and is carried out by six different software-

   - **RFI mitigation tool:** Radio frequency interference mitigation is an important step in detecting transients. These interferences come from multiple sources like TV channels, FM radios, Radars, jets, etc, and if not removed makes it difficult to detect actual signals. I have developed the current version of the RFI mitigation tool and this topic will be discussed in detail in later chapter.
   - **AstroAccelerate:** The AstroAccelerate package [8] carries out heavy GPU computation algorithms for -
     (a) **Dedispersion:** As discussed before, when the FRB signal travels through plasma present along the line of sight, the different frequencies travel at different speeds, causing the dispersion of signal. Higher frequencies reach faster

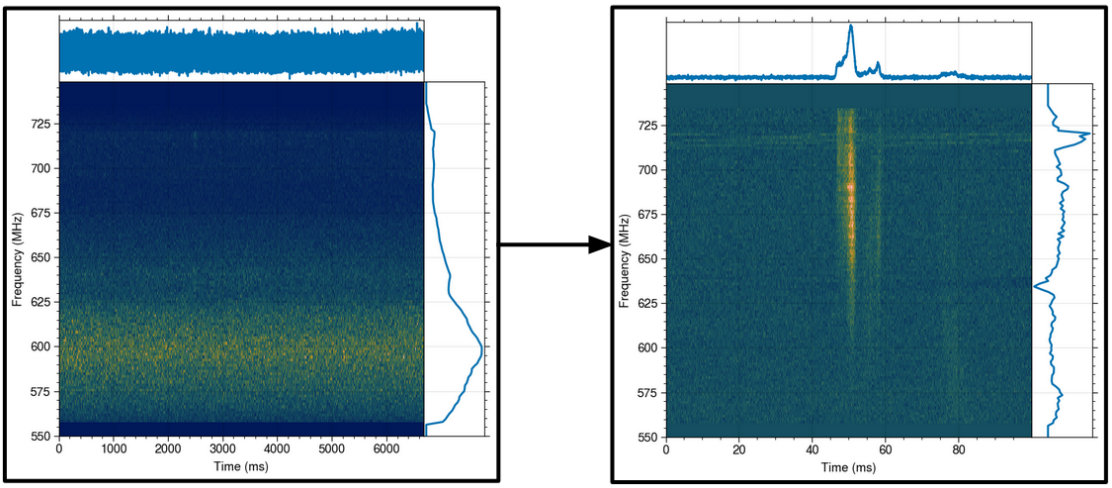

Figure 1.3: Left top: Dispersed time-series; Left bottom: Dispersed dynamic spectrum; Right Top: Time-series after dedispersion at the correct DM, with the pulse visible; Right bottom: Dedispersed dynamic spectrum with the pulse located between 40-50s

than lower frequencies and this delay is captured by the following equation:

$$\delta\tau = DM \times C_{DM}(\frac{1}{f_{low}^2} - \frac{1}{f_{high}^2})$$

where $C_{DM} = 4148.8 \times 10^3 MHz^2 pc^{-1} cm^3 s$. The DM is defined as the dispersion measure and is calculated as the integral of electron column density(n_e) along the line of sight.

$$DM = \int_0^{D_z} \frac{n_e(l)}{1+z(l)} dl$$

where z is the redshift of the source. The dispersed data needs to be dedispersed in order to retrieve the actual signal. But since, the DM by which the signal is dispersed is not known beforehand, dedispersion needs to be done across many trail DM values (from 100-2000). This process involves integrating the frequency-time intensity data, known as the dynamic spectrum, into a time series while accounting for the frequency-dependent delays corresponding to each DM value. [1]

(b) **Single pulse Search:** In the previous step, multiple time series for data dedispersed at multiple DMs were created. Each time series is searched for

single pulses by match filtering the signal with boxcars of varying widths. The main idea behind match filtering is when the width of the trial boxcar matches the width of the actual dedispersed pulse, the signl-to-noise ratio (SNR) of the signal is amplified and the noise is suppressed. A range of boxcar widths are trailed for each time-series. When the SNR of the pulse is above a certain threshold (usually 7σ), that pulse is considered as an actual signal by the searching tool.

In the end, astro-accelerate provides us with a long list of candidates, with their possible DM, time of arrival, width, and SNR.

- **Clustering:** The number of candidates generated by AstroAccelerate can range from a few hundreds to millions, most of which are spurious candidates due to RFI or redundant candidates. Therefore, we need to cluster candidates in both the DM-time plane and the RA-Dec plane across multiple beams, significantly reducing the number of candidates by several orders of magnitude. This topic will be discussed in much more detail in later chapter.
- **Feature extraction and classification:** The list of candidates obtained after clustering undergo feature extraction using our software Candies. The two main features are the dynamic spectrum and the DM transform (DMT). These characteristic features can be used to distinguish between FRBs and RFI. The GPU based code extracts this features from the beam data, and passes them onto the CNN classifier FETCH [2]. FETCH is a binary classifier which classifies each candidate as an FRB or RFI. I will discuss these tools in more detail in 5.

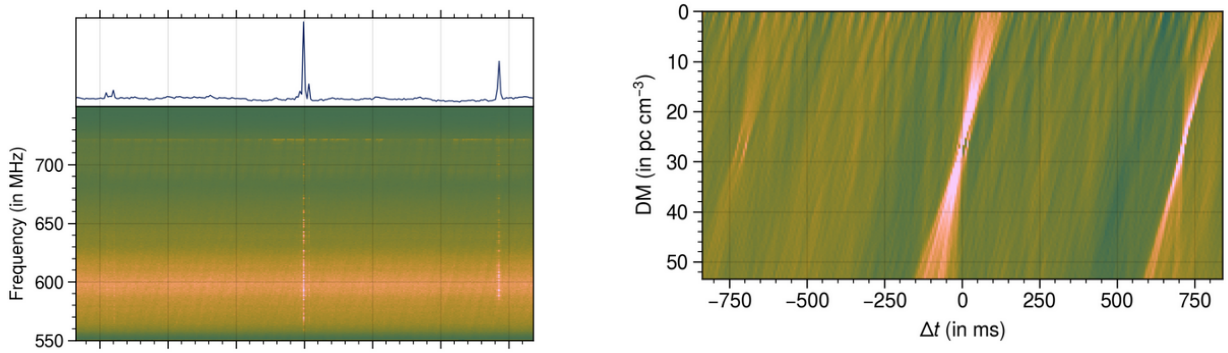

Figure 1.4: left: dedispersed dynamic spectrum with a few pulses; right: DM transform plot for the same pulse

- **Coincidence and anti-coincidence filtering:** Sometimes, we may observe the

same pulse with same DM and time of arrival in very far off beams (¿ 10-15 arc minutes). But, this is not physically possible as the pulse has originated from the same point source in sky. The main purpose of anti-coincidence filtering is to detect these kind of RFI and remove them. A pulse of certain SNR, if observed in the central beam will also be observed in nearby beams with similar DM and time and arrival. Considering all these as different candidates is redundant and computationally expensive. The main purpose of the coincidence filter is to remove these redundant candidates. This topic will be discussed in more detail in future chapter.

4. **Baseband processing:** For every detection, the detection pipeline (discussed above) will dump both beamformed data (with resolution $\sim 1.3ms$) and the Nyquist sampled baseband data (with resolution of a few nanoseconds). This data is utilized by the BurFi(our algorithm) (burst fitting) pipeline to better estimate various burst parameters like burst morphology, spectral characteristics (like emission bandwidth and peak emission frequency), drift rate, scattering, etc. Drift rate analysis with BurFi at low resolution investigates the frequency evolution of bursts, offering insights into the dynamics of the emitting region. This analysis helps distinguish between intrinsic spectral features and propagation effects caused by turbulent media. Collectively, these low-resolution analyses by BurFi provide a comprehensive understanding of FRB characteristics and play a crucial role in designing follow-up strategies by identifying potential repeater-like behavior. this pipeline is triggered only when a candidate is detected.

5. **Image localization pipeline:** Radio surveys with a wide field of view can detect a large number of FRBs; however, their poor localization accuracy makes it challenging to reliably associate FRBs with their host galaxies. As highlighted in [15], achieving a probability of chance coincidence below 1% requires an FRB to be localized within 0.5 arcseconds, given the density of galaxies in the field of view at typical FRB distances. The lack of precise host galaxy associations hinders efforts to model progenitors, analyze propagation effects, and explore potential cosmological applications. In contrast, commensal surveys conducted with interferometric arrays detect fewer FRBs but offer significantly improved localization accuracy. precise localization of detected FRBs at a later time is only possible for repeaters and not for one-offs. In order to accurately localize signals immediately (after a short pipeline delay), SPOTLIGHT has

incorporated this real-time imaging pipeline which gets triggered when a candidate is detected.

6. **Monitoring system:** The monitoring tool is a web interface which displays various properties of the system in real time to a remote observer. The interface also stores all the detected candidate lists with their characteristics.

Chapter 2

An Algorithm for Flux Calculation of GMRT Bursts using the Radiometer Equation

As discussed in the previous chapter, upon detection of a candidate by the detection pipeline, part of data (both beamformed and baseband) containing the burst is dumped to disk for further processing and analysis. But, there is always the possibility of the data being corrupt, or wrong data being dumped due to internal machine delays. In order to bypass this issue, if it arises, I have written an algorithm to calculate the flux density of a burst using the radiometer equation (fine-tuned for GMRT). The calculation of a radio burst is really beneficial to estimate its luminosity and energy output. It allows to classify sources and get better idea about the source environment.

2.1 Radiometer Equation

The power received per unit area, solid angle, and frequency from a point source in the sky, by a receiver is called Specific intensity (I_ν) or brightness ($Wm^{-2}Hz^{-1}sr^{-1}$). The fundamentals for this part are described in [4].The radio sources are not point sources and subtend certain solid angle on the receiver. Integrating I_ν over the solid angle subtended by the source gives us the flux density (S_ν) of the source. These sources are really faint with FRBs having flux

densities of few Janskys (1 Jy = $10^{-26}\mathrm{Wm}^{-2}\mathrm{Hz}^{-1}$). The brightness temperature is defined as for such a radio source with intensity (I_ν) is given as-

$$T_b = \frac{I_\nu c^2}{2k_B\nu^2}$$

where:

- c is the speed of light ($\approx 3.0 \times 10^8$ m/s),
- k_B is the Boltzmann constant ($\approx 1.38 \times 10^{-23}$ J/K),
- ν is the frequency (Hz).

The radio antenna absorbs power from incoming radio waves, which is typically expressed in temperature units (Kelvin).. To understand this power, lets assume a resistor in thermal equilibrium at temperature T. It experiences random electron motion, generating fluctuating currents. Although the average current is zero, the power — proportional to the square of the current — is nonzero. According to the equipartition principle, the power per unit frequency in the radio regime is given by the Nyquist formula:

$$P = kT$$

where k is Boltzmann constant. In analogy to this, if a radio antenna receiver has a power P (per unit frequency), then the corresponding temperature is known as antenna temperature (T_a).

$$T_a = \frac{P}{k}$$

This temperature is not the actual temperature of the antenna.

Now, the antenna consists of multiple electronic components which contribute to some noise power. Additionally, there may be extra power received from the sky, picked up from the ground, etc. The temperature corresponding to sum total power is called the system temperature (T_{sys}).

$$T_{\mathrm{sys}} = \frac{\text{Total Power referred to receiver inputs}}{k}$$

The system temperature when not observing a source is the measure of total random noise.

So, it is highly desirable to make the system temperature as low as possible.

$$T_{\rm sys} = T_{\rm rec} + T_{\rm sky} + T_{\rm Atm} + T_{\rm scat} + T_{\rm ground} + \ldots$$

where:

- $T_{\rm rec}$: Receiver temperature,
- $T_{\rm sky}$: Sky temperature (including contributions from the source and the CMB),
- $T_{\rm Atm}$: Contribution from the atmosphere,
- $T_{\rm scat}$: Scattering due to feed legs,
- $T_{\rm ground}$: Pickup from the ground.

Now, considering a single GMRT antenna observing a point source with flux density S_ν, the received power per unit frequency due to the source is given by:

$$P_\nu = k.T_A = \frac{[A_e.S_\nu]}{2}$$

where A_e is the effective area of the dish of the antenna. $A_e = \eta A$, where η is the aperture efficiency (typically $\sim 0.6 - 0.7$)and A is the geometric area.

Rearranging the values, the measured antenna temperature due to a signal is -

$$T_A = [\frac{A_e}{2k}]S_\nu$$

Hence, a larger collecting area would provide more intense signal. The term $A_e/2k$ is called the Antenna gain and it provides information about the increase in system temperature due to presence of a source. So, the temperature corresponding to the signal is the antenna temperature. When an antenna starts observing a source, this antenna temperature is added on to the previous system temperature. During the measurement of this T_{sys}, the RMS noise that is present is given by -

$$\sigma = \sqrt{2}.T_{sys}$$

If we average N such independent measurements of the output power, the RMS noise on the average estimate will be -

$$\sigma_N = \sigma/N^{1/2} = \sqrt{2}.T_{sys}/N^{1/2}$$

For a signal of bandwidth $\Delta\nu$, raw voltage data are separated by time $\Delta\tau \sim \frac{1}{2\Delta\nu}$ are independent of each other. So, the number of independent samples in an integration time Δt is -

$$N = \delta t/\delta\tau = 2.\Delta t.\Delta\nu$$

The RMS noise for the same integration time is given as-

$$\sigma_N = T_{sys}/(\Delta t.\Delta\nu)^{1/2}$$

This is also known as the Radiometer equation for a single antenna.

Now taking the ratio between the signal and noise, we get -

$$S/N = [A_e/(2k.T_{sys})] \times [1/(\Delta t.\Delta\nu)^{1/2}] \times S_\nu = G/T_{sys} \times [1/(\Delta t.\Delta\nu)^{1/2}]$$

So, the source of the flux is given as-

$$S_\nu = S/N \times \frac{T_{\rm sys}}{G \times \sqrt{\Delta t \cdot \Delta\nu}}$$

This equation is valid for a single dish antenna. As discussed before, the sensitivity increases by N for PA mode, and increases by $\sqrt{N}$ for IA mode (where N is the number of antennas). Also, presence of multiple polarizations effects the RMS noise by $1/\sqrt{N_{pol}}$. For a single pulse burst, we consider the integration time to be equal to the effective width of the observed pulse (W_{eff}). Applying all these factors, the flux density for a single pulse as observed by an interferometric array of N antennas, recording N_{pol} data, is given by -

$$S_\nu = \mathrm{S/N} \times \frac{T_{\rm sys}}{G\sqrt{N_{\rm PA}^2 \times N_{\rm pol} \times \Delta\nu \times W_{\rm eff}}} \quad \text{(PA mode)}$$

$$S_\nu = \mathrm{S/N} \times \frac{T_{\rm sys}}{G\sqrt{N_{\rm IA} \times N_{\rm pol} \times \Delta\nu \times W_{\rm eff}}} \quad \text{(IA mode)}$$

2.2 Algorithm

2.2.1 Inputs

In order to calculate the flux density of a single pulse, a few parameters are required as input. First is the band information of the data in which the pulse is observed. The table below provides the list of parameters stored as dictionary for each band -

Band	**BW (MHz)**	**Flow (MHz)**	**Fmid (MHz)**	**Fhigh (MHz)**	**BW Usable (MHz)**	**Ref. Gain**
2	200	125	200	250	50	0.33
3	200	260	400	500	120	0.33
4	400	550	650	850	200	0.33
5	400	980	1260	1500	280	0.22

Table 2.1: Band Information Table

Apart from this, we also require the following inputs -

- RA, Dec of the source (coordinates)
- Number of antennas used
- Number of polarizations observed
- channel resolution - When observing a source in radio frequency, the total bandwidth is divided into smaller channels (few KHZ-MHz). The ratio of the total observing bandwidth with the number of channels is known as the channel resolution. For SPOTLIGHT, the channel resolution is 0.0488 MHz for band 2,3; and 0.0976 MHz for band 4,5. The G/T_{sys} is calculated for each of this channels and integrated to get the final value.
- scattering width and intrinsic width observed for the pulse. The intrinsic width is obtained from the detection pipeline, and the scattering width is approximated using the effects of scatter broadening and is given by [7] -

$$\log(W_{\text{scatt}}) = -6.46 + 0.154 \times \log(\text{DM}) + 1.07 \times \log(\text{DM})^2 - 3.86 \times \log(\nu)$$

where W_{scatt} is in ms, and ν is the representative frequency

- The dispersion measure (DM) for the pulse. This is used to calculate the smearing caused by intra-channel dispersion:

$$W_{\mathrm{DM}} = K_{\mathrm{DM}} \times \frac{\mathrm{DM} \times \Delta\nu}{\nu^3}$$

2.2.2 Steps of calculation

1. The RA, Dec coordinates are used to estimate the sky temperature (T_{sky}) using the all sky temperature dataset. [22]

2. The G/T_{sys} for each frequency value is defined using polynomial functions as a function of frequency for each band as -

$$S(\nu) = a_0 + a_1\nu + a_2\nu^2 + a_3\nu^3 + \cdots + a_n\nu^n$$

where ν represents the frequency in MHz. The plots for the same are given in figure 2.1

Band	**Frequency Range (MHz)**	a_0	a_1	a_2	a_3	a_4	a_5
2	125 - 250	-0.0274	0.000653	-5.75×10^{-6}	2.26×10^{-8}	-3.30×10^{-11}	—
3	260 - 500	-3.9427	0.06092	-3.88×10^{-4}	1.31×10^{-6}	-2.46×10^{-9}	2.44×10^{-12}
4	550 - 850	-60.966	0.5298	-1.91×10^{-3}	3.67×10^{-6}	-3.94×10^{-9}	2.25×10^{-12}
5	980 - 1500	-57.791	0.2832	-5.77×10^{-4}	6.25×10^{-7}	-3.80×10^{-10}	1.23×10^{-13}

Table 2.2: Polynomial Coefficients for Sensitivity Functions

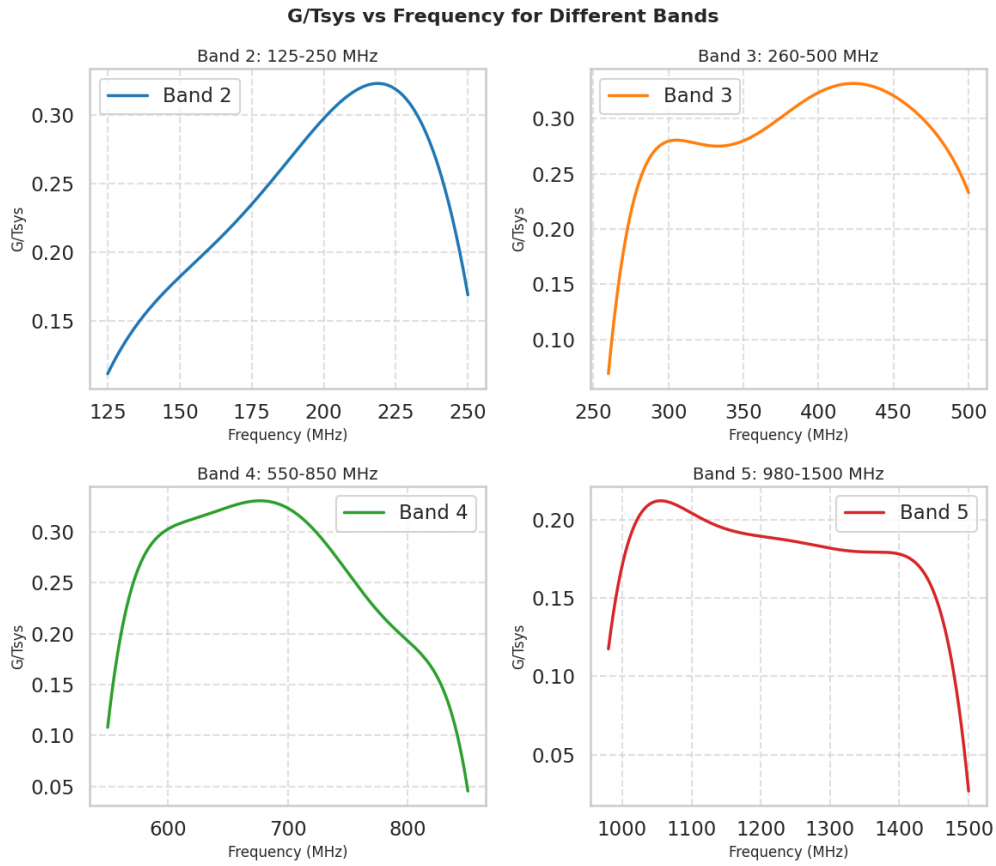

Figure 2.1: Polynomial functions for sensitivity

3. Now, we define SEFD = T_{sys}/G which is defined for a particular frequency.

4. Next, we calculate sumSEFD as the summation $(\text{SEFD})^2$ with a step size of 1 from starting frequency to ending frequency of the pulse. For better approximation, we can use the channel resolution as the step size.

$$\text{sumSEFD} = \sqrt{(\text{SEFD}_1)^2 + (\text{SEFD}_2)^2 + \cdots + (\text{SEFD}_n)^2}$$

5. Next, we calculate the average SEFD as the rms average of sumSEFD.

$$\text{avg SEFD} = \sqrt{\frac{(\text{SEFD}_1)^2 + (\text{SEFD}_2)^2 + \cdots + (\text{SEFD}_n)^2}{N}}$$

where N is the total frequency range of the pulse.

