Final Project Report
CMPT 318: Cybersecurity Analytics
Uwe Glässer
Spring 2020

Mackenzie Craig - 301371202
Queen Dominique Dela Cruz - 301273336
Andrew Do - 301265100
Praneeth Ellanti - 301381164
Vafa Dehghan Saei - 301379021

Table of Contents

Table of Figures

Table of Tables

Abstract

Due to the increasing threat landscape in cybersecurity, it is becoming increasingly important to improve the security of critical infrastructure and services to prevent damages and to maintain safe operation. By using a large daily power consumption time series dataset, we trained univariate and multivariate Hidden Markov Models (HMM) to detect anomalies. We compared anomaly detection of HMM with the Moving Average anomaly detection technique in order to verify the efficacy of the univariate and multivariate HMMs. The results, validated by using the moving average anomaly detection technique, show that HMMs trained on a specific time frame could be used to detect anomalies for future days on the same time frame.

Project scope

Power Infrastructure is a vital part of a well functioning society. Although we don't usually think about the importance of a safe and reliable electric grid, it directly affects all aspects of our lives including: how the food we eat is produced, how we get to school or work, the goods & services we can consume, how we communicate with others and many more important activities which far exceed the word count of this document. Despite our reliance on a safe & reliable electrical grid, there has been a lack of infrastructure investment even as society becomes more dependent on digital systems. In most cities and towns the power systems used today are nearly identical to the ones used decades ago. This is a problem because cyber crimes have begun to target public institutions such as hospitals, police departments and government institutions. By targeting the electrical infrastructure an intruder is effectively disabling all of these services in one attack, with a chance of causing long lasting damage if companies are inadequately prepared to respond to cyber attacks. In order to fight back, companies can

analyze data monitored by SCADA systems. Abstract concepts about “normal” usage can be derived from historical data & be used to sense active intrusions allowing for precious response time. The purpose of this project is to find out which data measurements are relevant to intrusion detection using PCA analysis. Using HMMs, we created a variety of univariate & multivariate models. By comparing the complexity (BIC) & faithfulness (Log. Like) of the models we were able to select the best models to move onto the testing phase. In order to test the effectiveness of our models, new data with injected “anomalous” data points were fed into the trained HMMs and a rolling average analysis was used to detect anomalies as well as their level of severity. This analysis is important since we want to build a system that detects anomalies but does not have too many false-positives are failures of detection. When an anomalous intrusion is detected, operators can react accordingly to scale up/down power generation & avoid overloading of electrical important electrical components.

Methodology

Feature Selection

Feature selection is the process of choosing the best subset of response variables from a large multivariate data set which is used to train a predictive model. Its ultimate goal is to reduce overfitting, improve accuracy, and reduce training time[1]. This process involves looking at the relationship or trends between each response variable to distinguish which subsets of variables would have redundant information. Reducing the number of response variables reduces unnecessary complexity of the model but increases the utility of relevant training information. This process seeks to extract useful information from the input without modifying it.

One of the important benefits of feature selection is to reduce overfitting, this phenomena occurs when models are fit well to training input and begin to pick up on noise present in real-world data. By selecting fewer response variables, less noise is included in the training input and it is less likely that the trained model will be overfitted. Moreover, selecting which features can eliminate information that is known to not have a significant relationship to the problem being solved. For example, using an amount of visible stars in the evening will not yield an accurate predictive model for the weather for the next morning. Using domain knowledge and relationships between each response variable can improve a predictive model's accuracy and reduce its training time.

This process is an integral part of training a predictive model because when done correctly, it offers significant improvements in the overall performance of the model. Two methods are used as the basis understanding and choosing the optimal subset of response variables for this project: correlation and PCA.

Correlation

Correlations graphs help us identify between important & unimportant response variables which do not contribute to the explanation of an observed data set. Highly correlated features will have redundant information, since features that change in the same pattern will yield similar information[2]. For example when training multivariate HMMs, we select a main feature along with secondary features that do not correlate with one another but share a high correlation with the main feature. This technique minimizes correlation among response variables while still keeping most of the information from the original training data.

Principal Component Analysis

PCA applies linear orthogonal transformation on highly dimensional data, and outputs uncorrelated (i.e. independent) principal components. These components are ranked by variance wherein the first component has the most variance of the original data, and the second component has the second highest variance and so on[2]. The goal in this section is to find response variables that contribute the most variance to PC1 and PC2, while still minimizing correlations. The 'prcomp' function, from the stats package in R, is used to perform PCA and biplots were used to interpret the results.

Training and Testing

A Hidden Markov Model is a model where the states are hidden from the observer and can be indirectly observed through a visible output of the system being modeled. This model is characterized by the number of states, number of distinct observable outputs, and probability distributions for the state transition and observation outputs[3]. To obtain an HMM model from the training data, we used the depmixS4 package in R and tried different numbers of hidden states. To test these models, a testing set partitioned from the original data set is fed into the model, and its log-likelihood and BIC values are compared with the trained model. This process will find the optimal model that fits the input data set, which can be used to detect anomalies.

Anomaly Detection

Anomaly detection is the process of finding abnormal behaviour. This is done by finding a model of normal behaviour, which is compared to the latest incoming data of the system to

determine any patterns that differ from normal. For this project, two different methods are used to determine anomalous behaviour: moving averages and HMM models.

One method of anomaly detection that we used was Moving Average (also known as Rolling Average). The concept of a moving average is to iterate over a time series considering n points at a time. For each point, n/2 points are considered on each side of the point. These points are averaged to plot the moving average line. This average line is used to represent normal behaviour of the data.

To detect anomalies using the Moving Average, we defined 3 thresholds that we used to flag anomalies depending on their severity. They are defined as follows:

Minor Anomaly (Yellow): 1 standard deviation away from the Moving Average

Major Anomaly (Orange): 2 standard deviation away from the Moving Average

Extreme Anomaly (Red): 4 standard deviation away from the Moving Average

In our charts shown in Table 6, we plot the original time series data using a grey line, and then the Moving Average line in blue. It is important to note that the moving average line begins and ends 5 samples after and before the original time series line. This occurs because it can only be plotted when there are 5 samples available to either side of its x axis value (our Moving Averages used 10 points total). We use the colors listed above to place dots on any anomalies found and to distinguish their severity. Points with no dot fall under the “Minor Anomaly” threshold and are not considered to be anomalies at all.

We calculated a moving average on the response variable Global_active_power. Table 5 shows the results of our moving average on each of the 5 test data sets, for the days Saturday (2009-12-07) and Monday (2009-12-09) from 07:00 AM until 09:00 AM.

Using HMM models, we can easily determine if a set of observations contains anomalous behaviour. The trained model's log-likelihood, which shows how well the model matches that sequence of observation, can be compared to a new set of observations's log-likelihood. If the new observations produced lower log-likelihood then these observations do not match the HMM's computed normal behaviour.

Characteristics of the solution

Data Exploration

To choose an appropriate time window, we visualized the averages of response variables in each minute of the same year. When plotted, the response variables show clear and consistent patterns in the morning, specifically between 7:00 AM to 9:00AM.

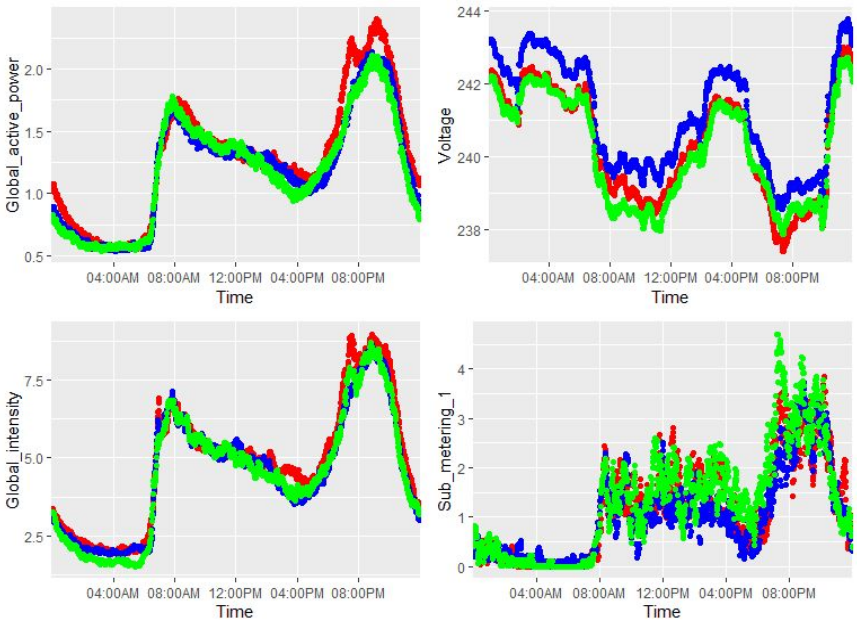


Figure 1. Averages of Response Variables

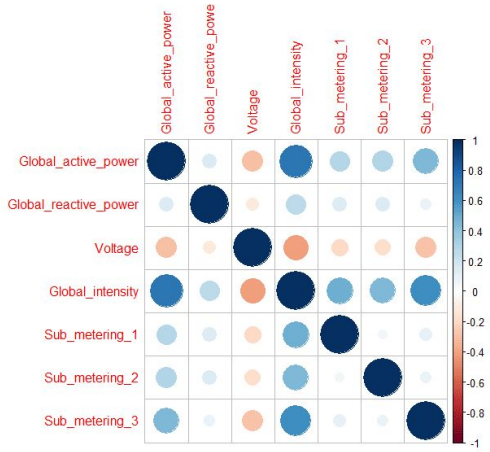


Figure 2. Correlation Heatmap

Figure 2 shows the correlation between each pair of response variables. Global active power and Global intensity have high positive correlation, while Voltage has high negative correlation with Global intensity and has low negative correlations with all other variables. This shows that these three could potentially be part of the optimal response subset for training.

