Cybersecurity Final Report

CMPT 318 - Fall 2018

Prepared by Group 8

Aarish Kapila	akapila	301269929
Che Jung (Kent) Lee	cjl27	301249826
Karan Sharma	ksa95	301238119
Razvan Andrei Cretu	rcretu	301255203
Yernur Nursultanov	ynursult	301255175

Abstract

A statistical analysis of individual household electricity consumption was performed for the purpose of classifying data points as either normal or anomalous. The dataset used ranged from December 16, 2006 to December 1, 2009 with a granularity on the scale of minutes, and containing metrics on global active power, global reactive power, voltage, global intensity, and three sub-metering variants. The dataset was partitioned according to seasons, dates, and times of day (mornings, afternoons, and evenings), of which Wednesdays and Saturdays were selected as suitable bases for weekdays and weekends. The *R* programming environment was used to explore the dataset and observe overall changes relative to the expected normal behaviour, to detect point anomalies by identifying minimum and maximum thresholds for the values of the features using a moving-average approach, and to detect contextual anomalies by training Hidden Markov Models on the dataset and evaluating their respective likelihoods on separate test datasets.

In the given dataset, global active power was found to be the most meaningful feature. The trends observed included relatively increased electricity consumption during the colder months compared to the warmer ones, and generally increasing consumption from 8:00 AM to 11:00 AM and likewise decreasing consumption from 9:00 PM to 12:00 AM. Training Hidden Markov Models to detect contextual anomalies was more practical compared to a moving-average approach to detect point anomalies in this circumstance, and this method was able to provide more substantial results.

Table of Contents

List of Figures

List of Abbreviations

BIC	Bayesian information criterion
GAP	Global Active Power
GRP	Global Reactive Power
GI	Global Intensity
HMM	Hidden Markov model

Introduction

Electricity is fundamental to the functioning of modern society and the economy. Today, critical infrastructure including electrical power grids depend heavily on automatic monitoring and automation in order to maximize efficiency and respond promptly to disadvantageous incidents

This project focuses on analyzing electrical grid data using a dataset collected from observing household power consumption, with a primary objective to reduce the risk of human-related threats by facilitating early warning and mitigation of a wide range of disasters. Situational awareness is a key concept that requires continuous detection of different types of anomalies by a behaviour-based intrusion detection system. The main challenge is to identify anomalies with appropriate probabilistic models capable of distinguishing an attack from noise such as power outages, natural disasters, etc.

This paper intends to describe the methodology which one might use to perform such an analysis when handling a large dataset and will cover the key findings from the experimental analysis and problems encountered while progress was made.

Background

This report was built upon several concepts of which one may not necessarily have a thorough understanding; to address this, such concepts shall be briefly discussed to provide a background for this paper.

Critical Infrastructure

Critical infrastructure refers to processes, systems facilities, technologies, networks, assets and services that are essential to the health, safety, security or economic well-being of Canadians and the effective functioning of government (Glässer, 2018a). Furthermore, they can stand-alone or be interconnected and interdependent within provinces and territories and across them as well as across national borders (Glässer, 2018a). Breakdown of a critical infrastructure can result in adverse economic effects, harm to public confidence, and even loss of life (Glässer, 2018a). In industrialized countries, critical infrastructure includes hospitals, railways, and nuclear defence systems, water management systems, and electric power grids (Glässer, 2018a).

Behaviour-based Intrusion Detection

Behavioural-based detection comprises of two parts. The first part defines a normal pattern of behaviour for a network or system. The second part involves continuously scanning for patterns or behaviours that deviate from the normal defined behaviour in the first part sufficiently to cause information system operators to

presume malicious activity. The deviated patterns are also called anomalies (Glässer, 2018a, 2018c).

Situational Awareness

Using behavioural-based intrusion detection allows defenders to be situationally aware. Situational awareness refers to “the perception of elements and events in the environment with respect to time and space, the comprehension of their meaning, and the projection of their status in the near future” (Glässer, 2018b). This means that one is aware of what is happening in the environment with regard to a particular subject and how information, events, and actions can impact the goals immediately and in the future.

Types of Anomalies

There are 3 common types of anomalies: point, collective and contextual. The simplest type of anomaly is a point anomaly (Glässer, 2018c). If only a single data instance can be deemed an anomaly compared to the rest of the data, then it is called a point anomaly. Collective anomalies are groups of associated data instances that are anomalous with respect to the overall data set. On the other hand, if a data instance is an anomaly in a particular context but not otherwise, it is referred to as a contextual anomaly (Glässer, 2018c).

In this paper, the data-processing pipeline will employ behaviour-based detection to examine the critical infrastructure of electricity, accounting for all of the aforementioned types of anomalies.

