Using machine learning to improve SuperDARN data classification

Summary

This project aims to develop a new approach of classifying SuperDARN (Super Dual Auroral Radar Network) data using machine learning algorithms. In the past, this data has been classified using a formula based on elevation angle, which is not always reliably available, or using another formula based on doppler velocity and spectral width which is biased to miscategorize low-velocity ionospheric backscatter (IS) as ground scatter (GS). Recently, researchers successfully applied machine learning techniques to this data, including k-means [Cousins et. al. 2012][1] and a method similar to a decision tree or a depth-first-search [Ribiero et. al. 2011][2]. These approaches improved on past methods, but they used a very limited set of features and relied on simple machine learning methods that do not easily capture non-linear relationships or subtle probability distributions. This project will apply machine learning methods with a focus on using a larger number of well-selected features and using more nuanced algorithms. Initial tests using a Gaussian Mixture Model (GMM) classifier showed high accuracy when compared to an empirical classification method [Bland et. al. 2014, Burrell et. al. 2015][3,4] and to k-means, so this will be a promising algorithm to start with. In this project I will create a generalized machine learning toolkit, use model selection to determine the best set of features, add new scatter labels, and validate predictions using an independent method. At the end of the summer, I will deliver a GitHub toolkit with the machine learning tools, a set-up and usage guide, a report on accuracy and validation, and a graphing tool on the SuperDARN website[5].

Project description

The major goal of this project is to produce a well-documented toolkit that can work with a wide variety of data, is well-validated, and is easy to use. To start, I will test GMM and other algorithms on various types of data to develop a tool that is generally effective. Also, the accuracy is difficult to obtain objectively, because there are no absolute ground-truth labels for this type of data. To address this, the next goal will be to develop validation methods to assess accuracy independent of the empirical model, which is sometimes visibly inaccurate and is based on elevation angle which is sometimes not available or unreliable. Additionally, there are other scatter types aside from IS and GS that are useful to identify, such as meteor scatter and mixed scatter, but those classifications are to obtain with existing methods. For this reason, I will investigate methods of categorizing these other scatter types and add what I find to the machine learning toolkit. Finally, this software should be easy to use and access, so at the end of the summer I will add an interactive graphing page to the SuperDARN website.

As a test, I trained a Gaussian Mixture Model (GMM) on a month of data from the Saskatoon (SAS) high-latitude radar, and the resulting model is 85-95% accurate when compared to the best known empirical model [Bland et. al. 2014, Burrell et. al. 2015]. Figure 1 shows the results over one day of data. The accuracy was good on this month of Saskatoon data, but the GMM algorithm may not generalize to data from different parts of the world (see Figure 2) or from different time periods. Note that the SuperDARN data used in this project is obtained through Davitpy[6], which is an open-source library originally started at Virginia Tech that has become an international collaboration to access and visualize SuperDARN data as well as other space science datasets and models.