PCA

After performing PCA on all the response variables for the time window (Monday and Saturday 7 am to 9 am) we chose, we were left with 7 principal components. The first principal component (PC1) represents the response variables that contribute the most to the variance observed in the dataset. We observe that PC1 accounts for 34.57% of the variance in the dataset for Monday, and 36.31% for Saturdays. By visualizing each response variable's contributions towards the principal components, we were able to derive the following figures.

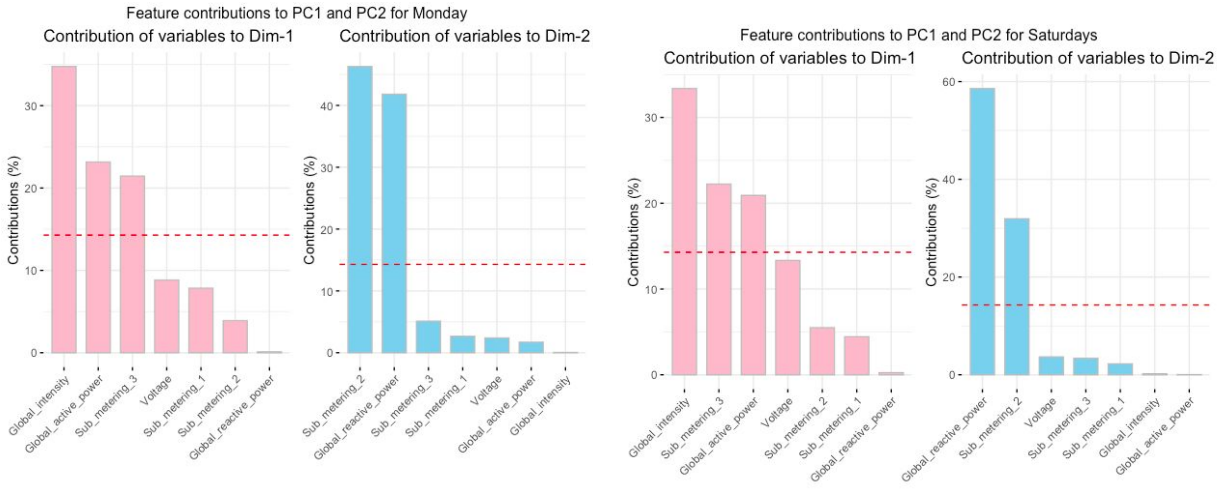


Figure 3.

As seen in the contribution plots to PC1 and PC2 for Mondays and Saturdays, Global intensity, and Global active power response variables contribute the most to PC1 for both days. Although Sub-metering 3 contributes to PC1 for both Monday and Saturday, it was not used for training our HMM models as it individually represents incomplete data, and domain expertise was needed inorder to incorporate Sub-metering data into our models. By using this information, we were able to rationalize that Global intensity and Global active power were both useful response variables that could be used to train the hidden markov models.

Training HMMs

The original data set is partitioned to a training set (from 2006-12-16 to 2008-12-31) and testing set (from 2009-01-01 to 2009-12-01) for both weekday and weekend data. Initially, the models we trained only had nstates = 5, 10, 15, 20 to see the trend of the log-likelihood and BIC values as we increase the number of states. These results are compared to the corresponding testing set results to determine whether the models are overfitting or underfitting at what number

of hidden states. The models with promising log-likelihoods and BIC values are further explored by training those features with different numbers of states starting at 5.

For univariate models, we choose Global active power, Global Intensity, and Voltage as features to train the models with because they have the most influence in PC1 for both weekdays and weekends. The models that had the best log-likelihood and BIC values are models using Global active powers as the response variables. Moreover, Table 1-4 contains unnormalized log-likelihoods and BIC values for univariate models, but the normalized values are shown in Fig 5-6. Models with Global active power with N = 11 have similar normalized values between training and testing sets. For models with states higher than 11, the training log-likelihood was much lower than the test log-likelihood which was indicative of the training model being overfitted to the training data as seen in Fig 5 and 6.

Table 1. Univariate model training (weekday) Monday 7am to 9 am

	Voltage	**Global Intensity**	**Global Active Power**
N = 5	'log Lik.' -18504.07 (df=34) AIC: 37076.13 BIC: 37329.75	'log Lik.' -21455.42 (df=34) AIC: 42978.84 BIC: 43232.46	'log Lik.' -5935.088 (df=34) AIC: 11938.18 BIC: 12191.79
N = 6	'log Lik.' -17351.52 (df=47) AIC: 34797.03 BIC: 35147.62	'log Lik.' -20851.81 (df=47) AIC: 41797.63 BIC: 42148.21	'log Lik.' -5684.702 (df=47) AIC: 11463.4 BIC: 11813.99
N = 7	'log Lik.' -16198.66 (df=62) AIC: 32521.31 BIC: 32983.79	'log Lik.' -17890.26 (df=62) AIC: 35904.53 BIC: 36367	'log Lik.' -5180.581 (df=62) AIC: 10485.16 BIC: 10947.63
N = 8	'log Lik.' -15452.99 (df=79) AIC: 31063.97 BIC: 31653.25	'log Lik.' -17241.88 (df=79) AIC: 34641.75 BIC: 35231.03	'log Lik.' -4913.908 (df=79) AIC: 9985.815 BIC: 10575.09
N = 9	'log Lik.' -14956.65 (df=98) AIC: 30109.3 BIC: 30840.3	'log Lik.' -15356.13 (df=98) AIC: 30908.26 BIC: 31639.26	'log Lik.' -4851.096 (df=98) AIC: 9898.192 BIC: 10629.2
N = 10	'log Lik.' -14593.88 (df=119) AIC: 29425.76 BIC: 30313.41	"log Lik.' -14733.66 (df=119) AIC: 29705.32 BIC: 30592.96	'log Lik.' -4597.881 (df=119) AIC: 9433.762 BIC: 10321.41
N = 11	Failed to converge	Failed to converge	'log Lik.' -3581.129 (df=142) AIC: 7446.258

			BIC: 8505.468
N = 12	Failed to converge	Failed to converge	'log Lik.' -3112.648 (df=167) AIC: 6559.297 BIC: 7804.988
N = 13	Failed to converge	Failed to converge	'log Lik.' -3353.314 (df=194) AIC: 7094.627 BIC: 8541.718
N = 14	Failed to converge	Failed to converge	'log Lik.' -2797.716 (df=223) AIC: 6041.432 BIC: 7704.84
N = 15	Failed to converge	Failed to converge	'log Lik.' -2668.612 (df=254) AIC: 5845.225 BIC: 7739.869
N = 16	Failed to converge	Failed to converge	'log Lik.' -2341.003 (df=287) AIC: 5256.006 BIC: 7396.805
N = 17	Failed to converge	Failed to converge	'log Lik.' -2123.032 (df=322) AIC: 4890.063 BIC: 7291.935
N = 18	Failed to converge	Failed to converge	Failed to converge

Table 2. Univariate Model Testing (weekday) Monday 7am to 9 am

	Voltage	**Global Intensity**	**Global Active Power**
N = 5	LogLik: -9687.578 BIC: 19669.83	LogLik: -9700.319 BIC: 19695.31	LogLik: -2399.115 BIC: 5092.908
N = 6	LogLik: -9141.97 BIC: 18691.29	LogLik: -9576.872 BIC: 19561.09	LogLike: -2334.048 BIC: 5075.444
N = 7	LogLik: -8448.432 BIC: 17434.22	LogLik: -8215.698 BIC: 16968.75	LogLik: -2282.076 BIC: 5101.505
N = 8	LogLik: -8149.867 BIC: 16984.43	LogLik: -8127.428 BIC: 16939.55	LogLik: -2216.584 BIC: 5117.86
N = 9	LogLik: -7926.186 BIC: 16701.74	LogLik: -6715.511 BIC: 14280.39	LogLik: -2179.311 BIC: 5207.987

N = 10	LogLik: -7713.655 BIC: 16458.68	LogLik: -6484.997 BIC: 14001.37	LogLik: -2158.21 BIC: 5347.791
N = 11	Failed to converge	Failed to converge	LogLik: -1673.78 BIC: 4578.272
N = 12	Failed to converge	Failed to converge	LogLik: -1566.314 BIC: 4580.015
N = 13	Failed to converge	Failed to converge	LogLik: -1612.074 BIC: 4905.544
N = 14	Failed to converge	Failed to converge	LogLik: -1405.401 BIC: 4743.541
N = 15	Failed to converge	Failed to converge	LogLik: -1551.814 BIC: 5305.044
N = 16	Failed to converge	Failed to converge	LogLik: -1306.711 BIC: 5100.849
N = 17	Failed to converge	Failed to converge	LogLik: -1219.88 BIC: 5230.531
N = 18	Failed to converge	Failed to converge	Failed to converge

