CMPT 318 TERM PROJECT REPORT

FALL 2022

GROUP 15

Bhakti Bhanushali 301448639

Karim Khoja 301379869

TABLE OF CONTENTS

1. Introduction

Present society extensively depends upon several crucial infrastructures like communication networks, oil and gas pipelines, transportation railway, banking system, internet services etc for everyday tasks. Any disruption would have severe consequences for public safety and national security. As the automation in these sectors increases the likelihood of threats like advanced persistent threats also increases. This project explores anomaly based intrusion detection methods used for cyber situational awareness of automated control processes in electric power grids, one of such vital infrastructure.

2. Background Information

This report is based on unsupervised intrusion detection systems using time series analysis and forecasting applied to stream data from a supervisory control system. In this project we have used the electricity consumption of households over several years. We start our analysis by finding out the Principal Components that have the most effect on the entire variance of the dataset which is used to extract the variables that are the most significant. These variables serve as the basis of our further analysis. Then, we assume that the train data follows a Markov process and apply the Hidden Markov Model. This determines the ideal state of our model. This state is applied on the test data to find out the accuracy of our model. Lastly, we test our model on anomalous datasets.

3. Technical Overview

Hidden Markov models (HMMs) are sequence models. That is, given a sequence of inputs, such as words, an HMM will compute a sequence of outputs of the same length.HMM has two parts: hidden and observed. The hidden part consists of hidden states which are not directly observed, their presence is observed by observation symbols that hidden states emit.

Anomaly detection (aka outlier analysis) is a step in data mining that identifies data points, events, and/or observations that deviate from a dataset's normal behaviour. Anomalous data can indicate critical incidents, such as a technical glitch, or potential opportunities, for instance, a change in consumer behaviour. It is often applied on unlabeled data which is known as unsupervised anomaly detection

4. Methodology

4.1 Information Engineering

We started off by extracting the dataset provided and converting the response variables to their appropriate data types.

In order to simplify the problem and only consider the response variables that have the most influence on the variation of the dataset we conducted the Principal Component Analysis. Principal component analysis, or PCA, is a dimensionality-reduction method that is often used to reduce the dimensionality of large data sets, by transforming a large set of variables into a smaller one that still contains most of the information in the large set. PCA analysis returns components that are independent of one another unlike our dataset where variables share some dependency. It also returns the percentage of variation that can be explained by each of the components.

There is a slight tradeoff of accuracy while conducting the PCA analysis. PCA is very sensitive to variation of the response variables. If the variation in variable A is significantly larger than that variable B then our result would indicate that variable A has more influence on the entire result, making the result biassed. To avoid such a situation we standardise all the response variables before conducting the PCA analysis. Standardisation is done by using the scale command in R.

The percentage variation can be seen in the bar graph below:

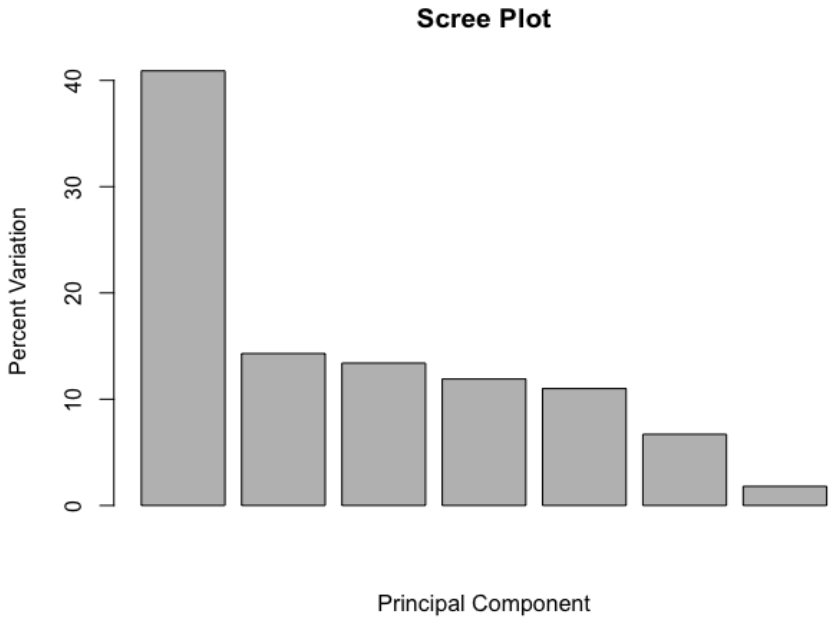

Figure 1.

Plotting the PCA on a biplot to gain insight into the effect that each variable has on PCA1 and PCA2. We observe that Global Intensity ,Global Active power and Voltage have a significant negative impact on PCA1 and Global reactive power and Sub metering 3 has a high impact on PCA2.

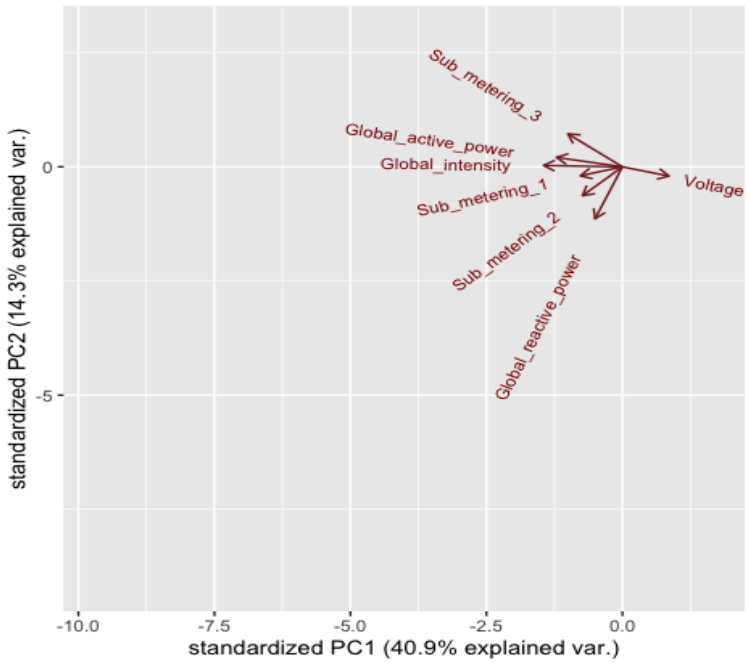

Figure 2.

