Proposal for Anomaly Detection of Energy Consumption Data

Team: Anomaly Detectors
Fei Wang
Yumeng Ding
Gautam Moogimane

Sponsor: Jones Lang LaSalle Americas, Inc. (JLL)
Linnea Paton

Table of Contents

1. Executive Summary

Anomaly detection of energy consumption is an important component in multi-family apartments' and commercial buildings' management process. Comprehensive and automated anomaly detection rules can help management companies identify potential overbilling and energy insufficiency efficiently and effectively. However, the differentiation between normal fluctuations in energy consumption due to seasonal variations and actual anomalies are not always easy and the incorrect detections can cause unnecessary investigations. In order to help our capstone project sponsor, Jones Lang LaSalle Americas, Inc. (JLL), correctly identify actual energy consumption anomalies, we propose the development of an automatic rule based anomaly detection system.

In this project, we will leverage the historical data from JLL and New York City Housing Authority (NYCHA)[1] public data to model the trend in energy consumption across buildings in New York and help JLL successfully detect the anomalies caused by infrastructure malfunction and billing errors.

Ultimately, this project will include (1) performing extensive data cleaning and data munging to understand the potential causes for billing issues and manual data entry errors, (2) evaluating existing anomaly detection rules established by JLL, (3) building predictive models with confidence intervals to detect anomalies in observed data entries.

2. Problem Statement & Objectives

In recent years, natural resource consumption and conservation have become major areas of focus across academic, industry and political debates. One of the big components of natural resource consumption comes from energy usage in commercial buildings. Therefore, the monitoring of energy usage trends and detection of abnormal activities in these public buildings are essential to a more efficient usage of these resources. Energy usage anomaly can come from many different sources, for example, error in manual data entry, broken infrastructure, seasonality of energy consumption and so on.

Our capstone project sponsor, Jones Lang LaSalle Americas, Inc. (JLL) is in charge of collecting energy usage data for properties under their management, to ensure the clients' energy usage are compliant with local energy disclosure laws and measure progress towards sustainability goals. The key to success of this task relies on the accuracy of the monthly-reported utility data. More specifically, we are tasked to output potential abnormal data entries due to either data quality issues or other factors like broken infrastructures for the team to investigate. This will help increase the productivity of JLL's analysts (i.e. time will be saved by targeting only the sites with anomalies for audits), lead to real cost savings (e.g. fixing building operation issues that are causing energy or water waste), and increase confidence in greenhouse gas sustainability reporting data. JLL has created a set of Data Quality Checking rules based on their experience with the data. Our team's objective will be to evaluate existing anomaly detection rules, advise on their statistical validity, come up with five new rules based on data analysis and potentially make suggestions on how to pipeline the detection process.

There are also numerous challenges regarding this anomaly detection problem. First, the utility bill datasets of JLL's clients are generally very small, each about 3,000 data points with 1-year's worth of historical data. This poses a significant challenge for us to model the energy consumption pattern of the clients. Second, the utility data may incorporate multiple types of energy consumption pattern changes at the same time, including seasonal shifts, additive outliers or random level shift caused by change of building occupancy or purpose. Therefore, our predictive model and anomaly detection algorithm need to separate all these above individually. Last, it's difficult to quantitatively evaluate our work. With our proposed rules to detect outliers, we need to obtain the ground-truth of energy consumption for each suspected outlier data point by working with the JLL data team, which usually take a long turnaround time for coordination. However, in order to improve our models and anomaly detection rules, feedback with real data is critical.

3. Background Information

3.1. Domain knowledge

3.1.1. Technology

The energy industry in general is undergoing a lot of advancements in terms of technology use. Artificial Intelligence and machine learning are no longer just buzzwords, but strategies that are being applied widely to improve efficiencies and reduce overall costs. In this project we are mainly concerned with anomaly detection, a remote facilities management tool, that detects outliers in a given data set that do not conform to established normal behavior. These errors in reported energy figures could be due to faulty equipment, irregular readings or manual entry mistakes and being able to automatically identify these erroneous inputs, could be extremely beneficial in reducing energy wastage.

An example of a tool that is currently being used across the industry is PEERS, a linear multivariate regression model, that takes in details like month, year, different property details as well as historical data, and comes up with a calculated model, that predicts the estimated value of an account for the current time period. Comparing this value, against the actual values that are reported, is a good way to check for variation and detect anomalies.

Another widely used strategy is to visualize the data as a time series and then split this signal into 3 different components, seasonal, trend and residual data. This way, it can automatically account for seasonality and cyclical trends in the data, and adjust for them accordingly. The residuals can then be tested with something like the Generalized ESD(Extreme standard deviation) to identify abnormal behavior.There are several inbuilt libraries that can do this in Python and R.

For the NYCHA dataset of electricity consumption and cost, we mainly focus on two types of energy usages and charges – the energy consumption (KWH) and peak power demand or 'capacity' (KW). A client needs to pay for not only the amount of electricity consumed, but also the maximum capacity of electricity delivered since the utility company need resources to offer it that capacity during its peak energy demand period.