Datasets

Data analysis was conducted on individual electricity consumption using 1 training dataset (containing 1,556,444 instances) and 5 testing datasets (containing 518,817 instances each). All datasets contain 9 attributes, as described in Table 1.

Table 1: Attributes of household electricity consumption dataset (Hebrail & Berard, 2012)

Attribute	Description
Date	Date in dd/mm/yyyy format
Time	Time in hh:mm:ss format
Global_active_power	Household global minute-averaged active power (kW)
Global_reactive_power	Household global minute-averaged reactive power (kW)
Voltage	Minute-averaged voltage (V)
Global_intensity	Household global minute-averaged current intensity (A)
Sub_metering_1	Kitchen appliance energy usage (Wh)
Sub_metering_2	Laundry appliance energy usage (Wh)
Sub_metering_3	Air and water conditioning energy usage (Wh)

Methodology and Assumptions

Phase 1

It was decided to use a subset of all days of the week; in this case, only Wednesdays and Saturdays were deemed necessary to describe weekdays and weekends respectively. An intuitive assumption one might make is that weekdays and weekends would differ in electricity consumption trends between the two types of a day of the week, but not differ significantly when comparing two days within the same of these two classes. The data subset was then divided into mornings, afternoons, and evenings, based on similar intuition. This division produced six subsets of data based on the combinations of these attributes. Each set was then aggregated in groups with granularities of minutes, months, and seasons.

After preprocessing the data in this manner, the data was plotted on minutely, monthly and seasonal bases for the purpose of visually observing the training and test data differed both quantitatively and quantitatively.

Figure 1 (below) presents an example of Saturday morning minutely, monthly and seasonal data for training and test datasets.

Figure 1: Minutely Saturday morning for training and test dataset.

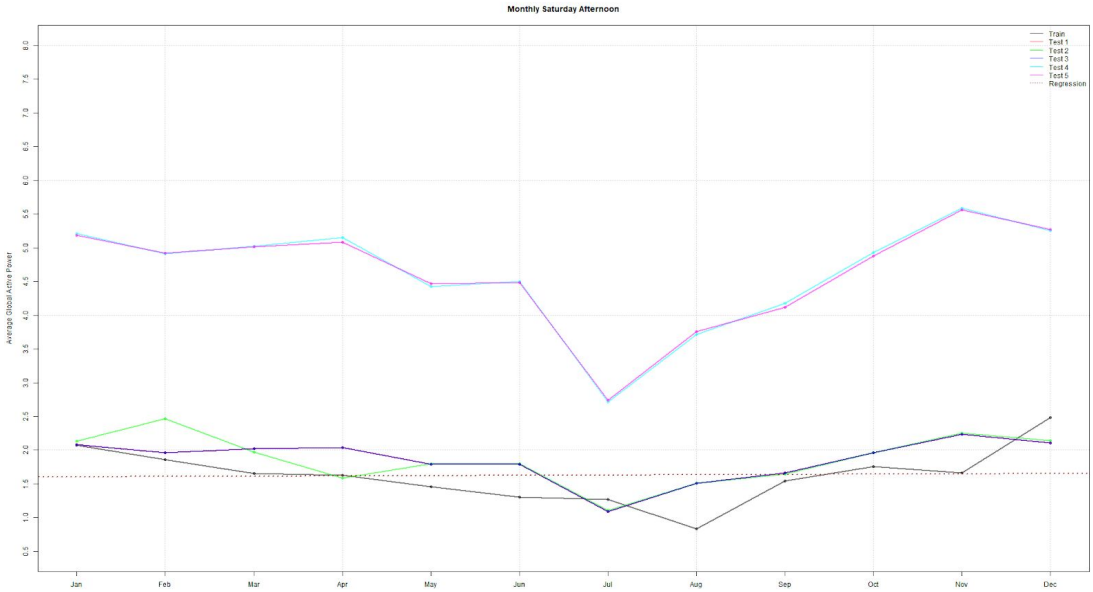

Figure 2: Monthly Saturday morning for training and test dataset.

