J. Astrophys. Astr. (2025) 46:26
https://doi.org/10.1007/s12036-025-10052-0

Innovative web tool for remote data acquisition and analysis: Customized for SKA low frequency beamforming test bed LPDA array at Gauribidanur Radio Observatory

ANUMANCHI AGASTYA SAI RAM LIKHIT[1,2,*], KATTA NAVEEN[1,2], B. ARUL PANDIAN[1], R. ABHISHEK[1] and T. PRABU[1]

[1]Electronics Engineering Group, Raman Research Institute, Bengaluru 560080, India.
[2]Department of Physics, Indian Institute of Science Education and Research Bhopal, Bhopal 462066, India.
*Corresponding author. E-mail: astropi.2003@gmail.com

MS received 20 June 2024; accepted 21 January 2025

Abstract. With the evolution of radio astronomy and related education and training, the demand for scalable, efficient, and remote systems in data acquisition, storage, and analysis has significantly increased. Addressing this need, we have developed a web interface for a log-periodic dipole antenna array integral to the SKA Test activities at the Gauribidanur Radio Observatory (77.428 E, 13.603 N). This interface, employing Python-based technologies such as Streamlit and PyVISA, along with Standard Commands for Programmable Instruments (SCPI) commands, offers a seamless and user-friendly experience. Our solution introduces a unique data acquisition approach, employing SCPI through Python to communicate with the setup's data acquisition system. The web interface, accessible remotely via a secure WLAN network or VPN, facilitates user-initiated observations and comprehensive logging and offers advanced features like manual radio frequency interference masking, transit plotting, and fringe plot analysis. Additionally, it acts as a data hub, allowing for the remote downloading of observational data. These capabilities significantly enhance the user's ability to conduct detailed post-observation data analysis. The effectiveness of this interface is further demonstrated through a successful solar transit observation, validating its utility and accuracy in real-world astronomical applications. The applications of this web tool are expandable. They can be tailored according to the Observatory's goals and instrumentation, as well as the growing radio astronomy instrumentation and observing facilities at various educational institutions.

Keywords. Radio-astronomy—remote web interface—log periodic dipole antennas—SKA low—Gauribidanur Radio Observatory—Streamlit.

1. Introduction

The field of radio astronomy has witnessed a rapid evolution, driven by the quest to explore the universe more deeply. We also observe the increased awareness and capabilities to set up small-scale radio telescope observational facilities that are coming up at the diverse educational institutes specializing in astronomy. This evolution brings with it a demand for advanced observational capabilities, particularly in terms of data acquisition, storage, and analysis. A significant challenge in this domain is the development of systems that are not only efficient and scalable but also capable of remote operation, given the often inaccessible locations of radio observatories.

These expanding needs, particularly at the Gauribidanur Radio Observatory, a hub of important astronomical research, are what drive our work. A sophisticated approach is required to conduct observations and handle the data remotely using the observatory's log-periodic dipole antenna (LPDA) array, which is used for Square Kilometre Array (SKA) India (Gupta *et al.* 2023) test activities. Traditional methods often fall short in terms of efficiency, user-friendliness, and flexibility, especially when it comes to remote accessibility.

Recognizing these challenges, our goal was to create a solution that addresses these technical demands and enhances the overall user experience for astronomers and researchers. A significant response to this need emerged with developing a web interface that utilizes

Published online: 10 May 2025

state-of-the-art Python-based technologies like Streamlit and PyVISA, along with Standard Commands for Programmable Instruments (SCPI). The design of this interface streamlines the process of data acquisition. It offers robust features for manual radio frequency interference (RFI) detection and masking and transit plotting – all accessible remotely, a crucial feature aiding modern astronomical research.

The design presented in this paper has been developed and successfully operational in the observatory for the past six months for a new broadband two-element radio interferometer. The hardware details of the interferometer implementation are discussed in detail in Dora *et al.* (2024) and this paper presents the salient aspects of the novel web-based remote data acquisition and data analysis features developed for the interferometer.

Astronomy communities have developed various software tools for monitoring and operating radio telescopes. We describe two SKA precursor MWA telescope tools: the Common Gateway Interface (CGI-BIN) based wMARC (Gupta 2011) receiver commissioning and NAGIOS-based telescope monitoring tools. The receiver commissioning required executing several discretely developed standalone codes and OS scripts in a flexible sequence. The wMARC integrated these executables to provide a customizable menu-driven environment, and a remote client's web browser could invoke the executables using HTTP and HTTPS calls. The tool was used extensively during the MWA telescope's commissioning phase. The Nagios (2009) is an event monitoring system typically valuable for monitoring servers, network switches, and applications. A Nagios-based system was used to monitor the status of the MWA telescope functions. Inherent to the Nagios is extensive logging of the events, system configuration retrieval, and alert mechanisms with an interface that includes email message alerts and authorization mechanisms. This system uses CGI-bin and plugins. Plugins are external executables written in any language. Sometimes, a CGI-based implementation concerns sensitive networks, exposing the system to external programs.

The tool presented in this work provides comprehensive functionalities for remotely operating a simple radio interferometric telescope and collecting and analyzing both current and archived data. It is based on a free and open-source framework, the Streamlit, which is popular among machine learning and data science web applications.

The paper's outline is as follows: Section 2 presents the background of our work, focusing on the LPDA we employed. Section 3 elaborates on the signal flow within our observatory's model telescope setup. Section 4 discusses the data acquisition methods discovered, detailing the chosen observation approach. Section 5 covers the data processing pipeline for the collected data. The design and implementation of our web interface are outlined in Section 6, with Section 7 explaining its functionalities. Section 8 addresses testing and validation of the web tool, including its application in actual astronomical observations. We conclude by reflecting on the web tool's success and its potential for broader application.

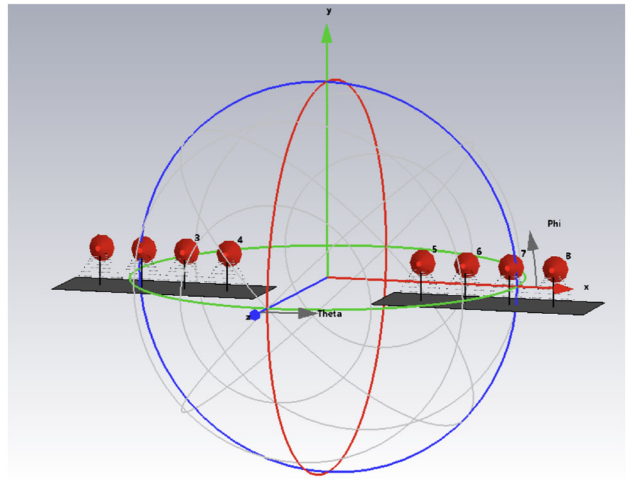

Figure 1. LPDA array at Gauribidanur Radio Observatory.