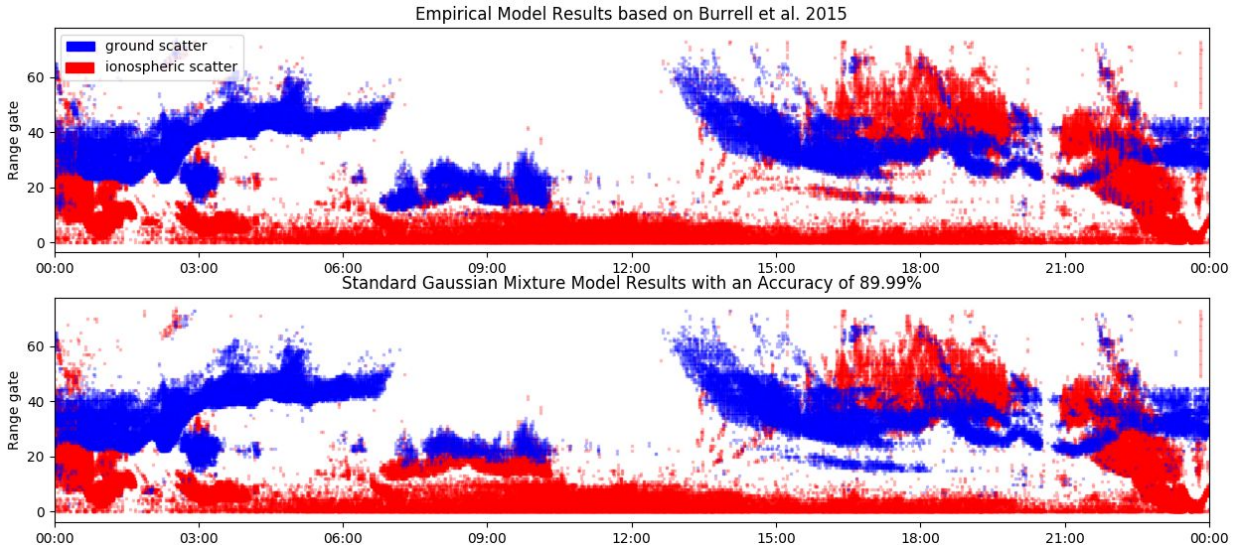

Figure 1: *Gaussian Mixture Model compared to empirical model*

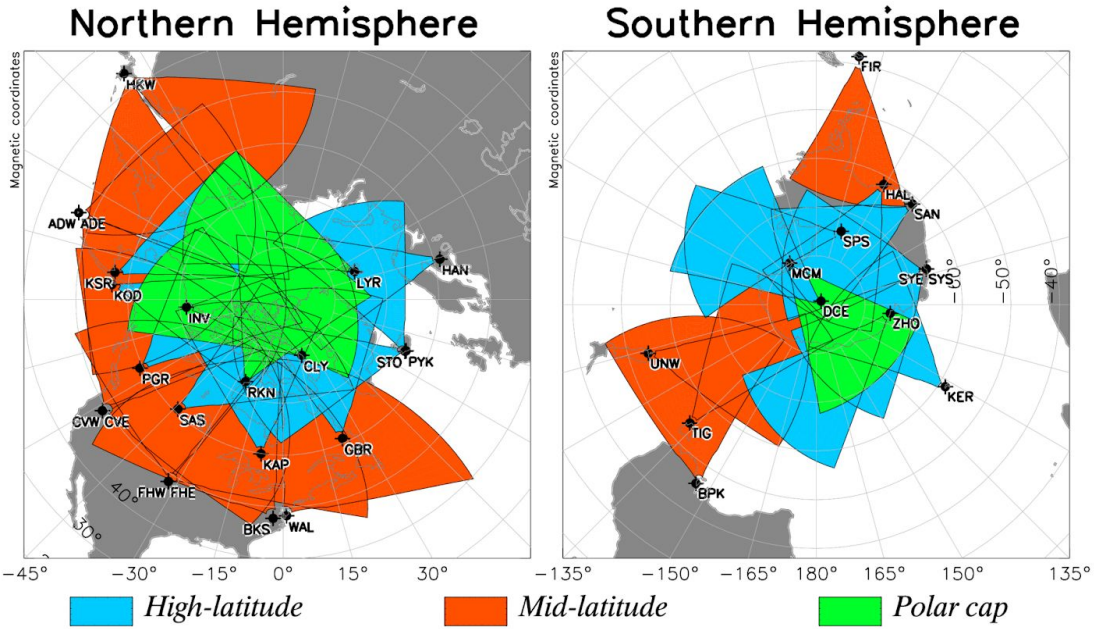

Figure 2: *SuperDARN radar coverage in the Northern and Southern hemispheres*

Deliverables

- A GitHub project with the machine learning tools and documentation.
- A report on validating the machine learning algorithms independent of the Bland et. al. empirical method.
- Implementation of new scatter categories such as meteor scatter and mixed scatter.
- A page on the SuperDARN website with interactive graphing, so that a visitor can tune parameters and choose to use any available SuperDARN data.

Optional deliverables

If there is any extra time, I will also work on one or more of these projects:

- Modify the machine learning algorithms to work with other dataset types, such as satellite data.
- A publication discussing the project and results.
- A presentation at the 2018 SuperDARN summer workshop.

Timeline

I will be unavailable the week of 5/14, and available 40 hours per week for the rest of the GSoC session, 5/21 to 8/14. Figure 3 shows the timeline.

5/21	5/28	6/04	6/11	6/18	6/25	7/02	7/09	7/16	7/23	7/30	8/06
Create a machine learning toolkit											
			Validate the predictions								
							Add new types of scatter				
										Add to the website	
Documentation											
			Eval				Eval				Due 8/14

Figure 3: *Timeline*

1. Create a machine learning toolkit [5/21 to 6/11, 3 weeks]

- Add new features to the machine learning algorithm, such as the Kp geomagnetic index.
- Add statistical information and refine model selection by implementing Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC).
- Test the algorithm under various conditions – no elevation angle, various latitudes, different seasons, time periods with different solar activities. Document and debug problems.
- [Optional] Implement neural network and/or Hidden Markov Model (HMM) for comparison.
- Create a GitHub toolkit with the code.
- Produce online documentation.

2. Validate the predictions [6/11 to 7/09, 4 weeks]

- Research methods of validating radar scatter. Start with how the Bland et. al. method was validated. Also make note of any methods for validating the types of scatter which will be added later, including mixed scatter and meteor scatter.
- Based on that research, write code to test the algorithm's predictions.
- If problems become apparent from that testing, attempt to debug.
- Produce a report on the results of validation, and update the GitHub repository with the new code and documentation.