6. The average SEFD can replace the G/T_{sys} in the flux calculation equation, giving the

final form for the IA beam to be:

$$S_\nu = \mathrm{S/N} \times \frac{1}{\text{avg SEFD} \times \sqrt{N_{\mathrm{IA}} \times N_{\mathrm{pol}} \times \Delta\nu \times W_{\mathrm{eff}}}}$$

$$= \mathrm{S/N} \times \frac{sumSEFD}{\sqrt{\Delta\nu} \times \sqrt{N_{\mathrm{IA}} \times N_{\mathrm{pol}} \times \Delta\nu \times W_{\mathrm{eff}}}}$$

$$= \mathrm{S/N} \times \frac{sumSEFD}{\Delta\nu \times \sqrt{N_{\mathrm{IA}} \times N_{\mathrm{pol}} \times W_{\mathrm{eff}}}}$$

Similarly for the PA beam:

$$S_\nu = \mathrm{S/N} \times \frac{sumSEFD}{\Delta\nu \times \sqrt{N_{\mathrm{PA}}^2 \times N_{\mathrm{pol}} \times W_{\mathrm{eff}}}}$$

2.3 Results and Discussion

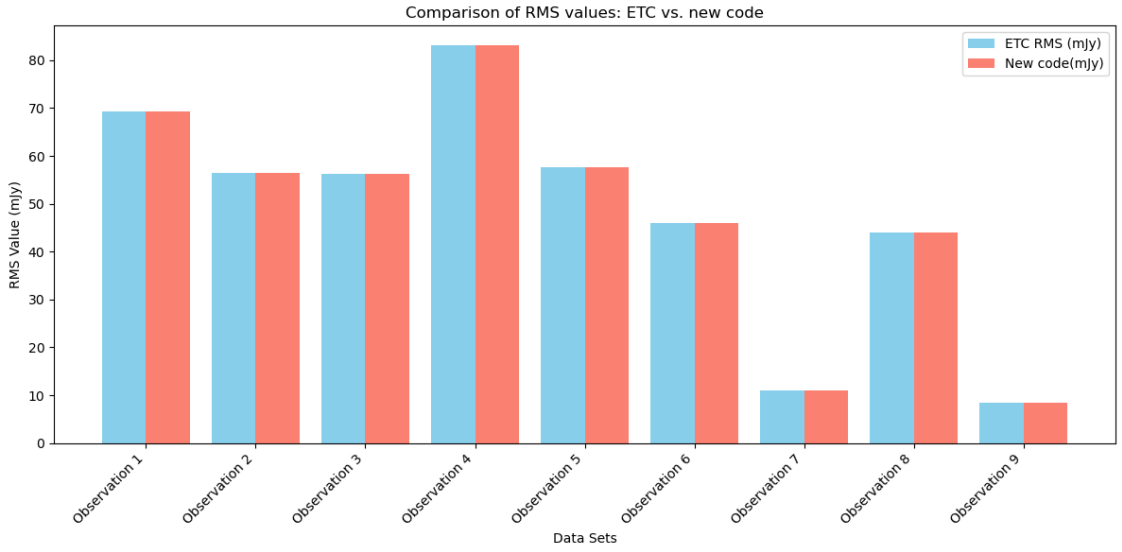

Figure 2.2: Comparison of RMS values

The flux of the source is calculated by taking a product of the signal-to-noise (SNR) of the single pulse and the RMS noise value. The SNR value of the signal is returned by the detection pipeline and the RMS can be calculated using the described algorithm. National

Centre for Radio Astrophysics (NCRA) follows a previous algorithm for calculating the RMS of sources, called Exposure Time Calculator (ETC) [30] but the there are few issues with the same. Firstly, it is a website and cannot be implemented for real-time processing and is only applicable for offline processing. Further, it can't calculate the RMS for FRBs to higher precision as it uses the full-bandwidth of the frequency band for the calculation of G/T_{sys} while the current algorithm uses the width of the single pulse for estimation of the G/T_{sys} giving better results (if required). Below plot shows the comparison of RMS values estimated by ETC and my code for five different FRB observations (recorded before). It can be seen in figure 2.2, both the algorithms estimated similar values for the RMS, making the above code applicable for FRB pulse flux estimation in the real-time pipeline.

Chapter 3

A RFI mitigation algorithm

In radio astronomy, the received signals of interest are typically extremely weak, with signal-to-noise ratios (SNR) often around -30 dB and, in extreme cases, as low as -60 dB. This makes them highly vulnerable to interference from various terrestrial sources. The presence of Radio Frequency Interference (RFI) poses a significant challenge for the sensitive radio antennas used in ground-based radio telescopes. RFI introduces unwanted artifacts into astronomical data, reducing the telescope's sensitivity and generating numerous false positives. The primary sources of RFI are man-made radio emissions stemming from various activities, including radar operations, communication and radiolocation systems such as mobile phones, and broadcasting services like television signals and FM radio bands. Mitigating RFI is crucial to preserving the integrity of astronomical observations and ensuring the accurate detection of faint cosmic signals.

RFI can be broadly classified into two main types based on its impact on the spectrum of the received signal:

- **Broadband RFI:** This type of RFI manifests as a noise-like signal, elevating the power level across a wide range of frequency channels for a brief period. It is typically caused by temporally impulsive events such as automobile ignitions, switching of inductive loads, and electrical discharges like sparking and corona effects in high-voltage transmission lines. These events are usually unintentional and lack periodicity. In Figure 3.1, the vertical bright lines between time bins 20000 and 25000 illustrate broadband RFI, spreading over thousands of frequency channels.

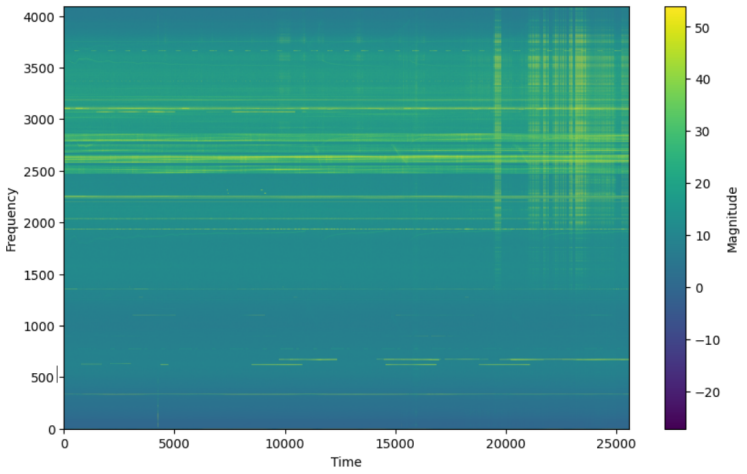

Figure 3.1: Dynamic spectrum containing broadband and narrowband RFI

- **Narrowband RFI:** appear as unwanted signals confined to discrete frequencies or narrow bands, affecting only a small portion of the receiver's bandwidth. It often persists across the entire observation period and is commonly caused by intentional transmissions from communication systems, such as TV and radio broadcasts, short-range radio services, and signals from aircraft and satellites. In Figure 3.1, the horizontal lines affecting channels between 2500 and 3500 are examples of narrowband RFI.

In time domain astrophysics, any signal effected by RFI can be represnted as:

$$x(t) = x_{src}(t) + x_{noise}(t) + x_{RFI}(t) \tag{3.1}$$

where, $x_{src}(t)$ is the contribution of the desired source, $x_{noise}(t)$ is the system noise (contribution of background sky noise and receiver noise), and $x_{RFI}(t)$ is the contribution of the RFI. Both the noise and source signal voltage data follow a zero mean Gaussian distribution, hence the intensity data follows a chi-square distribution with two degrees of freedom. But the RFI signal usually follows a non-Gaussian distribution [19]. These astronomical signals being really faint can get lost in the sea of RFI, and without proper removal of the RFI, finding transient signals is equivalent to finding needle in a haystack. Over the years, several methods have been implemented for RFI removal, but yet there is no equivalent method RFI removal. The techniques for RFI removal strongly depend on location of the telescope, type of observation, etc. So, it is not necessary the best RFI mitigation technique for one telescope will workout for a different telescope too.

3.1 RFI mitigation techniques

Since, most of the terrestrial RFI is human-borne or due to other communication services, the radio telescopes are built in regions with low population density and far from industrial developments. The majority of extra-terrestrial RFI arises from radio propagation through the ionosphere and troposcatter from distant transmitters [55].

The RFI mitigation techniques can be broadly classified into two ways, as shown the flowchart below:

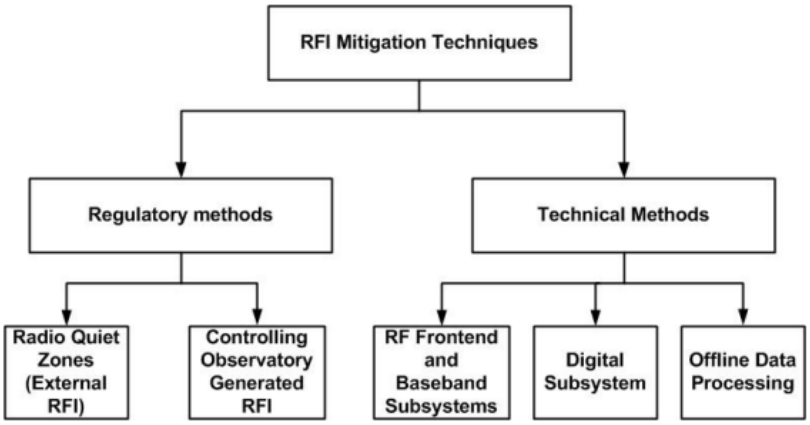

Figure 3.2: Various RFI mitigation methods[17]

Regulatory methods include physical precautionary methods to reduce radio pollution by establishing a radio quiet zone around the radio telescopes. Further, observatory generated RFI are also reduced by incorporating proper shielding of RF components, electrical equipments, and digital devices. To reduce leakage from working labs, Faraday cages are build around them to shield leakage of radio interference. But these methods are not enough as mostly these precautions are bypassed and also due to urbanization, the previously radio quiet zones start producing more and more RFI. Hence, several technical methods are implemented.

Technical methods are mainly used to counter the loss of desired candidates among RFI. Different types of observations require different mitigation techniques. In the case of transient search, periodic broadband RFI. Regardless of the type of RFI, mitigation is applied at various stages of the signal processing chain. At the electronic/receiver level, it is

addressed using analog techniques, while at the digital level, additional mitigation strategies (both in real-time and offline processing) are employed to further reduce interference. In order to mitigate RFI at the analog level, several measures like proper antenna patterns for low amount of side lobes, modified receivers for gain compression of RFI, notch filters to reject known communication bands, etc are applied. [17]

In the digital domain, High Performance Computation (HPC) through GPUs can be applied to mitigate temporal, spectral and spatial RFI in real-time. Real-time RFI mitigation techniques can be classified into three types, excision, cancellation, anti-coincidence [17]. **Excision** includes detecting the RFI from the desired data and removing that by either nulling or clipping the data corrupted by the RFI in both temporal and spectral domain. Excision in time domain is generally done using various robust estimates of variance in the data [18].Thresholding, based on various statistical moments of the data, is used to identify points with exceptionally high intensity values, often indicative of RFI. A more robust method for detecting these outliers is the Median Absolute Deviation (MAD). Once RFI points are identified, they can be mitigated by replacing them with appropriate values, such as zero, the median, or random zero mean Gaussian noise with small variance, depending on the specific requirements. Similar techniques are also applied in the spectral domain for better mitigation. Further, higher moments of variables like Spectral Kurtosis (4th moment) are also sometimes used as they can show significant deviation from predictable distribution for Gaussian variables in the presence of RFI. But calculation of these parameters in real-time come with higher computational costs [20]. There are many other tools like zero dming [14], z-dot filtering [52], etc which we will look into later.

Spatial filtering helps in removing known RFI sources by forming a null in the beam pattern towards the RFI source. This is useful for interferometric arrays like GMRT and specifically our multi-beam system SPOTLIGHT with steerable beams. So, known RFI sources of this sort can be mitigated during beamforming itself. **RFI cancellation**, on the other hand, subtracts the interference signal from the astronomical data. It works best when a strong reference signal or an accurate model of the RFI is available. The process involves detecting and estimating the RFI, synthesizing its noise-free version, and subtracting it from the corrupted data. Often, a separate antenna records the interfering signal, which is then fed into an adaptive filter to optimize its coefficients. Unlike excision, cancellation can effectively remove low-level RFI and, if applied before correlation, can recover most of the affected data [13]. Anti-coincidence approach to RFI filtering is a multi-beam level technique and will be

discussed in chapter 6 .

Finally, **offline processing** of the data is really helpful for properly studying the properties of the astronomical signal. The data itself can be visually inspected and necessary channels can be flagged to improve the SNR of the desired candidates. Offline pipeline can also integrate more computational expensive but robust techniques involving the study of the Fourier space properties of the channels [42]. But, in a search pipeline, it is necessary to have a real-time pipeline with high recall(sensitivity) value, focusing on balancing between increasing true positives and reducing false positives

3.2 Steps of RFI Mitigation

The RFI mitigation algorithm I am building is a multi-staged process, where each step is focussed on removing a particular type of RFI.

3.2.1 Bandpass correction

In radio astronomy, observations are carried out over a range of frequencies — for example, 300 MHz to 500 MHz in Band 3. Our pipeline receives data from different frequencies, captured by the telescope antennas. But the receiver system may respond differently to different frequencies. The bandpass of an observation refers to how the telescope (receivers) respond to different frequencies. In ideal scenario, the bandpass should be flat/uniform across all frequencies. But, in reality it is often non-uniform due to varying amplifier gains across frequencies, instrumental noise, RFI, ionospheric effects, etc. In figure 3.3, it can be seen that the bandpass is not uniform and the mean value is fluctuating across channels. So, the observed intensity signal can be described as:

$$S(f,t) = A(f).I(f,t) + N(f,t) \tag{3.2}$$

where A(f) is the bandpass function, N(f,t) is the noise term, and I(f,t) is the astrophysical signal

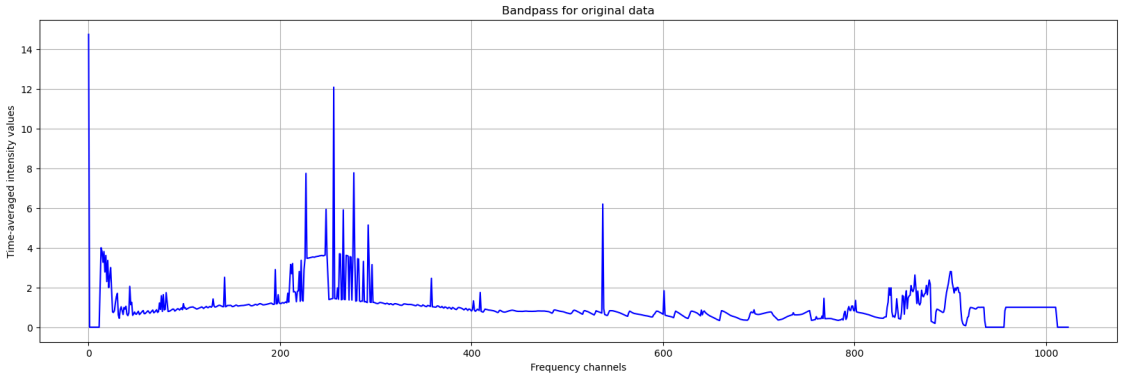

Figure 3.3: Bandpass for unfiltered intensity data captured through 4096 frequency channels(down sampled to 1024 channels)

In order to smooth this bandpass, we first estimate the bandpass by averaging the intensity over time for each frequency channel.

$$B(f) = \langle S(f,t) \rangle_t = \frac{1}{N_t} \sum_{t=1}^{N_t} S(f,t) \tag{3.3}$$

where N(t) is the number of time-step, and S(f,t) is the intensity value at a certain time and frequency. In our real-time pipeline, the data streams in buffers of ∼100s, so the algorithm calculates the bandpass for each channel for every buffer that is processed. After, calculating, the bandpass, the streaming data is normalized using the bandpass. So, each intensity value of the ith frequency channel is normalized by dividing by B(i)

$$S_{corrected}(f,t) = \frac{S(f,t)}{B(f,t)} \tag{3.4}$$

In Figure 3.4, the bandpass appears smoother than before. It can be seen previously there were fluctuations on very high scales, but after normalization, the maximum fluctuation between frequencies is less than 1. But the channels at both ends are generally not reliable for bandpass smoothing due to anomalous responses and the inherent unpredictability of the data.

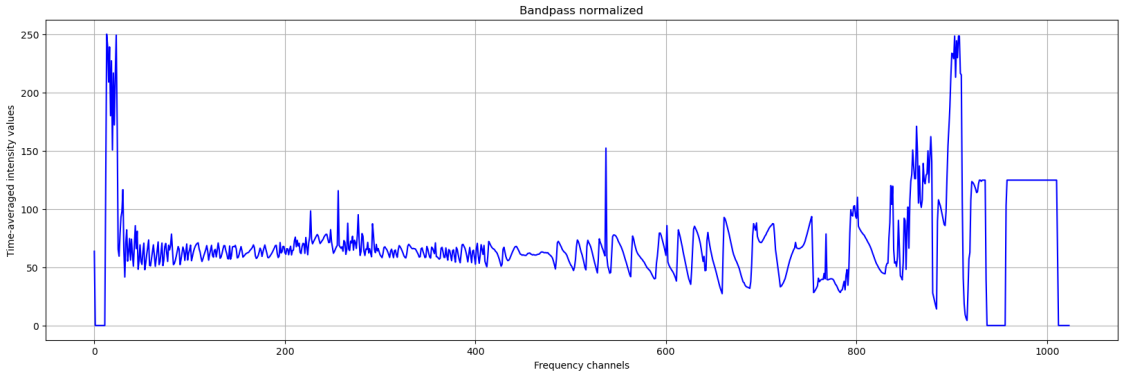

Figure 3.4: Normalized bandpass for 4096 channels (downsampled to 1024)

The effects may not be evident in the bandpass plot themselves, but normalizing the bandpass, automatically removed a lot of non-Gaussian features,instrumental noise, RFI from the data.