2. Log periodic dipole antenna array

At the Gauribidanur Radio Observatory, we have a set-up of eight LPDAs (these are similar to what was mentioned in Raghunathan *et al.* 2023), which are organized into two elements, each containing four antennas. As shown in the image in Figure 1, the antennas are aligned east-west and operate within the 150–350 MHz range, overlapping with the upcoming SKA Low radio telescope.

The baseline of the two-element antenna setup spread over 13.1 m along East-West (consisting of about 5 m and 5.3 m for two elements, with a free space of about 2.8 m between the elements). Using the actual parameters of the setup, we simulate each element in an antenna simulation software CST. The simulation shows, at a frequency of 200 MHz, the main lobe is directed towards the zenith with a magnitude of 11.6 dBi and an angular width of 18.1° at the 3 dB point.

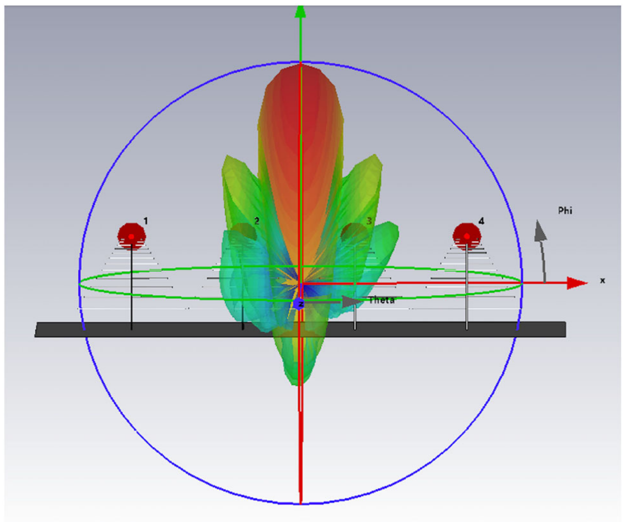

Figure 2. Simulated 3D far-field radiation pattern of a single element at 200 MHz.

The side lobe measured level is −12.7 dB, which indicates effective sidelobe suppression as it is sufficiently below the main lobe peak. The two-element beam pattern obtained from the simulation is shown in Figure 2.

These characteristics are crucial for the accurate interpretation of the observed radio signals and for ensuring that the primary reception is from the intended direction of a celestial source. We aim to create a web tool that can efficiently manage all the observations conducted on this log periodic dipole antenna array. This tool includes tasks such as data collection, storage, and processing. Before exploring the web tool, let us review the RF path of the setup.

3. RF signal path for the log periodic dipole antenna array

The signal processing chain for each element of the LPDA at the Gauribidanur Radio Observatory is designed in the following way: This RF path begins with the collection of radio signals by four LPDAs, which are engineered to operate over a broad frequency range. These antennas are particularly sensitive to the target frequency band of 150 to 350 MHz, making them ideal for the observatory's research purposes.

Signals from the celestial radio sources are very weak, so we use the above-mentioned sensitive RF path (Figure 3) to preserve the signal strength for the analysis. Upon capture by the antennas, the signals are individually amplified by custom-designed low-noise

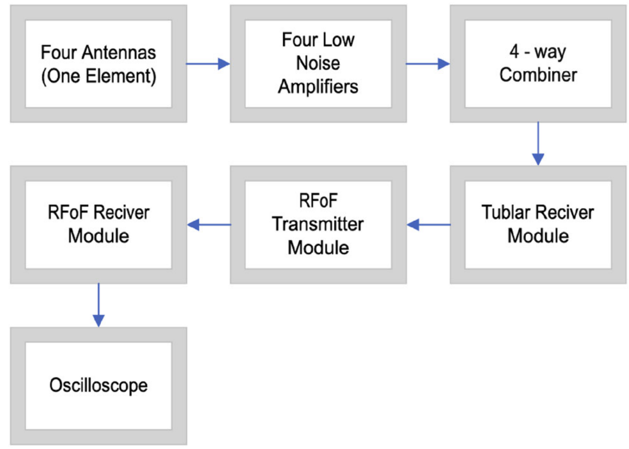

Figure 3. Radio frequency signal path tailored for an element.

amplifiers (LNA). These LNAs, custom-developed in-house, are critical in boosting signal strength while minimizing the system noise, thereby preserving the quality of the received astronomical signals. Following initial amplification, the signals from the four antennas are combined using a 4-way combiner. This process increases the overall signal strength and enhances the signal-to-noise ratio.

The combined signal is then processed through a front-end receiver module, another custom component designed at RRI, containing a series of bandpass filters, high-pass filters, amplifiers, and attenuators. These components collaboratively filter out frequencies outside the intended band, provide further amplification, and adjust the signal level, ensuring the output signal is precisely within the 150 to 350 MHz band with a gain of +53 dB. Next, the signal is sent to a Radio Frequency over Fiber (RFoF) transmitter module, which converts the RF signal into an optical signal to be transmitted over a fiber-optic cable. This transmission method preserves the signal quality over long distances with minimal loss. The corresponding RFoF receiver module then converts the optical signal back into an RF signal for electronic processing.

Finally, the RF signal is fed into an oscilloscope, which visualizes the signal in the time domain. This allows for real-time observation and analysis of the signal's properties, such as amplitude and frequency. The oscilloscope can be utilized to confirm the functionality of the complete RF channel by examining the signal properties and confirming they correspond to the anticipated 150 to 350 MHz frequency range and the overall amplification of 63 dB.

Thus, the RF path is carefully engineered to capture, amplify, filter, and analyze radio signals within a

Figure 4. Tektronix MSO3054 mixed signal oscilloscope-backend for our observations.

