

Utah State University

DigitalCommons@USU

---

Physics Capstone Projects

Physics Student Research

---

5-2024

## Analyzing Atmospheric Gravity Waves Over Antarctica and Visualizing Machine Learning Data

Anastasia N. Brown  
*Utah State University*

Follow this and additional works at: [https://digitalcommons.usu.edu/phys\\_capstoneproject](https://digitalcommons.usu.edu/phys_capstoneproject)



Part of the [Physics Commons](#)

---

### Recommended Citation

Brown, Anastasia N., "Analyzing Atmospheric Gravity Waves Over Antarctica and Visualizing Machine Learning Data" (2024). *Physics Capstone Projects*. Paper 116.

[https://digitalcommons.usu.edu/phys\\_capstoneproject/116](https://digitalcommons.usu.edu/phys_capstoneproject/116)

This Article is brought to you for free and open access by the Physics Student Research at DigitalCommons@USU. It has been accepted for inclusion in Physics Capstone Projects by an authorized administrator of DigitalCommons@USU. For more information, please contact [digitalcommons@usu.edu](mailto:digitalcommons@usu.edu).



# Analyzing Atmospheric Gravity Waves Over Antarctica and Visualizing Machine Learning Data

Anastasia N. Brown

## Abstract

In an effort to streamline the identification of “clean” windows of airglow images in all sky imager data for the ANGWIN experiment, we have developed a Light Gradient Boosted Machine (LightGBM) learning algorithm that sorts “clean” (marked as 0) wave images from “obscured” (marked as 1) images. These “clean” windows are then processed and undergo FFT-spectrum analysis. We have already successfully created LightGBM models that accurately sort through images taken at the Davis, McMurdo, and Halley research stations in Antarctica. Imager data from the Davis and McMurdo station has been fully processed from the years 2012 to 2022 with clean windows identified by using their respective LightGBM Models. The LightGBM model for the Halley station was recently verified and already several years’ worth of data have been processed. To gauge the effectiveness of the three models, phase velocity spectrums from a season’s worth of data from each station were compared against each other as well as previous findings from each station.

## Introduction

The ANTartic Gravity Wave Instrument Network (ANGWIN) is an international collaboration dedicated to investigate the upper atmosphere dynamics over a continent-size region, using a network of all-sky imagers. Before the operation reached full capacity in 2012, it was feasible to manually sort and clean the imager data “by hand”, but that is no longer possible due to the quantity of data (Table 1).

	2000	2001	2002	2003	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	2019	2020	2021	2022	2023
South Pole																								
McMurdo																								
Rothera																								
Halley																								
Davis																								
Palmer																								



 All-sky imager       AMTM + All-sky imager

Table 1: Timeline leading up to and during the ANGWIN experiment to show when the data was collected at each station.

For context, each station in use is capable of producing between 1 and 2 million images per winter season (usually lasting from March to September). Days with identified “clean” windows are flat-fielded, calibrated, rotated, and stretched to remove the fish-eye effect caused by the lens, in preparation for FFT-spectral analysis. We use two requirements to define “clear” windows, which are largely determined by the software used to perform FFT-spectral analysis: “clear” windows are defined as windows of 2 hours (usually equating to 720 frames) or longer containing at least one “clear” frame every 3 minutes (Fig. 1). For FFT-spectral analysis, we then average 6 frames together to represent 1 minute of data and perform

analysis using an algorithm based off of Matsuda et al., 2014 (Zia, 2022). Before the 2020 pandemic, Utah State University’s Atmospheric Imaging Lab had a large workforce dedicated to processing and identifying “clean” imager windows to use in FFT-spectral analysis. Unfortunately, as a result of the lock down, the workforce dedicated to processing “Clean” windows shrank.

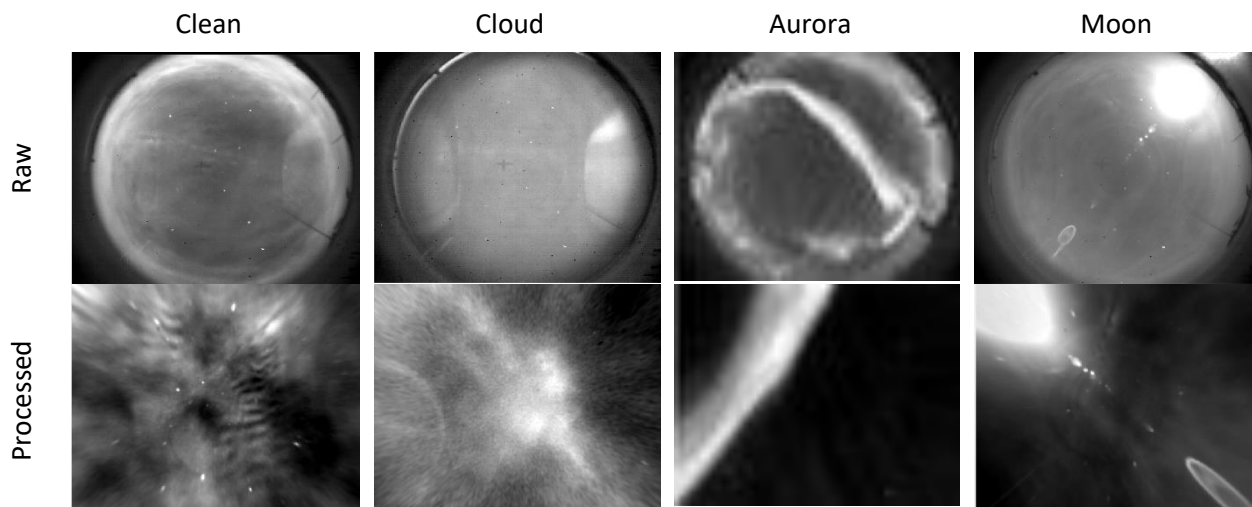


Figure 1: “Clean” windows are defined as free of excessive cloud, aurora, moon, or twilight contamination.

To streamline the sorting process, the lab now uses a machine learning algorithm to sort “clean” (marked as 0) images from “obscured” (marked as 1) images (Zia, 2022). The machine learning algorithm was inspired by one used to sort THEMIS (Time History of Events and Macroscale Interactions during Substorms) aurora all-sky imager data (Clausen et al., 2018). Generated using Convolutional Neural Network, Clausen et al.’s algorithm used a training set of 5,824 labeled images and reported 96% accuracy detecting images that contained aurora using binary classification. They initially used a 6-label ridge classifier system — labels being cloudy, moon, clear/no aurora, arc, discrete, and diffuse — that reported an accuracy of 82% when considering its ability to identify phenomena in each label.

