A Data Science Approach to Operations Data at the ALMA Observatory

Haley Egan, Matthew Litz, Diana Morris

Abstract

This study aims to optimize the data pipeline and reduce downtimes at the Atacama Large Millimeter Array (ALMA) observatory by employing data science and machine learning techniques. We analyzed three primary data sources—Jira tickets, weather data, and Shiftlog data—and identified challenges and inconsistencies in the current data pipeline. Using BERT models for Natural Language Processing (NLP), we successfully classified Jira tickets into hardware and software issues and categorized downtime types in the Shiftlog data. Using weather data, we created a classification of weather downtimes by cause, and identified a recovery time for each weather downtime. NLP was employed again to predict the root cause of hardware/software problems in Jira tickets. Survival analysis curves were prepared to help ALMA anticipate the time required to fix a downtimes caused by hardware and software problems. Our findings highlight the potential for machine learning-driven insights to significantly improve ALMA's operational efficiency, ultimately benefiting the broader scientific community. Future work should focus on refining the ML models, standardizing data entry, and integrating these insights into ALMA's daily operations. More information, including source code, can be found on the project Github.

Introduction

The Atacama Large Millimeter Array (ALMA) is an international astronomical research observatory located in Chile. ALMA specializes in observing millimeter and submillimeter wavelengths, offering unparalleled insights into the universe. As a vital scientific asset, the observatory continuously seeks ways to enhance its operational efficiency.

Various factors contribute to downtimes at ALMA, including weather, technical, and software issues. These downtimes hinder research observations, reducing the time available for scientific investigation. ALMA aims to minimize downtimes by leveraging data-driven strategies, machine learning, and data science. By employing these approaches, ALMA hopes to predict weather trends that cause downtimes, schedule observations based on future weather conditions, and plan maintenance in response to predicted hardware failures. Our team of data scientists conducted a study to support ALMA in achieving these goals. Our study focused on the following objectives:

1. Understand the types of data ALMA has collected over their decade of operations;
2. Identify the most valuable data for achieving ALMA's long-term data-driven operational goals;
3. Determine the most effective techniques and tools for organizing, cleaning, and analyzing their data;
4. Employ these techniques for exploratory data analysis, data cleaning, data preparation, and analysis;
5. Recognize gaps, roadblocks, and challenges within the current data and data pipeline;
6. Conduct initial machine learning modeling and predictions to establish baselines and ground truth;
7. Outline a roadmap towards an optimized data pipeline and the development of predictive systems;
8. Determine baseline performance and estimate future improvements to forecast cost savings and increased observational hours per year.

Data

Shiftlog Data

The shiftlog data consists of daily reports detailing operations and experiments performed at ALMA. This dataset contains information about the antennas and arrays used for each experiment, the types of downtime issues that occurred (weather, technical, scheduling), automatic executions, manual executions, and research shifts. The dataset has 16,472 rows and 49 columns, spanning the years 2012-2022.

Jira Ticket Data

ALMA uses Jira tickets to report, track, manage, and solve technical and software-related operational problems. Not all Jira tickets are related to downtimes. A ticket is filed when a technical or hardware problem occurs, but it is typically only classified as an official downtime in the Shiftlog data if the problem cannot be solved within 10 minutes. Jira tickets are initially labeled 'PRTSPR' for 'problem reports' and are later manually classified as either 'ICT' tickets for software

issues or 'PRTSIR' for hardware-related issues. There are a few other ticket classifications, but they were not relevant to our study. Each Jira ticket contains summary details, a description section, and comments. The description section may contain tables, screenshots, and/or written text, while the summary section contains short written text without a specific format or criteria. We received the Jira ticket data in the form of JSON files, with one JSON file corresponding to one Jira ticket. We were given 22,522 JSON files, spanning from 2013-2023. Since the majority of the text describing downtimes occurs in the Jira tickets, rather than the Shiftlog data, the Jira tickets were focused on for NLP.

Weather Data

The weather data consists of over a million data points from nine weather stations located within the center of the array field and surrounding the operations facility. Humidity readings, temperature, pressure, wind direction, and wind speed, PWV and phase RMS were collected every six minutes from 2011-2022, although various stations reported only intermittently. Data from the center of the array was reported as two different stations, one reporting from 2011-2015 and the other from 2015-2022. The most reliable station is at the technical building, where the correlators are located.

These three data sources were chosen because they offer the most potential for providing a holistic view of ALMA's daily operations and insights into downtimes. The data includes numerical, time series, and text data, resulting in a wide range of analysis, modeling, and predictive potential. Collectively, the shiftlog data, Jira tickets, and weather data have the potential to contribute significantly to ALMA's long-term data-driven operational goals.

Data Processing and Exploratory Data Analysis

Shiftlog Data

Exploratory data analysis was performed on the Shiftlog data to get a general sense of daily operations and identify areas needing further exploration.

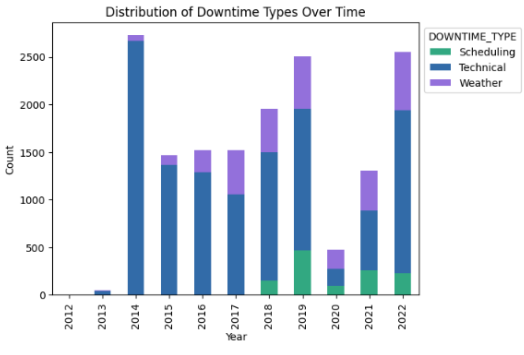

Figure 1: Distribution of downtime types over time

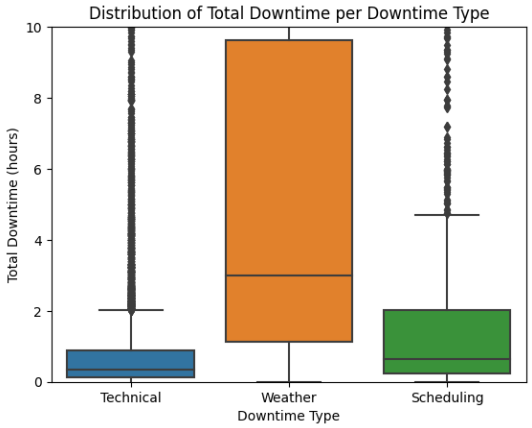

Figure 2: Distribution of downtime duration within shifts per downtime type. While weather downtimes are less common than technical, they often last much longer.

Jira Tickets

Each JSON file for each Jira ticket was parsed and cleaned in Dataiku. Some JSON files were long and complex, resulting in tables with hundreds of rows. Due to this complexity, only key features were extracted, including ticket id, ticket summary, description, assignee, start and end dates, location, etc. This information was extracted for each ticket, resulting in a dataset where each ticket is a row, and the key contents are columns.

Exploratory data analysis was performed to understand the Jira data, where hardware tickets are labeled by ALMA as 'PRTSIR', and software tickets labeled as 'ICT'. The data that was used for modeling consists of 6,940 PRTSIR tickets, and 673 ICT tickets. The summary and description columns were cleaned, removing symbols, non-important punctuation, and stopwords. Tokenization, keyword and key phrase extraction, topic modeling, and LDA were performed to obtain an overall sense of the data (key results results are documented in the appendix). These steps were repeated for each ticket type, allowing for a focused inspection of software-related tickets and hardware-related tickets.