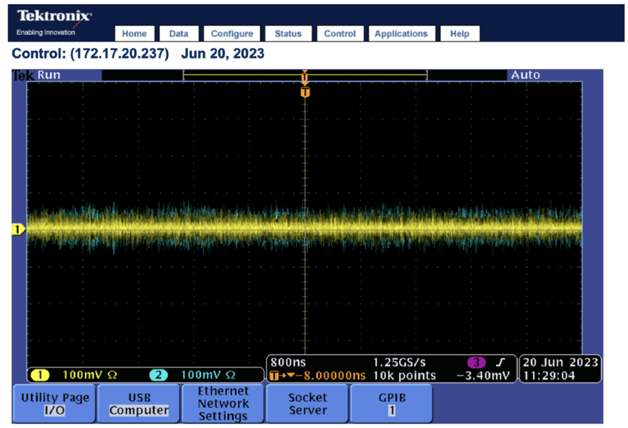

Figure 5. e*Scope feature of the oscilloscope.

specific frequency range, enabling astronomical observations in the SKA Low range at the Gauribidanur Radio Observatory using the LPDA array.

4. Data acquisitions methods for log periodic dipole antenna array

Accurate data gathering and control are essential for observing transient phenomena continuum source transits and for tracking them. The Tektronix Mixed Signal Oscilloscope MSO-3054,[1] shown in Figure 4, served as a crucial data gathering device for our observation.

The RF signals from each antenna element are fed into two RF input channels of this oscilloscope. The oscilloscope is connected to the local network via Ethernet, enabling remote access. In the following subsections, we present two innovative methods for data acquisition that we utilized in our work.

*1. Automated remote control with e*Scope:* One of the primary challenges in our data acquisition was obtaining dual-channel data synchronously to facilitate correlation and phased array mode of observations. We developed a method to address this challenge, utilizing the oscillipscope's built-in remote viewing and control feature, e*Scope, as shown in Figure 5.

As mentioned earlier, an Ethernet cable connects the oscilloscope to the internet. The web interface of the oscilloscope allows access to remote control of the data. This interface allows saving waveform as a comma-separated-value (CSV) data format file containing data for all channels synchronously gathered simultaneously. To achieve continual data acquisition over extended periods, such as full-day observations, we wrote a Python program utilizing the Selenium python package (Selenium 2021; Gojare *et al.* 2015), which provided a robust connection to the oscilloscope's web interface. This package helps to integrate the oscilloscope's front panel operations into a Python program script. By analyzing the webpage's structure, we discover the universal resource locator (URL) for the 'Save Waveform' button and automate the task of clicking it at regular intervals. The user could set the start time, end time, and interval, and the program would automatically open the URL, initiating the save waveform action accordingly. The oscilloscope saved the gathered two-channel data as CSV format data files into a USB storage device. As the data gathering continues, the oscilloscope saves each scan data in a separate file along with the time information. The oscilloscope named the files sequentially, such as $tek\_ALL\_0000.csv$, $tek\_ALL\_0001.csv$, and so on. The oscilloscope collects the data without manual intervention, ensuring accurate timing for repeated scans.

While this method allowed for precise data collection at simultaneous intervals for two channels, it did come with specific limitations. Firstly, the USB storage capacity could become a constraint depending on the size of the data files and the duration of the observation. Secondly, the oscilloscope can only save up to 9999 CSV files, after which it will start rewriting the first file. This means careful planning is required for very long observations.

2. Direct data transfer using PyVISA: The second method of data acquisition was designed to overcome the limitations of the first, providing a more flexible and robust solution, particularly for longer observations. This approach utilized the run/stop command of

[1] More about Tektronix Mixed Signal Oscilloscope MSO-3054 in the Appendix.

the oscilloscope, offering a novel way to collect dual-channel data synchronously at the same time.

By using the run/stop button, the oscilloscope is made to freeze by holding the gathered signal waveform data at a particular time t1, effectively stopping the live visualization and subsequent acquisition of the signals until a run command is used. Once the signal gathering is stopped, the waveform data of each channel is transferred to the computer using PyVISA (Grecco *et al.* 2023), a Python-based instrument control package. After data transfer, the live signal is promptly resumed by pressing the run/stop button, then paused again after a period of t2 is repeated throughout the observation. This data transfer based on PyVISA appeared to be faster.

This repeated run/stop requires specific minimum time gaps to be adhered to maintain data integrity. The time constraints come from the following elements: (a) Time to gather one record signal waveform, typically a few thousand to a million times the chosen sampling rate in the oscilloscope, (b) Time to transfer the recorded data over the internet as standard TCP packets, and (c) The oscilloscope's internal turnaround time to restart a new acquisition.

5. Data processing

The data gathering pipeline collects the data at regular intervals over the selected period. It saves the data in a CSV format file for each acquisition data packet with voltage readings. Each data packet contains 10,000 measurement points or voltage readings. We organized these packets into arrays for each channel, creating extensive datasets ready for analysis. We divided these packets into smaller segments, each with 1,024 data points, resulting in nine segments per packet.

To convert this data from the time domain to the frequency domain, we applied the Real Fast Fourier Transform (RFFT). The 'Real' in RFFT indicates the inputs are only real numbers, optimizing the computation. This method converts the time-domain signal into its constituent frequencies, producing 513 output points per segment. We discarded the first point to eliminate the DC voltage component, leaving 512 frequency bins. Each bin represents a bandwidth of 1.22 MHz, calculated by dividing the oscilloscope's 0 to 625 MHz range by 512.

We then calculated the signal's power by squaring each RFFT point's magnitude. By averaging these squared values, we obtained a 512-point array for each acquisition, which measures signal power. To express this power in dBm, a standard unit for spectral density, we took the -10 logarithm of these values. Averaging the power across the same frequency bins produced a spectrum, and averaging across all bins for each packet gave us a time series of signal power.

5.1 *Correlation algorithm*

The correlation analyzes the relationship between signals captured by two different channels (CH1 and CH2) from the antenna array. We divide the data from each channel into smaller segments to facilitate detailed analysis. We specifically break down each packet of data containing 10,000 voltage readings into segments of 1024 points. For each segment, we apply the RFFT to convert the time-domain signal into the frequency domain. This step is crucial because it allows us to analyze the frequency components of the signal. In this process, we discard the first frequency channel of the RFFT output to eliminate the DC component, focusing on the remaining frequency data.

After computing the RFFT for each segment, we calculate the power by taking the square of the magnitude of the Fourier-transformed data. We then average this power across all segments within each packet to obtain a representative power spectrum for each packet.

To measure the similarity between the signals from the two channels, we compute the cross-correlation of the Fourier-transformed data. This involves multiplying each segment's Fourier-transformed data from CH1 by the complex conjugate of the corresponding segment from CH2. The resulting correlated data measures the similarities between the two signals, allowing us to identify any time delays or phase differences. Finally, we calculate the average of the correlated data's absolute values for each packet. This mean value summarizes the correlation between CH1 and CH2, which is critical for understanding the relative positioning and timing of the observed signals.

