Detecting Anomalies in Electricity Consumption

Semester: Spring 2020

Course: CMPT 318 - Cybersecurity

Authors:

Wael Yakoub Agha - 301348817

Rabie Ali - 301320609

Sam Gouneili - 301306477

Giovanni Hosang - 301295511

Russell Wong - 301310883

Jack Wright - 301315736

Abstract

The purpose of this report is to demonstrate the understanding of specific characteristics of multivariate time series data and to develop an analytic approach to detect different types of anomalies. Anomaly detection based intrusion detection methods were utilized in order to gain cyber situational awareness in the analysis of automated control processes. Point anomalies, contextual anomalies, and behavioral anomalies were analyzed during this project. Point anomalies are seen as outliers in the data and are identified through a sliding window approach. Contextual anomalies follow a similar method. For behavioural anomalies, we built Hidden Markov Models (HMM) and applied them to test data.

Table of Contents

Table of Figures

1 Data Exploration

1.1 Data Preparation

Cleaning and reformatting data is a necessary step prior used to reduce and catch errors that can influence a model's predictive ability. Observing the provided Electricity Consumption dataset, our group chose to impute these missing values using R's Multivariate Imputations via Chained Equations (MICE) package with the Predictive Mean Matching (PMM) model. This strategy was chosen because MICE provides more accurate results than deleting observations with missing variables or looking at K nearest neighbors. These K values are dependent on choosing an accurate K amount of neighbors to look at and create a predictive value. Considering our dataset which contained many missing values, we would most likely get very inaccurate results. Using the imputation model PMM, calculations of missing values were done by looking at predicted values based on a set of candidate donors. Candidate donors are observation cases that have similar predicted values to our missing entry. Generally, 3,5 or 10 candidates are picked, and then lastly one is chosen to be used. As a middle-ground of computational time and accuracy, we decided to look and choose from 5 candidate donors.

1.2 Correlation Analysis

Upon fixing the data set by replacing null and corrupt values as mentioned in the section above, the next task was to determine the correlation matrix. Correlation analysis was done for each pair of disjoint pairs of dependent variables. The following figure is the visual representation of the result:

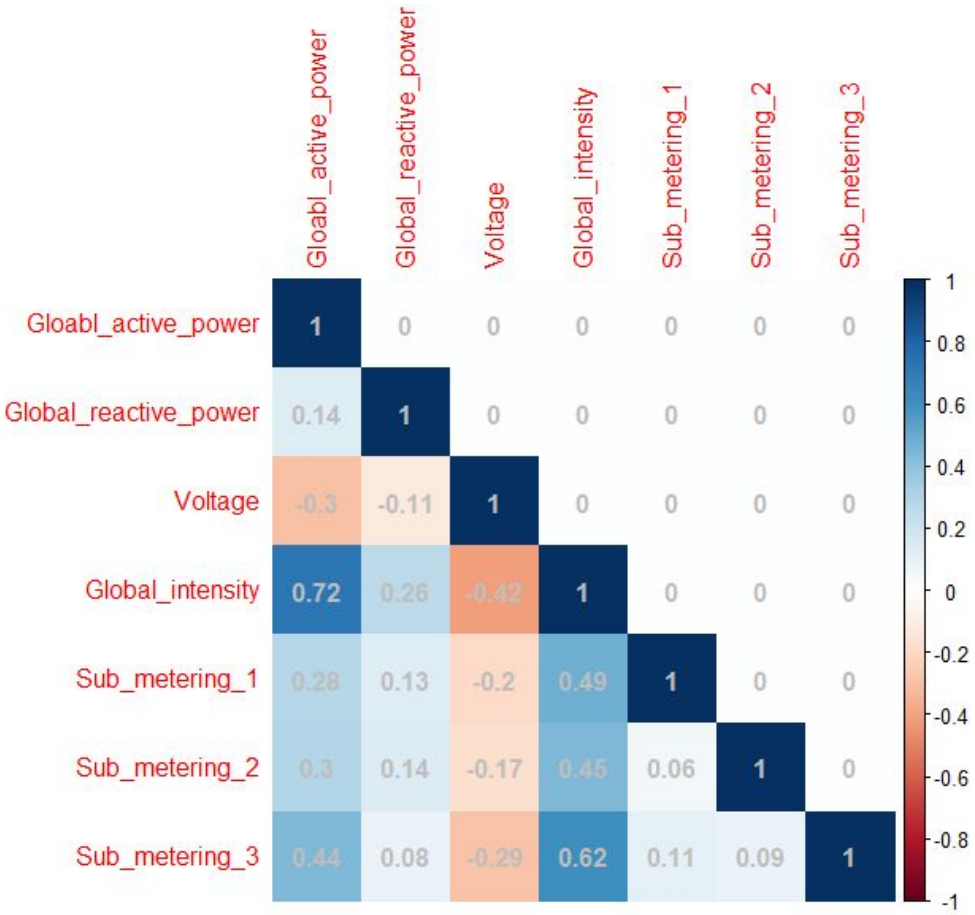


Figure 1.2: Correlation Matrix Visual Representation

It can be observed that Global Intensity and Global Reactive Power have the highest correlation factor. Additionally, Global Intensity is highly correlated with all three features

of sub metering. In terms of negative correlation, Voltage is least correlated with Global Intensity, Global Reactive Power, and all three sub meterings .

1.3 Observation Time Window

The observation time window was determined by finding a common pattern in electricity consumption over several hours between a couple of days. First, Global Intensity was chosen as the comparison feature, and its values for random five consecutive weekdays were plotted on top of each other. The following graph was the result:

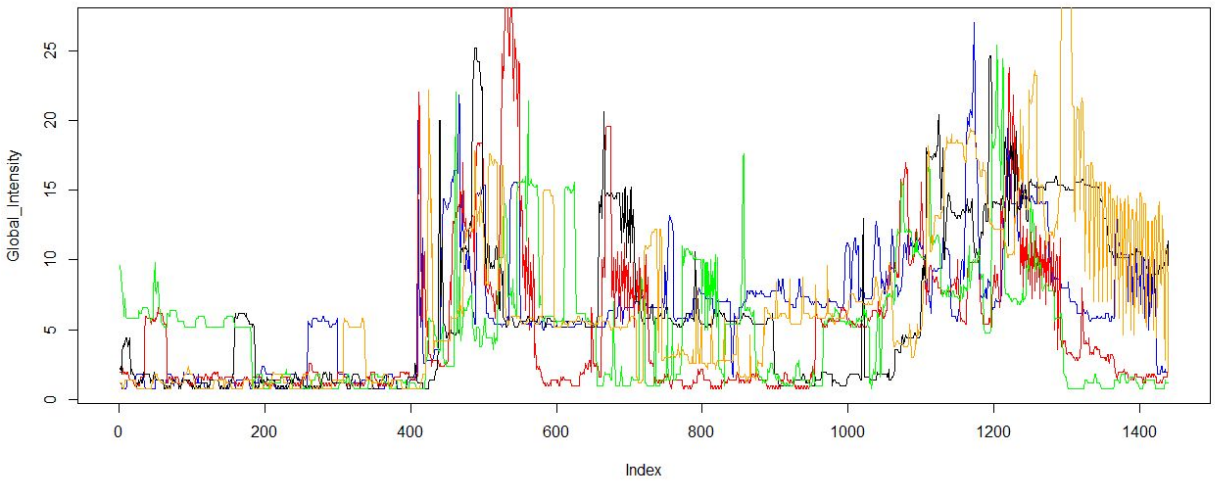


Figure 1.3.1: Global Intensity values for five weekdays on top of each other

It can be observed that the five graphs have similar values of Global Intensity on the interval [200, 400] which corresponds to the time interval [3:20 am, 6:40 am]. This pattern corresponds with real-life electricity consumption patterns since during this time

interval over weekdays, people are usually sleeping, and thus, electricity consumption is low. Therefore, the chosen time window is [3:20 am, 6:40 am].

The chosen time window also shows a pattern in Global Intensity values over weekend days. The following figure is the result of plotting the values of Global Intensity of three random weekdays on top of each other:

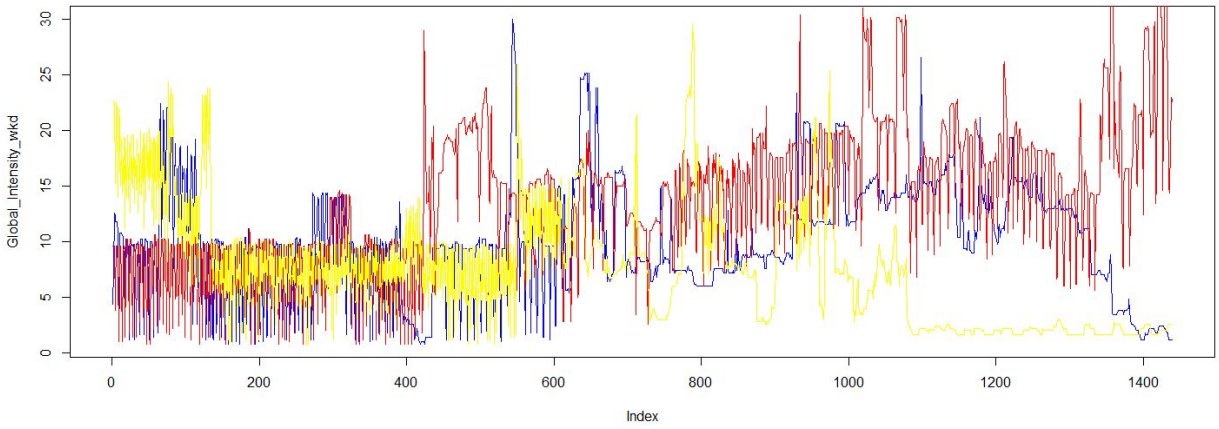


Figure 1.3.2 - Global Intensity values for three weekend days on top of each other

It can be observed that the graphs have similar behavior of Global Intensity on the interval [200, 400].

Eventually, the following graphs are the values of the seven features of the data set on the time interval [3:20 am, 6:40 am] for a randomly chosen weekday and a weekend day:

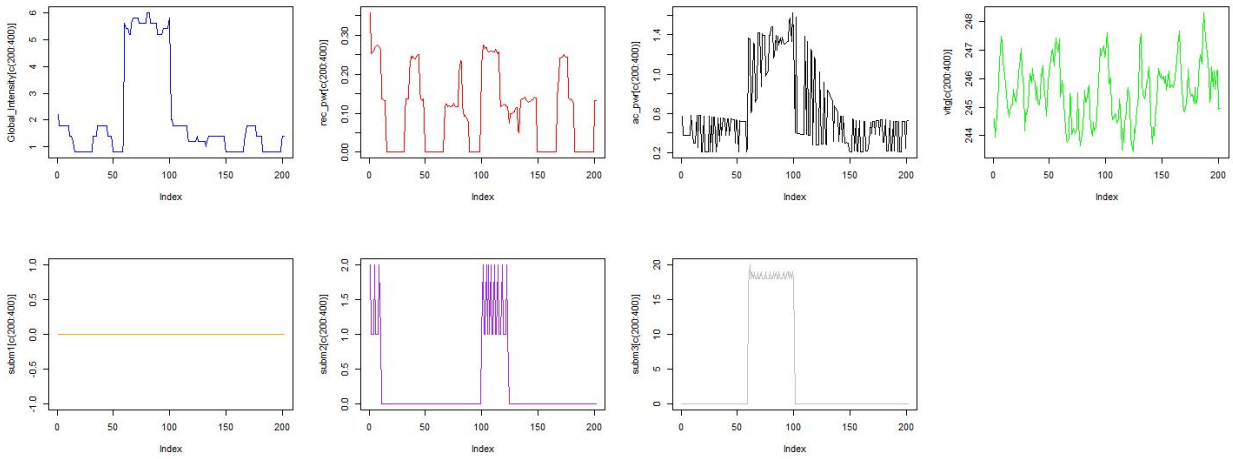


Figure 1.3.3: Values of each feature for the chosen time window over a weekday

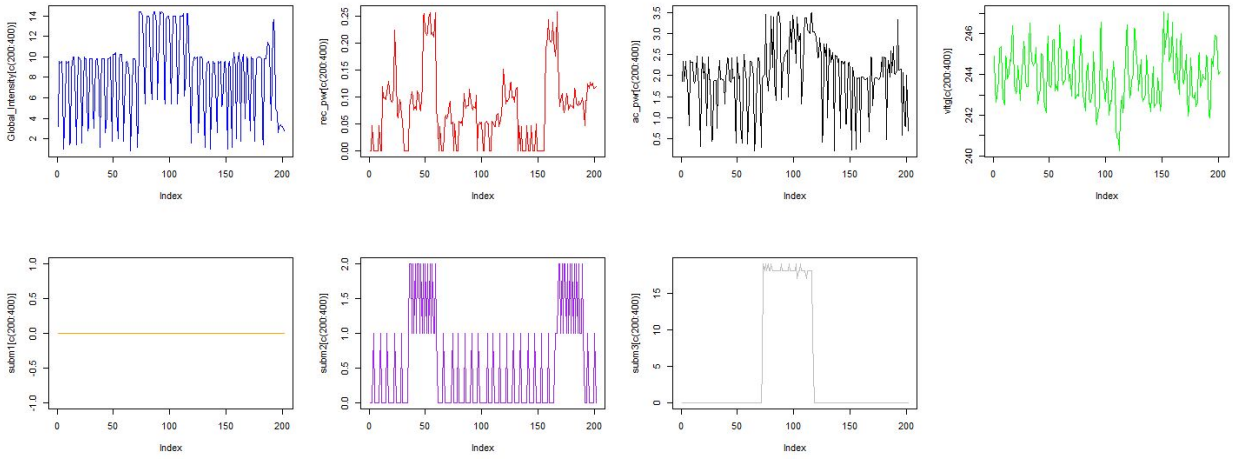


Figure 1.3.4: Values of each feature for the chose time window over a weekend day

2 Feature Engineering

2.1 Principal Component Analysis (PCA)

Principal Component Analysis is used when unsupervised data needs to be used for training.

