# Energy Consumption Analysis

**Group 14**

Arshdeep Chhokar - 301360719

Jason Chung - 301279878

Jacob He - 301391374

**CMPT 318 - Intro to Cybersecurity**

Spring 2022

## Abstract

Energy consumption in the form of electricity has become an integral part of the successful function of contemporary society. We rely on this energy to fuel our vehicles, heat our homes, and consume technology as members of the twenty-first century. This dependence makes anomaly detection in energy consumption data all the more vital in order to ensure the wellbeing of society.

## Table of Contents

**1. Introduction** **3**

1.1 Problem Scope and Background 3

1.2 Methodology 4

**2. Team Contributions** **5**

**3. Selection of Variables** **6**

3.1 PCA Analysis 6

3.2 Selected Time Window 8

**4. Training HMMs** **9**

4.1 State Selection 9

4.2 Test-train split 11

**5. Identifying Anomalies** **11**

**6. Conclusions** **12**

6.1 Problems Encountered 12

6.2 Lessons Learned 13

**7. References** **14**

## Table of Figures

**Figure 1** **3**

**Figure 2** **4**

**Figure 3** **6**

**Figure 4** **7**

**Figure 5** **8**

**Figure 6** **9**

**Figure 7** **10**

**Figure 8** **10**

**Figure 9** **12**

# 1. Introduction

## 1.1 Problem Scope and Background

As cyber attacks continue to grow in frequency and severity, the need for increased security measures has grown significantly. Advanced persistent threats in addition to existing vulnerabilities pose a great risk to critical infrastructure. Certain systems such as electricity grids rely on automated control processes to operate successfully. These processes must be continuously analyzed for suspicious behavior. This project will implement an intrusion detection mechanism on electricity consumption data that was generated from an automated control system.

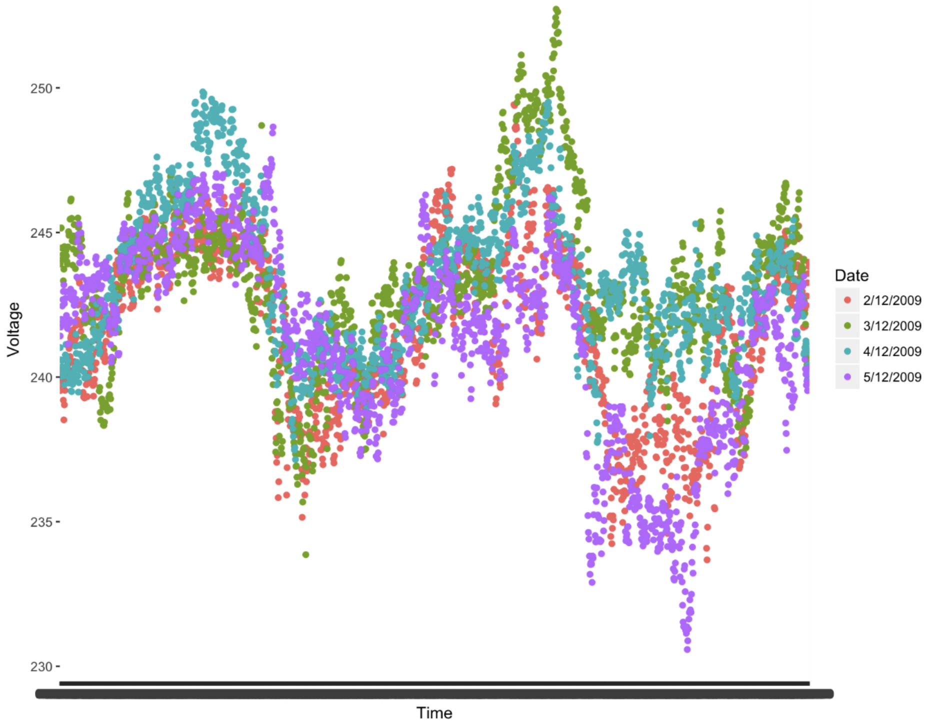


*Figure 1 - A time series of voltage consumption (Courtesy of CMPT 318 project overview slides)*

## 1.2 Methodology

The process of building a detection system begins with the selection of predictors that adequately represent the patterns found in the electricity consumption data. After scaling the data to ensure consistent values across all variables, a principal component analysis was performed in order to find a smaller subset of variables that maximize variance while minimizing information loss. Using a training subset of the data, numerous HMMs were trained with various numbers of states in order to see which model was most appropriate based on the Log-likelihood and BIC values. This trained HMM was then applied to three different datasets containing anomalies in order to determine the degree of anomalous behavior in each dataset.