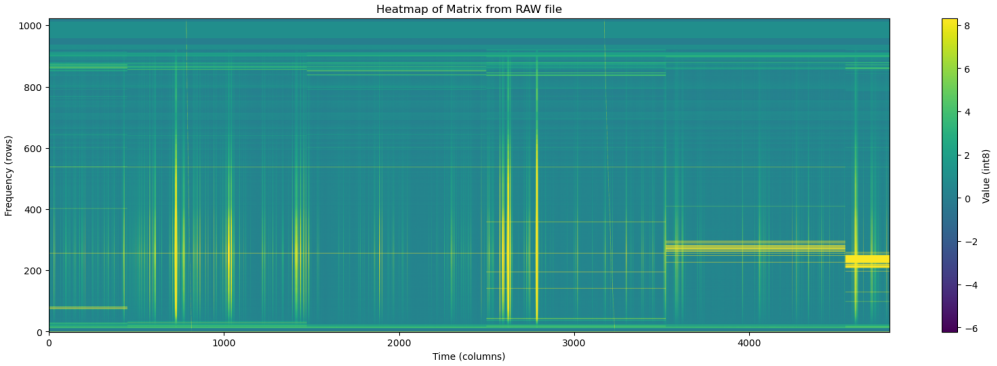

Figure 3.5: Intensity heatmap for original data

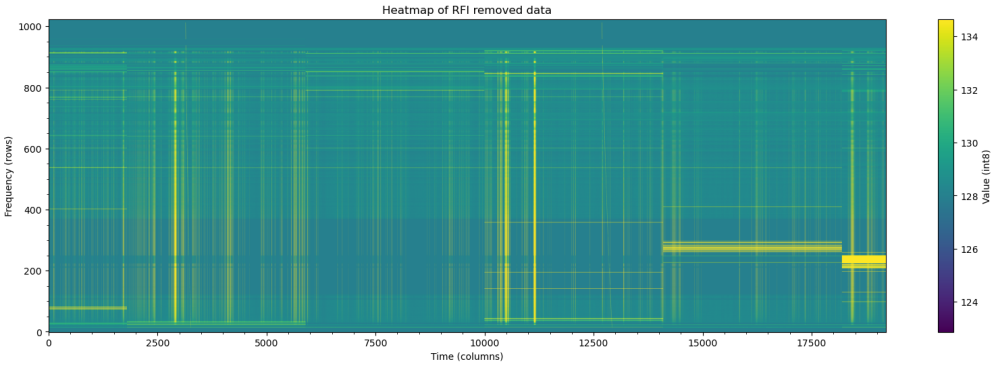

Figure 3.6: Intensity heatmap for bandpass removed data

As it can be seen from figure 3.5 and 3.6, quite a chunk of bright RFI has been removed from the data after baseband normalization. Baseband normalization also takes out the effects of telescopic respose, and now actual RFI features in the data show up and can be removed in next stages.

3.2.2 Zero DM filtering

Transient signals undergo frequency-dependent dispersion (give part) as a result of the intervening medium. The raw intensity data that we received is not dedispersed and is technically dedispersed at zero DM. This zero DM filter [14] can be used to reduce/remove the contribution of signals that have dispersion measure of zero (most terrestrial RFI). Using the zero DM filter, we remove the baseline from the signal (which just the mean value of the data across all frequencies at a particular timestep). Some of the RFI (eg-, from satellites or ground-based transmitters) and can appear as a flat, baseline-like signal in the time or frequency domain. Applying the zero DM filter can remove these kind of components from the data. The equation for zero-DM filtering [32]:

$$S'(f_i, t_j) = S(f_i, t_j) - \frac{1}{n_{\text{chans}}} \sum_{i=1}^{n_{\text{chans}}} S(f_i, t_j) \tag{3.5}$$

- $S(f_i, t_j)$ is the original signal at frequency channel f_i and time step t_j,
- $S'(f_i, t_j)$ is the zero-DM corrected signal,
- n_{chans} is the total number of frequency channels.

So, the zero DM filter subtracts frequency average intensity at each time step from the corresponding the intensity values. But, due to the simplicity of zero-DM filter, it sometimes can cause over-subtraction in channels containing the actual signal.

To counter this, we apply a Z-Dot filter [52] The z-dot filter is built up on the zero-DM filter but adds a correction term. This technique involves the calculation of the zero-DM timeseries for all frequency channels ($t_{DM=0}$) and also the timeseries for each frequency channel (t_i). Then, we estimate how much baseline is present in each frequency channel, this

is done by minimizing the χ^2 value, given below:

$$\chi^2 = (t_i - \alpha_i t_{dm=0} - \beta_i)^2 \tag{3.6}$$

where β_i is the baseline value for that frequency channel, α_i is the scaling factor for that frequency channel (how strongly the zero-DM signal should influence the data in that channel). Solving for α_i, we get:

$$\alpha_i = \frac{\sum_{k=1}^{N} t_i(k) \cdot t_{\mathrm{DM}=0}(k) - \frac{1}{N} \sum_{k=1}^{N} t_i(k) \sum_{k=1}^{N} t_{\mathrm{DM}=0}(k)}{\sum_{k=1}^{N} t_{\mathrm{DM}=0}(k)^2 - \frac{1}{N} \left(\sum_{k=1}^{N} t_{\mathrm{DM}=0}(k) \right)^2} \tag{3.7}$$

Finally, we can filter the timeseries as -

$$t' = t_i - \alpha_i t_{dm=0} \tag{3.8}$$

This technique helps to retain the astrophysical signals while removing most of the zero-DM RFI. Below are the results of applying the zdot filter.

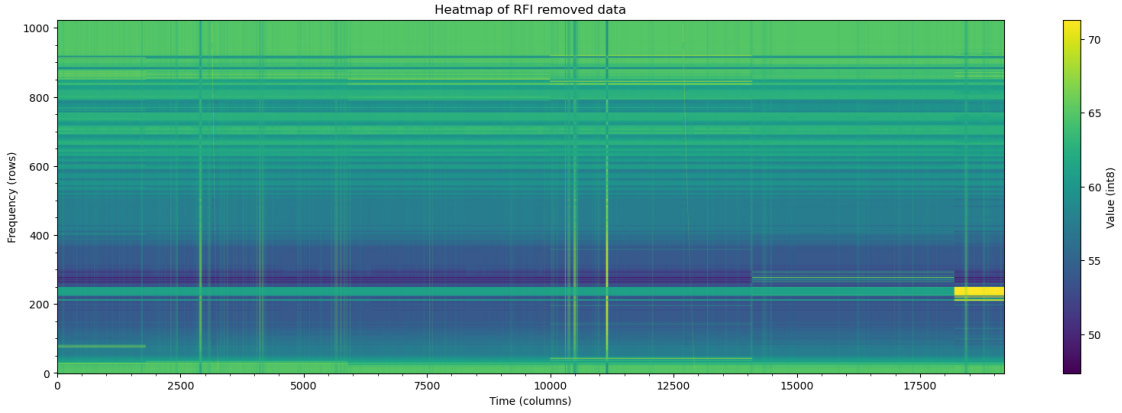

Figure 3.7: Intensity heatmap after applying Z-Dot filter

As, you can see, a lot of the RFI has been removed by applying the Z-Dot filter. We can already see the two FRB signals present in the data. However, one thing to notice is the mean value of the data has become 128, which may seem suspicious since the subtracting the baseline from data should bring down the mean around zero. The main reason for this is that I have added 128 to each element in the data stream during the bandpass correction. This is mainly because in out pipeline we deal with unsigned 8-bit integer (`uint8`) whose value

range from 0 to 255. If I don't add the 128, the negative values due to mean subtraction will get lost or clipped at zero, causing some serious discontinuity in the data that may be later picked up as false positive. The below plot depicting the comparison of timeseries of the original data and z-dot filtered timeseries shows this difference more better.

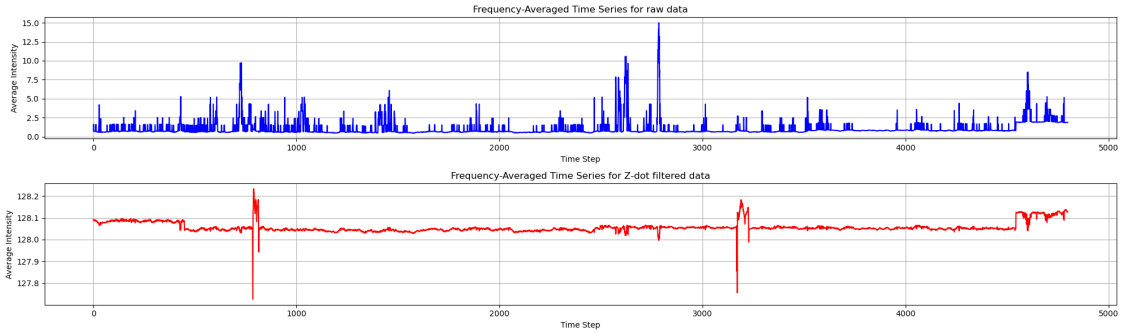

Figure 3.8: Frequency average timeseries for both original data and Z-dot filtered data

3.2.3 RFI removal steps

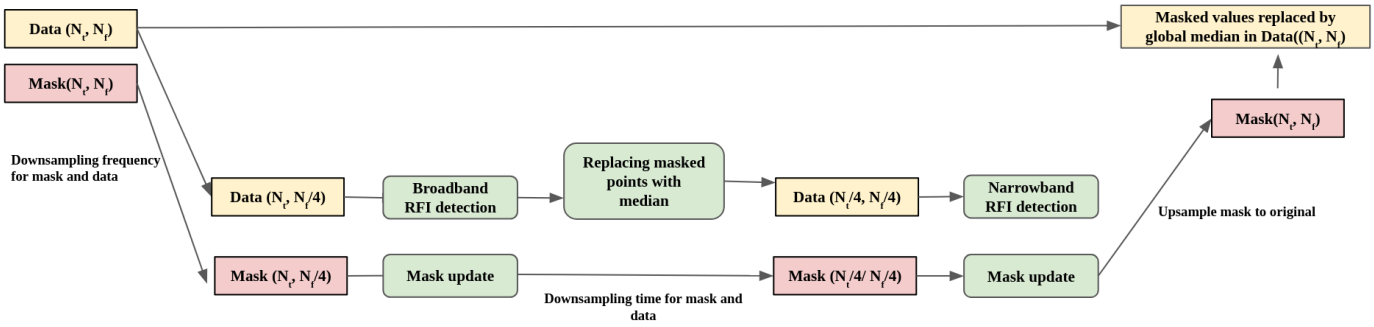

Figure 3.9: Mitigation algorithm

The data has 4096 frequency channels N*25600 time bins (where N is the number of blocks and 25600 is the number of time bins present in each block). After processing the data with baseband normalization and z-dot filtering, a series of key steps are performed to remove both narrow-band and broadband RFI, without removing the actual signal. In order to keep a track of which positions in the data contain RFI, the algorithm creates a mask matrix which has the same size as the data matrix, but only contains zero and one. A position marked as 0 represents RFI and 1 represents non-RFI points. So, firstly, the algorithm initializes a mask matrix containing all ones of size (N_t, N_f).

Downsampling in Frequency axis

We downsample both the data and mask matrices along the frequency axis from 4096 to 1024 channels by a factor of 4. This reduces noise fluctuations, effectively smoothing the data for improved RFI detection. By averaging across frequencies, small fluctuations in broadband transient signals are smoothed out, making them less likely to be mistakenly identified as RFI during broadband RFI removal. The downsampling is carried out by averaging out the frequency values of every 4 consecutive channels, as shown below-

$$\tilde{D}(t, f') = \frac{1}{K} \sum_{k=0}^{K-1} D(t, Kf' + k)$$

where $D(t, f)$ is the original data matrix with time t and frequency f, $\tilde{D}(t, f')$ is the downsampled data, K is the downsampling factor (4 in our case), and f' represents the new frequency index after downsampling.

Broadband RFI detection

In order to remove the broadband RFI, several statistical measures ae involved. Considering the data and mask to be two arrays of size (N_t, $N_f/4$), we iterate over each column (frequency channel), and calculate the fluctuation in intensity value for a particular frequency channel over the total time bins. So, we calculate the rate of change for each time bin (at frequency channel) as:

$$\text{Rate\_of\_change} = I(t, f) - I(t-1, f)$$

Fluctuations sometimes can be very random due to systematic errors. That's why the algorithm performs an exponential smoothening to smooth the smaller fluctuations and bring out the fluctuations due to high power points.

$$S(f) = \alpha \cdot \text{Rate\_of\_change} + (1 - \alpha) \cdot S(f - 1) \tag{3.9}$$

where α is the factor which determines what proportion must the previous transit scores contribute to the current one. This smoothens out the fluctuations as the current transit score is effected by the current rate of change and the previous transit scores. Next we carry

out analysis on the list of transit scores for the every frequency channel. The algorithm performs a modified Z-score analysis using MAD. For each list of transit scores, Median Absolute Deviation (MAD) is caluclated as follows:

$$\text{MAD} = \text{median}\left(|x_i - \text{median}(x)|\right)$$

Where x_i are the data points in the dataset, median(x) is the median of the transit score dataset, $|x_i - \text{median}(x)|$ is the absolute deviation of each data point from the median. The MAD value is most robust to outliers compared to standard deviation. In the next step, a modified z-score is calulated as follows:

$$Z(v) = 0.6745 \cdot \frac{|v - \text{median}|}{\text{MAD}}$$

where v is the value for which the Z-score is being calculated, median is the median of the data points, MAD is the Median Absolute Deviation, x_i are the individual data points and n is the number of data points.

Next, we compare the z-score value for each time step (for a frequency channel) with a threshold value ($\text{Z}_{threshold}$). If the z-score for that position is greater than the threshold, then the same position is marked as RFI in the mask matrix as shown below:

$$M(t, f) = \begin{cases} 0 & \text{if } z > z_{\text{threshold}} \\ 1 & \text{if } z \leq z_{\text{threshold}} \end{cases}$$

The same procedure is applied across all frequency channels, updating the mask matrix accordingly. However, this approach treats both RFI and very bright FRBs similarly, potentially masking transient signals. A crucial distinction between broadband RFI and transients is that transients are dispersed over multiple time bins. To address this, we refine the mask by scanning each time bin with a sliding window of size 128 across the 1024 frequency channels. Within each window, we count the number of masked points (zeros in the mask matrix). If the count exceeds a set threshold, the masked points within that window are retained. Conversely, if the count is below the threshold, all data points in the window are unmasked. This process is repeated for all time bins, resulting in the final mask matrix that effectively identifies and retains broadband RFI points while mitigating the risk of falsely masking transient signals.

In the next step, the global median of the whole dataset is calculated and the positions in the data matrix corresponding to the the masked positions in the mask matrix are replaced by the global median. Even though this data matrix is not our final matrix, this is very helpful for detecting the narrowband RFI.

Downsampling along the time axis

Since the time-axis contains longer datasets to compute, it is really beneficial to downsample the number of time bins. Since, we are processing a few blocks of data in every buffer (i.e $\sim$ 100000 time bins), we downsample the number of time bins by a factor of 4 by using the same averaging technique.

$$\tilde{D}(t', f) = \frac{1}{K} \sum_{k=0}^{K-1} D(Kt' + k, f)$$

So, we end up with a downsampled data matrix of size ($N_t/4$, $N_f/4$)

Narrowband RFI detection

Since, we can't downsample the previous updated mask matrix, we allocate a new mask matrix of size ($N_t/4$, $N_f/4$) and initialized with ones. Now, firstly we want to remove the narrowband RFI which are present over the whole observation length. In order to do this, we calculate the median of the whole range of data for each of the frequency channels. This gives us a list of $N_f/4$ (1024) medians. Next, we calculate the median and standard deviation (σ) for this list. Iterating each of these values, we compare between the value and the median of the medians. If the difference is more than n$\times\sigma$ (where n is a factor), then we mask the whole frequency channel. For, the frequency channel which are not masked, we perform a z-score analysis. We calculate the mean and standard deviation for all the data values in each of those frequency channels (over all time bins), and compute the z-score for each of the points, as:

$$z = \frac{x - \mu}{\sigma}$$

where$z$ is the z-score, x is the data point, μ is the mean of the data set, σ is the standard deviation of the data set. If the z-score is above a certain threshold than we mask those

positions in the mask matrix. Similar to broadband RFI search, we again move a window to restore back actual transient signals which may have been masked as RFI. This time, we move a window of size 200 along the time direction in each of the frequency channels, and use the same logic that if the number of masked points, exceeds the threshold set for the window, we don't change anything. But if the number is less than threshold, then we unmask all the points of that window in the mask matrix.

Upsampling the mask in time bins

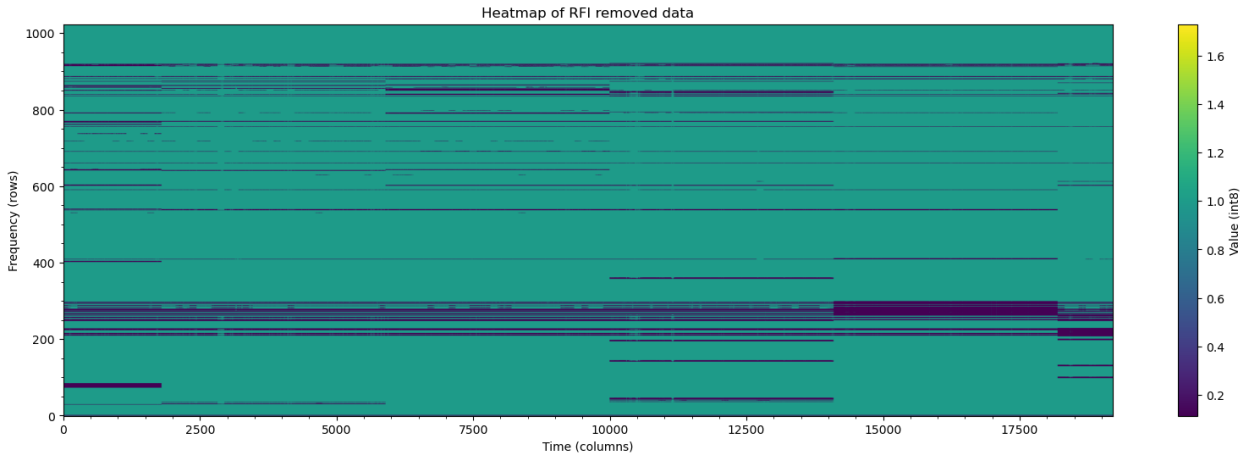

Figure 3.10: Final frequency downsampled mask after AND operation

After the previous steps, we have two masks - one of size (N_t, $N_f/4$), obtained after broadband RFI detection, and the other one of size ($N_t/4$, $N_f/4$), obtained after narrowband RFI detection. In the next step, the algorithm upsamples the the time downsampled mask back upto the size of the other mask. The upsampling is simply carried out by duplicating the each time step four times. For example - if we upsample a sequence $[x_1, x_2, x_3]$ by a factor of 2, the result becomes: $[x_1, x_1, x_2, x_2, x_3, x_3]$. Now, we have two mask matrices of the same size, M_{ij} and M'_{ij}. To combine the effects of both RFI mitigation steps, we compare their elements and perform a logical AND operation, defined as:

$$M_{ij}^{\text{final}} = M_{ij} \wedge M'_{ij}$$

The result of the AND operation follows these cases:

$$\begin{cases} 1 & \text{if } M_{ij} = 1 \text{ and } M'_{ij} = 1 \\ 0 & \text{if } M_{ij} = 1 \text{ and } M'_{ij} = 0 \\ 0 & \text{if } M_{ij} = 0 \text{ and } M'_{ij} = 1 \\ 0 & \text{if } M_{ij} = 0 \text{ and } M'_{ij} = 0 \end{cases}$$

Thus, a point is unmasked (value of 1) in the final mask only if it is unmasked in both M_{ij} and M'_{ij}.

