Semester Report Fall 2020

Anna-Marie Bals

Research paper

Semester Report Fall 2020

Anna-Marie Bals

December 9, 2020

Department of Physical Sciences
Embry-Riddle Aeronautical University

Anna-Marie Bals. *Semester Report Fall 2020.* Research paper, Embry-Riddle Aeronautical University, Florida, 2020.

Department of Physical Sciences , Embry-Riddle Aeronautical University, Daytona Beach, https://daytonabeach.erau.edu/college-arts-sciences/physical-sciences.

Abstract

Irregularities in Global Navigation Satellite System (GNSS) signals in high latitudes have different sources that cause scintillating signatures, which are mostly precipitation, gradient drift instability and Kelvin-Helmholtz-instabilities. Those structures are only understood yet for certain events and case studies. In order to gain a deeper insight into scintillations in high latitudes and their sources, we are making use of the huge volume of recorded data over the last years and therefore investigate different machine learning approaches to classify and categorize scintillation events and draw conclusions about physical background processes. For the the geomagnetic storm on the 9th of March 2012 we apply a simple machine learning approach to categorize the temporal scintillation signatures according to their geomagnetic source region. By use of a hierachical clustering analysis on high rate data in phase and power we are able to distinguish manually selected events from stations inside the polar cap vs those from the auroral oval. Together with geomagnetic background data we are finding input features and indices that can be used by our model to detect the scintillation signatures caused by particular irregularities and extract those candidate events to be further analyzed with inverse modeling. Based on the evolving database of sets of events we expect to understand more about the importance of the major sources of scintillation in each of the source regions. On top of that, we are investigating how to predict the physical characteristics of the irregularity and its source from the scintillation signatures we observe. We are applying a basic version of a CNN model to understand if we can automate the ML classification and clustering. This work is done to have a preliminary understanding of which method would work best for our requirement of analyzing high rate scintillation phase and amplitude data and classifying it based on scintillation signatures. This to our knowledge has not been attempted thus far.

Contents

1 EP800 Fall 2020

1.1 Introduction

The following report describes the progress achieved during the Fall 2020 Semester for the EP 800 In-depth Dissertation Study.
This project started as work on the classification of irregularity signatures in GPS radio data that has travelled through the ionosphere in high latitudes. The idea is to engage Machine Learning techniques to extract signatures due to different irregularities from data at different high latitude receiver locations.
The ionosphere is a region in earth's atmosphere that contains charged particles in cloud-like structures that can be moving at possibly very high speeds. Therefore, as global navigation satellites are passing over the high latitudes, the EM radio waves can be severely distorted by those irregularities and the signal received on ground will show rapidly oscillating signatures in phase and amplitude, so-called ionospheric scintillations, which is shown in Figure 1.1 schematically. While this is bad news for all GPS applications, this effect gives us the chance to learn more about the irregularities and their structures based on years of recorded data at many different locations. It is not clear yet how ionospheric irregularities develop in different regions in the high latitudes. From the modeling approaches we expect that different scales and structures will lead to particular signatures. If we were able to extract those signatures from observed data, we could describe the irregularities, their parameters and how they are created. As a consequence this could help tune current models for high latitude ionospheric scintillations and extend their applicability into much more general use cases.
As a fist step we conducted a feasibility study in form of a hierarchical clustering study for

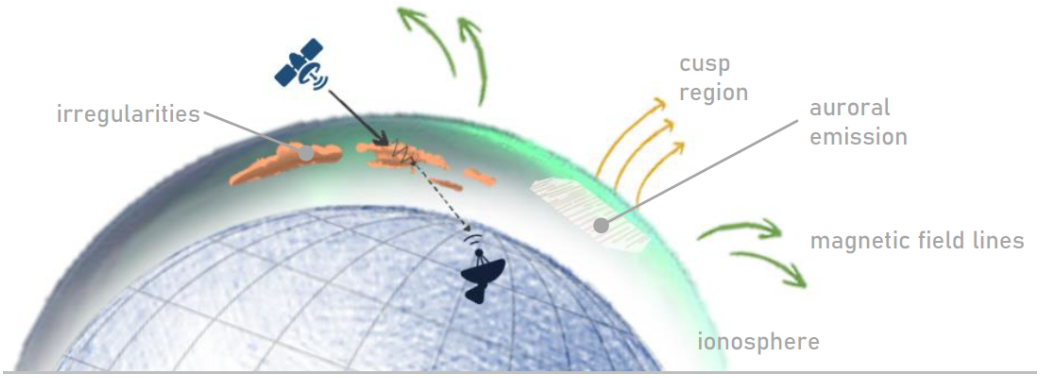

Figure 1.1: Schematical conditions in the ionsophere in high latitudes

one case study on 9th of March 2012. The following chapters are going over the results of the first study and based on that we describe the progress achieved in this semester in the different aspects. At the end we draw conclusions and look out onto future work.

1.2 Motivation

In order to gain a better understanding of what kind of signatures we are looking for and what the circumstances are for the corresponding irregularities to develop, we will take a closer look at the high latitude ionosphere. Therefore we will consider some more details about the three major source regions that we are distinguishing due to different geomagnetic conditions. The cusp region appears on the dayside and describes the area where the ionosphere is coupled to the magnetosphere and to the solar radiation, which leads to active regions of ionized particles in the ionosphere as shown in yellow in the schematic diagram in Figure 1.2. Due to magnetic reconnection, smaller clouds can be separated from the active region and start drifting towards the pole region, which are called polar cap patches and shown in pink. Thirdly, if energized particles enter regions of current and through collision they can cause auroral emission, the region is called auroral oval and is indicated in green in the diagram.

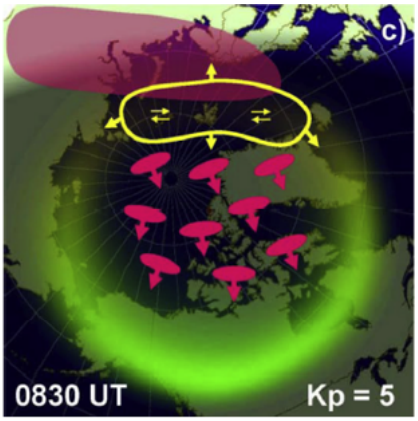

Figure 1.2: A schematic illustration of active space weather regions in the polar cap ionosphere when IMF BZ is south (Kp = 5). The active regions for creation of polar cap patches/plasma irregularities are shown in yellow color and move under the influence of IMF as indicated with yellow arrows. For IMF BZ south the tongue of ionization (pink) extends into the dayside auroral oval, where magnetic reconnection chops it into polar cap patches (pink) that begin to drift across the polar cap. In the production region there are flow channels and strong flow shears that initiate the growth of ionospheric irregularities. [Moen et al. 2013]

There are three predominant irregularities present in these three regions in mixed contributions. As to how often they occur in which region and how they look structurally in detail, these are still open questions, but they are caused by different mechanisms. In the first case, energetic particles arriving from the sun enter into the ionosphere mainly in the cusp region but flow also into the auroral region and therefore lead to precipitation

in different scale sizes. Another big influence factor can be found regions where strong velocity shear occurs and introduces instabilities in intermediate-scale size that lead to turbulence by cascading into smaller scales. This irregularity is mostly present in the auroral oval and also in the cusp region. Finally, regions with different density gradients will try to form an equilibrium by mixing in intermediate-scale sized structures. This happens mostly in the polar cap region where polar cap patches with different density gradients are drifting.
So now we have these three different geomagnetic regions and three different predominant effects appearing. But in order to learn more about the Physics behind each of these instabilities and the interaction between different regions, our goal is to extract observed signatures of each of the irregularities for different geomagnetic background conditions in order to tune our models for as general use cases as possible. On the bottom of this slide there is a collection of simulation and signature obtained through inverse modeling for one specific event for each of the three predominant instabilities. So the questions we are asking ourselves are: How can we describe the relationship between scintillation signatures we have recorded and the sources of irregularities that are responsible for the scintillation? So basically how can we connect the signature to actual geomagnetic conditions and parameters? And in order to narrow it down, we want as a first step to categorize the high latitude irregularities for both temporal and spectral scintillation signatures first with a Machine Learning infrastructure and once we have enough data from enough events with analytics. Also, what is not yet certain as well is how frequent the major sources occur in different regions and how important their contribution is to the generation of scintillation. And to break it down, the question guiding our work for the next steps of analysis is: Can we see differences between signals recorded at different source regions?

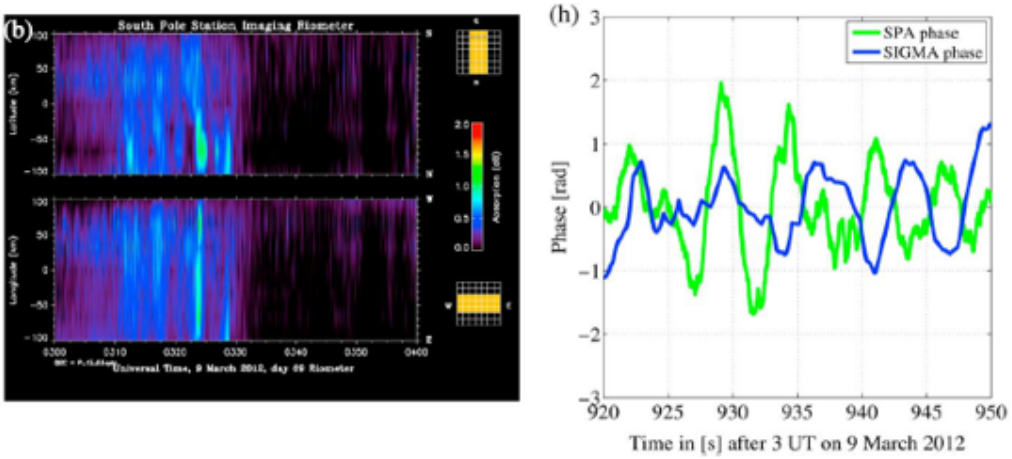

Figure 1.3: Example of a signature likely containing precipitation with imaging riometer data [3, Deshpande et al., 2016]

The first step to find the answer to that question is a very basic hierarchical cluster-

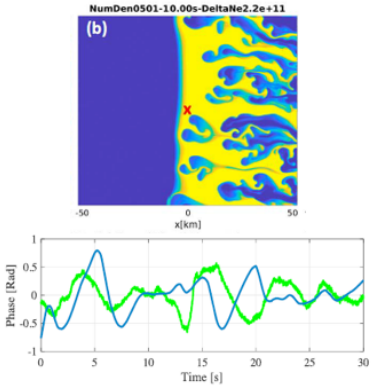

Figure 1.4: Example of a gradient-drift instability signature and spatial electron density [4, Deshpande and Zettergren, 2019]

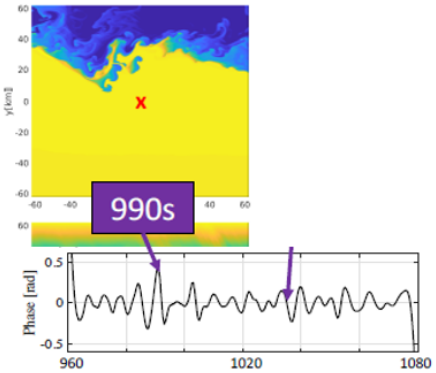

Figure 1.5: Example of a Kelvin-Helmholtz instability signature and spatial electron density [11, Spicher et al., 2020]

ing approach that investigates the correlation and similarity of event time series from different stations.

1.3 Previous Work and Objectives

The most basic question to classify high latitude irregularities is whether or not the events look different for different source regions. Since different irregularities are believed to be present in different high latitude regions, the different scales and structures of the predominant irregularities are expected to lead to different features in the scintillation signatures. Specifically, for the following work, we posed as question: Can we distinguish whether a signal was recorded at a polar cap station vs. an auroral oval station based on the signatures it contains in phase and amplitude?
Since to our knowledge no extensive studies investigating differences between polar cap and auroral oval stations have been conducted yet, we will start by analyzing the similarity of signatures by using basic Machine Learning to cluster the event signatures. If we can see patterns in the hierarchical clustering that lead to grouping the signatures from the stations into two different groups, we know that the data can be distinguished and we can search for the criteria to keep them apart. Therefore, in the following we describe the first approach as a feasibility study and based on the results we define the objectives for the work presented in this report.