Unfortunately, it was not possible to directly replicate Clausen et al.’s algorithm due to a large update to the Machine Learning platform they used, TensorFlow, that took place in 2019. Instead of a Neural Network, the lab’s machine learning algorithm uses a Light Gradient Boosted Machine (LightGBM) to generate its machine learning models (Ke et al., 2017). Despite generating our machine learning algorithm using a different method, we were able to mimic closely Clausen et al.’s results using their training set (Zia, 2022). Using the code Dr. Kenneth Zia wrote, and a training set of 6,526 images for the Davis station, we were able to achieve 98.5% accuracy detecting images that contained clear sky characteristics (Zia, 2022). The main drawback to LightGBM is that a model generated for one station does not transfer well to others. We found it necessary to develop a unique model for each station that we have data left to process, Davis, Halley, McMurdo, and Rothera. Dr. Zia completed the training set and model for McMurdo. The Davis station was completed back in 2022, which left the Halley and Rothera stations, both of which I have been in charge of developing. I am in the late stages of testing Rothera model, and this report will focus on the development and completion of Halley model.

## Background

In model testing, I have found that the usefulness of the machine's reported frame identification accuracy is often misleading, and not overly beneficial for our lab. For reference, we considered a study similar to Clausen et al.'s that used a larger training set (240,000 images), a Neural Network, and a trinary classification system (cloudy, clear, and dynamic) (Sedlak et al., 2023). Sedlak et al. reported a mean average precision of 82% with individual average precision values being 90% for cloudy, 85% for calm, and 63% for dynamic. A major similarity between the Clausen and Sedlak's machine learning models is that they exhibit 82% accuracy when judging their model's ability to identify all features, and both gained greater accuracy when choosing just one feature to report. In our lab, the ability to correctly identify multiple features is more desirable than correctly identifying waves and chunks of clear sky in a picture. The perceived accuracy of our models dropped when considering the machine's ability to accurately identify windows that fit the criteria for our FFT spectral analysis (Table 2).

Station	"Clean" Nights	Percent Validated	Frame Accuracy	Training Set Size	Labeled Steps	Image
Davis	169	89%	98.5%	6526	5	
McMurdo	271	79%	99.8%	6757	5 and 10	

Table 2: 2013-2017 Accuracy of ML. The machine reported the frame accuracy and found the windows. Students reported the validated accuracy. Training set size and labeled image steps are in units of frames.

Recalling the sheer number of frames required to analyze one window, it is easy to see that the reported frame accuracy is only partially reflective of the model's effectiveness. As such, the goal for the Halley model was to achieve verified window identification rates and accuracy within a range comparable to the Davis and McMurdo stations.

The second part of my work was to compare the results of the FFT analysis to the results obtained by Matsuda et al., 2017. They found that over Antarctica for the period April-May 2013, the directionality of atmospheric gravity waves (AGW) vary greatly in space and time. With that said, they found that AGW over Davis often transport 5-6 times more energy than those reported over McMurdo. They suggested that the AGW energy observed at stations near Davis latitude could be significantly higher than AGW observed near Halley and McMurdo latitudes. Compared to Halley and McMurdo, Davis has most uniform directionality. For McMurdo and Halley, averaged phase velocity showed westward propagation preference. The fewest eastward waves were reported over McMurdo and fewest northeastward AGW were reported at Halley. Matsuda et al. speculated that these differences were possibly caused by critical level filtering by background winds. Imagers at all three stations imaged OH (0.9-1.7 $\mu$ m) airglow at a sampling interval of 10s, exposure time of 3s, and a detector size of 320 $\times$ 256 pixels (Matsuda et al., 2017).

McMurdo		Halley		Davis	
78°S 167°E	81°S	76°S 27°W	67°S	69°S 78°E	77°S

Table 3: Station's latitude, longitude and magnetic latitude (Matsuda et al 2017).

## Procedure

During the development of the Davis and McMurdo models, our goal was to track the contents of our training sets by replicating the documentation styles of past students. Students who were not involved with machine learning. More precisely, the original model development method was as follows:

1. Label images and manually add the index, label, and classification of each frame to the training set csv file, and write a brief description detailing the window.
2. Feed the training set into the Python machine learning algorithm, MachineLearning.py.
3. Use the generated model to clean a year's worth of data using the ML based cleaning software, ASICleaning.py.
4. Use a window flagging software, MLShellrunner.pro, to identify "clean" window picked out by the model. The same program is used to perform FFT-spectral analysis on the windows found after they have been processed.
5. Judge the results of the first model and feed image types it misidentified back into the training set, and repeat until the model correctly identifies a majority of "clean" windows correctly.

In-depth documentation used in early development was lacking. Because of this, the results from the Davis and McMurdo models were not easily replicable. Early attempts at generating the Halley model, following a documentation style similar to the Davis and McMurdo models, only yielded a 62.3% validated "clean" window accuracy. As such, a major goal for developing the Halley model was to simultaneously develop documentation methods that improved our ability to summarize the contents of the training set and output of the models. The production of the Halley Model can be broken down by improvements to the ASI Imager data documentation processes, planned improvements to the computational tools associated with the Machine Learning algorithm, and unplanned improvements to the computational tools associated with the Machine Learning algorithm.

For the training set, one of the planned improvements was to rework the documentation of labeled images used in this set. The issue with tracking labels in training set was that it was difficult to manage a .csv file with 5000+ entries without additional documentation. Even with using a "reference sheet" to manually track changes to the training set, it was often difficult to accurately confirm that the changes made to the training set accurately reflected changes made to the "reference sheet". The solution that worked best was to computationally generate a formatted "reference sheet", and to use it to computationally generate the training sets. Changes made to the training set are easy to track, it takes less time to generate new models, and the formatted structure of the "reference sheet" makes it easy to graphically represent the contents of the training set. However, relying on code to generate the training set allows for less flexibility in labeling mixed data. For example, instead of labeling mixed data as "Moon and Cloudy" the program I wrote is only equipped to handle one label. It is not impossible to change the code to allow for greater flexibility, but such changes would take time, and the formatted "reference sheet" already has a "notes" section which allows for additional information.

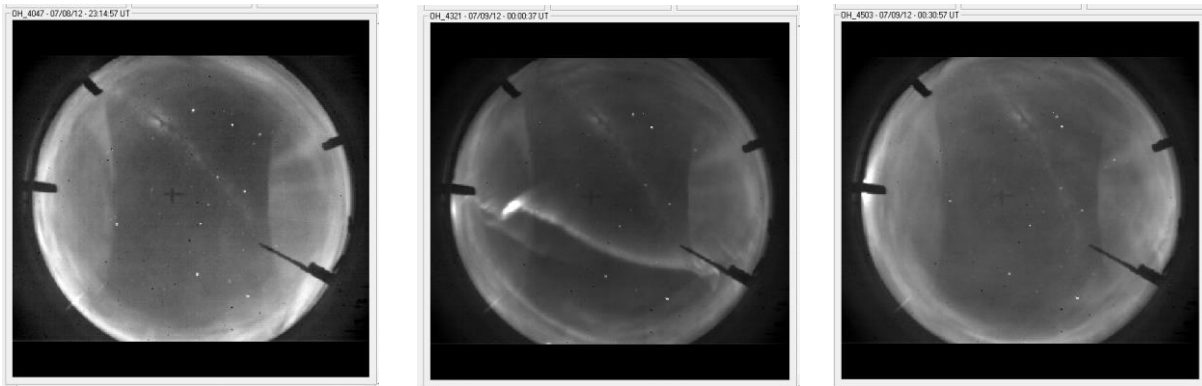


Figure 2: Three frames taken within a span of roughly 1.25 hours.