5.2 *Radio frequency interference masking*

Radio frequency interference is the unwanted disturbance in the observed radio signal that is caused by man-made sources like satellites and local electronic equipment. These interferences can corrupt the data collected by the telescope, making it challenging to extract meaningful information. Effective RFI mitigation is crucial in radio astronomy to ensure that the data accurately reflects the true signals from the sky, free from unwanted noise sources.

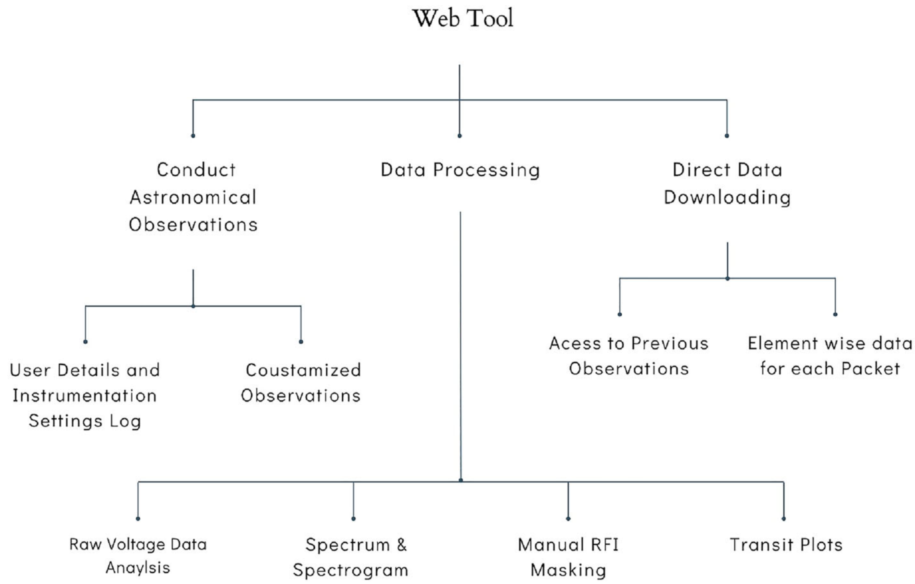

Figure 6. Outline of the web tool functionalities.

To address RFI in our data processing pipeline, we focused on analyzing the power of each frequency bin over time. Specifically, heat maps are plotted to represent the power levels across different frequency bins throughout the observation period. These heat maps let us visually identify the bins and corresponding time intervals where RFI was present. Once we identified these regions, we manually mask the affected bins by setting their values to zero.

For instance, we observed RFI from satellites and interference from the local oscillator, which introduced a consistent, predictable noise in specific bins. By manually identifying and masking these interferences, we were able to reduce their impact on the data. However, while this manual masking process is effective, it is labor-intensive and leaves room for improvement. There is significant potential for automating this process using machine learning techniques (Mosiane *et al.* 2017), which could identify and mitigate RFI more efficiently.

6. Design and implementation of web tool

To address the challenges of remote data acquisition and analysis, we developed a novel web tool that allows simultaneous data acquisition from the two elements at regular intervals and is remotely accessible from the Raman Research Institute campus. A virtual private network (VPN) can also be accessed globally. The system offers a range of functions, including data downloading, logging of observations and instrument settings, data processing, and RFI masking. All these features are seamlessly integrated into a user-friendly web interface, providing a comprehensive solution for data management needed in radio telescope observations.

Streamlit (Streamlit 2019) is an open-source Python library that simplifies creating web applications for data science and machine learning. With its user-friendly design, Streamlit allows data scripts to be transformed into shareable web apps. Given these features, we chose Streamlit to design a web tool that serves as a hub for observations, data storage, and data analysis. The platform offers numerous advantages, such as rapid prototyping, interactive analysis, extensive customization options, and ease of use. Making it an ideal choice for our project.

We developed the Python-based web tool using Streamlit, which is hosted on a local system at the Gauribidanur Radio Observatory. We connected it to remote users on the main campus via WLAN.

A schematic representation of the network configuration between the Gauribidanur Radio Observatory and the Raman Research Institute in Figure 7. The diagram illustrates the data flow from the observatory's oscilloscope to the web tool hosted on a local PC, accessible via a secure WLAN or remotely through a VPN.

7. Functionalities of web tool

The web tool offers various functionalities, from initiating observations to conducting advanced data analysis

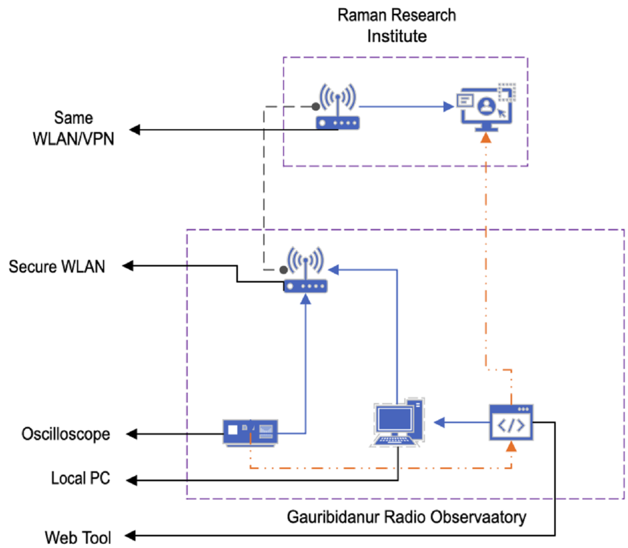

Figure 7. Schematic representation of the network configuration.

and plotting results. The main page of the web interface displays an image of the LPDA array, welcoming users and inviting them to navigate through three distinct sections: (i) data acquisition, (ii) data center, and (iii) data processing as shown in Figure 8. The following text delves into each of these sections of the web tool to explore what they offer to the user.

7.1 *Data acquisition*

This section of the web tool enables users to initiate observations with the antenna array. Users can manage various aspects of the observation, such as setting the start and end times of the observation, selecting the number of data channels to collect (in our case, we have two elements), and adjusting the time delay between acquisitions based on the specific requirements of their observations. Also, we have to enter a name for the observation. Using the name, a folder is created to save the collected data.