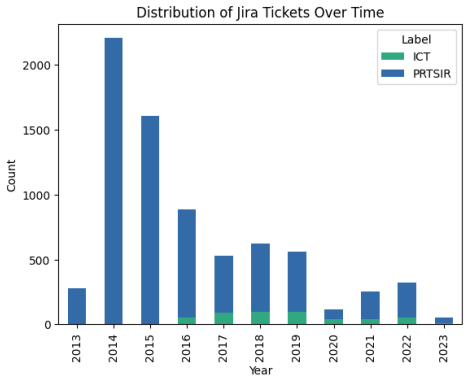

Figure 3: Distribution of Jira Hardware and Software tickets over time

Weather Data

Before using the weather data, several rounds of cleaning and joining were required. One glaring problem with the weather data, apart from intermittent reporting, was the incorrectly calibrated data, e.g., humidity readings greater than 100 or less than 0. Also, when sensors break, they often report the same data repeatedly for several consecutive readings. To address this, a script was written to establish the number of consecutive duplicates. Humidity readings over 100 that could be attributed to calibration errors rather than defective sensors were assigned 100. Negative readings were re-assigned as 0 (consistent with the fact that ALMA has extremely low humidity). Then the average for each weather type (e.g. windspeed, humidity, temperature) was computed across all reporting stations for each datapoint.

The weather and downtime datasets were then joined, linking all weather data within the starting and ending points of each downtime.

Separately, all shiftlog events detailing successful science operations were also merged with the weather data and stacked onto the downtime/weather set for future analysis.

ALMA Operations Performance Analysis

In the initial meetings with ALMA, they provided us with an overview of their process for conducting science observations. This overview was transferred visually to a process map (refer to Figure 21). A visual representation of the process helped the team understand the full process and also helped us probe further into their processes.

The map helped the team conceptualize three (3) areas where data science could improve operations:

1. Classifying the cause of downtimes (technical, scheduling, weather)
2. Predict the length of repair times once a ticket was initiated
3. Correctly classifying weather downtime causes

The shiftlog data provided by ALMA provided a window into their operational efficiencies. The daily count of failed and successful observations was examined in the following chart. The left side of the chart presents the number of failed observations, which has a median of 3 failed obs/day. The right chart presents the number of successful observations, which has a median of 15 successful obs/day. A trend of increasing successful observations can be seen over time as ALMA's operations matured. The gap seen in 2020 is due to the operational shutdown caused by the COVID-19 pandemic.

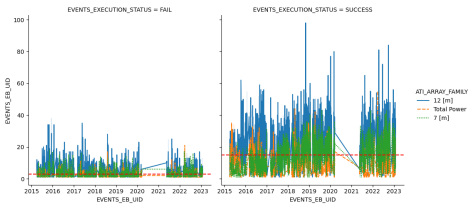

Figure 4: Failed (left) and successful (right) science observations per day.

The shiftlog data contains a daily record of the daily time allotted for observations. This enabled us to calculate utilization as defined by the following equation:

$$Percent\ Daily\ Utilization = \frac{Successful\ Science\ Obs.}{Num.\ of\ Obs.\ Attempted}$$

The following charts presents the utilization by array family. The Total Power array has periods of under-utilization that stands out in comparison to the 7m and 12m array families.

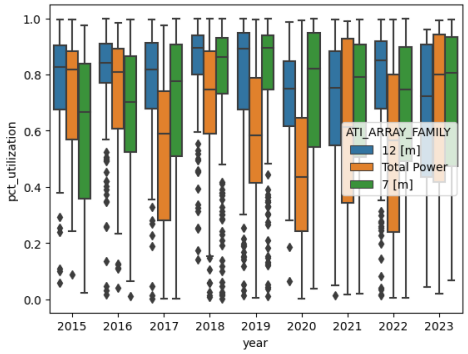

Figure 5: Percent utilization across the antenna array families.

During performance analysis, the following items were identified as warranting future investigation:

1. The ALMA site utilizes several antenna and pad configurations (refer to Figure 24) for their science observations. It was commented that some pads experience more downtime than others, therefore an investigation into antenna/pad location correlation to downtime is recommended.
2. Array name continuity (i.e. number of array names created in a day) could signal the onset of a technical downtime and should be investigated

Identifying Gaps and Challenges

ALMA possesses a wealth of valuable data. Over the years, as the observatory has grown and adapted its operations, it has continued to develop and improve its data collection and tracking systems.

However, after a thorough exploration of the data, we identified several challenges with the data and data pipeline. One issue is the lack of a consistent format for documenting problems and downtimes when they occur. As a result, each text entry in the Shiftlog comments and Jira tickets is unique. Entries often contain typos, inconsistent data types that cannot always be processed (such as screenshots and tables), inconsistent formatting, incomplete or lacking information, and potential errors in communicating or identifying the root issue.

Another challenge is that the same problem may be reported multiple times, appearing as separate problems, or labeled, classified, or called different things. Closing Jira tick-

ets can also be neglected, which leads to inaccurate recordings of how long a problem took to resolve. Some downtimes in the Shiftlog reports may be inaccurately classified; for example, a downtime may be reported as a weather issue when it is actually a technical issue.

The weather data is often unreliable due to calibration issues, other errors in the readings, inconsistent operation of some stations, and the changing use of stations over time. These factors contribute to inconsistent readings of weather patterns.

The downtimes themselves are difficult to track in duration and frequency because the (often unrelated) beginning and end of science shifts can break a downtime into two, incorrectly suggesting that the first downtime ended in recovery rather than merely because the shift ended.

In all datasets, missing data is a common issue, particularly in the weather data. Many errors, inconsistencies, and untrustworthy data points can be attributed to human error and the inability to verify the documented information.

Machine Learning Models and Predictions

Shiftlog Text Modeling

BERT was also used to classify downtime types based on the text comments in the Shiftlog dataset. The purpose of using BERT on this dataset was to perform simple predictions on unknown project comments to classify them as either weather, technical, or scheduling downtime issues. Multi-class classification was used with the BERT model, with the three classes - 'technical' (0), 'weather' (1), and 'scheduling' (2) - representing each downtime type. The dataset was split into train, test, and validation sets. The BERT model was defined with a dropout and dense layer, 'softmax' as the activation function, and 'adam' as the optimizer. The metrics analyzed were: loss, accuracy, true negatives, false positives, AUC ROC curve, and balanced recall, balanced precision, and balanced F1-score (due to the imbalanced dataset).

The model was trained on 7,759 technical, 1,734 weather, and 507 scheduling downtimes. Using multi-class classification, the model aimed to predict downtime types (weather, technical, and scheduling). The model's performance was assessed using various metrics, yielding a 96% accuracy, AUC of 99.45%, balanced recall of 89.8%, balanced precision of 94%, and a balanced F1 score of 91.4% (based on the validation set).

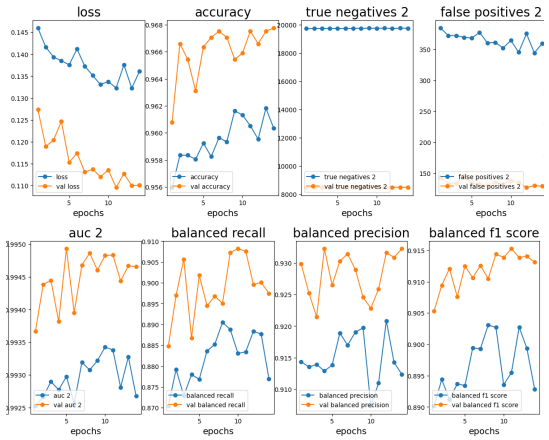

Figure 6: Evaluation metrics for classifying weather, technical, and scheduling downtime-types from Shiftlog comments with BERT. 96% accuracy, 99.45% AUC, 89.8% balanced recall, 94% balanced precision, 91.4% balanced F1 score.

