Anomaly Detection: Wednesday Night Household Global Power Usage

Group 15 Members

Daniel Tavaszi - 301128307

Stefano Macri - 301229307

Khizr Pardhan - 301314376

Darien Flamont - 301272292

Andrei Hristea - 301352475

CMPT 318 - Cybersecurity, Fall 2018

Abstract:

Electrical grid systems need reliable anomaly detection and automated controls to function at high efficiency. Critical anomaly detection systems must isolate vulnerabilities that cause interference to the electrical grid. Our goal is to quantify and train multiple HMMs (Hidden Markov Models) to detect anomalous behaviour in our test data that deviates from the expected behaviour in our training set. We use a semi-supervised approach with one set of training data and five sets of test data, with the chosen parameter of Global Active Power as a basis for “standard household electrical use”. The chosen time period is Wednesday nights, separating test and training data up between seasons giving us a total of 4 time periods. General data exploration is used to calculate the mean and STD (standard deviation) errors, afterwards point and contextual anomaly detection between the expected normal behaviour of the training set and the test data was done using a maximum and minimum bound on the chosen time period (Wednesday summer nights) and training HMMs in R Studio respectively.

Table of Contents:

1. **Introduction**
    a. Problem
    b. Time period
    c. Packages and functions
    d. Methodical approach
        i. Phase 1
        ii. Phase 2 - Approach 1
        iii. Phase 2 - Approach 2
2. **Observations**
    a. Phase 1 - General Data Exploration
        i. Error Calculations
        ii. Regression for summer Wednesday night
    b. Phase 2 - Approach 1 Point Anomalies
        i. Out of Range
        ii. Moving Average
    c. Phase 2 - Approach 2 Contextual Anomalies (HMMs and log-likelihood)
3. **Conclusion**
    a. Key findings
    b. Challenges
    c. Lessons learned

Figures

Fig. 1.1 - Mean and STD errors using MSE calculation between training and test data using Wednesday morning time period and Global Active Power as the parameter.

Fig. 1.2 - Mean and STD errors using MSE calculation between training and test data using Wednesday night time period and Global Active Power as the parameter.

Fig. 1.3 - Correlation between all observations when time was aggregated hourly

Fig. 1.4 - Correlation between observations when time was aggregated daily

Fig. 1.5 - Predicted values against recorded values for Global Intensity

Fig. 1.6 - Training set GAP Max and Min Wednesday nights

Fig. 2.1 - Wednesday summer nights training set rolling average vs Test set #1 values

Fig. 2.2 - Test set #1 Wednesday summer nights point anomaly identification

Fig. 2.3 - Wednesday summer nights training set rolling average vs Test set #2 values

Fig. 2.4 - Test set #2 Wednesday summer nights point anomaly identification

Fig. 2.5 - Wednesday summer nights training set rolling average vs Test set #3 values

Fig. 2.6 - Test set #3 Wednesday summer nights point anomaly identification

Fig. 2.7 - Wednesday summer nights training set rolling average vs Test set #4 values

Fig. 2.8 - Test set #4 Wednesday summer nights point anomaly identification

Fig. 2.9 - Wednesday summer nights training set rolling average vs Test set #5 values

Fig. 2.10 - Test set #5 Wednesday summer nights point anomaly identification

Fig 3.1 - Normalized Log likelihoods of training sets vs training set

Fig 3.2 - Training data BIC Wednesday summer nights per n-states

1. Introduction

a. Problem

Detecting simple and complex anomalies using household electrical data is a crucial step in determining if power consumption is behaving normally. The first problem is exploring our datasets to find differences between the test data and the training data. From this data exploration, mean sum error was computed between each test and training sets of data. Having compared measurements between each test set and the expected normal behaviour of the training set gave insight into standard deviation and average mean errors between the data. Time windows of Wednesday mornings and nights for different seasons were used to collect metrics on standard deviation and mean average. Then, summations of error differences between test and training sets show relative errors between the training and test data.

The two approaches for detecting anomalies between the training and test data were point anomaly detection and complex anomaly detection. Point anomalies are found using an out of range approach that determines a min-max range on the time windows chosen (i.e. Wednesday nights). During this time window, we use a moving average for a fixed window size in minutes (i.e. the average for every 7 minutes). If this moving average on our test sets was out of range of our total maximum and minimum range on our training set, then that window has a point anomaly associated with it.

b. Time Period

We used a time window of Wednesday nights (7 to 11 pm) and Wednesday mornings (7 to 11 am) which was further partitioned into four seasons. This approach gives a fixed time series with the assumption that seasonal variability will change, but that overall household power consumption remains consistent on Wednesday morning

and nights - it would not have as high variability as, for example, a Saturday night would. This approach decreases noise and reduces the unusable outlying data points for data analysis. Wednesday time windows were determined to be the most stable and will be the basis of our exploration going forward.

For Phase 1, all four seasons and Wednesdays (mornings and nights) were used. For Phase 2, however, we used only Wednesday nights - with the summer months as our variable time windows. It was infeasible to train HMMs that find point anomalies for four different seasons on two different time windows, so we only used a singular time window and a singular season.

c. **Packages and functions**

i) GGPlot - plot data; create figures of different average metrics; point anomaly detections for a time period; and graph BIC's of different HMM parameters to determine a best fit for the data.

ii) Depmixs4 - training basis for HMMs using different number of states and certain parametric criteria.

iii) Seasons function - our time periods were found through a function that split our data into different seasons. Season 1, 2, 3, 4 corresponds with spring, summer, fall, and winter respectively. Seasonal partitioning of our training and test data enabled us to focus on a time period, and instead have different windows to search for anomalies based on season.

iv) Zoo - built in R library that calculates our moving average over a window of size 7; helped plot our point anomalies that resided outside of our training set bounds

d. **Methodical Approach**

i. **Phase 1**

General data exploration was done by splitting our training data into four seasonal variables, then filtering the data as required for regression and exploration (using the time periods specified in section **b**). After filtering, the aggregate function in R Studio will then compile mean and standard deviations for each minute in our time series (morning and night). These are ordered by increasing minutes, to ensure our time series match at every minute between the training and test sets of data. Finally, the mean squared error calculation was used to determine the mean average and standard deviation errors for Wednesday mornings and nights, for each season, as shown in *Fig. 1.1* and *Fig. 1.2*.

ii. **Phase 2 - Approach 1**

The minimum and maximum range of the chosen GAP (Global Active Power) feature was found by aggregating all the training data in our time window into a list. This list is searched to find the minimum and maximum values for each season of our training set (see *Fig. 7*); these training max and min are the upper and lower bounds to compare to the test sets and locate point anomalies.