Representing the images and windows in training set using text also presented itself as a limiting factor when it came to model development. As familiarity with ASI images increased, the definition of “clear” and “obscured” images evolved. Each time the quality of a window was manually verified, it was necessary to open the images on an image viewing program (named “Norway.exe”) which is time intensive to use. The first solution was to take screenshots of frames within the window. Individual images were quick and easy to acquire and described obscuring factors more efficiently than words could (Fig. 1). However, each image only describes a 10s blip within the training set. This method does not capture the quality of the entire window. Still frames effectively document if there are obscuring features in a frame, but they do not best document the variance of a window (Fig. 2). To combat the still frame’s inability to correctly describe long periods of observation, it was attempted to use Excel files to create a weather log that describes weather features in 30-minute periods (Fig. 3).

All Days	Aurora	Cloud	Stars	Waves	Moon	Opaque Clouds
9:30						
17:30 Jul23-24						
18:00						
18:30						
19:00						

Figure 3: An example of a weather log made for tracking weather conditions.

Weather logs allowed for days, months, and potentially seasons to be documented in detail. Additionally, it allows for cross comparison between models and by counting the number of 30-minute periods exhibiting a given weather feature. One thing to note is that weather logs describe windows with less detail and less flexibility than using descriptive text. However, a detailed weather log paired with still frames allows for both the general weather conditions and time variability of a given window to be shown. Using these methods leads to Halley Model 4, which reported 73% accuracy predicting “clean” windows and 90% accuracy predicting “clean” nights in testing. However, when Halley Model 4 was used to clean an entire season, it only flagged 9 nights as being clean for the Halley station in 2013. Halley Model 4’s ability to correctly point to a “clean” window is reasonable, but most of the actual “clean” windows in the data set were missed.

Around the same time, the MLShellrunner.pro's window flagging algorithm was separated from the FFT-spectral analysis algorithm. The result of this endeavor was two additional window flagging algorithms, in addition to the original, and a separate FFT-spectral analysis algorithm, MLShellrunner.test.pro, that reads in already found "clean" windows. The window flagging algorithms have their various strengths. In short, the first two window identification algorithms, MLShellrunner Version 1 and Version 2, were written in IDL programming language. They enforced the "clean" window requirements for FFT spectral analysis by assuming each frame was precisely 10 seconds apart from the last and counting both the number of "clean" frames that separate "obscured" along with ensuring the total number of "clean" frames in a window added up to a minimum of two hours. The third, MLShellpy, was written in Python. Comparing the MLShellrunner Version 1, MLShellrunner version 2 and MLShellpy results provides more checks to increase the number of "clean" windows found. Transferring the window flagging algorithm to Python also allowed for the FFT-spectral analysis requirements for the clean windows to be met by examining the timestamps associated with the imager data. This allowed for more precise window identification at the cost of including windows with a lower minimum count of "clean" frames in the window. The differences of the window identification algorithms will be discussed more in depth later, but because most of the data processing was done with the window identification algorithm associated with MLShellrunner Version 1, the station comparison was done using MLShellrunner Version 1 data.

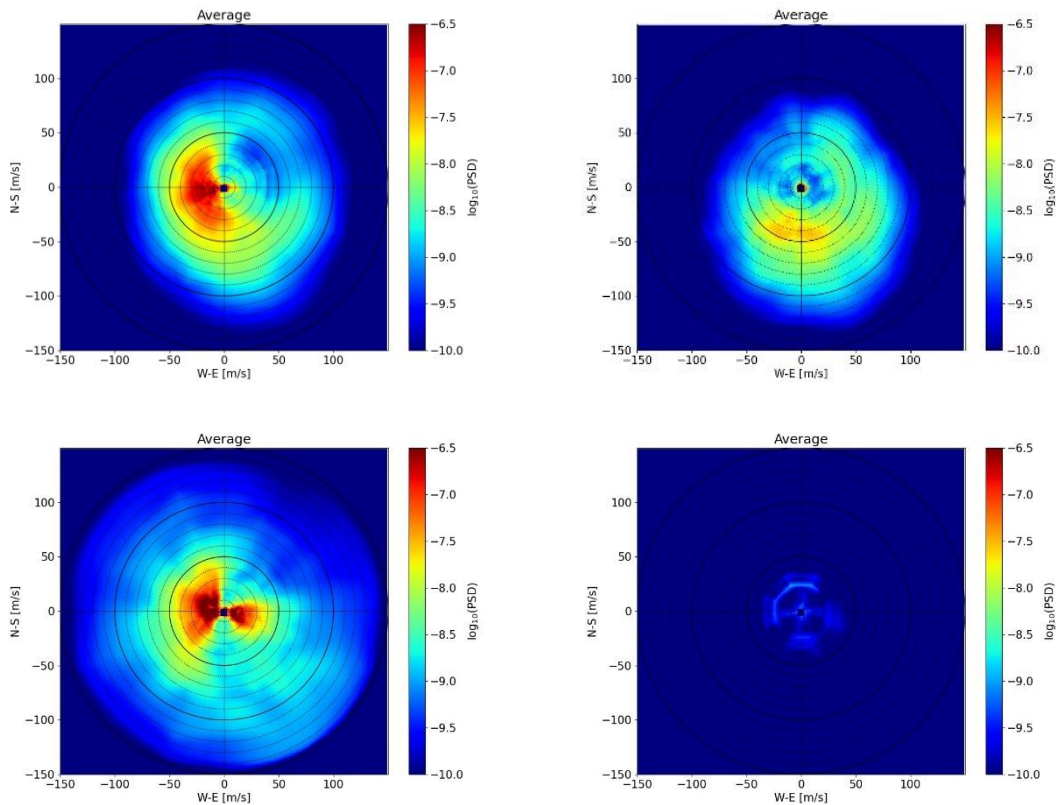


Figure 4: Phase velocity spectrums from the Halley Model 4 training set, comprised of 2012 data. Top left: Aug09-10 2500-5000, originally labeled as "clear", kept as clear. Top right: Sep09-10 1600-2600, originally labeled "obscured" but changed to "clear". Bottom left: Aug16-17 1900-2600 originally labeled as "Aurora", removed from the training set. Bottom right: Jul05-06 2250-3175 originally marked as "moon", window length extended to 1000-3175.