No convergence was achieved with the Shiftlog data, suggesting that the model struggled to learn the distinctions between comments related to different downtimes. This issue may arise from the "messy" nature of the data, which is likely due to human errors. Implementing a standardized data entry process could significantly mitigate this problem and enhance the effectiveness of future machine learning models.

The BERT model running on the Shiftlog data took approximately 6 hours to complete 14 epochs (with early stopping) using CPU power, emphasizing the necessity for more efficient computational resources. If ALMA intends to utilize this model regularly in the future, it is essential to invest in more powerful hardware, such as GPUs, to decrease training time and facilitate more frequent and efficient use of the model in their operations. This investment would substantially improve the practicality and utility of machine learning models for ALMA's data-driven objectives.

Using BERT to classify downtime types from comments in the Shiftlog may be a useful tool in the future for ALMA. A user interface could be built to assist researchers and astronomers during their experiments to reduce human error. This interface could harness the BERT predictions, and when the user inputs a comment about a real-time issue, the model can read the comment and either suggest a downtime type or correct and verify the downtime type provided.

Jira Text Modeling

For Jira tickets, NLP with BERT was employed to classify technical software and hardware issue categories. The first model was built using cleaned and pre-processed text data from the summary and description fields in the Jira tickets, proving successful. However, the model built with cleaned and pre-processed text data from just the summary field showed even better results. Software-related problems, labeled 'ICT' tickets, were assigned a value of 0, while hardware-related problems, labeled 'PRTSIR' tickets, were assigned a value of 1. The model's performance was as-

sessed using various metrics, yielding a 94% accuracy, AUC of 97.9%, balanced recall of 93.8%, balanced precision of 94%, and a balanced F1 score of 93.9%.

The BERT model was trained with binary classification to learn the difference in the text between hardware and software issues. Then, when passed a new unknown Jira ticket, it can classify it as either a hardware or software-related issue and suggest a label. The results suggest that using just the summary text was better for classifying the tickets. This is likely due to the simplicity of the summary text data compared to the complexity of the description text. Even with thorough pre-processing, the description data can be long and 'noisy'.

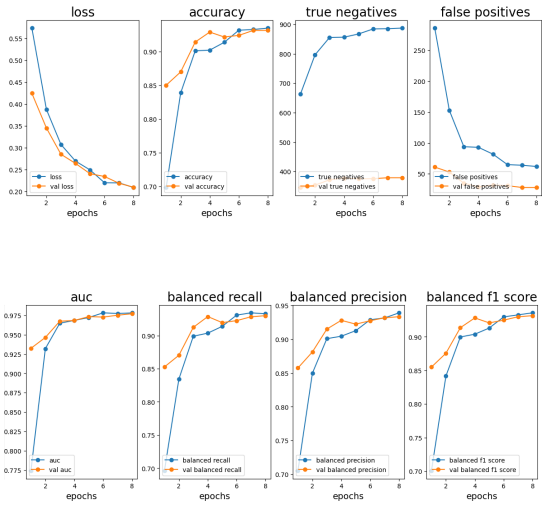

Figure 7: Evaluation metrics for classifying Hardware vs Software Jira tickets from summary text with BERT. 94% accuracy, 97.9% AUC, 93.8% balanced recall, 94% balanced precision, 93.9% balanced F1 score.

Using BERT classification has the potential to significantly reduce human error in the ticketing process. A user interface could be developed to assist ALMA personnel when creating and managing Jira tickets. This interface could incorporate the BERT model's predictions to streamline the ticket creation process and improve ticket categorization.

When a user creates a new Jira ticket, the BERT model could analyze the inputted summary text and automatically suggest whether the issue is software-related (ICT) or hardware-related (PRTSIR). This would not only save time for the user but also reduce the likelihood of misclassification due to human error. Additionally, the model could be used to identify and flag potentially duplicate tickets, helping to keep the ticketing system more organized and efficient.

In cases where the description field is necessary, or more detailed and advanced classification is desired, preprocessing techniques or text summarization tools, such as GPT and the OpenAI API, could be utilized to enhance the BERT model's performance. This would enable the model to better understand and classify complex descriptions, further improving the accuracy and efficiency of the ticketing process.

By integrating text classification with a model like BERT into the Jira ticketing system, ALMA can optimize its issue tracking and management, leading to faster resolution times and more accurate data for future analysis.

Weather Data Modeling

After merging the downtimes and successful science operations with all the weather data occurring within the timeframe of each event, we worked with our sponsor at ALMA to establish thresholds to identify flags, potentially causes of downtimes. These thresholds are given in the chart below. Note that at windspeeds greater than 15 m/s, some arrays are automatically shut down.

Classification	Conditions
Wind	windspeed > 12 m/s
Precipitation	humidity > 80, pressure < 544 mm
PWV	any pwv reading > 5 mm
Atmospheric Instability	phase RMS > 300

To classify the downtimes, the average of each weather category was computed using the first five weather datapoints per event (themselves averages of all reporting stations for that time). The idea was that the first five weather datapoints (corresponding to the first half hour of each downtime) should establish the cause of the downtime.

Each new weather datapoint (the average of the first 5 readings) was assigned a flag if it met the conditions given in the chart. If an observation was assigned multiple flags, the hierarchy of wind, precipitation, PWV and atmospheric instability established a single cause. By separately examining all weather datapoints in the downtime, and finding the first weather datapoint with no flags, we identified the starting point of each downtime recovery. Comparing to the overall length of each downtime, we could observe how long the recovery phase lasted.

Several problems arose during the analysis. First, we observed that around 48% of all downtimes had no recovery time. Either there was never a discernible cause for the downtime in the first place, hence it began in a recovery phase (by our definition of recovery, namely when the weather has no flags) or the downtime ended without any easing of the conditions leading to the original flag(s). In fact, around 30% of all weather downtimes have no weather flags in the first half hour of the downtime, hence have no cause. It should be pointed out for comparison, however, that in an analysis of events related to the 12 m array, over 76% of successful science observations had no initial weather flag. In other words, the weather data is correctly identifying problems, but not to the extent that we would expect.

While puzzling over the lack of recovery time, we noticed that a disproportionately high number of weather downtimes seemed to end around noon or midnight. After consulting with our sponsor at ALMA, we discovered that the official policy is to end the weather downtime with the shift (i.e. the scheduled block of time for the corresponding science observation). Many downtimes did not follow this rule due to technician error, adding to the confusion.

Although we did receive a new dataset showing all downtimes by shift, which will in theory allow us to link weather downtimes that span consecutive shifts, by the time of this paper, we were not able to apply this to our model. Instead, we simply restricted the analysis to downtimes that live entirely within a single shift.