	PC1	PC2	PC3	PC4	PC5	PC6	PC7
Global_active_power	-0.4687199	0.13475701	-0.087364024	-0.06854240	0.26144877	-0.76918472	-0.29968496
Global_reactive_power	-0.1947535	-0.74422839	0.166001666	0.60786960	0.06446593	-0.03290397	-0.07677664
Voltage	0.3305256	-0.13245740	-0.035064907	-0.13266015	0.91900003	0.08172722	0.05602833
Global_intensity	-0.5595970	0.01912909	0.001155733	-0.06409154	0.13818872	0.08117594	0.81036445
Sub_metering_1	-0.2988388	-0.12874880	0.728446337	-0.47839198	0.04294132	0.24064565	-0.27362731
Sub_metering_2	-0.2837926	-0.41294122	-0.651209714	-0.42742059	-0.05496931	0.28686667	-0.23846336
Sub_metering_3	-0.3874711	0.47218329	-0.094190943	0.43879699	0.24282899	0.50378618	-0.33574973

Figure 3.

We observe that PC1 explains 40.9% of the variation in our dataset and PC2 explains 14.3% of the variation. On further evaluation, we find that Global intensity has a negative correlation of 0.55 with PC1 and Global activepower has a negative correlation of 0.46 with PC1. Since PC1 dominates all the other components we pick the response variables i.e. Global Intensity and Global active power ,that are responsible for the majority of variation in PC1. These are the response variables we choose to conduct our Hidden Markov Analysis.

We decided to conduct our study on Friday's from 6:00 PM to 9:00 PM. Intuitively, on Fridays people are at home and since it's typically dark at this time we expect to see certain peaks and observe patterns.

When a random Friday is plotted, we can observe that there is a peak at around 7:00 PM and another at around 9:00 PM, as expected.

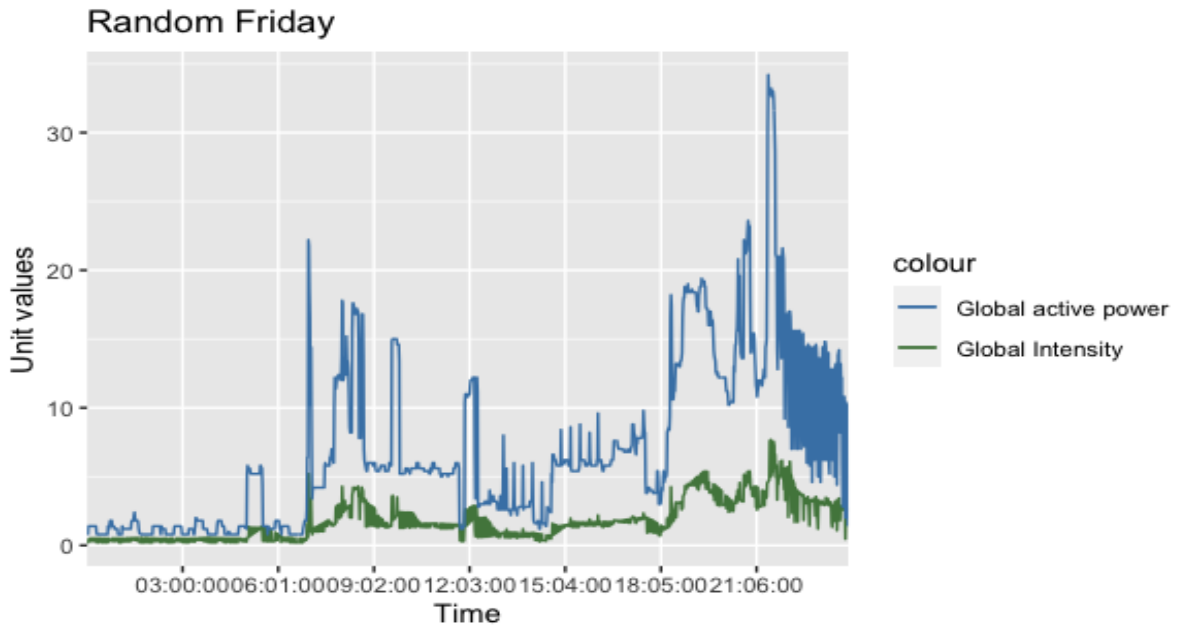

Figure 4.

We also plotted the data points from this time frame over 5 Fridays of the year and we observed a certain pattern.