What is interesting about MLShellrunner.test.pro is that it allows me to test the phase velocity spectrums of the specific training set windows and not just the output produced by the model. For model development, what this accomplished was testing how similar windows classified as “clean” were to windows classified as “obscured” (Fig. 4). This helped to find mislabeled data and ambiguous windows in the training set. Mislabeled data was relabeled and kept in the training set, and the ambiguous data was re-evaluated.

Relatively few pixels distinguish moon glare with waves from clear sky with waves. The aurora flickers on and off the frame quickly. Furthermore, after reviewing the spreadsheet made to log the weather conditions of July Halley 2012 for this experiment, 23 nights were reported as having stars visible (clear sky), 18 of those nights were logged as having potential wave activity, 10 of the nights with stars visible were reported to have aurora activity, as well, 8 of the nights that had aurora activity were also reported to have exhibited potential wave activity. Upon examining the “Basic Cloud Diary 2012” that another student made in years past, it appears they recorded 116 nights with possible wave activity for the entire 2012 season at Halley (but when it was done manually, <2 hours periods of clear sky were still considered good and processed). Of the nights that included reports of auroral activity, 45 out of 48 nights (93%) also included reports of wave activity (meaning that 38% of nights with reports of wave activity also included reports of auroral activity). These numbers could be biased as the student who made the cloud diary might have not been focusing on reporting aurora activity, but the correlation appears to be impressive. Given the possible correlation between waves and aurora, it was decided to develop Halley Model 8 by not including obscuring examples of aurora in the training set and leaving it to the data cleaners to decide which of those windows should or should not be processed.

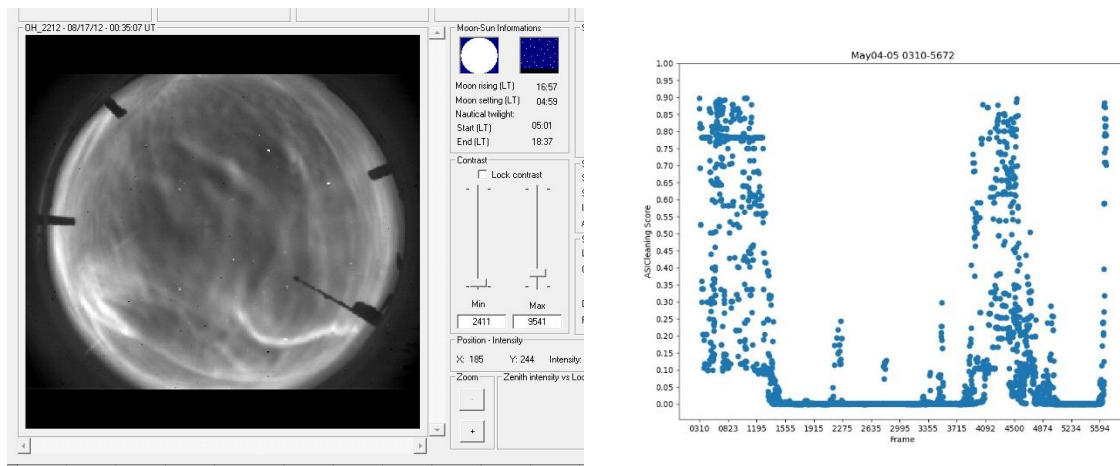


Figure 5: Improved techniques for developing ML model. (Left) Image from Halley 2012 Aug16-17 1900-2600 training set window. Although there are clear instances of geomagnetic activity, the majority of the frame features strong wave activity. Training against the aurora also inadvertently trains against clear sky phenomena. (Right) A graph of ASICleaning scores from Halley Model 8, 2013.

With the FFT algorithm now able to read in “clean” windows, the criteria for a window being validated and processed has softened. It became less necessary to have a machine learning model that precisely determine the start and end of each window, as it is now possible to manually adjust the length of the windows. By filtering out only the most extreme obscuring phenomena, the chances of missing clear sky and wave phenomena decrease. With this in mind, the subtlety of the Halley training set decreased until the model correctly identified frequently missed clean windows in March of 2013. Once that was



achieved, efforts were shifted validating Halley Model 8's ability to correctly identify and catch clean windows in the Halley 2013 dataset.

For consistency during this experiment, the length of the windows flagged by the machine learning model were not adjusted. However, using graphs of the ASICleaning scores, the model's capacity to evaluate Halley 2013 data was tracked and notes of the model's ability to correctly judge clear and obscured windows with anything noteworthy. To validate Halley Model 8 across a greater range of data, instead of making a weather log for Halley 2013 data, a comparison was done between the ASICleaning score graphs from each day to the imager data associated with the graphs. When the imager data and ASICleaning scores disagreed, a video comprised of the frames in the window was made for comparison. Then a summary file was used (the format of which and supporting code was designed by a fellow student, Connor Waite) to track the clean windows. The following was tracked: windows found by each flagging algorithm, windows found by examining the ASICleaning score graphs, and unique windows found by manually examining the imager data that were entirely missed by the other window identification methods. Fellow undergraduate students, Connor Waite, Dallin Tucker, and Max Haehnel helped to prepare the 2013 data for the Davis and McMurdo stations for comparison.

## Analysis

Below are the window length and average power of every clear sky window found for Halley 2013 (Figure 6). For both the computationally identified windows and the visually identified windows, their average power tends to be below 0.0020. Additionally, MLShellpy tends to find slightly longer windows than the IDL based window flagging algorithms. The number of windows found via each method and the percentage of those windows validated as correct can be seen in table 4.

	Unique Manually Identified windows*	All Graph Estimates	Unique Graph Estimates*
Window	22	111	29
Validated	95%	82%	48%
Night	22	84	25
Validated	95%	92%	52%
	MLShellrunner Version 1	MLShellrunner Version 2	MLShellpy
Window	31	65	76
Validated	80%	80%	79%
Night	27	57	66
Validated	85%	84%	83%

Table 4: Windows identified compared to the percent of windows correctly identified using various window identification methods.

\*Windows identified in addition to what was found by other means. The nights are not necessarily unique.