Modeling Technical Downtimes

Forecasting Future Utilization Using the historic utilization data, we next developed a tool for ALMA to forecast future utilization. Long Short-term Model (LSTM) recurrent neural networks (RNN) model future outcomes for a given time step and was selected as the method for forecasting. Prior to selecting a univariate LSTM model, the daily averaged weather was analyzed for any correlations with utilization. However no statistical significant correlation was found. By constantly running this model within a pipeline, it is expected that any recurring trends will be properly captured and displayed in the model. This initial prediction model

Survival Analysis The ALMA data also provided time spans between when a Jira ticket was issued and the time that the problem was fixed or determined that the problem cannot be fixed. This data was fitted with Kaplan-Meier survival curves to assist ALMA with making operational plans once a hardware/software issue arose.

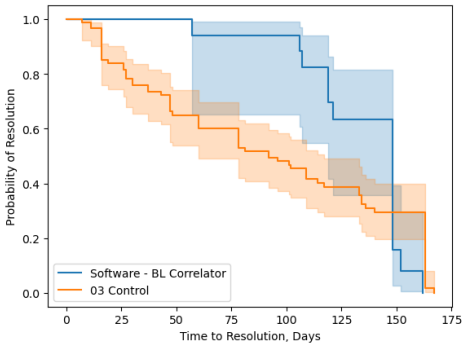

Figure 8: Survival Analysis curves for selected software Jira ticket causes.

Jira Ticket Hardware / Software Prediction Another area of operational improvement that ALMA would like to implement with Dataiku is correctly diagnosing the root cause of a hardware or software issue as early as possible. Often several Jira tickets are appended to each other as new information is collected about the problem during the investigation. By being able to immediately identify the root cause once it arises, this could direct personnel to find and fix the problem root cause, thus significantly reducing the span of downtimes.

Each Jira ticket contains a natural language summary of the problem written by the operator. Our team decided to apply NLP to the summary text and use that data to predict the root cause of the problem. A BERT model was once again used as the analysis method and applied to classifying a problem's affected hardware (435 labels), location (203 labels), and software root causes (91 labels). A multiclass confusion matrix for the classifications is provided in Figures 23 and 22. The results are reasonably accurate and useful to provide ALMA with a good starting point for expediting problem resolution. If placed within a data pipeline that continually ingests new data, it is expected that accuracy will increase significantly.

ALMA indicated that the data quality for the software ticket classification is accurate. However, the root causes provided for the hardware was not necessarily the "true" root cause. It is recommended that at the closure of a Jira ticket, ALMA would require the final root cause to be appended to the initial Jira ticket created for the technical problem.

Challenges with Modeling

One of the challenges we faced during the machine learning modeling and prediction stage was the limitation of only having access to CPU power. This constraint significantly affected the speed and efficiency of our work, as some models, particularly those utilizing BERT for natural language processing, took hours to train and evaluate. The lack of GPU resources restricted our ability to explore more complex models or perform extensive hyperparameter tuning, potentially limiting the overall performance of our models. Moving forward, it is essential to consider the computational resources available when conducting machine learning tasks, as access to GPU or more powerful hardware can substantially improve the efficiency and capabilities of the models being developed.

One of the seemingly obvious models associated with the weather data was linking the weather at the beginning of each downtime with the duration of the downtime. We used random forest, decision trees, and regression with the humidity, winds, pressure, temperature, PWV, and phase RMS readings as predictors, but were never able to attain an R^2-value higher than 0.1. After realizing how unreliable the downtime durations were (given that many simply end with the end of the shift allocated to the concurrent science observation), the reasons for the lackluster results became more clear. After eventually linking all downtimes across science shifts, we hope to redo this analysis.

Roadmap for Optimized Data Pipeline and Predictive Systems

In order to develop an optimized data pipeline and predictive systems at ALMA, several key steps should be undertaken. First, standardizing data entry is crucial. By establishing consistent formats, templates, and guidelines for entering information in Jira tickets and Shiftlog comments, data quality will be improved, and the process of cleaning and analyzing the data will be streamlined. To further enhance data quality, data validation rules and automated checks should be implemented, minimizing errors and missing data across Jira tickets, Shiftlog data, and weather data.

A centralized data storage system that integrates all data sources, including Jira tickets, weather data, and Shiftlog data, will facilitate easier access, maintenance, and analysis. Developing automated data preprocessing pipelines to clean, process, and transform raw data from different sources into a format suitable for analysis and machine learning is also

essential. Moreover, implementing data versioning to track changes in data over time will ensure transparency, reproducibility, and traceability in data analysis and machine learning processes.

In terms of predictive systems, refining the current BERT models by incorporating additional features, optimizing hyperparameters, and leveraging advanced NLP techniques, such as text summarization, is a valuable step. Investigating the use of other machine learning algorithms, such as decision trees, random forests, and support vector machines, can complement the BERT models for improved prediction accuracy. Additionally, training custom machine learning models tailored to specific ALMA operational tasks, such as predicting equipment failure, optimizing observation scheduling, and forecasting weather conditions, will contribute to more effective predictions. Continuous monitoring of model performance and updating or retraining them as needed will ensure optimal performance and accuracy.

Integrating ML-driven insights into ALMA operations requires creating user interfaces that incorporate these insights, making it easier for ALMA personnel to access and use predictive information in their daily work. Seamless integration of developed ML models and predictive systems with existing ALMA tools, software, and platforms will enhance the overall efficiency and effectiveness of operations. Establishing a feedback loop between the ML-driven insights and ALMA operations will allow for continuous learning from and adaptation to changing circumstances, thereby improving the quality of predictions and recommendations. Finally, providing training and support for ALMA personnel on how to effectively use the new data-driven tools and insights will contribute to improved decision-making and overall operational efficiency.

By following this approach, ALMA can transition to a more data-driven operational model, optimizing its data pipeline and implementing predictive systems that will reduce downtimes, improve operational efficiency, and ultimately contribute to more productive research observations.

Conclusion

In conclusion, this study focused on the analysis and optimization of ALMA's data pipeline, leveraging machine learning techniques to better understand and predict downtimes. By employing BERT models for NLP, we successfully classified Jira tickets into hardware and software issues and categorized downtime types in the Shiftlog data. These ML-driven insights can significantly enhance the understanding and management of ALMA's daily operations.

For the weather data, obtaining the distribution of recovery times for weather downtimes gives ALMA a fuller picture of how long from the resolution of weather conditions it takes to get back to operations. Unfortunately, no definitive conclusions can be reached until the original shift data is updated to reflect the true beginning and end of weather downtimes, across shifts. The results discussed here and shown in the appendix are for the limited subset of weather downtimes that lived entirely within the bounds of a single shift, which necessarily excluded multi-day weather events from the analysis.

The implications of this study for ALMA and the scientific community are substantial. With improved data management and predictive capabilities, ALMA can reduce downtimes, enhance the efficiency of its operations, and increase the overall productivity of astronomical research. The scientific community, in turn, can benefit from more reliable access to ALMA's facilities and data, enabling them to generate new insights and discoveries in the field of astronomy.

As for future work, we recommend several avenues for exploration and improvement. Further research could focus on refining and expanding the current ML models, incorporating additional data sources and features, and exploring alternative ML techniques to enhance predictive performance. ALMA should continue to invest in standardizing data entry, implementing data validation rules, and developing automated data preprocessing pipelines. This will improve the quality and reliability of its data, ultimately facilitating more accurate and actionable insights. Furthermore, the integration of ML-driven insights into ALMA's operations and the development of user-friendly tools and interfaces will ensure that these insights are effectively utilized by researchers and operational staff.