The *zoo* library in R Studio was then used to determine, for a window size of seven, what the local average of the window would be for the whole time period chosen. If any of these windows in our test sets were above or below the bounds determined by the training set, we would consider it a point anomaly and graph it.

iii. **Phase 2 - Approach 2**

Using our partitioned training set for Wednesday nights, we trained an HMM on the summer months with GAP as our chosen feature. After training this HMM for summer months, our lowest BIC was when n-states was 19. After 20 states, we saw a linear increase in BICs and did not expect them to fall again to the previous low. Using this HMM we calculated the log-likelihood of our five different test sets against this HMM and received *Fig 3.1* after normalizing our training data set log-likelihood. We normalized our training data by using the amount of observations present (i.e. how many weeks in our data set time period). We received a value of 39 observations where as our test sets only had 13 observations. We thus just divided the training set by our ratio instead of dividing each test set as well and receiving the same constant log-likelihoods.

2. Observations

a. Phase 1 - Data exploration

i. Error Calculations

*Fig. 1.1 - Mean and STD errors using MSE calculation between training and test data using Wednesday **morning time** period and Global Active Power as the parameter.*

	Mean Error Test 1	Mean Error Test 2	Mean Error Test 3	Mean Error Test 4	Mean Error Test 5	STD Error Test 1	STD Error Test 2	STD Error Test 3	STD Error Test 4	STD Error Test 5
Spring - Season 1	0.287	0.370	0.287	12.943	13.230	0.2558	0.2262	0.2558	5.3846	5.5031
Summer - Season 2	0.128	0.139	0.128	6.666	6.581	0.0952	0.1078	0.0952	3.6175	3.5694
Fall - Season 3	0.385	0.376	0.385	16.507	16.488	0.1782	0.1693	0.1782	5.2776	5.2382
Winter - Season 4	0.176	0.237	0.176	14.279	14.398	0.1462	0.1447	0.1462	5.1911	5.4687

*Fig. 1.2 - Mean and STD errors using MSE calculation between training and test data using Wednesday **night time** period and Global Active Power as the parameter.*

	Mean Error Test 1	Mean Error Test 2	Mean Error Test 3	Mean Error Test 4	Mean Error Test 5	STD Error Test 1	STD Error Test 2	STD Error Test 3	STD Error Test 4	STD Error Test 5
Spring - Season 1	0.210	0.231	0.210	10.238	9.830	0.1093	0.0992	0.1093	3.9950	3.7116
Summer - Season 2	0.246	0.272	0.246	8.413	8.183	0.1125	0.1262	0.1125	4.6450	4.5540
Fall - Season 3	0.139	0.124	0.139	8.733	8.372	0.0972	0.0825	0.0972	4.8271	4.4512
Winter - Season 4	0.185	0.411	0.185	16.468	16.741	0.1265	0.1564	0.1265	6.9111	7.0086

In the general data exploration, based on the error metrics conveyed from *Fig. 1.1* and *Fig. 1.2*, we make two observations about the connection between test sets and the training set provided.

For every single error estimation using MSE calculation, whether it was mean or STD, test sets #1 and #3 returned identical values for Wednesday (nights and mornings), and all four seasons. This observation is important for later anomaly detection methods in **Phase 2** - we should receive the same point and contextual anomalies between the training set and these two identical test sets.

The test sets greatly increase in error as we move from one to the other. Both test sets #4 and #5 have an error average that ranges from four to eight times as bad as test sets #1 through #3. This error increase of 400 to 500% shows that test sets #4 and #5 (for the chosen parameter of Global Active Power) differ significantly in values between the test and training sets. This means we should have higher anomalous activity in both test set #4 and #5, since the error calculations greatly differ from our training set - our expected normal behaviour.
The second observation is crucial going forward. If we can successfully train an HMM to find contextual anomalies in test sets #1 through #3, we will likely be able to find the anomalies in test set #4 and #5 as well, because based on the error analysis, their values will differ on a larger scale.

ii. Regression for summer Wednesday night

When creating a linear regression model we wanted to predict global_active_power in relation to time. We tried creating multiple models, both aggregating global_active_power hourly, daily, weekly, and finally month, but the resultant predictions were not very accurate. We attributed this to how volatile global_active_power was during the day, and realised a regression model

would not be appropriate for exploring the data that way. This led us to exploring the correlation between all the variables and understand how they are linked together.