1.3.1 Feasibility Study: Hierarchical Clustering

In order to distinguish signatures from polar cap or auroral stations, we have developed a very basic approach consisting of three major steps

- extracting a set of events
- determining their similarity

- searching for patterns in the resulting hierarchical cluster

for which we have investigated different procedures to compare the performance of the analysis depending on various influence parameters.

Event Extraction

We decided to use data from the geomagnetic storm on March 9th 2012 for a very active time segment during UTC 03:00 - 04:00. The reason being is that there is a lot of literature available and the event has been pretty well studied in its impact on the ionosphere in high latitudes. As a result, we can formulate an expected outcome for our study and compare our results to well understood geomagnetic conditions.

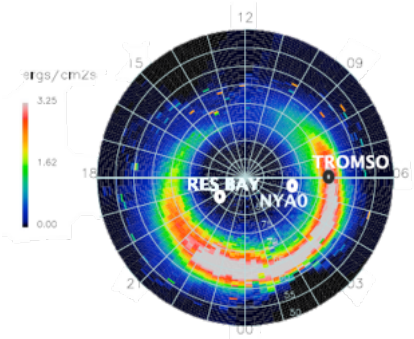

Figure 1.6: OVATION Prime forecast of Northern auroral oval at 3:30 UT on 9 March 2012 displayed in magnetic local time and magnetic coordinate system. The yellow squares are the locations of GPS stations from where TEC measurements were obtained. Pink squares are radio occultations, and grey circles are SSUSI 1356 emissions. [3, Deshpande et al., 2016]

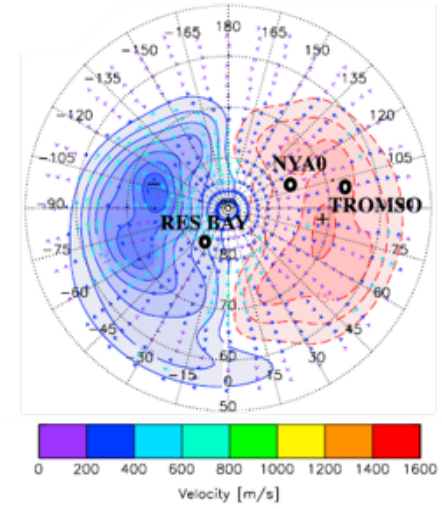

Figure 1.7: Convection velocity maps derived from SuperDARN data in northern high-latitude region at 3:30 UT on 9 March 2012. The colored convection velocity vectors display the magnitudes based on the color bar. [3, Deshpande et al., 2016]

In overlap with [3, Deshpande et al., 2016], we chose as the four stations to be analyzed in the Northern and Southern Hemisphere as shown in Table 1.3.1. As shown in Figures 1.6 and 1.7, Tromso And Ny-Alesund are located in similar longitudes, leading to both stations lying in the nightside. This is necessary since we want only to analyze the polar cap and the auroral region for now, not the cusp region on the dayside. From Figure 1.6, it is confirmed that Tromso is lying inside the auroral oval form the OVATION Prime forecast. Ny-Alesund on the other hand by referring to both Figures is located inside the polar cap.
The event was indicating the beginning on a geomagnetic storm and had a Kp index of about 6. [10, Prikryl et al., 2015] It contains enough active segments that are longer that

station	institute	receiver	MLAT
Tromso	Bath	Novatel	66.7 ° N
Ny-Alesund (NYA)	INGV	Novatel	78.9 ° N
McMurdo (MCM)		Novatel	80 ° S
South Pole (SPA)		Novatel	74.3 ° S

Table 1.1: stations with 2012 data used for feasibility study

30 [s] and is very well recorded by different instruments over the whole globe.
For the length of the active events we decided to search for 30 [s] segments. This allows for a long enough window to also perform spectral analysis of the signatures later. And from statistical analysis it is known that for a segment length of 30 [s] the optimal trade-off between segment length and number of active segments can be achieved. [5, Jiao et al., 2013]
Before events can be extracted, the raw data has to be parsed and processed. In order to get rid of the (probably memory) effects in very low frequencies, we are using a Butterworth filter with 0.1 [Hz] as cutoff frequency.
Since UT hours 03:00-04:00 were very active, long segments of scintillation were extracted from the data manually and later split into 30 [s] intervals. An example time series for Tromso and Ny-Alesund is shown in Figure 1.8. The manual extraction has the great advantage that it can distinguish spikes due to anomalies in the data from actual active events and therefore mostly prevent faulty data to enter the analysis. For a trained data analyst, the manual extraction will lead to the highest event detection performance.
However, in some cases it is really hard to determine whether or not scintillation is present, especially if background noise is present, as can be seen in Figure 1.8. On top of that, for this step only longer segments were extracted that contained activity from the hourly PRN data. Because of the 30 [s] interval length it cannot be guaranteed that all the intervals that were split apart contained significant activity.

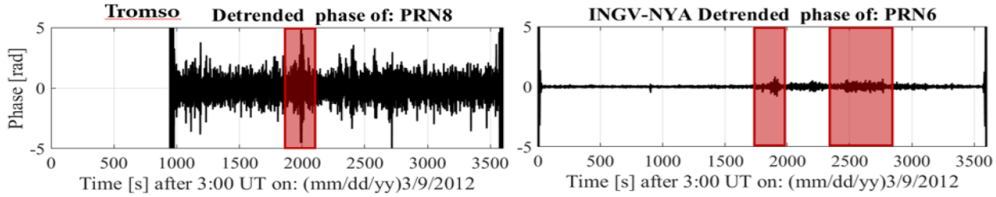

Figure 1.8: Example phase time series (PRNs) for Tromso and NYA on the 9th of March 2012

In addition to the manual extraction of segments, an automated approach was written and tested. The detection criterion is simple thresholding in phase and amplitude for different intensities of activity. However, as can easily seen from Figure 1.8, every station

and PRN has different noise levels and therefore needs its own thresholds which have to be set manually by at least looking through the data beforehand. So it is safe to say that this approach is unfortunately only semi-automatic and still too time-consuming to follow up on.

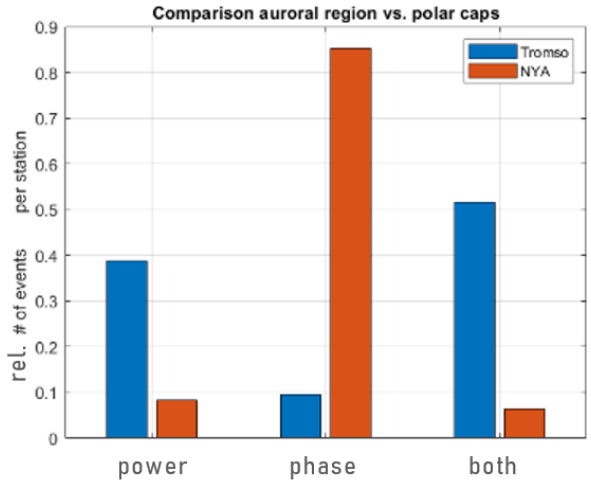

Figure 1.9: Relative number of events in phase, power and phase and power combined for Tromso and Ny-Alesund on March 9th 2012 in UT 03:00 - 04:00 after automatic event extraction

As can be deduced from Figure 1.9, mostly phase events were extracted by the automatic approach for Ny-Alesund, while Tromso has a lot of simultaneous events in phase and power as well as significantly more power events.

Agglomerative Hierarchical Cluster

One of the simplest techniques to cluster data according to its properties is a dendrogram or so-called hierarchical clustering. The goal is to compare the similarity of two signals with each other and describe their similarity as a number. Segments that look very similar will be grouped together via connection with a link, then on the next level these branches of groups are connected so that a multi-layer tree-structure emerges.
We are determining the similarity of the time series by calculating the Euclidean distance for all elements pairwise of the input matrix compared against each other with the cost function (Matlab function *linkage*). We do this for corresponding power and phase segments (15 per station per run to not create artificial bias) and combine their correlation matrices to determine their proximity in a 2-D plane consisting of phase and power. The linkage function will group the elements in a tree, where each branch connecting two events describes that they were most similar and therefore forming a binary cluster. This will lead to a tree with several branches that have many sub-levels of binary clusters of close events. In the last step, it has to be determined how to cut the branches of the tree to achieve the most significant and meaningful labels of events into either auroral oval or

polar cap stations. We are plotting dendrograms for power only, phase only and power and phase simultaneously as in Figure 1.10.

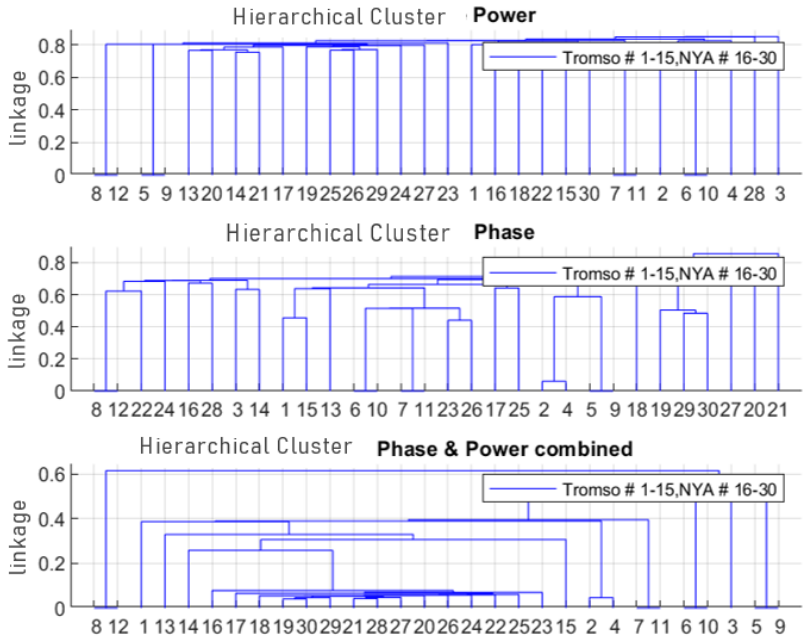

Figure 1.10: Amplitude, phase and simultaneous phase and amplitude linkage for Tromso (samples 1-15) and Ny-Alesund (samples 16-30) on March 9th 2012 in UT 3-4

Results

Taking a closer look at the three dendrograms in Figure 1.10, we can see that the segments in power have very similar linkages with no apparent pattern. In the phase hierarchical tree we can see few patterns, but none that apparently help us answer the question about the differences between polar cap and auroral oval stations. However, considering the 2-D case for simultaneous linkage in power and phase, we observe a big difference in linkage between polar cap segments (samples 16-30) and auroral oval segments (samples 1-15). All polar cap segments have linkages below 0.15 and lead into a single branch. In contrast, all the auroral oval segments join at linkages significantly larger than 0.2.
As a consequence, we determine 0.2 as cutoff linkage to classify the samples into polar cap vs auroral oval stations. However, also a few auroral oval samples have linkages smaller than 0.2 before they join their part of the tree. This is probably due to samples without significant activity which are therefore containing mostly white noise looking similar in different samples.
By visually extracting the samples with their labels based on the branching of the tree and not on the absolute values of the linkage (meaning for example sample 2 and 4 still

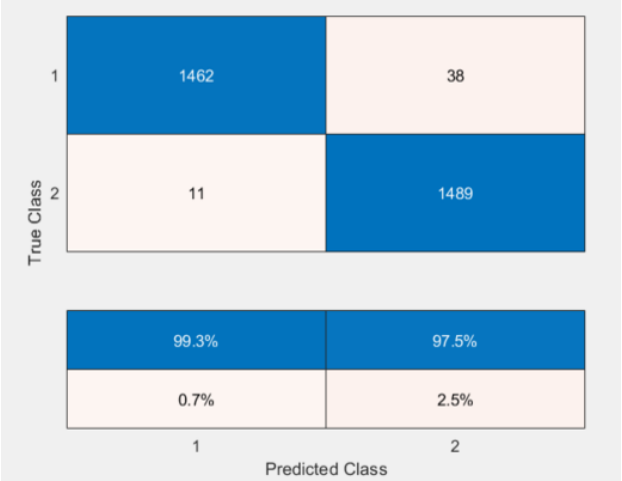

Figure 1.11: Confusion matrix for 2012-03-09 Tromso vs. NYA (left); Class 1 = Tromso (auroral station), Class 2 = NYA (polar station)

count as Tromso even though they have linkage values well below 0.2), we achieve the confusion matrix shown in Figure 1.11. With our procedure we obtain a very high ability of classifying the sample correctly as to which region it was recorded in.