By adopting a data-driven approach and continuously iterating on the ML models and data pipeline, ALMA can significantly advance its operational efficiency, thereby contributing to the broader scientific community's pursuit of astronomical knowledge and understanding.

Appendix

Using Dataiku

Dataiku is a collaborative data science platform that streamlines the process of data wrangling, analysis, and modeling, with the intention of evaluating its suitability for ALMA's future data science and machine learning projects. Dataiku was utilized throughout all stages of our research, and we found it to be a valuable tool in managing our data and enhancing the efficiency of our work. However, we also encountered some downsides to the platform, which we discuss below.

1. Data Wrangling and Preprocessing: Dataiku facilitated the data import, cleaning, and preprocessing tasks. With its intuitive visual interface, handling Jira ticket JSON files, weather data, and Shiftlog data became a seamless process. Dataiku's built-in functions and the ability to incorporate custom Python scripts enabled us to effectively deal with missing values, inconsistencies, and data transformation.
2. Exploratory Data Analysis: Dataiku's visual exploration tools provided a comprehensive overview of the datasets, allowing us to identify patterns, trends, and anomalies in the data. The platform enabled us to generate various visualizations such as histograms, scatterplots, and heatmaps, which greatly assisted our understanding of the relationships between different variables.
3. Machine Learning Modeling: Dataiku's built-in machine learning algorithms, along with the ability to incorporate external libraries, allowed us to experiment with different models for our research. We were able to easily train,

evaluate, and compare various machine learning models, including BERT for natural language processing tasks, to identify the best-performing models for our specific objectives. However, one of our limitations was that we only had access to a CPU, which significantly slowed down our work and limited its capabilities. Some models took hours to run.

4. Collaboration and Reproducibility: Dataiku's collaborative environment enabled our team to work together efficiently, with version control and sharing of data, code, and visualizations. This ensured the reproducibility of our work and allowed us to maintain a clear record of the project's progress.

Despite the positive aspects of Dataiku, we encountered some downsides during our evaluation. These include:

1. Limited customization: Although Dataiku offers a wide range of built-in functions and algorithms, there may be instances where the platform does not provide the level of customization needed for specific tasks or projects.
2. Learning curve: Dataiku's extensive suite of tools can be overwhelming for new users, and there may be a steep learning curve for those who are not familiar with data science platforms.
3. Scalability: For large-scale projects or those involving very large datasets, Dataiku's performance might not be as efficient, and additional resources may be required to ensure optimal processing and modeling capabilities.

In conclusion, our experience with Dataiku was generally positive, and the platform's comprehensive suite of tools and its collaborative environment greatly contributed to the success of our research. However, the downsides should be considered when assessing its suitability for ALMA's future data science and machine learning projects. We recommend that ALMA carefully evaluate Dataiku in the context of their specific needs and requirements to determine if it is the best choice for their ongoing and future projects.

Other BERT Models

In our exploration of BERT models, we experimented with various approaches, including classifying antenna-related issues within the Jira ticket text and incorporating both summary and description text for classification. While using only the summary text yielded better performance, we believe that further preprocessing and potentially text summarization using a tool like GPT and the OpenAI API could improve accuracy with description text. These additional experiments provided valuable insights into the model's performance, potential applications in a real-world operational setting, and the opportunities and challenges associated with implementing machine learning models for ALMA's data analysis and prediction tasks.

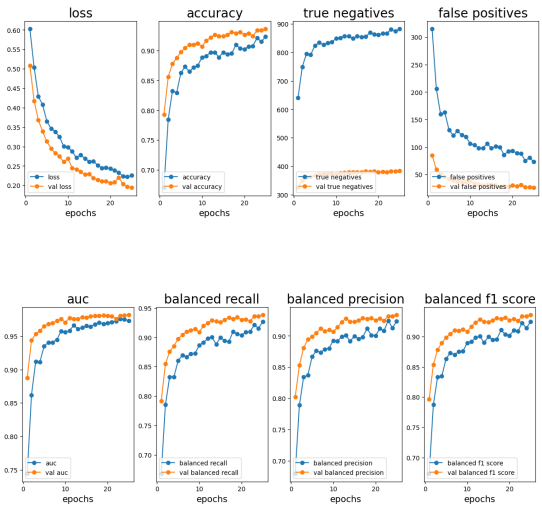

Figure 9: Evaluation metrics for classifying Hardware vs Software Jira tickets from summary and description text with BERT. 95% accuracy, 98% AUC, 94% balanced recall, 94% balanced precision, 94.5% balanced F1 score.

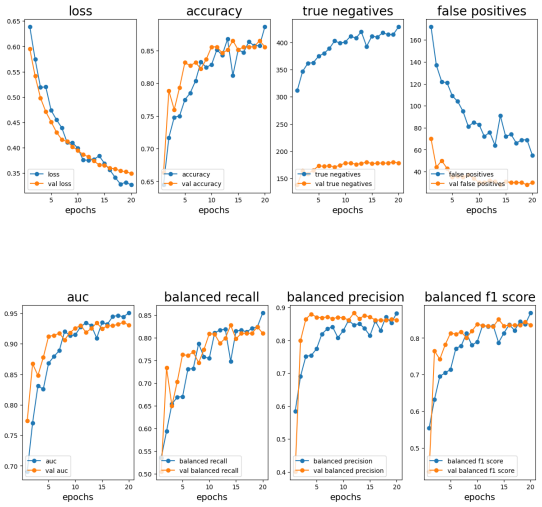

Figure 10: Evaluation metrics for classifying antenna-related Jira tickets from description and summary text with BERT. 85% accuracy, 93% AUC, 80% balanced recall, 85% balanced precision, 83% balanced F1 score.

Key Phrase Extraction with N-Grams

Our analysis of the Shiftlog comments involved creating trigrams for each downtime type, which revealed the most common three-word phrases for each category. For technical downtimes, the top phrases included 'pr1 bl aos', 'receive correlation data', 'cannot calibrate correlator', 'invoking observing mode', and 'error invoking observing'. In contrast, the most frequent phrases for weather downtimes were 'speed 20 ms', 'wind speed 20', and 'high wind aos'. Lastly, for scheduling downtimes, the prominent phrases were 'end shift gap' and 'projects available current'; upon closer examination of the data, the full phrase for the latter was 'no

projects available currently', or 'no projects currently available'. This information can provide valuable insights into the common issues faced during each downtime type, aiding in the development of targeted solutions.