Fig 1.3 - Correlation between all observations when time was aggregated hourly

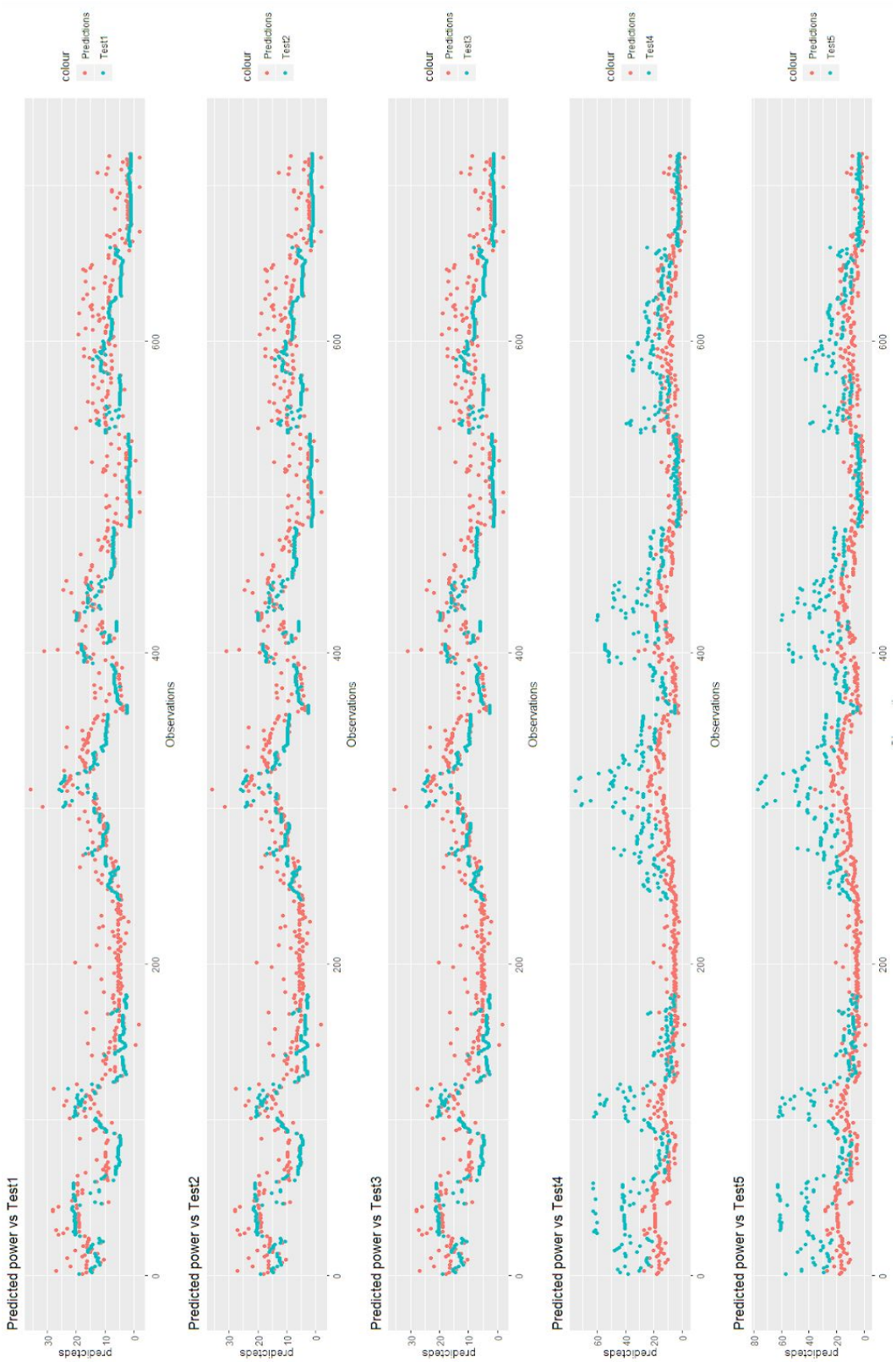
	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
Global_active_power	1.0000000	0.2121330	-0.3073528	0.8092017	0.3691985	0.3467608	0.5165714
Global_reactive_power	0.2121330	1.0000000	-0.1544859	0.3346676	0.3541653	0.2567012	0.1124776
Voltage	-0.3073528	-0.1544859	1.0000000	-0.3979771	-0.2099719	-0.1710845	-0.2991657
Global_intensity	0.8092017	0.3346676	-0.3979771	1.0000000	0.5056849	0.4598049	0.6772855
Sub_metering_1	0.3691985	0.3541653	-0.2099719	0.5056849	1.0000000	0.1237429	0.2042420
Sub_metering_2	0.3467608	0.2567012	-0.1710845	0.4598049	0.1237429	1.0000000	0.1372597
Sub_metering_3	0.5165714	0.1124776	-0.2991657	0.6772855	0.2042420	0.1372597	1.0000000

Fig 1.4 - Correlation between observations when time was aggregated daily

	Global_active_power	Global_reactive_power	Voltage	Global_intensity	Sub_metering_1	Sub_metering_2	Sub_metering_3
Global_active_power	1.00000000	-0.09430221	0.15414334	0.86313850	0.39828051	0.38089763	0.60700967
Global_reactive_power	-0.09430221	1.00000000	-0.06300499	0.05077379	0.33947151	0.17856335	0.04455324
Voltage	0.15414334	-0.06300499	1.00000000	0.08135940	-0.07240053	-0.07435111	0.13252725
Global_intensity	0.86313850	0.05077379	0.08135940	1.00000000	0.54520350	0.48912452	0.73306878
Sub_metering_1	0.39828051	0.33947151	-0.07240053	0.54520350	1.00000000	0.24212045	0.32911920
Sub_metering_2	0.38089763	0.17856335	-0.07435111	0.48912452	0.24212045	1.00000000	0.22734383
Sub_metering_3	0.60700967	0.04455324	0.13252725	0.73306878	0.32911920	0.22734383	1.00000000

Our correlation calculations showed a fairly strong relationship between Global_active_power and Global_intensity. Using this correlation, we create a linear regression model to predict the Global_intensity for summer evenings. Our regression model is able to predict the Global_intensity during the time in which we have no observations. The figure below shows the predictions done by the linear model against the test data. The initial three tests display a strong relationship between the prediction from the regression model and the actual measured values - a strong indication of correct values. Test #4 and #5 values differ greatly from the predicted values of the model which is an indication that anomalies are present in that data.

Fig. 1.5 - Predicted values against recorded values for Global Intensity



b. Phase 2 - Approach 1 Point Anomalies

i. Out of Range: When calculating our min-max range to detect point anomalies, the aggregate function was used. In the point anomalies approach for each test set, we only focus on Wednesday nights for the summer season, though each seasonal maximum and minimum for Wednesday nights was calculated for the training set. Our min and max for training set GAP (Global Active Power), for each season during Wednesday nights, are listed in *Fig. 1.5*.

The highest total minimum and maximum GAP are observed during Wednesday nights, in the winter season, which is expected. The lowest maximum GAP is observed in the fall, and the lowest minimum GAP is observed during the summer season, which also conforms to our expectations.

Fig. 1.6 - Training set GAP Max and Min Wednesday nights

	Max Global Active Power	Min Global Active Power
Spring - Season 1	8.06	0.184
Summer - Season 2	7.436	0.08
Fall - Season 3	7.03	0.188
Winter - Season 4	8.974	0.204

ii. Moving Average: Using the chosen time window of summer Wednesday nights, every test set was tested against our min and max values for the summer season of our training set. Using *zoo*, a built in R Studio library, the training data set rolling averages, with a window size of 7, were plotted against the actual data of each test set. The point anomalies for each test set values, which fall outside the minimum and maximum values of our training set, were plotted in red; the test set

data points that were within our training set bounds were plotted in black. Observing *Fig. 3* through *Fig. 12,* the assumptions from **Phase 1** are strengthened. Both the graphs for test set #1 and #3 have 51 point anomalies in the same windows and the same rolling average graphs. We can confidently say that test set #1 and #3 are comprised of the same data points. In test sets #4 and #5, we observed an increase of point anomalies by about 500%, which is what we expected from our error calculations in **Phase 1.** From the graphs of test set #4 and #5 against the rolling average of our training set, we see that the values of #4 and #5 are much higher than the normal data points in our training set. This leads to a high number of point anomalies and will more than likely lead to a lower log-likelihood in Phase 2 - Approach 2. The two latter data sets are comprised of data points that are nowhere near the expected outcome of normal behaviour represented by our training set. These are likely the two outlier data sets that have been chosen to determine if our metrics are trained well enough to uncover anomalies. Whereas test sets #1-3 are a better representation of normal behaviour with a few anomalies present.