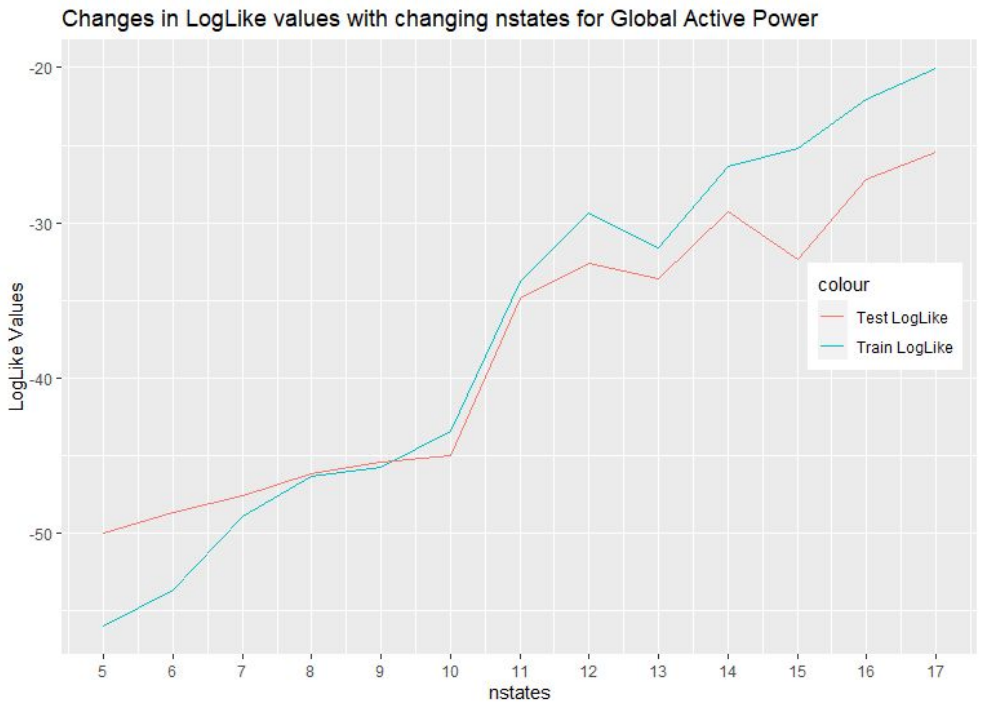


Figure 4. Weekday Plots for LogLik

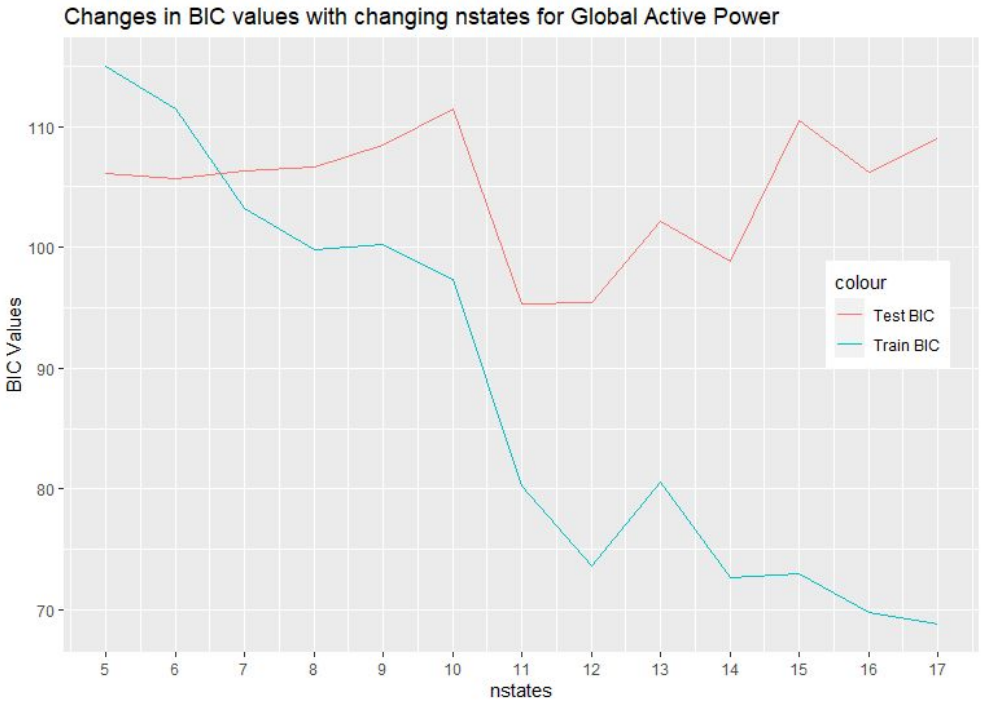


Figure 5. Weekday Plots for BIC

Table 3. Univate Model Training (weekend) Saturday 7am to 9 am

	Voltage	GI	GAP
N = 5	'log Lik.' -17349.76 (df=34) AIC: 34767.51 BIC: 35020.48	'log Lik.' -18173.14 (df=34) AIC: 36414.28 BIC: 36667.24	'log Lik.' -4588.505 (df=34) AIC: 9245.011 BIC: 9497.977
N = 6	'log Lik.' -16178.88 (df=47) AIC: 32451.75 BIC: 32801.44	'log Lik.' -16119.33 (df=47) AIC: 32332.66 BIC: 32682.35	'log Lik.' -4140.969 (df=47) AIC: 8375.937 BIC: 8725.626
N = 7	'log Lik.' -15555.12 (df=62) AIC: 31234.23 BIC: 31695.52	'log Lik.' -15936.48 (df=62) AIC: 31996.95 BIC: 32458.24	'log Lik.' -3548.424 (df=62) AIC: 7220.849 BIC: 7682.14
N = 8	'log Lik.' -14794.24 (df=79) AIC: 29746.48 BIC: 30334.26	'log Lik.' -14353.47 (df=79) AIC: 28864.94 BIC: 29452.71	'log Lik.' -3027.542 (df=79) AIC: 6213.084 BIC: 6800.859
N = 9	'log Lik.' -14276.72 (df=98) AIC: 28749.44 BIC: 29478.58	log Lik.' -13403.56 (df=98) AIC: 27003.12 BIC: 27732.26	'log Lik.' -2568.024 (df=98) AIC: 5332.047 BIC: 6061.185
N = 10	Fails to converge	Fails to converge	'log Lik.' -2428.681 (df=119) AIC: 5095.363 BIC: 5980.744
N = 11	Fails to converge	Fails to converge	'log Lik.' -2266.884 (df=142) AIC: 4817.768 BIC: 5874.274
N = 12	Fails to converge	Fails to converge	'log Lik.' -1878.27 (df=167) AIC: 4090.541 BIC: 5333.051
N = 13	Fails to converge	Fails to converge	'log Lik.' -1690.495 (df=194) AIC: 3768.989 BIC: 5212.385
N = 14	Fails to converge	Fails to converge	'log Lik.' -1277.848 (df=223) AIC: 3001.697 BIC: 4660.857
N = 15	Fails to converge	Fails to converge	'log Lik.' -789.3298 (df=254) AIC: 2086.66 BIC: 3976.466
N = 16	Fails to converge	Fails to converge	Fails to converge

Table 4. Univate Model Testing (weekend) Saturday 7am to 9 am

	Voltage (log like)	GI (log like)	GAP
N = 5	LogLike: -11771.9 BIC: 23838.35	LogLike: -10244.05 BIC: 391109.8	LogLike: -2607.393 BIC: 5509.34
N = 6	LogLike: -10876.56 BIC: 22160.3	LogLike: -8908.903 BIC: 391222.4	LogLike: -2514.196 BIC: 5435.571
N = 7	LogLike: -10714.13 BIC: 21965.39	LogLike: -9784.685 BIC: 391352.4	LogLike: -2316.405 BIC: 5169.938
N = 8	LogLike: -9351.089 BIC: 19386.58	LogLike: -7958.016 BIC: 391499.6	LogLike: -2216.331 BIC: 5117.068
N = 9	LogLike: -9151.285 BIC: 19151.58	LogLike: -7747.033 BIC: 391664.2	LogLike: -1855.411 BIC: 4559.833
N = 10	Fails to converge	Fails to converge	LogLike: -1826.709 BIC: 4684.358
N = 11	Fails to converge	Fails to converge	LogLike: -1640.741 BIC: 4511.68
N = 12	Fails to converge	Fails to converge	LogLike: -1641.586 BIC: 4729.955
N = 13	Fails to converge	Fails to converge	LogLike: -1937.297 BIC: 5555.288
N = 14	Fails to converge	Fails to converge	LogLike: -1759.829 BIC: 5451.589
N = 15	Fails to converge	Fails to converge	LogLike: -1459.2 BIC: 5118.896