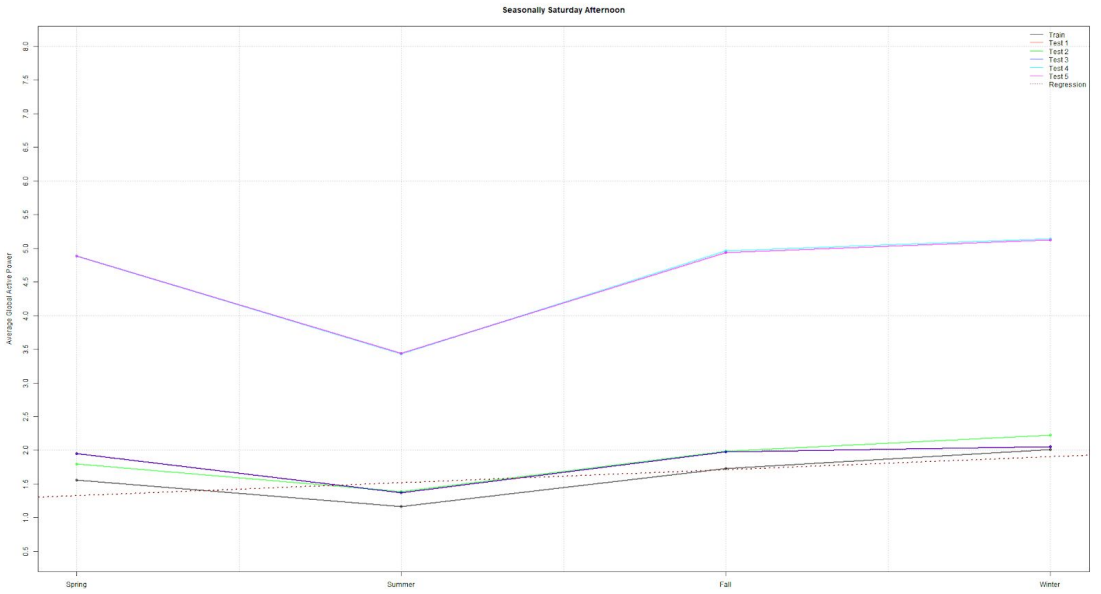

Figure 3: Seasonal Saturday morning for training and test dataset.

Phase 2, Approach 1

Point Anomalies

The point anomaly detection consists of two processes: 1) finding the average minimum and maximum GAP values from the train data, and 2) comparing it with the test data to identify the outliers. As shown in the graphs below, the points outside of the range of green (minimum) and red (maximum) lines are considered anomalies and thus are recorded in the output directory. Note that the whole process is based on minutes for accurate estimation purpose. Figures 4-6 below illustrate point anomalies for Saturday mornings, afternoons, and evenings on a minutely basis.

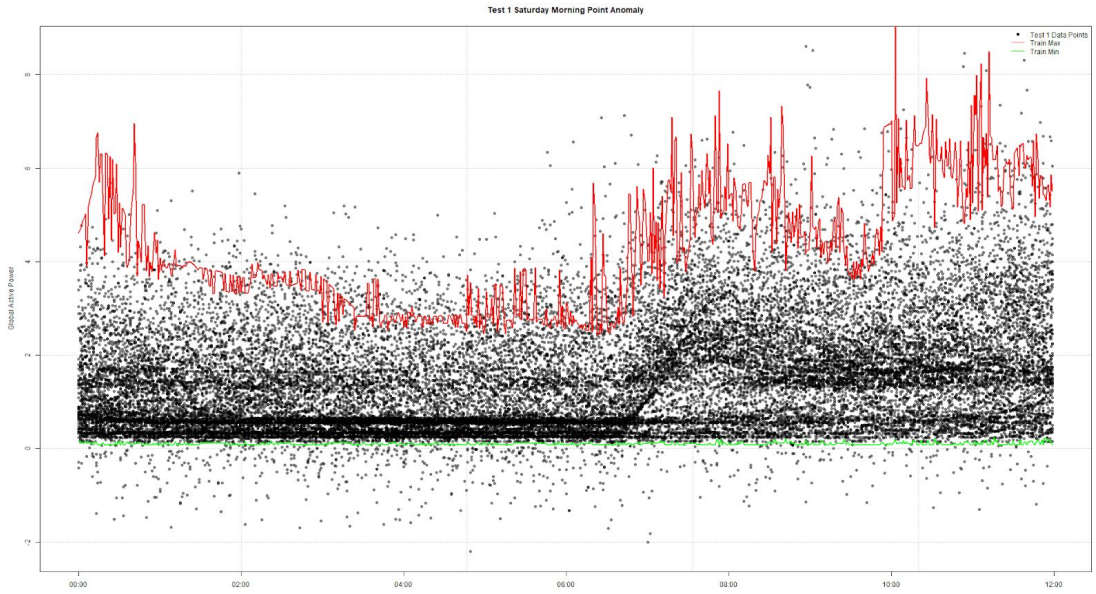

Figure 4: Test 1 Saturday Morning Point Anomalies.