Fig. 2.1 - Wednesday summer nights training set rolling average vs Test set #1 values

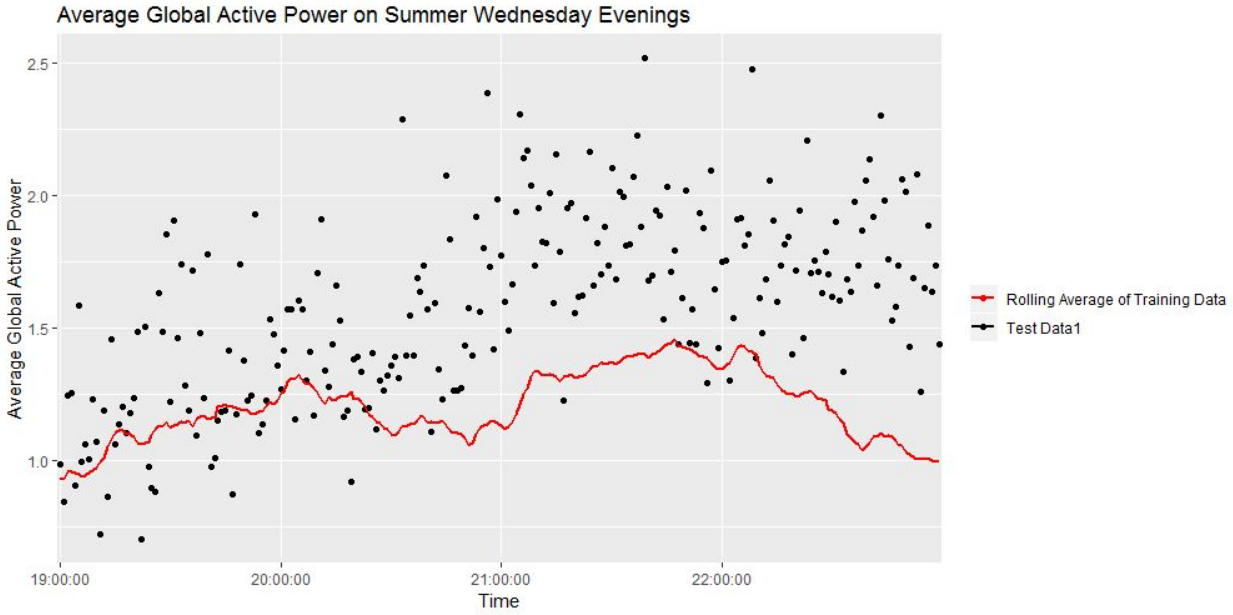


Fig. 2.2 - Test set #1 Wednesday summer nights point anomaly identification

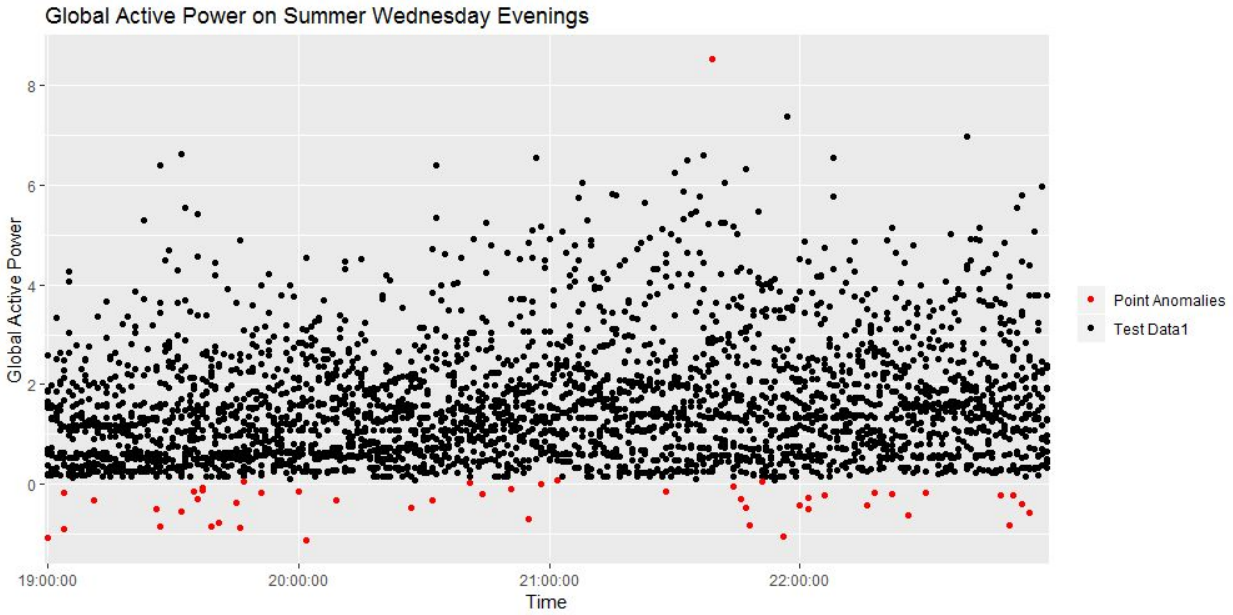


Fig. 2.3 - Wednesday summer nights training set rolling average vs Test set #2 values