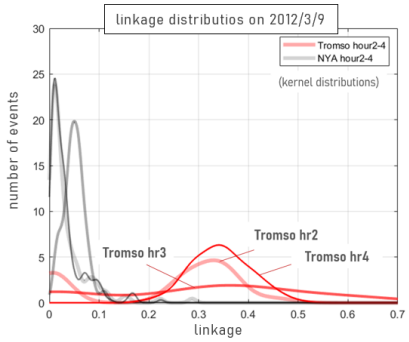

Figure 1.12: Simultaneous phase and amplitude linkage between signatures from Tromso and NYA over 3 hours of UT on March 9th 2012 in UT 3-4

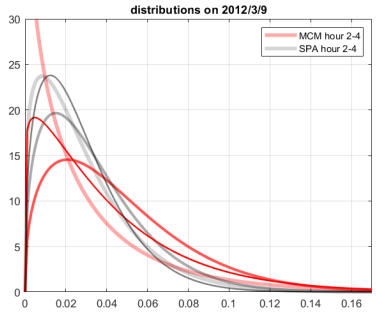

Figure 1.13: Simultaneous phase and amplitude linkage between signatures from McMurdo and South Pole over 3 hours of UT on March 9th 2012 in UT 3-4

Since the activity was ongoing for several hours on the 9th of March 2012, we are investigating how the linkage changed for both stations over the course of three hours. The results can be found in Figure 1.23 in the form of distributions that have been estimated based on histograms with the kernel function. From what we know about the geomagnetic storm, especially UT 3 was the most active hour. Both stations show a wider spread of linkage values in hour three, as well as a peak at higher linkage values. Ny-Alesund is very clearly distinguishable from Tromso at lower linkages. Tromso distributions can have two significant peaks, one around zero and one at higher linkages.

This is due to Tromso samples that contain mostly noise and therefore look very similar to each other. Even Ny-Alesund in hour 3 has another part of the distribution left of the peak probably due to less significant activity.
After performing the same kind of analysis for McMurdo and South Pole station, we obtain the distributions over time displayed in Figure 1.24. Again especially for McMurdo in UT 3, the spread gets wider and the peak happens at higher linkages for more activity. However, both stations have really low linkages and overlap for most of their distributions. When correlating these results with the geomagnetic conditions at that time at both stations in [3, Deshpande et al., 2016], it can be observed that probably neither of the two stations was located inside the auroral oval in UT 3-4 with McMurdo even possibly more trending towards the cusp region. The results show that the data for the Southern Hemisphere stations is much quieter and has less dramatic events as in Tromso. On top of that, there are less significant events happening in phase and power simultaneously.

1.3.2 Extended Approach

Based on these results that we obtained only from the feasibility study, we want to collect a set of questions and ideas that will guide the work described in the next sections to explore further the possibilities of simple Machine Learning approaches for classification. These questions are defining the tasks leading on to the results presented at the end of this report. Especially, we want to make sure to be aware of the limits and challenges we are encountering while building the Machine Learning infrastructure and therefore need to ask critical questions in order to make sure that our results are reliable and that we understand what our framework can and cannot achieve.

Data

This feasibility study was only performed for events from two hours UT for four stations in total. The really dramatic effect of Tromso being very distinguishable from Ny-Alesund could unfortunately not be reproduced for the stations in the Southern Hemisphere, since non of them was inside the auroral oval.
Therefore, the first big question is how the procedure will perform on data from other stations? Specifically, what will happen for data from different receivers and in different regions of the auroral oval or polar cap?
On top of that, every geomagnetic storm has a slightly different morphology. That is why it will be important to assemble a set of data from different years to account for changes in solar activity and interaction with geomagnetic conditions.
So a big objective is to incorporate as much more data as possible also from different stations and with different parameters.
A big impact that was not known yet at the beginning of the study was an issue with the Tromso receiver in the week of 9th of March 2012. Apparently, the receiver was struggling with oscillator issues, which leads to the records being very noisy throughout the whole week. This might be a strong explanation as to why Tromso is so well distinguish-

able from Ny-Alesund in the hierarchical clusters.
In order to eliminate background noise from PRNs, a procedure has been developed by [2, Deshpande et al., 2012] to cut quiet segments from PRNs that don't show as much activity in order to add them up to a reference PRN for the whole hour UT. This reference PRN will then be subtracted from the PRNs which actually contain the signatures to enhance the signal-to-noise ratio.
When implemented, the background noise level can be significantly decreased as shown in a spectral analysis of Tromso data in hour 3 in Figure 1.14. With noise elimination through reference satellite subtraction the power spectral density of the PRNs could be reduced from the blue line to the red line.

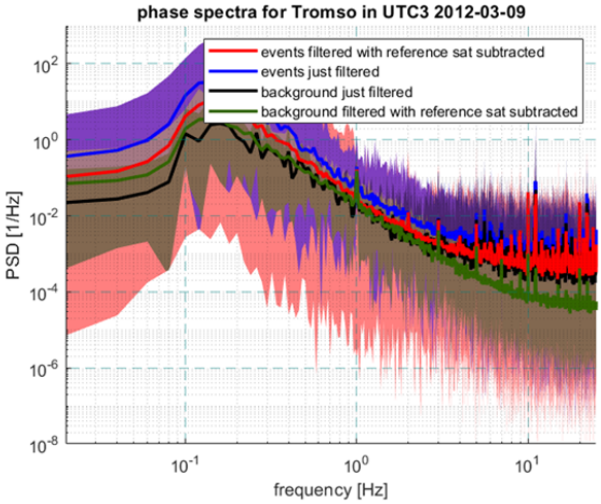

Figure 1.14: Spectral analysis of time series recorded at Tromso on March 9th 2012 in UT 3-4 during a geomagnetic storm that were extracted with and without comparison to background segments extracted from reference satellites [2, Deshpande et al., 2012]

A challenge in using reference satellites is that the quiet segments have to be identified manually, which takes a lot of time. On top of that this technique so far is only implemented for PRNs that are complete within one hour of UT. That reduces the amount of available data for the actual classification task significantly. Since we don't have too much data available so far and are trying to work with data already well known in the literature, an open question is to whether the routine can be altered to be able to also work on incomplete PRNs.
And finally, there is so far no way of determining the significance of the extracted segments, so that we cannot really tell how close they are to the actual background noise. However, in implementing the classification we also want to find an answer to the question until which significance of events the procedure still confidently classifies the segments correctly. As we are mostly working with extremely active segments for now, we want to be able to make a statement until what level of activity the classification is possible before the events become to similar to the background noise and cannot be

distinguished between stations.
So the question is: Is the effect showing up in the feasibility study repeatable for other stations in other years? And is there a way to prepare the data to become comparable between different stations even though they might use different receivers, from different times and different quality?

Event Extraction

So far, the segments used for our analysis have mostly been extracted from the data by hand to make sure they are trustworthy and significant enough. Manual event detection is still considered to be the optimal performance event detection that all automatic approaches are compared to. That is especially true for the field of scintillation detection where each time instance has a binary label as scintillation present or absent. This said, the enormous downside of this approach is the time effort required to process years of data for many different locations, which is an impossible mission.
This is the reason why we are exploring several possible methods to detect and extract events automatically, since it will play a crucial role in whether our Machine Learning infrastructure will be successful or not.
The automatic thresholding may be a start, but it is too rigid to cope with the data we are working on. How can we develop an event detection algorithm that can adapt to different receivers and different noise levels that might be changing within the hour? How can we make sure to identify issues with filtering or gaps in the data leading to an exclusion of low-quality data segments?
And finally: How can we assign a measure of significance to the segments we have extracted in comparison to the overall noise floor of the PRN in that hour/day UT? How can we extract favorably segments that have for example a similar length to our Machine Learning input segments?

Classification

The more datasets of significant time segments we have for different stations, the more thoroughly we can train and improve our Machine Learning approach.
Using the feasibility study as a first step, one interesting question is: Can we use the linkages determined by the hierarchical clustering for phase, power and phase and power simultaneously to train a Machine Learning algorithm to classify the stations based on their similarity?
What other approaches can we employ to compare time series and search for their differences? What have other people already been successfully using? What implementation would be completely new in the field of ionospheric scintillations?

1.4 Data Processing

Another really prominent geomagnetic storm apart from the March 2012 case was the so-called St. Patrick's day storm of 2015, with strong activity especially on the 17th of March. Due to very drastic solar activity on March 15th, this storm falls into the category of a super storm. [1, Cherniak and Zakharenkova 2016]

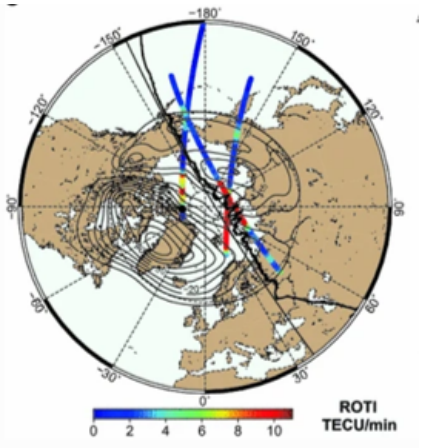

Figure 1.15: SuperDARN polar potential maps for the Northern Hemisphere at 18.0 UT with superimposed low earth orbit (LEO) Rate of TEC (ionospheric total electron content) index (ROTI) (colored lines) and in situ (thick black line) observations. Black dot indicates the position of the magnetic pole. TEC data are the relative slant TEC measurements. [1, Cherniak and Zakharenkova 2016]

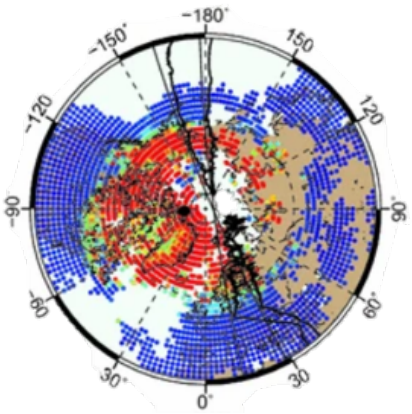

Figure 1.16: Hourly rate of TEC (ionospheric total electron content) index (ROTI) maps over the Northern Hemisphers in geographical projections at 18 UT on March 17th 2015. Geomagnetic poles are marked by black dots. [1, Cherniak and Zakharenkova 2016]

Since different groups have different policies on distributing the data and communicating availability, we are using two of the most transparent groups in the following. Unfortunately, Bath has a big gap in Tromso data for the spring of 2015. That is why we are conducting the following studies with data from the PokerFlat Research Range in Alaska and from Resolute Bay in the CHAIN network, as shown in Table 1.4. Luckily for us, both of the stations have Novatel receivers, nearly the same longitude and very similar latitude to the 2012 Northern Hemisphere stations.
From Figures 1.15 and 1.16, we deduce that both stations were still on the nightside for the active hour UT 11:00, with PokerFlat being located inside the auroral oval and Resolute Bay inside the polar cap.
PokerFlat high rate data can just be requested on the PokerFlat website and is distributed via email as zip-files (see `http://apollo.tbc.iit.edu/~spaceweather/live/?q=download_data/PFRR`). For downloading CHAIN data, access has to be requested for certain email addresses to the FTP server. However, unfor-

station	institute	receiver	MLAT
PokerFlat	PFRR	Novatel	65.4 ° N
Resolute Bay	CHAIN	Novatel	76.4 ° S

Table 1.2: stations used with data from March 17th 2015

tunately only Novatel receiver data can be found in the online depository, while Septentrio high rate data has to be requested separately from the CHAIN maintenance (see `http://chain.physics.unb.ca/chain/pages/stations/`).
In order to unpack the .gps and .nvd files from both stations, we are using the *parsesin* script called by python *novatel_processing* routine, which extracts the raw high rate data into excel files.
The data processing before feeding it into analysis consists of a basic data cleaning with reference satellite subtraction using two reference PRNs. That way we are reducing the background noise level of the stations. Low-frequency effects will be excluded by sending the data through a high-pass Butterworth filter. We are converting signal components I and Q into phase in [rad] and power in [db]. For PokerFlat, the high rate data has to be downsampled from 100 [Hz] to 50 [Hz] in order to be comparable to CHAIN data with 50 [Hz].