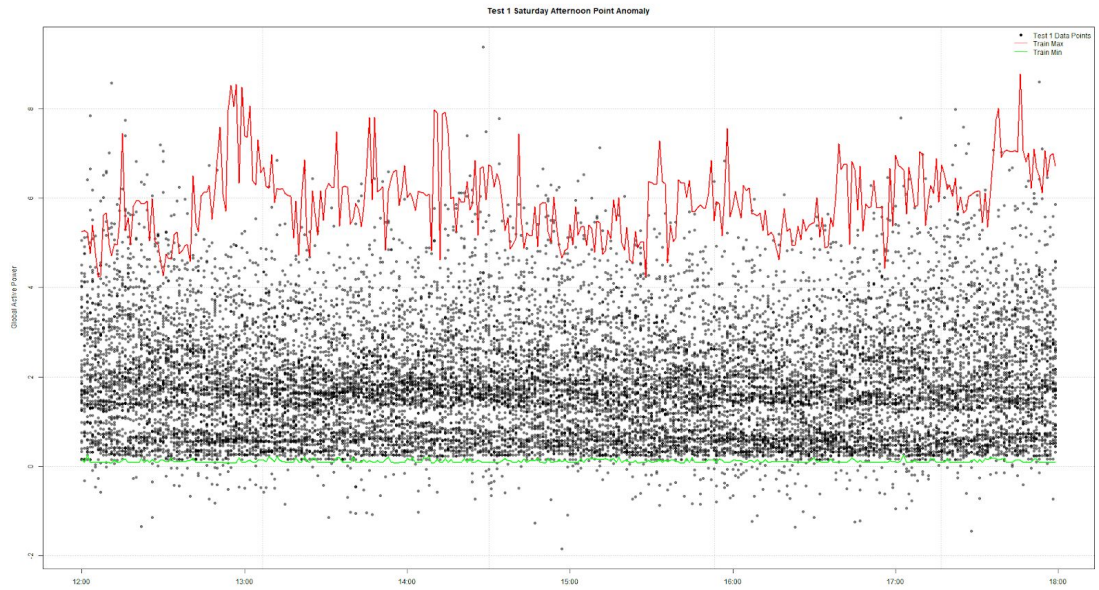

Figure 5: Test 1 Saturday Afternoon Point Anomalies.

Figure 6: Test 1 Saturday Evening Point Anomalies.

Collective Anomalies

The collective anomaly detection also consists of two processes: (1) finding the average GAP value over a given window size (set to 60 in our case). (2) sliding the window by one observation; if the difference of the observation GAP value and the calculated average is either above or below a certain threshold (set to 1.5 in our case), then it is considered as an anomaly. As shown in below graphs, the points outside of the range of green (average value obtained from the previous window - threshold) and red (average value obtained from the previous window + threshold) lines are considered as anomalies and thus are recorded in the output directory. Figures 7-9 illustrate the results of this method for Saturdays.

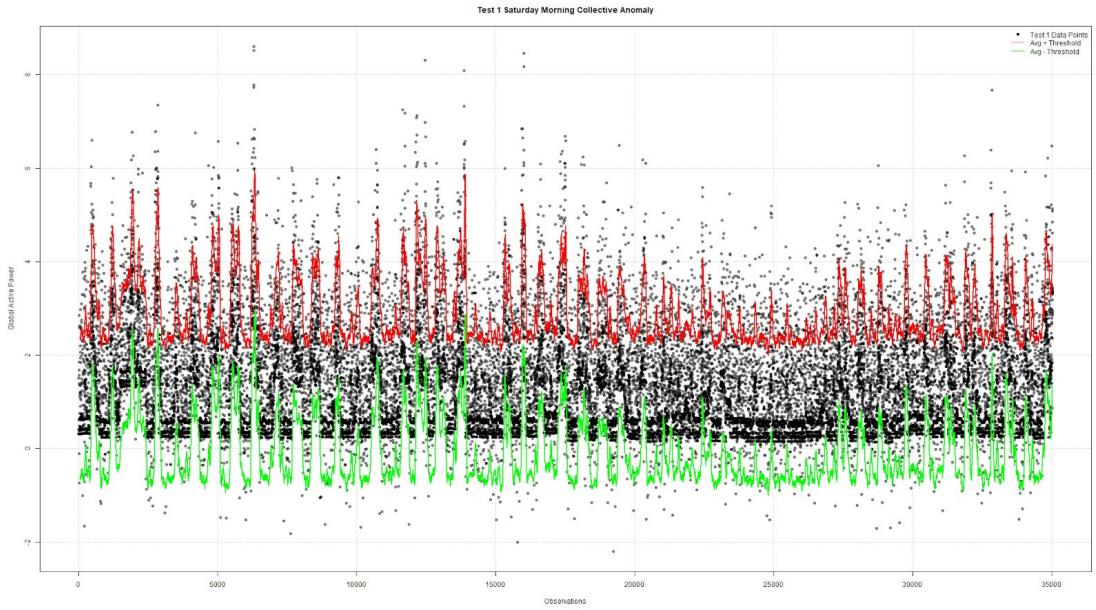

Figure 7: Test 1 Saturday Morning Collective Anomalies.