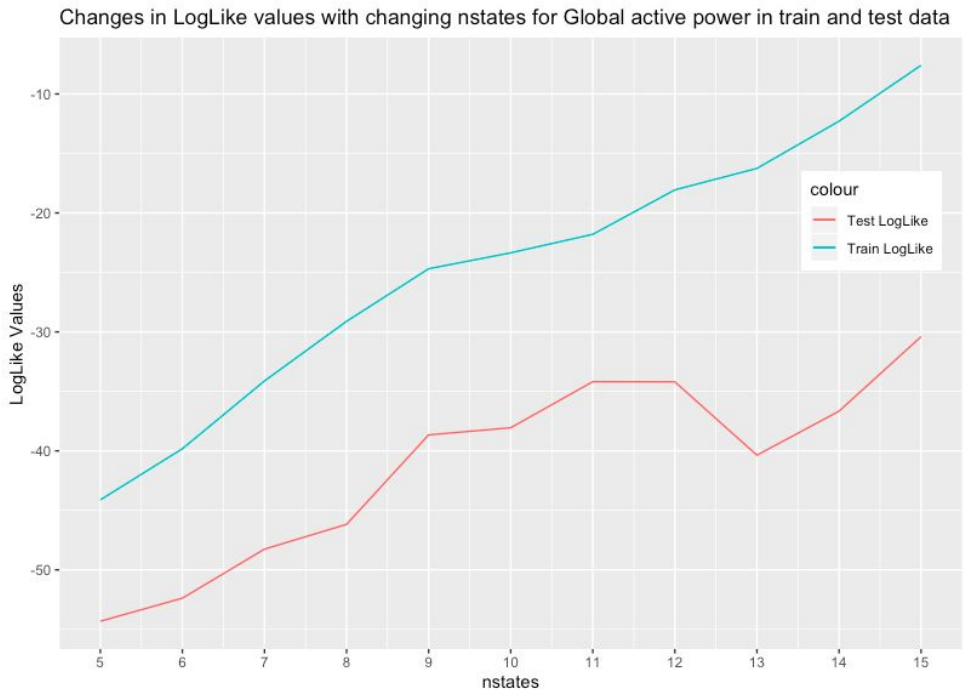


Figure 6. Weekend plots for LogLike (Univariate) for global active power

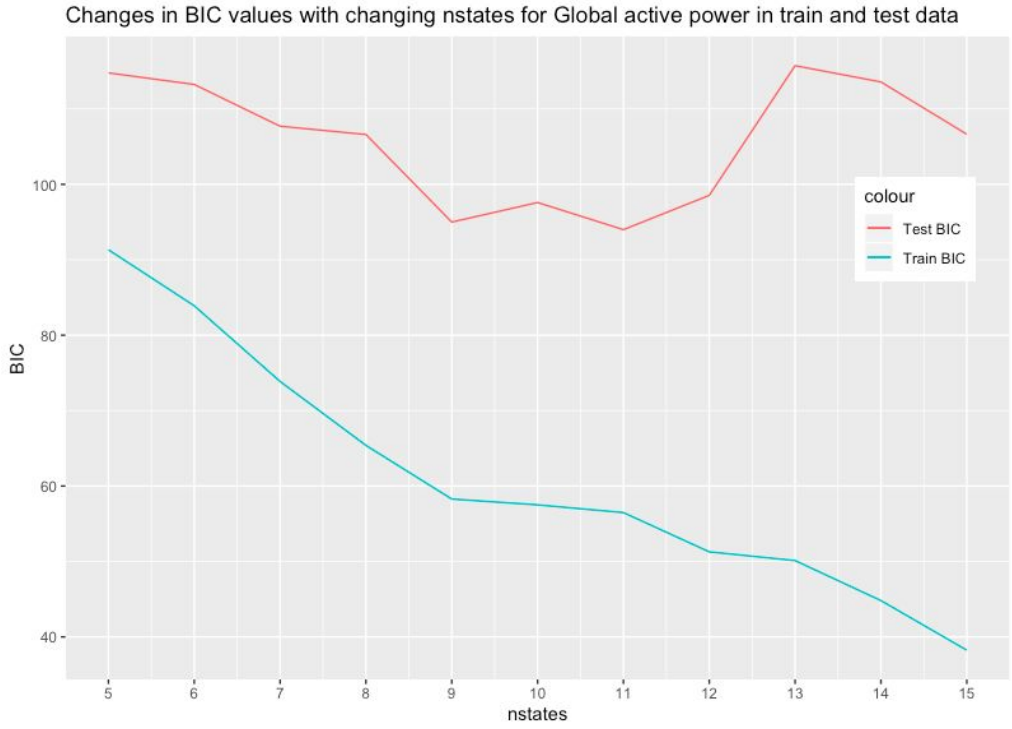


Figure 7. Weekend plots for BIC (Univariate) for global active power

For multivariate models, we trained models using different combinations of these three response variables. The multivariate models that used Global active power and Global intensity had the highest log-likelihoods and lowest BIC values. We observed that N > 14, the trained models starts to overfit the data since the test log-likelihood stops decreasing while train log-likelihood keeps decreasing. Moreover the test BIC starts increasing as well, hence the best multivariate model is Global active power and Global intensity with N = 14.

nstates	train_loglik	train_bic	test_loglik	test_bic
5	-33537.73	67491.66	-15074.709	30530.77
6	-30717.02	61992.14	-13643.470	27798.29
7	-29626.29	59971.49	-13051.345	26761.38
8	-27084.34	55067.30	-11862.680	24548.72
9	-26146.40	53390.06	-11708.343	24422.06
10	-25584.57	52483.97	-11567.768	24340.25
11	-24942.16	51435.62	-11036.463	23494.31
12	-23831.67	49470.06	-10701.946	23059.29
13	-23070.95	48222.92	-10310.172	22527.08
14	-22562.18	47498.62	-10297.794	22771.00
15	-21909.57	46505.57	-9764.149	21989.72
16	-21869.41	46756.31	-10046.828	22858.43
17	-21767.70	46902.89	-9765.287	22616.02
18	-20916.66	45569.72	-9752.852	22929.17
19	-20932.20	45988.62	-10082.925	23944.66
20	-20228.76	44988.48	-9185.136	22521.76

Figure 8. Multivariate Training and Testing (Global Active Power & Global Intensity)

Monday 7:00AM - 9:00AM

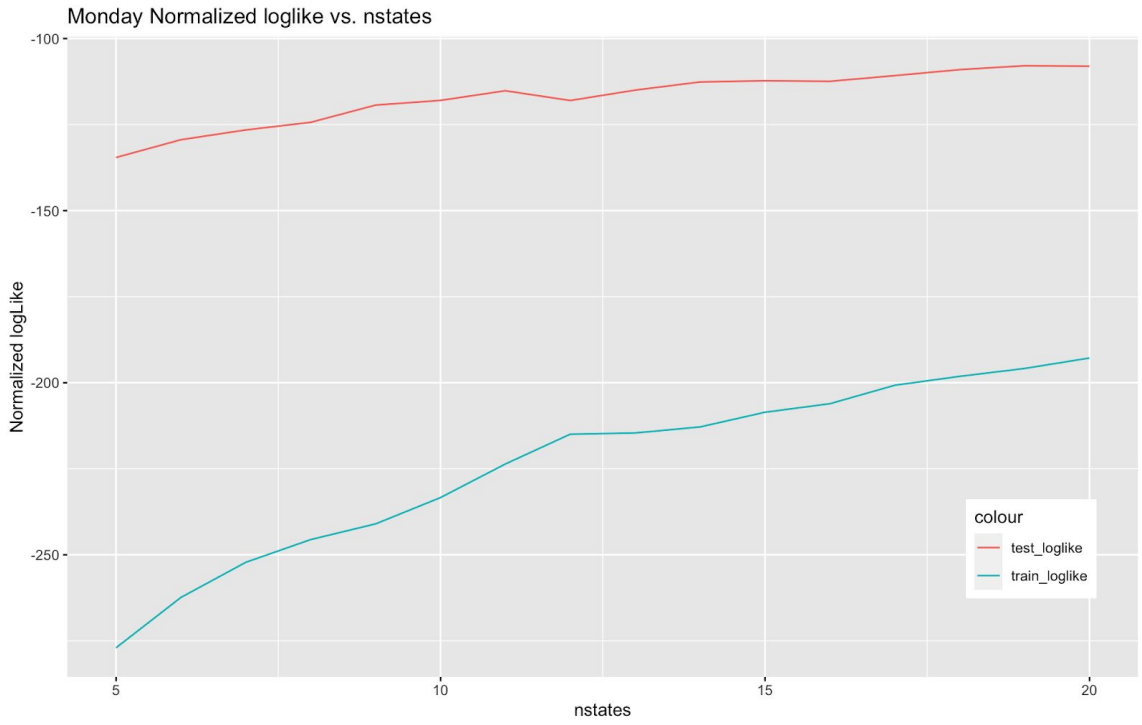


Figure 9. Weekday Plots for Log-Like (Global Active Power & Global Intensity)

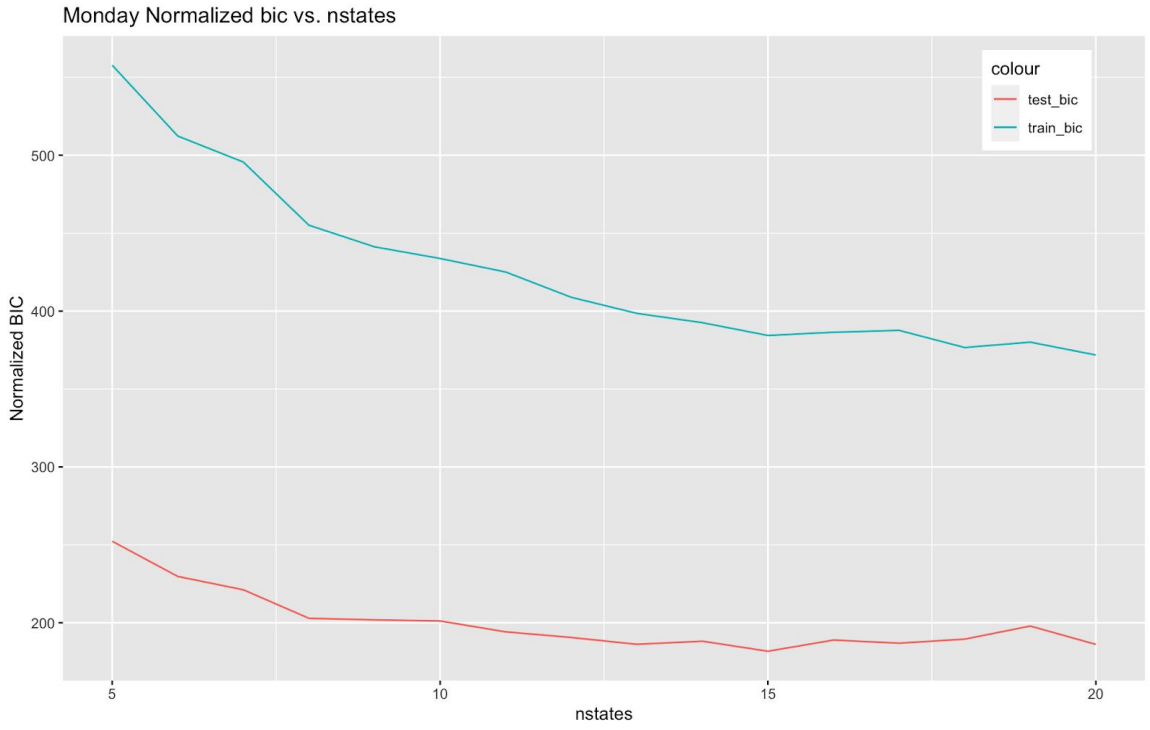


Figure 10. Weekday Plots for BIC (Global Active Power & Global Intensity)