This method transforms and simplifies several sets of observations into a set of values of linearly uncorrelated variables. This can assist in choosing the proper component for creating a model.

From our dataset we can extract the following information after applying the PCA algorithm:

```
Importance of components:
                          PC1    PC2    PC3    PC4    PC5     PC6     PC7
Standard deviation     1.6907 0.9991 0.9698 0.9136 0.8779 0.68648 0.35529
Proportion of Variance 0.4084 0.1426 0.1344 0.1192 0.1101 0.06732 0.01803
Cumulative Proportion  0.4084 0.5509 0.6853 0.8045 0.9146 0.98197 1.00000
```

We have obtained 7 principal components, which we call PC1-7. Each of these explains a percentage of the total variation in the dataset. That is to say: PC1 explains 41% of the total variance, which means that nearly two-fifths of the information in the dataset (7 variables) can be encapsulated by just that one Principal Component. PC2 explains 14% of the variance. So, by knowing the position of a sample in relation to just PC1 and

PC2, we can get a very accurate view on where it stands in relation to other samples, as just PC1 and PC2 can explain 55% of the variance.

tbl_f.pca	list [5] (S3: prcomp)	List of length 5
sdev	double [7]	1.691 0.999 0.970 0.914 0.878 0.686 ...
rotation	double [7 x 7]	-0.46864 -0.19467 0.33041 -0.55972 -0.29883 -0.28369 0.13465 -0.74428 -0.13304 ...
center	double [7]	1.227 0.122 240.548 4.639 1.154 1.356 ...
Global_active_power	double [1]	1.22712
Global_reactive_power	double [1]	0.121822
Voltage	double [1]	240.5479
Global_intensity	double [1]	4.638983
Sub_metering_1	double [1]	1.153558
Sub_metering_2	double [1]	1.356428
Sub_metering_3	double [1]	6.164008
scale	double [7]	1.056 0.112 3.260 4.575 6.268 6.015 ...
Global_active_power	double [1]	1.055796
Global_reactive_power	double [1]	0.111635
Voltage	double [1]	3.260495
Global_intensity	double [1]	4.574518
Sub_metering_1	double [1]	6.268301
Sub_metering_2	double [1]	6.014877
Sub_metering_3	double [1]	8.313937
x	double [1556444 x 7]	-4.216455 -5.716708 -5.459262 -5.879703 -3.832515 -4.024042 -0.733028 -0.602789 ...

Figure 2.1: PCA calculation output

This PCA object contains the following information:

- The center point ($center), scaling ($scale), standard deviation(sdev) of each principal component
- The relationship (correlation or anticorrelation, etc) between the initial variables and the principal components ($rotation)
- The values of each sample in terms of the principal components ($x)

2.2 Complete Dataset PCA Plot

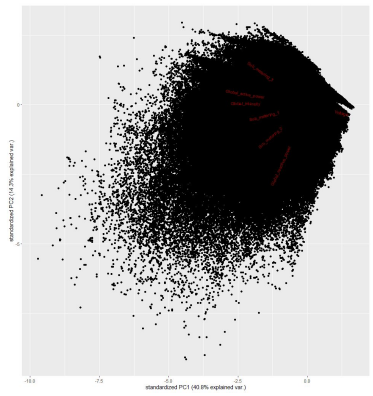


Figure: PC 1 vs 2 (40.8% : 14.3%)

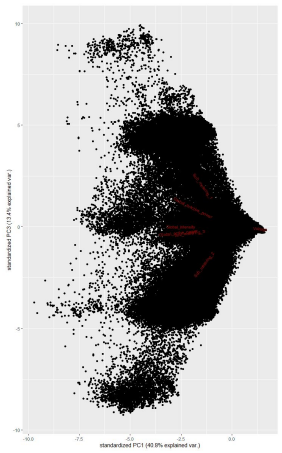


Figure: PC 1 vs 3 (40.8% : 13.4%)

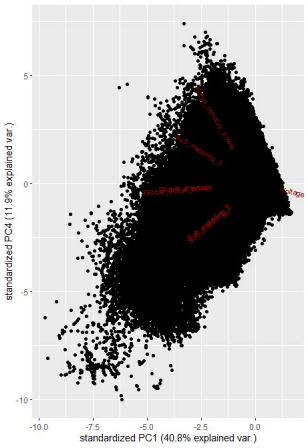


Figure: PC 1 vs 4 (40.8% : 11.9%)

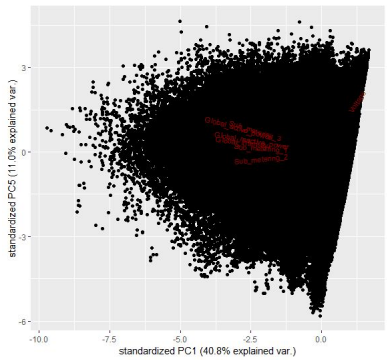


Figure: PC 1 vs 5 (40.8% : 11%)

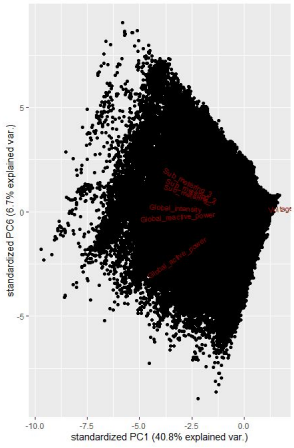


Figure: PC 1 vs 6 (40.8% : 6.7%)

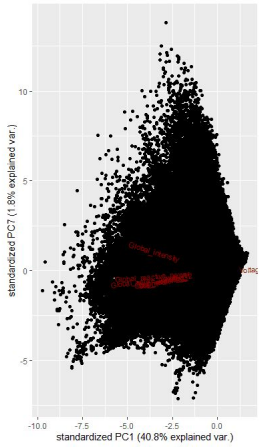


Figure: PC 1 vs 7 (40.8% : 1.8%)

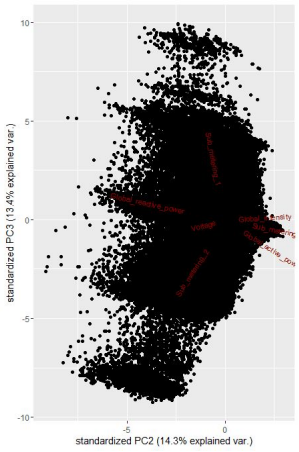


Figure: PC 2 vs 3 (14.3% : 13.4%)

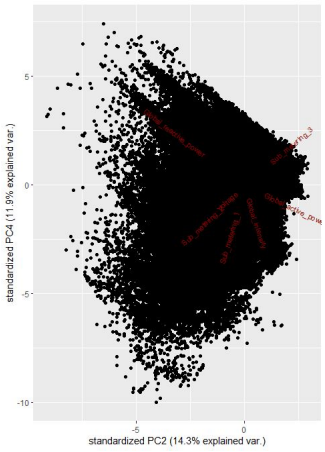


Figure: PC 2 vs 4 (14.3% : 11.9%)

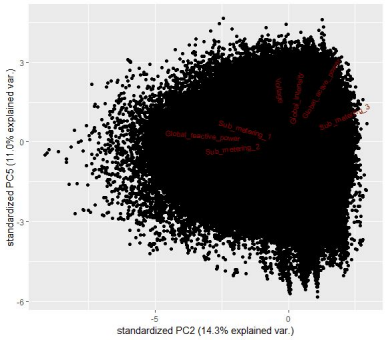


Figure: PC 2 vs 5 (14.3% : 11%)

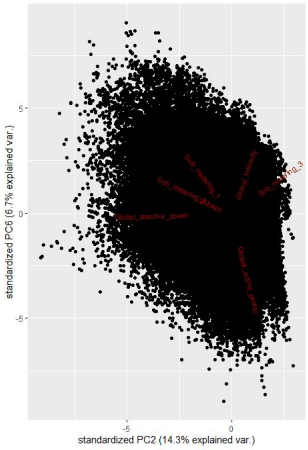


Figure: PC 2 vs 6 (14.3% : 6.7%)

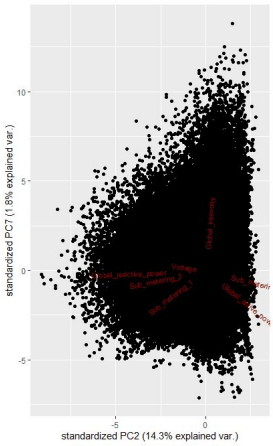


Figure: PC 2 vs 7 (14.3% : 1.8%)

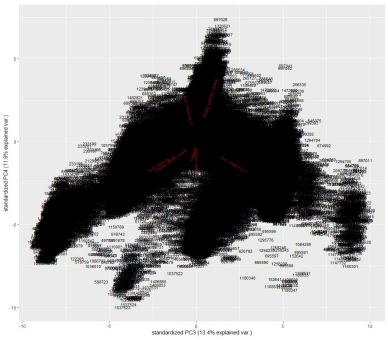


Figure: PC 3 vs 4 (13.4% : 11.9%)

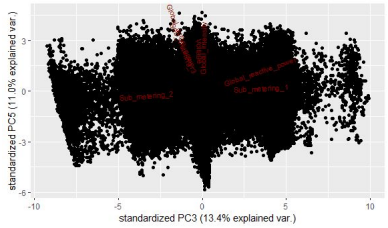
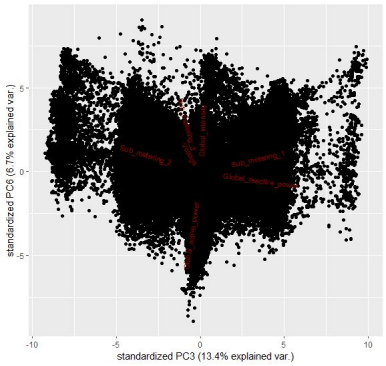
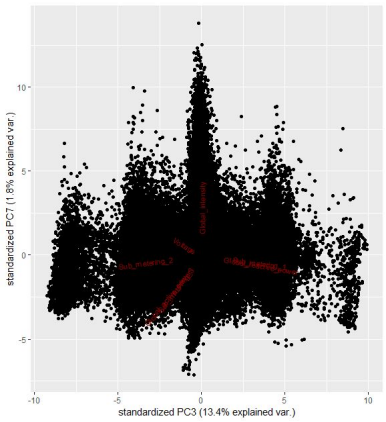
		

Figure: PC 3 vs 5 (13.4% : 11%)	Figure: PC 3 vs 6 (13.4% : 6.7%)	Figure: PC 3 vs 7 (13.4% : 1.8%)

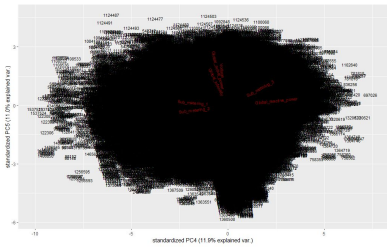
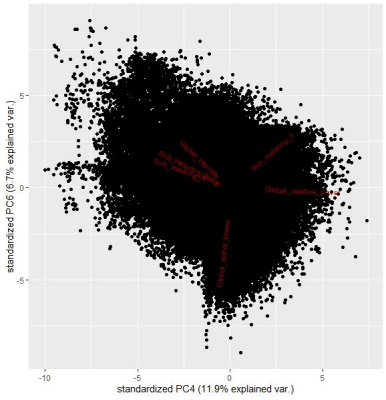
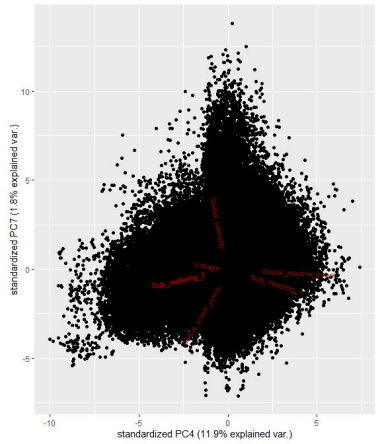

		

Figure: PC 4 vs 5 (11.9% : 11%)	Figure: PC 4 vs 6 (11.9% : 6.7%)	Figure: PC 4 vs 7 (11.9% : 1.8%)

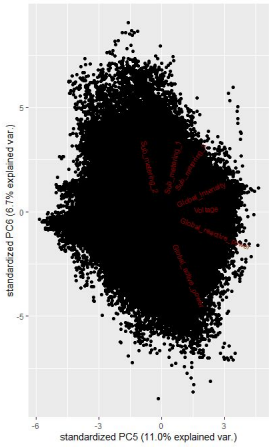
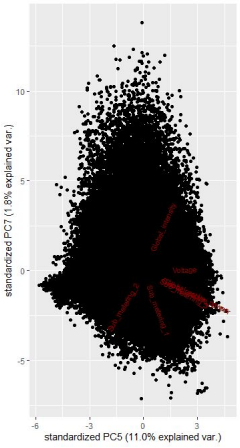
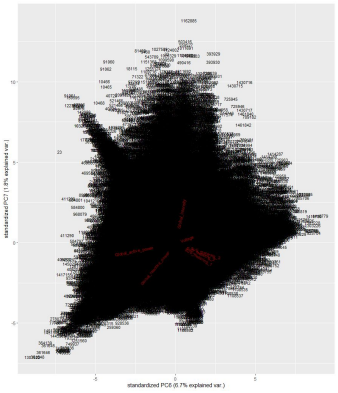
		

Figure: PC 5 vs 6 (11% : 6.7%)	Figure: PC 5 vs 7 (11% : 1.8%)	Figure: PC 6 vs 7 (6.7% : 1.8%)

Figure 2.2: Principal Component Analysis plot

This Principal Component Analysis plot makes it clear that the best Component to choose for model training would be PC 1 as it covers the most variance of the dataset.