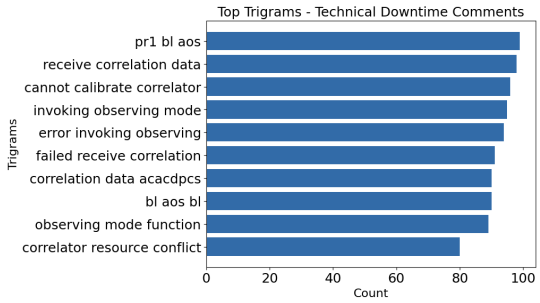

Figure 11: Top three-word phrases for Technical downtime comments from Shiftlog data

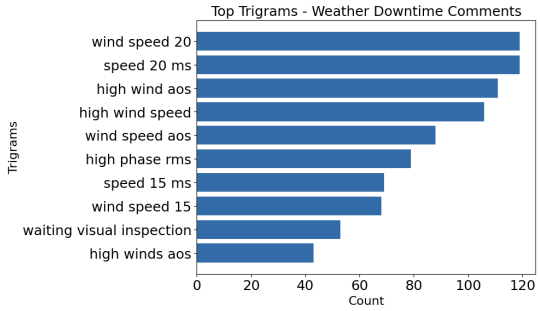

Figure 12: Top three-word phrases for Weather downtime comments from Shiftlog data

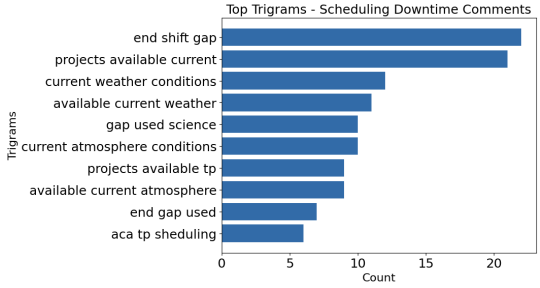

Figure 13: Top three-word phrases for Scheduling downtime comments from Shiftlog data

In a similar manner, we analyzed the Jira tickets by creating trigrams for both hardware (PRTSIR) and software (ICT) issues. The most common three-word phrase for hardware-related Jira tickets, as identified from the summary text column, was 'went to shutdown'. For software-related tickets, the top phrase was 'antenna container crash'. These findings can help shed light on the prevalent issues associated with hardware and software, allowing for better understanding and more focused efforts to address these problems in ALMA's operations.

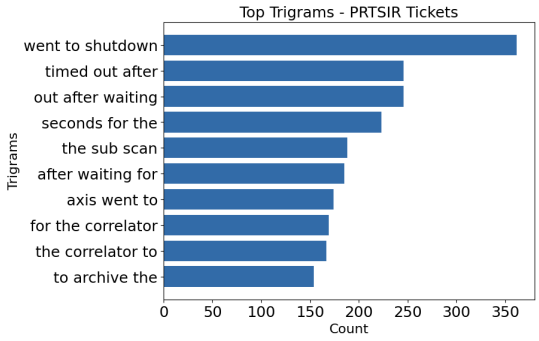

Figure 14: Top three-word phrases for PRTSIR tickets from Jira summary text data

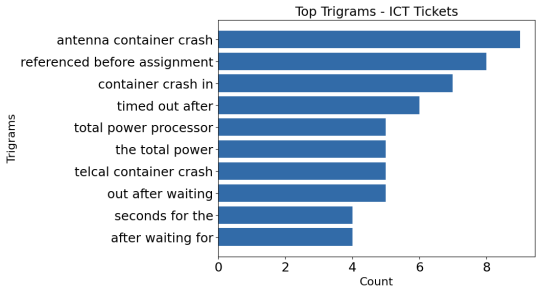

Figure 15: Top three-word phrases for ICT tickets from Jira summary text data

OpenAI API

We experimented with connecting to the OpenAI API to improve model accuracy by making the description text from Jira tickets more understandable and summarized. We connected to the davinci-002 model using 18,000 free trial tokens and passed in 40 rows of text data, which included uncleaned summary and description text from Jira tickets. Unfortunately, not all rows were summarized, resulting in only 35 usable columns, and the output summary we received from the API was only 100 tokens long.

After cleaning the data, we used the same BERT model on this OpenAI summarized text. The results were poor, with only 60% accuracy, 65% AUC, 60% balanced recall, 60% balanced precision, and 60% balanced F1 score. The poor results are likely due to the summary output from the API model being too short, as the description sections of the Jira tickets are long and complex, requiring a much longer summary to capture their true meaning.

We recommend that ALMA pursue this avenue further as a potential solution for using the description text in the Jira tickets for analysis. However, this will be a costly exercise, likely costing a minimum of $1,000 to read and summarize all of the historic Jira tickets. The experimental pipeline for connecting the data to the API and receiving usable results is provided to ALMA in the Dataiku project 'Haley Capstone (Final)', and can be used in the future to scale up the project.

Weather Charts

The first two of the following charts indicate the distribution of weather downtimes by cause and the downtimes vs.

recovery times for those weather downtimes with recovery times. The weather downtimes depicted here are restricted to those that occur unproblematically inside of one shift.

The final weather chart indicates weather downtimes in the top row and successful science observations in the second row, arranged by the presence (green) or absence (blue) of a weather flag in the first weather datapoint of the downtime. This analysis was restricted to events affecting the 12 m array. Notice the significantly higher presence of weather flags in the weather downtimes.

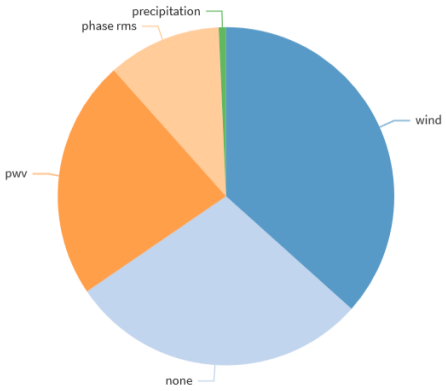

Figure 16: Downtimes by Cause

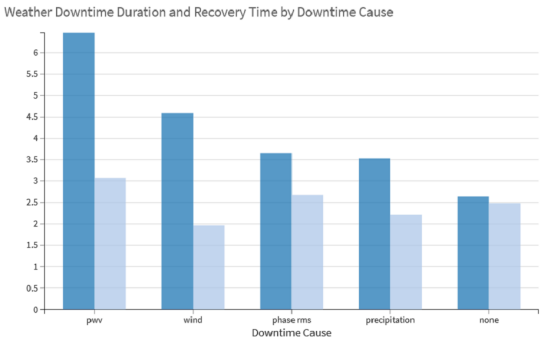

Figure 17: Average Downtime Length per Cause, with Average Recovery Length

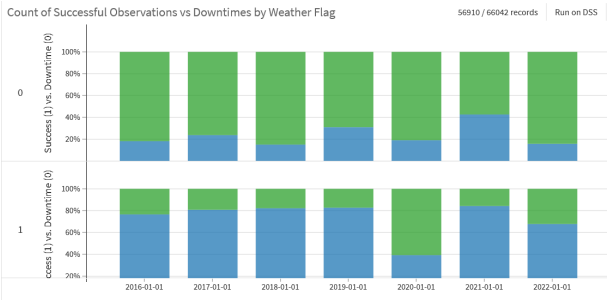

Figure 18: Downtimes vs. Successful Science Operations with Weather Flag Proportion