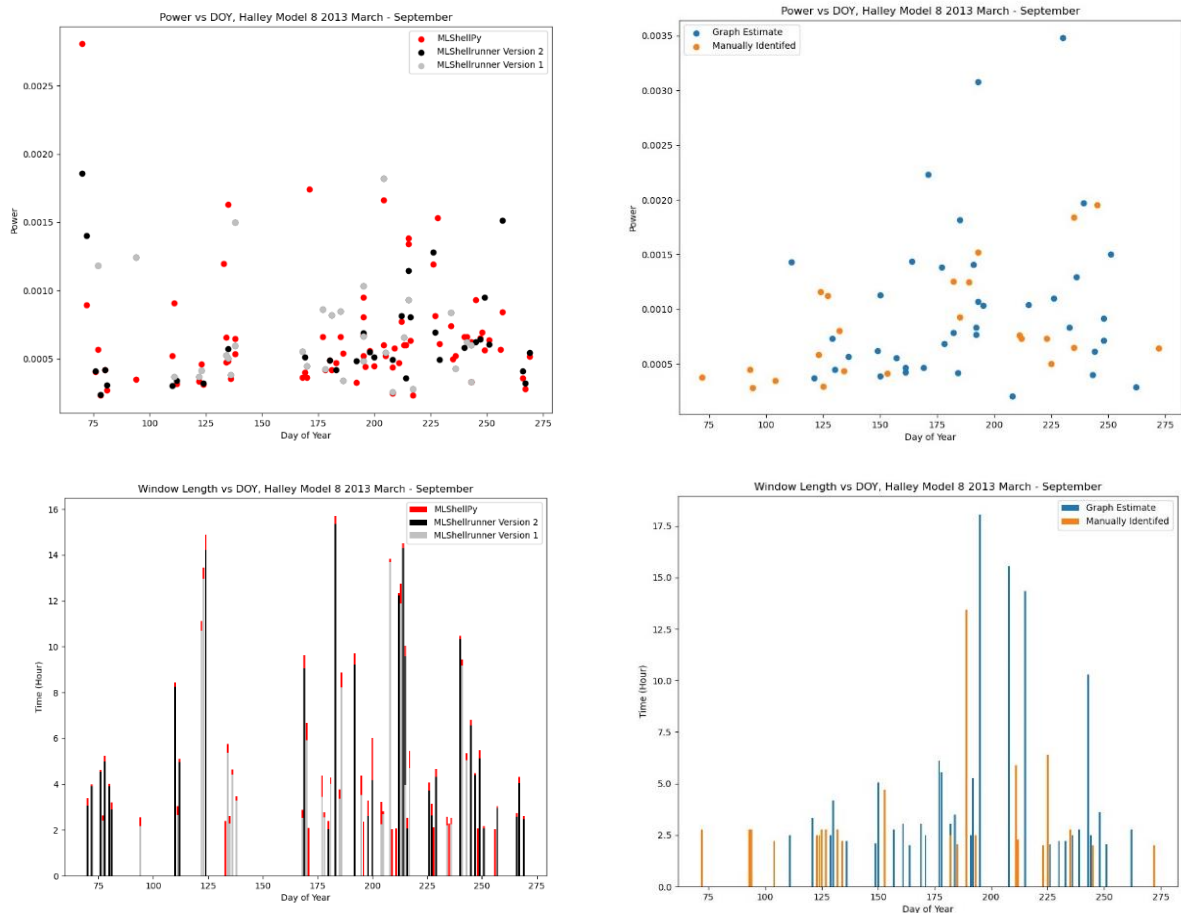
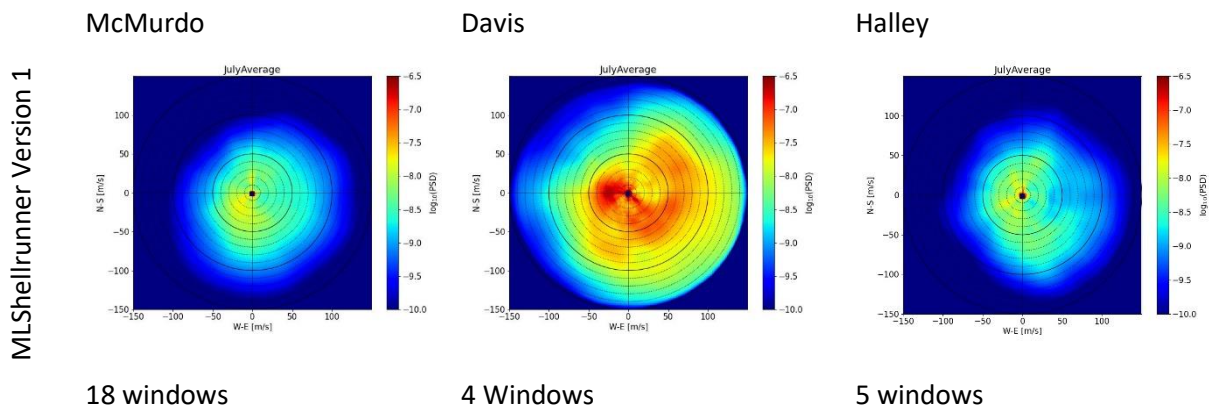


Figure 6: Left: Windows found by one of the MLShellrunner window flagging algorithms. Right: unique windows found by either examining graphs of ASICleaning scores or manually sifting through the data. Top: spectrum power, bottom: windows duration.

Even though the percent of validated “clean” windows is consistent across all three computational window flagging algorithms, the phase velocity spectrums vary noticeably, even after manually removing obvious ‘obscured’ windows the algorithm erroneously flagged (Figure 7).



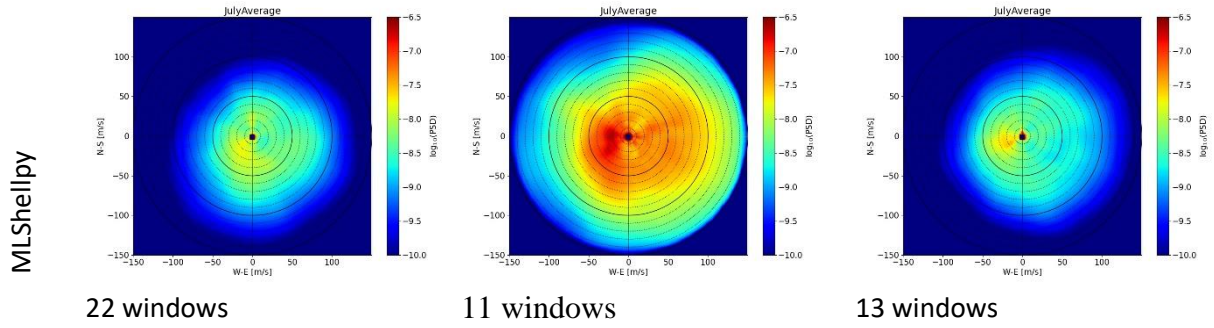


Figure 7: The number of windows found by various window flagging algorithms affect the perceived physical properties of the monthly phase velocity spectrums.