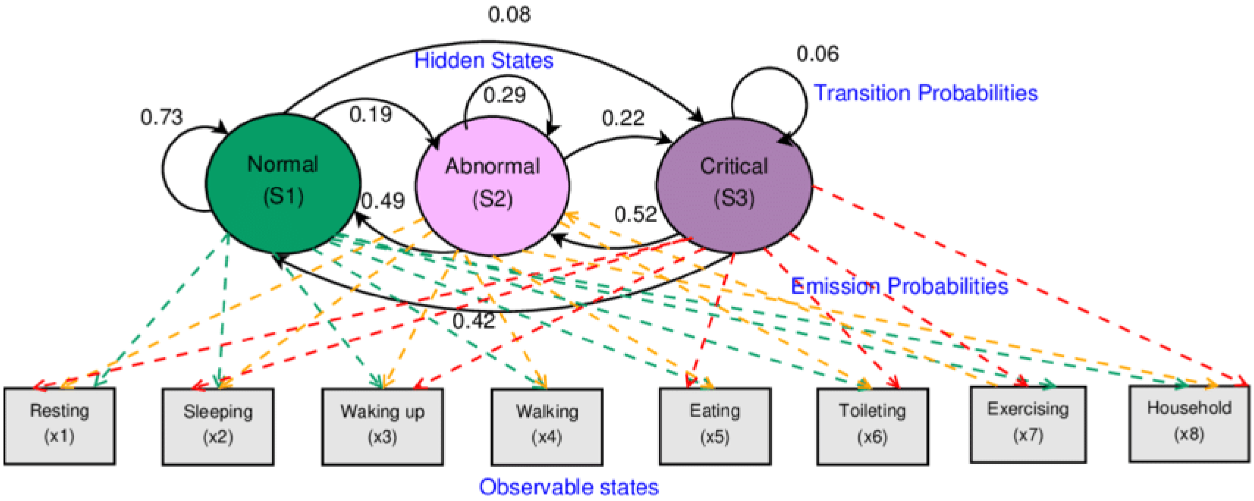


*Figure 2 - An example of a Hidden Markov Model used for modeling daily activities (Courtesy of ResearchGate)*

# 2. Team Contributions

**Arshdeep Chhokar**

- Worked on implementing R code and writing the project report
- Complete slides for presentation
    - Introduction
    - Training of models
    - Problems encountered
    - Lessons learned

**Jacob He**

- Worked on implementing R code and writing the project report
- Contributed to presentation slideshow
    - Selection of variables
    - PCA analysis

**Jason Chung**

- Worked on implementing R code and writing the project report
- Contributed slides for presentation
    - Identifying anomalies
    - Conclusions and recap

# 3. Selection of Variables

## 3.1 PCA Analysis

The following bar plot illustrates that the first principle component accounts for more than 35% of the variance in our dataset. This means the variables within this principal component provide the most accurate representation of our dataset. As a result, variables corresponding to the first principle component will be selected.