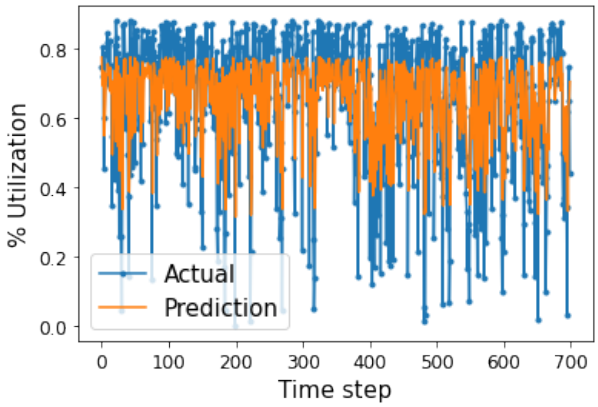

Figure 19: LSTM model percent utilization prediction results vs. actuals

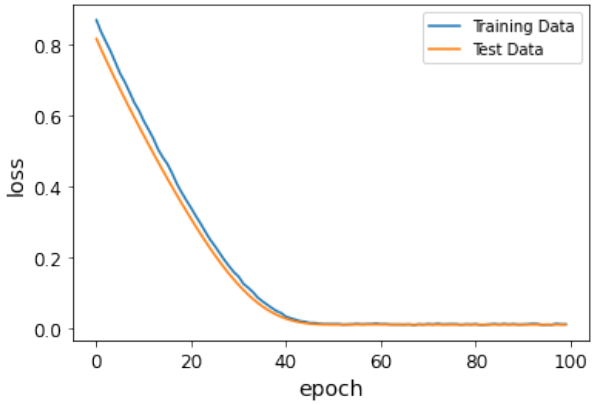

Figure 20: LSTM model performance

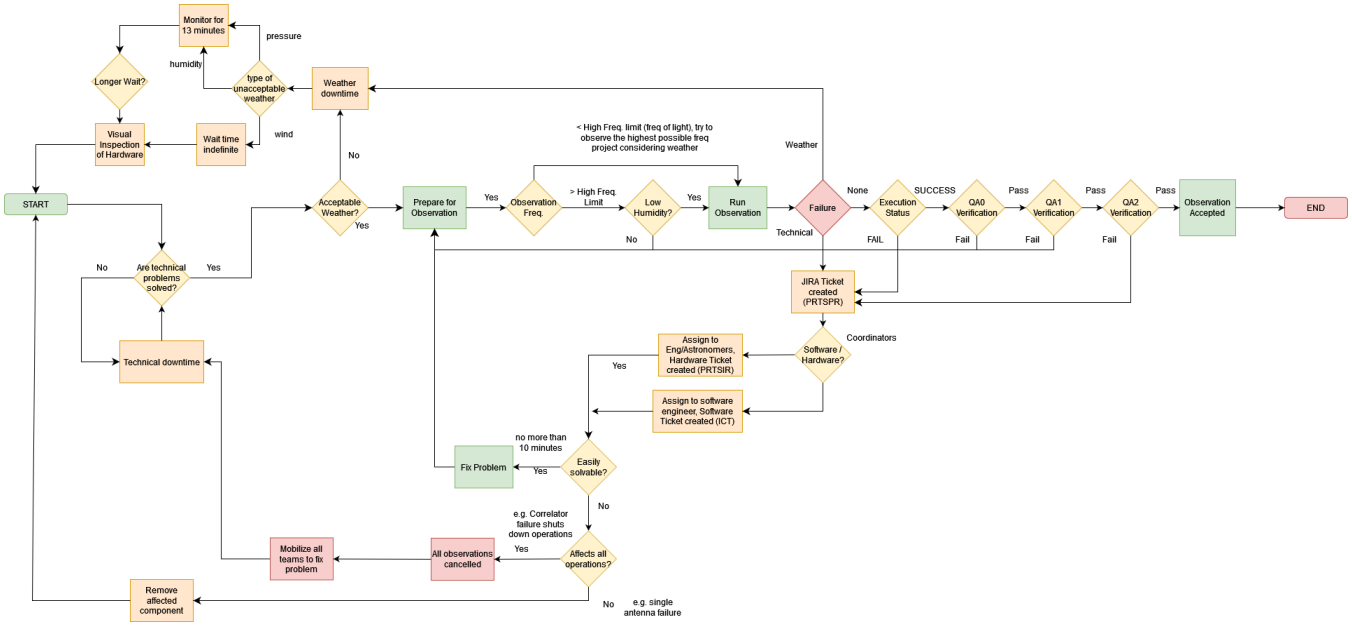

Figure 21: Process map of ALMA science operations.

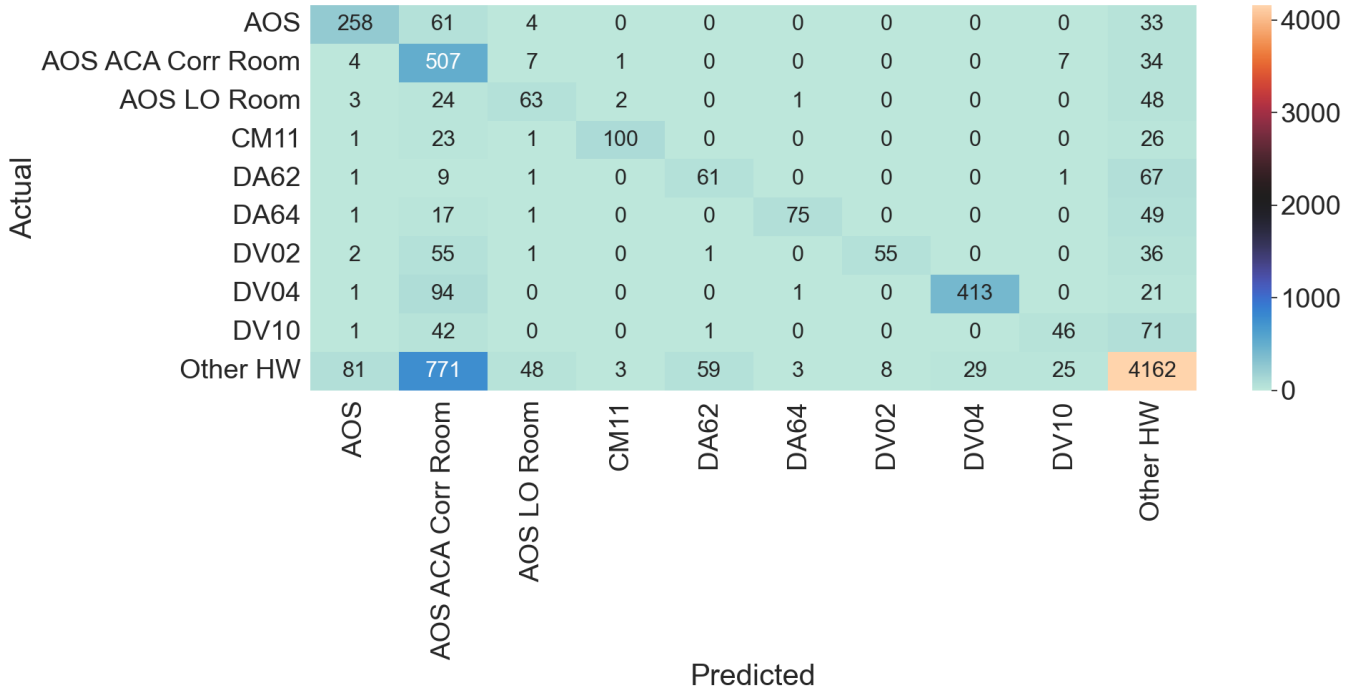

Figure 22: Multiclass Confusion Matrix for classifying Hardware Jira Ticket root causes.