nstates	train_loglik	train_bic	test_loglik	test_bic
5	-29977.813	60370.99	-15882.99	32147.17
6	-27559.415	55675.80	-14458.04	29427.21
7	-26082.847	52883.15	-13826.06	28310.54
8	-24046.274	48989.37	-13331.38	27485.79
9	-23271.979	47639.02	-12861.42	26727.78
10	-23439.042	48190.27	-13152.55	27509.30
11	-21945.211	45438.61	-12320.28	26061.35
12	-21866.155	45535.38	-12489.45	26633.61
13	-20759.377	43595.59	-11746.99	25399.93
14	-12289.185	26947.86	-11521.59	25217.68
15	-20342.907	43366.83	-11727.79	25915.97
16	-20140.905	43293.23	-12240.15	27243.91
17	-18556.902	40474.51	-10969.06	25022.27
18	-17732.456	39193.78	-10407.21	24236.46
19	-9564.393	23244.71	-10255.07	24287.36
20	-8776.896	22075.64	-10202.19	24554.13

Figure 11. Multivariate Training and Testing (Global Active Power & Global Intensity)

Saturday 7:00AM - 9:00AM

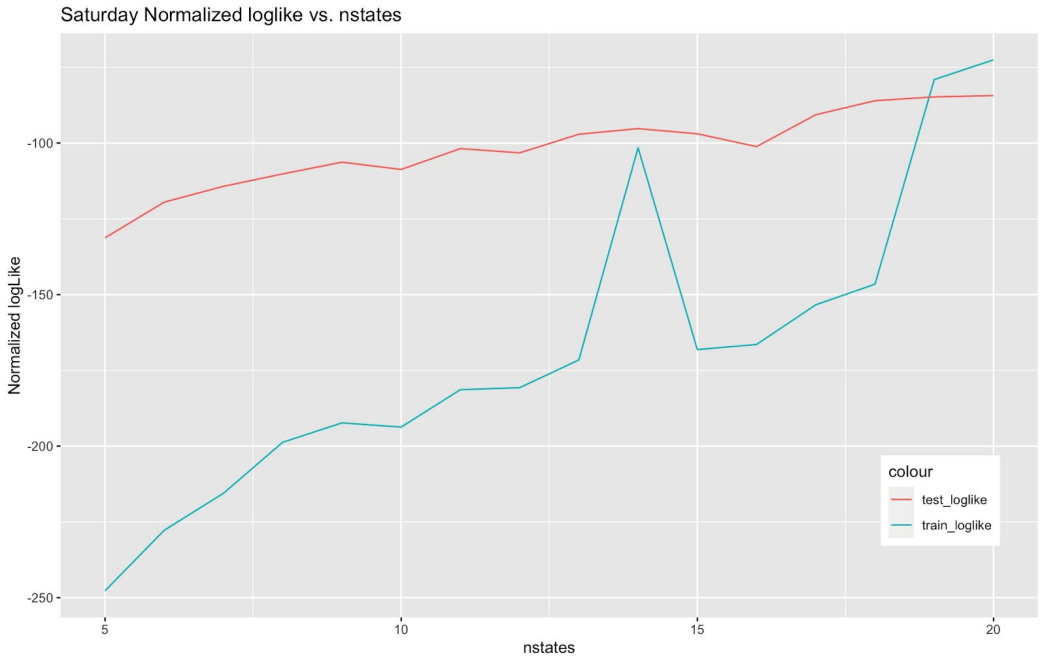


Figure 12. Weekend Plots for Log-Like (Global Active Power & Global Intensity)

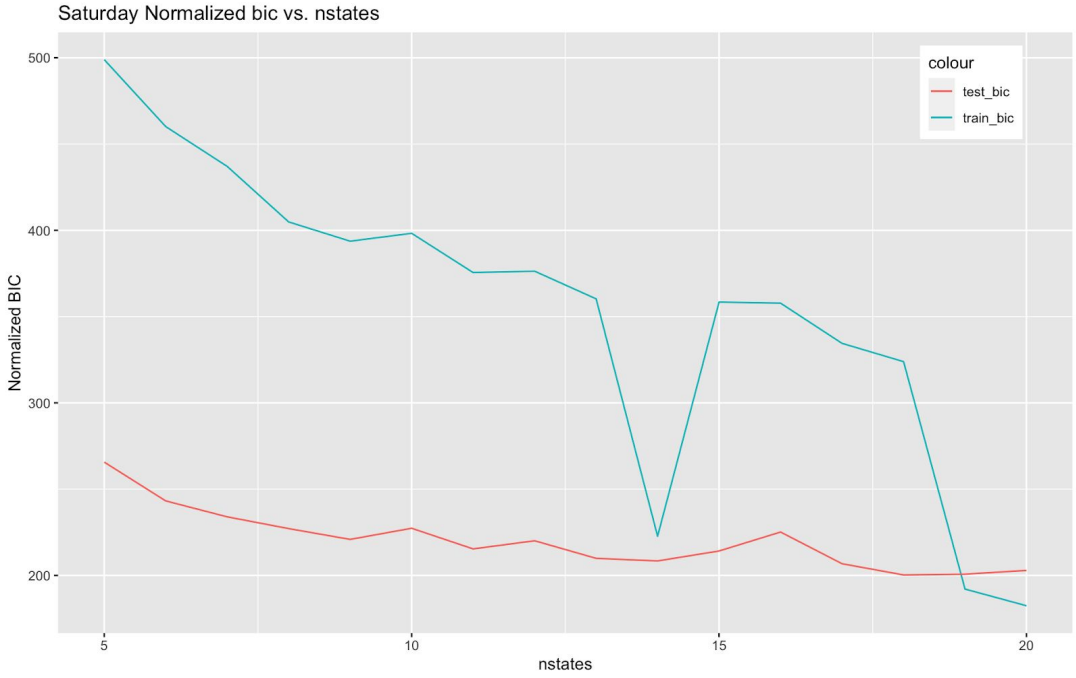


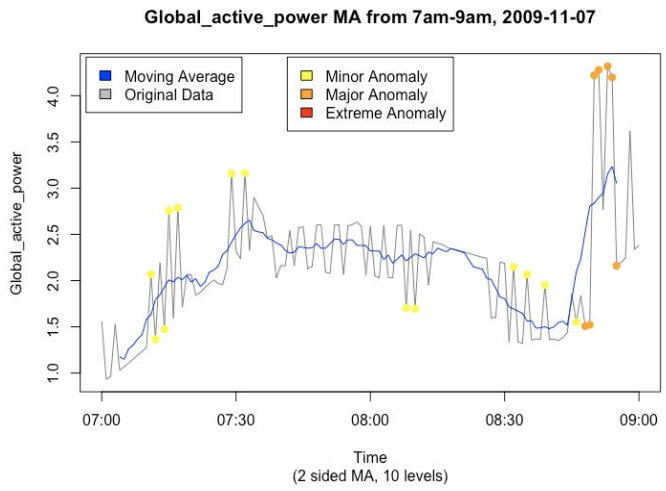
Figure 13. Weekend Plots for BIC (Global Active Power & Global Intensity)

Anomaly detection using Moving Average

By using the moving average anomaly detection technique on a day from the last year of the training (validation) dataset, we observe that there are no extreme anomalies for 2009-11-07; Saturday 7 am to 9 am in the data. This time window had 13 minor anomalies, and 7 major anomalies. For 2009-11-02; Monday 7 am to 9 am from the training (validation) dataset; there were 14 minor anomalies, and zero major/extreme anomalies as seen in the table below.

The number of anomalies in the injected anomalies test data could be found in the table below.

Table 5.