1.5 Event Detection

Same as in the feasibility study, we are primarily extracting active segments from the data manually to ensure optimal input data for the algorithm as a sort of benchmark to show how much we could actually classify based on optimal input data sets. The challenge is that in a statistical analysis of the data most of the events have very short lengths, as discussed in [5, Jiao et al., 2013]. In order to actually capture a signature with time-dependent dynamics, it has been found that 30 [s] segments lengths are optimal. They contain enough data points to find a solidly covered spectral signature. While indices like S4 and σ_ϕ are based on 60 [s] intervals, we want to avoid segments longer than 30 [s] in order to prevent the data from containing larger fractions of background noise. That is why we are trying to find the most significant and loudest and most active 30 [s] segments in all PRNs. And this task is most achievable by a manual detection process.
So, breaking it down, basically we are searching for events that ideally last around 30 [s] and not just a spike, without containing background noise and they should have a large signal-to-noise ratio and high scintillation amplitude, optimally both in phase and amplitude. This is what we define as the most significant events: optimal length and loudest amplitudes.
Various successful approaches to detect scintillation events have already been implemented in the past. Especially lately, the scope has been widened to try Machine Learn-

ing to get a hold of years of data. We will go into a few of those references a little more in detail to understand what will be the best approach in our case for the actual Machine Learning task in the next section 1.6. But since they have achieved great progress in event detection itself, we will shortly go through the approaches and results and discuss their relevance to our task. So, does it make sense to use Machine Learning already in the event detection?

In [9, McGranaghan et al., 2018], the authors utilize a Support Vector Machine (SVM) approach with between 28 to 51 input features from numerous auxiliary input channels and the phase index to detect active events and predict phase scintillation for a lead time of one hour. Since we want a computationally efficient detection process only based on signal components phase and amplitude, we are ruling out this approach as an overkill to the requirements.

A very promising approach comes from [7, Linty et al., 2018], that is simply based on the phase and amplitude of the radio signal. Based on the signal components I and Q, the indices S4 and σ_ϕ are calculated and compared with a training set via a decision tree structure. The huge advantage of this approach is its efficiency and easy applicability to high rate data. However, since we are working on a lot of different data from different stations, receivers and different time epochs, the morphology of the data can vary a lot as well, as discussed in section 1.3.2. As a consequence, the decision tree algorithm would have to be trained for each and every station and day. And since we are not yet working with a lot of data yet and have to go manually through it first anyways, this is just too much effort to validate at this point of the analysis. We are merely trying to develop the infrastructure for a Machine Learning event classification for now, which is one step further. As we extract more and more data, we are simultaneously preparing for refining the Machine Learning event detection.

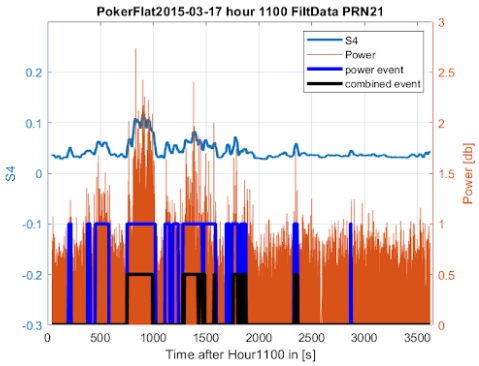

Figure 1.17: Time series with S4 (blue line) for PokerFlat PRN 21 in Hour 11:00 on the 17th of March 2015 with significant power detections in blue binary flag and simultaneous phase and power in black binary

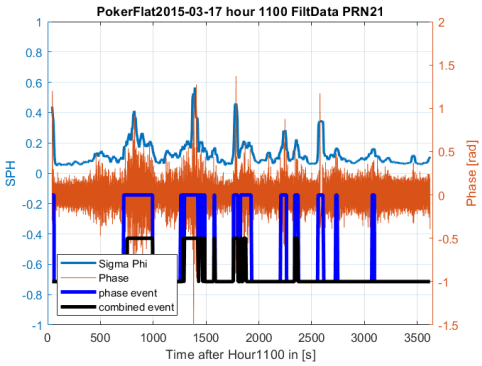

Figure 1.18: Time series with Sigma Phi (blue line) for PokerFlat PRN 21 in Hour 11:00 on the 17th of March 2015 with significant phase detections in blue binary flag and simultaneous phase and power in black binary

Now, in order to still explore some automatic detection methods, we are going one step back into statistics. First we are calculating the amplitude index S4 and the phase index

σ_ϕ as suggested by [7, Linty et al., 2018]. Taking a closer look at the high rate data with its indices, we realized that for segments with lengths somewhere between 5-60 [s] and no activity, the distribution of their average indices over 30 [s] will likely look like a beta distribution with a lot of segments with very small indices and fewer and fewer loud segments. An example of such a beta distribution is shown in Figures 1.19 and 1.20 for 30 [s] time segments. The corresponding time series for PokerFlat PRN 21 in UT 11:00 on th 17th of March 2015 are plotted in Figure 1.17 for power and S4 and in Figure 1.18.

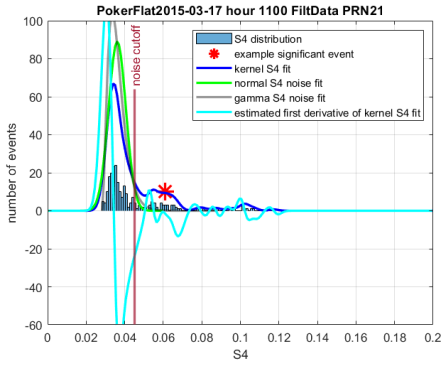

Figure 1.19: Distribution of S4 for PokerFlat PRN 21 in Hour 11:00 on the 17th of March 2015 with example significant event (red star) and approximated distributions with cutoff

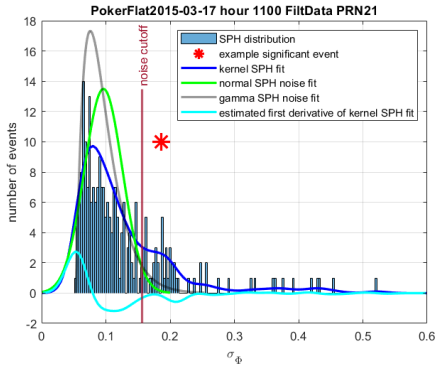

Figure 1.20: Distribution of Sigma Phi for PokerFlat PRN 21 in Hour 11:00 on the 17th of March 2015 with example significant event (red star) and approximated distributions with cutoff

The red star in Figure 1.19 and 1.20 stands for a significant event that was unambiguously identified manually (around 900 [s] in the time series). Now, we can see in the graphs that there are actually two different distributions present and the red star belongs to the one with the higher indices. PRN 21 is also a really active event with a lot of scintillation segments happening in both amplitude and phase. Assuming that the background noise of the PRNs populates the distribution with the smaller average indices, the second distribution in those graphs has to be describing the actual events and is clearly distinguishable from the noise distribution.
For an optimal PRN like this with a lot of difference between background and activity, we can find the kernel distribution estimate and through its derivative find the turning point where the two distributions touch and can be separated, described by the red cutoff line in the graphs. Any segments with average indices higher than the red line will therefore be significant events that can be extracted from analysis. Using the integral underneath the kernel distribution we can determine how much percent of segments are actually above the cutoff. The higher this percentage, the louder and more frequent the scintillations in that PRN. Figure 1.17 and 1.18 contain the blue flag for only power/phase detections and the black flag for power & phase detections. The question is whether this percentage can be formulated mathematically to describe the significance of the event to help detect the strongest activity with a certain segment length?

We are still working on whether the scale of the kernel and beta distribution estimates corresponds to the actual scale of the histograms. On top of that, we are getting S4 indices mostly between 0 and 0.15 and σ_ϕ between 0 and 0.6. Something seems to be the issue with the units for the power index S4, which should also be in the same range as σ_ϕ.
Due to a few issues with the data, noise outliers can be found above cutoff indices. This becomes apparent for example due to windowing in σ_ϕ at the beginning and end of the time series in Figure 1.18. Another issue are gaps in the data or recalibration or instrumentation noise segments that look like scintillation. In our case, they would also be extracted. At that point a Machine Learning detection would have a clear advantage.
On top of that, some PRNs have so few data points or such dramatic noise issues that the distributions are actually not distinguishable. With the approach presented here, these PRNs would unfortunately have to be discarded as a consequence.
Nevertheless, the unbeatable advantage of this approach in comparison to other implementations like [12, Sreenivash, Su and Datta-Barua, 2020] or thresholding itself is that the cutoffs are dynamic and depend solely of the morphology and noise levels of the PRN in that very hour. Also, since mostly phase activity is expected in polar cap regions while stations in the auroral oval might show more simultaneous amplitude scintillations as in [3, Deshpande et al., 2016], we don't even have to adjust thresholds for different regions but just extract whatever stands out from the background noise and use that for further analysis. And on top of that, the length of the segment to be extracted is already considered in forming the beta distributions. We don't just detect scintillation, but we detect the optimal 30 [s] long scintillations.

segment length [s]	phase event detections [%]	amplitude event detections [%]	simultaneous phase & amplitude detections [%]
10	17	24	
20	18	23	
30	16	23.6	15.6

Table 1.3: stations used with data from March 17th 2015

Changing the length of the segments in which the average indices are formed does not influence the outcome significantly, as shown in Table 1.5 for PokerFlat PRNs in UTC 11:00.
One big improvement of the presented approach would be forming averages with overlap to find the beta distributions of phase and amplitude. That way, the exact most active segments could be extracted even better without background noise that comes from a cut that starts too early or too late.

1.6 Classification with Machine Learning

Now we want to take a look at some very interesting Machine Learning applications that are already implemented in the field of ionospheric scintillations and one from substorm analysis. In general, Machine Learning methods can be categorized in different groups, based on whether they are to be applied to continuous vs. discrete data or whether we know details like labels or patterns about the data we are investigating or if we want to cluster the data based on patterns and correlations determined by the Machine Learning model and not by any predefined relationships. A rough overview of approaches in Machine Learning for different requirements is depicted in Figure 1.21.