2.3 PC for One week only

PCA method: Singular value decomposition with Imputation

Component loadings (8 dimensions in rows, 7 components in columns):

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Day 1 - 17 Sun	-0.35	-0.21	-0.14	0.33	-0.31	0.55	0.54
Day 2 - 18 Mon	-0.35	-0.05	0.39	0.26	0.15	0.34	-0.45
Day 3 - 19 Tue	-0.35	0.55	-0.34	0.28	-0.45	-0.37	-0.13
Day 4 - 20 Wed	-0.35	-0.06	0.36	0.40	0.43	-0.50	0.30
Day 5 - 21 Thu	-0.35	0.17	0.23	-0.18	-0.11	0.21	-0.43
Day 6 - 22 Fri	-0.35	0.06	0.38	-0.67	-0.25	-0.14	0.36
Day 7 - 23 Sat	-0.35	-0.73	-0.37	-0.14	-0.10	-0.30	-0.28
Day 8 - 24 Sun	-0.35	0.27	-0.50	-0.29	0.64	0.20	0.09

Figure 2.3.1: Component loadings (8 dimensions in rows, 7 components in columns)

Principal components (7 data points in rows, 7 components in columns):

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Global Active Power	1.12	0.00	0.00	-0.01	-0.00	-0.00	-0.00
Global Reactive Power	1.17	0.03	0.00	-0.01	0.01	0.00	-0.00
Voltage	-6.41	0.00	-0.00	-0.00	0.00	-0.00	0.00
Global Intensity	0.94	-0.06	-0.04	0.00	0.00	0.00	0.00
Sub-Metering 1	1.14	0.01	0.01	-0.02	-0.01	0.00	0.00
Sub-Metering 2	1.11	0.06	-0.01	0.02	-0.00	0.00	0.00
Sub-Metering 3	0.93	-0.04	0.04	0.01	0.00	0.00	0.00

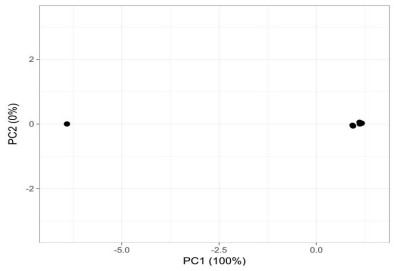
Figure 2.3.2: Principle components (7 dimensions in rows, 7 components in columns)

Variance explained by principal components (7 components):

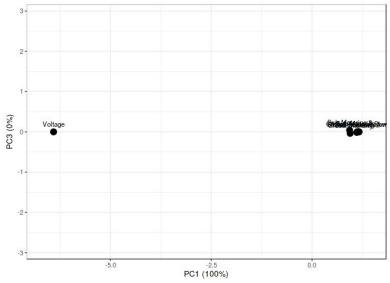
	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Individual	1.00	0.00	0.00	0.00	0.00	0.00	0.00
Cumulative	1.00	1.00	1.00	1.00	1.00	1.00	1.00

Figure 2.3.3: Variance explained by the principal (components)

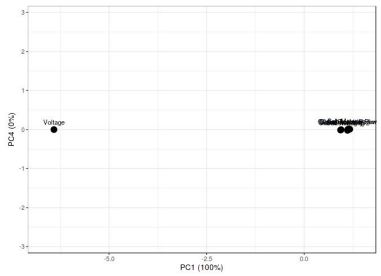
One Week's Worth of PC Plots:



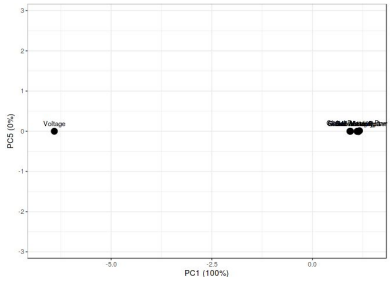
PC1-2



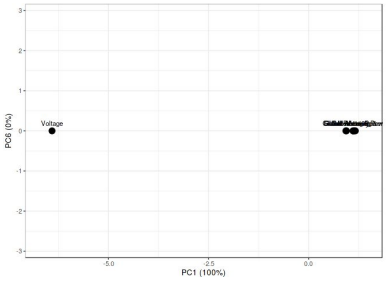
PC1-3



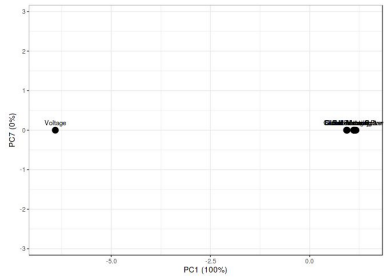
PC1-4



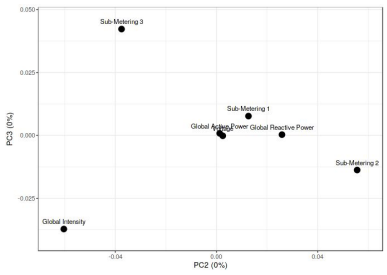
PC1-5



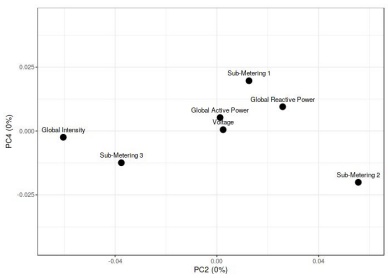
PC1-6



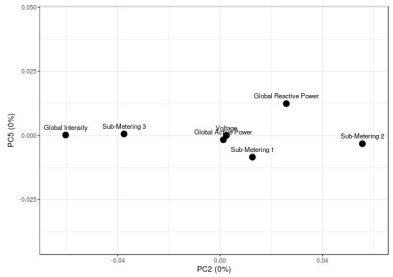
PC1-7



PC2-3



PC2-4



PC2-5

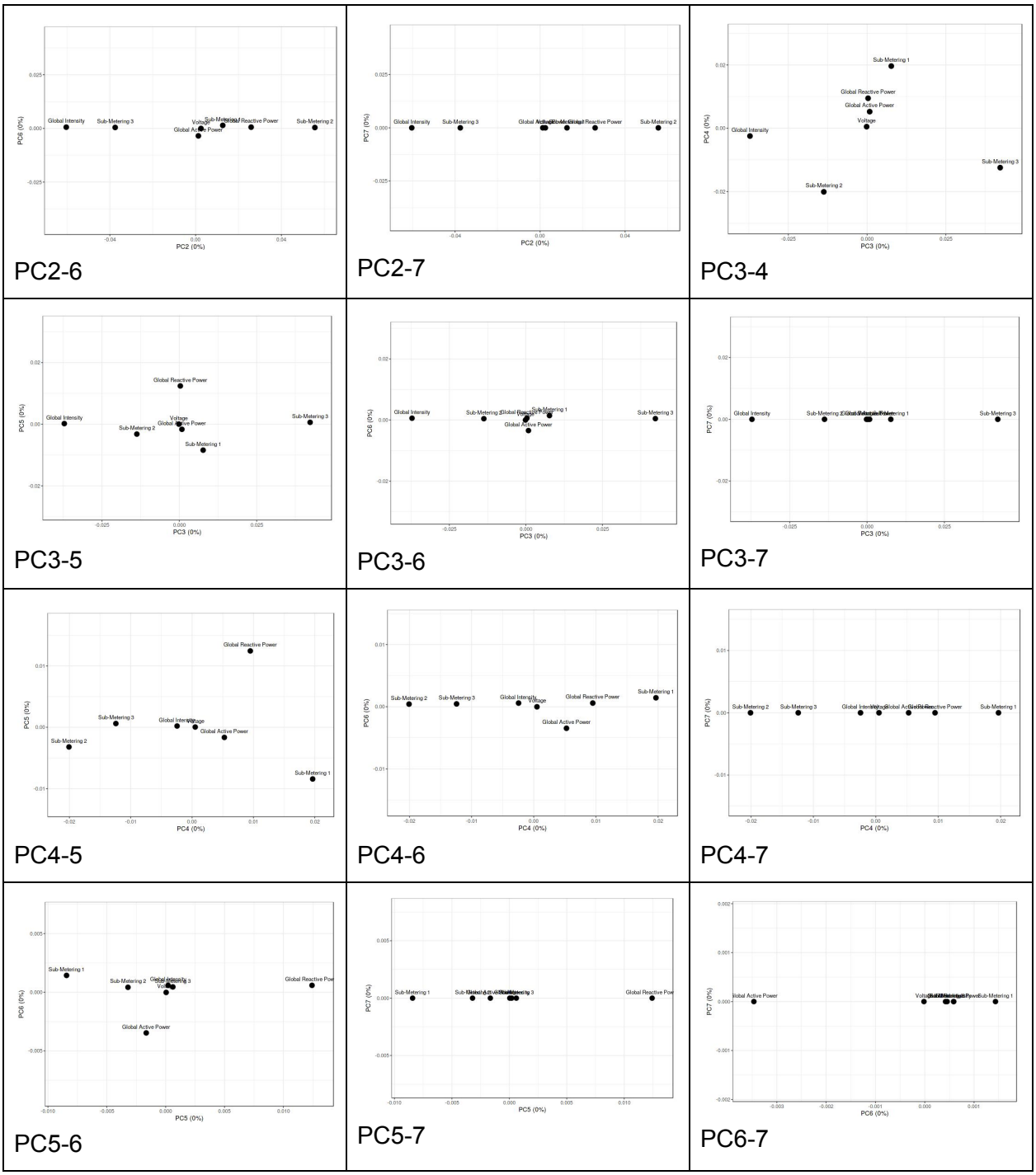


Figure 2.3.4: PC plots over a span of a week

2.4 Heat Map

Rows are centered; unit variance scaling is applied to rows. Both rows and columns are clustered using correlation distance and average linkage. 8 rows, 7 columns.

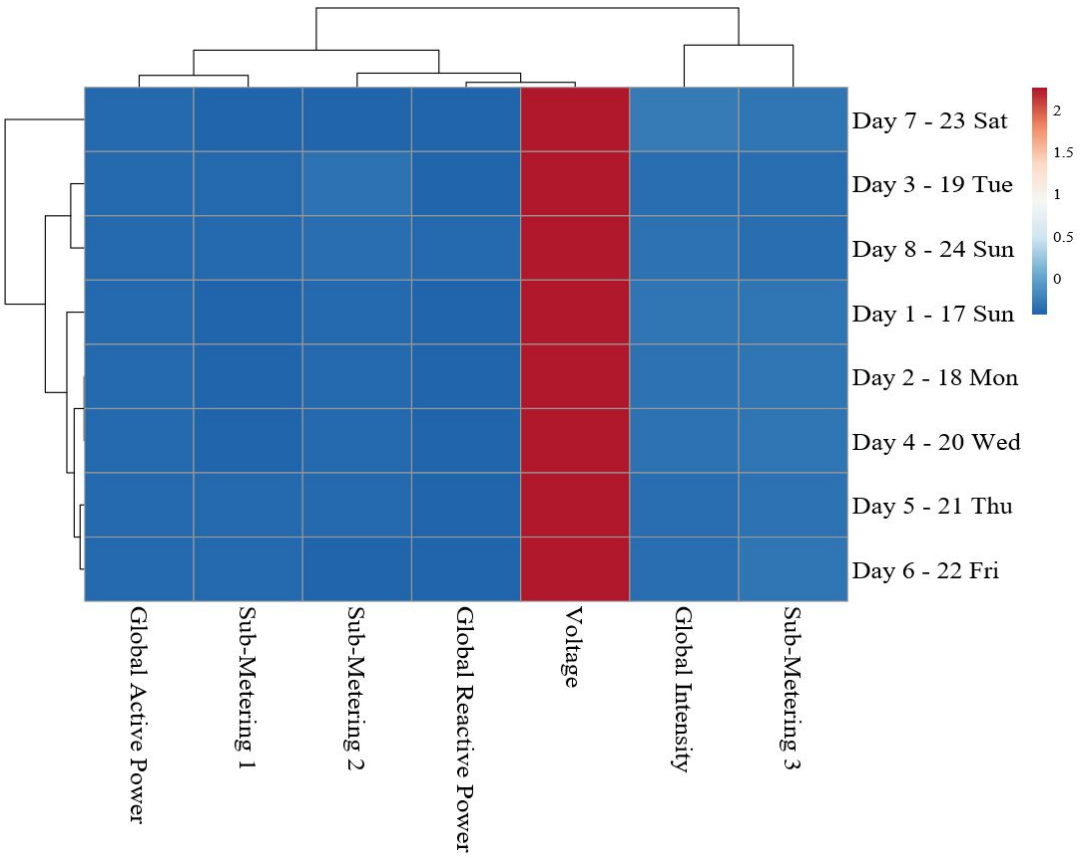


Figure 2.4: Heat Map

3 Hidden Markov Models (HMM) and Log Likelihood

A final approach our group considered was to make use of HMM's. Tools provided by the depmixS4 package were used in order to see if our HMM was applicable to the test data. For this course project, we were tasked with partitioning the original dataset into a training and testing dataset. The training dataset was composed of two years worth of data, namely January 1st, 2007 to December 31st, 2008, while the test dataset was made up of data from January 1st, 2009 to December 1st, 2009. This splits the original data set at about 70% training and 30% testing. Tools provided by the depmixS4 package were used in order to see if our HMM was applicable to the test data. On our first pass through training the test data, it was discovered that these datasets included N/A values. To work around this, we subbed in values for these N/A entries.