The obvious differences between the phase velocity spectrums of the MLShellrunner version 1 generated windows and the MLShellpy windows is concerning. To test the severity of the physical differences reported, the phase velocity spectrum produced by the IDL MLShellrunner code was subtracted from the phase velocity spectrum produced by the Python MLShellpy code. The results can be seen in Figure 8.

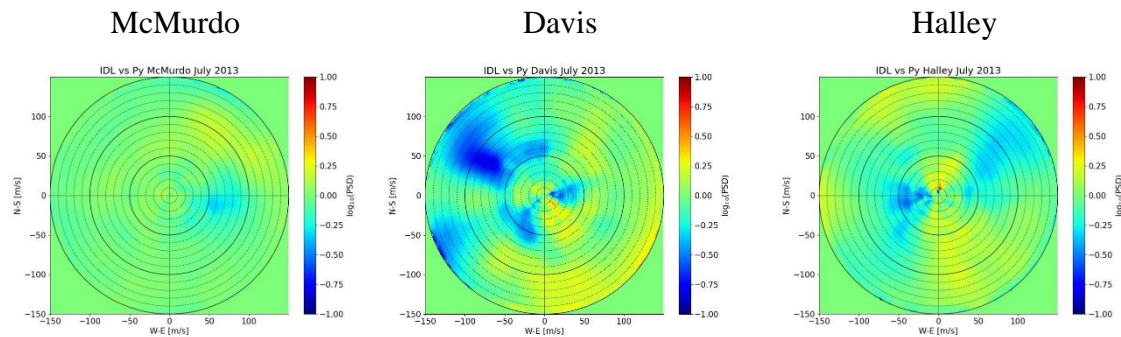
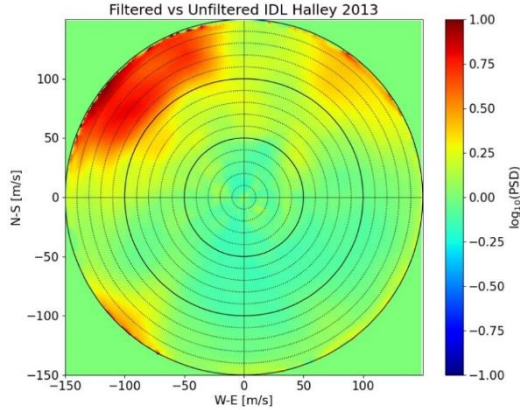


Figure 8: The differences between July 2013 average phase velocity spectrums generated using windows found from MLShellPy and MLShellrunner Version 1, for McMurdo (left), Davis (center), and Halley (right) stations.

For the McMurdo station, there is not much difference between the phase velocity spectrums generated by MLShellrunner Version 1 and MLShellPy, but for Halley and Davis the difference between the phase velocity found by the two window flagging algorithms varies by as much as an order of magnitude. To test the cause of this, the “Filtered” and “Unfiltered” windows from Halley July 2013 that were found by the two window flagging algorithms were compared to gauge which flagging algorithm most closely matched reality. This was done by generating an averaged phase velocity spectrum of each data set and subtracting them. “Filtered” refers to windows identified by the flagging algorithm and manually verified as “clear”. “Unfiltered” refers to the entire set of windows identified by a given window flagging algorithm. As can be seen in Figure 9, the data that was filtered out of the MLShellpy windows left less of an imprint than the data that was filtered out of the MLShellrunner version 1 windows. What this means is that, even though MLShellrunner version 1 has a somewhat higher percentage of “verified clean” windows it identifies, the obscuring data in the “unfiltered” set of windows found by MLShellpy had less of an impact on the phase velocity spectrums.

MLShellrunner Version 1



MLShellpy

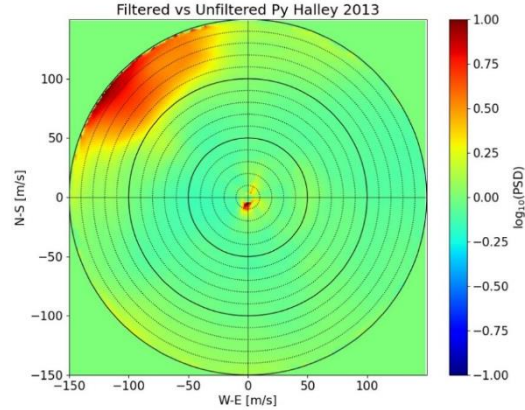
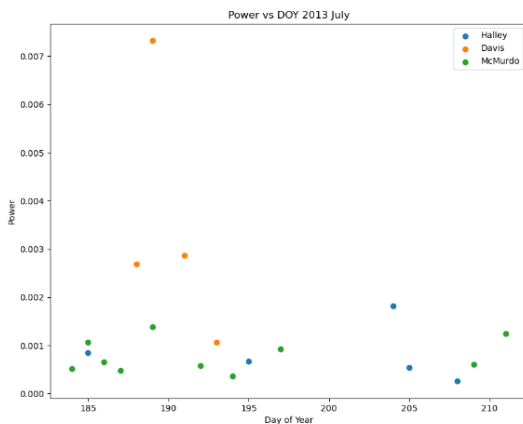


Figure 9: The difference between the phase velocity spectrums made using filtered and unfiltered windows identified by IDL's MLShellrunner and Python's MLShellpy.

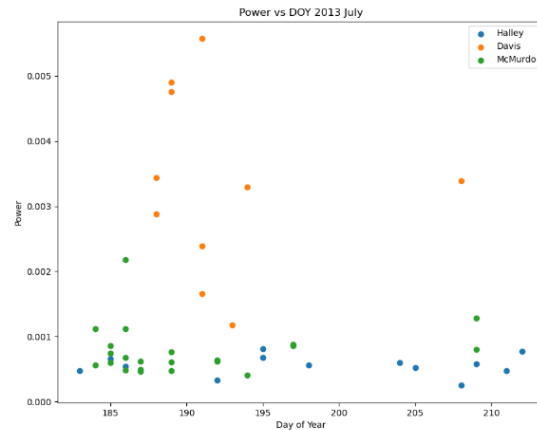
To verify that the windows were correctly filtered, videos were made for reference. It should also be noted that to save time, windows with a long period of both obscured and clean data were filtered out. Future investigation may benefit from testing how manually adjusting the length of mixed windows and adding them to the "filtered" set affects the phase velocity spectrums.

Another thing to consider is that by comparing windows found by MLShellPy and MLShellrunner Version 1, it becomes apparent that increasing the number of windows analyzed does not greatly affect the range of energies observed but can affect what is considered to be an outlier. Below is a comparison of average power and window length gathered from filtered MLShellrunner Version 1 and MLShellpy windows found for July 2013 (Figure 10). Particularly when considering the power measured from the Davis station and the length of windows gathered from the Halley station, data points that appear to be outliers in the MLShellrunner Version 1 set of windows appear to be statistically significant in the MLShellpy window set. This implies that the small sample size of windows gathered by MLShellrunner Version 1 acts to skew the perceived physical properties of the data set.