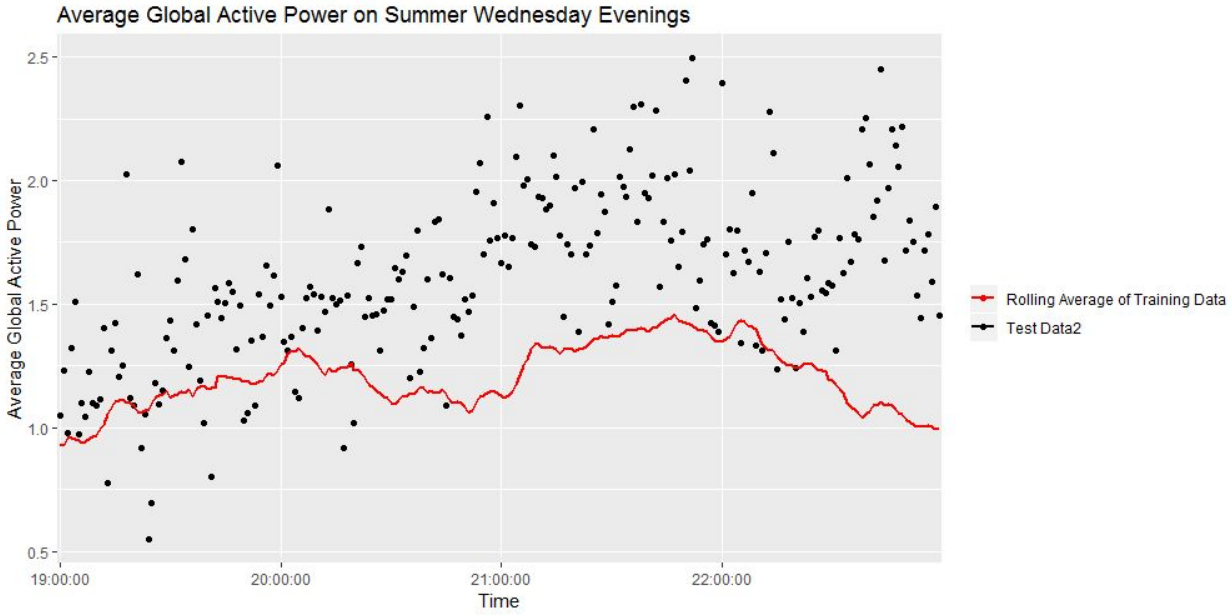


Fig. 2.4 - Test set #2 Wednesday summer nights point anomaly identification

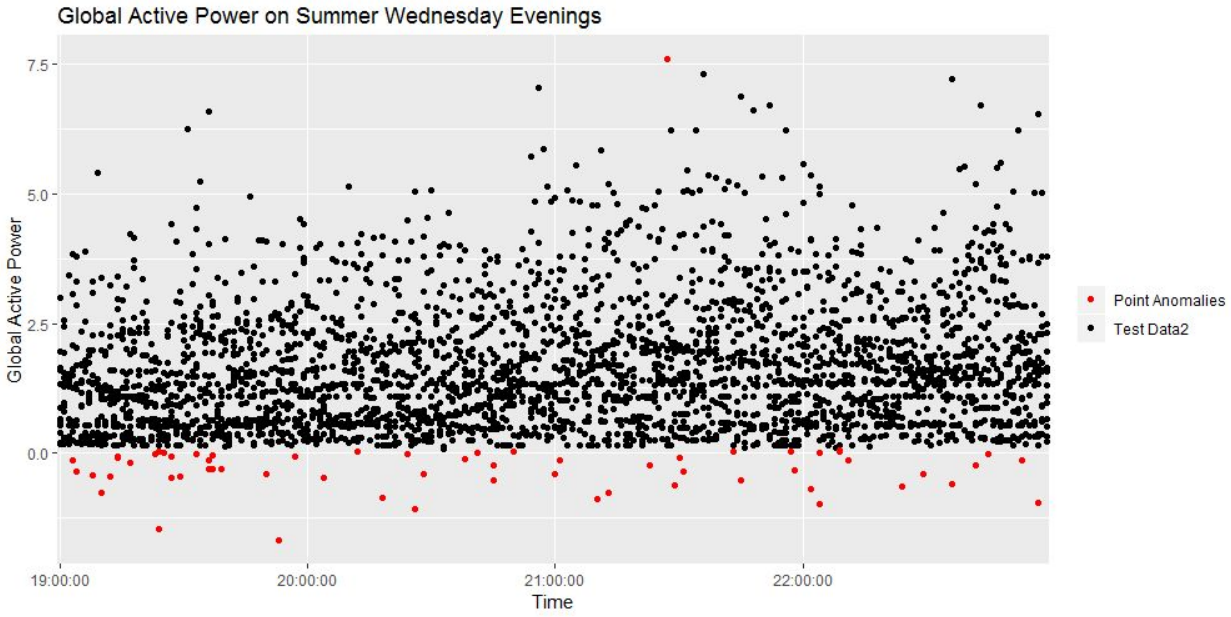


Fig. 2.5 - Wednesday summer nights training set rolling average vs Test set #3 values