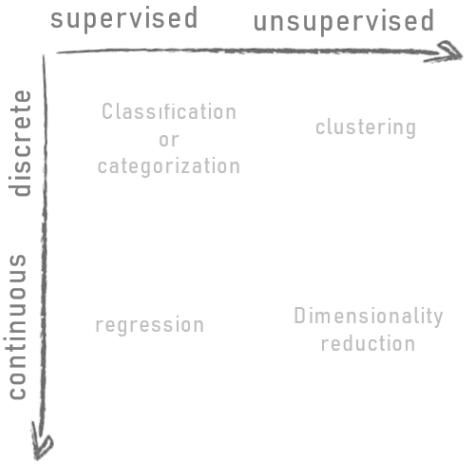

Figure 1.21: Machine Learning approaches depending on their application

Linty et al., 2018 In [7, Linty et al., 2018], the authors use a decision tree classification algorithm to detect whether active scintillation is present in the data as a binary yes or no. The charme of this approach is its computational efficiency, since it does not require any additional channels but only the raw phase and amplitude in form of the I and Q correlator.
Its independence from data processing by using signal components I and Q avoids influence on the performance through filtering methods. On top of that, the trained algorithm can separate between actual scintillation and multipath issues. Also, the algorithm is able to work with data recorded at smaller elevations than 30 $^\circ$.
The optimal performance presented in the paper is the result of a subset of features based merely on the raw signal

$$\mathcal{L} = \{\langle I\rangle, \langle Q\rangle, \langle SI\rangle, \langle I^2\rangle, \langle Q^2\rangle, \langle SI^2\rangle\} \tag{1.1}$$

for 50 [Hz] data in equatorial latitudes. Those input features turned out to be much more accurate than S4, the signal-to-noise ratio and the elevation angle. The authors

obtained a detection accuracy of 98 % for the final training data set with 140,000 points for a decision tree with a 10-fold cross validation (training data : test data 9:1).

McGranaghan et al., 2018 Another very interesting approach is from McGranaghan et al using a support vector machine to also detect and predict scintillations but linked to the geomagnetic, interplanetary and Sun parameters that influence ionospheric scintillations the most. This work is great as it links active phase scintillation to certain geomagnetic background conditions.
The main goal of [9, McGranaghan et al., 2018] is to provide a prediction benchmark with machine learning. Machine Learning is used since the knowledge about scintillations is still very general and the models are tuned for very specific cases only. A climatological prediction exists, but its reliability depends on certain solar activity. This paper is one of the first of its kind for high latitude prediction with ML. There is a lot of data available since the mid of the 1990s, but it is very heterogeneous (different sats, receivers, storage techniques...). The keys to ML according to the authors are: 1) large volume of high-quality data, (2) well-defined task and (3) adequate computational resources - which are all given in this case. So the main task lies in predicting the occurrence of ionospheric phase scintillation at any given location with 1-hr lead time given input solar wind and geospace data. Important questions guiding the work are: What data are most important? How high is the spatial resolution? How can we understand the space weather phenomena to improve ML?
The authors are using the SVM (support vector machine) approach with : observed solar wind, geomagnetic activity, ionospheric behavior and future phase scintillation occurrence. Reasons for the SVM are that it's explainable AI, supervised ML, high accuracy, able to handle high-dimensional data and flexibility with modelling diverse data. Mayor results from this paper are a creation of a new database for ML analyses, a new phase scintillation prediction method for high latitudes and a benchmark for high latitude scintillation prediction.
Datawise the goal of the paper is to identify (1) features which describe scintillation well as predictors and (2) same scintillation data at a later point in time for training and validation. Features are included from the sun, interplanetary space, magnetosphere and upper atmosphere. The phase component of the scintillation data was taken from Novatel CHAIN receivers.
The supervised ML database consists of known input feature and output as predicted label. Data conditioning for CHAIN data involves: elevation mask at 30deg, removing loss-of-lock events (also <200s lock time), creating a super observation which is median of all satellite signals in view into slant-to-vertical projection (height-dependent) which causes rather rough spatial resolution. As soon as geomagnetic data and solar wind comes in, resampling becomes a big issue as well as observation delays. In total 9.6 mio data samples were used, which were most dense between 60 and 70 deg MLAT.
In the next steps the authors (1) create a well-defined and explicit prediction task, then (2) explore the 51 input features, (3) select an algorithm and finally (4) measure performance (based on some robust evaluation metric) assuming zero latency. The task will

be predicting σ_Φ and whether or not scintillation occurs, while using σ_Φ >0.1 [rad] as threshold to keep positive-to-negative class imbalance low (one hot encoding).
Featurization means removing features from the set that are not necessary by using the univariate Fisher ranking score (F score) which assumes independent input features (F score < 1.5 -> feature rejected). In this case phase scintillation and spectral slope have highest F scores for a lead time of 1hr, which might be due to memory effect of large scale structures or due to long duration of geomagnetic activity. Next highest F scores are particle precipitation parameters, especially accelerated electrons, before solar wind features. Multivariable approaches chosen to ensure distinguishing between classes. SVM models with different sets of input features were generated and tested for performance through calculating average and standard deviation TSS. Overfitting occurs after 25 parameters added. All features are normalized to zero mean unit variance in preparation to training the ML model.

Maimaiti et al., 2019 From the different field of substorm analysis another very interesting approach by [8, Maimaiti et al., 2019] built a convolutional neural network that takes 5 geomagnetic input features of substorm onsets that have been prelabeled in order to make a binary substorm onset detection. In cases where the onsets are not always unambiguous in all 5 channels the neural network approach brings great advantages as it can be trained very well for those special cases.
Fortunately, a database with the last 20 years of substorms exists that are prelabeled, which is not the case for us. Since they included geomagnetic and solar storm parameters, the sampling frequency of the data is 1 [Hz] using time series which is much less than in our study.
The reason for exploring this paper from another field is that there are no CNN approaches yet for scintillation classificaiton. We can learn a lot from this paper how to set up a model structure with a very rectangular input matrix that can work with timeseries. On top of that, all the input features have to be normalized and scaled in order to avoid biases towards one parameter. Lastly, distinguishing substorm onsets is not in every case unambiguous and through the years of prelabeled data the really close distributions can still be distinguished through training the algorithm.

And there is a regression approach by **[6, Lamb et al., 2019]** using geomagnetic background data as well for 1h prediction. Since phase scintillation events are rather rare in a dataset over several months, this approach proposes the use of a scarcity and loss function to avoid discarding the actual events due to averaging.

Coming back to our analysis of the different source regions of ionospheric scintillations, we require our algorithm to be able to work with temporal signatures as discrete instances. Instead of a binary detection whether or not scintillation is present we want to introduce a binary classification as to whether the signature is from the auroral oval or the polar cap region. Since we are working with 50 [Hz] data and segments of

30 seconds, we can obtain a fairly huge dataset for many events just based on the signal components. That is why for now we are not considering auxiliary channels like geomagnetic background data with significantly lower sampling rates yet. Thus the goal is to distinguish scintillation signatures for auroral oval vs. polar cap stations merely based on the high-rate data in one frequency.

1.6.1 Hierarchical Clustering and Decision Tree

Since it is not known yet whether the scintillation signatures will be different depending on the source region they were recorded in, our first approach investigates the correlation of the signals for segments with 30 seconds length from either an auroral oval or a polar cap station. The segments with active events were extracted first manually from the data to avoid artefacts or filtering issues that could be extracted incorrectly by automatic thresholding etc. The idea is to build up a decision tree comparing phase correlation, amplitude correlation and phase and amplitude simultaneous correlation. Since we want to compare temporal signatures and not single time instances, we use the hierarchical clustering to compare them with each other and quantify their similarity.

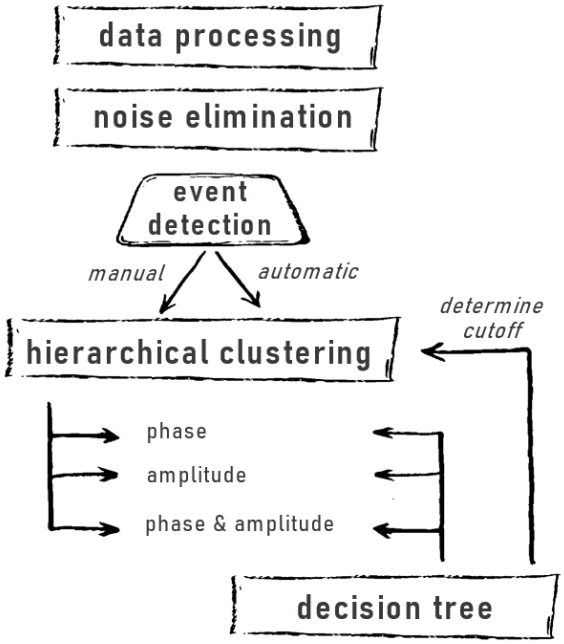

Figure 1.22: Flow Chart for the hierarchical clustering decision tree approach

In a second step we use those correlation measures as training sets for a decision tree model to determine the optimal cutoff defining the groups in the hierarchical cluster. The correlation between the segments is determined by comparing their similarity with the cost function and calculating the Euclidean distance of two segments after. The same number of auroral oval and polar cap segments is used as input. The whole procedure

is shown in Figure 1.22.
An example of manual detection of a very significant phase event from the time series is shown in Figure 1.8 for Tromso in 2012. For Tromso in the auroral oval and Ny-Alesund in the polar cap in March 2012 the hierarchical cluster in Figure 1.10 shows a clear separation for Ny-Alesund samples smaller than 0.2 and Tromso samples considerably larger than 0.2 in simultaneous phase and amplitude. While the amplitude signatures seem to be very similar, the phase correlation shows more distinguishable patterns. Over the course of 3 hours, we see a slight variation in the spread of correlation for simultaneous phase and amplitude in Figure 1.23, where UT 3 is the most active hour trending to wider spreads and peaks in higher correlations for both stations. The higher the contribution of noise is to the signature, the lower the linkage since amplitude and features are similar in the background noise and the less spread of the distribution. Some Tromso noise events can be seen in UT 2 close to 0. Therefore, noise plays a decisive role in the processing of the data and the event detection. First, as much background noise as possible has to be eliminated in order to avoid articial correlation effects. This can be achieved by subtracting quiet PRNs mostly consisting of background noise from active PRNs [2, Deshpande et al., 2012]. The impact of noise elimination through reference PRNs is shown in Figure 1.14 for the power spectral signatures. And secondly, it has to be made sure in the detection process to extract as significant segments as possible without noise intervals to compare only actual signal features, as described in section 1.3.2.

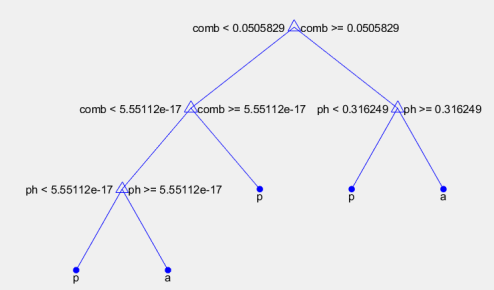

Figure 1.23: Decision Tree for **manually** extracted segments with 3 layers, 4 nodes and 5 leaves

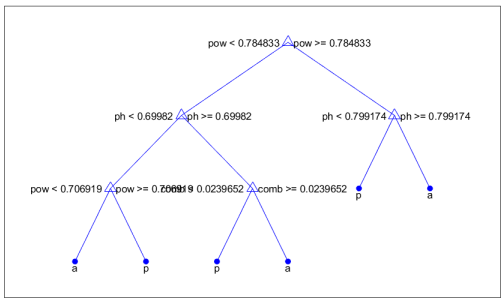

Figure 1.24: Decision Tree for **automatically** extracted segments with 3 layers, 4 nodes and 5 leaves

We are presenting here the decision tree study for year 2015 at PokerFlat and Resolute Bay. For hour 11:00 UTC we extracted 50 training and 25 validation samples manually per station and 150 training vs 75. validation sets of automatically detected samples per station, which are active 30 [s] segments with 50 [Hz] sampling where PokerFlat had to be downsampled. The automatic detection selects the statistically most significant events from the distributions of S4 and sigma Phi amplitude and phase indices. As a result, the confusion matrices for the trained decision tree models with 3 layers phase, amplitude and simultaneous amplitude and phase linkage are shown in Figures 1.25 and 1.26, where the first observation is that the performance of the manual detections

is significantly better than for the automatic detections. This is a consequence of the automatic detection selecting the more significant events that in some cases can also include artefacts due to filtering or gaps or not the loudest events in PRNs where the background is too high and we haven't improved the automatic detection through significance far enough yet. The decision tree model tends to detect the polar cap segments with greater accuracy than the auroral oval segments. From the distributions we know that the incorrectly labeled auroral segments that are actually polar segments have very similar linkages and are therefore hard to distinguish.