In the beginning, all of the models were trained exclusively using the gaussian family. This resulted in a surprisingly high number of feature combinations that resulted in a positive log-likelihood no matter the number of states. An attempt was made to correct this by training the models using the multinomial family. Upon making this switch, we experienced a persistent memory error for each of the different models, and R Studio could not allocate enough space for the depmix vectors and thus, could not complete the fitting process. It was later revealed that this memory error was due to the extremely large test data set that we had chosen. Instead of our choice of a 70% - 30% split, in hindsight, we should have instead picked a single time window and repeated it across

the two-year span. For this reason, we had to use our results from the gaussian family models

Running each different combination of features we tested from 4-10 states we determined that global active power using 8 states was our best univariate model, and global reactive power and global intensity with 6 states was our best multivariate model. Global intensity, voltage, global active power, and global reactive power were features chosen for univariate models. While global reactive power did not converge for any combination of states that were tested, global active power was found to have the most reasonable log-likelihood and BIC values. Furthermore, global intensity and voltage likelihood values were significantly lower in value in comparison to global active power, and for that reason was not chosen. When dealing with multivariate models, the combinations of the features chosen were:

- Global reactive power and global active power
- Global reactive power and global intensity
- Voltage and global intensity
- Global active power and global intensity
- Global active power, global reactive power, and global intensity

We found that global reactive power and global intensity provided the best values for log-likelihood and BIC that were tested, while also having very high positive

log-likelihood values. Global active power and global intensity had much more negative log-likelihood values for all states than global reactive power and global intensity did. Our three feature models did converge as well but it's log-likelihood values were much more negative than that of the GRP and INT two feature models.

Constricting the data sets further by using a time window of 12:00 pm to 3:00 pm every day for a span of 2 years gave us a negative log-likelihood value for our test and training data. This was a result of using the multivariate model with the feature of global reactive power and 4 states. We concluded that the univariate model surrounding global intensity with 6 states was our best choice. This model gave us negative log-likelihood values for the test and training. Unfortunately, due to time constraints, not all possible combinations of features and states will be tested. While we are on the right track compared to our first attempts, our most successful results being passed on to part 4 might not be our best results overall.

	Training Results		Testing Results	
Model	Log-Likelihood	BIC	Log-Likelihood	BIC
Type: Univariate Feature: Global Active Power States: eight Family: Gaussian	-98840.4	198776.76	96136.99	107668.7
Type: Multivariate Feature: Global Reactive Power and Global Intensity States: six Family: Gaussian, Gaussian	-66931.22	134680.59	165922.1	-81399.81

Figure 3.1: Table using 2-year span of data

	Training Results		Testing Results	
Model	Log Likelihood	BIC	Log Likelihood	BIC
Type: Univariate Feature: Intensity States: six Family: Gaussian	-144687.5	252950.32	-44124.95	120331.2
Type: Multivariate Feature: Global Reactive Power and Global Intensity States: four Family: Gaussian, Gaussian	-58359.3	117083.83	-3329.521	31382.46

Figure 3.2: Table using 2 years of data & a three hour time window from 12 pm - 3 pm

4 Anomaly Detection

4.1 Moving Average Method

Three different types of anomalies have been analyzed in our research: First, anomalies over the entire data set. These are data points far from the overall mean by three or more standard deviations in both the positive and negative directions. Second, two types of anomalies were being searched for using the Moving Average window. The average window method was used in order to detect anomalies that might not be considered outliers in the overall data set, but given the window, they do not conform to the window's expected behavior. The window size is the one defined in the first part of the report which is 200 data points of length. The window starts at the first point in the data set and gradually moves until the end of the set. At each iteration, it checks whether each value is far from the window mean by three or more window's standard deviations in both the positive and negative directions. Additionally, it checks whether the window average is far from the average of the maximum and minimum of the window by three or more window's standard deviations in both the positive and negative directions. In conclusion, we have defined three different thresholds in order to distinguish the different types of anomalies in the given test data sets. The following graphs will show a comparison between the three different types of anomaly detection thresholds for the same weekday and weekend day chosen from the first week of all the five given test data sets. In the graphs, anomalies are being indicated using red points.

Test Table 1

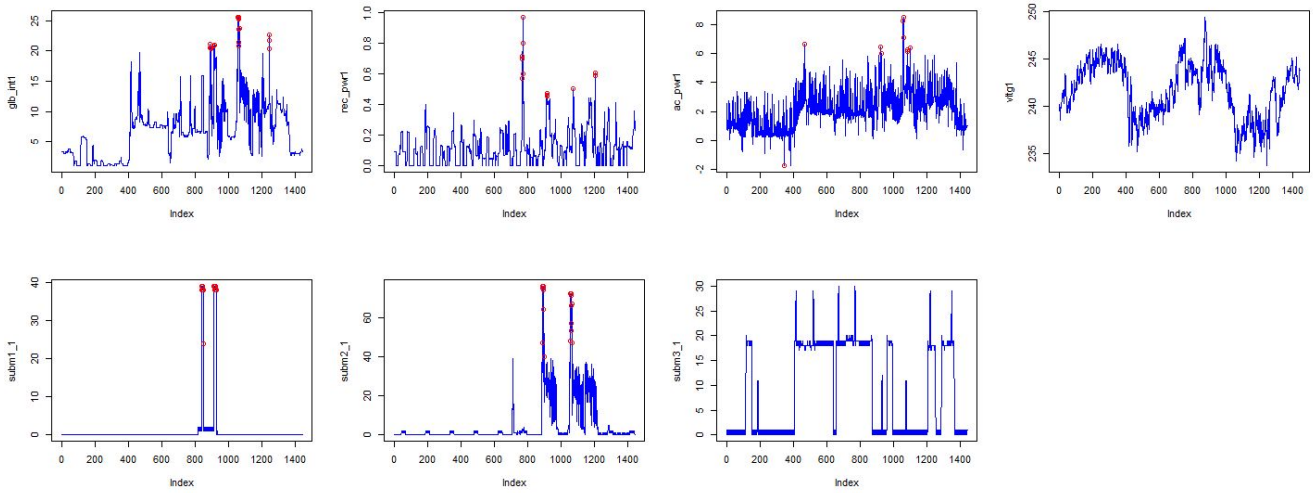


Figure 4.1.1: Table 1 weekday overall anomalies

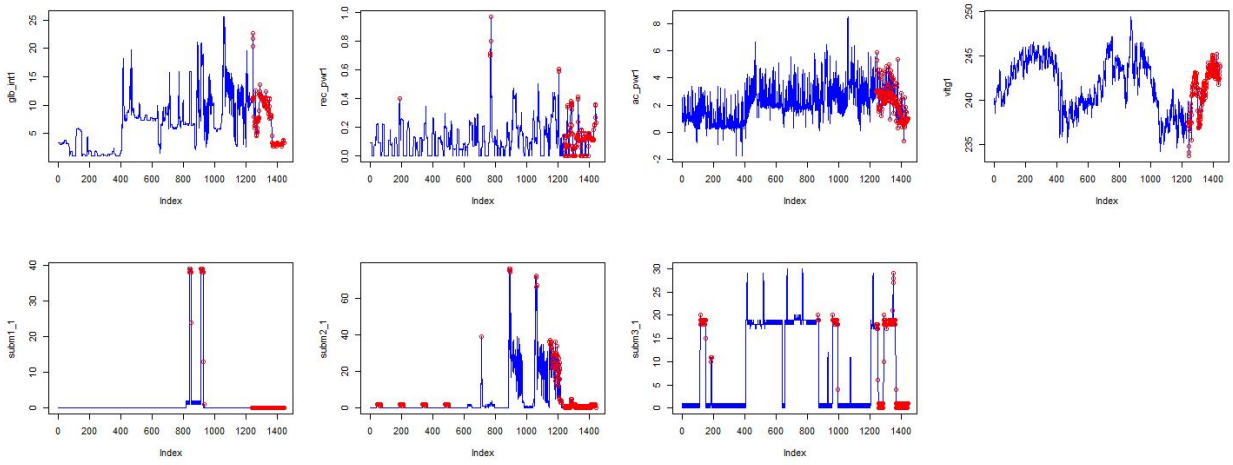


Figure 4.1.2: Table 1 weekday window anomalies (1)

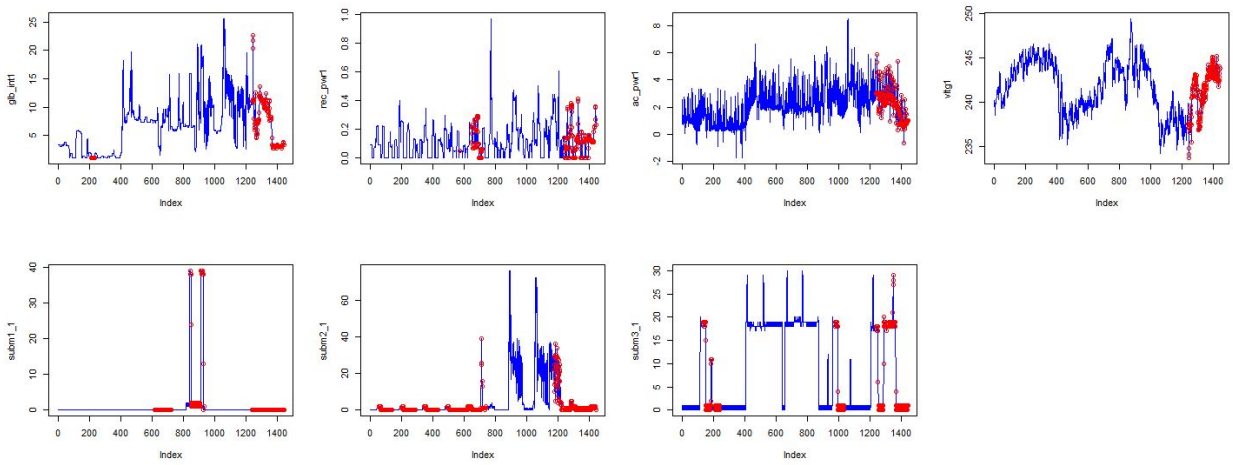


Figure 4.1.3: Table 1 weekday window anomalies (2)

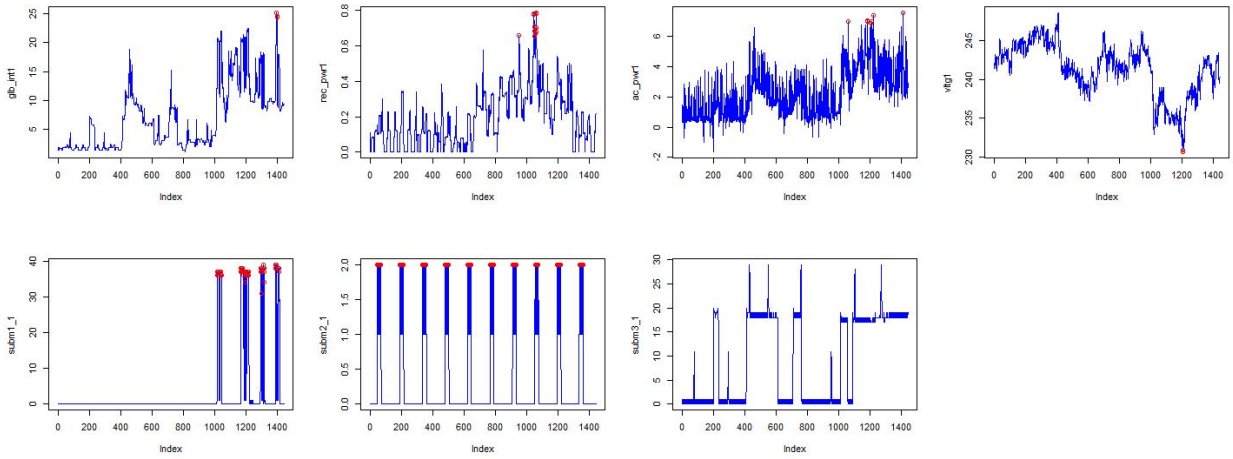


Figure 4.1.4: Table 1 weekend overall anomalies

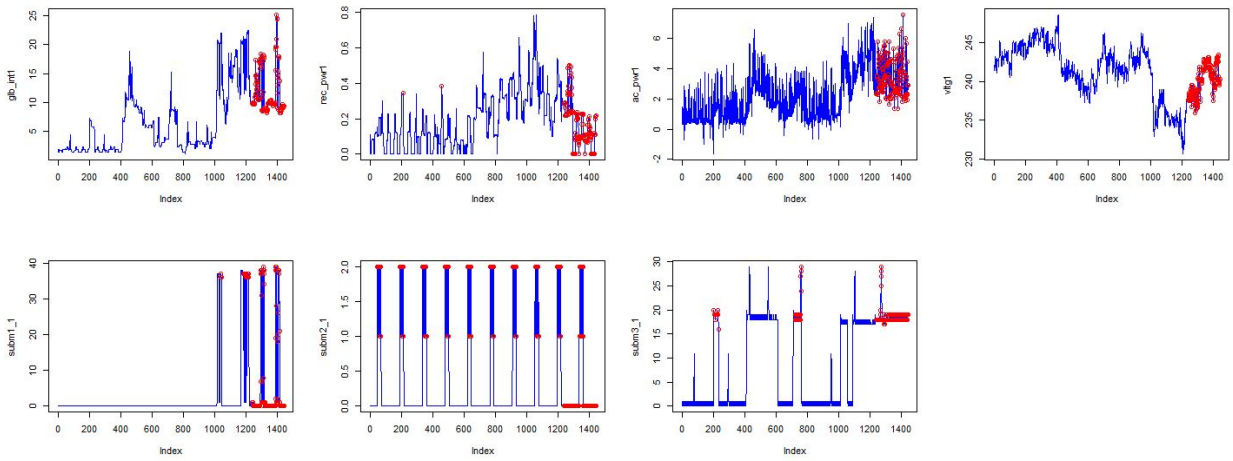


Figure 4.1.5: Table 1 weekend window anomalies (1)