3. Add new types of scatter [7/09 to 7/30, 3 weeks]

- Investigate characteristics of other types of scatter. Determine which categories will be most useful to researchers, and which categories are well-understood and can be modeled. Look for validation methods.
- Add new categories to the algorithm. Which categories are added will depend on the results of research.
- Test, validate, and document.

4. Add to the website and wrap up the project [7/30 to 8/13, 2 weeks]

- Communicate with the team that is overhauling the Space@VT website to learn the system for creating interactive graphing pages.
- Set up a dummy webpage on the Space@VT site with the desired interface, then slowly add functionality.
- Also use this time to clean up and improve on documentation.
- Submit code, project summary, and final mentor evaluation by the 8/14 deadline.

Benefits to the Community

Improving the way we classify radar backscatter puts better tools in the hands of researchers studying the Earth's atmosphere, substorms, and other space phenomena. Many existing methods for classifying backscatter rely on elevation angle, which is sometimes inaccurate or unavailable. Other methods are biased to classify low-velocity ionospheric scatter (IS) as ground scatter (GS) in order to obtain clean IS data, but that method causes them to exclude a certain amount of low-velocity IS data and produce unreliable GS data. Currently, it is also difficult to obtain meteor scatter and mixed scatter data, and those categories get misclassified as IS or GS. This project's machine learning method may provide more reliable scatter classifications without using elevation angle, because it mainly relies on time and distance and doppler velocity. It also does not use data filtering, unlike other existing machine learning methods, so it will preserve the resolution of the data. Additionally, this project will attempt to add new classes of scatter, which will produce cleaner GS and IS as well as providing new types of data for research. As a result, more good-quality data will be available from SuperDARN radars for researchers to use.

Task report

I. Problem Statement

This project aims to predict the Disturbance Storm Time (DST) index using Interplanetary Magnetic Field (IMF) data, particularly the Z component of magnetic field, Bz. The DST index measures magnetic field produced by ring current flowing around the earth's equator. The Interplanetary Magnetic Field is produced by solar wind, which is plasma that has been ejected from the sun (coronal mass ejection). This project uses a neural network to model a year of IMF data and predict DST indices. The resulting model can predict the general trend of DST, but it usually fails to predict the exact value. However, this is just a prototype, and the accuracy could be improved using a few different methods that have not been tried.

II. Solution

Since we have both the input data (IMF Bz) and the output labels (DST), this is a supervised learning problem. Neural networks are good at capturing relationships between input and output that may or may not be linear, and they are often effective, so I chose to model the data using a neural network. The only feature I used was IMF Bz data from the full year of 2015 (1-hour averages), and the labels were the hourly DST indices for the same time period.

At first, the predictions would cut off at a certain threshold, not making any prediction above a certain value. This turned out to be a problem with the data not being processed well for the ReLU activation function. I scaled the Bz data to be all positive and have a larger magnitude,

which fixed the problem. The R^2 value was low at around 0.1-0.2, because the predictions follow the trend somewhat but rarely predict the correct value. I experimented with adding different features including proton density and plasma flow speed from the IMF data set, but found that they decreased accuracy. This may indicate that they also need special scaling and preprocessing, or they may not be well-correlated with the DST.

Figure 4 shows the predictions (orange) vs. truth (blue) over 3 months on test data. You can find this and more graphs at: https://github.com/e-271/SpaceVT-GSoC-Demo/tree/master/graphs