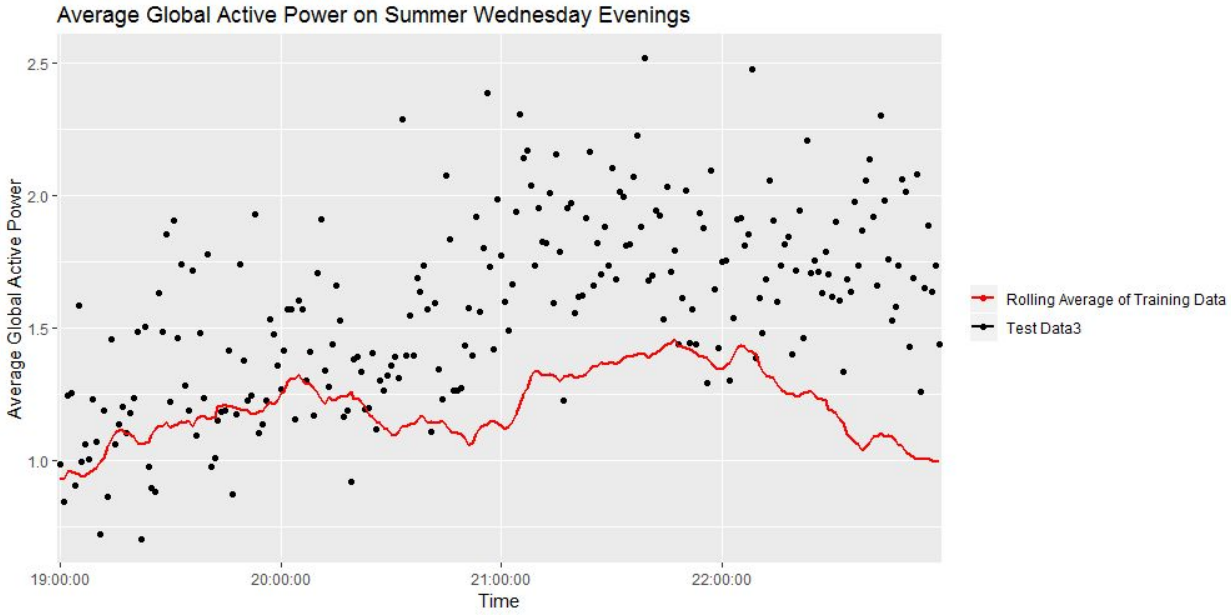


Fig. 2.6 - Test set #3 Wednesday summer nights point anomaly identification

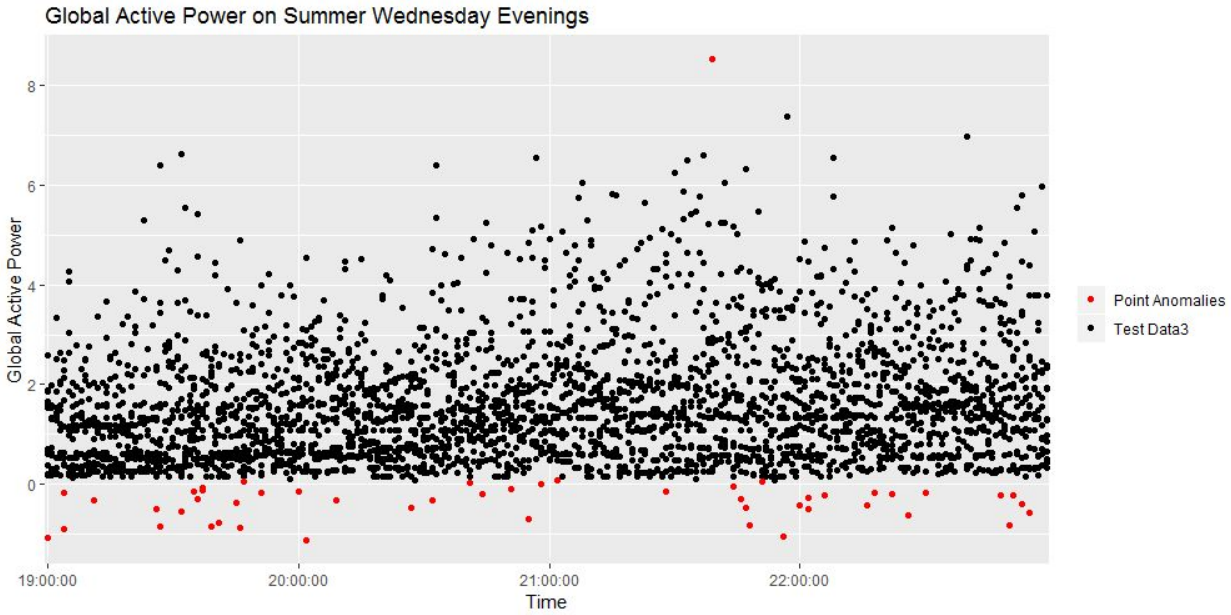


Fig. 2.7 - Wednesday summer nights training set rolling average vs Test set #4 values

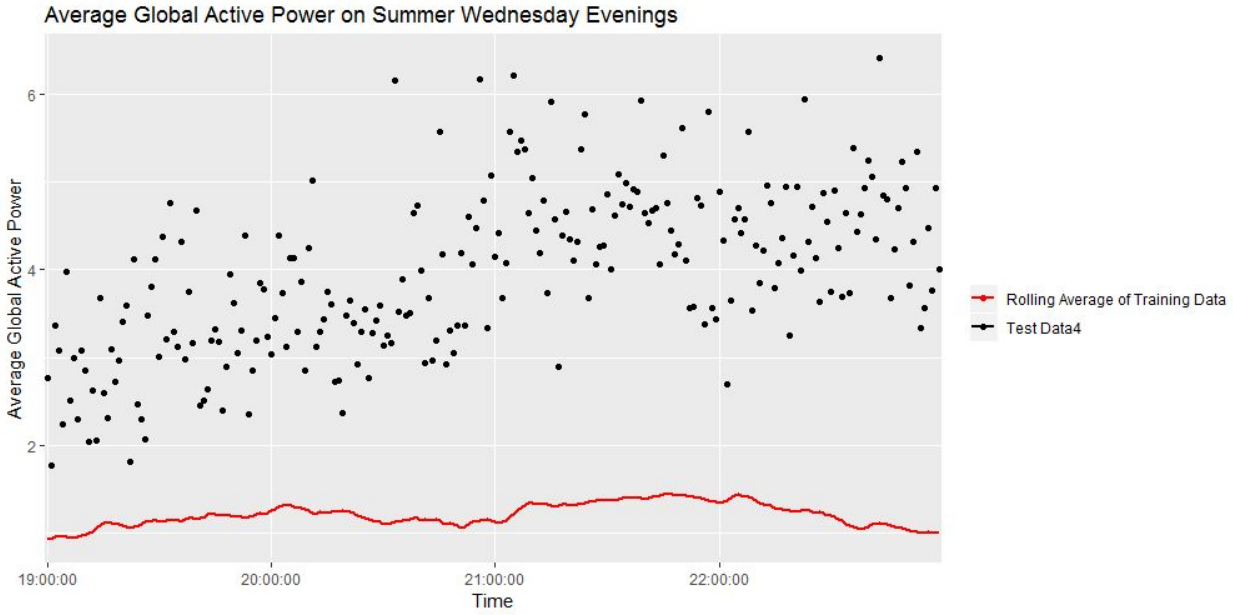


Fig. 2.8 - Test set #4 Wednesday summer nights point anomaly identification

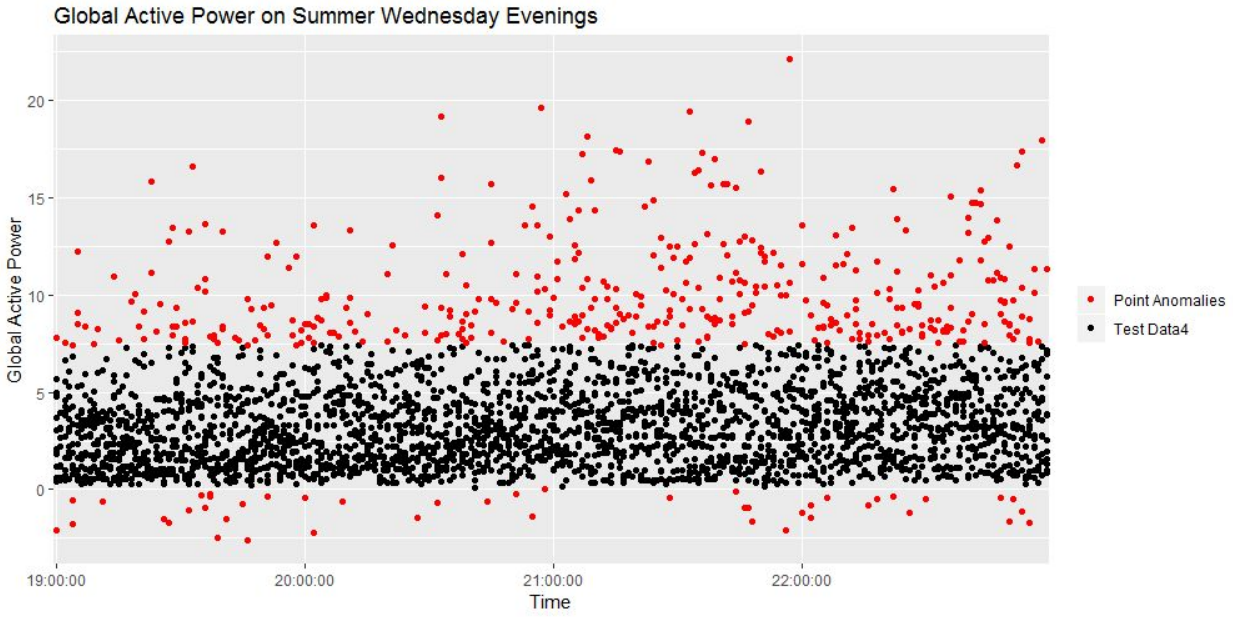


Fig. 2.9 - Wednesday summer nights training set rolling average vs Test set #5 values