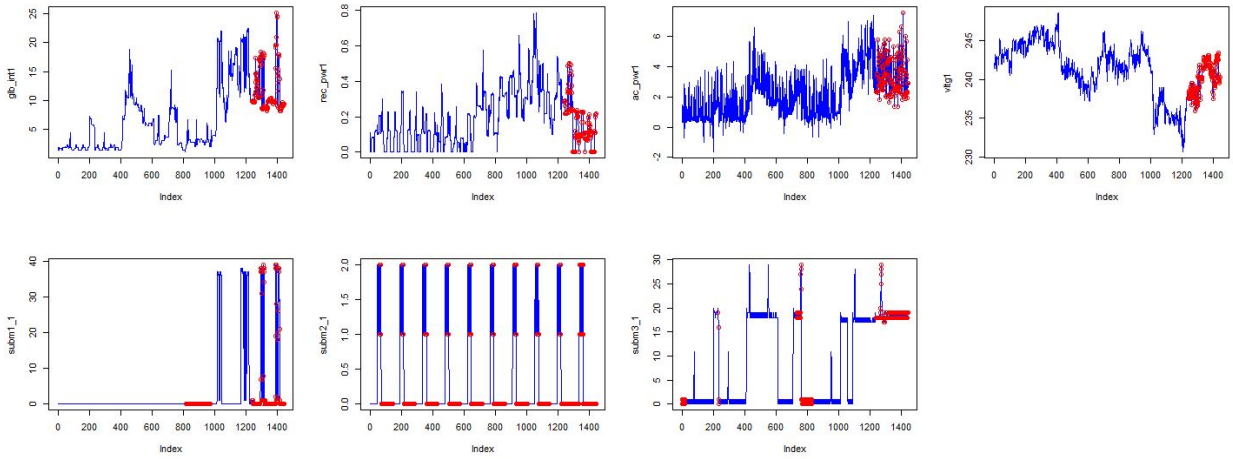


Figure 4.1.6: Table 1 weekend window anomalies (2)

Test Table 2

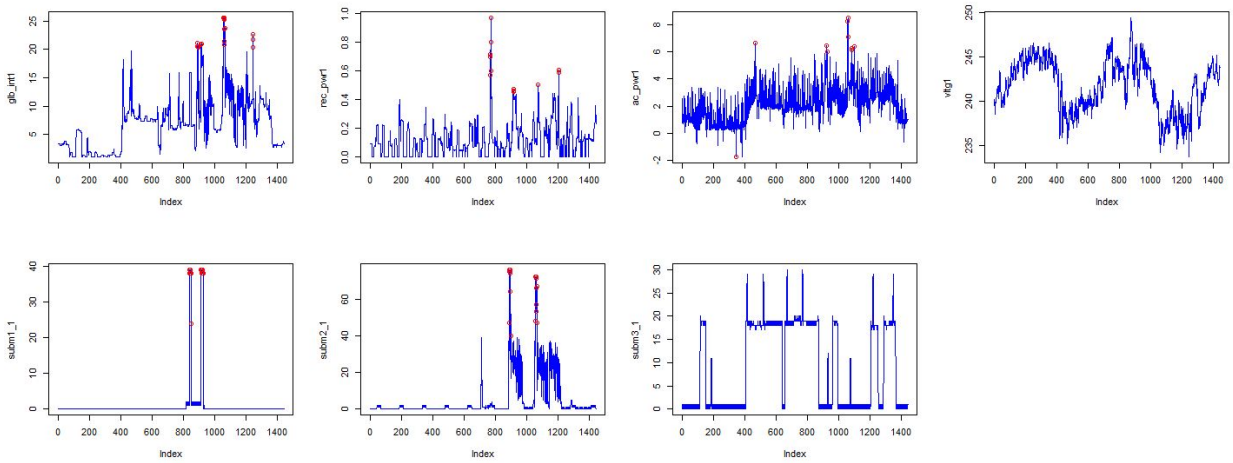


Figure 4.2.1: Table 2 weekday overall anomalies

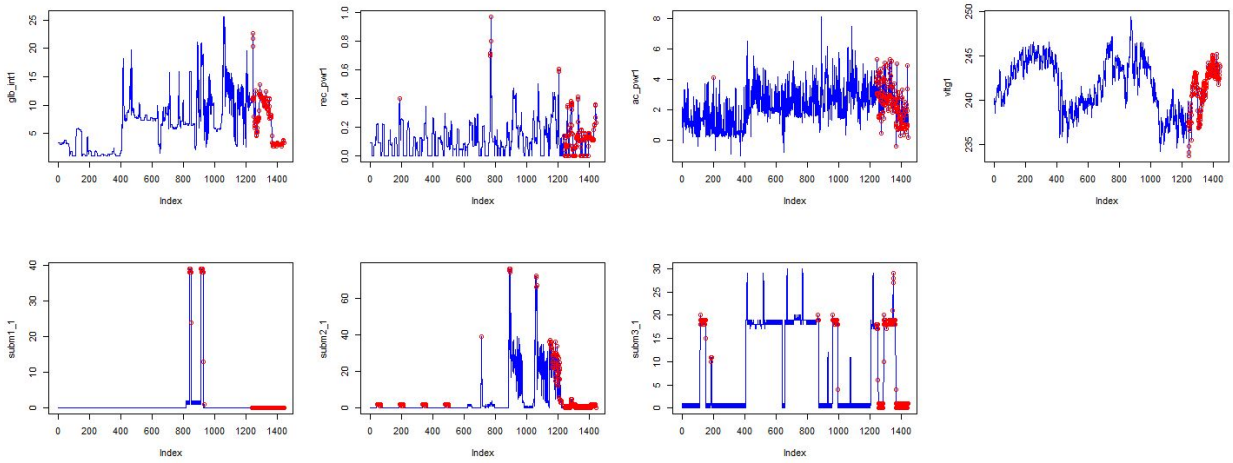


Figure 4.2.2: Table 2 weekday window anomalies (1)

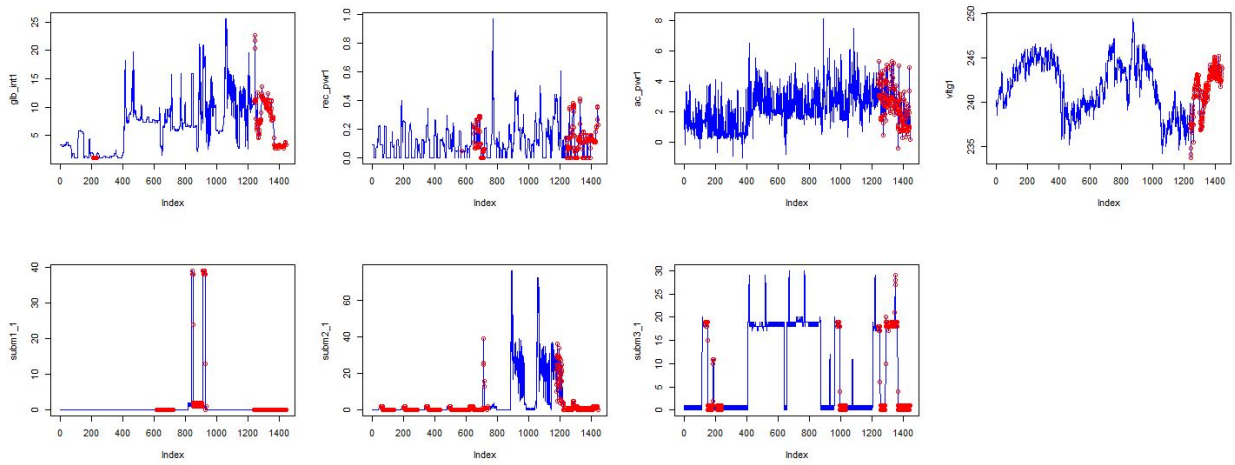


Figure 4.2.3: Table 2 weekday window anomalies (2)

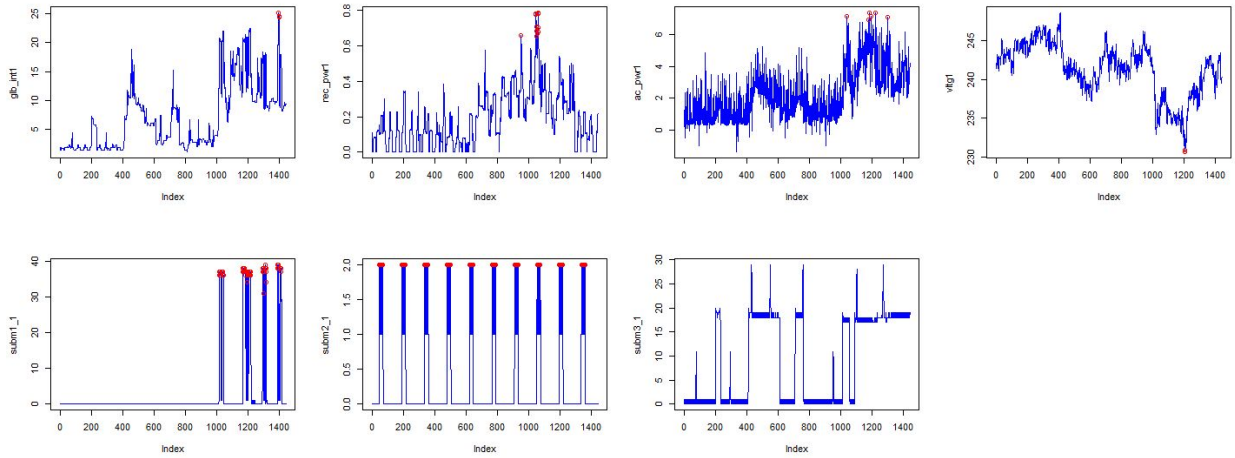


Figure 4.2.4: Table 2 weekend overall anomalies

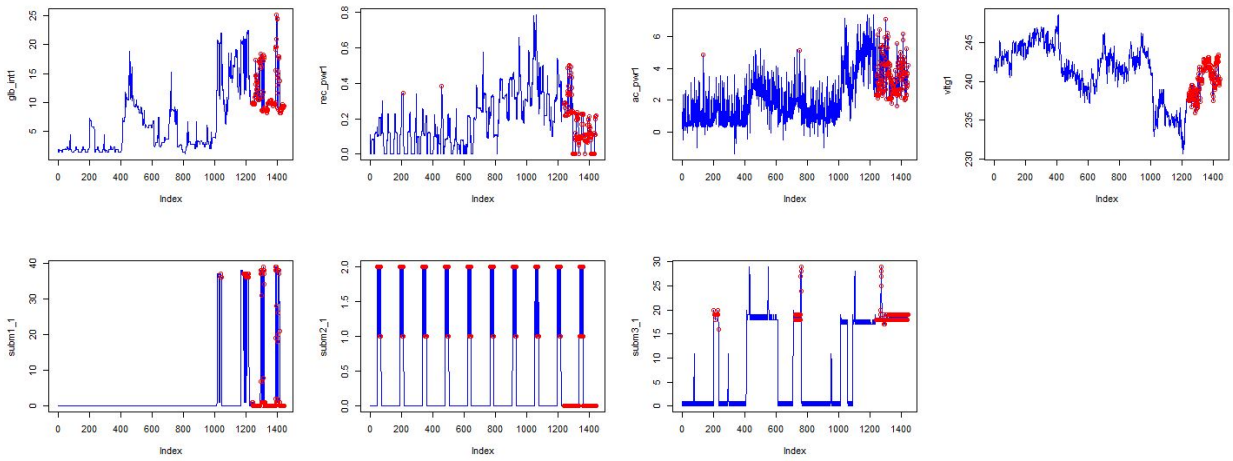


Figure 4.2.5: Table 2 weekend window anomalies (1)