Dataset: TrainingData.txt

Date: 2009-11-07 (Saturday)

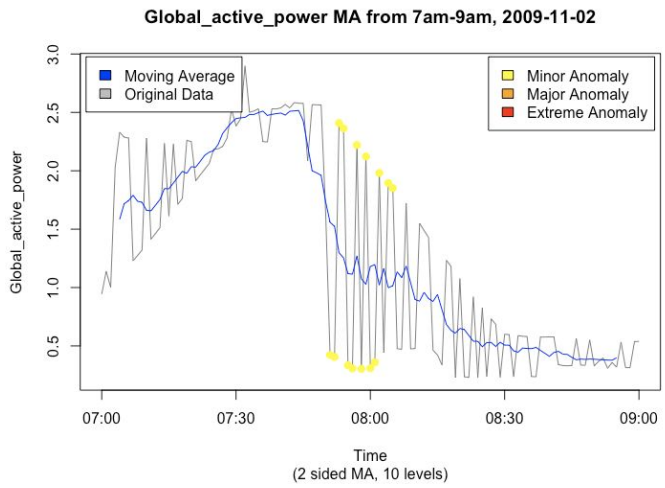
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 13

Major Anomalies: 7

Extreme Anomalies: 0



Dataset: TrainingData.txt

Date: 2009-11-02 (Monday)

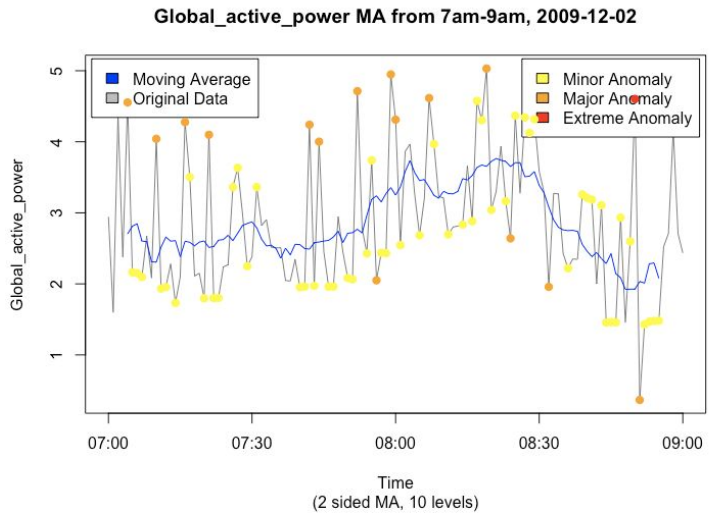
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 14

Major Anomalies: 0

Extreme Anomalies: 0



Dataset: test1.txt

Date: 2009-12-07 (Saturday)

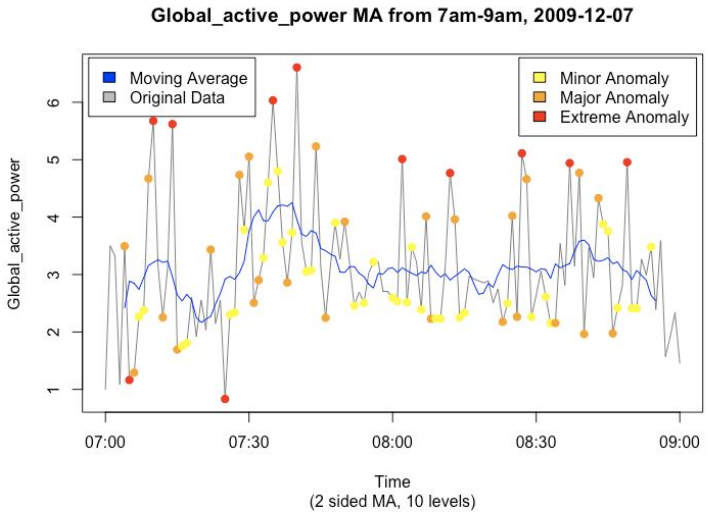
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 53

Major Anomalies: 15

Extreme Anomalies: 1



Dataset: test1.txt

Date: 2009-12-09 (Monday)

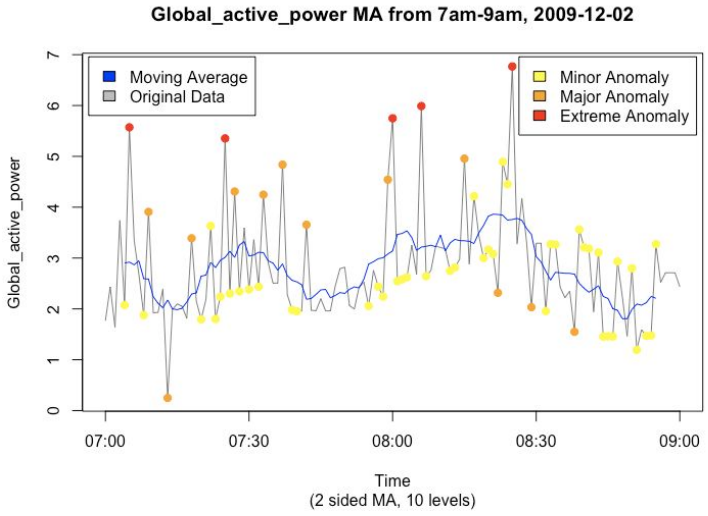
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 37

Major Anomalies: 26

Extreme Anomalies: 11



Dataset: test2.txt

Date: 2009-12-07 (Saturday)

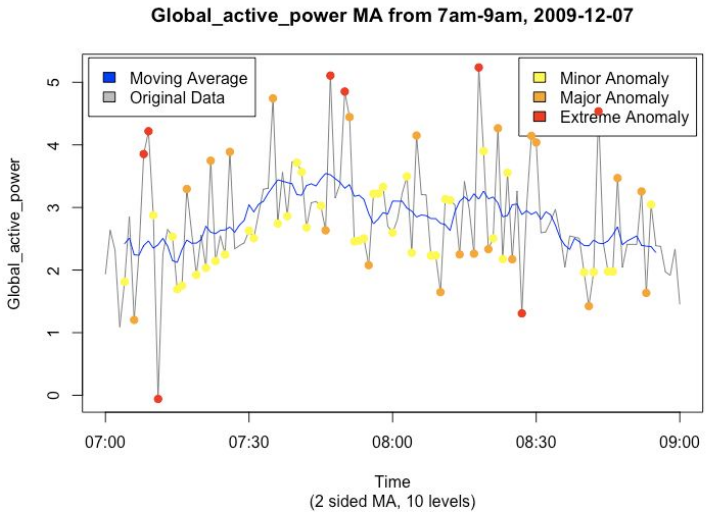
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 43

Major Anomalies: 12

Extreme Anomalies: 5



Dataset: test2.txt

Date: 2009-12-09 (Monday)

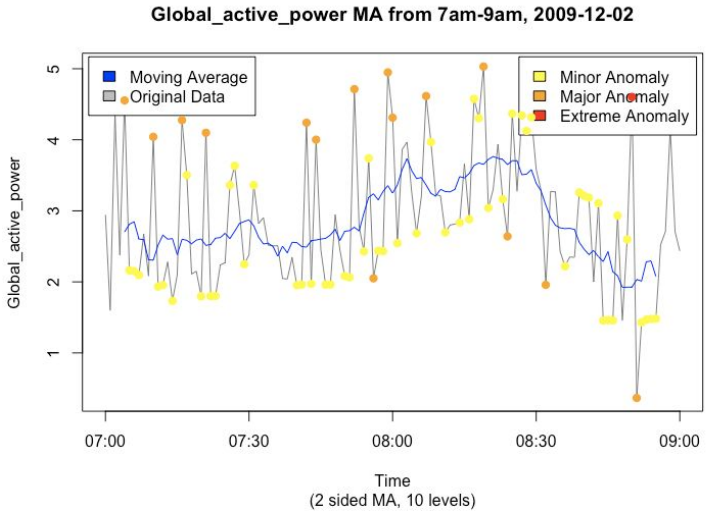
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 39

Major Anomalies: 21

Extreme Anomalies: 8



Dataset: test3.txt

Date: 2009-12-07 (Saturday)

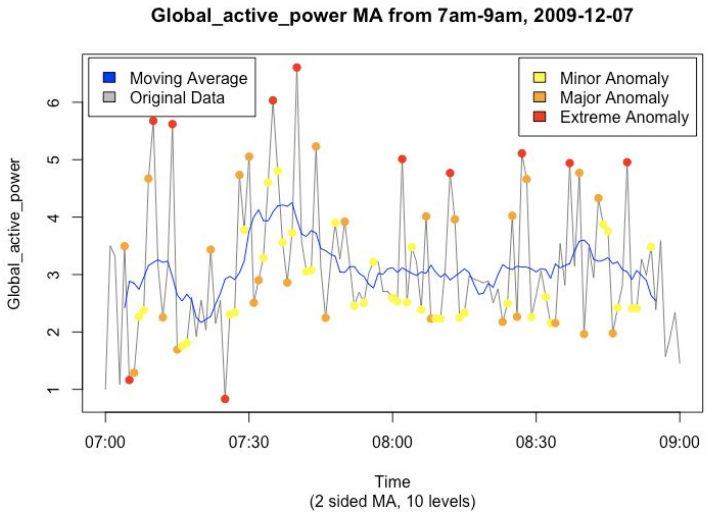
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 53

Major Anomalies: 15

Extreme Anomalies: 1



Dataset: test3.txt

Date: 2009-12-09 (Monday)

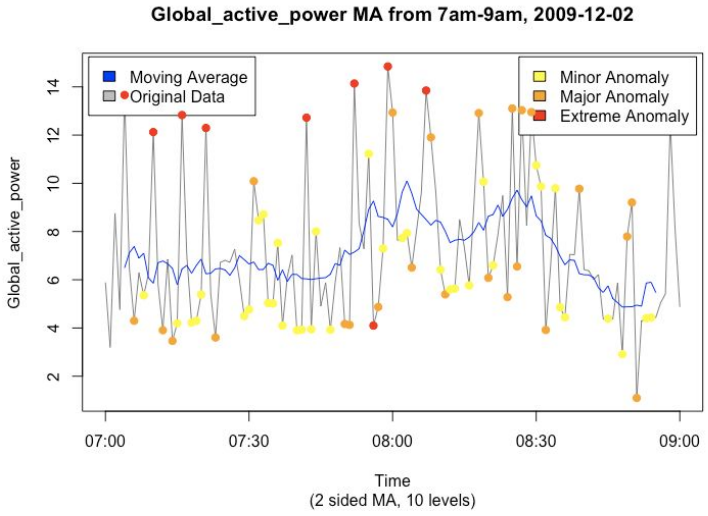
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 37

Major Anomalies: 26

Extreme Anomalies: 11



Dataset: test4.txt

Date: 2009-12-07 (Saturday)

