CMPT 318 Spring 2022

TERM PROJECT: Critical Infrastructure Protection

Group 21

Lee Roy Gomos (301313035)
Khushwant Parmar(301370994)
Paolo Sy-Quia (301346727)

Table of Contents

Abstract **3**

1 Introduction **3**

1.1 Problem Scope 3

1.2 Technical Background 4

2 Variable Selection **5**

2.1 Principal Component Analysis 5

2.2 PCA Results 5

Figure 1: PCA plot 5

Figure 2: Rotation (variable loadings) matrix 6

2.3 Factors Taken Into Account 7

2.4 Final Choice of Variables 8

2.5 Observation Time Window 8

Figure 3: Time window comparison 8

3 Multivariate Hidden Markov Model **9**

3.1 Preparing the Dataset for Training 9

3.2 Data Partitioning 9

3.3 Overview of log-likelihood and BIC values 9

Figure 4: Log-likelihood plot 10

Figure 5: BIC plot 11

Figure 6: Log-likelihood and BIC 11

4 Testing Our Model **12**

4.1 Evaluation on Train and Test Data 12

4.2 Evaluation on Datasets with Injected Anomalies 12

5 Conclusion **13**

6 References **14**

Abstract

With the major adoption of automation for critical infrastructure and the landscape of increasingly complex cyber threats, the potential for damage and harm from cyberattacks has never been greater. In particular, the rise of advanced persistent threats and their risk of cascading effects has emphasized the importance of good intrusion detection. In this paper, we demonstrate an anomaly-detection based method for intrusion detection using an analysis of electrical consumption data from an automated control process as an example. We performed Principal Component Analysis (PCA) to select appropriate variables for training a multivariate Hidden Markov Model (HMM). We experimented with multiple models to find the optimal number of states. After training our model, we evaluated it on test data in addition to three different datasets with anomalies.

1 Introduction

1.1 Problem Scope

Functioning and monitoring of critical infrastructure has experienced a shift towards automation by large scale integration of physical systems with network and other cyber systems. This has led to the formation of cyber physical systems which not only increase efficiency and production but also increase the attack surface for advanced persistent threats. The project explores anomaly-detection based intrusion detection methods used for cyber

situational awareness in the analysis of automated control processes. The data being studied here is real-time control data from the continuous operation of a cyber-physical system monitoring electricity consumption. Technically, such data forms a multivariate time series that needs to be analyzed in (near) real-time. Continuous data analysis is vital for early threat detection to mitigate the impact of attacks by launching countermeasures.

1.2 Technical Background

A hidden Markov model (HMM) is a Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. These models have initial and transition probabilities governing what states the system starts in and how the transition between states take place. This means for each of the N possible states a hidden variable at time t can be in, there is a transition probability from this state to each of the n possible states of the hidden variable at time t+1, for a total of N^2 transition probabilities. Also, there are transition probabilities governing the distribution of observed variable value at a particular time given the state of the hidden variable at that time. We can be dealing with a single variable changing with time or multiple variables. A multiple variable changing Hidden Markov Model is called a multivariate model, which will be what we are working with in this project. Furthermore, to evaluate these models there are some useful parameters available at our disposal. First, the likelihood of an observation sequence is the multiplication of a number of probabilities (i.e., transition and emission probabilities) and calculating their sum. A greater likelihood of an observation sequence implies that the given HMM can generate this sequence

more likely than the other sequence with the lower likelihood. If a sequence of observations has a very low likelihood, it can be that this sequence does not match with the (normal) behavior captured by the HMM. For longer sequences the likelihood values can drop below interpretable levels and thus, to avoid that we take the log of the total calculation producing the log-likelihood for the model. It is possible to increase the likelihood of a sequence by increasing the number of states of the model but it may result in overfitting, to counter that we rely on another parameter called BIC, which determines the complexity of the model. BIC penalizes models for complexity, leading to more complex models having larger scores. We aim to maximize log-likelihood while keeping the BIC low (a simple model).

2 Variable Selection

2.1 Principal Component Analysis

Principal Component Analysis (PCA) is a technique most commonly used to analyze datasets with many variables. PCA works by reducing the dimensionality of a dataset through a linear transformation that simplifies a dataset with many variables to a dataset with a smaller number of variables, called principal components (PC).