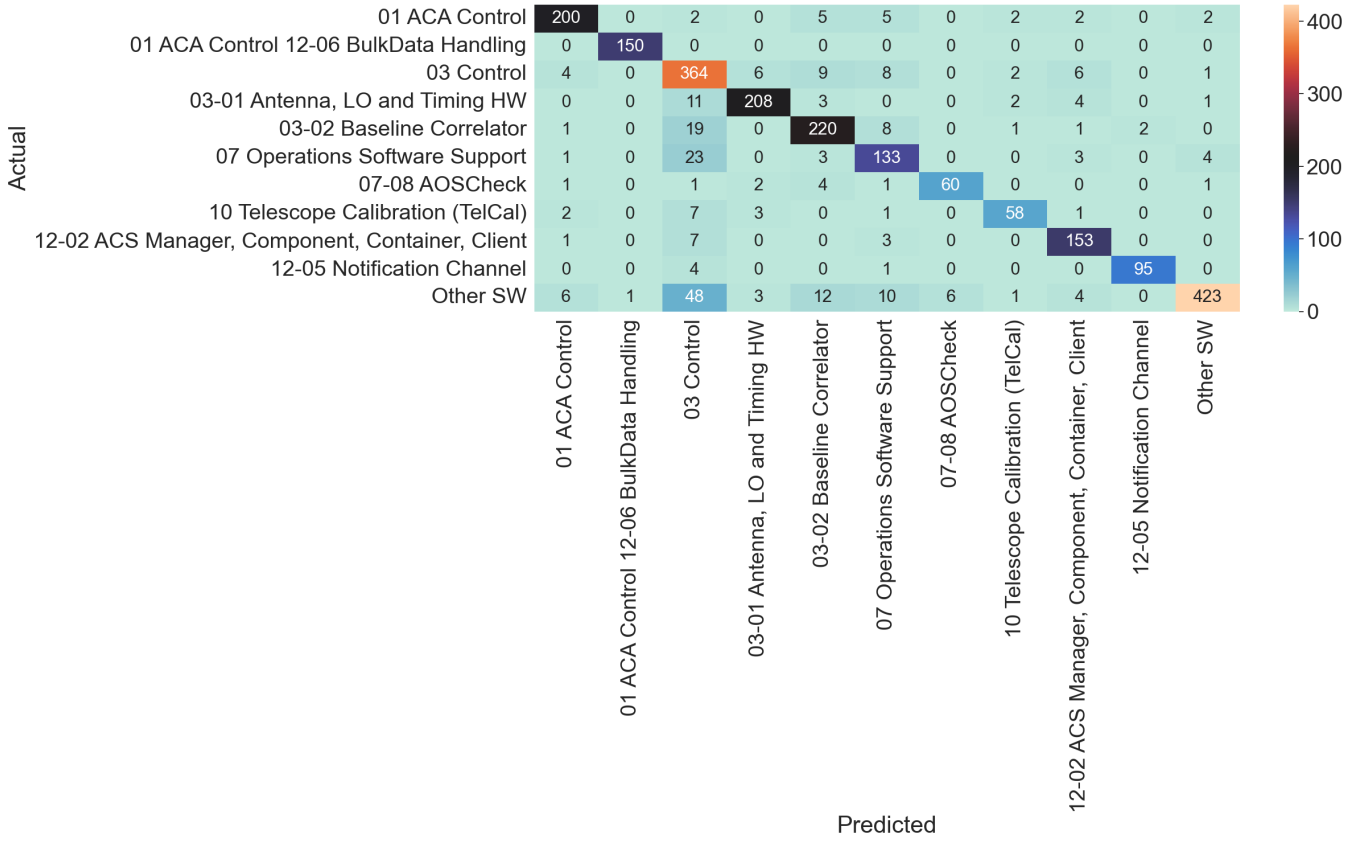

Figure 23: Multiclass Confusion Matrix for classifying software Jira Ticket root causes.

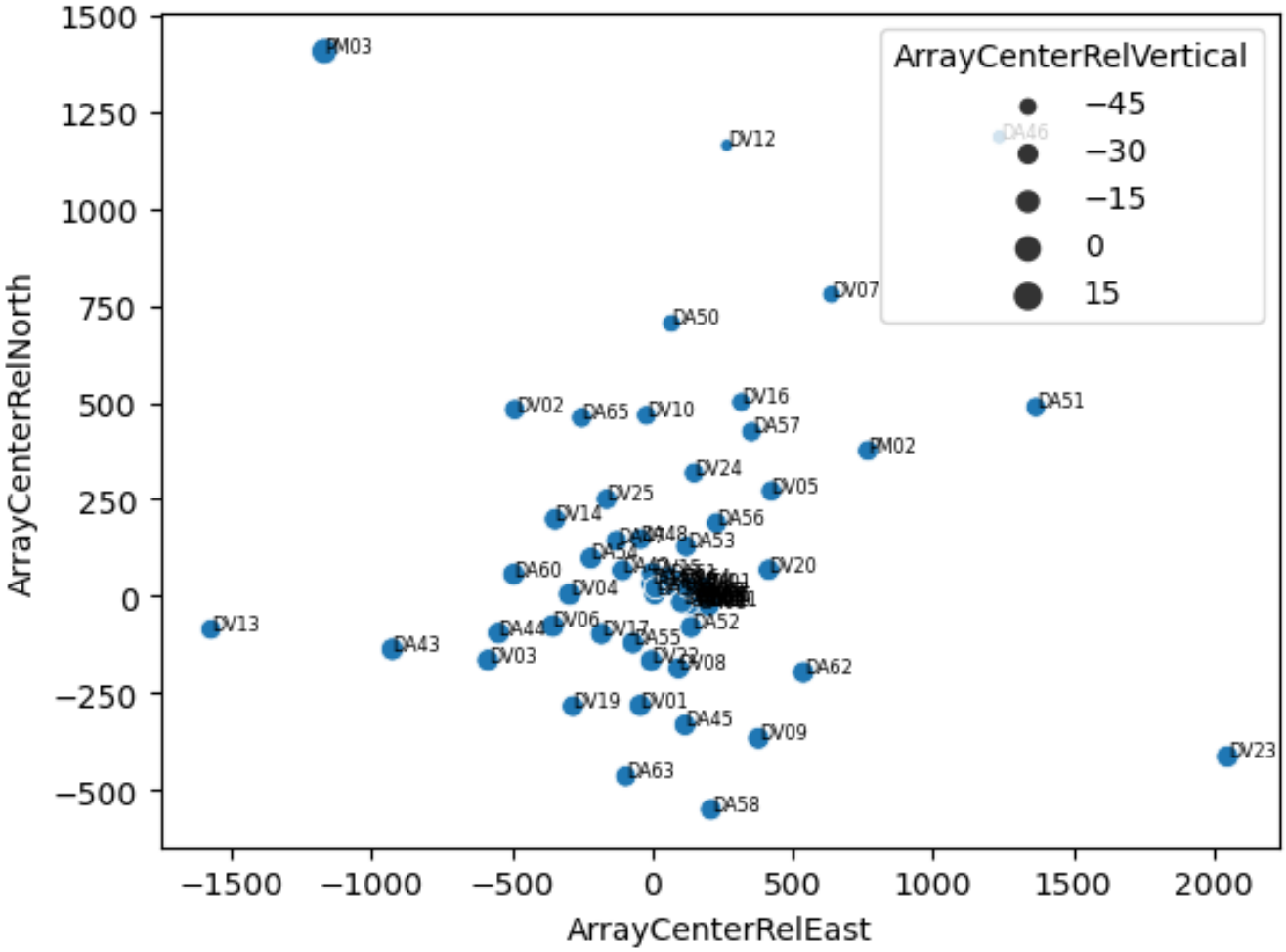

Figure 24: Example antenna configuration