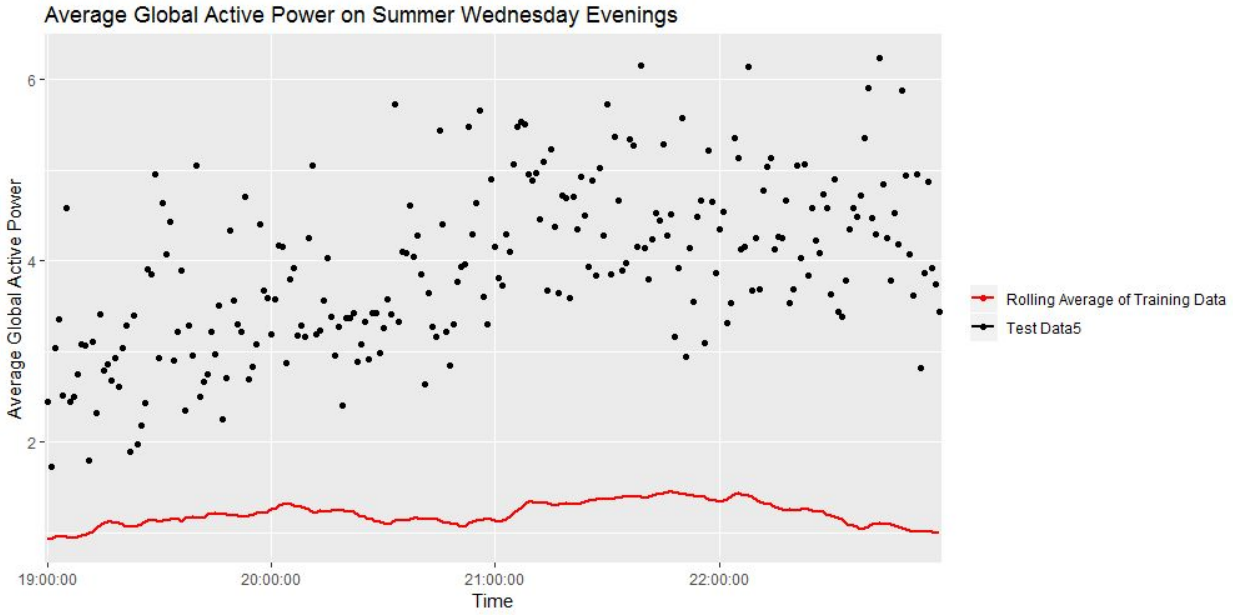
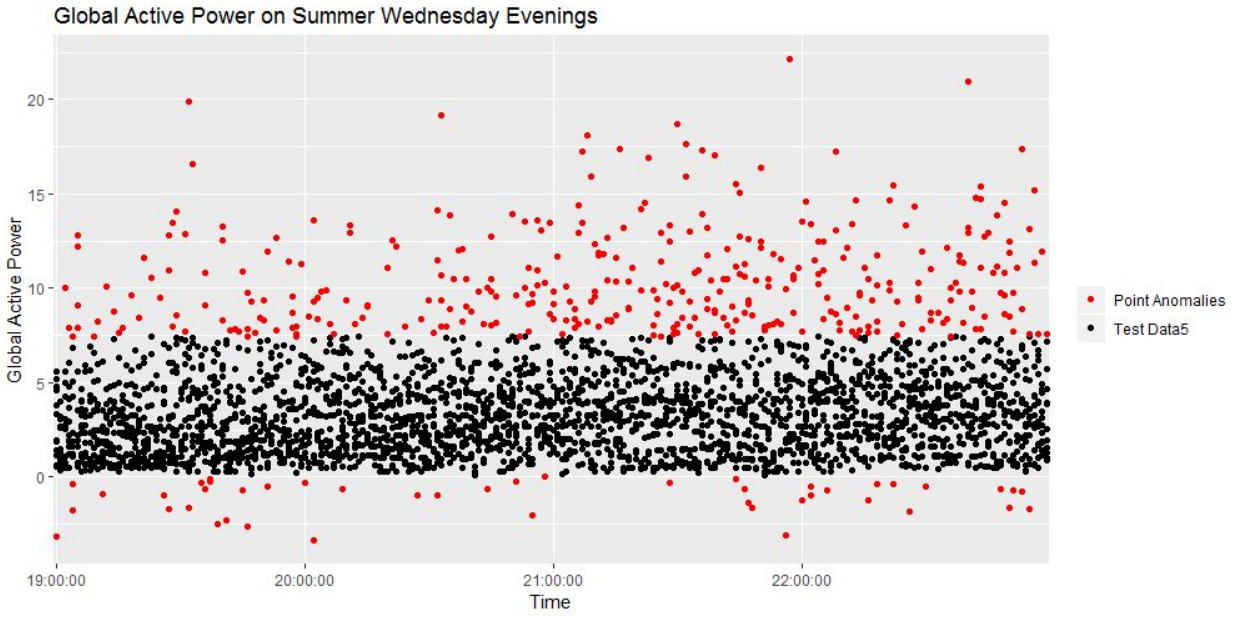


Fig. 2.10 - Test set #5 Wednesday summer nights point anomaly identification

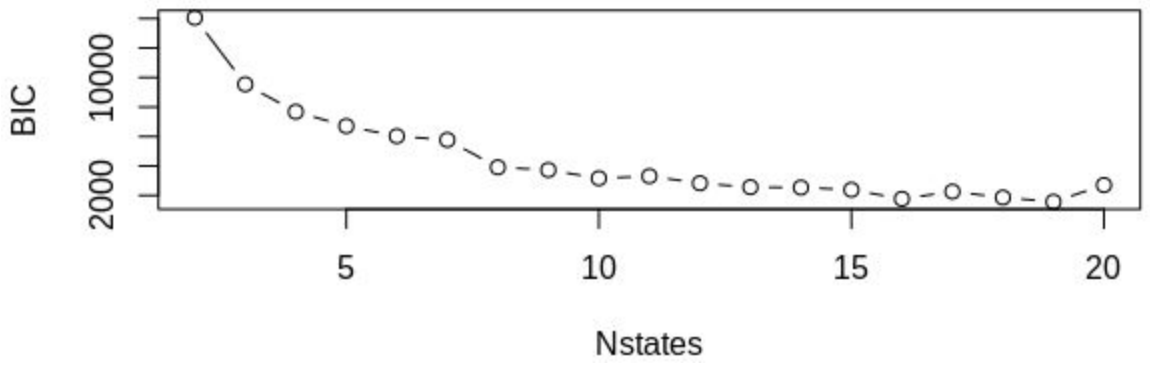


c. Phase 2 - Approach 2 Contextual Anomalies (HMMs and log-likelihood)