Figure 8: Test 1 Saturday Afternoon Collective Anomalies.

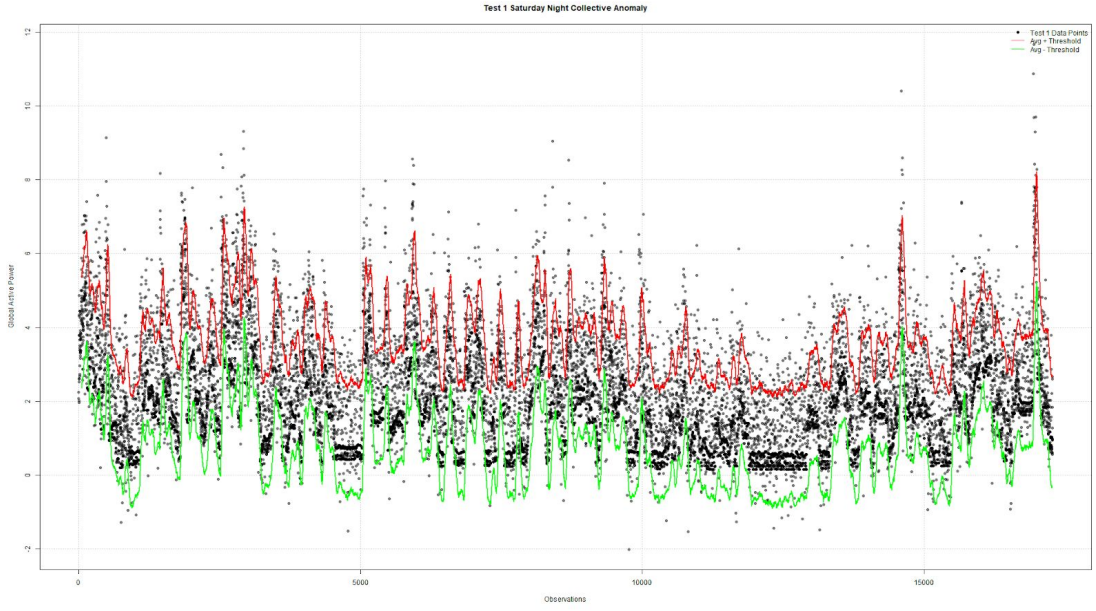

Figure 9: Test 1 Saturday Evening Collective Anomalies.

Figures and anomalous records have been produced for all the test data; the above graphs are the random examples selected to show our work. This is the same for the point anomaly detection.

Phase 2, Approach 2

In order to identify the optimal number of states to use for a Hidden Markov model, a function was designed to evaluate the GAP feature from 1 to k states. The evaluation is determined by two factors: log-likelihood (the higher the better) and BIC value (the lower the better).

The reason for the GAP feature being selected is based on the results of another function that tests all combinations of two features. The comparison criteria is the BIC pattern, which is the pattern that produces positive BIC values in mostly descending order and has the highest number of fits (given 1 to k states). As mentioned, the resulting optimal feature is the Global Active Power.

Next, HMMs were trained with the *depmixS4* package and fitted for all time periods of training data (six in total) using both the `depmix` and `fit` functions. Models for all test data were constructed using `depmix`. This creates `depmix` objects that can be updated with parameters obtained in the fitted models from train data using `setpars` and `getpars`. To calculate the probability of an observation sequence, we used the `forwardbackward` function, which allows us to obtain the log-likelihood.

So, for each time period, (1 + k) log-likelihood values were obtained: one from the train fitted model, k from the test data (five test datasets were used in this case). The log-likelihood values were divided by the appropriate time window size which

corresponded to the number of distinct observations, then the results were plotted for comparison.

Log-likelihood represents the estimation of certain model parameters which result in the curve that best fits the data. The higher the log likelihood values, the better the model characterizing the data. In theory, the results between train data and test data should be similar, with train data performing better than test data.

Results and Explanation

After training and fitting the six models (one for each time-window), the following results were obtained:

Log-likelihood of HMM on Wednesday time-windows

As seen in Figure 9 below, the log-likelihood for the training and test datasets suggest that the respective HMMs had the best performance for Test 1, Test 2 and Test 3. Both Test 4 and Test 5 had equally poor log-likelihood scores in comparison.