Upon entering this page, the user is prompted to input their username. Figure 9 illustrates the user interface requesting the username entry. After successful verification, options to start the observation become available, as shown in Figure 10. The web tool is designed to track and log all inputs the user provides into a log file, continuously appending to a CSV file. Furthermore, it communicates with the oscilloscope, generating a new text file with detailed oscilloscope settings at the start of each observation. These files are distinctly labeled with the date and time of the observation and are stored on the local system.

Access to these files is restricted to administrators, allowing them to review previous observation details. This level of inspection into user inputs and oscilloscope settings provides comprehensive information about the observations conducted. The various features incorporated in the Web Tool allow the users to customize their observation needs. The Web Tool has been regularly used for periodic remote observations for a long time.

7.2 *Data center*

This section of the web tool allows users to log and review data from previous observations. Users can view individual files for each observation.

Additionally, as depicted in Figure 11, there are options to download individual files or to download all files from an observation in a zipped folder. This functionality enables users to exclusively download data collected from the antenna array at the Gauribidanur Radio Observatory and conduct their analysis.

7.3 *Data processing*

This section of the web tool enables users to process observational data as shown in Figure 12. It offers the option to select data from a specific observation for processing, including full data analysis or a quick check by selecting a partial dataset. Users can also choose the FFT segment size for processing. Upon clicking the ‘Start Processing’ button, the tool begins processing the data and displays results such as initial raw voltage checks, time plots for each channel’s raw voltage data, and voltage histograms of an acquisition. Subsequently, it presents the Power Spectrum for each channel and correlated data. It also generates plots of average power across All frequencies with time for Channels 1 and 2 and correlated data, where transit plots can be observed in the absence of RFI.

Additionally, the tool provides Spectrograms for Channels 1 and 2 and correlated data. These plots, created using the Plotly library, allow users to zoom into specific regions of the spectrogram, aiding in identifying RFI-affected bins in the correlated spectrum.

The advanced data processing feature permits users to specify the number of frequency bin regions to mask and to enter their ranges. After processing with the RFI-affected bins set to zero, the tool produces updated results, including Average Power Across All Frequencies With Time for correlated data, the updated correlated spectrogram, and the fringe pattern at 200 MHz. This enables users to view transit plots with RFI regions

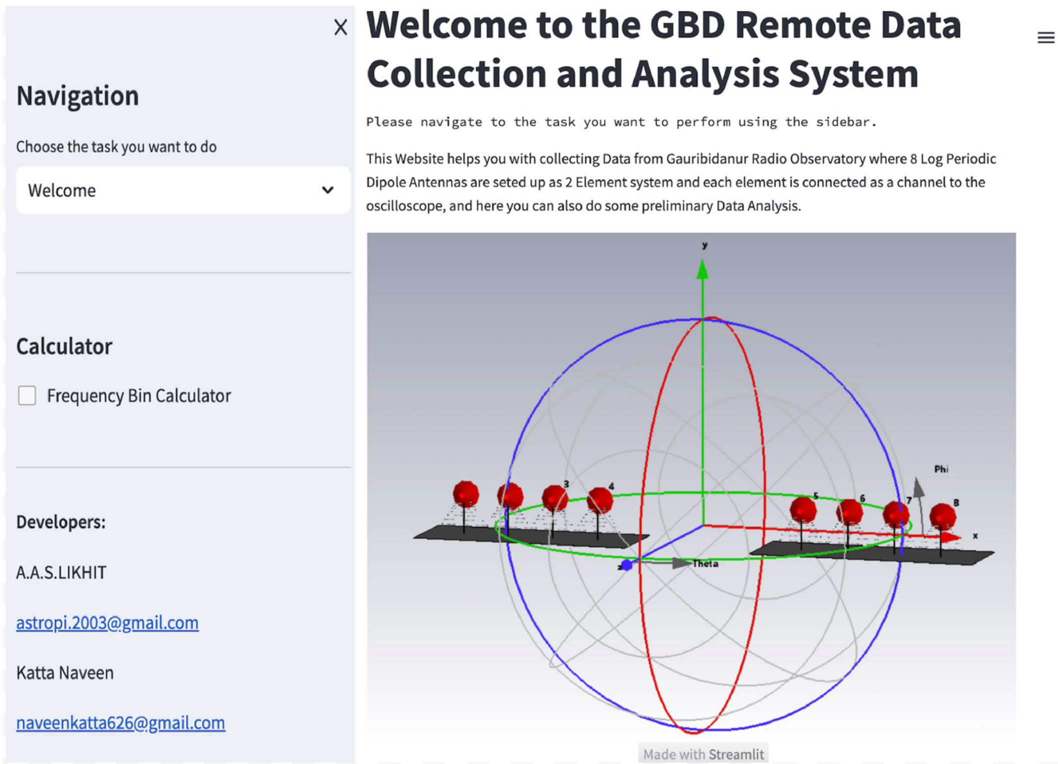

Figure 8. Home page of the Gauribidanur radio observatory LPDA observation tool (GLOT), displaying the navigation bar with available features and the interactive 3D visualization of the array setup.

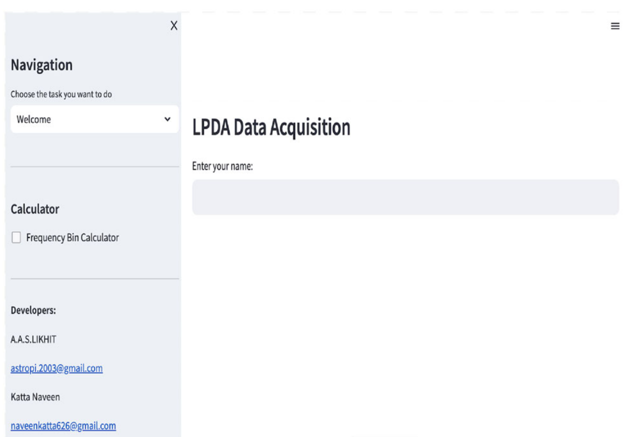

Figure 9. Initial user authentication prompt asking the user for their name to proceed further to start an observation.

Figure 10. Data acquisition section of the web page with prompts for user inputs for the observation settings needed to start the data acquisition.

removed and observe the fringe pattern at 200 MHz frequency.

7.4 *Operational flow*

The web tool's functionality is outlined in Figure 6, which depicts its key components: conducting astronomical observations, data processing, and direct data downloading. Detailed workflows for each component are further illustrated in Figures 13–15, representing the respective operational phases of data acquisition, data center, and data processing.