Upsampling the final mask in frequency direction and median replacement

Finally, the algorithm upsample the downsampled mask matrix to the original size (N_t, N_f) using the same upsampling techniques used in the previous steps. Once, the mask is upsampled, we calculate the global median of the original bandpass normalized and z-dot filtered matrix. Looping through the mask matrix, we find the positions in M_{ij} (mask) which are marked zero, and change value of those positions in the data matrix D_{ij} by the global median. Hence, replacing the RFI values with values which follow a Gaussian distribution and are with the standard deviation (σ) of the data.

3.3 Results and Discussion

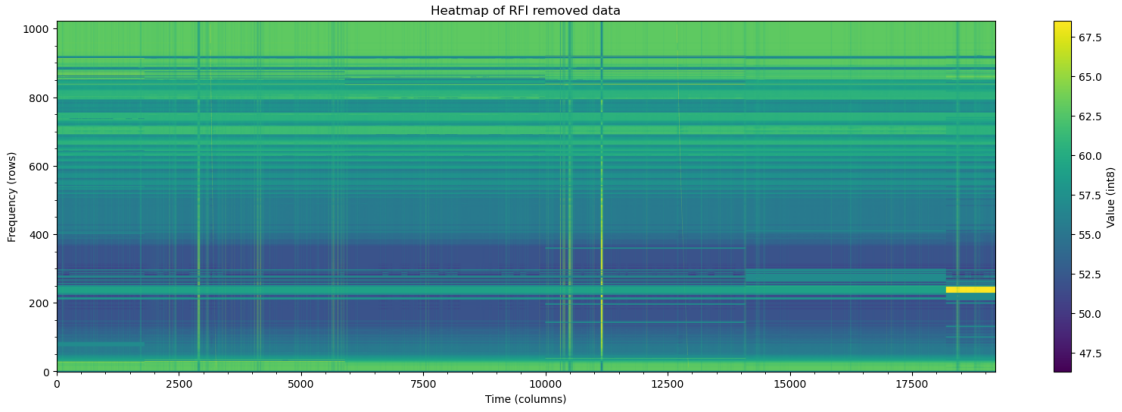

Figure 3.11: RFI removed data

The dataset portrayed in the previous figures consists of two simulated single pulses injected at 50 seconds and 100 seconds at a dispersion measure value of 100 pc cm^{-3} and 200 pc cm^{-3}. Even though the data is filled with RFI as seen in figure 3.5, the pulses can be seen in the final RFI removed dataset. Figure 3.11 shows the RFI removed plot, where the two injected pulses are visible. For a $\sim$ 100 second data, after passing through Astro-Accelerate

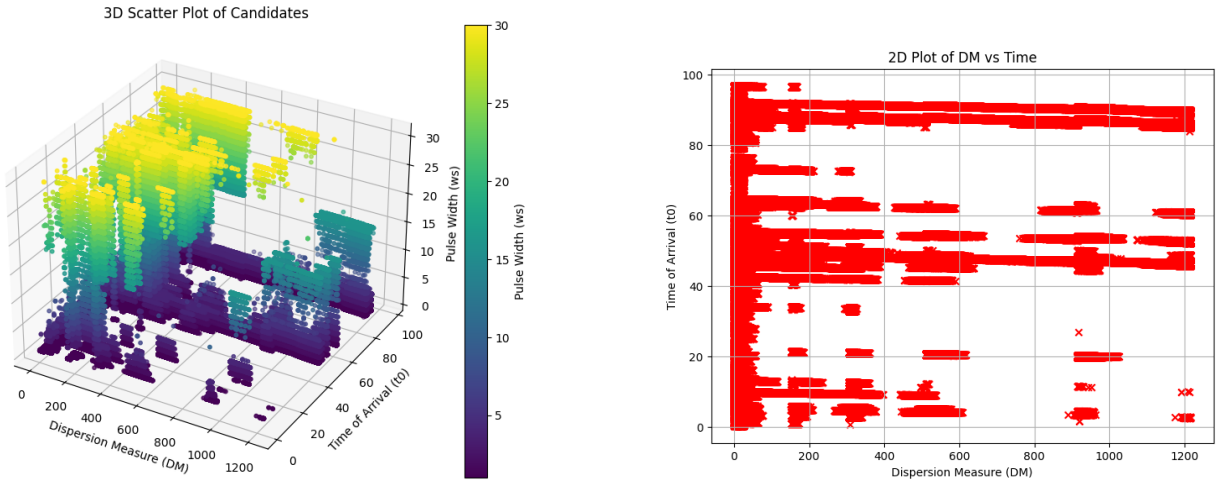

Figure 3.12: Single pulse candidates for non-RFI removed data: a) 3D plot with DM, time and width; b) 2D plot with DM and time

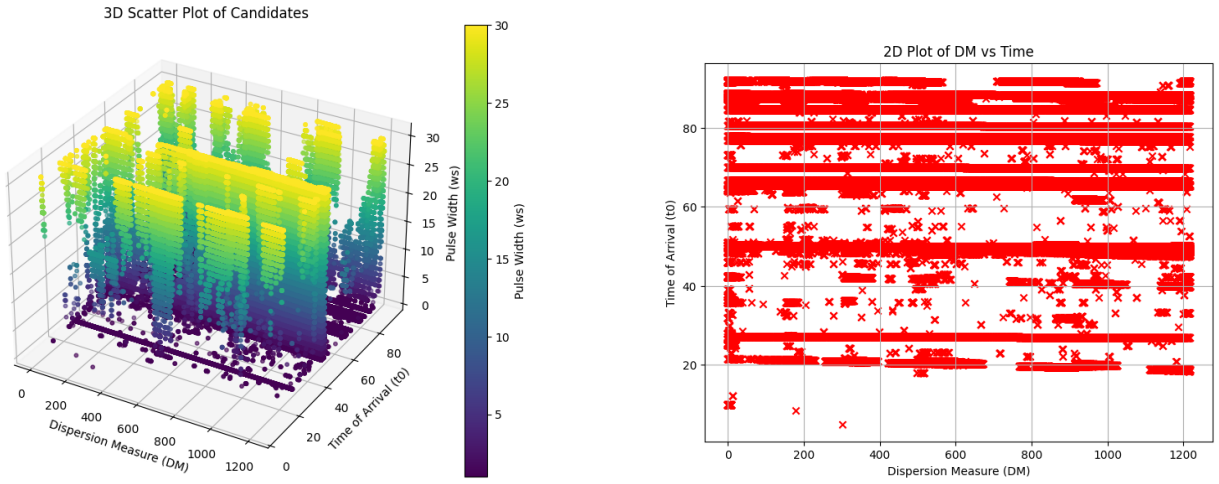

Figure 3.13: Single pulse candidates for RFI removed data: a) 3D plot with DM, time and width; b) 2D plot with DM and time

[1] for single pulse search, the number of candidates detected for non-RFI mitigated data is 5321872, while the number of candidates detected for RFI mitigated data is 2218568,

showing a reduction go $\sim$ 300,000 detected candidates after RFI mitigation. Similar decline is seen for other datasets too. It can be seen in figure 3.13b, the actual candidates injected at 50s and 100s, at a DM of 100 pc cm^{-3} and 200 pc cm^{-3} are still present. This shows the importance of a RFI mitigation algorithm.

Timing results

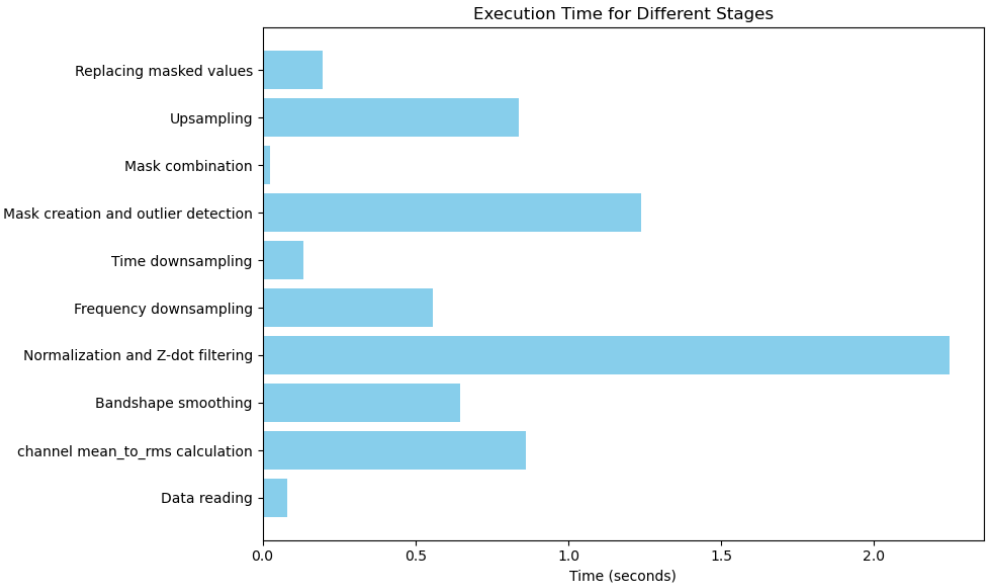

Figure 3.14: Execution time for the various RFI removal steps

The RFI removal algorithm outlined above consists of several steps, including data reading, masking frequency channels based on their mean-to-RMS values, bandshape smoothing, zero-DM correction, downsampling of the data and mask, outlier detection and masking, upsampling, and more. As shown in Figure 3.14, for a data size of approximately 100 seconds, the RFI removal process takes a computation time of 6.48 seconds. This results in a real-time factor of approximately 0.64, indicating that the algorithm is faster than previous approaches. However, further acceleration is possible through the application of GPU acceleration on specific parts of the algorithm. Currently, the algorithm employs OpenMP for CPU parallelization in certain parts of the code. The components that would benefit most from further acceleration are highlighted in Figure 3.14. In particular, the data preprocessing steps, including bandshape smoothing and zero-DM filtering, require

significant acceleration, as these steps currently take around 2 seconds of execution time for data of approximately 100 seconds in length. The two key factors for a good real-time algorithm—speed and accuracy—are addressed by the current implementation. However, there are still parts of the algorithm that could be further optimized to enhance overall performance.

Chapter 4

Developing a real-time clustering algorithm

The primary goal of a clustering algorithm in a single-pulse search pipeline is to reduce the number of detected candidates after the initial search. The single-pulse search process (Astro-accelerate) consists of two key steps: dedispersion and matched filtering [1][47], both of which give rise to a lot of redundant candidates and false positives.

- Since this is a search pipeline and the dispersion measure (DM) of the source is unknown, the dedispersion step scans over a wide range of DM values, typically from 0 to 2000, with step sizes of 0.1, 0.2, or 0.5. This means that closely spaced DM values are searched over, leading to the detection of multiple candidates corresponding to the same astrophysical pulse. The fundamental idea behind dedispersion is that when a pulse is corrected with the correct DM, the delay introduced by interstellar dispersion is removed, making the pulse peak more prominent in the data. However, since the exact DM is unknown, neighboring DM values can still produce a strong pulse detection, leading to duplicate candidates from the same astrophysical event. Observations have shown that the detected DM values for a single pulse can vary by 0.1 to 0.8 due to multiple factors.

    1. **DM variations and multiple detections:** Since the dedispersion step iterates over many DM values, multiple candidates from the same pulse appear due to

small differences in DM. These variations can depend on factors like signal-to-noise ratio (SNR), pulse width, and instrumental effects.

2. **Low DM Contamination from Radio Frequency Interference (RFI):** Another issue arises from the fact that dedispersion also searches over low DM values. Since RFI originates from terrestrial sources such as FM radio, television signals, and RADARs, it has very low dispersion measures. Some of these bright RFI signals can be falsely selected as candidates when searching at low DM values. As a result, a significant fraction of detected candidates consists of spurious RFI detections rather than real astrophysical pulses.

- After dedispersion, matched filtering or peak filtering is applied to enhance the signal-to-noise ratio (SNR) of detected pulses. The core idea of matched filtering is to convolve the signal with a set of predefined template functions optimized for various pulse widths. This method significantly improves the SNR of weak pulses that might otherwise be lost in noise. Since the intrinsic width of the pulse is unknown, the search is performed over a range of template widths, from narrow pulses (spanning only a few time bins) to broad pulses (spanning hundreds of time bins). When the template width closely matches the actual pulse width, the SNR is maximized. However, this approach introduces challenges, such as detecting the same candidate multiple times for slightly different template widths and mistakenly classifying bright radio frequency interference (RFI) as astrophysical candidates. The above reasons increase the number of candidates after the single pulse search by Astro-Accelerate.

4.1 Actual Number vs. Detected Candidates

4.1.1 Creating a Robust Dataset

To assess the impact of RFI and false positives on candidate detection, we conducted an extensive, months-long evaluation of our pipeline. This involved comparing the actual number of pulses present in the data with the candidates detected after running a single-pulse search using Astro-Accelerate.

Testing such a pipeline in real-time with fast radio bursts (FRBs) is inherently challenging due to two key reasons:

- FRBs are non-periodic. Even when following up on known repeaters, there is no guarantee that a burst will be detected during an observation.
- For real observational data, the true number of pulses present is unknown, making it difficult to quantify detection efficiency.

An alternative approach is to use pulsars as test sources. While pulsars are periodic and provide a known reference for the number of pulses present in the data, they are still not ideal for comprehensive testing. The limitations of using pulsars include:

- Most detected pulsars are of Galactic origin and have relatively low dispersion measures (DMs), typically ranging from 20 cm^{-3} pc for nearby pulsars to $\sim$ 1000 cm^{-3} pc for those near the Galactic center, where the interstellar medium (ISM) is denser. The average pulsar DM lies in the range of 10–300 cm^{-3} pc (excluding Galactic center pulsars) [41]. This limited DM range is insufficient for testing the pipeline over the full DM parameter space (0–2000 cm^{-3} pc).
- Pulsar signals are significantly weaker than FRB pulses, often by orders of magnitude. A bright pulsar may have a peak flux density of a few Jy, whereas FRBs can reach tens to thousands of Jy [37].

To overcome these limitations, we adopted a hybrid testing approach that combined real observational data with simulated pulses. Observations were conducted using pulsars or calibrator sources (depending on source availability within the telescope's zenith). However, to ensure robust testing across a wide DM range, we injected simulated pulses into the data stream in real time.

The simulated pulses were generated using Arachne (cite) and were designed to cover a broad range of dispersion measures and flux densities: **Dispersion Measures:** Pulses were inserted at DM values ranging from 100 to 2000 cm^{-3} pc, with increments of 100 cm^{-3} pc. Typically, the first simulated pulse (DM = 100 cm^{-3} pc) was injected at 150s, followed by subsequent pulses (DM = 200, 300, etc.) after every 100 seconds.; **Flux Densities:** The

tests were repeated for multiple flux levels: 0.2, 0.5, 1.0, 1.5, 2.0, 5.0, 10.0, and 20.0 Jy (not all tested on the same day). Furthermore, similar tests were conducted across all GMRT observing bands—Band 3, Band 4, and Band 5—to evaluate the pipeline's performance under different radio frequency conditions. One thing to note is that all these trials were not performed every time. This approach allowed us to rigorously assess our pipeline's ability to detect real astrophysical pulses while quantifying the effects of RFI contamination and false positives across a wide parameter space.

4.1.2 Results and observations

All the tests were performed on data from a single beam. In nearly every trial across the entire parameter range, we observed an increase in the number of candidates detected by Astro-Accelerate compared to the number of injected candidates. Although the ratio of increase varied depending on factors such as the pulse flux, the bandwidth of the data, and other parameters, a consistent global trend of increased detections was observed. Importantly, while no robust statistical patterns were found in the ratio of increase, a notable difference was observed in detection rates when RFI (Radio Frequency Interference) filtering was disabled as opposed to when it was enabled. This suggests that RFI plays a significant role in candidate detection, likely contributing to false positives.

To illustrate these results, we refer to the comparison shown in Figure 4.1. The figure shows one of the datasets comparing the number of injected candidates to the number of detected pulses for observations made on 1st May 2024. This dataset, covering Band 3 and Band 4, clearly demonstrates that across all flux levels, the number of detected candidates consistently exceeds the number of injected pulses. This outcome highlights the heightened sensitivity of the detection algorithm, which appears to register a larger number of candidates, especially when RFI filtering is relaxed.

As seen in the figure 4.1, the gap between the number of injected and detected candidates is pronounced, especially under the conditions of relaxed RFI filtering. This increase in detections underscores the impact of the algorithm's sensitivity to weak signals, as well as the possible role of RFI in producing false positives. These trends were consistent across different bands and flux levels, supporting the hypothesis that RFI and algorithm sensitivity are crucial factors influencing candidate detection

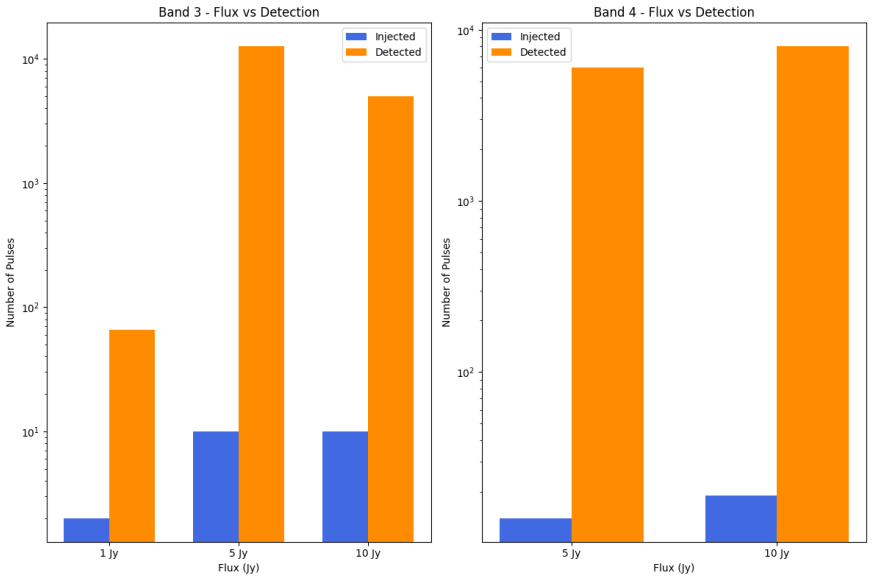

Figure 4.1: Comparison of the number of injected candidates versus the number of pulses detected (in logarithmic scale) for test observations on 1st May 2024 (Band 3 and 4).