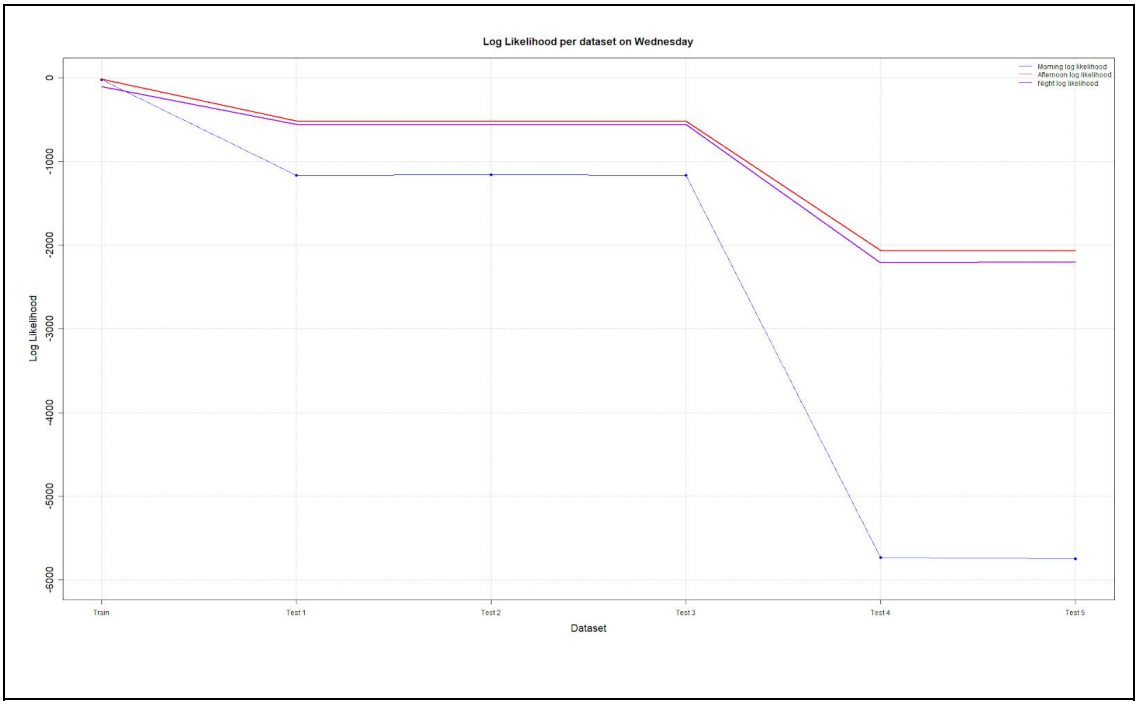

Figure 10: Log-likelihoods for Train, Test 1, Test 2, Test 3, Test 4 and Test 5 datasets (from left to right) on Wednesdays. Blue lines represent morning values, red lines represent afternoon values and purple lines represent night values.

Note that the individual line plots in Figure 10 and Figure 11 have been combined together for the sake of conciseness, rather than comparison. Since each model is investigating a different time-window (and therefore a different context), their log-likelihoods are not comparable to each other.

Log-likelihood of HMM on Saturday time-windows

Similarly to Figure 9 for Wednesday time-windows, Figure 10 conveys a comparable story regarding each model's relative performance between datasets. Once again, all three models struggled with estimating parameters for Test 4 and Test 5 in comparison with the other Test datasets.