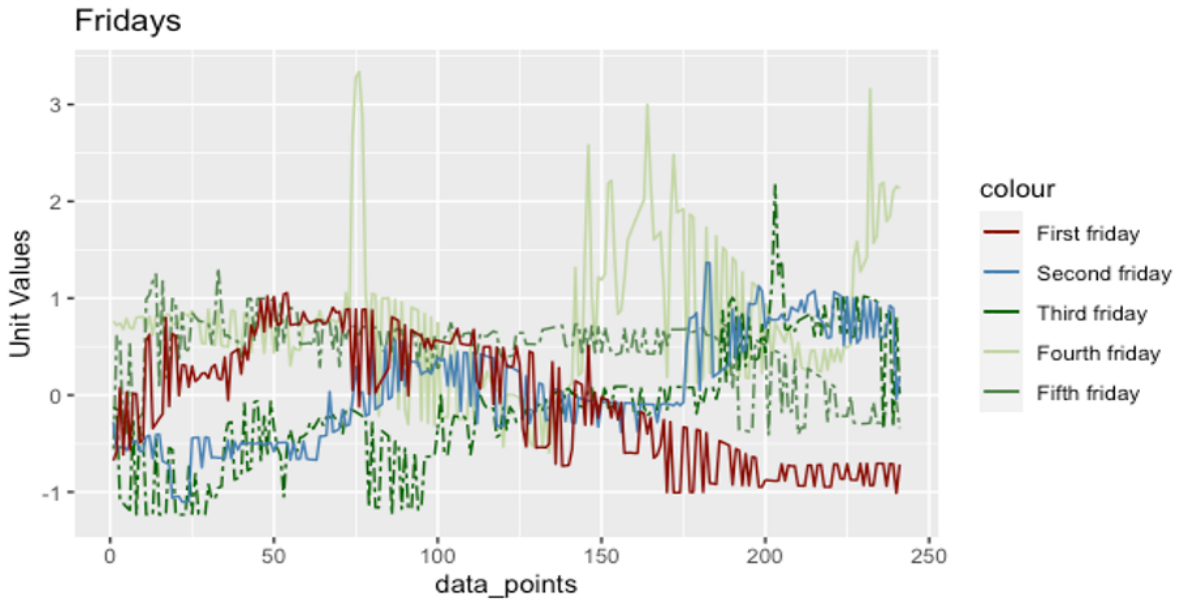

Figure 5.

After selecting the response variables we split the dataset into test and train data. Test and train data had 107 and 47 weeks worth of data.

4.2 Hidden Markov Models

Hidden Markov models are probabilistic frameworks where the observed data are modelled as a series of outputs generated by one of several hidden internal states.

Lastly, we assume that the dataset follows the Markov process .i.e. the output at the present state only depends on the value at the previous state and not the ones before that. We trained our dataset using the Hidden Markov Model with different states ranging from 4 to 24. Since the states are hidden we cannot observe the state themselves but only the result of some probability function(observation) of the states.

We picked different nstates : 4,6,8,10,12,14,16,20 and 24. The respective BIC and Log likelihood are presented in the table below.

	states	BIC	logs
1	4	49452.263	-24573.1258
2	6	33780.468	-16599.0297
3	8	25173.882	-12118.0528
4	10	24533.919	-11580.9023
5	12	14368.467	-6241.5217
6	14	10892.089	-4207.1930
7	16	8246.918	-2548.9825
8	20	5452.896	-362.2656
9	24	3691.762	1465.9486

Figure 6.

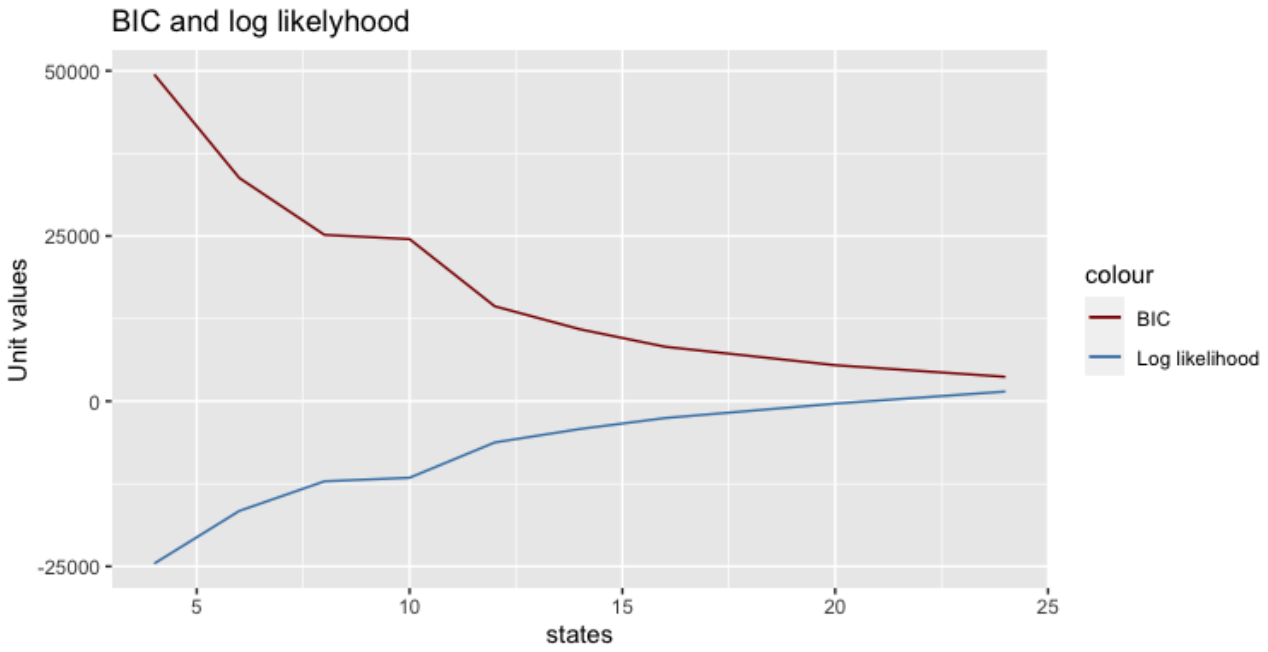

Figure 7.

BIC and Log likelihood of the states can be seen in the graph above.

It can be observed that the log likelihood and the BIC reduces as the number of states increases but at state 24 the model fails to converge to a negative value. Since states 20 and 16 have a very high log likelihood and an acceptable BIC we pick them as our ideal states to run the model on the test data. Running the model on test data and finding out the log likelihoods indicate the accuracy of our model. If the log likelihood of the train data is large and that of the test data is small it suggests that we are overfitting our model. Overfitting occurs when our model is working too hard to find patterns and finds patterns in the noise. Since these patterns do not exist in the test dataset the log likelihood becomes smaller.