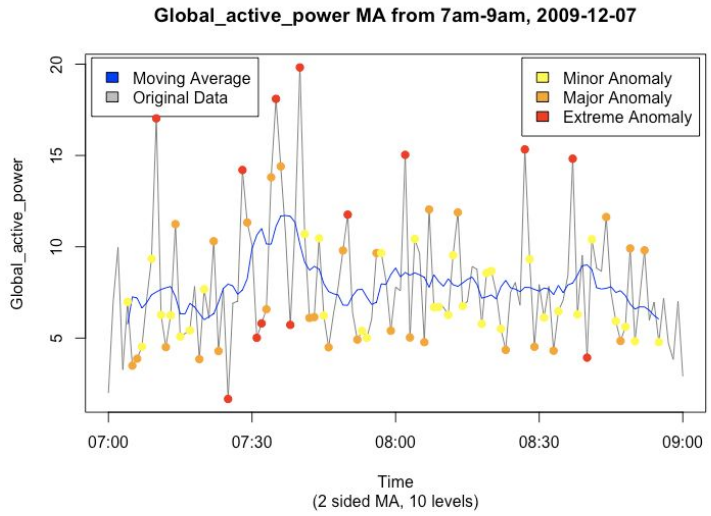
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 37

Major Anomalies: 24

Extreme Anomalies: 9



Dataset: test4.txt

Date: 2009-12-09 (Monday)

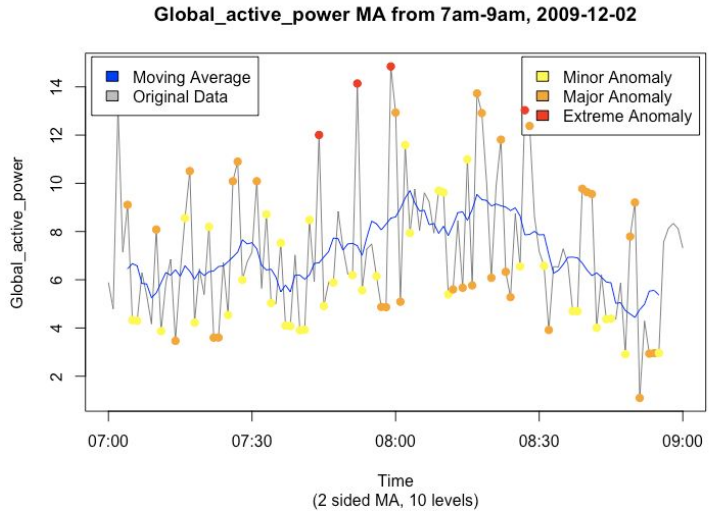
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 33

Major Anomalies: 29

Extreme Anomalies: 13



Dataset: test5.txt

Date: 2009-12-07 (Saturday)

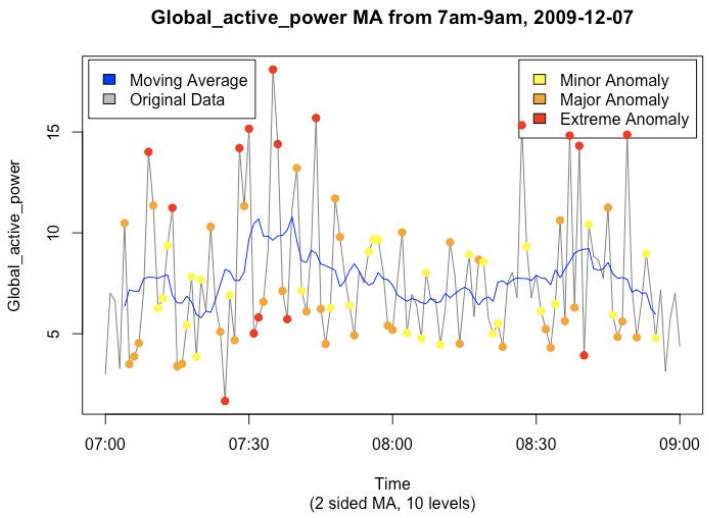
Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 36

Major Anomalies: 32

Extreme Anomalies: 4



Dataset: test5.txt

Date: 2009-12-09 (Monday)

Time window: 7:00AM - 9:00AM

Moving window size: 10 samples

Minor Anomalies: 29

Major Anomalies: 36

Extreme Anomalies: 16

Anomaly detection using HMM

Univariate model

After fitting the univariate HMM model to the training dataset (first three years) using the Global active power for Mondays 7 am to 9 am, the model had a normalized log-likelihood of -38.305 (BIC = 89.281). On the validation dataset, which consisted of data that our model did not have access to (last year of the training dataset), our model had a log-likelihood of -40.2568 (BIC = 106.1535). The log-likelihood and BIC values for both training and validation datasets were very similar. This indicates that the univariate HMM model was not overfitting to the training data. The results for HMM models trained on data from Saturday 7 am to 9 am were similar to Monday (see Table 6 for the results). The validation dataset log-likelihood can be interpreted as there being no anomalies in the validation dataset because the log-likelihood is so similar to the training dataset. This conclusion is further supported by the moving average anomaly detection which showed that there were no extreme anomalies in the dates selected from the validation dataset (2009-11-02 - Monday, and 2009-11-07 - Saturday).

When this model was used to detect anomalies on the five provided test datasets with injected anomalies, the log-likelihood values were much different from the validation dataset. The log-likelihood values of the first three test datasets (Test 1, Test 2, Test 3) were similar to each other (-193.0328, 190.3317, -193.6763 respectively). From the moving average anomaly detection, these three test datasets had a similar amount of anomalies; 26, 21, 26 major and 11, 8, 11 extreme anomalies respectively for the time window chosen.

The log-likelihood of the Test 4 (-1112.323), and Test 5 (-1107.113) were significantly different from the log-likelihood of the training and validation datasets. These datasets had 29, and 36 major, and 13, and 16 extreme anomalies respectively from the moving average anomaly detection. These significant differences in log-likelihood could be attributed to the anomalies detected in the data by the HMM model. Overall, the HMM results are validated by the moving average anomaly detection technique.

Table 6. Normalized LogLikelihood and BIC values for Univariate HMM (Global active power, nstates = 11)

Mondays							
	Training	**Test (from TrainData.txt)**	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**
LogLike	-38.305	-40.25682	-193.0328	190.3317	-193.6763	-1112.323	-1107.113
BIC	89.281	106.1535	410.7958	404.9639	411.653	2248.946	2238.527
Saturdays							

LogLike	-69.743	-36.01246	-183.4843	-182.1047	-183.484	-678.5189	-675.3685
BIC	51.967	97.74705	391.2691	388.5098	391.269	1381.338	1375.037

Multivariate model

The multivariate HMM model trained using Global active power, and Global intensity (for Monday 7 am to 9 am) resulted in a log-likelihood of -213.537 on the training dataset, and -215.698 on the validation dataset. The similarities in log-likelihood once again indicates that overfitting did not occur in the model we trained. (See Table 8 below for Saturday multivariate HMM results).

Using the multivariate HMMs to detect anomalies in the provided Test datasets, the differences between training log-likelihood (-213.537) and Test 1, 2, and 3 log-likelihood (-350.830, -324.704, -347.419 respectively) were not as significant as the univariate model. Once again, Test 4 (-1867.038) and Test 5 (-1848.149) log-likelihood were significantly different from training log-likelihood.

A possible conclusion that could be drawn from comparing univariate models with multivariate models is that univariate models are much more sensitive to fluctuations in the data because they only depend on one variable. Whereas, the multivariate model is more robust. This could mean that the univariate model may detect more false positives.

Table 7. Normalized LogLikelihood and BIC values for Multivariate HMM (Global active power, and Global intensity, nstates = 14)

Monday							
	Training	**Test**	**Test 1**	**Test 2**	**Test 3**	**Test 4**	**Test 5**

		(from TrainData .txt)					
LogLike	-213.537	-215.698	-350.830	-324.704 3	-347.419	-1867.03 8	-1848.14 9
BIC	449.472	476.7189	745.3748	692.264	737.696	3776.932	3739.153
Saturday							
LogLike	-200.474	-257.3894	-351.822 8	-350.828 6	-351.822 8	-1247.72	-1239.78
BIC	423.732	560.2455	746.3	744.3115	746.2999	2538.11	2522.215

Problems encountered

After completion of the PCA analysis, we found that the PCA graphs between monday & saturday showed different eigenvectors for the same variables. This meant that if we were to create a model for saturdays, we could only refer to data from saturdays to explain phenomena occurring on saturdays. We see in this analysis that despite having access to the same data types, the interaction between them differ drastically due to the characteristics of the underlying state changes that produce these observables (i.e. people getting ready for work Monday morning vs. staying home on Saturday mornings) . The observed data exclusivity just mentioned points to the challenge of unavailability of labels to differentiate normal & outlier data. To counteract this, each model must be built without any preconceptions about variable interaction as well as selections of appropriate time windows and an analysis such as PCA must

be used to identify important variable interactions that a human operator may not be able to perceive simply by looking at graphs.

When training the multivariate HMMs we had originally intended to use sub metering 1, 2 & 3 to support the global active power time series models. We tried to interpret these variables as multinomial sequences because they had low correlation with one another but a strong correlation with global active power. Unfortunately we could not obtain results for this model because after fitting the trained models, they had a mismatch of the number of parameters when compared to the test models. This meant that we could not get the trained model parameters & set them on the test data. We believe the issue came from an invalid dataset, values in the sub meter time series followed a somewhat binary system, alternating between high and low values with very few values existing in between these values. We could have interpreted these data points as binary by setting a cutoff point. However we decided against this since the unit of measurement was not supplied to us and we did not feel qualified to set an arbitrary cutoff point for this data.