We are interested in principal components that account for a large proportion of variance because these components capture most of the information from our original dataset. Each PC is a linear combination of the original variables where each of the variables has a coefficient (called variable loading). We utilized Principal Component Analysis to inform our decision on selecting variables for HMM training. For computing PCA, we used *prcomp* from the R stats package and for plotting our PCA graph, we used *ggbiplot.*

2.2 PCA Results

This is the resulting plot from our PCA of our standardized electricity consumption dataset.

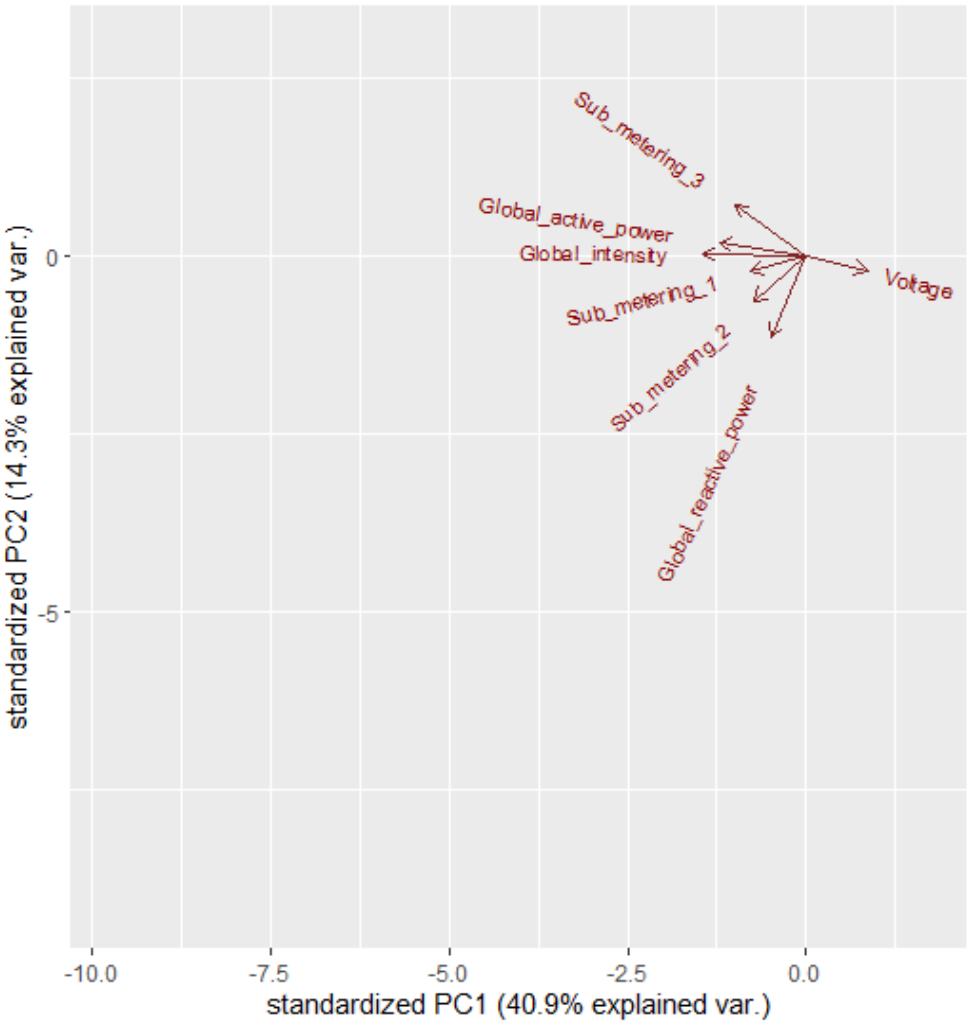

Figure 1: PCA plot (samples omitted for clarity)

We get the variable loadings corresponding to each of the seven PC's from *pca$rotation* (where pca is the result of *prcomp)*

	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
Proportion of Variance	0.4085600	0.14262000	0.134350000	0.11916000	0.11006000	0.06724000	0.01802000

Figure 2: Rotation (variable loadings) matrix

2.3 Factors Taken Into Account

Loading Score

A good indicator of a variable's overall impact is its loading score. We primarily focused on PC1 loading scores because PC1 accounts for 40.9% of total variance in the dataset. Looking at Figure 2, we can see that the top three variables in terms of loading score magnitude are *Global_intensity*, *Global_active_power*, and *Sub_metering_3*. This can also be observed in Figure 1, where the three variables mentioned have the longest arrow lengths. Despite these three having the largest loading scores, we opted to select Voltage, Global_intensity**,** Global_active_power, but **not Sub_metering_3**. We will explain this choice with the next factor taken into account.