MLShellrunner Version 1



MLShellpy



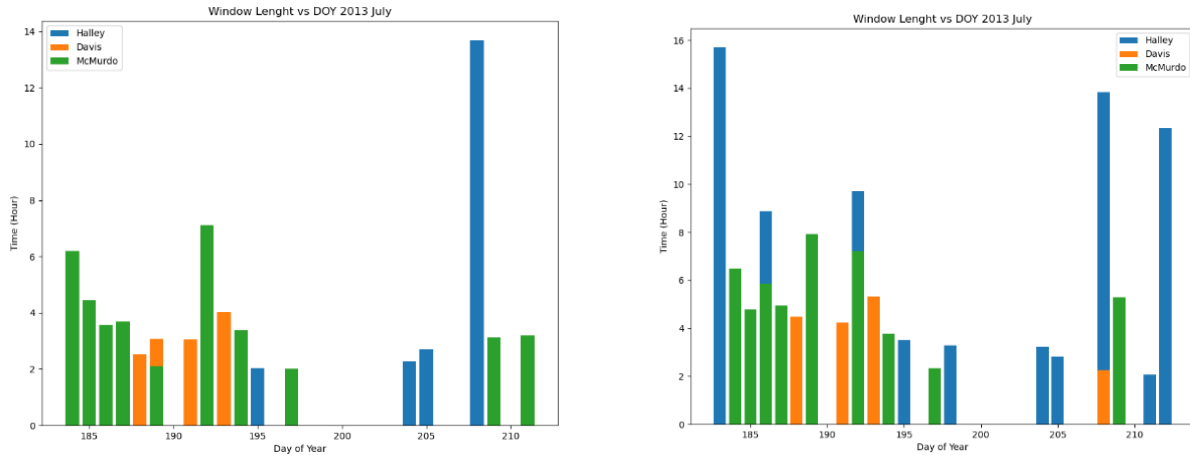


Figure 10: Windows found by MLShellrunner Version 1 appear to exhibit several outliers in terms of average power and window length found. However, the increased number of windows found by MLShellPy fills in this gap.

So long as the “clean” windows acquired from MLShellPy are adequately verified, it is fair to suggest that MLShellPy yields a closer depiction of reality compared to MLShellrunner version 1. Further investigation is necessary to see how the three station’s machine learning models compare when using MLShellPy to flag “clean” windows.

## Results

In terms of the quantity of verified windows found between the three stations, average power found from each window, and the total length of each window, a comparison can be seen in Figure 11. Using MLShellrunner Version 1 to flag windows found from each model, McMurdo had the greatest number of windows and Halley had the longest average windows. The ratio of validated and unfiltered nights found at Halley in 2013 was within the range of validated and unfiltered nights found from 2013 to 2017 at the Davis and McMurdo stations. Moreover, the ratio of validated and unfiltered nights observed using the Halley model stayed consistent across each computational window flagging method.

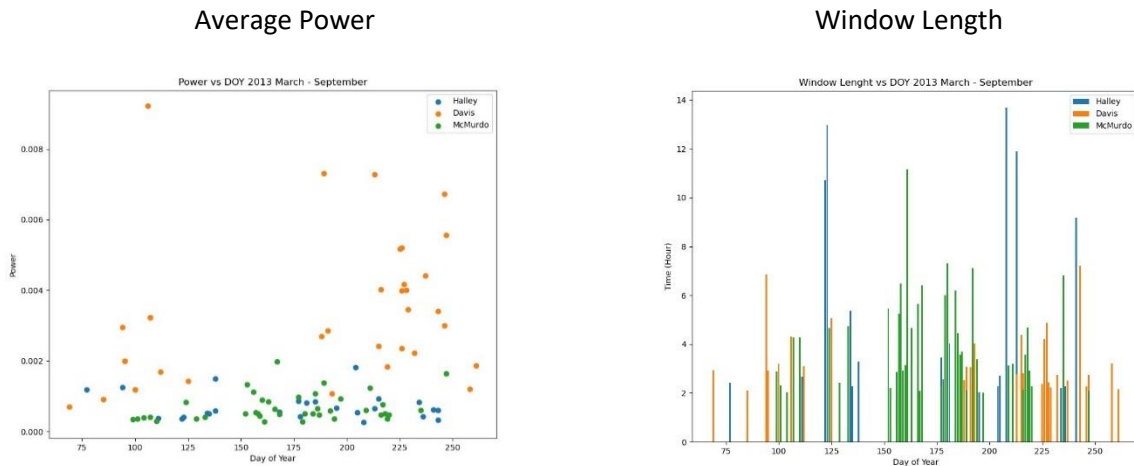
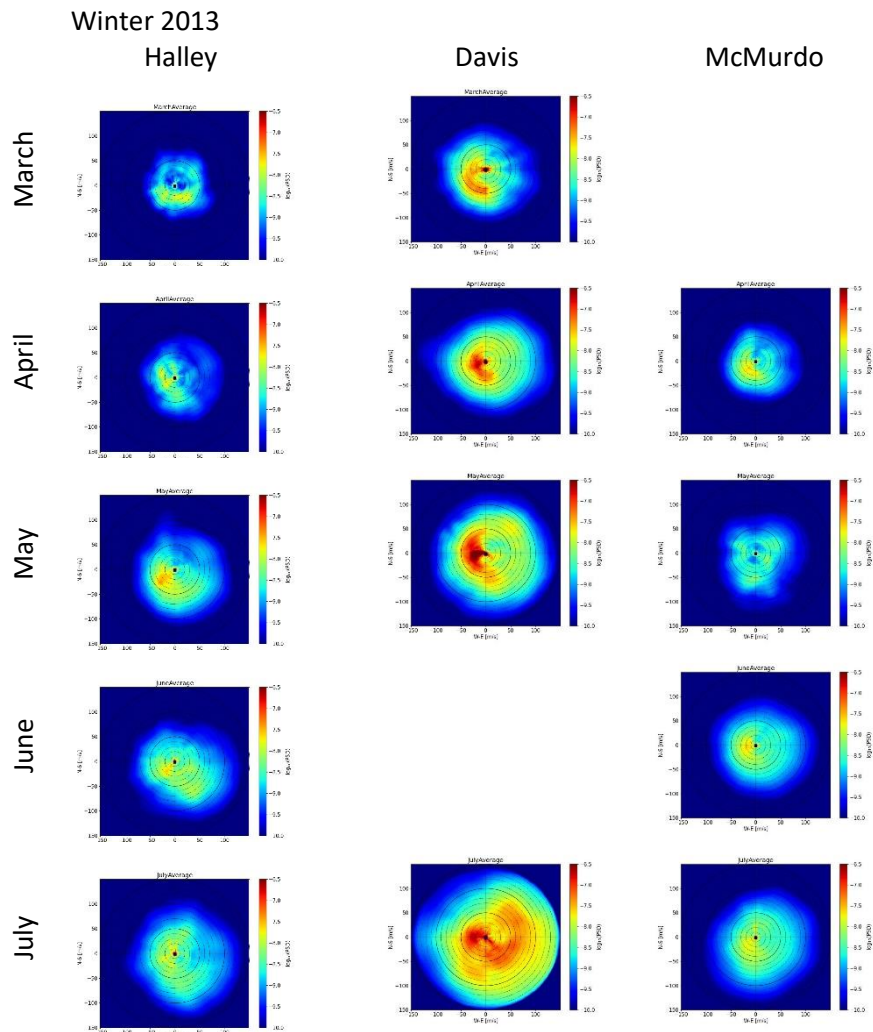


Figure 11: Average power and window length for Halley, Davis, and McMurdo stations in 2013.