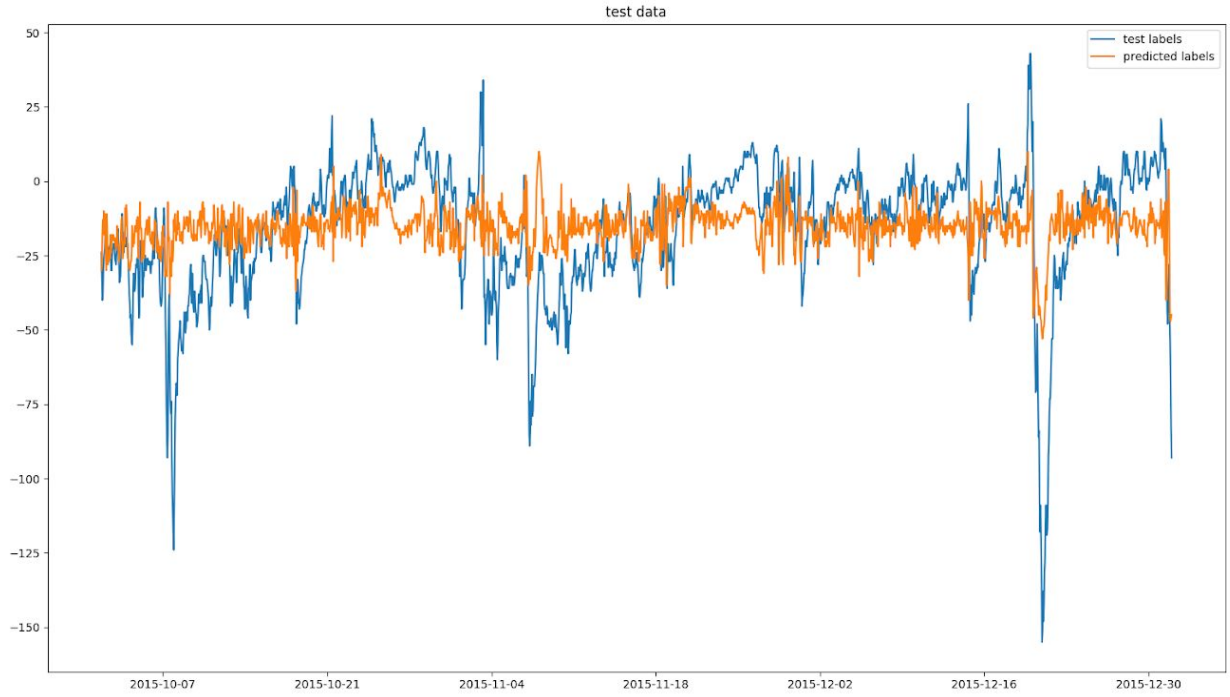

Figure 4: *Recorded DST (blue) and predicted DST (orange) over 3 months*

III. Conclusions

Neural networks may be a promising method for predicting the DST, but this method would require adding more features, using more data, and using careful scaling in order to make accurate predictions. It is also possible that there is a time lag between Bz and DST. Since we are using 1-hour averages it would need to be on that timescale to have a significant effect, but if we can calculate that lag we could correct for it. IMF data is available as far back as 1963, and DST data is available from 1957, so we could potentially train the network on 50 years of data. We could also pull in features from datasets other than IMF. In conclusion, this algorithm is decent at predicting the trend of DST, but is inaccurate in predicting the actual values. However, the predictions might be improved by a number of methods, including adding new features with good pre-processing and using more data.

IV. References

OMNI data documentation https://omniweb.gsfc.nasa.gov/html/HROdocum.html
Source of DST data http://wdc.kugi.kyoto-u.ac.jp/index.html
Source of IMF data https://cdaweb.gsfc.nasa.gov/pub/data/omni/
GitHub page for this project https://github.com/e-271/SpaceVT-GSoC-Demo

Bio-sketch

I am studying Computer Engineering at Virginia Tech, and will graduate with a Bachelor of Science in December 2018. My education background includes classes in statistics, linear algebra, machine learning, and artificial intelligence, and my major interests in Computer Engineering are statistics and machine learning. My work background includes designing a website front-end for a machine learning script. I recently completed a research internship where we investigated Android 7's newly-added GNSS data logging capabilities, with a goal of obtaining centimeter-level positioning, which we found was possible on devices with a certain GNSS chip. I served as the coding lead for a Python toolkit we created to process data and graph the experiment results. I am interested in this project because I see a lot of potential for machine learning algorithms to improve our understanding of scientific data, and I want to develop tools that will help researchers study the final frontier.

References

1. *Cousins, E. D. P., and S. G. Shepherd (2012), Statistical characteristics of small-scale spatial and temporal electric field variability in the high-latitude ionosphere, J. Geophys. Res., 117, A03317, doi:10.1029/2011JA017383.*
2. *Ribeiro, A. J., J. M. Ruohoniemi, J. B. H. Baker, L. B. N. Clausen, S. de Larquier, and R. A. Greenwald (2011), A new approach for identifying ionospheric backscatter in midlatitude SuperDARN HF radar observations, Radio Sci., 46, RS4011, doi:10.1029/2011RS004676.*
3. *Bland, E. C., A. J. McDonald, S. de Larquier, and J. C. Devlin (2014), Determination of ionospheric parameters in real time using SuperDARN HF Radars, J. Geophys. Res. Space Physics, 119, 5830–5846, doi:10.1002/2014JA020076.*
4. *Burrell, A. G., S. E. Milan, G. W. Perry, T. K. Yeoman, and M. Lester (2015), Automatically determining the origin direction and propagation mode of high-frequency radar backscatter, Radio Sci., 50, 1225–1245, doi:10.1002/2015RS005808.*
5. *VT SuperDARN website http://vt.superdarn.org/*
6. *DaViTPy https://github.com/vtsuperdarn/davitpy*