The BIC and log likelihood for the test data for the selected states i.e 20 and 16 is as follows:

States	**TestBIC**	**TestLogs**
16	6165.804	-1639.6432
20	6068.509	-867.1039

Table 1.

We can see that both the states give extremely good log likelihood and BIC's but 20 has a higher log likelihood and lower BIC. We need to normalise the log likelihoods of the test and train data in order to accurately choose the state.

States	**Test data logs**	**Train data logs**
16	-34.88603	-23.82227
20	-18.44902	-3.38566

Table 2.

After normalising the values of the test and training data set we observe that for state 20 the log likelihoods are extremely close whereas for state 16 there is a wide gap. Therefore, we pick state 20 as our ideal state.

4.3 Anomaly Detection

We now test our model on datasets with anomalies and make conclusions based on the log likelihoods and BIC returned.

Dataset	BIC	Log likelihood
1	9677.169	-2647.221
2	11434.964	-3526.119
3	10286.065	-3683.649

Table 3.

We can observe that the gap between the log likelihoods of the anomalous data and the test/train data is huge. This is because of the outliers which disrupt our model. This breaks the patterns that the model looks for, resulting in a higher log likelihood. Intuitively, since the log likelihood of dataset 3 is the lowest we can claim that it is the most anomalous followed by dataset 2 and 1.

4.3.1 Univariate Model

Grubbs' test would not be appropriate for finding anomalies in the electricity data. Firstly, Grubbs' tests only the points that lie farthest from the overall mean, which is often not a meaningful reference for time series because the data may contain repeating periodic patterns. Secondly, Grubbs' test can only test one anomaly at a time, but we may expect to find multiple anomalies in the time series.

Therefore we use Seasonal-Hybrid ESD. It is a statistical test that can find multiple anomalies in time series that have seasonal patterns. The primary algorithm, Seasonal Hybrid ESD (S-H-ESD), builds upon the Generalised ESD test for detecting anomalies. S-H-ESD can be used to detect both global and local anomalies. This is achieved by employing time series decomposition and using robust statistical metrics, viz., median together with ESD. GESD is a simple statistical approach used to detect one or more outliers

in a univariate data set that follows an approximately normal distribution. Statistical approaches assume that regular data follow some statistical model and the data not following the model are outliers. The algorithm is implemented using the AnomalyDetectionVec function from the AnomalyDetection package.

To prove this hypothesis we conducted tests with the AnomalyDetectionVec function which looks at our dataset and pin points anomalies in it.

Dataset 1:

Global_active_power

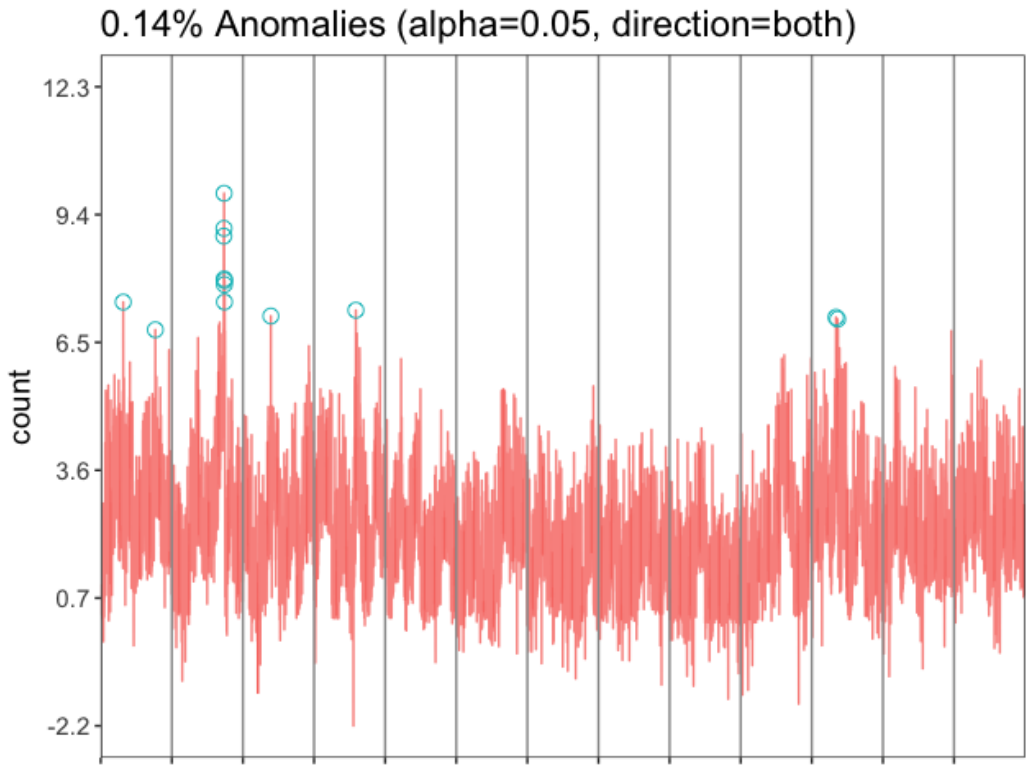

Figure 8.