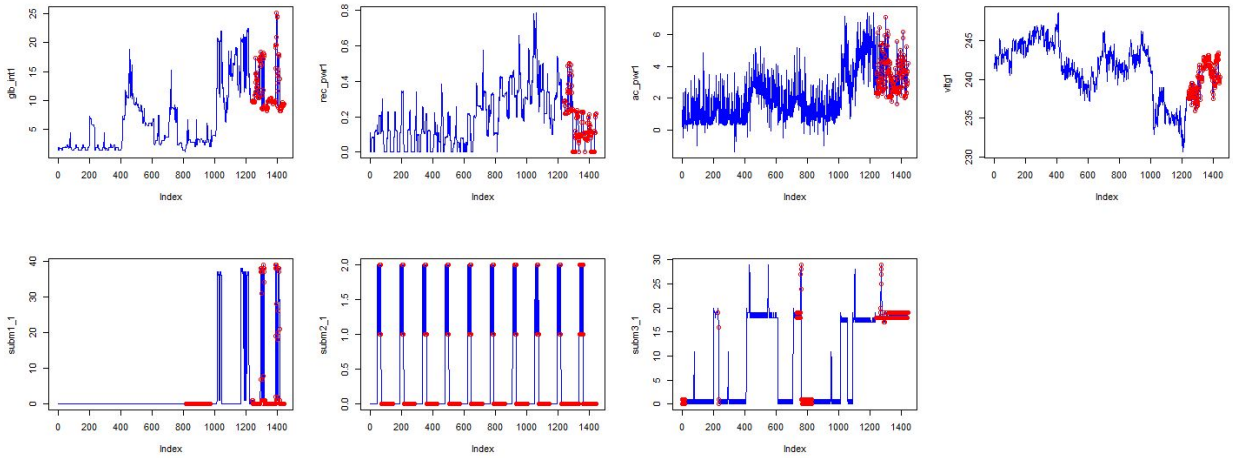


Figure 4.2.6: Table 2 weekend window anomalies (2)

Test Table 3

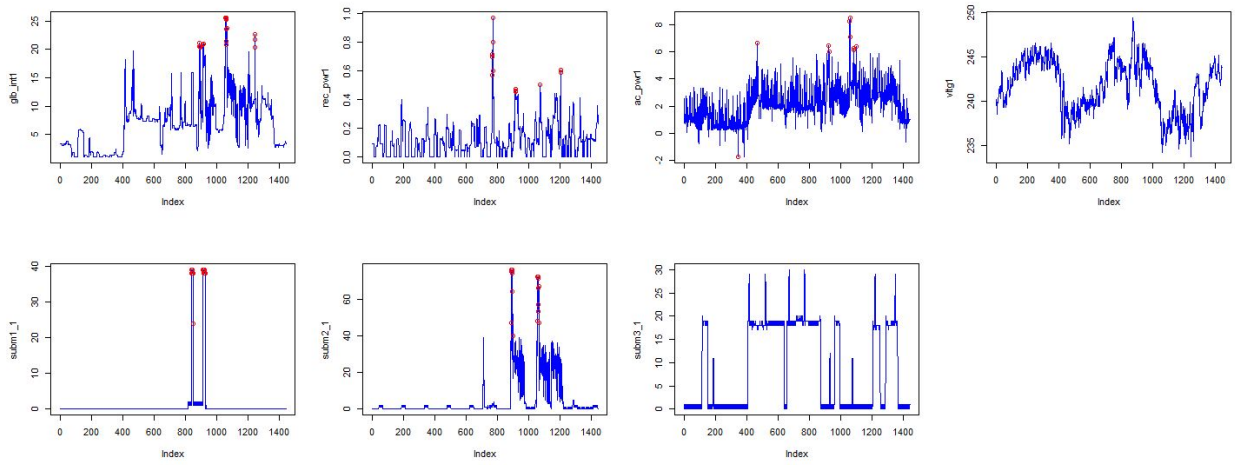


Figure 4.3.1: Table 3 weekday overall anomalies

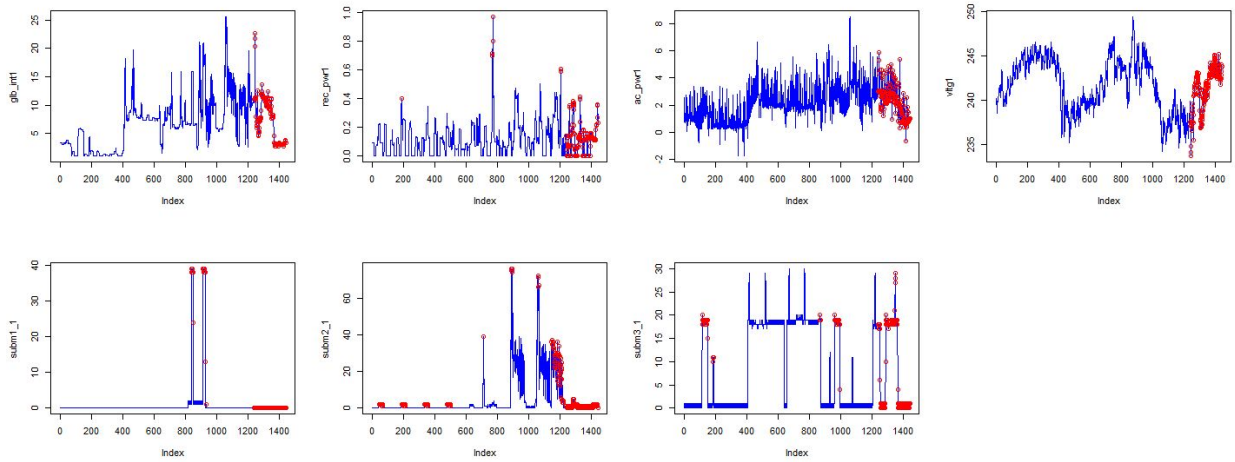


Figure 4.3.2: Table 3 weekday window anomalies (1)

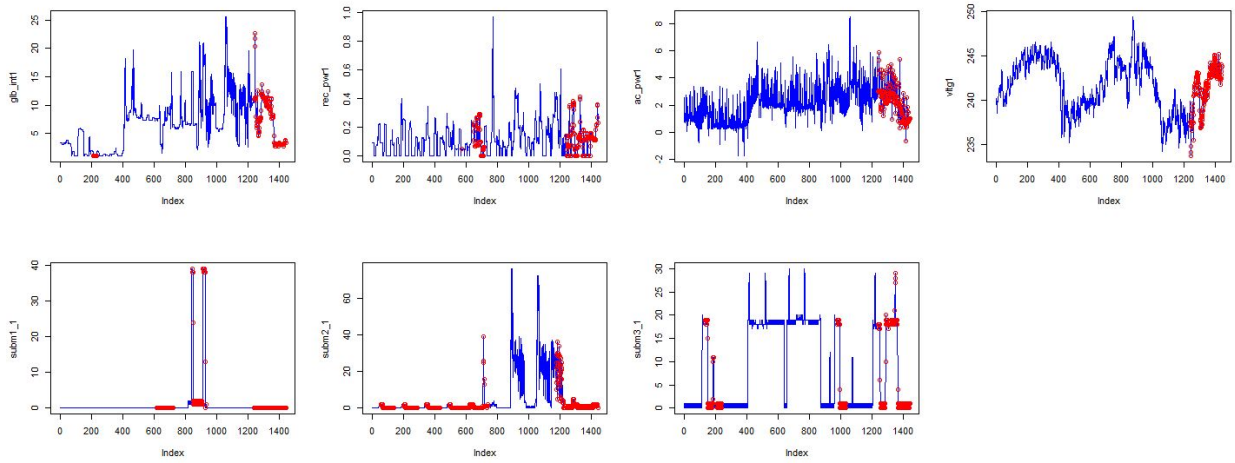


Figure 4.3.3: Table 3 weekday window anomalies (2)

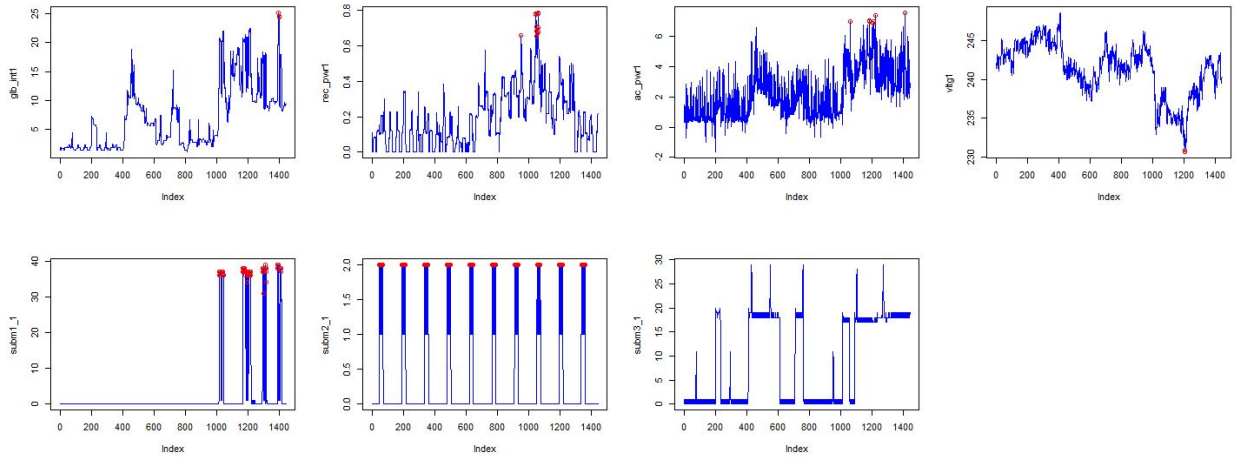


Figure 4.3.4: Table 3 weekend overall anomalies

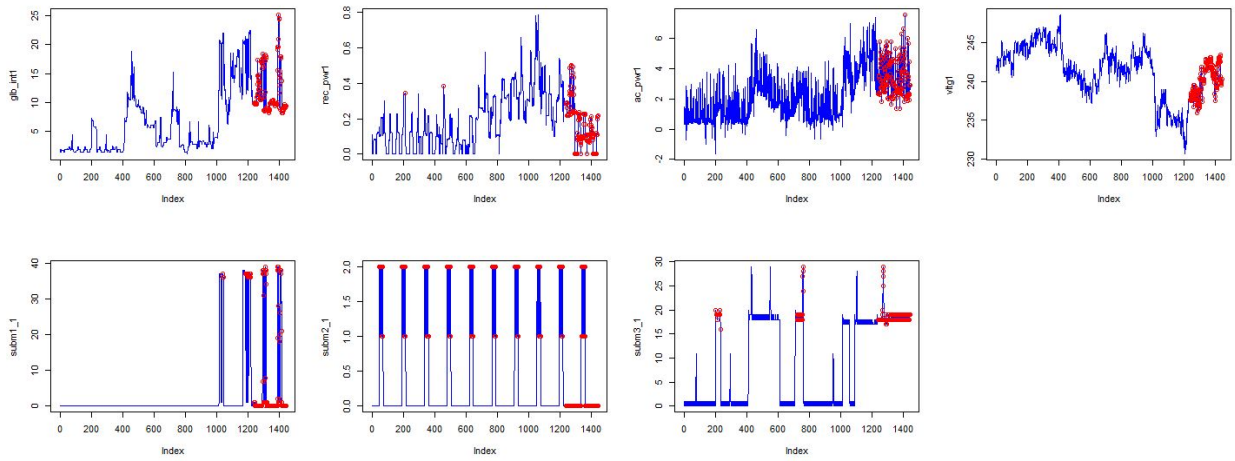


Figure 4.3.5: Table 3 weekend window anomalies (1)

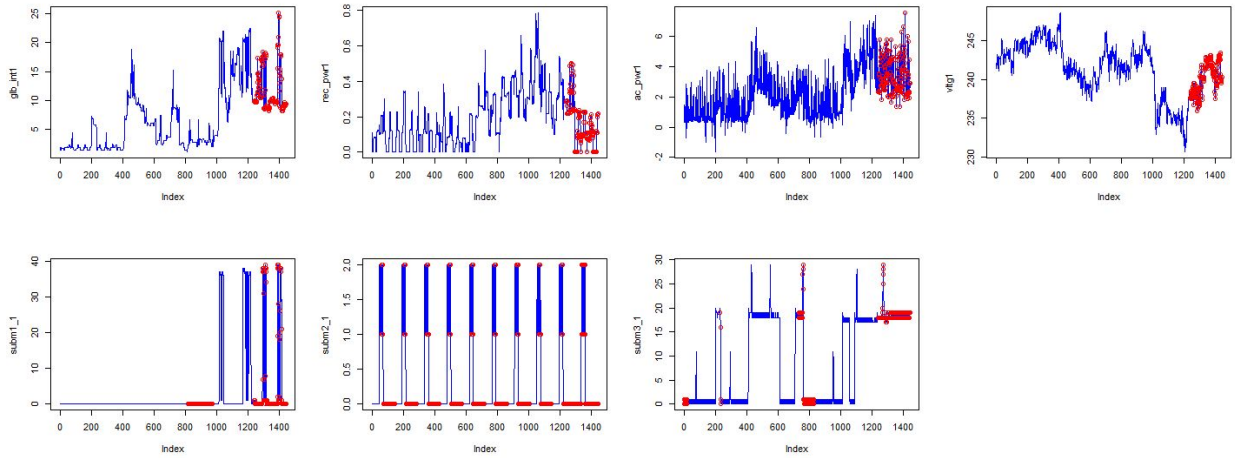


Figure 4.3.6: Table 3 weekend window anomalies (2)