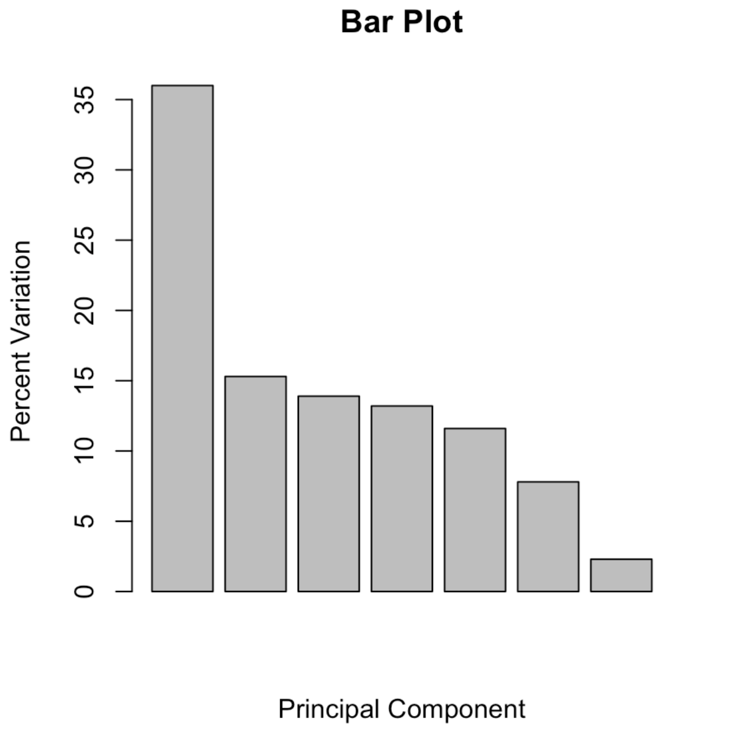
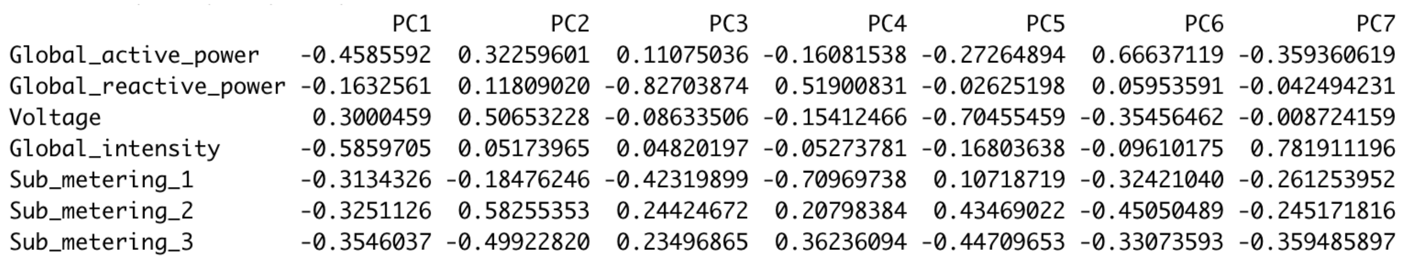


*Figure 3 - Bar Plot showing results of PCA*

The table of principal components in Figure 4 shows the loading scores for each variable in our data set. These loading scores show how much effect each variable has on the principal components. Looking at the absolute values in PC1, we can see that Global-intensity and Global-active power were the top two variables with scores of -0.5859 and -0.4585 respectively.

This means that these two variables have a strong negative correlation to the principal component and have a greater effect on the first principal component than any other variables.



```
                              PC1         PC2         PC3         PC4         PC5         PC6          PC7
Global_active_power   -0.4585592  0.32259601  0.11075036 -0.16081538 -0.27264894  0.66637119 -0.359360619
Global_reactive_power -0.1632561  0.11809020 -0.82703874  0.51900831 -0.02625198  0.05953591 -0.042494231
Voltage                0.3000459  0.50653228 -0.08633506 -0.15412466 -0.70455459 -0.35456462 -0.008724159
Global_intensity      -0.5859705  0.05173965  0.04820197 -0.05273781 -0.16803638 -0.09610175  0.781911196
Sub_metering_1        -0.3134326 -0.18476246 -0.42319899 -0.70969738  0.10718719 -0.32421040 -0.261253952
Sub_metering_2        -0.3251126  0.58255353  0.24424672  0.20798384  0.43469022 -0.45050489 -0.245171816
Sub_metering_3        -0.3546037 -0.49922820  0.23496865  0.36236094 -0.44709653 -0.33073593 -0.359485897
```

*Figure 4 - Table showing results of loading scores for each PC*

Looking at the PCA plot for all seven variables in Figure 5, we can gain even more insight into the selection of variables. In this case, the two selected variables global intensity and global active power are good picks because they are most parallel to the axis of PC1. This means they contribute greatly to PC1. In addition, the length of the vectors representing global intensity and global active power are greater than the length of the vectors representing global reactive power and submetering1 which are also close to PC1. This means that global intensity and global active power have greater variability represented in the two principal components than the other two variables and hence they are the better picks.