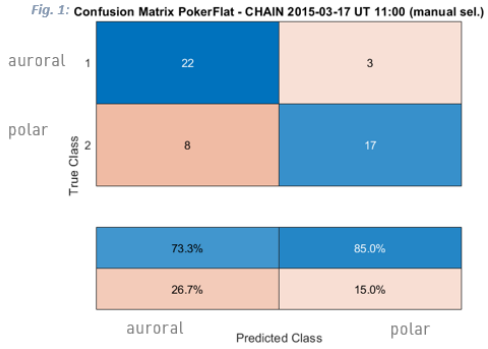

Figure 1.25: Confusion Matrix for **manually** extracted segments PokerFlat vs . CHAIN-resc with Decision Tree

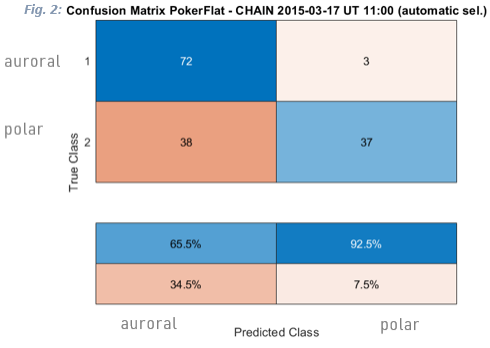

Figure 1.26: Confusion Matrix for **automatically** extracted segments PokerFlat vs . CHAIN-resc with Decision Tree

The True Skill Score was calculated according to [6, Lamb et al., 2019] as

$$TSS = \frac{TP}{TP + FN} - \frac{FP}{FP + TN} \tag{1.2}$$

yielding a TSS of 0.56 for the manual detections and a TSS of 0.45 for the automatically extracted events.
A problem occurred when downsampling PokerFlat data from 100[Hz] to 50[Hz]. Right in the middle of the 30 [s] time series, the data made a complete jump of one. We have improvised and just added or subtracted 1 from the halves of the outcoming timeseries for this analysis. It is not clear yet what causes this issue.

1.6.2 Convolutional Neural Network

Based on the findings of [8, Maimaiti et al., 2019], we decided to try the deep learning approach on our data. Especially in recognizing patterns, convolutional neural networks (CNN) show a deep learning curve. One of the big advantages is that convolutional neural networks work even better if the data is noisy. The background component in the data prevents the model from overfitting.
CNN usually work with image material, but in our case we are building the input matrix with the parameters as discovered most efficient by [7, Linty et al., 2018] in Equation 1.1 since for now we do not have auxiliary channels available. Initially we started off with

segment lengths of 30 [s] corresponding to 1,500 data points, but we also experimented with shorter segments to try to keep the input matrix less long and more rectangular.

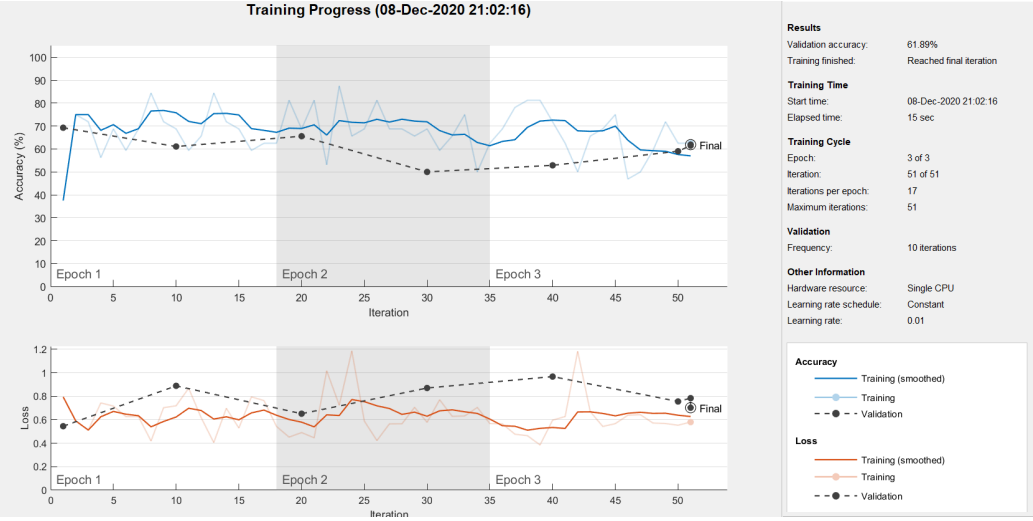

Figure 1.27: Example for first attempts of training a Convolutional Neural Network in Matlab

At first, the signal components in Equation 1.1 have to be found as averages. In a next step, we find the highest value for each of the parameters for both auroral oval and polar cap stations. In order to use them as input features for the CNN, we have to scale the parameters to values between 0 and 1 that will be on the grayscale. Therefore we divide all the segments by the max of all timeseries in polar cap or auroral oval. It is not clear if the transformation into an image is actually successful. When opening the images, they are just black lines, even though they should contain structures.
Perhaps this is the most dominant problem for why the CNN is not converging so far, as shown for an example in Figure 1.27. Depending on the segment length (10-30 [s]) we have between 100 and 350 input files for each station available. They are selected randomly, whereas 2/3 are used for training the model and the remaining 1/3 serve as validation samples.
The CNN used in this analysis consists of two sets of convolution and max pooling layers, followed by another convolution layer and then the final fully connected layer. The details can be found in the appendix 1.10.4. Another issue with our non-converging CNN is probably due to the very long input images, which most likely requires only one stack of convolution and max pooling layer (as in [8, Maimaiti et al., 2019]).

1.7 Conclusion and Future Work

We note that signatures from auroral oval and polar cap stations seem to be partly different especially if extracted manually, even though some overlap between the stations remains. Also, a thorough processing of the data beforehand is extremely important,

especially the noise elimination in order to avoid artificially introduced correlation. The more significant the selected segments, the higher the detection performance. Currently, we are working on the implementation of a convolutional neural network infrastructure that will be purely based on amplitude and phase features without any other geomagnetic background data, to compare to the decision tree analysis. Since the statistical significance of the extracted events plays a crucial role in the classification process, we are steadily improving the automatic event extraction process determined by the hourly intensity distributions. This might also allow for an hourly normalization to ensure stations with generally different amplitudes can be compared. And finally, since the temporal signatures as shown in Figure 1.4 can look very different even though containing similar effects, we are working on a spectral analysis also considering wavelets as shown for gradient drift instability in Figures 1.28 and 1.29.

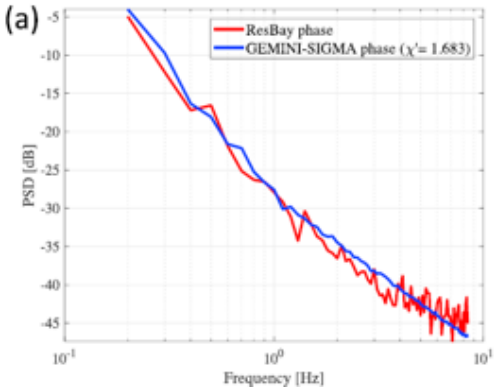

Figure 1.28: spectral phase signature of gradient-drift-instability observed (red) and modeled (blue) [4, Deshpande et al., 2019]

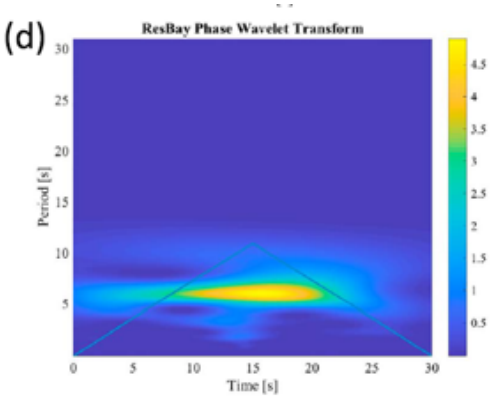

Figure 1.29: wavelet transform of observed time series of gradient-drift-instability [4, Deshpande et al. 2019]

In very rough terms, the next steps will involve:

- more data from different events and stations
- improvement of automatic detection process
- more thorough data cleaning with reference satellites
- decision tree analysis with more input parameters (for example the Linty features), which could be compared in several 2-D clusters
- decision tree analysis using the hierarchical cluster of spectra instead of time-series
- a revised CNN with input images that can be checked and that works on very asymmetrical data
- CNN with wavelet analysis

- a plan on how to incorporate auxiliary channels, as to which ones we need and how to compare them with low sampling rates against high-rate scintillation data

1.8 Project Management

Rough goals of this semester were achieved. We have a first detection algorithm that is unique and makes sense for the requirements of our framework. The hierarchical clustering is developed enough to serve as input for a decision tree model that can be trained fairly well. A CNN has been started but has some technical issues. As to data acquisition, we have unlimited access to CHAIN Novatel stations and PokerFlat data. However, much more bothering of the guardians of the data has to be undertaken to achieve a fuller picture. The recent literature on Machine Learning is familiar, as well as basic scintillation Physics. Starting to dive more into irregularity-specific reads. Writing has made a great leap in creating a round picture thanks to the AGU talk opportunity.

1.9 Acknowledgements

Thanks to Bath, INGV, CHAIN and PokerFlat for the access to the 2012 and 2015 scintillation data.
Many thanks to Luca Spogli for the input on the spectral analysis of the Tromso noise levels.
Many thanks to Bharat Kunduri for the CNN discussion!

1.10 Codes

Excerpt of the four most central scripts serving as a base for the results presented in this work. Since steps have to be reiterated so often, they are still scripts but will be turned into rigid functions as soon as the framework will remain fixed in its application.