Global_intensity:

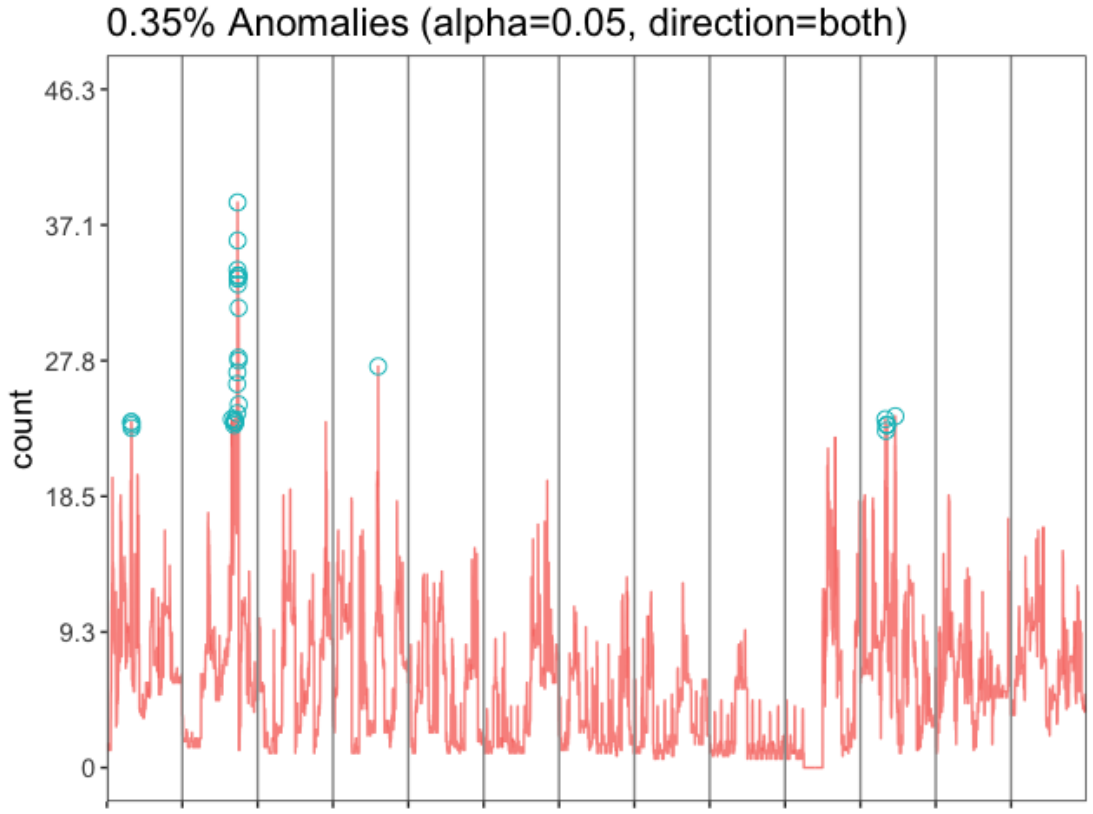

Figure 9.

Dataset2:

Global_active_power

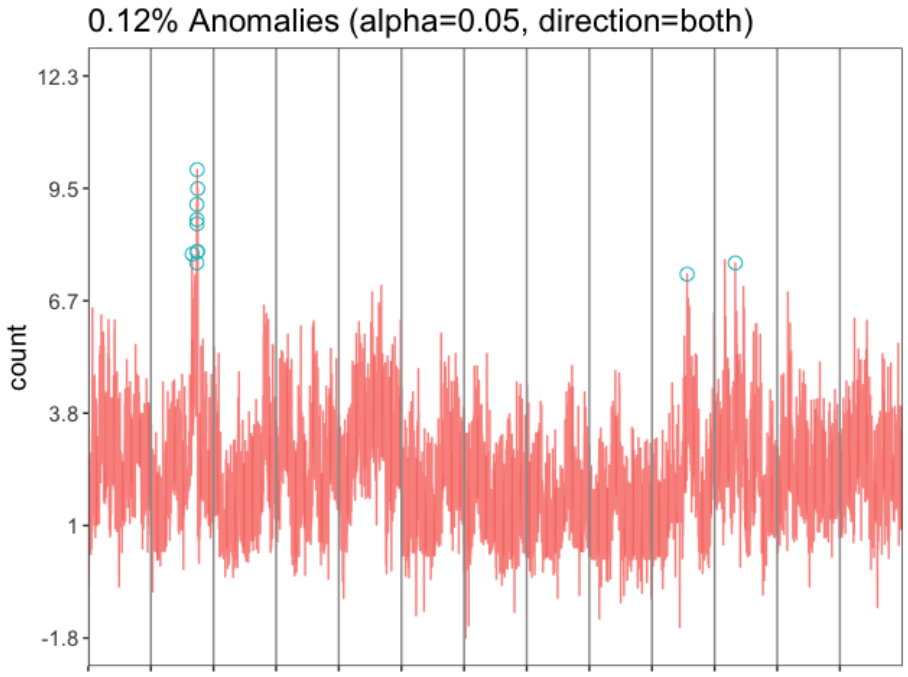

Figure 10.

Global_intensity:

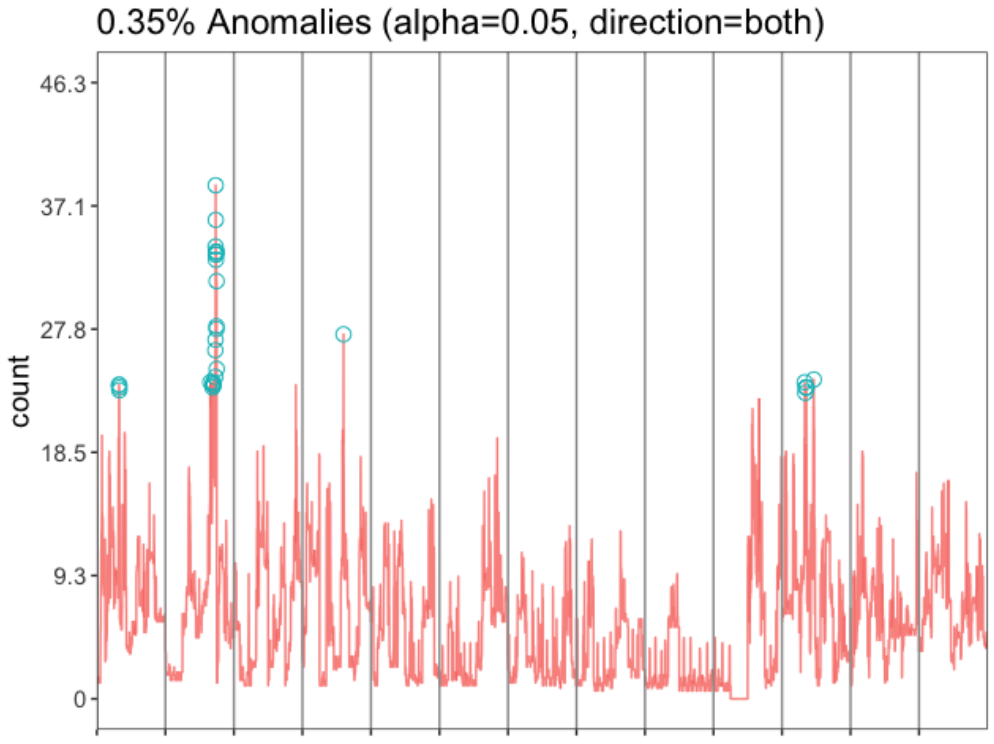

Figure 11.

Dataset3:

Global_active_power

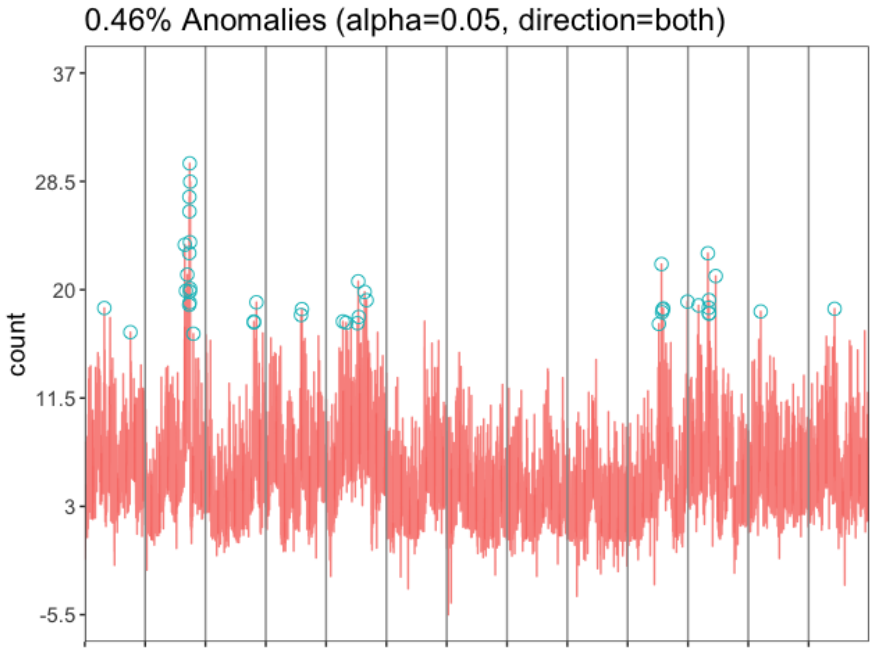

Figure 12.

Figure 12.

Global_intensity

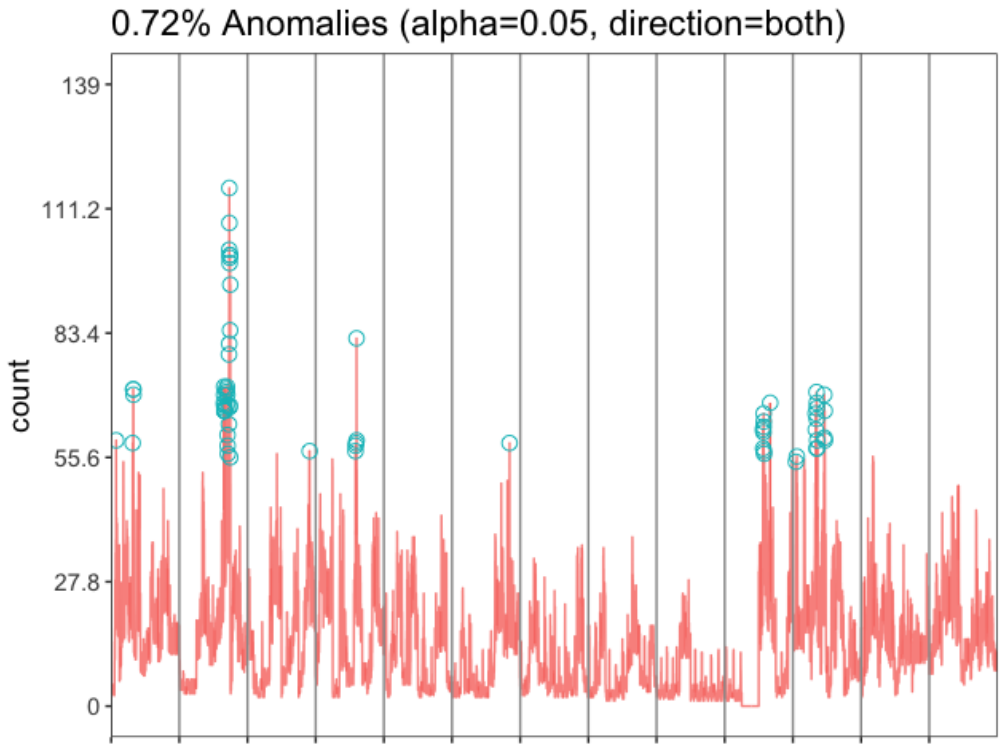

Figure 13.