Test Table 4

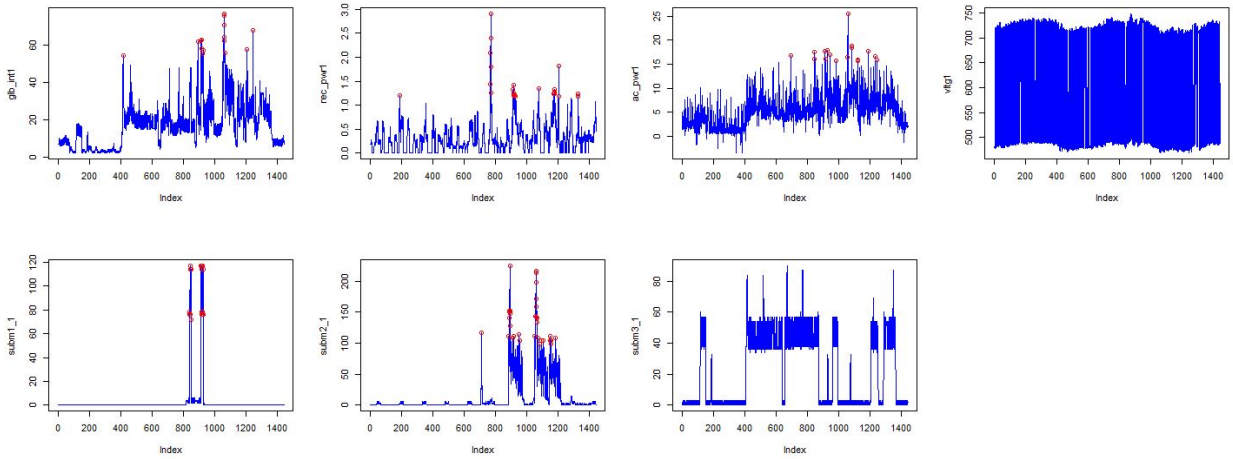


Figure 4.4.1: Table 4 weekday overall anomalies

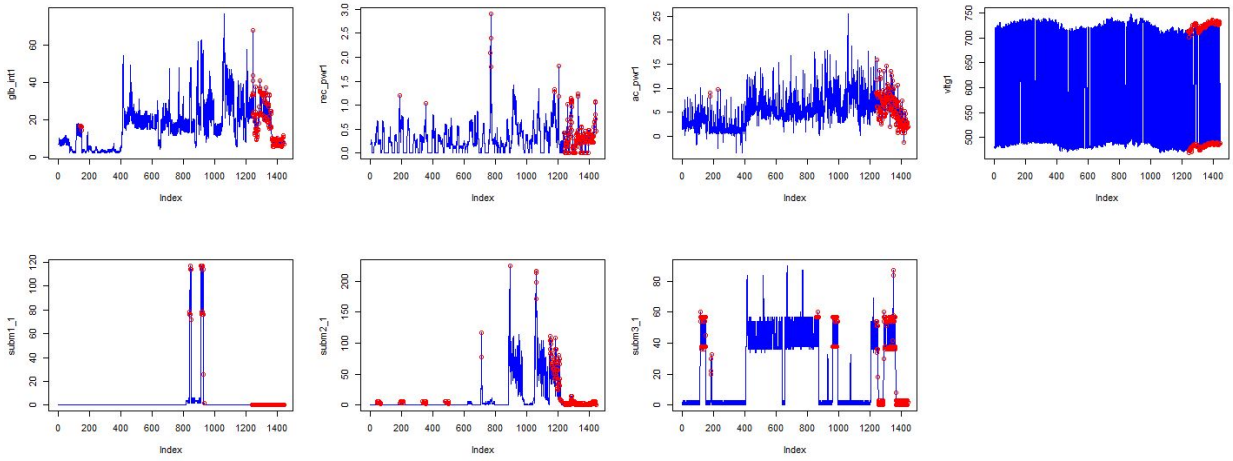


Figure 4.4.2: Table 4 weekday window anomalies (1)

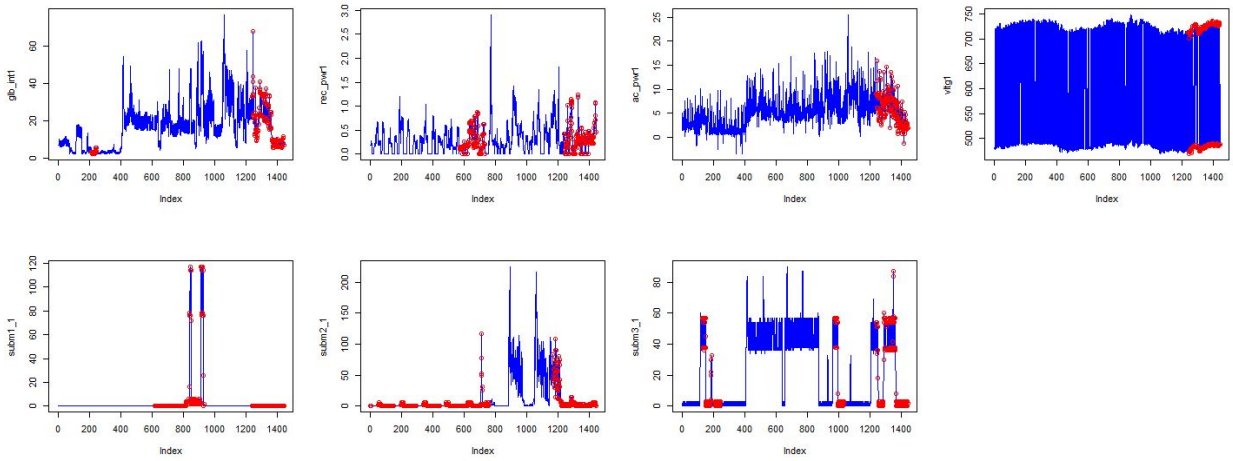


Figure 4.4.3: Table 4 weekday window anomalies (2)

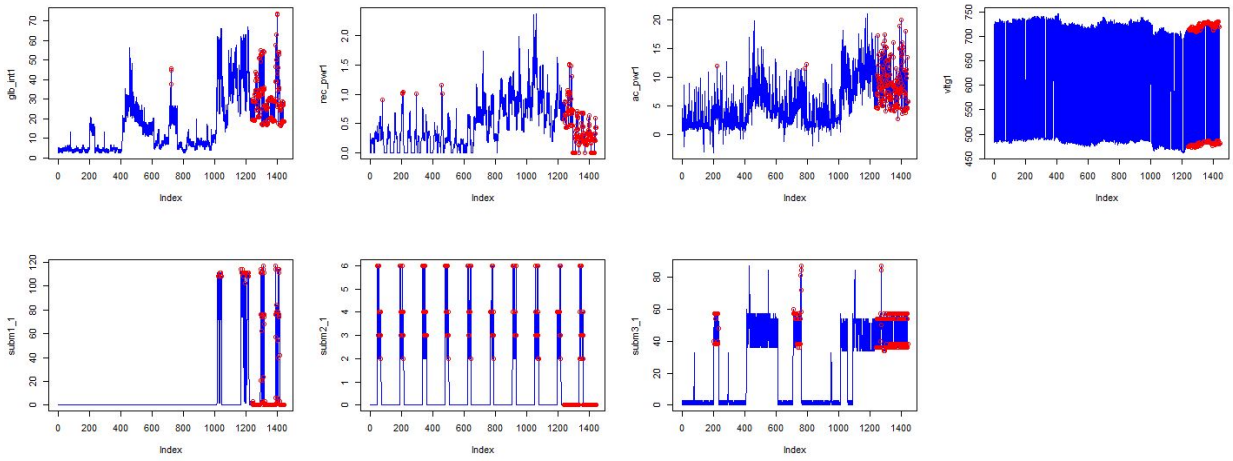


Figure 4.4.4: Table 4 weekend overall anomalies

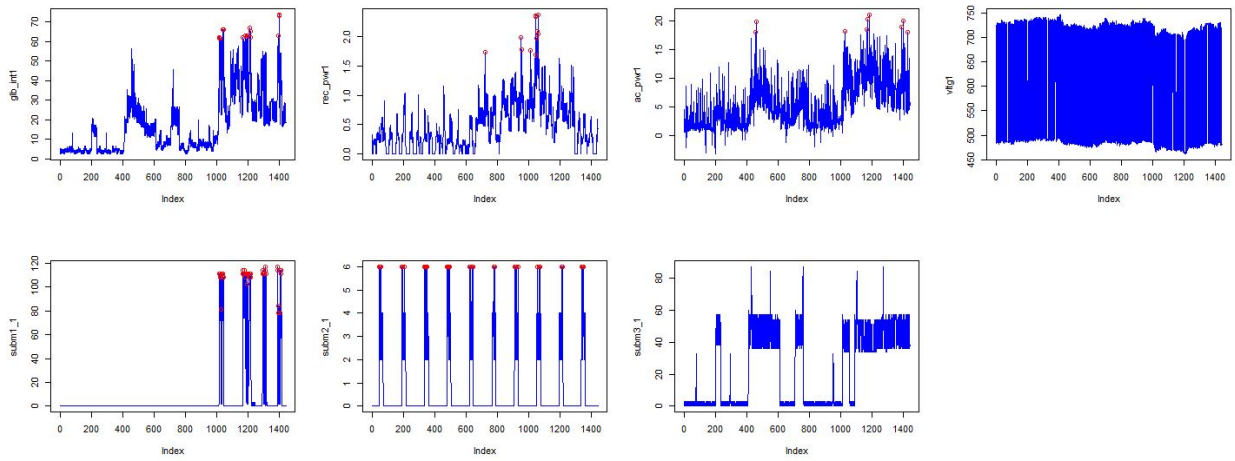


Figure 4.4.5: Table 4 weekend window anomalies (1)

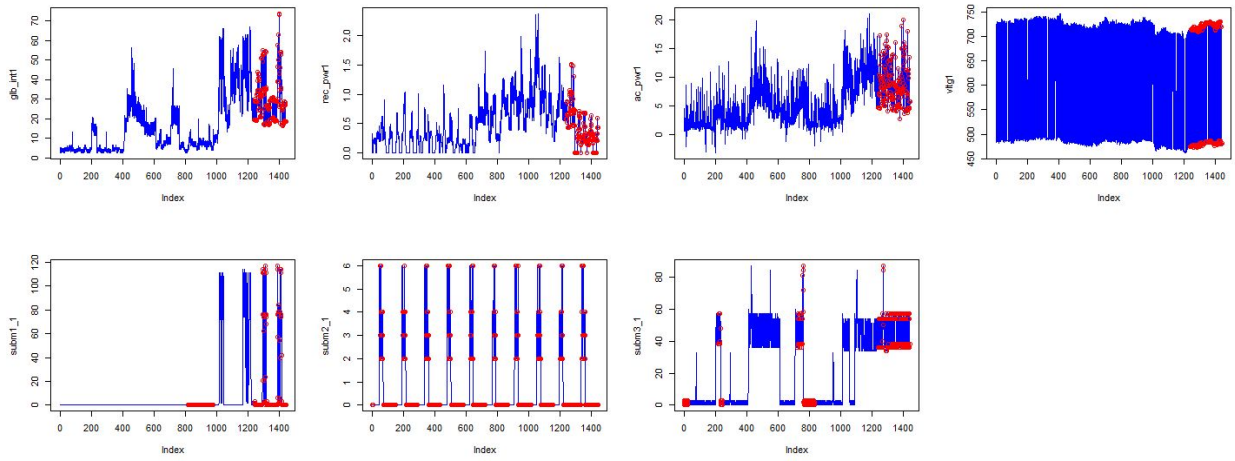


Figure 4.4.6: Table 4 weekend window anomalies (2)