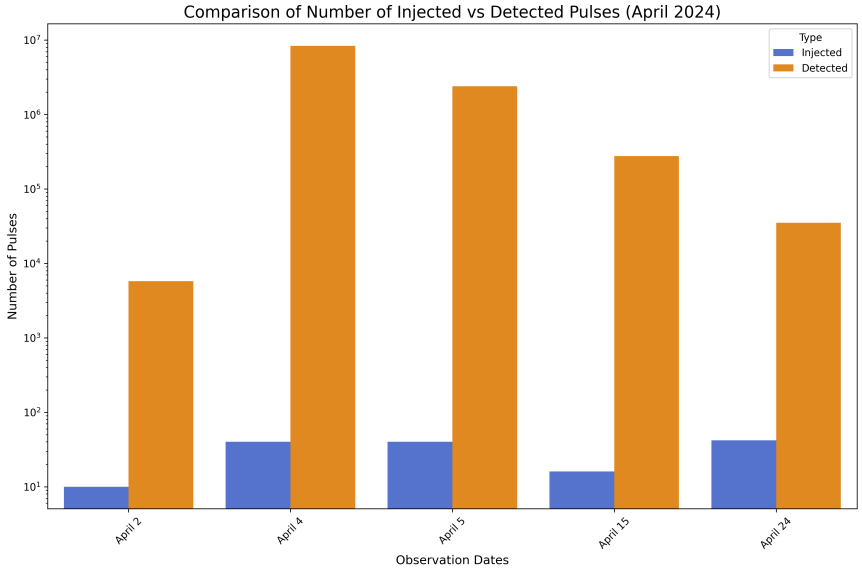

Figure 4.2: Comparison plot for all the tests done in April 2024

As shown in Figure 4.2, the total number of detected candidates is consistently much higher than the number of injected candidates. Notably, the tests conducted on April 4, 5, and 15, which did not include RFI mitigation, exhibit a significant increase in the number of detected candidates—by nearly an order of magnitude—compared to the other tests.

These observations emphasize the critical role of a clustering algorithm in filtering out false positives, ensuring more accurate detection of real astrophysical signals.

4.2 Algorithm for reducing the number of candidates

The algorithm follows a three-step process to minimize false positives and redundant candidates. Since all these steps are performed in real-time, computational efficiency is crucial. To achieve this, the process is accelerated using GPUs (Graphics Processing Units).

4.2.1 Thresholding

Given that we are searching for bright astronomical signals, we can impose constraints on certain parameters to filter candidates effectively. We focus on the time of arrival, dispersion measure (DM), signal-to-noise ratio (SNR), and width of each candidate. By setting thresholds on DM and width, we can significantly reduce the number of candidates, albeit with some trade-offs.

Dispersion Measure (DM): A threshold of 10–20 cm^{-3} pc can be applied to DM, meaning any candidate with a DM below this value will be discarded. The rationale behind this is that terrestrial radio frequency interference (RFI) typically exhibits very low DM values, rarely exceeding a few cm^{-3} pc. At the same time, almost all known FRBs are extragalactic, and even in high Galactic latitudes, the minimum expected Milky Way DM contribution is around 30–50 cm^{-3} pc. The lowest recorded DM for an FRB so far is 141 cm^{-3} pc [49]. However, since our pipeline also aims to detect pulsars, we must keep the DM threshold low enough to avoid inadvertently filtering out genuine pulsar signals.

Width: FRBs are typically characterized by intense, millisecond-duration pulses. Most detected FRBs have widths on the order of a few milliseconds. For instance, FRB 010621 has a width of approximately 7.8 ms [31]. To ensure we do not exclude relevant transients, we can conservatively set an upper limit of 100 ms. This threshold should also be sufficient for detecting pulsars. Given this constraint, we can reasonably assume that most astronomical candidates will not be missed.

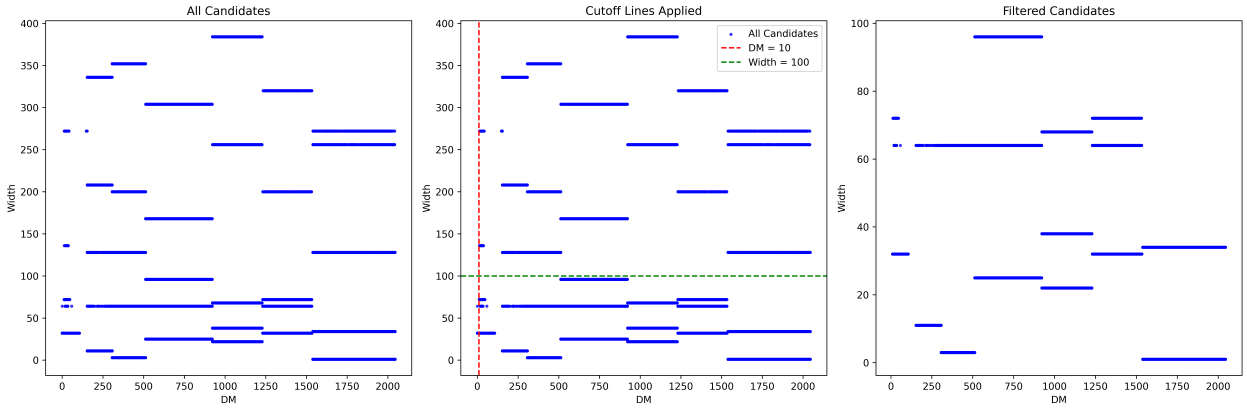

Figure 4.3: Left: All candidates plotted on DM-width plane; Middle: Candidates with cutoff; Right: Filtered candidates

The above plots show all the candidates after single pulse search in the DM-width plain and how the number decreases after the putting the cutoffs. It is seen that a major portion of the RFI and false positive candidates can be removed by using necessary thresholds. For this case, the number of candidates dropped from $\sim$ 70000 to $\sim$ 40000 after thresholding. It is seen that higher the number of candidates after single pulse search, more amount of candidates are removed by thresholding. But this is obvious as more number of candidates arise due to presence of more amount of RFI.

4.2.2 Clustering

The clustering process involves two key steps. First, candidates with similar characteristics—such as time of arrival, dispersion measure (DM), and pulse width—are grouped together. Then, from each group, the candidate with the maximum signal-to-noise ratio (SNR) is selected. This approach effectively reduces redundancy caused by dedispersion and single-pulse searches, allowing us to retain only the most scientifically valuable candidates—those with the highest SNR. In order to group the candidates together, we tried out various machine learning algorithms for grouping candidates like DBSCAN, FoF, etc. [54] For a real-time pipeline, there are two important factors that make an algorithm better than other- accuracy, precision of the algorithm, and efficiency. Based on our requirements, we opted for DBSCAN based algorithm for clustering.

DBSCAN(Density-based Spatial Clustering of Applications with Noise) is an algorithm for data clustering. It is a density-based, non-parametric clustering algorithm, meaning that given a set of points in some dimensional space, it groups together points that are closely packed (i.e., with many neighboring points) and marks outliers as noise if they are located in low-density regions. The operation of DBSCAN is based on two main parameters: ϵ, which specifies the radius of the neighborhood with respect to a given point, and N, the minimum number of points required within the neighborhood for a point to be considered a core point.

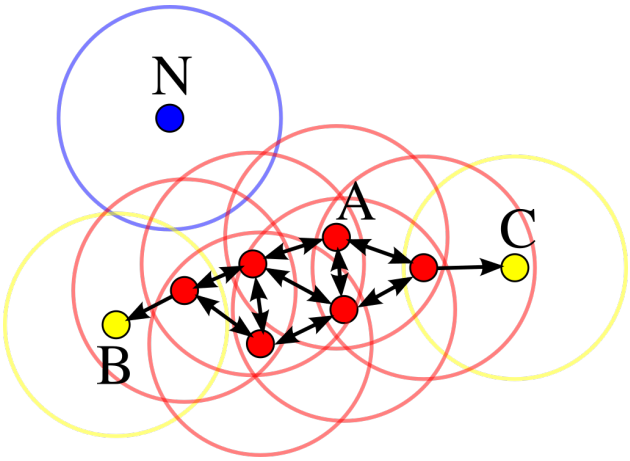

Figure 4.4: In this diagram, the minimum number of points required to form a core point (minPts) is set to 4. Point A and the other red points are classified as core points because their -radius neighborhoods contain at least 4 points, including themselves. Since these core points are all reachable from one another, they form a single cluster. Although points B and C do not qualify as core points, they remain part of the cluster because they are connected to A through a path of core points. In contrast, point N is neither a core point nor reachable from any core point, so it is considered a noise point.

A point a is classified as a *core point* if there are at least N points within its ϵ-radius neighborhood. If another point b lies within this ϵ-radius of a, then b is said to be *directly reachable* from a. Furthermore, b is *reachable* from a if there exists a sequence of points $a_1, a_2, \ldots, a_n$ satisfying

$$a_1 = a, \quad a_n = b, \quad \text{and} \quad a_{i+1} \text{ is directly reachable from } a_i \text{ for all } i = 1, \ldots, n-1.$$

All points along this path, including a and the intermediate points, must be *core points*—with the possible exception of b. Points that are not reachable from any other point are classified as *outliers* or *noise points*. If a is a core point, it forms a cluster that contains all core and

non-core points reachable from it. Each cluster must include at least one core point; non-core points, while they may belong to a cluster, form its *boundary* since they do not contribute to further expanding the cluster.

In our project, we apply a two-dimensional DBSCAN algorithm using the time of arrival and dispersion measure of the candidates. To achieve more effective clustering, it is important to work with features on comparable scales. Rather than using standard scaling, which maps values to a range between 0 and 1, we keep the time of arrival values unchanged and convert the dispersion measure into time units. This conversion reflects the time delay of the signal between the highest and lowest frequencies. By transforming both parameters into the same unit of time, we eliminate inconsistencies arising from different scales, allowing the clustering algorithm to treat both features uniformly.

$$\text{delay} = k_{\text{DM}} \cdot DM \cdot \left(f^{-2} - f_{\text{ref}}^{-2}\right)$$

where $k_{\text{DM}} = 4.1488064239 \times 10^3$, f is the frequency (in MHz), $f_{\text{ref}}$is the reference frequency (in MHz). From each cluster, we pick the maximum SNR candidate and

4.3 Results and Discussion

GPU accelerated clustering

Further, in order to improve the computational efficiency of our code, we implemented a a GPU based DBSCAN algorithm [58] to process our candidates. Figure 4.5 shows that the CPU-based DBSCAN algorithm performs better than the GPU-based algorithm when the number of samples is relatively small (around 70,000). This is primarily because the GPU algorithm requires the creation of a CUDA Data-frame, a process that introduces some overhead, making it less efficient for smaller datasets. However, as the number of samples increases beyond this threshold, the GPU-based algorithm significantly outperforms the CPU-based one.

In Figure 4.6, we observe that the CPU algorithm starts to reach a bottleneck at around 10,000–20,000 samples, after which its performance degrades rapidly. This highlights the

necessity of using the GPU-based approach for handling large datasets, as it effectively leverages parallel processing to accelerate clustering, confirming its suitability for our work.

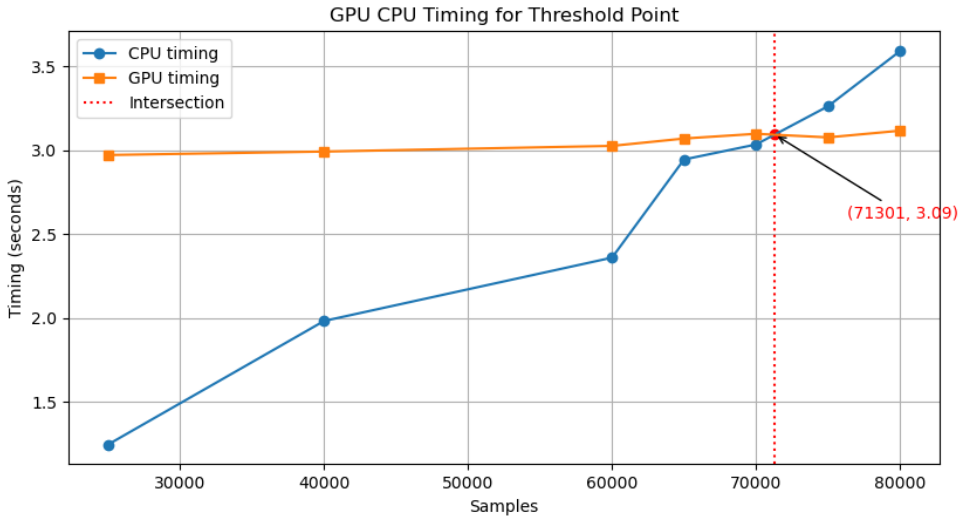

Figure 4.5: Processing time vs number of samples for GPU and CPU based algorithms

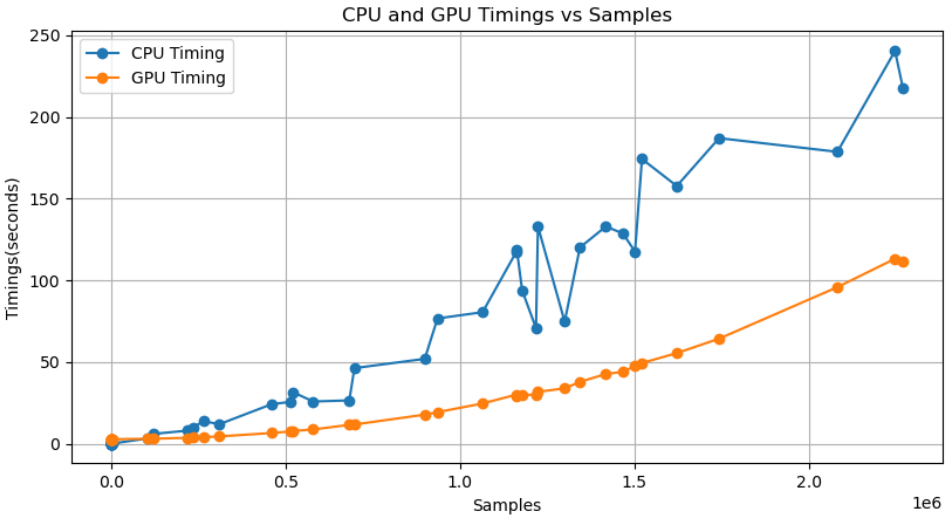

Figure 4.6: Timing as function of number of samples for both GPU and CPU based algorithms

Fine-tuning epsilon

As described before, epsilon (ϵ) is the parameter which specifies the radius of the neighborhood with respect to some point. Using a proper value of epsilon is essential for properly clustering the candidates, which in turn increases the number of true positives and reduces the number of false positives. Clustering using DBSCAN requires one other parameter- minimum number of points required to be present within the neighborhood (N). The value of N is set to 3 based on previous observations. It is seen that an actual pulse is detected at-least 3 times with nearby DM and time of arrival. Fixing the value of N to 3, we vary the value of epsilon to find the best value of epsilon.

In order to do this, I used five different datasets observing five different pulsars and processed those data for list of single pulses. The clustering algorithm was run on each of these data with ϵ ranging from 0.05 to 1, with step sizes of 1. After clustering, we got the final list of candidates fro each epsilon value for each observation. The five observations were on five different pulsars with different DM values. In order to check the effectiveness of the clustering algorithm in each case, we calculated the number of actual pulses present after clustering for each case. For a pulsar of DM α, all candidates with detected DM values ranging from (α - 0.5, α + 0.5) were considered as real candidate. An average list of number of actual candidates detected for each ϵ was calculated for the five observations. The results are plotted in figure 4.7. It is evident that an epsilon value of 0.2 to 0.3 is optimal to get the best results.

4.3.1 Clustering Results

The clustering results are divided into two parts - one focusing on the reduction of number of candidates after clustering, compared to the list of candidates after single pulse search in the data, the other is the validity of the clusters formed after clustering in the real-time system.

The clustering algorithm consistently reduces the number of candidates by several orders of magnitude. This trend is observed across various tests — from offline, single-beam simulations to real-time, multi-beam pulsar observations. In every case, a significant reduction in candidate numbers is evident after clustering. Figure 4.8 illustrates a two-order decrease in

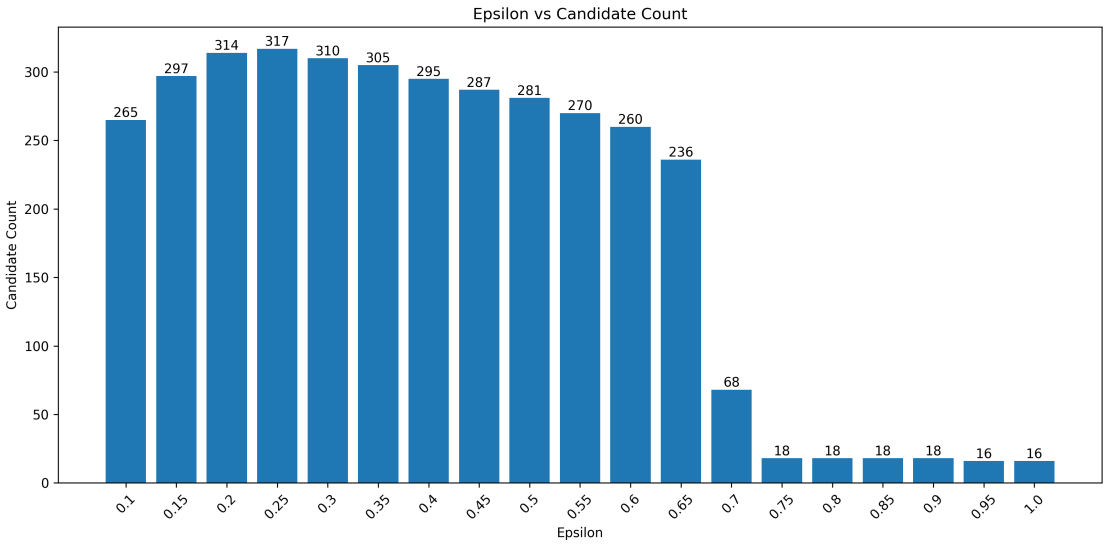

Figure 4.7: Average total number of actual candidates against epsilon

the number of candidates across a hundred beams during the observation of pulsar B0329+54 across band 3.