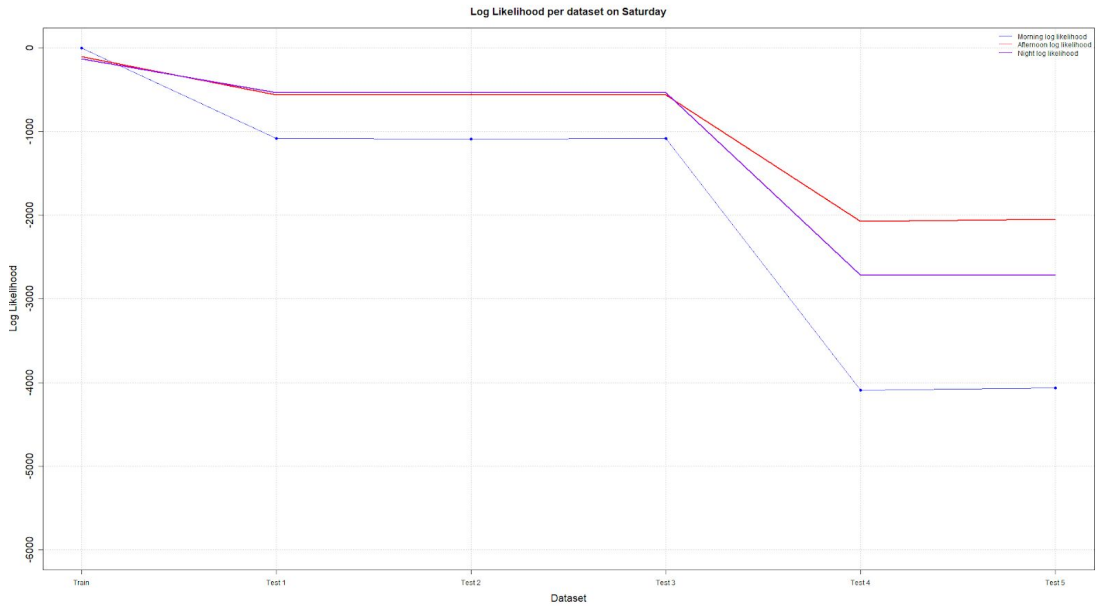

Figure 11: Log-likelihoods for Train, Test 1, Test 2, Test 3, Test 4 and Test 5 datasets (from left to right) on Saturdays. Blue lines represent morning values, red lines represent afternoon values and purple lines represent night values.

Differences in test datasets

In order to understand the consistent patterns in the log-likelihood scores, we took a closer look at the differences between the datasets and noted a few common trends.

1. The data points for the GAP feature generally had a maximum value of 12 for the Test 1, 2 and 3 datasets, whereas, the maximum value was around 25 for the Test 4 and 5 datasets.
2. As indicated by Figure 11, a huge number of data points in the Test 4 dataset were above the maximum threshold determined by the training dataset. Similar trends can be observed among other time-windows, which are replicated in Test 5.

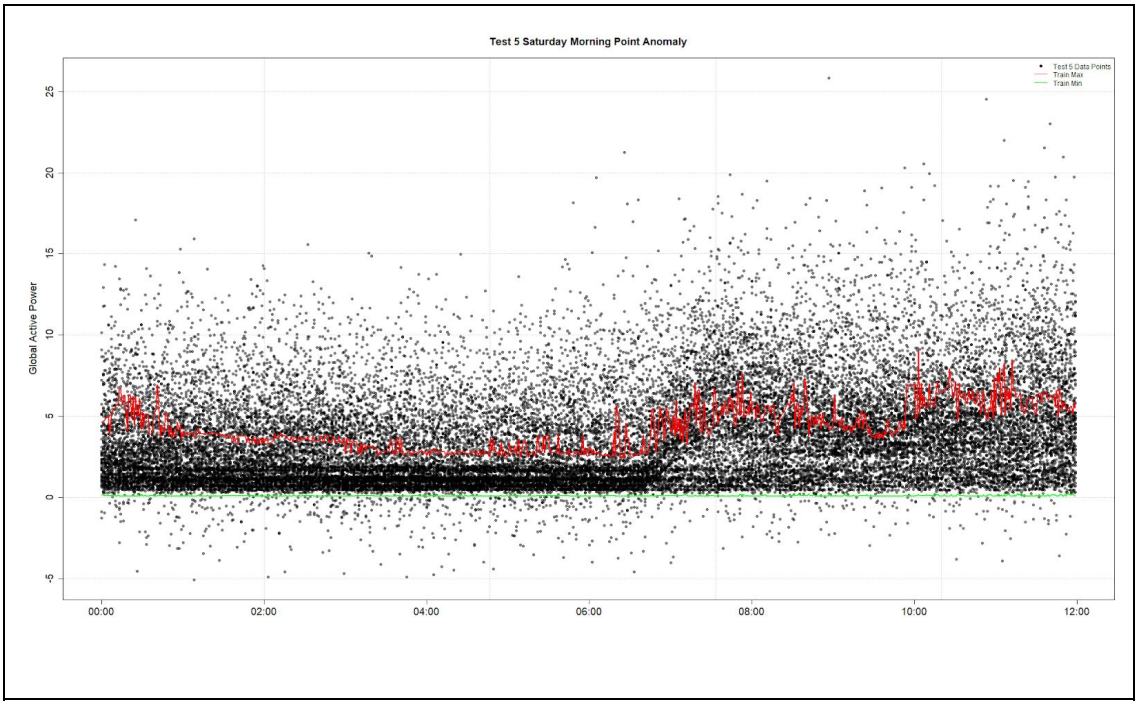

Figure 12: Test 4 Saturday Morning Point Anomalies.

Taken together, the differences in the latter 2 test datasets suggest that they do not represent the inherent trends in the training dataset as well as the first 3 test datasets. Consequently, the trained HMMs also struggled to predict the actual test instances after learning from the training instances.