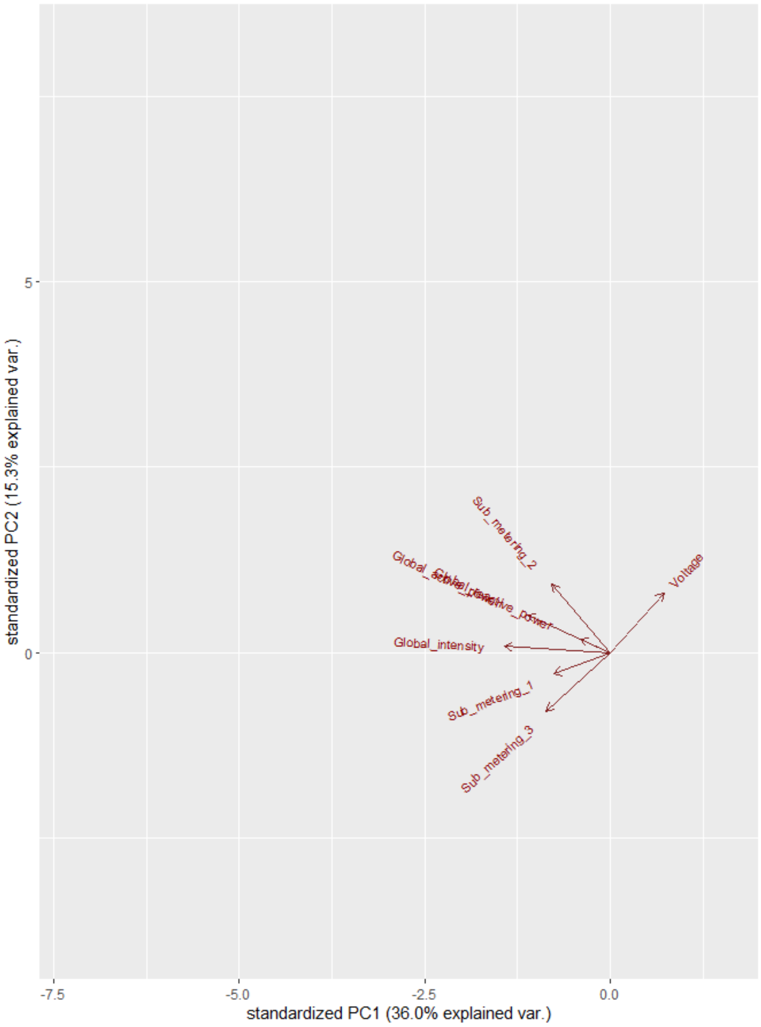


*Figure 5 - A PCA plot showing relationship between variables and PCs*

## 3.2 Selected Time Window

The time window selected from the data set was every Sunday from 8 am to 11 am. It was decided that a 3 hour time window would be best because it contains enough data points to sufficiently recognize patterns in the data.

# 4. Training HMMs

## 4.1 State Selection

In order to select the right number of states, various HMMs were trained with states ranging from 4 to 20 in increments of 4 (4, 8, 12, 16, etc). In order to avoid the significant time delay of training models with a higher number of states, it was decided to make 20 states the upper bound.

In Figure 6 we can see that as we move up the number of states the number of iterations required for convergence increases significantly. Therefore, the runtime for HMMs trained on greater than 20 states will also require significant memory and runtime.

| Number of States | Number of Iterations | Log Likelihood |
|---|---|---|
| 4 | 51 | -18860 |
| 8 | 94 | -7386 |
| 12 | 121 | -2415 |
| 16 | 276 | 240 |
| 20 | 441 | 2917 |

*Figure 6 - A table showing log likelihood results for various states*

In Figure 7, we can observe that as the number of states increase the BIC values steadily decrease while the log-likelihood values increase. This means that the complexity of our model will decrease and its goodness of fit will increase as we move up the number of states. This plot also shows that the log likelihood goes above zero after a certain point and intersects with the BIC plot. In this case we had two different options. We could either select the number of states at the intersection point (roughly 17) or the point where BIC has the lowest value and log-likelihood is the highest (20 states). Looking at Figure 8 we can see that the difference between normalized training and test log-likelihoods for 17 and 20 states is similar (roughly 70). However, it was decided to select an HMM with 20 states because it is the easiest to interpret while also providing the best fit.