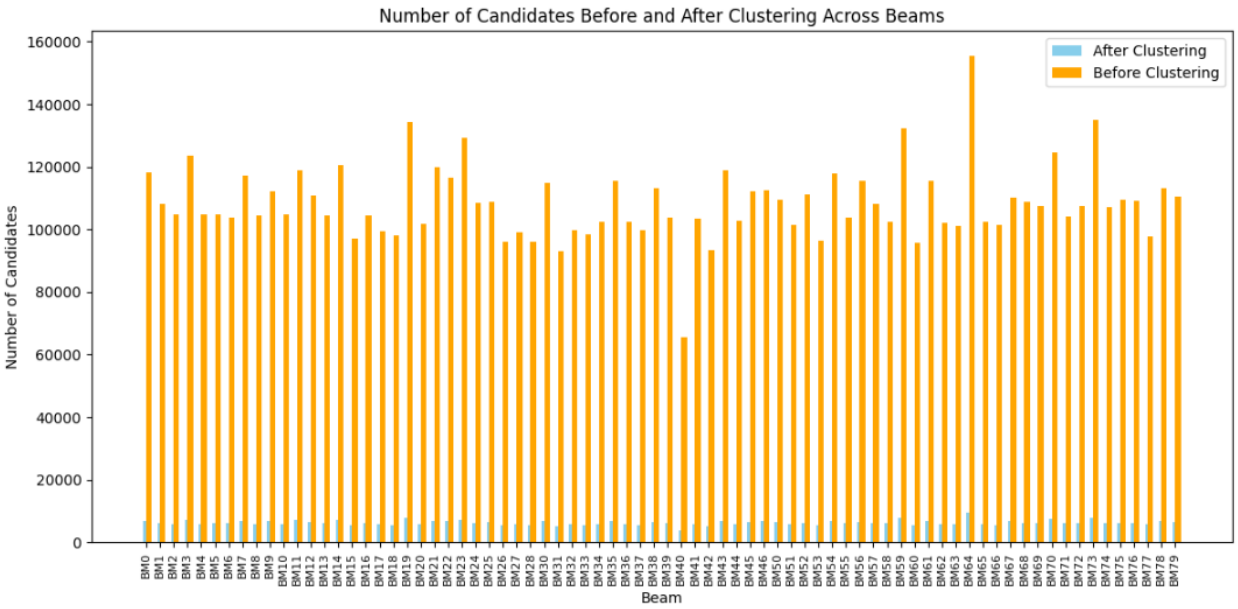

Figure 4.8: Number of candidates before and after clustering

The clustering algorithm effectively identifies clusters containing the actual candidates. Out of the 800 beams, the algorithm successfully detects the true candidates in beams where

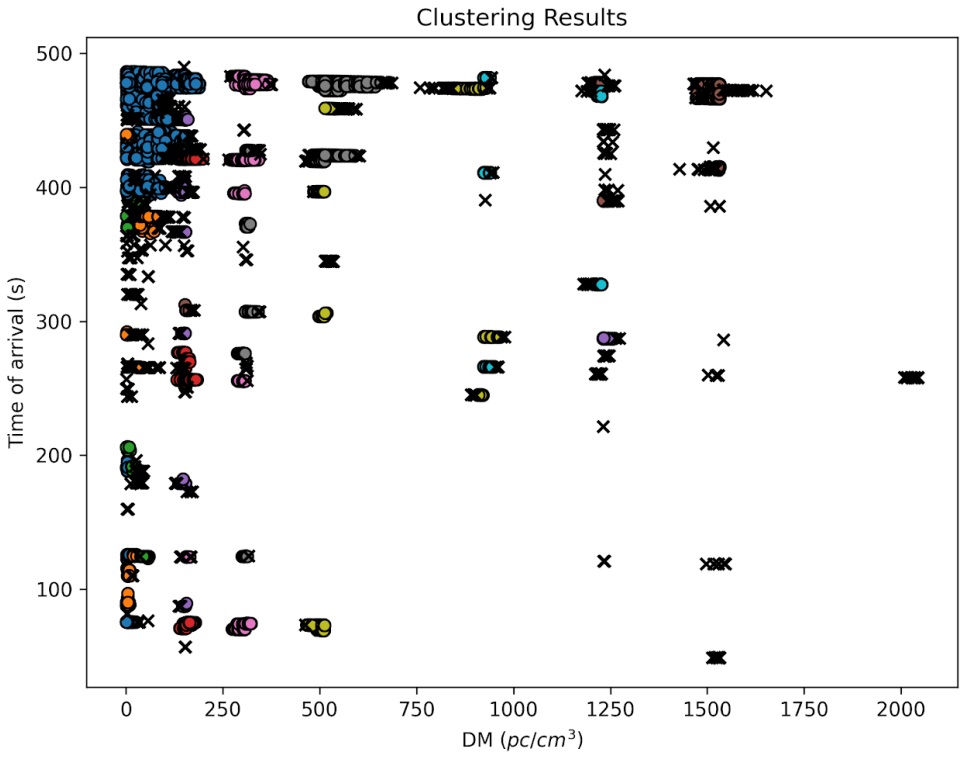

Figure 4.9: Results of clustering for Crab Pulsar data in Band 4 (550-750 MHz) for one beam out of 800. Different colours show the different clusters formed by DBSCAN in the DM-time plane. The candidates marked "x" are the outliers.

they were identified through single-pulse searches. The core concept behind the clustering code is to group candidates into clusters and select the candidate with the maximum SNR from each cluster. An optimal clustering algorithm should minimize the number of filtered candidates while ensuring true positives are not missed. Figure 4.9 illustrates the clustered candidates for a single beam during a Crab pulsar observation in Band 3. Before clustering, the average number of candidates per beam after single-pulse searches was approximately 10,000. After applying the clustering algorithm, the number of filtered candidates was reduced to around 100. These results show the efficiency, accuracy, and necessity of the GPU-based clustering algorithm.

Chapter 5

Real-time Candies and FETCH

Candies is a candidate feature extraction tool for single pulses. Using the information of dispersion measure. time of arrival, beam number ,and width for a candidate, the tool can cutout required portion from the actual data and produce three features/plots, namely:

- **Dedispersed dynamic spectrum:** is a time-frequency representation of radio intensity data, corrected for dispersion delays using the appropriate dispersion measure (DM). In this plot, the x-axis represents time bins, while the y-axis shows the 4096 frequency channels spanning the band's total bandwidth. Accompanying the dynamic spectrum is a dedispersed time series, which displays the average intensity across all frequency channels for each time bin.

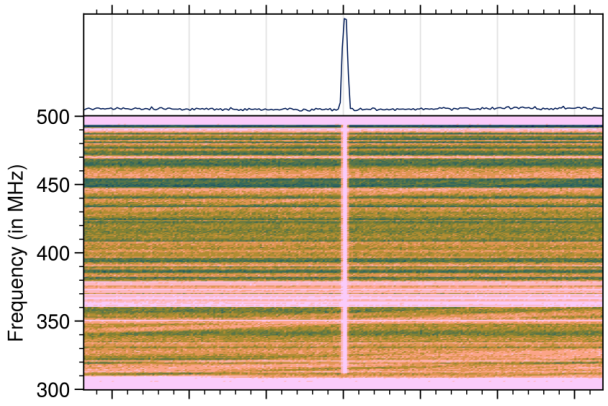

Figure 5.1: Dedispersed Dynamic Spectrum

As both the dynamic spectrum and time series are dedispersed, any astrophysical pulse appears as a bright vertical feature in the dynamic spectrum and as a distinct peak in the time series. These plots are crucial for distinguishing real astrophysical candidates from radio frequency interference (RFI).

- **DM transform** is a plot to characterize actual signals for RFI. The main idea behind DM transform is the smearing of the signal when dedispersed at incorrect DMs. Each frequency channel will experience certain delay based on $\Delta t \propto \text{DM} \propto \nu^{-2}$. In DM transform, he signal is dedispersed at a range of DM values near the near the actual value, and the time delay curve for each each trial is plotted.

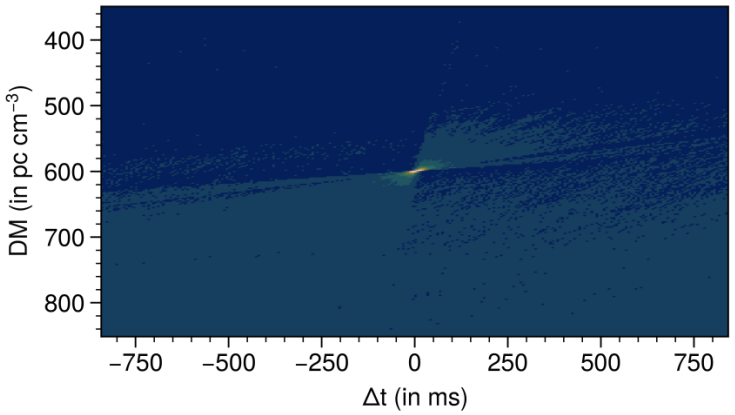

Figure 5.2: DM transform

A bow-tie pattern is created for actual signals due to stretching of the signal at incorrect DMs. Above the true DM, the signal appears over corrected causing higher frequencies to be shifted far more back in time, and lower frequencies not enough. Below the true DM, the signal is under corrected causing the opposite effect compared to the previous one.

FETCH [2] is a convolutional neural network (CNN) [48] designed to classify astronomical signals and false positives, distinguishing between genuine sources such as pulsars and Fast Radio Bursts (FRBs) and noise or Radio Frequency Interference (RFI). It uses the dedispersed dynamic spectrum and DM transform as key features for classification.

Both tools operate on offline data and process SIGPROC [36] filterbank files, which contain raw radio data along with a header providing essential metadata about the observation.

These tools are GPU-accelerated to handle large datasets efficiently. In this workflow, Candies takes the filterbank file and a CSV file containing candidate information — specifically DM, time, SNR, and width — and outputs an HDF5 [51] file for each candidate. Each HDF5 file contains the candidate's dynamic spectrum, time series plots, and additional metadata. FETCH then reads these HDF5 files and classifies candidates as either true positives or false positives. Finally, it produces a CSV file, labeling candidates with 1 for true positives and 0 for false positives. In a real-time pipeline, this offline processing won't work. Instead of a filterbank file, the data is stored in a memory (capable of storing $\sim$ 400 seconds of data) which is overwritten after every $\sim$ 400 seconds. So, all the data processing for that data needs to be in real-time. I have modified the candies and FETCH to be capable of working in real-time by reading data straight from the memory. In-order to properly understand the working of candies and FETCH, we need to understand the data streaming architecture properly.

5.1 Multi-beam FRB shared memory

A shared memory is a process to communicate between different processes by allowing different processes to access a common region of the memory. Instead of each process creating its own separate memory, shared memory allows for data to be exchanged efficiently between processes by allowing them to read from and write to the same memory segment. Our pipeline creates a data buffer shared memory called FRB shared memory in order to stream the intensity data. Further, the shared memory contains multi-beam data. Each node hosts its own shared memory. The total N beams covering the total field of view, are split across M nodes. Thus each node hosts N/M beams. Currently, a total of 800 beams are split over 16 nodes, so each node hosts 50 beams.

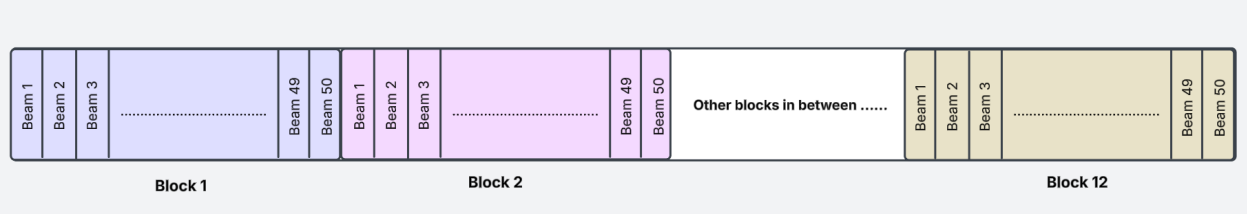

Figure 5.3: FRB shared memory for 50 beams

The FRB shared memory contains a total of 12 blocks. Each block has 25600 time bins, with each time bin being 1.31072 ms. Thus one block contains $25600 \times 0.00131072 = 33.55$ seconds of data. Considering a single beam in one block, the size of the block would be 25600 (number of time samples) $\times$ 4096 (number of frequency channels = 104857600 bytes. But a single block contains data of 50 beams placed in series, making the total size of one block shared memory 52428.8 MB. Hence the size of the shared memory as $\sim$ 629 GB. So, we can imagine the data structure as a 4096×25600×50 matrix, where each value corresponds to intensity. Also, the data for all the beams are updated at the same time, so different parts of the memory are updated at the same time. Hence, we need some rigorous pointer algorithm to retrieve the necessary data.

5.1.1 Reading data from FRB shared memory

The real-time Candies pipeline processes a CSV file, which is written to disk after clustering. This file contains information about each candidate, including the dispersion measure (DM), time of arrival, signal-to-noise ratio (SNR), width, and beam number. Since the focus is on identifying the candidates with the highest SNR, the algorithm first reorders the candidates in descending order of their SNRs. The candidate with the maximum SNR is then selected for further processing. To locate the position of the selected candidate in the data, the algorithm uses the candidate's time of arrival. It calculates the maximum delay caused by dispersion, using the dispersion measure (DM) along with the highest and lowest frequencies of the band, according to the following equation:

$$\text{max-delay} = k_{\text{DM}} \cdot DM \cdot \left(f^{-2} - f_{\text{ref}}^{-2}\right)$$

where $k_{\text{DM}} = 4.1488064239 \times 10^3$. Now that we have the maximum delay, we calculate the staring bin and ending bin to cut data (binbeg and binend).

$$\text{bin}_{\text{beg}} = \left\lfloor \frac{t_0 - \text{maxdelay}}{\Delta t} \right\rfloor - w_{\text{bin}} \tag{5.1}$$

$$\text{bin}_{\text{end}} = \left\lfloor \frac{t_0 + \text{maxdelay}}{\Delta t} \right\rfloor + w_{\text{bin}} \tag{5.2}$$

The wbin in the above equations refers to the detected width of the pulse. The time we get is an absolute time and is zero at the start of the observation. So, the time bin also starts from zero and increments as the observation goes. The binbeg refers to the **offset** of the data, and (binend - binbeg) is the **count** of the data to be chopped.

Using the count and offset information, the algorithm calculates the blocks containing the beginning and end bins by performing integer division of `binbeg` and `binend` with the total number of bins in a block (25600): binbeg_block_loc = binbeg/25600. The block numbers start from 0 and increment uniformly until the end of the observation. However, since the blocks are cyclic and get overwritten after 12 blocks, the algorithm accounts for this by calculating the cyclic block locations using the modulo operation with the total number of data blocks (12): binbeg_block_loc_cyclic = binbeg_block_loc%12. Finally, to identify the exact bin within each block, the algorithm computes the bin location by taking the modulo of `binbeg` and `binend` with the total number of bins in a block (25600): binbeg_bin_loc = binbeg%25600.

Now, we need to read the data by moving our pointer to the exact memory location. There are factors which influence this - which blocks do the binbeg and binend lie in, and which beam do we want to read. There are three cases to deal with -

- If both binbeg and binend lie on the same block, then we just move to pointer to the appropriate beam within the block and read the data from binbeg to binend within that beam.

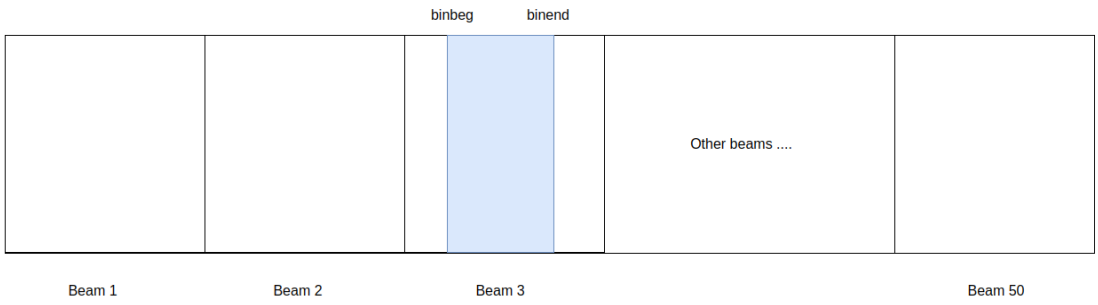

Figure 5.4: If the binbeg and binend are present in the same block and we want to cut data for beam 3

- If the binbeg and binend are present in different blocks then the algorithm needs to skip the other beams present in between, as the two blocks of the same beam are not

continuously placed in the memory. The first block's data starts at the correct offset and extends to the end of the block. Intermediate blocks are copied in their entirety. The final block's data starts at the beginning and ends at the target bin.

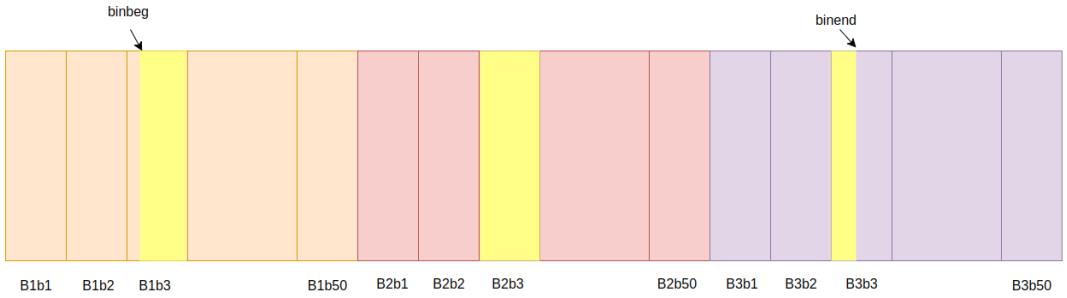

Figure 5.5: Binbeg and binend located at different blocks. Here, B is the block number and b is the beam number. The yellow part represents the data that is chopped

This data is read from the shared memory and converted to a 2D matrix with time as x-axis and frequency as y-axis. The FRB shared memory is only accessible using C codes, but Candies is written in python. So, I have used pybind toolkit which can transfer data between C and python. After reading the required data from the shared memory, it is converted to a numpy array and passed into python code. This matrix is further processed by the original candies code to produce dedispersed dynamic spectrum and the DM transform.

5.1.2 Passing information between Candies and FETCH

Candies and FETCH are two separate software tools originally designed to work independently. Previously, FETCH would process the h5 files produced by Candies, performing further classification of candidates. However, in the real-time pipeline, I have integrated FETCH's classification functions, specifically the `get_model` function, directly into Candies.

This integration eliminates the need to write h5 files during intermediate steps. Instead, the feature information required for classification is passed through shared memory. Once clustering is complete, each candidate undergoes feature extraction and immediate classification. If a candidate is identified as a real signal, an h5 file is then written to disk, and the final CSV file is updated accordingly.

5.2 Testing and Results

The above algorithm for real-time candies and FETCH is still under development for the 800 beam detection pipeline. Even though the algorithm for real-time implementation is ready and working, the whole real-time system is not coherently connected. As discussed before, the real-time candies requires for the single pulse search to dump the candidate list. And the real-time candies and classification code picks up those candidates and classifies them. This is required to be carried out within the time period of the $\sim$ 400 second buffer. So, without a pipeline manager, which manages the coherence of the processes, it is not possible to carry out a full fledged test. Based on the current availability of a real-time 800 beam data stream, in order to test the above algorithm, I cut out a chunk ($\sim$ 33s) of data from the data stream and checked the presence of the observing pulsar and calculated it's period using PRESTO [53] to compare with the literature.