Maximizing Representation

The loadings of our variables represent the direction and magnitude by which they will push the samples in each PC. For example, this can be seen in the PCA plot (Figure 1) with *Sub_metering_3*, its arrow points left (negative in PC1 axis) and up (positive in PC2 axis). When selecting variables, we believed including *Voltage* was critical because it was the only variable

with a positive loading in PC1 and its loading magnitude is comparable to *Sub_metering_3*. Additionally, we selected *Global_intensity* and *Global_active_power* which have the largest negative loadings. Note: we greatly prioritized PC1 representation and loading scores when selecting variables due to the proportion of variance disparity (PC1 40.9% vs. PC2 14.3%).

2.4 Final Choice of Variables

For the reasons described in the previous section, our selected variables are **Global_intensity**, **Global_active_power**, and **Voltage**.

2.5 Observation Time Window

Our selected observation time window is Monday, 9 AM - 11 AM. As a common time people typically start working on a Monday, we expected this time window to feature consistently moderate to high energy consumption and this is what we observed.

	G_active_power.mean	G_active_power.var	G_intensity.mean	G_intensity.var
Monday, 9-11AM	1.428914	0.5762004	5.434894	11.72430
All entries	1.227778	1.1182153	4.642035	20.96628

Figure 3: Time window comparison

The table in Figure 3 shows that our selected time window has a higher mean for *Global_active_power* and *Global_intensity* when compared to the entire dataset. Additionally, our time window has a much lower variance for *Global_active_power* and *Global_intensity*.

3 Multivariate Hidden Markov Model

3.1 Preparing the Dataset for Training

To train our dataset, we used a range of n-states from 4 to 24. To simplify and speed up the process, we decided to train every even number of n-states within this range, training a total of 10 models. There will be 120 data entries per week for our chosen time window. The given dataset has 155 weeks worth of entries, therefore the total number of datapoints would be 18600. However, we noticed that some of the weeks had incomplete data. To deal with this, we removed those weeks from our dataset in order to ensure that each week had complete data, each having 120 entries. In the end we were left with 153 weeks with complete data for our specified time window.

3.2 Data Partitioning

We partitioned the data into approximately 75% training data to 25% test data. They are partitioned by weeks in order to not split up any entries from the same time range for that week, with the training set having 114 weeks (13,680 data entries) and the test set having 39 weeks (4,680 data entries). We chose this split in order to minimize the runtime, error, and the score differences between the training and test set [1].

3.3 Overview of log-likelihood and BIC values

As per figure 4, we can see that the log likelihood is steadily increasing as we increase the number of states. Once we reach n-states = 20, we notice a big increase from the previous

models. We also observed that at around n-states = 24 is when the log likelihood begins to increase past zero.