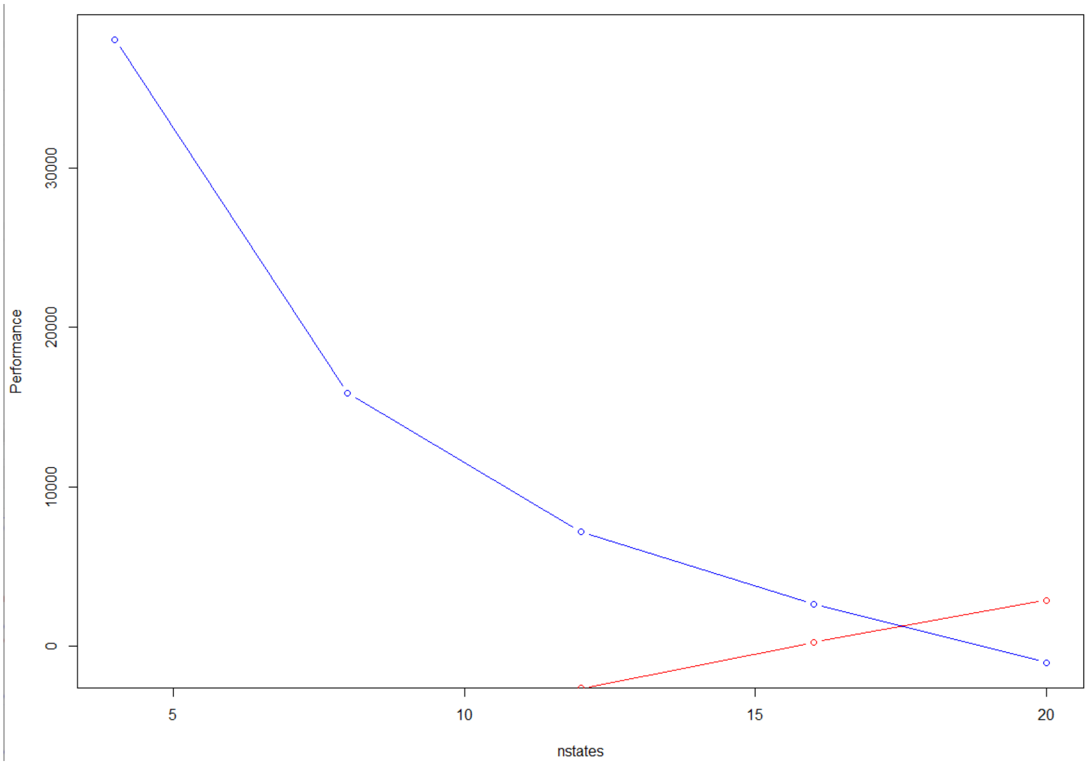


*Figure 7 - Graph comparing BIC(blue) and LL(red) values for various states*

| Number of states | Normalized log likelihood (training) | Normalized log likelihood (testing) |
|---|---|---|
| 17 | 10.1 | -51.7 |
| 20 | 27 | -42.7 |

*Figure 8 - Comparing the results of HMM with 17 and 20 states*

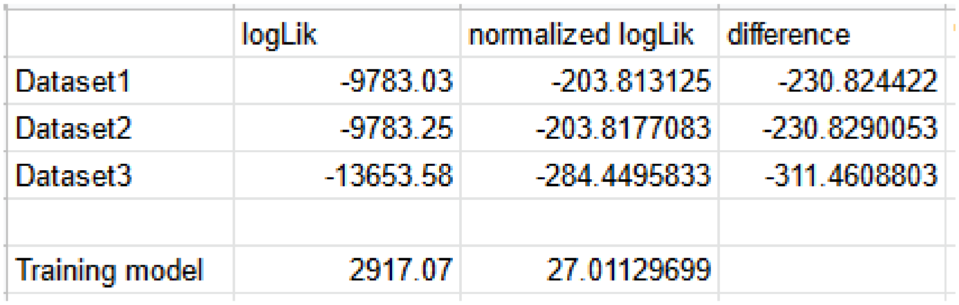
In Figure 8, we can look at the difference between the normalized train and test log likelihoods to see how well the model has performed in 20 states. In this case the difference is about 70 and more specifically, the normalized train log-likelihood is approximately 3x greater than the normalized test log-likelihood. It may be possible to train an HMM with an even greater number of states to achieve a smaller difference (hence better performance), but there is a risk of overfitting the data because the BIC values may suddenly increase after a certain number of states.

## 4.2 Test-train split

After selecting a time window of 3 hours (8am to 11am) for every Sunday, we partitioned the data set so that approximately 70% of the data points were used for the training set, and the remaining 30% was used for the testing set. The reason this was done is because the first 70% represents roughly 2 years worth of data to be used for training, while the remaining 30% which is about 1 years worth of data would be used for testing.