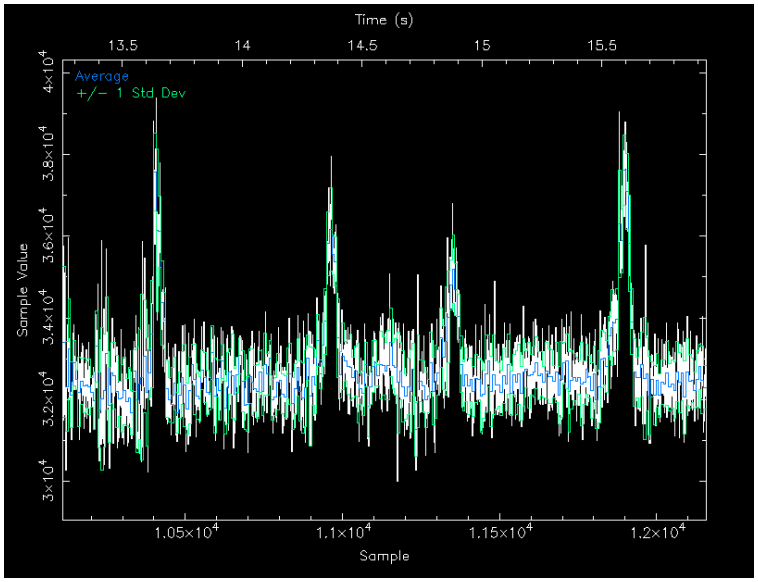

Figure 5.6: B0329+54 data folded and dedispersed at 26.83 pc cm^{-3}, generated by PRESTO

Figure 5.6, shows a part of the data chunk for pulsar B0329+54, cut out from the data stream and folded at 26.83 pc cm^{-3}. The calculated period of the pulsar is estimated to be $\sim$ 714 ms, which is similar to the period mentioned in the pulsar catalog [26]. Now that the data reading part is working fine, the creation of dedispersed dynamic spectrum and DM-transform, and finally classification was checked with a mock candidate list for the same real-time streaming data. The code was able to generate all the necessary plots and all the

mock candidates were detected as fake, as the exact DM, time-of arrival were not known through single pulse search. Now, that the proper functioning of the algorithm has been confirmed, the actual final tests with real candidate lists will be conducted once the system starts to run coherently.

Chapter 6

A coincidence and anti-coincidence spatial filtering algorithm

The coincidence and anti-coincidence filtering algorithms deal with spatial filtering of candidates across multiple beams spread over the whole field-of-view of the observation. As described in the previous chapter[4], the SPOTLIGHT pipeline deals with multi-beam pipeline. Even though, the pipeline includes RFI mitigation and clustering at the beam level to get rid of spurious candidates and false positives, this is not enough for a multi-beam system.

Coincidence filtering is a spatial filtering technique to remove multiple detections of the same candidate in the nearby beams. Since, the beams are tightly packed (for example - considering a field of view of 2 degrees, and 800 beams; a single occupies approximately a few arc-seconds). Bright transient pulses being extended sources light up multiple tightly packed beams at once. So, if there is a detection in the central beam, there will also be detections in the surrounding beams. The number of beams lighting up by a pulse depends on the fluence of the pulse. In figure 6.1, we can see how the SNR is distributed across different beams which one beam capturing the maximum SNR (the beam pointed towards the source) and the SNR decreasing away from it. The coincidence filter, picks out the same candidate (candidates observed in multiple beams with similar dispersion measure and time) from multiple beams and sorts them on the basis of SNR. Finally, picking up the maximum SNR candidate as the actual signal and removing the others. The generation of SNR maps for a pulse will be discussed in the next section.

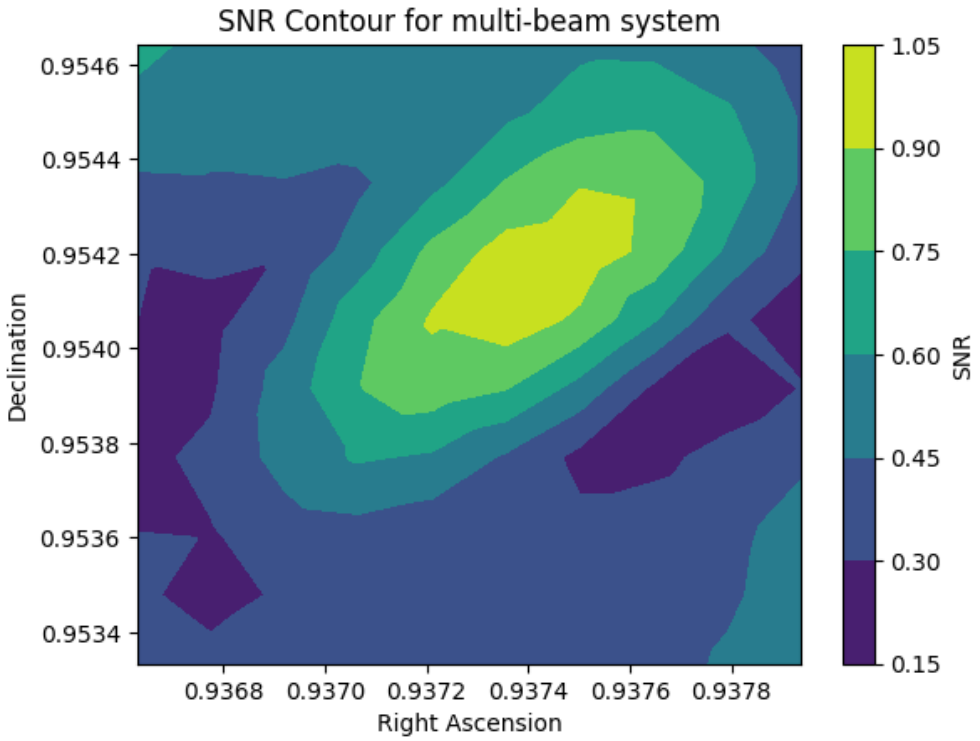

Figure 6.1: SNR map showing the distribution of SNR of detected pulse across the field of view

Anti-coincidence filtering is a spatial filtering technique to remove RFI present in different beams using spatial filtering. The main idea behind this filter is that RFI is mostly terrestrial is not localized spatially. A bright RFI pulse will be detected across multiple beams irrespective of how closely the beams are present. If similar candidate is observed across far away beams (separated by a few degrees), then they are classified as RFI by the anti-coincidence filter.

In a single beam pipeline we can let some bright spurious candidates to pass through and eventually get classified as RFI by the CNN classifier FETCH [2]. But, in a multi-beam pipeline, the number of these bright spurious candidates increase significantly. Since, currently, there are 16 nodes with each hosting 50 beams, it is not possible to perform both the coincidence and anti-coincidence filtering at the node level. The data from all the nodes need to be combined in a single node and then the spatial filtering can be carried out in full resolution.

6.1 Preliminary filtering algorithm

6.1.1 Data Generation

At the time of development of the first coincidence/anti-coincidence filter, SPOTLIGHT was a 100 beam system with offline data processing. So, the generation of a robust datasets for testing the filter required a few steps, as listed below:

- Using the 100 beam system, bright pulsars like B0329+54, B1929+10 [26] were observed across Band 3,4,5, over multiple observations. For each observation (for a particular source and frequency band), hundred filterbank files [36] containing the data were dumped for further processing.
- Even though bright pulsars are observed, they can't provide the high dispersion measure (DM) we are looking for in FRBs. So, we inject simulated FRBs into the pulsar data itself. But this injection is not as simple as single beam injection. The simulated FRB needs to be injected into all the 100 beams such that the SNR distribution for the pulse follows a natural pattern that is actually observed for a GMRT multi-beam system.
- In order to generate a natural SNR distribution across multiple beams, we use the pulsar data itself. Since, the pulsar signal itself is observed through the multi-beam system, calculating the SNR of the pulsar pulses across different beams gives a rough estimate for simulating the SNR distribution for the FRB.
- Since, pulsars are periodic pulses and not as bright as FRBs, the SNR calculation for pulsars requires the method of pulsar folding [38]. It is a technique to enhance the SNR of periodic pulsar signals buried in noise. The method includes adding multiple pulse periods by aligning the data based on the period of the pulse. This effectively averages out the noise, reinforcing the periodic signal.
- After pulsar folding, we are left with 100 SNR values, which can be used for injecting the simulated FRBs into the same pulsar dataset. The SNR distribution obtained from pulsar folding is depicted in figure 6.2b,6.3b
- In order to validate the SNR map obtained from pulsar folding we also generate a similar SNR map using a beam simulation code [46] as shown in figure 6.2a,6.3a. On

comparison, we can see that even though there are minute differences, the overall structure are similar.

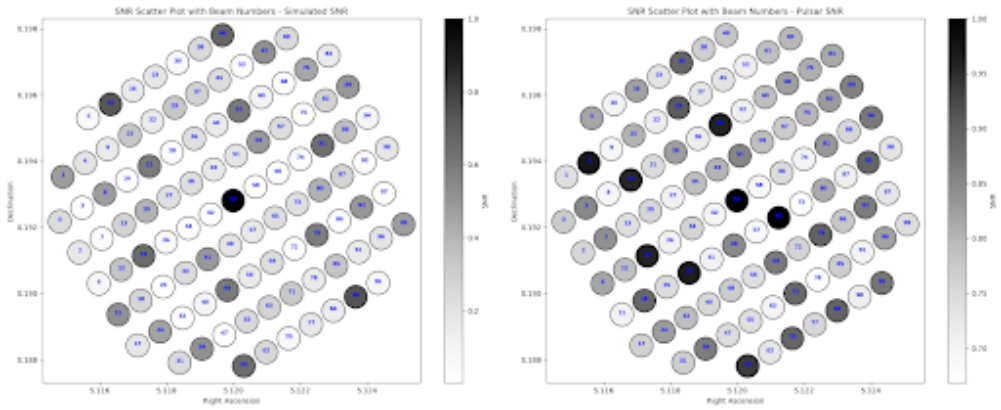

Figure 6.2: SNR distribution of pulsar B0329+54 (band3) across multiple beams: a)Simulated using beamforming code b)Obtained using pulsar folding

6.1.2 Filtering algorithm

The preliminary spatial filtering algorithm performs both the coincidence/anti-coincidence algorithm at the same time. The filtering algorithm consists of the following steps:

- Considering there are 100 beams data. Each beam data passes through AstroAccelerate and list of candidates containing information about dispersion measure, time of arrival, width, SNR, beam number is returned. Lets consider the 100 beams contain $(N_1, N_2, N_3, ..., N_{100})$ candidates.
- Each of those candidates are passed through an initial DM and width cutoff (similar to [4.2.1]) inorder to remove the spurious candidates.
- The candidates from each beam are passed through a DM-time clustering (similar to [4.2.2]) using DBSCAN algorithm. This beam level clustering is done to remove the spurious candidates and false positives from each beam. For each beam, we take the maximum SNR candidate from each cluster. This makes the number of candidates in each bea, to go from $(N_1, N_2, N_3, ..., N_{100})$ to $(M_1, M_2, M_3, ..., M_{100})$ (where M also corresponds to the number of clusters formed in each beam).

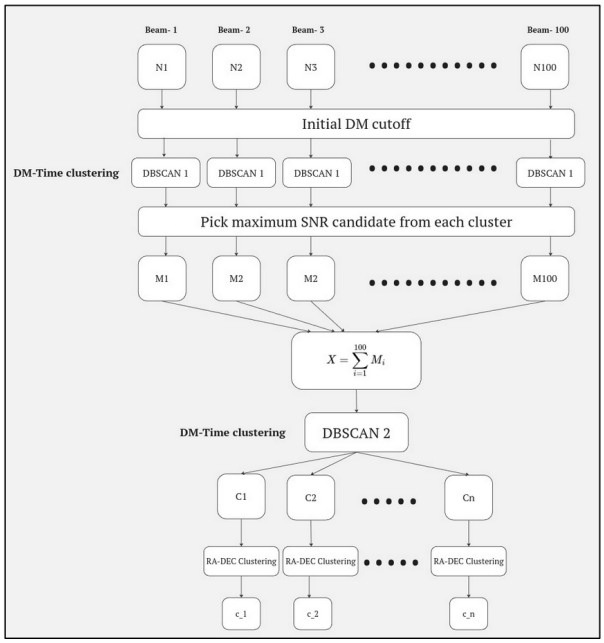

Figure 6.3: coincidence/anti-coincidence filtering algorithm

- Now the algorithm moves from beam level to global level. All the candidates from the 100 beams are combined into a common list. But now the candidate list also contains information about the position (RA, DEC) of each of the candidates (obtained from the beam). Another clustering is performed in DM-time. Let's consider we get 'n' clusters after this clustering, namely, $(C_1, C_2, C_3, ..., C_n)$

- For each cluster C_i, we perform a clustering in RA-DEC plane. Let the number of clusters formed for C_ith DM-time cluster in RA-DEC plane be c_i

- If $c_i = 1$: The C_i (i-th DM-time) cluster forms a single, closely spaced cluster in the RA-DEC plane. It is considered an actual candidate cluster, and the maximum SNR candidate from this cluster is selected for the final list. This is the coincidence filtering.

- If $c_i > 1$ or $c_i < 1$: The C_i cluster contains multiple or no clusters in the RA-DEC plane, with candidates spread far apart. It is not an actual candidate cluster and is rejected. This is the anti-coincidence filtering

6.2 Results and Discussion

6.2.1 Preliminary results

The following is result from one of the tests for observing B0329+54 over Band-4 (550 MHz-750MHz) using 4 antennas. The simulated FRB was injected at a dispersion measure of 500 pc cm^{-3} at a time of 30 seconds. All the 100 beams are processed and the below figure 6.4a shows the candidates present in one beam in DM-time.

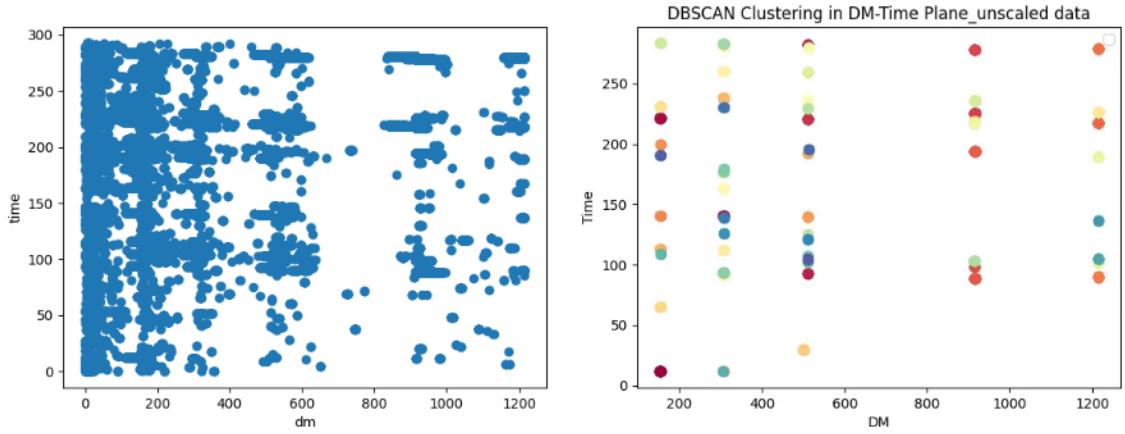

Figure 6.4: a) Candidates plotted in DM-time plane for Beam 49. The red line represents the DM cutoff. b)Candidates from multiple beams clustered together

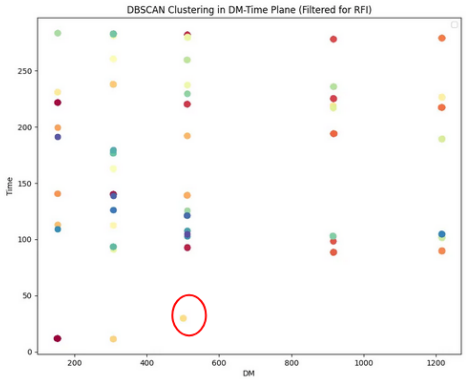

Figure 6.5: Final clusters after spatial filtering

After the maximum SNR candidate from each cluster in each beam are picked up, all

these candidates are clustered again in DM-time. The above figure 6.4b shows the merged list of candidates clustered in DM-time. The merged candidates form a total of 59 cluster in DM-time plane.

After carrying out the final RA-DEC clustering for each DM-time cluster, the number of actual candidate cluster goes down from 59 to 54. In the above figure it can be seen that the injected FRB candidate is getting detected in the end. Even though, the algorithm works on this dataset, the current above mentioned is not suitable for processing due to main reasons:

- The dataset used for these tests were not tightly packed over a bigger field-of-view. Since, the field-of-view was around $\sim$ 15 arc-minutes, anti-coincidence filtering was not that effective. So, we require a more robust dataset for the tests.
- The current algorithm does not account for patterns in the SNR map. Through thorough testing, it has been observed that the SNR distribution for transient sources often follows a distinct pattern, with high SNR values occasionally spreading across distant beams. This undermines the algorithm's effectiveness, as it relies on the assumption that a genuine candidate pulse cannot extend to far-away beams. As a result, the algorithm struggles to reliably differentiate between RFI and actual candidates.

6.2.2 Recent Development

In last last few months, after the SPOTLIGHT system came online with 800 beams, we have been able to get a lot of robust data to check the coincidence and anti-coincidence filtering algorithm. One of the main advantages of having 800 beams is the presence of a robust dataset covering a few degrees in the sky. By observing bright pulsars, we are checking the validity of the SNR map generation using pulsar folding and beam tilling code (figure-6.6). As it can be seen, the SNR map obtained from pulsar folding follows a more extended pattern than the SNR map obtained by beam tilling, which is just a theoretical prediction. In an ideal case, the residual between the two, should be zero throughout, but as we can see in figure 6.7a, the residual has a regions of bright peaks, suggesting the implicit differences between the two the SNR maps. Further, figure 6.7b was plotted by choosing the maximum SNR pulse from the whole 800 beam dataset, and then using the dispersion measure and time of arrival of that pulse to find the same pulse in all other beams. Using the SNR of

the same exact pulse recorded in different beams, the SNR map is generated. It can be seen that the brightest pulse is not observed throughout. This suggests that, having a rigorous algorithm like the previous one which uses the clustering of the position of the beams to carry out spatial filtering, is not the best choice. In order to solidify these findings, several other SNR maps were created for other datasets across different beams. Tests on off-axis sources (i.e the central beam doesn't point towards the source) were also performed.