Summarised account of the graphs is as below:

	Global_Active_power	**Global_intensity**
Dataset 1	0.14%	0.35%
Dataset 2	0.12%	0.35%
Dataset 3	0.46%	0.72%

Table 4.

Dataset 3 is the most anomalous dataset which aligns with our previous evaluation.

4.3.2 Multivariate Model

We used the Mahalanobis Distance (MD) to find the outliers in the datasets. It is an effective distance metric that finds the distance between a point and a distribution. It is quite effective on multivariate data. The reason why MD is effective on multivariate data is because it uses covariance between variables in order to find the distance of two points. In other words, Mahalanobis calculates the distance between point “P1” and point “P2” by considering standard deviation. We used the chi squared value of 0.99 probability and df =2 to find a cutoff value. Therefore everything that is above this cutoff .i.e has a less than 1% chance of occurring counts as an anomaly.

The Anomaly plots for each of the dataset is depicted below:

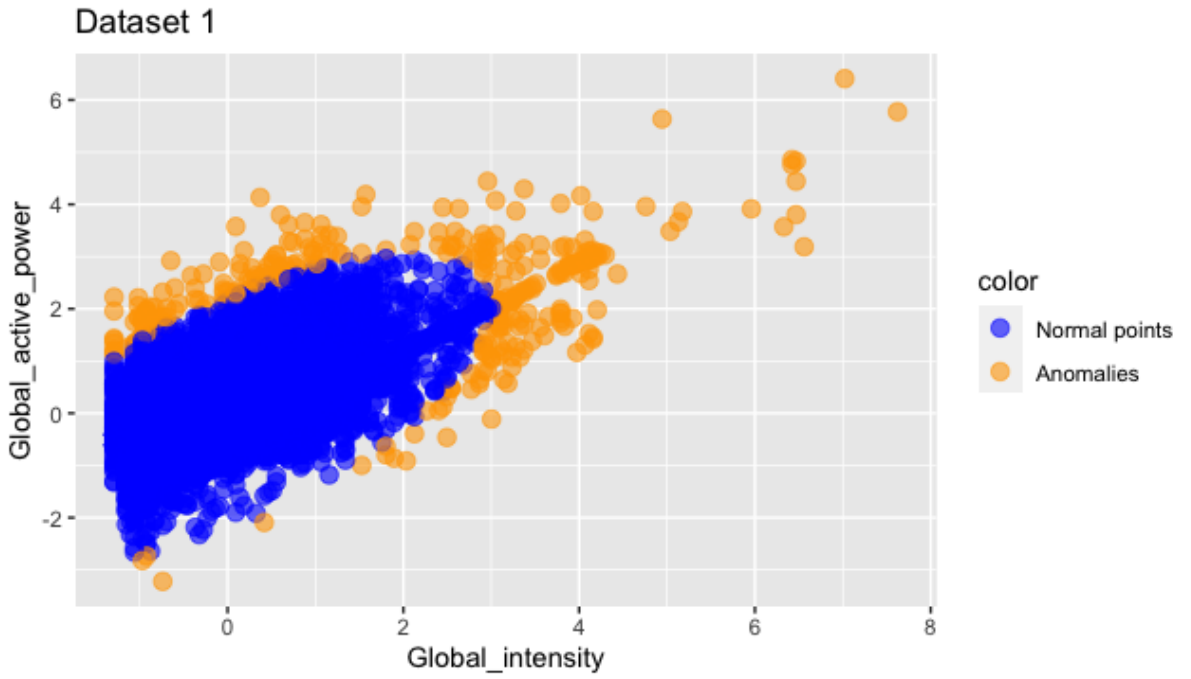

Figure 14.