Figure 13 (data acquisition) outlines the initial phase, where users establish the settings for capturing data. It starts with the user entering their name and the fixed IP address of the Oscilloscope, ensuring a stable connection for data flow. The user then inputs the number

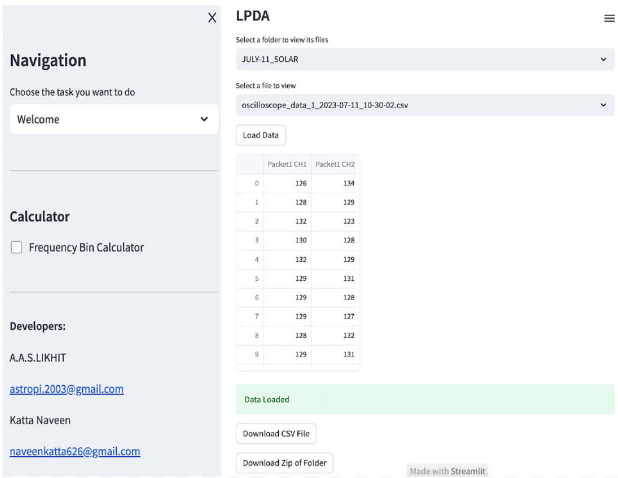

Figure 11. Data center section of web tool displaying the data of JULY-11 SOLAR observation folder.

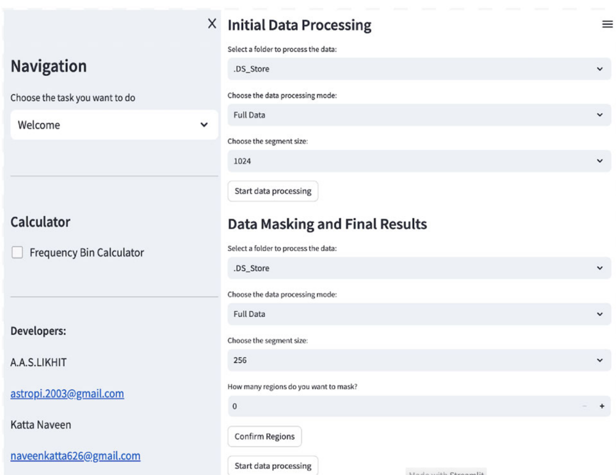

Figure 12. Data processing section of web tool displaying the options for the users input to process the desired observation data.

of channels (elements present) and specifies the observation's timeframe using a start and end timestamp (YYYY-MM-DD HH-MM format). Finally, the user sets the interval between data captures within the observation period and names a directory to save the data before initiating data acquisition.

Figure 14 (data center) describes the data retrieval and selection process. The user accesses the data center, where they select a folder containing available observational data. Options are provided to download the entire dataset as a zip file or to pick an individual CSV file for specific data points. Post-selection, there is a provision to either view the raw data directly within the web tool or to download the individual CSV file for offline use or further analysis.

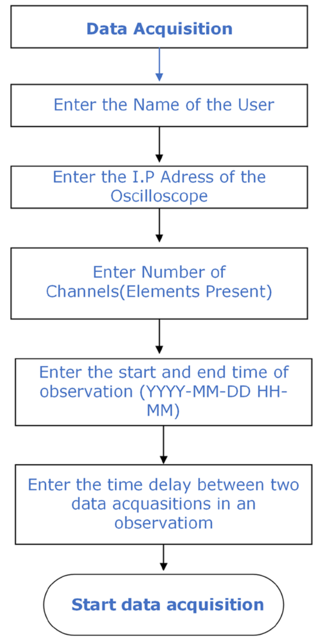

Figure 13. Operational flow of the data acquisition section.

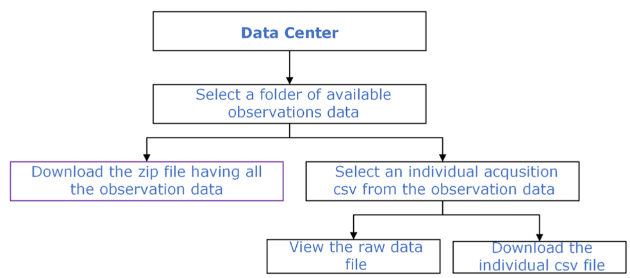

Figure 14. Operational flow of the data center section.

Figure 15 (data processing) describes the comprehensive steps for analyzing the gathered data. It begins with choosing the desired data analysis method – full or quick. For a quick analysis, the user selects a starting and ending CSV file from observation and then proceeds where in full all the data will be taken for the analysis, enters the segment size for performing Fast Fourier Transform (FFT), and inputs parameters for RFI masking, including the number of regions and their corresponding frequency bins. After confirming these regions, the user can proceed to process the data. The final output includes detailed spectrograms and transit plots, providing insights into the analyzed observations.

8. Testing and validation

During the testing and validation phase, we conducted an observation specifically aimed at capturing a solar transit and present here results from our observations

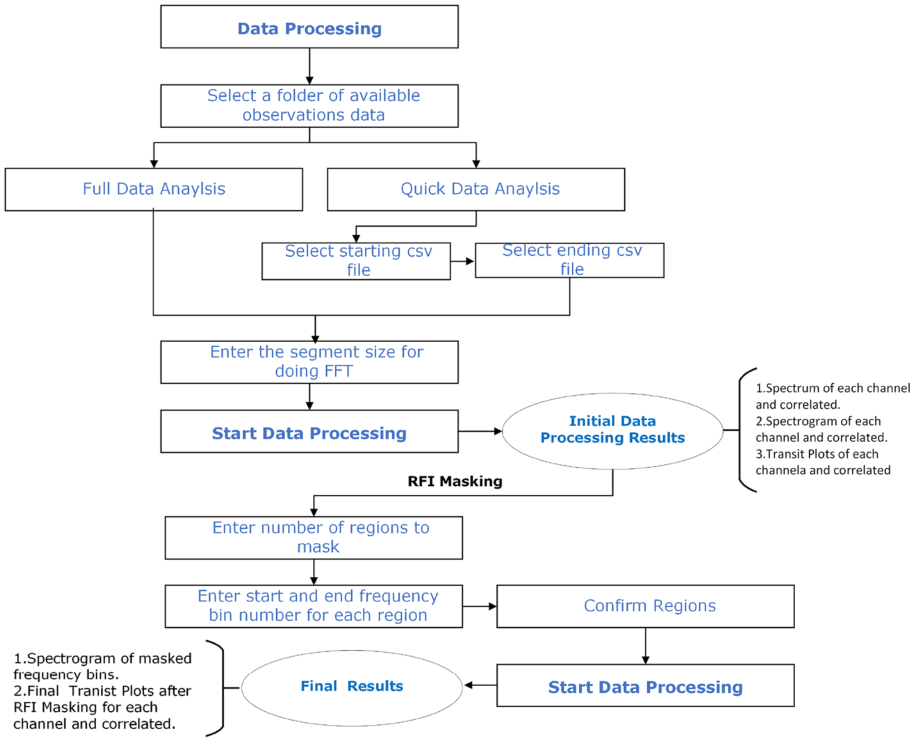

Figure 15. Operational flow of the data processing section.

of 11 July 2023. This event served as an essential test case to evaluate the web tool's capabilities in a real-world scenario.