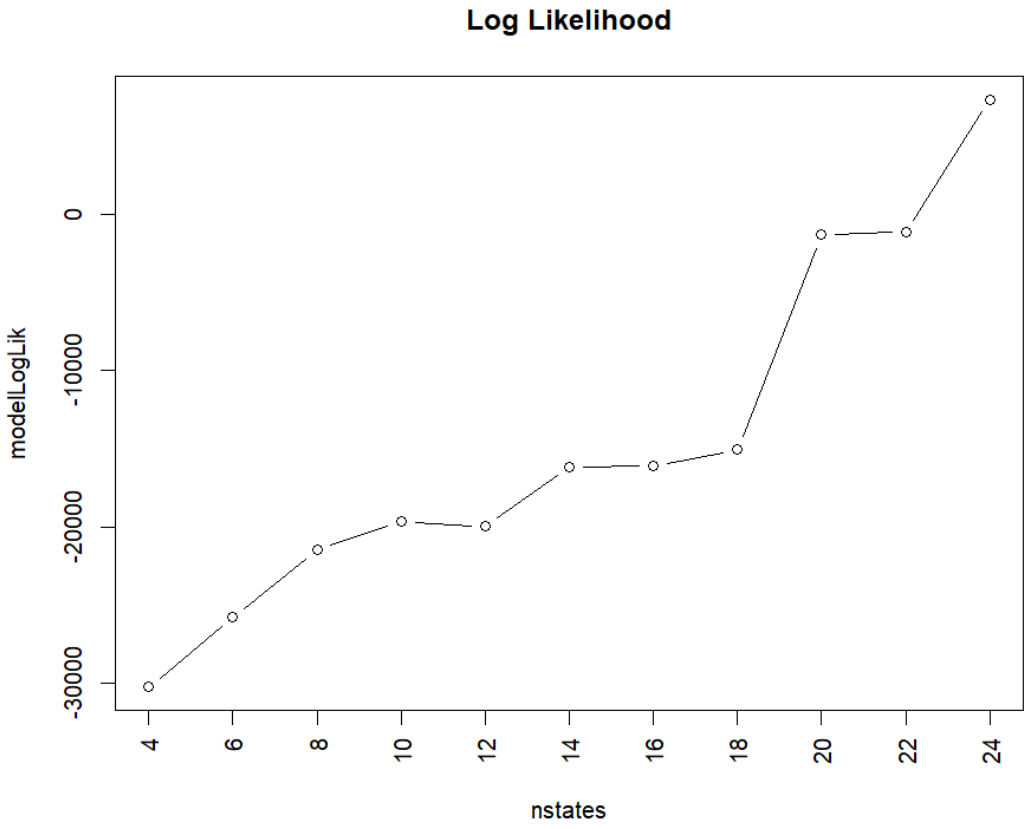

Figure 4: Log-likelihood plot

Taking a look at the BIC values (Figure 5), we can see a similar pattern occurring. At n-states = 20, we notice a big decrease in values, and at n-states = 24 is where we see the values begin to go below zero. From these results, we can see that the sweet spot for the number of states is around 20 to 22. We have chosen our final number of states to be 22 due to it having a slightly higher log likelihood than n-states 20, as well as having very similar BIC values.