1.10.1 Extracting significant phase and amplitude events from arbitrary station PRNs

```
%% extract significant

clear;
clc;
close all;


station = 'CHAIN-resc';
%station = 'PokerFlat';
date2 = '2015_3_17';
```

```
date = '2015-03-17';
hr = 1100;
prn = 7;

filename = strcat('C:\Users\Lenovo\Documents\stars\GNSS\ML\',...
    'Geomagnetic_Storm_2012\plots\statistical\data\',...
    '\', station,'\detected_',...
    station, date2, '-HR', num2str(hr),'_PRN', ...
    num2str(prn),'.mat');

new_loc = strcat('C:\Users\Lenovo\Documents\stars\GNSS\ML\',...
    'Geomagnetic_Storm_2012\plots\statistical\data\',...
    station, date, '-HR', num2str(hr),'_PRN', ...
    num2str(prn), '\');

load(filename);

if station(1) == 'P'
    power = downsample(power, 2);
    phase = downsample(phase, 2);
    Fs = 50;
end

%% extract combined events
if length(com_pos) >1
    for i= 1:length(com_pos)
        new_name = strcat(station, date, '-HR', num2str(hr),'
            _PRN', ...
            num2str(prn), '_comb-event', num2str(i), '.txt');
        path = strcat(new_loc, new_name);
        fileID = fopen(path,'w');
        txtdata=[power(((com_pos(i)-1)*(Fs*seglen)+1):com_pos(i)
            *Fs*seglen),...
            phase(((com_pos(i)-1)*(Fs*seglen)+1):com_pos(i)*Fs*
                seglen)];
        fmt = repmat('%11g ', 1, 2);
        fmt(end:end+1) = '\n';
        fprintf(fileID, fmt, txtdata.');
        fclose(fileID);
    end
elseif length(com_pos) == 1
    new_name = strcat(station, date, '-HR', num2str(hr),'_PRN',
        ...
            num2str(prn), '_comb-event', num2str(1), '.txt');
```

```
51          path = strcat(new_loc, new_name);
52          fileID = fopen(path,'w');
53          txtdata=[power(((com_pos(i)-1)*(Fs*seglen)+1):com_pos(i)
                *Fs*seglen),...
54              phase(((com_pos(i)-1)*(Fs*seglen)+1):com_pos(i)*Fs*
                    seglen)];
55          fmt = repmat('%11g ', 1, 2);
56          fmt(end:end+1) = '\n';
57          fprintf(fileID, fmt, txtdata.');
58          fclose(fileID);
59  end
60
61  %% extract power only events
62  if length(s4_events) >1
63      for j= 1:length(s4_events)
64          if isempty(find(com_pos==s4_events(j))) == 1
65          new_name = strcat(station, date, '-HR', num2str(hr),'
                _PRN', ...
66              num2str(prn), '_s4-event', num2str(j), '.txt');
67          path = strcat(new_loc, new_name);
68          fileID = fopen(path,'w');
69          txtdata=[power(((s4_events(j)-1)*(Fs*seglen)+1):
                s4_events(j)*Fs*seglen),...
70              phase(((s4_events(j)-1)*(Fs*seglen)+1):s4_events(j)*
                    Fs*seglen)];
71          fmt = repmat('%11g ', 1, 2);
72          fmt(end:end+1) = '\n';
73          fprintf(fileID, fmt, txtdata.');
74          fclose(fileID);
75          end
76      end
77  elseif length(s4_events) == 1 && isempty(com_pos==s4_events) ==
        1
78      new_name = strcat(station, date, '-HR', num2str(hr),'_PRN',
            ...
79              num2str(prn), '_s4-event', num2str(1), '.txt');
80          path = strcat(new_loc, new_name);
81          fileID = fopen(path,'w');
82          txtdata=[power(((s4_events(j)-1)*(Fs*seglen)+1):
                s4_events(j)*Fs*seglen),...
83              phase(((s4_events(j)-1)*(Fs*seglen)+1):s4_events(j)*
                    Fs*seglen)];
84          fmt = repmat('%11g ', 1, 2);
85          fmt(end:end+1) = '\n';
```

```
86          fprintf(fileID, fmt, txtdata.');
87          fclose(fileID);
88  end
89
90  %% extract phase only events
91  if length(sph_events) >1
92      for k= 1:length(sph_events)
93          if isempty(find(com_pos==sph_events(k))) == 1
94          new_name = strcat(station, date, '-HR', num2str(hr),'
                _PRN', ...
95              num2str(prn), '_sph-event', num2str(k), '.txt');
96          path = strcat(new_loc, new_name);
97          fileID = fopen(path,'w');
98          txtdata=[power(((sph_events(k)-1)*(Fs*seglen)+1):
                sph_events(k)*Fs*seglen),...
99              phase(((sph_events(k)-1)*(Fs*seglen)+1):sph_events(k
                    )*Fs*seglen)];
100         fmt = repmat('%11g ', 1, 2);
101         fmt(end:end+1) = '\n';
102         fprintf(fileID, fmt, txtdata.');
103         fclose(fileID);
104         end
105     end
106 elseif length(sph_events) == 1 && isempty(com_pos==sph_events)
        == 1
107     new_name = strcat(station, date, '-HR', num2str(hr),'_PRN',
            ...
108             num2str(prn), '_sph-event', num2str(1), '.txt');
109         path = strcat(new_loc, new_name);
110         fileID = fopen(path,'w');
111         txtdata=[power(((sph_events(k)-1)*(Fs*seglen)+1):
                sph_events(k)*Fs*seglen),...
112             phase(((sph_events(k)-1)*(Fs*seglen)+1):sph_events(k
                    )*Fs*seglen)];
113         fmt = repmat('%11g ', 1, 2);
114         fmt(end:end+1) = '\n';
115         fprintf(fileID, fmt, txtdata.');
116         fclose(fileID);
117 end
```

1.10.2 Agglomerative Hierarchical Clustering to determine linkage of event input time series

```
1  %% clustering and linkage
```

```
clear
close all

%% input
num='05';

hour1 = 1100;
station1 = 'PokerFlat';
fs1 = 50; % has been downsampled in extraction script

hour2 = 1100;
station2 = 'CHAIN-resc';
fs2 = 50;


%% read in timeseries
%           --- 1 st station ---
location1 = strcat('D:\GNSS\data\',...
    station1, '_', num2str(hour1), '_old_manual2\');
listing = dir(location1);
dirIndex = [listing.isdir];  %# Find the index for directories
C = {listing(~dirIndex).name}';

n_files = 15;
start_file=(num*n_files)+1;
s4_ts = [];
sig_ts = [];
max_count = length(C);
files = randi([1 max_count], 1, 15);
for k = 1:n_files
    name = C{files(k)};
    prn_start=strfind(name, 'PRN');
    if isnan(str2double(name(prn_start+4)))
        prn=str2double(name(prn_start+3));
    else
        prn=str2double(name(prn_start+3:prn_start+4));
    end
    read_dir=strcat(location1, name);
    data=dlmread(read_dir);
    s4_ts=[s4_ts;data(:,2)'];
    sig_ts=[sig_ts;data(:,3)'];
end

% only if automatic with PokerFlat and CHAIN
```

```
%s4_ts_new = s4_ts(:, 750:end)+1;
%sig_ts_new = sig_ts(:, 1:750)-1;
%s4_ts = [s4_ts(:, 1:749), s4_ts_new];
%sig_ts = [sig_ts_new, sig_ts(:, 751:end)];

%           --- 2 nd station ---
location2= strcat('D:\GNSS\data\',...
    station2, '_', num2str(hour2), '_old_manual2\');
listing = dir(location2);
dirIndex = [listing.isdir]; %# Find the index for directories
C = {listing(~dirIndex).name}';
max_count = length(C);
files = randi([1 max_count], 1, 15);

for k = 1:n_files
    name = C{files(k)};
    prn_start=strfind(name, 'PRN');
    if isnan(str2double(name(prn_start+4)))
        prn=str2double(name(prn_start+3));
    else
        prn=str2double(name(prn_start+3:prn_start+4));
    end
    read_dir=strcat(location2, name);
    data=dlmread(read_dir);
    s4_ts=[s4_ts;data(:,2)'];
    sig_ts=[sig_ts;data(:,3)'];
end

fs = 50;
seg_len=30;
x = linspace(0,seg_len,fs*seg_len);

% stacked plot
figure(1)
subplot(2, 1, 1)
for i = 1:(size(s4_ts, 1)/2)
    h2=plot(x,s4_ts(i+(size(s4_ts, 1)/2), :), 'b:','LineWidth'
        ,1);
    %h2.Color(4) = 0.3;
    h1=plot(x, s4_ts(i,:), 'r:','LineWidth',1);
    %h1.Color(4) = 0.3;
    hold on
    grid on
end
```

```
89  xlabel('t [s]')
90  ylabel('Power')
91  ylim([0 2.5])
92  legend({strcat(station1,' hour',num2str(hour1)),...
93      strcat(station2, ' hour', num2str(hour2))},...
94      'Location','northeast');
95
96  subplot(2,1,2)
97  for l = 1:(size(sig_ts, 1)/2)
98      h4=plot(x,sig_ts(l+(size(sig_ts, 1)/2), :), 'b:', 'LineWidth
            ', 1);
99      h4.Color(4)=0.3;
100     h3=plot(x,sig_ts(l, :), 'r:', 'LineWidth', 1);
101     h3.Color(4) = 0.3;
102     grid on
103     hold on
104 end
105 legend({strcat(station1,' hour',num2str(hour1)),...
106     strcat(station2, ' hour', num2str(hour2))},...
107     'Location','northeast');
108 xlabel('t [s]')
109 ylabel('Phase [rad]')
110 ylim([-3 3])
111
112 hold off
113
114 %% sigma plot
115 % subplot(2, 1, 2)
116 % for i = 1:size(sig_ts, 1)
117 %     plot(x, sig_ts(i,:))
118 %     hold on
119 %     grid on
120 % end
121 % xlabel('t [s]')
122 % ylabel('\sigma_{\Phi}')
123 % ylim([-4 4])
124 % hold off
125
126 fig1_newname = strcat('D:\GNSS\data\plots\timeseries_',...
127     station1, '_hour', num2str(hour1), '-', station2, '_hour',
            ...
128     num2str(hour2), '-run', num,'_new.png');
129 saveas(gcf, fig1_newname);
130
```

```
131
132 %%
133 Z_s4 = linkage(s4_ts, 'single', 'correlation');
134 Z_sig = linkage(sig_ts, 'single', 'correlation');
135 combined = [s4_ts, sig_ts];
136 Z_comb = linkage(combined, 'single', 'correlation');
137
138 figure(2)
139 subplot(3, 1, 1)
140 dendrogram(Z_s4) %, 'ColorThreshold', 0.85)
141 title('Decision tree Power')
142 legendtext= strcat('{', station1, ' # 1-15, ', station2, ' #
      16-30}');
143 legend(legendtext,'Location','northeast');
144 grid on
145 subplot(3, 1, 2)
146 dendrogram(Z_sig) %, 'ColorThreshold', 0.75)
147 title('Decision tree Phase')
148 legend(legendtext,'Location','northeast');
149 grid on
150 subplot(3, 1, 3)
151 dendrogram(Z_comb) %, 'ColorThreshold', 0.2)
152 title('Decision tree Phase & Power combined')
153 legend(legendtext,'Location','northeast');
154 grid on
155
156 fig1_newname = strcat('D:\GNSS\data\plots\decisiontree_',...
157     station1, '_hour', num2str(hour1), '-', station2, '_hour',
          ...
158     num2str(hour2), '-run', num,'.png');
159 saveas(gcf, fig1_newname);
160
161
162
163 count1 = 0;
164 count2 = 0;
165 countr1 = 0;
166 countr2 = 0;
167 rememberlines1 = 0;
168 rememberlines2 = 0;
169 count11 = 0;
170 count22 = 0;
171 countr11 = 0;
172 countr22 = 0;
```

```
rememberlines11 = 0;
rememberlines22 = 0;
count111 = 0;
count222 = 0;
countr111 = 0;
countr222 = 0;
rememberlines111 = 0;
rememberlines222 = 0;
for n =1:size(Z_comb, 1)
    if Z_comb(n, 1)<16
        count1 = count1 + 1;
        link1(count1)= Z_comb(n, 3);
    elseif Z_comb(n, 1)>=16 && Z_comb(n, 1)<31
        count2 = count2 + 1;
        link2(count2) = Z_comb(n, 3);
    end
    if Z_s4(n, 1)<16
        count11 = count11 + 1;
        link11(count11)= Z_s4(n, 3);
    elseif Z_s4(n, 1)>=16 && Z_s4(n, 1)<31
        count22 = count22 + 1;
        link22(count22) = Z_s4(n, 3);
    end
    if Z_sig(n, 1)<16
        count111 = count111 + 1;
        link111(count111)= Z_sig(n, 3);
    elseif Z_sig(n, 1)>=16 && Z_sig(n, 1)<31
        count222 = count222 + 1;
        link222(count222) = Z_sig(n, 3);
    end
    if Z_comb(n, 2)<16
        countr1 = countr1+1;
        rememberlines1(countr1) = n;
    elseif Z_comb(n, 2) >=16 && Z_comb(n, 2)< 31
        countr2 = countr2 + 1;
        rememberlines2(countr2) = n;
    end
    if Z_s4(n, 2)<16
        countr11 = countr11+1;
        rememberlines11(countr11) = n;
    elseif Z_s4(n, 2) >=16 && Z_s4(n, 2)< 31
        countr22 = countr22 + 1;
        rememberlines22(countr22) = n;
    end
```

```
217     if Z_sig(n, 2)<16
218         countr111 = countr111+1;
219         rememberlines111(countr111) = n;
220     elseif Z_sig(n, 2) >=16 && Z_sig(n, 2)< 31
221         countr222 = countr222 + 1;
222         rememberlines222(countr222) = n;
223     end
224     if n == size(Z_comb, 1)
225         l1= size(link1 , 2);
226         l2 = size(link2 ,2);
227         if ~rememberlines1==0
228             for m = 1:length(rememberlines1)
229                 link1(l1+m) = Z_comb(rememberlines1(m), 3);
230             end
231         end
232         if ~rememberlines2 ==0
233             for k = 1:length(rememberlines2)
234                 link2(l2+k) = Z_comb(rememberlines2(k), 3);
235             end
236         end
237     end
238     if n == size(Z_s4, 1)
239         l11= size(link11 , 2);
240         l22 = size(link22 ,2);
241         if ~rememberlines11==0
242             for m = 1:length(rememberlines11)
243                 link11(l11+m) = Z_s4(rememberlines11(m), 3);
244             end
245         end
246         if ~rememberlines22 ==0
247             for k = 1:length(rememberlines22)
248                 link22(l22+k) = Z_s4(rememberlines22(k), 3);
249             end
250         end
251     end
252     if n == size(Z_sig, 1)
253         l111= size(link111 , 2);
254         l222  = size(link222 ,2);
255         if ~rememberlines111==0
256             for m = 1:length(rememberlines111)
257                 link111(l111+m) = Z_sig(rememberlines111(m), 3);
258             end
259         end
260         if ~rememberlines222 ==0
```

```
261                 for k = 1:length(rememberlines222)
262                     link222(l222+k) = Z_sig(rememberlines222(k), 3);
263                 end
264             end
265         end
266 end
267
268 new_name = strcat('D:\GNSS\data\linkage\linkage_',...
269     station1, '_hour', num2str(hour1), '-', station2, '_hour',
        ...
270     num2str(hour2), '-run',num, '.mat');
271 save(new_name, 'link1', 'link2', 'link11', 'link22', 'link111',
    'link222');
```