The data was successfully acquired using the web tool remotely from the Raman Research Institute, located about 85 Km from the observatory, and the data was seamlessly displayed and promptly displayed in the data center section of the interface. This demonstrated not only the tool's data handling efficacy but also allowed for immediate dataset downloading.

Furthermore, the data was processed directly within the web interface, resulting in successfully generated plots that visualized the solar transit, as shown by the time series and histogram in Figure 16, and the power spectrum and spectrogram in Figures 17 and 18.

Figure 19 shows the plot before RFI removal, and Figure 20 shows the refined plot with minimal RFI interference, obtained by specifying frequency-bin regions to mask using advanced data-processing features. These comprehensive tests confirmed the web tool's functionality for both basic and advanced data analysis tasks.

During the past six months, the tool has successfully operated daily for solar transit and galactic plane transit observations. This extended period of operation has provided substantial evidence of the web tool's reliability and performance, demonstrating automated its robustness in consistent astronomical data acquisition and analysis. The consistent results, including the average power across all frequencies with time and the spectrogram analysis, are illustrated in Figures 18 and 19, respectively.

9. Web tool adaptability and future scope

The web tool's features and functionalities make it versatile and useful for radio astronomy observations at various remote locations. It integrates seamlessly with the instrumentation by connecting to the IP address of the data acquisition system (DAS), such as an oscilloscope. The code is open-source and available on GitHub (Astropi 2023), allowing users to modify the data processing techniques and even add more elements (antennas) to the array based on their specific observation goals. This flexibility ensures that users from diverse observatories can adapt the tool to their unique setups.

Furthermore, the web tool has the potential to be integrated into larger, more complex systems like the SKA. The SKA Observatory provides several support tools available on platforms like GitLab, such as the SKA-OST Sensitivity Calculator and the SKA-OST Simulation for Low Station Behavior (SKAO 2024). These tools are primarily designed for tasks like simulation of results, data processing, and observation management. While they focus on post-observation

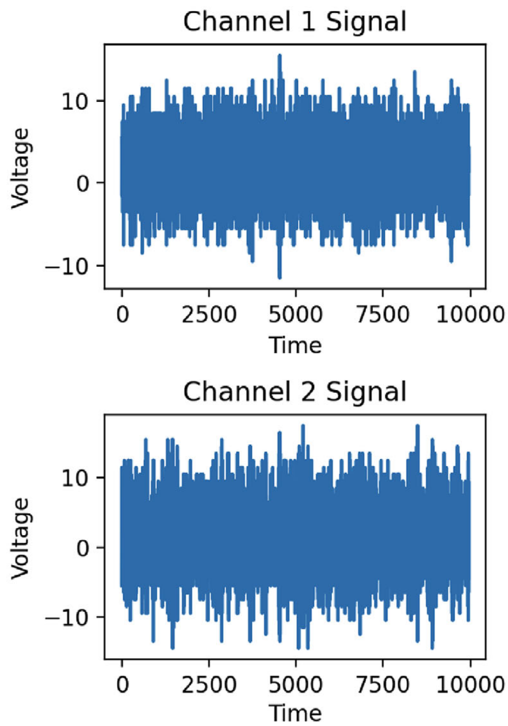

Figure 16. Raw voltage monitoring feature.

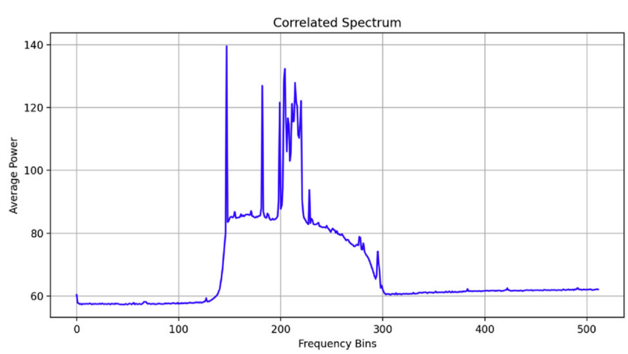

Figure 17. Power spectrum.

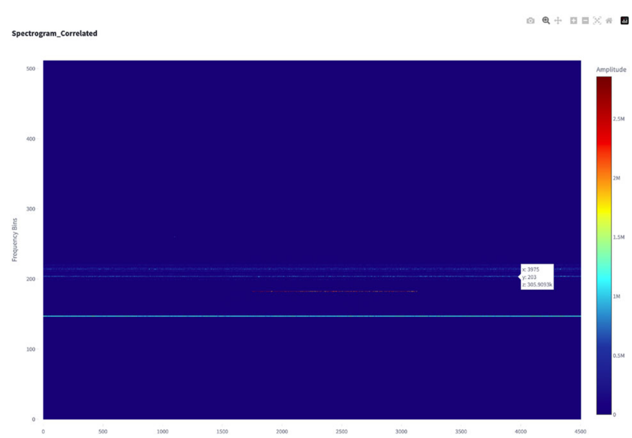

Figure 18. Spectrogram.

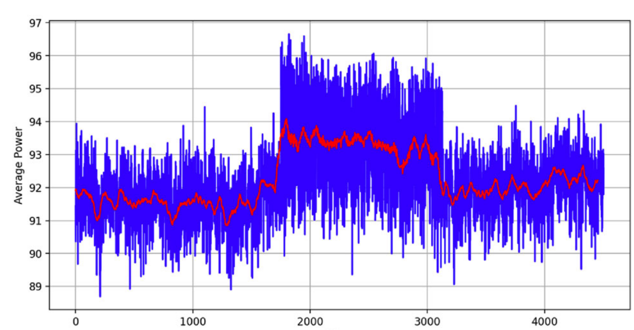

Figure 19. Average power across all frequencies with time for correlated data before RFI masking.

analysis and sensitivity simulation, our web tool is designed to interact directly with the data acquisition hardware in real time.