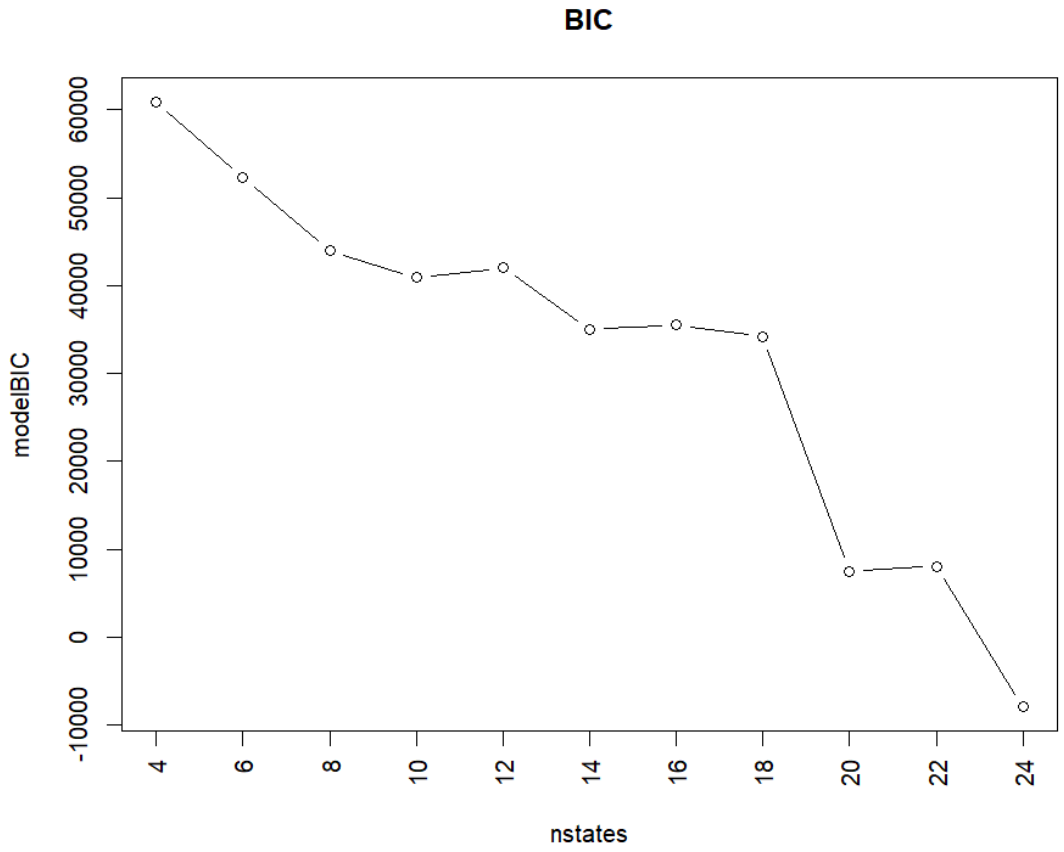

Figure 5: BIC plot

n-states	train logLik	train BIC	test logLik	test BIC
4	-30273.868	60919.161	-10405.826	21141.242
6	-25806.075	52288.332	-8484.384	17568.793
8	-21478.915	44014.959	-6541.037	14020.141
10	-19689.656	40893.579	-5563.807	12471.332
12	-19987.770	42023.133	-4346.877	10510.730
14	-16204.186	35065.482	-3753.692	9865.228
16	-16091.633	35526.081	-3598.980	10164.279
18	-15058.535	34221.780	-2299.059	8240.521

20	-1283.533	7509.861	-2640.606	9667.308
22	-1104.138	8065.346	-1377.708	7952.813
24	7389.674	-7931.814	-1331.904	8740.116

Figure 6: Log Likelihood and BIC

4 Testing Our Model

4.1 Evaluation on Train and Test Data

```
Normalized log-likelihood training data: -1104.138 / 13680 = - 0.0807
Normalized log-likelihood test data: -1377.708/ 4680       = - 0.2943
```

We see that the test log-likelihood is less than training data log-likelihood but the difference is not a significant one. The higher training data values are due to the fact that the model is fitted to the training data. Test data getting a log-likelihood value not so far off from train data values implies that our model is reasonably accurate in predicting values and is safe from over or underfitting.

4.2 Evaluation on Datasets with Injected Anomalies

Using our model, we calculated the log-likelihood of three different datasets which featured injected anomalies. The following are the results:

```
Normalized log-likelihood training data:  -1104.138 / 13680 = - 0.0807
Normalized log-likelihood anomalous data 1: -15777.5/6000   = - 2.6295
Normalized log-likelihood anomalous data 2: -15777.5/6000   = - 2.6295
Normalized log-likelihood anomalous data 3: -29059.2/6000   = - 4.8432
```

The first two anomalous data sets have the same log-likelihood but the third data set has significantly lower values. Inferring from the log-likelihood diversion from the training data set values we infer that the data sets have significantly different characteristic data.
Translating the lower likelihood values to the number of anomalies present in the data, we conclude dataset 3 has the highest number of injected anomalies, almost double than dataset 1 and 2.

5 Conclusion

In this project, we analyzed a set of real-time stream data from a supervisory control system for the purpose of threat detection, in order to potentially mitigate the impacts of malicious attacks. In this data set which contained about 3 years of continuous energy consumption data, we performed Principal Component Analysis (PCA) to determine the best response variables to use when training our Hidden Markov Model (HMM). Through PCA, we determined that Voltage, Global_active_power, and Global_intensity to be the best three variables to use in our model. After training 10 multivariate HMM in the range of 4 to 24 states, we determined that

the best number of states in order to maximize the log likelihood and BIC values is 22. We then used this final HMM to perform anomaly detection on the three given datasets injected with anomalies in order to evaluate the effectiveness of our model, which proved to be a success with the first two datasets showing likelihoods that indicate the presence of anomalies while the last dataset showing even greater signs of containing anomalies.

6 References

[1] G. Anne and D. Sblendorio, “What should my train/test split be? - machine learning resources,” *Watchful.io*. [Online]. Available: https://www.watchful.io/resources/what-should-my-train-test-split-be