Problems Encountered

Identifying point-anomaly thresholds

A problem inherent in detecting point anomalies using a moving-average approach — which became apparent as this technique was applied — is that of determining the appropriate minimum and maximum threshold values for use in such an algorithm. Visually observing the data did not provide clear suggestions for these values due to the sheer

volume of data points. Although many data points were obviously anomalous and likewise many demonstrated normal behaviour, it was difficult to determine a meaningful boundary between these two classes as a result of the inherent variance within the dataset. Put more broadly, the dataset was unlabelled, which presented the challenge of classifying the data in the face of noise.

A consideration for the future would be to examine alternative methods for detecting point anomalies. A moving-average approach has the property of being affected by anomalies, meaning an anomalous data point will push the trend by some magnitude proportional to the difference between that point and the previously-observed point. An alternative to the moving average is the moving median, which is able to exclude anomalous points from the trend instead of "absorbing" them, provided that anomalies are relatively infrequent (Moving average, n.d.).

Working around long model training times

As the team approached the final phases of the task, the script began to take a significant amount of computing time depending on the hardware used — more than an hour was needed for the script to complete (including the commented-out optimization of '*nstates*'). This time requirement limited the iterative testing of different parameters and approaches at certain parts of the script. In an effort to reduce the time spent computing, the team spent just as much time coming up with educated guesses for the next parameter combination we would try during training.

Conclusion

Over the course of this project, the team learned a lot about the practical realities of working with large datasets and the challenges of detecting anomalies. In addition to discovering the predictive strengths of Hidden Markov models, we also learned about the time constraints pertaining to hyperparameter optimization. The project also highlighted the fact that statistical models are significantly affected by incongruencies among the test and training datasets.

Lessons learned

Real data is messy

Real data is messy and noisy. We found that our dataset contained thousands of missing values and the data itself was noisy. We also found that the column 'Global_reactive_power' had a low correlation to every other column and it was difficult to work with. Lot of our work went into trying various cleaning approaches such as imputation (which was ultimately rejected in favor of omitting rows with any NA values) to ensure that our numbers were accurate and not providing any misinformation.

Hidden Markov models are powerful predictive tools

Hidden Markov models are best suited for evolving systems because the state of an evolving system uses variables that describe it not just currently, but a fixed number of

times in the past. Markov processes are evident in many observable processes, so HMMs are a logically congruent modelling choice. In our case, HMMs are apt because the household consumption of electricity is a stochastic phenomenon, where the future probabilities are determined by the most recent values (as indicated by the patterns over the day and over seasons).

Script efficiency is a priority for large datasets

Given the considerable size of the dataset, we quickly realized that the efficiency of data manipulation operations in the script matters. For example, performing multiple aggregations such as min, max, and mean over multiple calls to the R built-in "*aggregate*" function was found to be highly inefficient. Instead, the team decided to perform aggregations with the more performant and concise "data.table" package. Nevertheless, as noted earlier, computation time became a hurdle near the end, which required us to adapt our approach to scripting.

Distribution of tasks within team

- Aarish
    - Worked on phase 1 and 2.
    - Checked the data for the identified time windows.
    - Considered various trends using statistical model.
    - Contributed to the methodology and conclusion sections of the report.
- Kent
    - Worked on phase 2 approach 1.1 and 1.2.
    - Identified time windows for training dataset based on the Min/Max values.
    - Contributed to the results and explanation sections of the report.
- Karan
    - Worked on phase 2 approach 1.2.
    - Picked threshold for a moving average to smoothen the curve of that feature.
    - Contributed to the results, explanation, and conclusion sections of the report and proofread.
- Razvan
    - Worked on phase 2 approach 2.
    - Conducted research on the major problems for choosing suitable R packages.
    - Contributed to the abstract, introduction, and conclusion sections of the report and proofread.
- Yernur
    - Worked on phase 2 approach 2.
    - Helped with Classification of the datasets and approval of time periods.
    - Contributed to documenting the prob and the introduction sections of the report

References

Glässer, U. (2018a). *CMPT318: Cybersecurity, section 1 slides* [PDF slides].

Glässer, U. (2018b). *CMPT318: Cybersecurity, section 2 slides* [PDF slides].

Glässer, U. (2018c). *CMPT318: Cybersecurity, section 3 slides* [PDF slides].

Hebrail, G., & Berard, A. (2012, August 30). Retrieved November 3, 2018, from https://archive.ics.uci.edu/ml/datasets/individual+household+electric+power+consumption

Moving average. (n.d.) Retrieved November 23, 2018, from https://en.wikipedia.org/wiki/Moving_average#Moving_median