Lessons learned

For feature selection, we learned what Principal Component Analysis is and how to use it to determine the set of response variables to use for training a model. The first and second principal component captures the most variance in the original data, and so we need the response variables that have a strong influence in those two principal components. Furthermore, we realized the importance of knowing what the different response variables mean in the context of the problem. For instance, the Submetering response variables have strong influence in the first principal component for Mondays and so we used it for training models in the beginning of the project. Yet we later discovered that these submetering

responses are not useful individually, and all three must be used for training if one is to be included in training. This was interesting information that would have lessened our struggles in the beginning of this project.

Another interesting observation we made is that univariate models finish running quicker than multivariate models, but also stop converging at low numbers of states. For instance, when training Global active power at nstate=18 the depmix function reaches the max number of iterations before converging. When training with Global intensity at nstate=10, depmix outputs a warning and exits the function. This is not the case for the multivariate models, which take longer to train but converge for a high number of states. Although this could be a property of this specific data set, it could also be something to think about for future projects similar to this.

Conclusion

In order to preserve a safe and reliable power grid, it is paramount to have the ability to detect anomalies that could be indicative of intrusions in critical infrastructure relied upon by our society. In this project, we demonstrated the efficacy of HMMs trained on a small subset of past historic data to detect anomalies in the data collected in the future and validated this with the use of moving average anomaly detection techniques.

References

[1] “Dimensionality Reduction Algorithms: Strengths and Weaknesses.” *Elite Data Science*, 25 Jan. 2019, elitedatascience.com/dimensionality-reduction-algorithms.

[2] Shaikh, Raheel. “Feature Selection Techniques in Machine Learning with Python.” *Medium*, Towards Data Science, 28 Oct. 2018, towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e.

[3] Lawrence R. Rabiner. 1990. A tutorial on hidden Markov models and selected applications in speech recognition. Readings in speech recognition. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 267–296.

Scree plot for Monday 7 am to 9 am

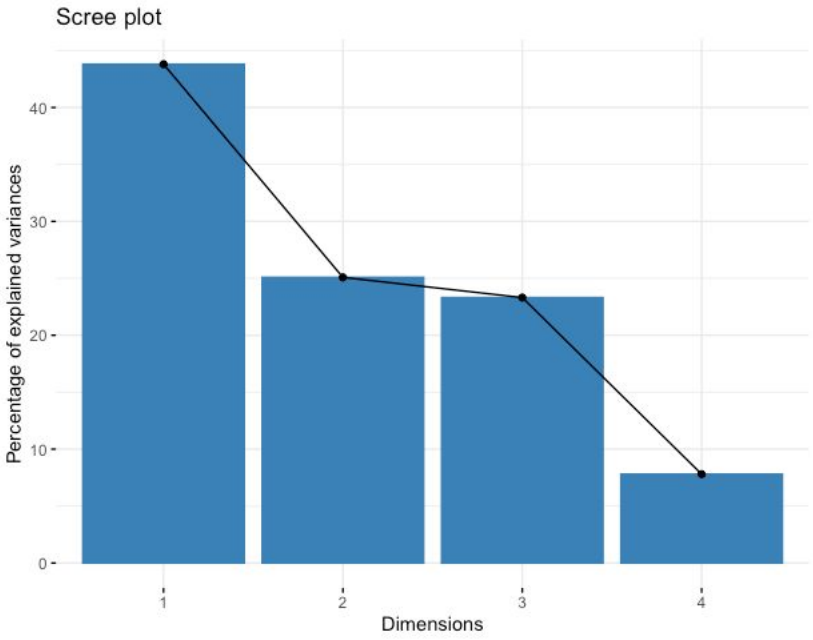


Figure 14.

Scree plot for Saturdays

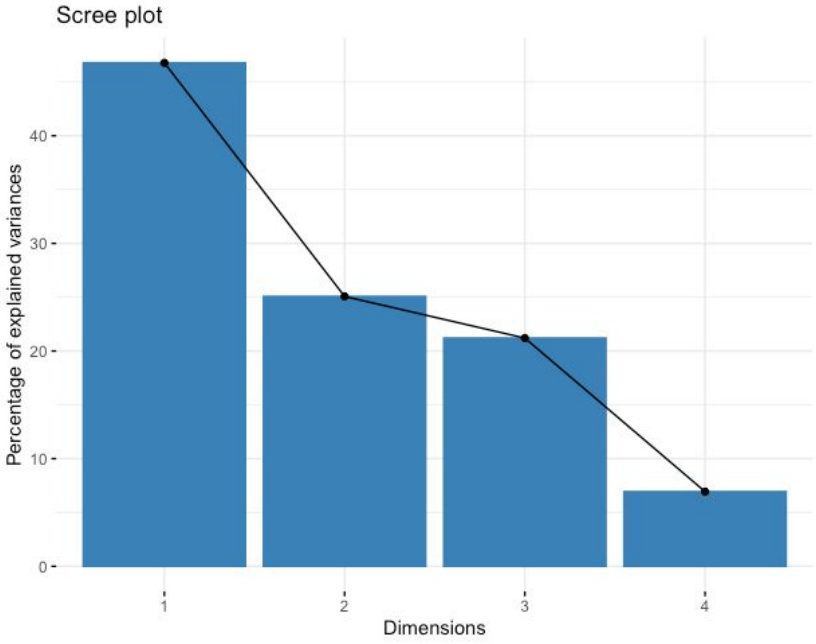


Figure 15.

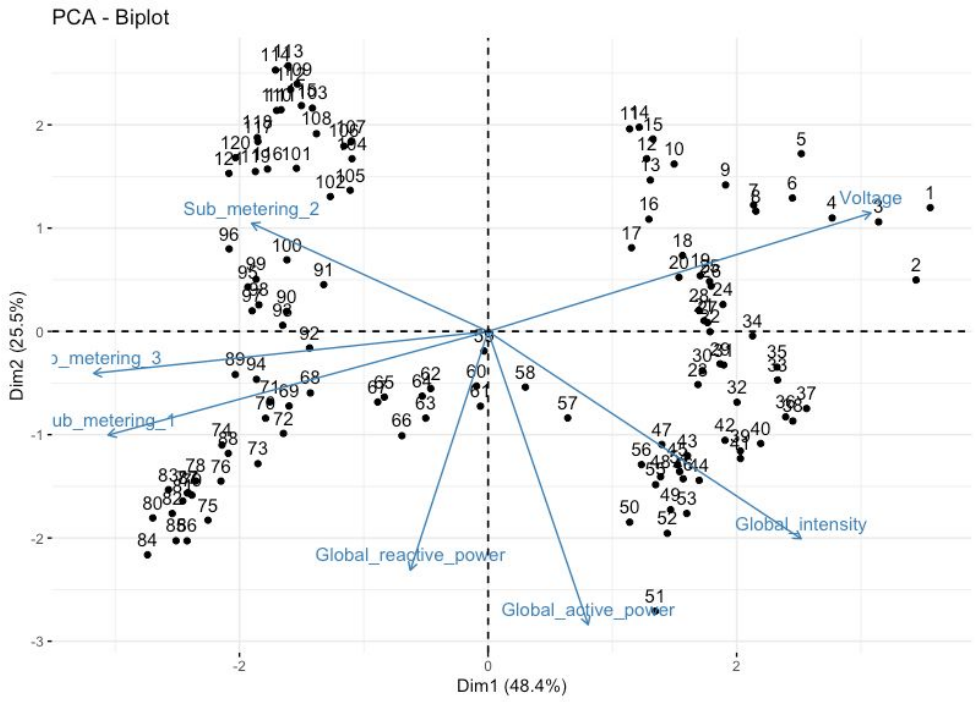


Figure 16.

Saturday

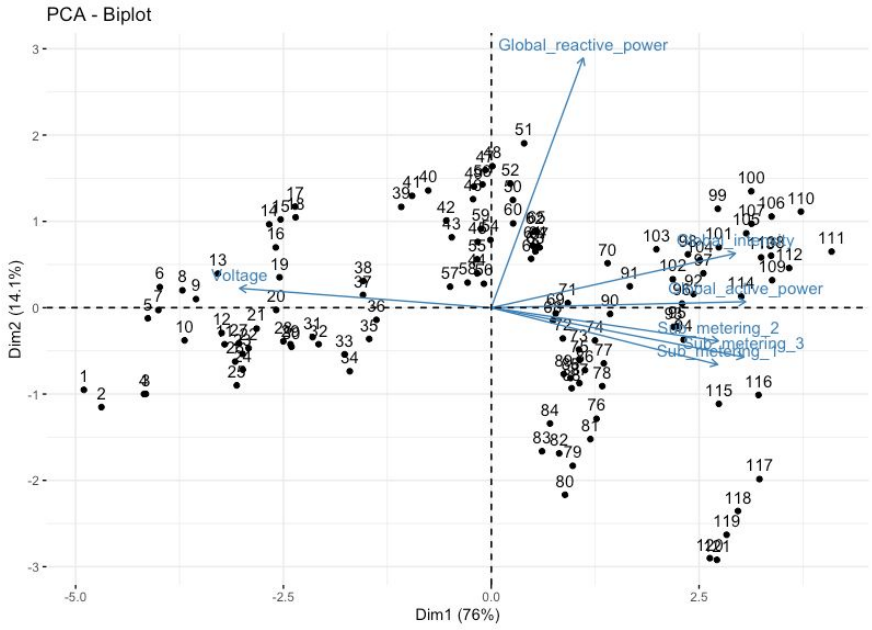


Figure 17.