Fig 3.1 - Normalized Log likelihoods of training sets vs test set

	Training set	Test set #1	Test set #2	Test set #3	Test set #4	Test set #5
Normalized Log Likelihood	185.6229	-5208.521	-5538.565	-5208.823	-16194.67	-15983.83

Fig 3.2 - Training data BIC Wednesday summer nights per n-states



After creating an HMM with 19 states and using our feature set of Global Active Power, we receive the following log-likelihood values. Our original training set log-likelihood was 556.8686, but since normalization had to be done, we divide the training set log-likelihood by 3, as it has 39 observations for summer Sunday nights. For each of our test sets, we only receive 13 observations for summer Sunday nights if we use the ratio 39/13=3, as opposed to dividing all log-likelihoods by a factor (see *Fig 3.1*). The values presented in *Fig. 3.1* are exactly what we expect when comparing the contextual anomalies of GAP with our training set. Both test sets #1 and #3 have the same log-likelihoods, up to 1 significant digits. Likewise, test sets #4 and #5 have log-likelihoods that are 3 times as low as our other test sets. Given our initial observations in **Phase 1** and **Phase 2 - Approach 1**, we confirm our

previous suspicions about the relationship of our data sets, and have found key consistencies between the training data and the anomalies of the test sets.

3. Conclusion

a. Key Findings

The key findings of this project, through the use of the semi-supervised anomaly detection method, were made apparent in our **Phase 1** error observations, and were later solidified by our anomaly detection. It is certain that test sets #1 and #3 are the same set of data values, seeing as we received the same values in errors, point anomalies, and contextual anomalies for our chosen time window of Wednesday summer nights. Test sets #4 and #5 differ greatly from the expected normal behaviour of our training set. Both test set #4 and #5 have a high number of point anomalies and large error ratios; their log likelihoods are the lowest for both test sets when compared to our training set.

i. **Phase 1** - The feature highlighted is GAP (Global Active Power). The wednesday mornings (7 to 11 am) of the Fall season contained significantly more errors than other seasons. A surprising event found is that Wednesday winter nights have significantly more error between our test data #4 and #5, as opposed to the first three data sets. The linear regression model that predicts global intensity from GAP supports this finding, as there are many values that have large differences from their predicted values. From this information, we deduce that Wednesday winter nights in test data #4 and #5 have significantly more abnormalities than any other combinations of season and test data.

ii. **Phase 2 Approach 1** - We established min and max values for GAP on Wednesday nights, and rolling averages, with a window size of 7. Using these methods, we confidently say that test set #1 and #3 are comprised of the same data points. A key example of the potential difference between training and test data is *Fig. 2.7 - Wednesday summer nights training set rolling average vs Test set #4 values.*

iii. **Phase 2 Approach 2** - We discover that the Normalized Log Likelihood of Test set #4 and #5 are more than three times less than that of the other test sets. Despite the power of HMMs, we could not accurately predict anomalous behaviors in test set #4 and #5. As expected, both test sets #1 and #3 have very similar log-likelihoods.

b. Challenges

Early on, we had to make decisions that would set the direction for rest of the project. We had to decide which identified time windows and feature we wanted to explore and to base our HMMs on. We decided on the feature Global Active Power, and the time window of Wednesday nights with seasonal variability, so that we would not have to repeat the training anymore than necessary, because the training requires a lot of time.

R being single threaded was an inconvenience. We had to ensure our code was bug free before training our HMMs. We had a bound of 20 states for training our HMMs, because after ~10 states, a trend occurred that caused the HMMs to increase with higher BICs. These state bounds assume that there is no perfect state past 20 that will receive the lowest and best BIC; we were able to follow this linear trend.

Working with GGPLOT2 was a major challenge - particularly in making graphs

with two variables that span the Y axis. Converting between different data types was also an inconvenience, but well worth the trouble.

The zoo library was exceptionally helpful in finding common statistical values for any given time frame. We established bounds, based on the training data, to use for detection of point anomalies. Since test sets #4 and #5 differed greatly from the expected normal behaviour of our training set, establishing bounds and using a moving average yielded a large number of suspected anomalies.

The observations provided do not have a linear pattern when changing over time. We tried creating multiple linear regression models to predict how variables changed over time. In the end, time did not create a linear regression model, so we instead used the strong correlation between Global_active_power and Global_intensity to create a model that is able to predict global_intensity.

c. Lessons Learned

We had to be certain, due to long training times, that the model we are training was bug free. We did not waste time pursuing increased speed with use of the package *parallel.* Linear models do not work well with real data that have large fluctuations in values. While using bounds and moving average is easy, it is not effective. When there are many variables for similar data, it is best to be very descriptive, even at the cost of having long variable names. When working with a weakly typed language such as R, it is helpful to comment the datatype of key non-primitive variables. Doing so allows multiple programmers to work on a single task with little explanation for their code. Furthermore, black lines should be utilized to differentiate between subtasks, as it compensates for the lack of modularity and encapsulation, caused by the absence of an object-oriented programming paradigm.

References:

Pastell, Matti. (2011, February 11). Calculating moving average. Retrieved from https://stackoverflow.com/questions/743812/calculating-moving-average/4862334#4862334

Svetunkov, Ivan. (2018, August 25). sma() - Simple Moving Average. Retrieved from https://cran.r-project.org/web/packages/smooth/vignettes/sma.html

O'Brien, Josh. (2012, February 29th). Find which season a particular date belongs to. Retrieved from https://stackoverflow.com/questions/9500114/find-which-season-a-particular-date-belongs-to

Zeileis, Achim. (2018, September 19th). zoo: S3 Infrastructure for Regular and Irregular Time Series. Retrieved from https://cran.r-project.org/web/packages/zoo/index.html

Ushey, Kevin. (2018, June 5th). Package 'RcppRoll'. Retrieved from https://cran.r-project.org/web/packages/RcppRoll/RcppRoll.pdf

Wickham, Hadley. (2016, June 8th). Package 'plyr'. Retrieved from https://cran.r-project.org/web/packages/plyr/plyr.pdf

Wickham, Hadley. (2018, October 25th). ggplot2: Create Elegant Data Visualisations Using the Grammar of Graphics. Retrieved from https://cran.r-project.org/web/packages/ggplot2/index.html

Ryan, Jeffrey. (2018, November 5th). xts: eXtensible Time Series. Retrieved from https://cran.r-project.org/web/packages/xts/index.html