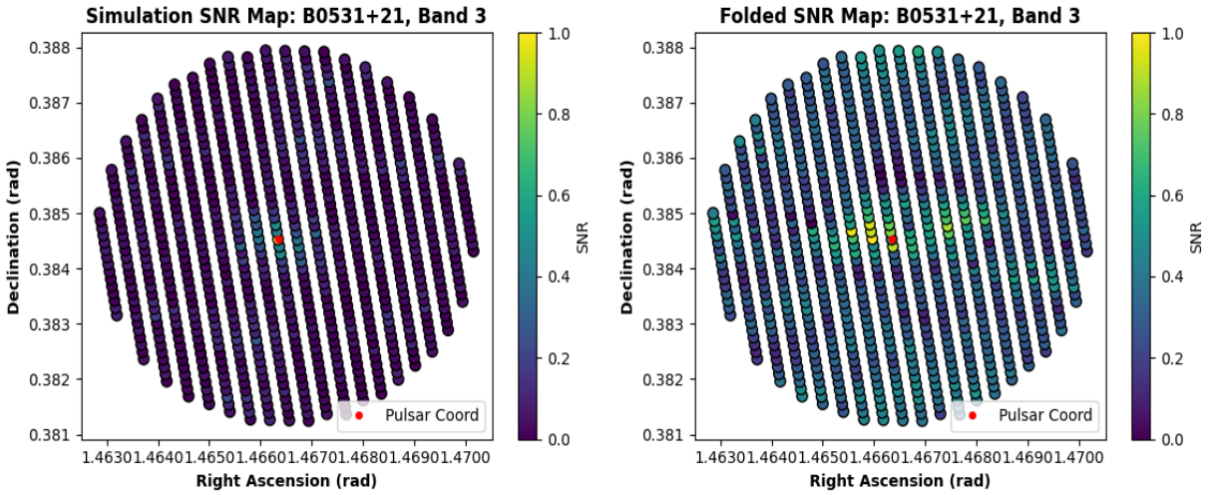

Figure 6.6: a) SNR map generated using beam tilling b) SNR map generated using pulsar folding

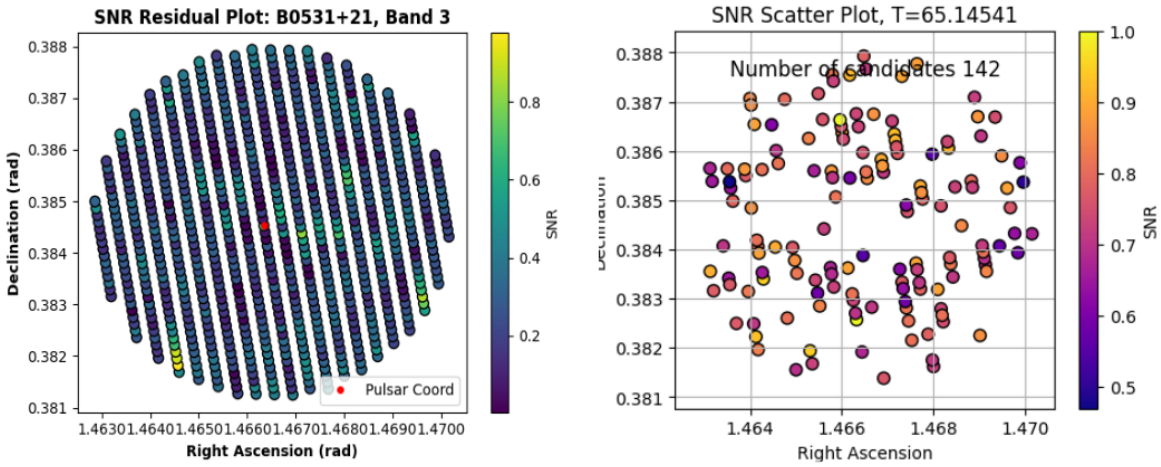

Figure 6.7: a) Residual SNR map b)SNR map from single pulse search

In order to solve this, we are trying out much simpler algorithms which are not as rigorous as before. The current algorithm separates the coincidence and anti-coincidence algorithm.

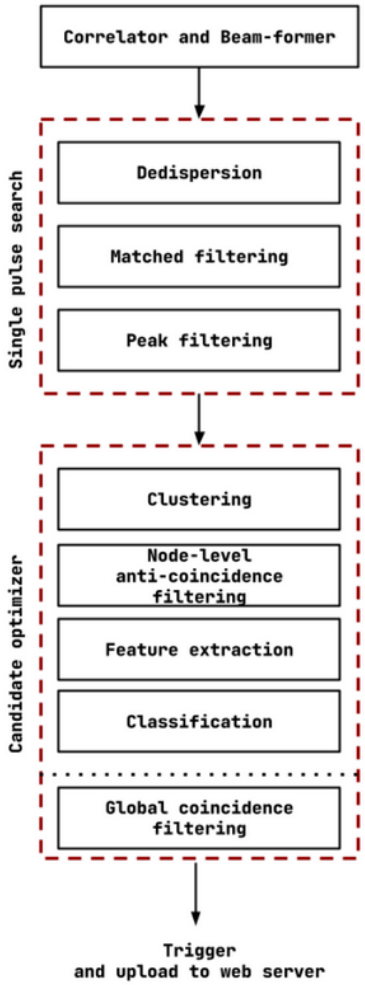

Figure 6.8: Updated flow for the detection pipeline

The previous anti-coincidence algorithm had a high chance of losing actual signals. So now, after clustering(showed in figure 6.8), the filtered candidates from each beam in a given node are combined to perform anti-coincidence filtering. Instead of looking for spatial positions, we remove candidates which are just observed in a single beam, as we saw before that actual pulses are observed over multiple beams. These candidates are most likely to be RFI. Even though, this is the most accurate technique, yet this doesn't lose actual candidates. Just by using such an approach, the number of false positives has seen to be decreasing by a factor of ~ 5. After, this the whole pipeline is run and after the classification of candidates

by the CNN classifier FETCH, the coincidence filter is run. The coincidence filter looks for similar candidates across all beams, and picks up the maximum SNR candidate.

Chapter 7

Conclusion

Currently, the SPOTLIGHT system is a 800 quasi real-time pipeline working over 16 nodes. The detection system which I have worked on, as described in the previous sections is running in quasi real-time (i.e. "X" hours of raw data are processed within the next "X" hours). Individual modules of the realtime pipeline that I have worked on including RFI mitigation, clustering, feature extraction (Candies) and classification, are ready and tested, but a pipeline manager to control the flow of the whole pipeline is still under development. Once, this is set up, all these modules will be able to run in real-time. A further modified and tested algorithm for coincidence and anti-coincidence filtering will also be developed in order to carry out better spatial clustering. Currently, various pulsars and known FRBs are being observed using the SPOTLIGHT system, but once the whole real-time pipeline is ready within the next few months, we will be able to search throughout the whole GMRT field of view and look for unknown FRBs. When the system is fully running, we will be able to run the detection pipeline on a total of $\sim$ 2000 beams, and be able to detect and arc-second localize FRBs in real-time. The deployment of the fully operational real-time pipeline will provide a robust dataset for studying compact objects and uncovering the underlying physics of the sources that generate FRBs. Additionally, it will advance our understanding of cosmology, as FRBs serve as powerful probes for exploring the structure and evolution of the universe. The real-time pipeline will be made accessible to the scientific community from the next observation cycle, offering more robust and impactful results. Our work has already been presented and published at prestigious international conferences such as FRB2024 and URSI. Furthermore, with GMRT serving as a pathfinder for the Square Kilometre Array (SKA),

the SPOTLIGHT system will strengthen both GMRT's capabilities and India's position as a key player in the future of radio astronomy.

Bibliography

[1] Karel Adámek and Wesley Armour. Single-pulse detection algorithms for real-time fast radio burst searches using gpus. *The Astrophysical Journal Supplement Series*, 247(2):56, 2020.

[2] Devansh Agarwal, Kshitij Aggarwal, Sarah Burke-Spolaor, Duncan R Lorimer, and Nathaniel Garver-Daniels. Fetch: A deep-learning based classifier for fast transient classification. *Monthly Notices of the Royal Astronomical Society*, 497(2):1661–1674, 2020.

[3] Mandana Amiri, Bridget C Andersen, Kevin Bandura, Sabrina Berger, Mohit Bhardwaj, Michelle M Boyce, PJ Boyle, Charanjot Brar, Daniela Breitman, Tomas Cassanelli, et al. The first chime/frb fast radio burst catalog. *The Astrophysical Journal Supplement Series*, 257(2):59, 2021.

[4] Radio Astronomy, A Tsioumis, W Ban, SH Chung, et al. Handbook, 2013.

[5] Matthew Bailes. The discovery and scientific potential of fast radio bursts. *Science*, 378(6620):eabj3043, 2022.

[6] P Beniamini, Z Wadiasingh, J Hare, KM Rajwade, G Younes, and AJ van der Horst. Evidence for an abundant old population of galactic ultra-long period magnetars and implications for fast radio bursts. *Monthly Notices of the Royal Astronomical Society*, 520(2):1872–1894, 2023.

[7] ND Ramesh Bhat, James M Cordes, Fernando Camilo, David J Nice, and Duncan R Lorimer. Multifrequency observations of radio pulse broadening and constraints on interstellar electron density microstructure. *The Astrophysical Journal*, 605(2):759, 2004.

[8] Cees Carels, Karel Adámek, Jan Novotnỳ, and Wesley Armour. Development of production-ready gpu data processing pipeline software for astroaccelerate. *arXiv preprint arXiv:1912.07704*, 2019.

[9] Hyerin Cho, Jean-Pierre Macquart, Ryan M Shannon, Adam T Deller, Ian S Morrison, Ron D Ekers, Keith W Bannister, Wael Farah, Hao Qiu, Mawson W Sammons, et al. Spectropolarimetric analysis of frb 181112 at microsecond resolution: implications for

fast radio burst emission mechanism. *The Astrophysical Journal Letters*, 891(2):L38, 2020.

[10] James M Cordes. Pulsars as probes of relativistic gravity, nuclear matter, and astrophysical plasmas. In *Solar, Stellar and Galactic Connections Between Particle Physics and Astrophysics*, pages 43–76. Springer, 2007.

[11] Fronefield Crawford, Kevin Stovall, AG Lyne, BW Stappers, David J Nice, IH Stairs, P Lazarus, JWT Hessels, PCC Freire, B Allen, et al. Four highly dispersed millisecond pulsars discovered in the arecibo palfa galactic plane survey. *The Astrophysical Journal*, 757(1):90, 2012.

[12] Cherie K Day, Adam T Deller, Ryan M Shannon, Hao Qiu, Keith W Bannister, Shivani Bhandari, Ron Ekers, Chris Flynn, Clancy W James, Jean-Pierre Macquart, et al. High time resolution and polarization properties of askap-localized fast radio bursts. *Monthly Notices of the Royal Astronomical Society*, 497(3):3335–3350, 2020.

[13] Weizhen Dong, Brian D Jeffs, and J Richard Fisher. A kalman-tracker-based bayesian detector for radar interference in radio astronomy. In *Proceedings.(ICASSP'05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 4, pages iv–657. IEEE, 2005.

[14] R P Eatough, E F Keane, and AG Lyne. An interference removal technique for radio pulsar searches. *Monthly Notices of the Royal Astronomical Society*, 395(1):410–415, 2009.

[15] Tarraneh Eftekhari and Edo Berger. Associating fast radio bursts with their host galaxies. *The Astrophysical Journal*, 849(2):162, 2017.

[16] Rob P Fender, AM Stirling, RE Spencer, I Brown, GG Pooley, TWB Muxlow, and JCA Miller-Jones. A transient relativistic radio jet from cygnus x-1. *Monthly Notices of the Royal Astronomical Society*, 369(2):603–607, 2006.

[17] John M Ford and Kaushal D Buch. Rfi mitigation techniques in radio astronomy. In *2014 IEEE Geoscience and Remote Sensing Symposium*, pages 231–234. IEEE, 2014.

[18] PA Fridman. Statistically stable estimates of variance in radio-astronomy observations as tools for radio-frequency interference mitigation. *The Astronomical Journal*, 135(5):1810, 2008.

[19] PA Fridman and WA Baan. Rfi mitigation methods in radio astronomy. *Astronomy & Astrophysics*, 378(1):327–344, 2001.

[20] DE Gary, GM Nita, X Wang, and H Ge. Spectral domain algorithms for rfi excision in real time. In *Proceedings of the URSI General Assembly*. Citeseer, 2008.

[21] Yashwant Gupta, B Ajithkumar, HS Kale, S Nayak, S Sabhapathy, S Sureshkumar, RV Swami, JN Chengalur, SK Ghosh, CH Ishwara-Chandra, et al. The upgraded gmrt: opening new windows on the radio universe. *Current Science*, pages 707–714, 2017.

[22] CGT Haslam, CJ Salter, H Stoffel, and WEz Wilson. A 408 mhz all-sky continuum survey. ii-the atlas of contour maps. *Astronomy and Astrophysics Supplement Series, vol. 47, Jan. 1982, p. 1, 2, 4-51, 53-142.*, 47:1, 1982.

[23] Kasper E Heintz, J Xavier Prochaska, Sunil Simha, Emma Platts, Wen-fai Fong, Nicolas Tejos, Stuart D Ryder, Kshitij Aggerwal, Shivani Bhandari, Cherie K Day, et al. Host galaxy properties and offset distributions of fast radio bursts: implications for their progenitors. *The Astrophysical Journal*, 903(2):152, 2020.

[24] JWT Hessels, LG Spitler, AD Seymour, JM Cordes, D Michilli, RS Lynch, K Gourdji, AM Archibald, CG Bassa, GC Bower, et al. Frb 121102 bursts show complex time–frequency structure. *The Astrophysical Journal Letters*, 876(2):L23, 2019.

[25] Antony Hewish. Pulsars. *Scientific American*, 219(4):25–35, 1968.

[26] G Hobbs, R Manchester, A Teoh, and M Hobbs. The atnf pulsar catalog. In *Young Neutron Stars and Their Environments*, volume 218, page 139, 2004.

[27] Russell A Hulse and Joseph H Taylor. Discovery of a pulsar in a binary system. *Astrophysical Journal, vol. 195, Jan. 15, 1975, pt. 2, p. L51-L53.*, 195:L51–L53, 1975.

[28] CW James, JX Prochaska, JP Macquart, FO North-Hickey, KW Bannister, and A Dunning. The fast radio burst population evolves, consistent with the star formation rate. *Monthly Notices of the Royal Astronomical Society: Letters*, 510(1):L18–L23, 2022.

[29] Victoria M Kaspi and Andrei M Beloborodov. Magnetars. *Annual Review of Astronomy and Astrophysics*, 55(1):261–301, 2017.

[30] Santaji N Katore, Aditya Chowdhury, and Nissim Kanekar. The exposure time calculator for the upgraded giant metrewave radio telescope. 2020.

[31] EF Keane. Classifying rrats and frbs. *Monthly Notices of the Royal Astronomical Society*, 459(2):1360–1362, 2016.

[32] EF Keane, DA Ludovici, RP Eatough, M Kramer, AG Lyne, MA McLaughlin, and BW Stappers. Further searches for rotating radio transients in the parkes multi-beam pulsar survey. *Monthly Notices of the Royal Astronomical Society*, 401(2):1057–1068, 2010.

[33] XJ Li, XF Dong, ZB Zhang, and D Li. Long and short fast radio bursts are different from repeating and nonrepeating transients. *The Astrophysical Journal*, 923(2):230, 2021.

[34] Malcolm S Longair. High energy astrophysics: vol. 1. *Comments on Astrophys., Vol. 17, p. 259-261 (1994)*, 17:259–261, 1994.

[35] Malcolm S Longair. *High energy astrophysics*. Cambridge university press, 2011.

[36] DR Lorimer. Sigproc-v1. 0:(pulsar) signal processing programs. Technical report, Arecibo Technical Memo, 2001.

[37] Duncan R Lorimer, Matthew Bailes, Maura Ann McLaughlin, David J Narkevic, and Froney Crawford. A bright millisecond radio burst of extragalactic origin. *Science*, 318(5851):777–780, 2007.

[38] Duncan Ross Lorimer and Michael Kramer. *Handbook of pulsar astronomy*, volume 4. Cambridge university press, 2005.

[39] R Luo, BJ Wang, YP Men, CF Zhang, JC Jiang, H Xu, WY Wang, KJ Lee, JL Han, Bing Zhang, et al. Diverse polarization angle swings from a repeating fast radio burst source. *Nature*, 586(7831):693–696, 2020.

[40] Andrew Lyne and Francis Graham-Smith. *Pulsar astronomy*. Number 48. Cambridge University Press, 2012.

[41] RN Manchester. The parkes multibeam pulsar survey and interstellar scattering. In *Sources and Scintillations: Refraction and Scattering in Radio Astronomy IAU Colloquium 182*, pages 33–38. Springer, 2001.

[42] Alexandra G Mannings, Wen-fai Fong, Sunil Simha, J Xavier Prochaska, Marc Rafelski, Charles D Kilpatrick, Nicolas Tejos, Kasper E Heintz, Keith W Bannister, Shivani Bhandari, et al. A high-resolution view of fast radio burst host environments. *The Astrophysical Journal*, 917(2):75, 2021.

[43] Benito Marcote, Kenzie Nimmo, JWT Hessels, SP Tendulkar, CG Bassa, Z Paragi, A Keimpema, M Bhardwaj, R Karuppusamy, VM Kaspi, et al. A repeating fast radio burst source localized to a nearby spiral galaxy. *Nature*, 577(7789):190–194, 2020.

[44] Ben Margalit, Edo Berger, and Brian D Metzger. Fast radio bursts from magnetars born in binary neutron star mergers and accretion induced collapse. *The Astrophysical Journal*, 886(2):110, 2019.

[45] Maura A McLaughlin, AG Lyne, DR Lorimer, M Kramer, AJ Faulkner, RN Manchester, JM Cordes, F Camilo, A Possenti, IH Stairs, et al. Transient radio bursts from rotating neutron stars. *Nature*, 439(7078):817–820, 2006.

[46] Mekhala Muley, Sanjay Kudale, Nishant P Deo, et al. Optimal tiling of spotlight field-of-view with multi-beam synthesis.

[47] Jan Novotnỳ, Karel Adámek, MA Clark, Mike Giles, and Wes Armour. Accelerating dedispersion using many-core architectures. *The Astrophysical Journal Supplement Series*, 269(1):29, 2023.

[48] Keiron O'shea and Ryan Nash. An introduction to convolutional neural networks. *arXiv preprint arXiv:1511.08458*, 2015.

[49] Emily Petroff, LC Oostrum, BW Stappers, Matthew Bailes, ED Barr, S Bates, S Bhandari, NDR Bhat, MARTA Burgay, S Burke-Spolaor, et al. A fast radio burst with a low dispersion measure. *Monthly Notices of the Royal Astronomical Society*, 482(3):3109–3115, 2019.

[50] Ziggy Pleunis, Deborah C Good, Victoria M Kaspi, Ryan Mckinven, Scott M Ransom, Paul Scholz, Kevin Bandura, Mohit Bhardwaj, PJ Boyle, Charanjot Brar, et al. Fast radio burst morphology in the first chime/frb catalog. *The Astrophysical Journal*, 923(1):1, 2021.

[51] Danny C Price, Benjamin R Barsdell, and Lincoln J Greenhill. Hdfits: Porting the fits data model to hdf5. *Astronomy and Computing*, 12:212–220, 2015.

[52] Kaustubh M Rajwade and Joeri van Leeuwen. A needle in a cosmic haystack: A review of frb search techniques. *Universe*, 10(4):158, 2024.

[53] Scott Ransom. Presto: pulsar exploration and search toolkit. *Astrophysics source code library*, pages ascl–1107, 2011.

[54] Erich Schubert, Jörg Sander, Martin Ester, Hans Peter Kriegel, and Xiaowei Xu. Dbscan revisited, revisited: why and how you should (still) use dbscan. *ACM Transactions on Database Systems (TODS)*, 42(3):1–21, 2017.

[55] G Swarup. Surveys of radio frequency interference (rfi) at the gmrt site from terrestrial transmitters part iv. Technical report, NCRA-TIFR, 2001.

[56] Shriharsh P Tendulkar, Armando Gil de Paz, Aida Yu Kirichenko, Jason WT Hessels, Mohit Bhardwaj, Fernando Ávila, Cees Bassa, Pragya Chawla, Emmanuel Fonseca, Victoria M Kaspi, et al. The 60 pc environment of frb 20180916b. *The Astrophysical Journal Letters*, 908(1):L12, 2021.

[57] Nithyanandan Thyagarajan, David J Helfand, Richard L White, and Robert H Becker. Variable and transient radio sources in the first survey. *The Astrophysical Journal*, 742(1):49, 2011.

[58] Bingchen Wang, Chenglong Zhang, Lei Song, Lianhe Zhao, Yu Dou, and Zihao Yu. Design and optimization of dbscan algorithm based on cuda, 2015.

[59] Kurt W Weiler, Nino Panagia, Marcos J Montes, and Richard A Sramek. Radio emission from supernovae and gamma-ray bursters. *Annual Review of Astronomy and Astrophysics*, 40(1):387–438, 2002.