Davis had windows with the greatest average power, and the average power found by Halley and McMurdo were consistent with each other. Without manually adjusting the mixed windows found by the machine learning models, the power observed at Davis was 4.6 times larger than the power observed at Halley and 4.8 times larger than the power observed at McMurdo. This is comparable to the findings of the 2017 Matsuda et al. study. There is evidence that using a different window flagging algorithm, MLShellPy, may impact the average power observed at each station, and further research is needed to gauge the effect of manually adjusting the computationally found windows has on the observed average power.

As seen in Figure 12, power increases over the course of the winter for all the stations and directionality is very similar for McMurdo and Davis (~SW) during most of the winter. It varies for Halley from W to E, which is similar to what was found by Nielsen et al., 2009, who analyzed individual wave events instead of power spectrums. Further testing revealed that there is evidence to suggest that “clean” windows flagged by MLShellPy more closely reflect phenomena that occurs in the data.





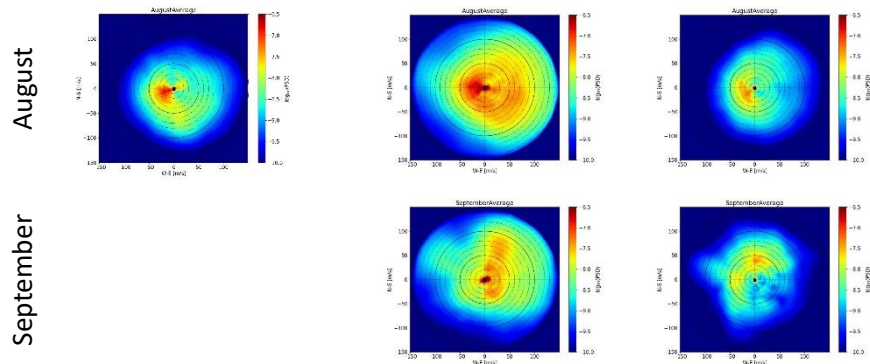


Figure 12: Phase velocity spectra from Halley, Davis, and McMurdo using windows gathered using their respective Machine learning models and MLShellrunner Version 1.

## Conclusion

A machine learning model, Halley Model 8, was developed for the Halley station airglow image dataset that sorts “clean” and “obscured” images with an accuracy comparable to how the McMurdo and Davis models performed for their respective stations. The accuracy of the clean window identification varies with the window flagging algorithm used. Comparing the three stations using MLShellrunner Version 1 resulted in the Halley model accurately identifying “clean” windows in 85% of the nights it flagged, compared to 89% of the nights the Davis model found, and 79% of the nights the McMurdo model found. On average, the power observed at Halley and McMurdo are similar, and much smaller than the power observed at Davis. All of them were comparable to what was found in the 2017 Matsuda et al. study. In the comparison of the phase velocity spectrums, the power increased throughout the winter for each station. The McMurdo and Davis stations performed similarly to the results seen in the 2017 Matsuda et al. study, but the Halley station phase velocity spectrums varied from W to E. There is evidence that windows obtained using the MLShelppy flagging algorithm yields results closer to what is apparent in the physical data. More research is required to test how the phase velocity spectrums and average power turn out when the mixed windows are manually adjusted to keep only the “clean” data as opposed to being entirely thrown out.

## Acknowledgments

This research project is supported by NSF grant 2029318. Mentors who helped with this project were Pierre-Dominique Pautet, Yucheng Zhao, and Michael J. Taylor. Efforts to implement the use of machine learning for clean window identification was started by Kenneth Zia. Students who assisted with data cleaning for this project were Connor Waite, Dallin Tucker, Max Haehnel, and Eli Kroeber.

## References

Clausen, L. B. N., and H. Nickisch, Automatic classification of auroral images from the Oslo Auroral THEMIS (OATH) data set using machine learning, *Journal of Geophysical Research: Space Physics*, **123**, 5640– 5647, doi.org/10.1029/2018JA025274, 2018.



Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, LightGBM: A Highly Efficient Gradient Boosting Decision Tree, *NeurIPS Proceedings*, NIPS'17, Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 3149–3157, 2017.

Matsuda, T. S., T. Nakamura, M. K. Ejiri, M. Tsutsumi, and K. Shiokawa, New statistical analysis of the horizontal phase velocity distribution of gravity waves observed by airglow imaging, *J. Geophys. Res. Atmos.*, **119**, 9707–9718, doi:10.1002/2014JD021543, 2014

Matsuda, T. S., T. Nakamura, M. K. Ejiri, M. Tsutsumi, Y. Tomikawa, M. J. Taylor, Y. Zhao, P.-D. Pautet, D. J. Murphy, and T. Moffat-Griffin, Characteristics of mesospheric gravity waves over Antarctica observed by Antarctic Gravity Wave Instrument Network imagers using 3-D spectral analyses, *J. Geophys. Res. Atmos.*, **122**, 8969–8981, doi:10.1002/2016JD026217, 2017.

Nielsen, K., M.J. Taylor, R.E. Hibbins, and M.J. Jarvis, Climatology of short-period mesospheric gravity waves over Halley, Antarctica (761S, 271W), *Journal of Atmospheric and Solar-Terrestrial Physics*, doi:10.1016/j.jastp.2009.04.005, 2009.

Welscher, R., A. Hannawald, P. Wüst, S. Lienhart, R. Bittner, M., Analysis of 2D airglow imager data with respect to dynamics using machine learning, *Atmos. Meas. Tech.*, **16**, 3141–3153, doi.org/10.5194/amt-16-3141-2023, 2023.

Zia, K. I., Investigating Atmospheric Gravity Waves Using 3-Dimensional Spectral Analysis, *All Graduate Theses and Dissertations, Spring 1920 to Summer 2023*, **8645**, <https://digitalcommons.usu.edu/etd/8645>, 2022.