Test Table 5

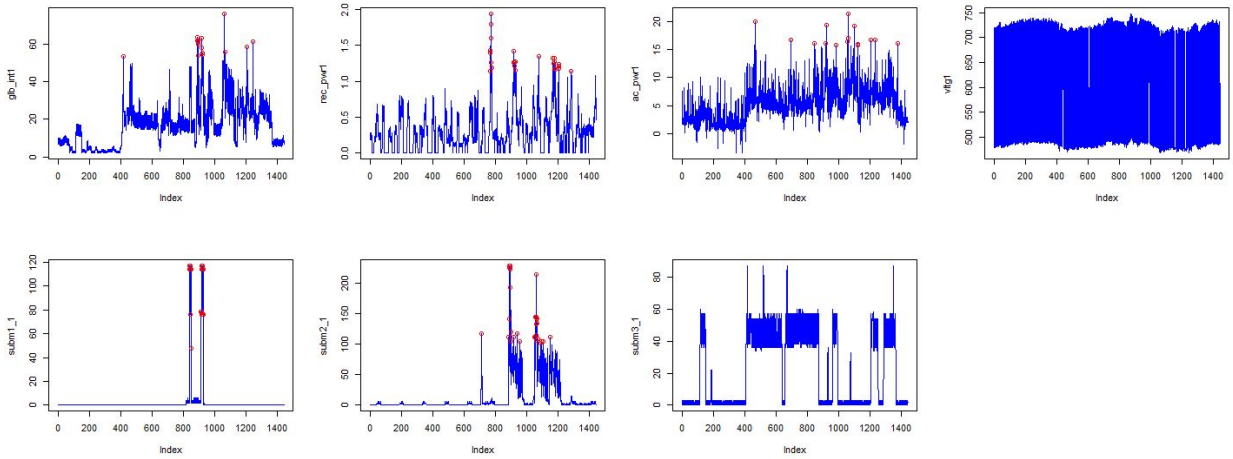


Figure 4.5.1: Table 5 weekday overall anomalies

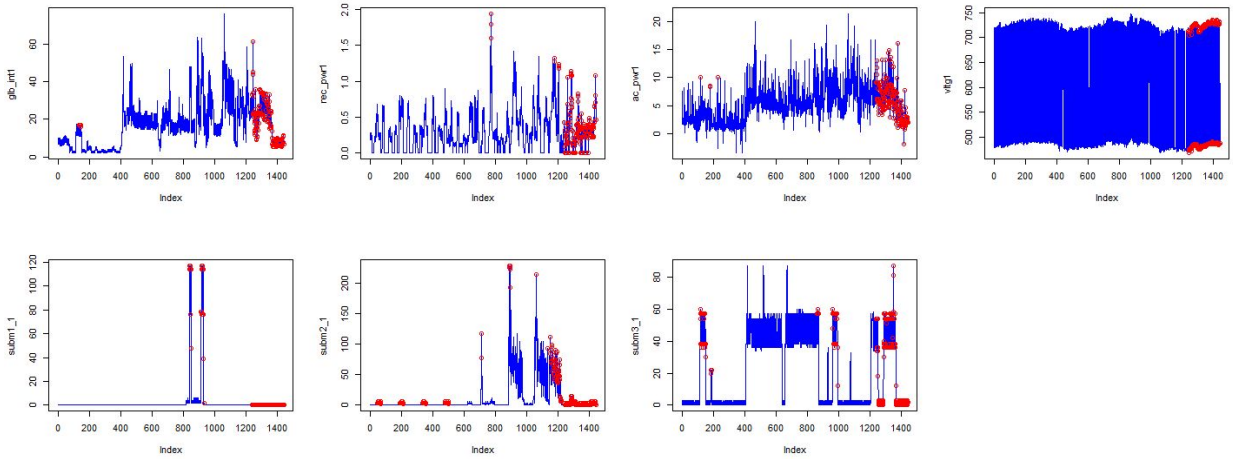


Figure 4.5.2: Table 5 weekday window anomalies (1)

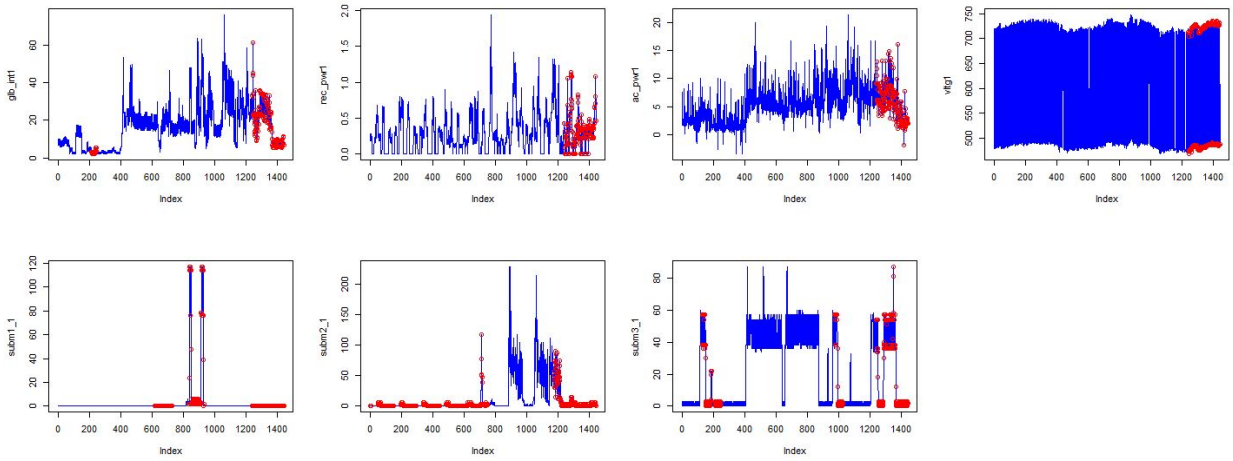


Figure 4.5.3: Table 5 weekday window anomalies (2)

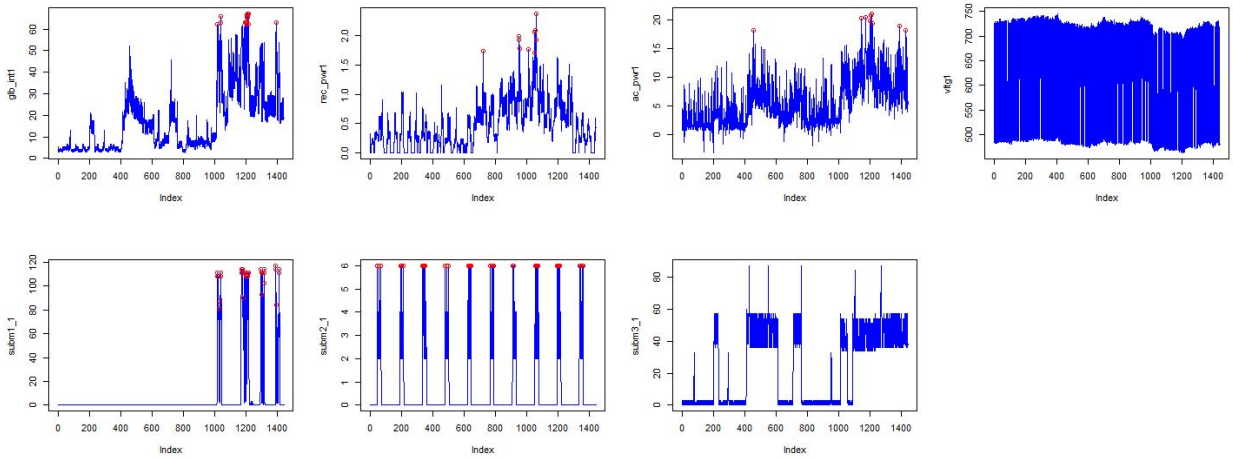


Figure 4.5.4: Table 5 weekend overall anomalies

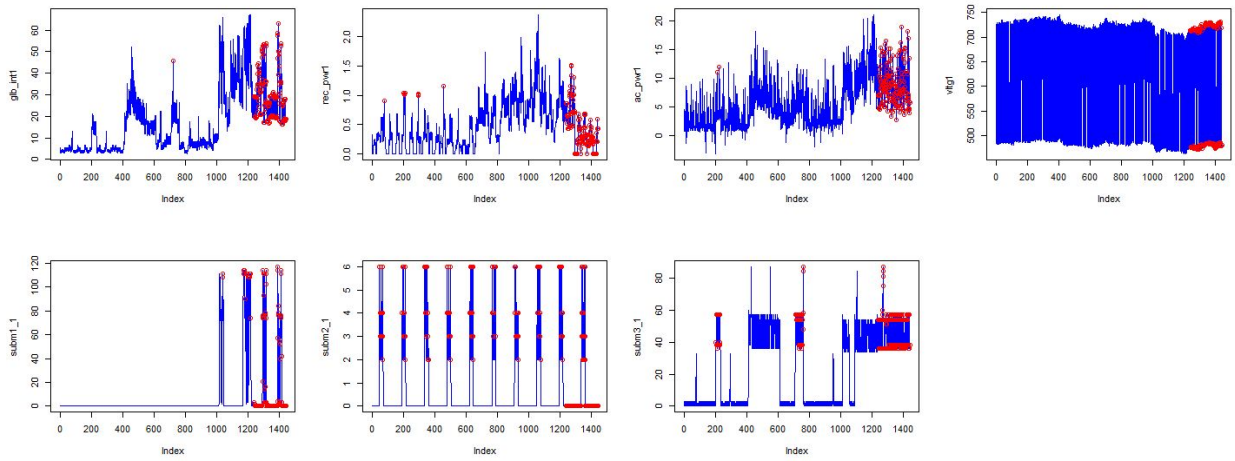


Figure 4.5.5: Table 5 weekend window anomalies (1)

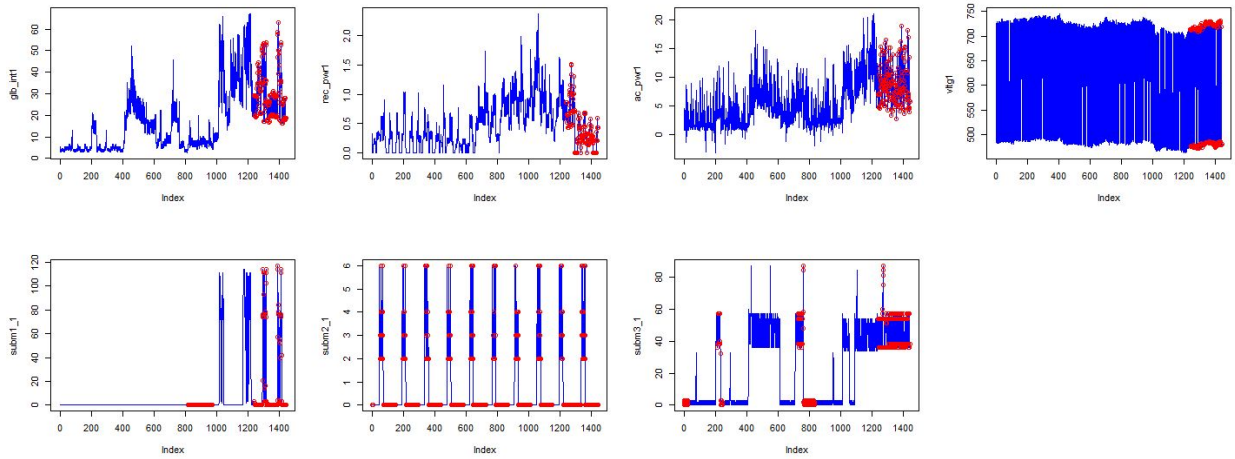


Figure 4.5.6: Table 5 weekend window anomalies (1)

4.2 Log-likelihood for observation sequence

Our second method of anomaly detection was to use our best univariate and multivariate HMM models to detect how “anomalous” each test data set is. For this purpose, each data set was first cleaned from NA values. Afterward, we took a subset of each test data set that includes all the occurrences of the time window identified in section 1.3 which is from [3:20 AM, 6:40 AM]. Using the functions setpars( ) and getpars( ) from the library “depmix”, we fitted each subset with the best univariate and best multivariate HMM that were the results of our previous training and testing. The following are the BIC and Loglike values per fitting each model:

Test Data Set	BIC	Loglike
1	739.6628	75.53099
2	739.6628	75.53099
3	739.6628	75.53099
4	5226.266	-1981.964
5	5304.966	-2023.236

Figure 4.6 - Best Univariate Model BIC and Loglike values

Test Data Set	BIC	Loglike
1	-2282.935	1488.304
2	-2282.935	1488.304
3	-2282.935	1488.304
4	6542.151	-2862.998
5	6373.344	-2778.934

Figure 4.6 - Best Multivariate Model BIC and Loglike values

4.3 Anomaly Detection Results Summary

Based on the previous results of testing with HMM, we can see that the best trained HMM models (both univariate and multivariate) have a much better fit over the fourth and fifth datasets given they have the lower (and only negative) loglike values. On the other hand, the first three data sets have positive loglike values which means our HMM has a very bad fit over these data sets. Given these results, we can conclude that the first three data sets are much more “anomalous” than the remaining and thus, they indicate a possible intrusion.

5 Conclusion

Identifying normal patterns in data is a complex process that requires a well-structured data set, well-trained models and availability of large enough data sets for training and testing. Firstly, data was cleared of NA values and a better data set was created in place of the old one. Secondly, the new data set was explored for patterns that correspond to real-life patterns of electricity consumption. Additionally, a correlation matrix was constructed to understand the relationship between the different features of the data. Thirdly, PCA analysis was conducted to have a mathematically backed argument for the choice of the most important features that will be chosen for training. Fourthly, Hidden Markov Models served as our machine learning model that aims to detect the patterns in data and construct an abstraction of the norm that can be used for validation and comparison. However, the choice of the best HMM required analysis and comparison of many different HMMs with a different numbers of states and choice of response features in terms of BIC and Loglike values. Upon finding the best HMM (univariate and multivariate), the last step was to detect anomalies. An analysis using both Moving Window Averaging and testing with HMM was conducted in order to compare the data sets in terms of being anomalous and detecting a possibility of intrusion. Based on the results of our research, the first three test sets have the highest anomalous behavior and thus, the highest probability of a possible intrusion.

6. References

Visser, I. (2007). depmix: An R-package for fitting mixture models on mixed multivariate data with Markov dependencies. *R-package manual*, *39*, p.65. Retrieved March 30, 2020, from http://finzi.psych.upenn.edu/library/depmix/doc/depmix-intro.pdf

Wickham, H. (2016). *ggplot2: elegant graphics for data analysis*, Springer. Retrieved March 30, 2020, from https://books.google.ca/books?hl=en&lr=&id=XgFkDAAAQBAJ&oi=fnd&pg=PR8&dq=ggplot2&ots=so44bM6YaV&sig=wjIcuho7g06SoLbyOj0MOsoWCdY&redir_esc=y#v=onepage&q=ggplot2&f=false

Pinheiro, J., & Bates, D. *Extract Log-likelihood*, Retrieved March 30, 2020, from https://astrostatistics.psu.edu/su07/R/html/stats/html/logLik.html