1.10.3 Decision Tree and confusion matrix based on hierarchical clustering input

```
1  %% decision tree
2
3  clear all
4  close all
5  clc
6
7  station1 = 'PokerFlat';
8  station2 = 'CHAIN-resc';
9  hour = 1100;
10 date = '2015_3_17';
11 location = strcat('C:\Users\Lenovo\Documents\stars\GNSS\ML\',...
12     'Geomagnetic_Storm_2012\plots\statistical\data\',...
13     'dec_tr_training_',...
14     station1,'-', station2, '-', date, '-HR', num2str(hour),'.
           mat');
15 input_data = load(location);
16
17 max = length(table2array(input_data.X))/2;
18
19 ratio = 2/3;
20 max_count = ratio*max;
21 files_a_train = randperm(max, max_count);
22 files_p_train = randperm(max, max_count);
23 x1 = [1:1:max];
24 x2 = [1:1:max];
25 for i = 1:max
26     if isempty(intersect(x1(i), files_a_train)) ~= 1
```

```
27          x1(i) = NaN;
28      end
29      if isempty(intersect(x2(i), files_p_train)) ~= 1
30          x2(i) = NaN;
31      end
32  end
33  x1(isnan(x1)) = [];
34  x2(isnan(x2)) = [];
35  files_a_val = x1;
36  files_p_val = x2;
37  train_a = [];
38  train_p = [];
39  for i = 1:max_count
40      train_a = [train_a; input_data.X(files_a_train(i), :)];
41      train_p = [train_p; input_data.X(files_p_train(i)+max, :)];
42  end
43
44  X = [train_a; train_p];
45  Y = [input_data.Y(1:max_count, 1); input_data.Y(max+1:max+
        max_count, 1)];
46
47  %tc = fitctree(input_data.X,Y,'MaxNumSplits',8,'CrossVal','on');
48  tc = fitctree(X, Y, 'MaxNumSplits',8);%,'CrossVal','on');
49
50  %view(tc.Trained{1},'Mode','graph')
51  val_a = [];
52  val_p = [];
53  for i = 1:(max-max_count)
54      val_a = [val_a; input_data.X(files_a_val(i), :)];
55      val_p = [val_p; input_data.X(files_p_val(i)+max, :)];
56  end
57
58  X_val = [val_a; val_p];
59  Y_val = [input_data.Y(max_count+1: max, 1); ...
60      input_data.Y(max+max_count+1:end, 1)];
61
62  [label,score,node,cnum] = predict(tc, X_val);
63  C = confusionmat(table2cell(Y_val),label);
64  cm = confusionchart(C);
65  cm.Title = strcat('Confusion Matrix PokerFlat - CHAIN 2015-03-17
        '...
66      'UT 11:00 (automatic sel.)');
67  cm.ColumnSummary = 'column-normalized';
68
```

```
tc = fitctree(X, Y, 'MaxNumSplits',5,'CrossVal','on');
view(tc.Trained{1},'Mode','graph')
```

1.10.4 Convolutional Neural Network with signal correlators I and Q

```
%% first cnn

digitDatasetPath = fullfile('C:\Users\Lenovo\Documents\stars\
    GNSS\ML\',...
     'Geomagnetic_Storm_2012\plots\cnn_data\CHAIN-Poker\');
imds = imageDatastore(digitDatasetPath, ...
     'IncludeSubfolders',true,'LabelSource','foldernames');
labelCount = countEachLabel(imds)
img = readimage(imds,2);
size(img)
numTrainFiles = 280;
[imdsTrain,imdsValidation] = splitEachLabel(imds,numTrainFiles,'
    randomize');

%digitDatasetPathT = fullfile('C:\Users\Lenovo\Documents\stars\
    GNSS\ML\',...
%     'Geomagnetic_Storm_2012\CNN\Train');
%imdsTrain = imageDatastore(digitDatasetPathT, ...
%     'IncludeSubfolders',true,'LabelSource','foldernames');
%digitDatasetPathV = fullfile('C:\Users\Lenovo\Documents\stars\
    GNSS\ML\',...
%     'Geomagnetic_Storm_2012\CNN\Validation');
%imdsValidation = imageDatastore(digitDatasetPathV, ...
%     'IncludeSubfolders',true,'LabelSource','foldernames');


layers = [
    imageInputLayer([6 250 1])

    convolution2dLayer(3,8,'Padding','same')
    batchNormalizationLayer
    reluLayer

    maxPooling2dLayer(2,'Stride',2)

    convolution2dLayer(3,16,'Padding','same')
    batchNormalizationLayer
    reluLayer

```

```
36      maxPooling2dLayer(2,'Stride',2)
37
38      convolution2dLayer(3,32,'Padding','same')
39      batchNormalizationLayer
40      reluLayer
41
42      fullyConnectedLayer(2)
43      softmaxLayer
44      classificationLayer];
45
46  options = trainingOptions('sgdm', ...
47      'InitialLearnRate',0.01, ...
48      'MaxEpochs',3, ...
49      'Shuffle','every-epoch', ...
50      'ValidationData',imdsValidation, ...
51      'ValidationFrequency', 10, ...
52      'Verbose',false, ...
53      'Plots','training-progress');
54  options.MiniBatchSize =32;
55  net = trainNetwork(imdsTrain,layers,options);
56
57  YPred = classify(net,imdsValidation);
58  YValidation = imdsValidation.Labels;
59
60  accuracy = sum(YPred == YValidation)/numel(YValidation)
```

Bibliography

[1] I. Cherniak and I. Zakharenkova. “High-latitude ionospheric irregularities: differences between ground- and space-based GPS measurements during the 2015 St. Patrick’s Day storm”. In: *Earth Planet Space* 68(136) (2016), p. 1. DOI: https://doi.org/10.1186/s40623-016-0506-1.

[2] K. B. Deshpande, G. S. Bust, C. R. Clauer, H. Kim, J. E. Macon, T. E. Humphreys, J. A. Bhatti, S. B. Musko, G. Crowley, and A. T. Weatherwax. “Initial GPS scintillation results from CASES receiver at South Pole, Antarctica”. In: *Radio Science* 47(5) (2012). DOI: https://doi.org/10.1029/2012RS005061. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2012RS005061. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2012RS005061.

[3] K. B. Deshpande, G. S. Bust, C. R. Clauer, W. A. Scales, N. A. Frissell, J. M. Ruohoniemi, L. Spogli, C. Mitchell, and A. T. Weatherwax. “Satellite-beacon Ionospheric-scintillation Global Model of the upper Atmosphere (SIGMA) II: Inverse modeling with high-latitude observations to deduce irregularity physics”. In: *Journal of Geophysical Research: Space Physics* 121(9) (2016), pp. 9188–9203. DOI: 10.1002/2016JA022943. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2016JA022943. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2016JA022943.

[4] K. B. Deshpande and M. D. Zettergren. “Satellite-Beacon Ionospheric-Scintillation Global Model of the Upper Atmosphere (SIGMA) III: Scintillation Simulation Using A Physics-Based Plasma Model”. In: *Geophysical Research Letters* 46(9) (2019), pp. 4564–4572. DOI: https://doi.org/10.1029/2019GL082576. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019GL082576. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019GL082576.

[5] Y. Jiao, Y. T. Morton, S. Taylor, and W. Pelgrum. “Characterization of high-latitude ionospheric scintillation of GPS signals”. In: *Radio Science* 48(6) (2013), pp. 698–708. DOI: https://doi.org/10.1002/2013RS005259. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/2013RS005259. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2013RS005259.

[6] K. Lamb, G. Malhotra, A. Vlontzos, E. Wagstaff, A. G. Baydin, A. Bhiwandiwalla, Y. Gal, A. Kalaitzis, A. Reina, and A. Bhatt. “Prediction of GNSS Phase Scintillations: A Machine Learning Approach”. In: *CoRR* abs/1910.01570 (2019). arXiv: 1910.01570. URL: http://arxiv.org/abs/1910.01570.

[7] N. Linty, A. Farasin, A. Favenza, and F. Dovis. “Detection of GNSS Ionospheric Scintillations based on Machine Learning Decision Tree”. In: *IEEE Transactions on Aerospace and Electronic Systems* PP (June 2018), pp. 1–1. DOI: 10.1109/TAES.2018.2850385.

[8] M. Maimaiti, B. Kunduri, J. M. Ruohoniemi, J. B. H. Baker, and L. L. House. “A Deep Learning-Based Approach to Forecast the Onset of Magnetic Substorms”. In: *Space Weather* 17(11) (2019), pp. 1534–1552. DOI: https://doi.org/10.1029/2019SW002251. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019SW002251. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019SW002251.

[9] R. M. McGranaghan, A. J. Mannucci, B. Wilson, C. A. Mattmann, and R. Chadwick. “New Capabilities for Prediction of High-Latitude Ionospheric Scintillation: A Novel Approach With Machine Learning”. In: *Space Weather* 16(11) (2018), pp. 1817–1846. DOI: 10.1029/2018SW002018. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018SW002018. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018SW002018.

[10] P. Prikryl, R. Ghoddousi-Fard, L. Spogli, C. N. Mitchell, G. Li, B. Ning, P. J. Cilliers, V. Sreeja, M. Aquino, M. Terkildsen, P. T. Jayachandran, Y. Jiao, Y. T. Morton, J. M. Ruohoniemi, E. G. Thomas, Y. Zhang, A. T. Weatherwax, L. Alfonsi, G. De Franceschi, and V. Romano. “GPS phase scintillation at high latitudes during geomagnetic storms of 7–17 March 2012 – Part 2: Interhemispheric comparison”. In: *Annales Geophysicae* 33(6) (2015), pp. 657–670. DOI: 10.5194/angeo-33-657-2015. URL: https://angeo.copernicus.org/articles/33/657/2015/.

[11] A. Spicher, K. Deshpande, Y. Jin, K. Oksavik, M. D. Zettergren, L. B. N. Clausen, J. I. Moen, M. R. Hairston, and L. Baddeley. “On the Production of Ionospheric Irregularities Via Kelvin-Helmholtz Instability Associated with Cusp Flow Channels”. In: *Journal of Geophysical Research: Space Physics* 125(6) (2020). e2019JA027734 10.1029/2019JA027734, e2019JA027734. DOI: https://doi.org/10.1029/2019JA027734. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2019JA027734. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019JA027734.

[12] V. Sreenivash, Y. Su, and S. Datta-Barua. “Automated Ionospheric Scattering Layer Hypothesis Generation for Detected and Classified Auroral Global Positioning System Scintillation Events”. In: *Radio Science* 55(1) (2020). e2018RS006779

2018RS006779, e2018RS006779. DOI: https://doi.org/10.1029/2018RS006779. eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/2018RS006779. URL: https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018RS006779.