# 5. Identifying Anomalies

In order to identify anomalies, we first processed the data from each of the three anomalous datasets to match our training model so that they each contain the same hour intervals and have the rows with missing values removed. Since we previously selected 20 states during the training and testing phases, we will create a model for each of the three datasets using 20 states. We then applied the forward backward function to each of these models after setting the parameters from the fitted training model to calculate their individual log likelihoods. Since the dataset corresponding to the training model contains a lot more observations, the results had to be normalized. This was done by dividing the log-likelihood by the number of days contained in the dataset. The results are displayed in Figure 9. From this table we were able to identify the degree of anomaly contained within each dataset by comparing their log likelihoods to the training model's as shown by the difference column. It can be observed that the log likelihoods of the first two datasets are quite similar which indicates that they contain similar amounts of anomalous data.  However, the third dataset has a much lower log likelihood and greater difference than the other two datasets which means that it contains the most anomalous data out of the three datasets.



| | logLik | normalized logLik | difference |
|---|---|---|---|
| Dataset1 | -9783.03 | -203.813125 | -230.824422 |
| Dataset2 | -9783.25 | -203.8177083 | -230.8290053 |
| Dataset3 | -13653.58 | -284.4495833 | -311.4608803 |
| | | | |
| Training model | 2917.07 | 27.01129699 | |

*Figure 9 - Results of applying trained HMM to 3 anomalous datasets*

# 6. Conclusions

## 6.1 Problems Encountered

### 6.1.1 Missing values

The data contained several missing (NA) values. In particular, one of the columns we were interested in (Global intensity) had this problem for the chosen time interval. Our initial approach was to fill in these missing values using the mean of the present data. However, we later found out that all the missing values emerged from the same day and the entire column was missing from that single day. We were able to identify this through the use of aggregation. First, we removed all rows with missing values and then aggregated the total row count for each of the remaining unique days. Upon inspecting the aggregated results, we found out that all the row counts for each day added up to 180 (corresponding to the chosen 3-hour interval) and thus we were able to confirm the missing values indeed arose from just one single day. Since the data is now mostly complete for the remaining days, we decided it would be better to drop the day with missing values rather than using the mean imputation approach.

### 6.1.2 Computation and Resource Limitations

Training and fitting the models was very time consuming and requires a lot of computational resources. This led to numerous system crashes, loss of progress, and time wasted. There were multiple factors that contributed to this. The number of states has a significant impact on the time it takes to fit a given model. As the number of states increases, the iterations required for convergence also increases significantly. Secondly, the selected distribution methods directly relate to the amount of memory required. For example, at one point we attempted to build the models using multinomial distributions for both responses and this resulted in several models taking up over 30GB and eventually crashed. This problem accounted for the majority of the time we spent working on the project.

## 6.2 Lessons Learned

We learned from doing PCA analysis to choose the variables which had the strongest impact on our models. From our calculations, we found that the two values were Global-intensity and Global-active power, which we used to build our models. We learned from our approach from the beginning that trying to fit N/A values would cause issues with the models produced. We realized that the best solution to deal with missing values was to completely eliminate the data points that would skew our resultant models, or to replace missing values with mean values. After building the HMM's with N states = {4, 8, 12, 16, 20}, we were able to decide the most appropriate models with respect to BIC and Log-Likelihood, which was a model with 20 states. Lastly, we learned how to calculate and evaluate log likelihoods to detect anomalies by using the trained model on the anomalous datasets. Specifically, we took the parameters of the trained model, passed them to the models representing the anomalous datasets in order to get the log likelihoods which show us the degree to which there are anomalies present in each of the datasets.

# 7. References

Brownlee, J. (2020, August 26). Train-Test Split for Evaluating Machine Learning Algorithms. Retrieved April 2, 2022, from https://machinelearningmastery.com/train-test-split-for-evaluating-machine-learning-algorithms/

Brownlee, J. (2019, October 29). Probabilistic Model Selection with AIC, BIC, and MDL. Retrieved April 2, 2022, from https://machinelearningmastery.com/probabilistic-model-selection-measures/

Hartmann, K. (2018). *Interpretation and Visualization*. Statistics and Geospatial Data Analysis. Retrieved April 3, 2022, from https://www.geo.fu-berlin.de/en/v/soga/Geodata-analysis/Principal-Component-Analysis/principal-components-basics/Interpretation-and-visualization/index.html#:%7E:text=The%20biplot%20is%20a%20very,in%20a%20single%20biplot%20display.&text=The%20plot%20shows%20the%20observations,principal%20components%20(synthetic%20variables)

Jaadi, Z. (2021, December 1). A Step-by-Step Explanation of Principal Component Analysis (PCA . Retrieved April 2, 2022, from https://builtin.com/data-science/step-step-explanation-principal-component-analysis

Jurafsky, D., & Martin, J. H. (2021, December 29). Hidden Markov Models. Retrieved April 2, 2022, from https://web.stanford.edu/~jurafsky/slp3/A.pdf

Skogholt, M. (2020, May 4). CRAN Packages By Name. Retrieved April 2, 2022, from https://cran.r-project.org/web/packages/available_packages_by_name.html