By integrating with SKA's existing tools, our web tool can complement the SKA infrastructure. For example, SKA tools like the sensitivity calculator simulate and process data after acquisition, but our tool can handle real-time data acquisition, RFI mitigation, and initial processing. This allows observatories to modify the data acquisition process and tailor data collection techniques according to the specific requirements of their observations.

Additionally, the web tool allows observatories to store their data locally on systems or hard disks, removing the need for costly cloud storage solutions likeAmazon AWS or Microsoft Azure, which some

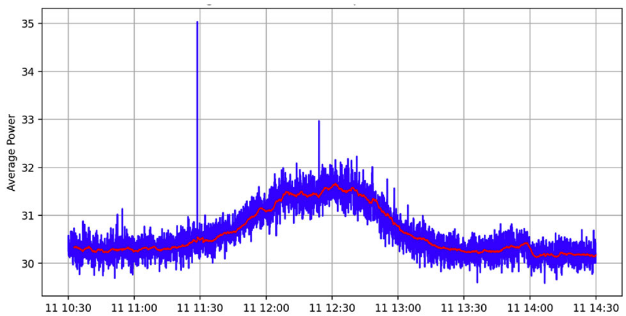

Figure 20. Average power across all frequencies with time for correlated data after RFI masking.

modern observatories commonly use for data handling. Our tool significantly reduces the operational costs associated with data management by offering a local storage option. Smaller observatories or educational institutions, lacking the budget for large-scale cloud infrastructure, can particularly benefit from this, as it guarantees secure access and processing of data for research purposes.

9.1 *Security measures*

The web tool works securely within the observatories using a secure WLAN network. This network ensures that only authorized personnel can access the DAS, minimizing the risk of external or unauthorized access.

We propose linking the web tool to a scheduling system via Google Forms as a future upgrade. This integration would allow users to schedule observations by submitting forms, with access restricted to specific Gmail accounts. We can then use the responses from Google Forms to automatically update the web tool's settings in the Streamlit app, allowing us to schedule and manage observations. This method would improve the tool's functionality and ensure that only authorized users with permitted Gmail accounts can schedule and access observations, improving the tool's overall security.

10. Conclusion

In this paper, we have presented the design details of an innovative Web tool meant for remote data acquisition and data analysis. This tool is developed for use with the SKA beamforming-related testbed based on the LPDA array at the Gauribidanur Radio Observatory. This web tool is primarily designed to address the typical needs of a remote observatory. Hence, its usability is anticipated for the many upcoming small radio astronomy facilities in educational institutions. We have presented the salient aspects of this work in this paper. This paper's web tool and ideas can be adapted for the different upcoming facilities. We also foresee using such remote tools for the forthcoming SKA antenna commissioning in the Western Australian deserts.

In conclusion, the web tool we created for the Gauribidanur Radio Observatory significantly addresses the observatory's operational requirements. This tool uses recent developments in data science and technology to offer a scalable and efficient solution for astronomical research. For further details and access to the tool, please refer to the Gauribidanur Radio Observatory LPDA array observation tool (GLOT) repository (Astropi 2023).

Looking ahead, the utility of systems like the one developed here is not limited to current needs but is expected to grow with upcoming projects such as the SKA. These tools are essential for modern research and offer a scalable base that can evolve with the field's future requirements. The deployment of our web tool not only addresses the immediate needs of the Gauribidanur Radio Observatory and supports the ongoing advancement of radio astronomy, enhancing research capabilities and encouraging further progress in the discipline.

Acknowledgements

We would like to thank the open-source libraries that played an important role in developing our web tool. We used PyVISA for instrument communication, allowing seamless interaction with the data acquisition hardware. Streamlit provided the framework for building a user-friendly web interface, allowing for simple data visualization and remote access. Finally, we utilized Selenium for web automation, facilitating efficient interaction with the oscilloscope's web interface.

Appendix. Tektronix MSO-3054

Tektronix MSO-3054 has a bandwidth of 500 MHz; the MSO 3054 provides the necessary frequency range for our observations. The oscilloscope features four analog and sixteen digital channels, allowing for simultaneous data acquisition from multiple sources. A sample rate of 2.5 Gs/s on all channels ensures high-resolution sampling of the signals, while the record length of 5 Megapoints on all channels offers substantial memory for recording waveform data. The MSO 3054also boasts a waveform capture rate of greater than 50,000 waveforms per second, capturing transient events precisely. Furthermore, it is equipped with USB for quick storage, a built-in Ethernet port for network connections, VISA connectivity for remote control, e*Scope for remote viewing and control, and Wave Inspector controls for managing long record lengths with zoom and pan functions, play and pause features, and search and mark capabilities.

References

Astropi B. 2023, Gauribidanur Radio Observatory LPDA Array Observation Tool (GLOT) [Computer software; GitHub Repository]. Available at: https://github.com/astropi-b/GLOT

Dora R. K., Likhit A. A. S. R., Naveen K. *et al.* 2024, in the 42nd meeting of the Astronomical Society of India (ASI), Vol. 42, P01

Gojare S., Joshi R., Gaigaware D. 2015, Procedia Computer Science, 50, 341. Big Data, Cloud and Computing Challenges

Grecco H. E., Dartiailh M. C., Thalhammer-Thurner G., Bronger T., Bauer F. 2023, Journal of Open Source Software, 8, 5304

Gupta S. 2011, wMon – A Web-Based Monitoring and Control Tool [Computer Software; GitHub Repository]. Available at: https://github.com/whoamishashi/wMARC_v1.1

Gupta Y., Bhattacharya D., Choudhury T. R., Wadadekar Y., Prabu T. 2023, J. Astrophys. Astron., 44, 27

Mosiane O., Oozeer N., Aniyan A., Bassett B. 2017, IOP Conf. Series. Mater. Sci. Eng., 198, 012012

Nagios 2009, Nagios: The Industry Standard in IT Infrastructure Monitoring [Computer Software]. Available at: https://www.nagios.com

Raghunathan A., Satish K., Sathyamurthy A. *et al*. 2023, J. Astrophys. Astron., 44, P43

Selenium H. Q. 2021, A browser automation framework and ecosystem, https://github.com/SeleniumHQ/Selenium

SKAO 2024, SKA Observatory Support Tools, https://gitlab.com/ska-telescope/ost

Streamlit 2019, Open-source Python framework for data scientists and AI/ML engineers to deliver dynamic data apps, https://github.com/streamlit/ streamlit