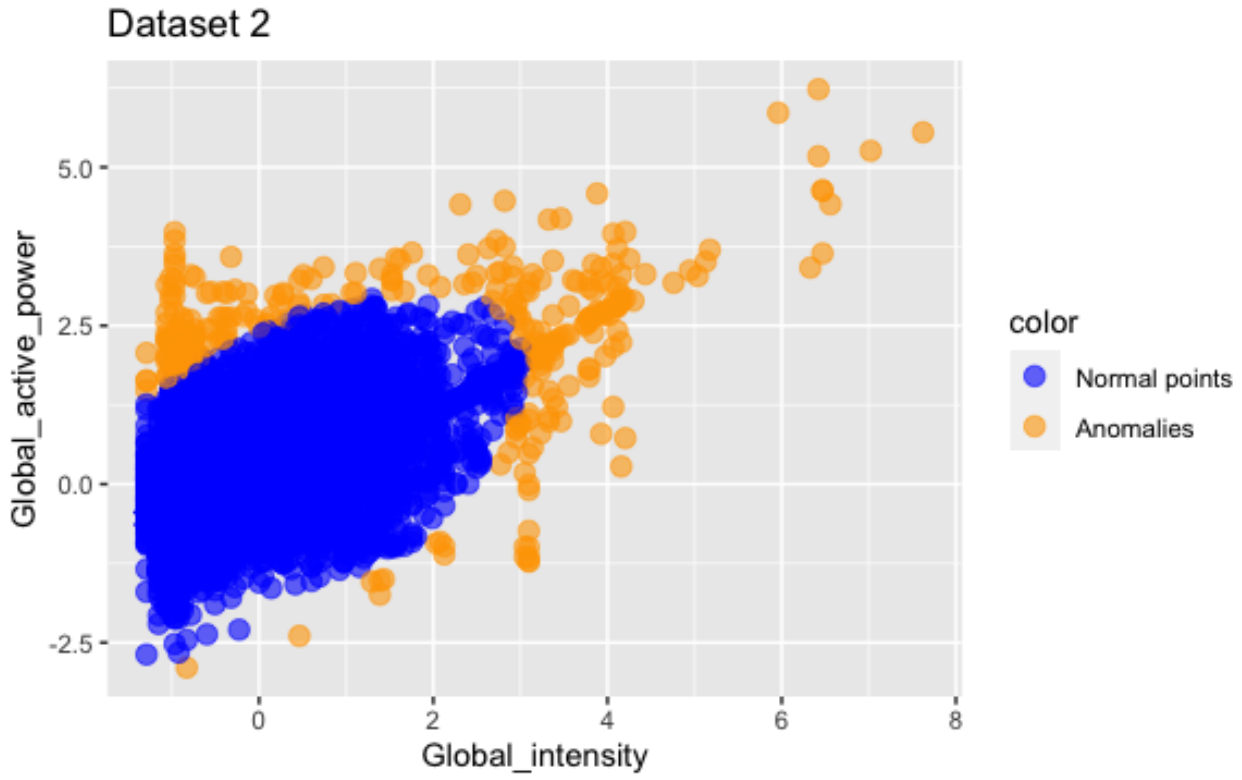

Figure 15.

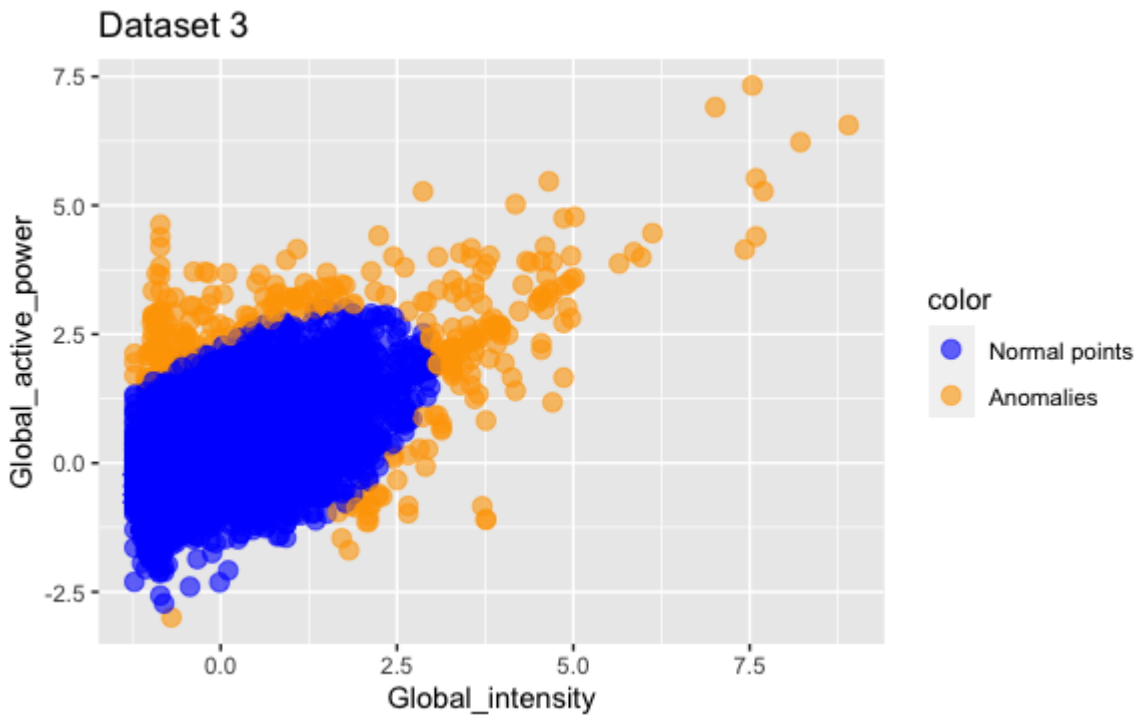

Figure 16.

On careful observation we notice that dataset 3 has the most yellow points making it the most anomalous, followed by dataset 2 and 1. The number of anomalous points in each of the dataset is presented below:

Dataset	**Number of anomalies**
1	310
2	299
3	314

Table 5.

5. Conclusions

All the 3 methods of anomaly detection – Hidden Markov Model, Univariate model and Multivariate model result in the same conclusion that dataset 3 is the most anomalous dataset.

6. Problems faced

1. We first faced the issue of missing values in the dataset that reduced the statistical power of the dataset. It also threatened the validity of our analysis. In order to maintain consistency between all of the groups of our model, we replaced all the missing values with 0's
2. We faced trouble finding the middle ground between complexity and likelihood. While trying to find a model that had the maximum log likelihood we had to be careful about the complexity of the model and choose cautiously.
3. There was also some trouble trying to make the log likelihood converge. Several attempts with different starting points were used to attain convergence.

7. Lessons learned

The project helped us in becoming acquainted with different methods of anomaly detection namely- hidden Markov Models which reproduce a log likelihood for the dataset, Univariate model and multivariate model that uses Mahalanobis distances to find outliers. We also gained insight into dimensionality reduction with Primary Component Analysis which is an important technique in statistical analysis to simplify problem solving and to obtain better results.

Furthermore, finding an optimal model helped us understand how to select a model from the train dataset and check its accuracy on the training dataset.

8. References

https://towardsdatascience.com/hidden-markov-model-hmm-simple-explanation-in-high-level-b8722fa1a0d5

https://towardsdatascience.com/anomaly-detection-for-dummies-15f148e559c1

https://www.anodot.com/blog/what-is-anomaly-detection/

https://blog.twitter.com/engineering/en_us/a/2015/introducing-practical-and-robust-anomaly-detection-in-a-time-series#:~:text=The%20primary%20algorithm%2C%20Seasonal%20Hybrid,.%2C%20median%20together%20with%20ESD.

https://towardsdatascience.com/mahalonobis-distance-and-outlier-detection-in-r-cb9c37576d7d

https://www.datacamp.com/tutorial/pca-analysis-r