3.1.2. Assumptions on the electricity consumption and cost dataset

1. Each entry should be uniquely identified by a combination of Building ID, Account ID, Revenue Month and billing window (including both Service Start Date and Service End Date)
2. Each entry in the dataset represents a monthly energy charges for a client, with non-negative values in both KWH and KW charges, Consumption and Capacity. It may also contain other info on billing adjustments reflected in the ‘Other charges’ attribute.
3. The sum of KWH charges, KW charges and Other charges should be equal to the value of Current charges. Consumption value and KWH charge value should be both positive or zero, so do Capacity value and KW charge value.
4. There are two levels of granularities of business entities – building level and account level. A building can be uniquely identified by its ‘TDS #’ and ‘Location’ values; An account can be uniquely identified by a ‘Meter Number’ value. Each building may contain multiple distinct meter numbers. Sometimes under the same account, the meter number may get updated by another.
5. Each account should contain both non-negative KWH and KW charges, consumption and capacity.
6. In case there is rebilling for the same billing window, there will be multiple rows per Building ID, Account ID, Service Start Date and Service End Date.
7. For any account there might be overlaps or gaps between billing windows. Any overlap or any gap longer than 3 days should be considered problematic.

3.2. Literature Review

Our utility billing data is time series data in nature. Generally speaking there are two categories of approaches to detect anomalies in time series data. [2]

Category I: Find anomalous data points directly from the time series with statistical tests

- STL Decomposition

    - Split the time series into 3 parts: seasonal, trend and residual
    - Analyze the residual part with Generalized ESD (Extreme Student Deviation) Test
    - Implementation example: Twitter's AnomalyDetection library
    - Pros: simple, robust, good for detecting additive outliers
    - Cons: rigid (can only tweak confidence interval), can't detect level change (which can occur often for energy consumption if a building's occupancy changes - baseline adjustments that Linnea mentioned)

Category II: Make predictions based on historical data with confidence intervals; compare the observed values with predicted values and use statistical tests to check if the observed data points lie inside of the confidence intervals. Those lie outside would be considered outliers.

- Classification and regression trees (CART)
    - Classify anomalous/normal points (need labels)
    - Predict the next points with confidence interval —> compare with real data (using G-ESD test or Grubb's test)
    - Implementation example: XGBoost Library
    - Pros: flexible; Cons: need careful feature selection

- Neural Networks (supervised and unsupervised??)
    - RNN such as LSTM
    - Pros: flexible; Cons: need careful feature selection

- ARIMA
    - Forecast using past data and compare with the real data
    - Cons: need parameter selection, signal must be independent on time
    - Implementation example: tsoutliers R package

- Exponential Smoothing
    - Holt-Winter's Seasonal Method
    - Cons: can only have one seasonal period (such as one from weekly, monthly, yearly)

General Tips [3]:

1. Try the simplest model and algorithm that fit your problem the best.

2. Switch to more advanced techniques if it doesn't work out.
3. Starting with more general solutions that cover all the cases is a tempting option, but it's not always the best.
4. Use different combinations of techniques starting with STL and ending with CART and LSTM models

4. Work to date

4.1. Data Summary

We have been using the NYCHA dataset as sample data for data evaluation since our sponsor is working on anonymizing the actual dataset. NYCHA dataset should be similar in format to the actual dataset.

The raw data file from NYCHA includes monthly electricity consumption and cost data for buildings in New York City from January 2010 to June 2018. The complete dataset has 313,147 rows and 27 columns. There are three sets of features in this dataset: there are 13 variables used to identify building information such as Development Name, Borough, Account Name and Meter Number; there are 8 variables used to describe individual billing information such as Bill Id, Revenue Month, Service Start/End Data, and Rate Class; lastly, there are 6 variables used to identify energy consumption and cost information associated with each bill, and the 6 variables are current charges, consumption in KWH, consumption in KW, KWH charges, KW charges and Other charges. These 27 features together are used to uniquely identify each bill entry and describe the amount of consumption by buildings in New York City.

However, we are not able to uniquely identify individual energy bill from any one of the 13 variables associated with each bill as all of the account names and development names can not describe 100% of the data entries. In the following section, we will be generating a primary key by concatenating several building identification variables.

Moreover, after consulting our sponsor, we learnt that the aggregation of KWH and KW charges is a better overall estimator for cost. Therefore, we will be adding two calculated fields for better comparison of the consumption cost and consumption rate. These calculated fields will later be used to model the overall trend of charges across buildings in New York.

4.2. Summary of Data pipeline

We took the following steps to clean and transform the data for model fitting and anomaly detection:

1. General Data Cleaning

First, we cleaned the raw data to exclude rows with null account name, duplicated values and estimated electricity charges since they will not have accurate predictive value for our model. Secondly, we cleaned data types of variables regarding electricity consumption and charges from string to float. These variables will be our main metrics to draw statistical conclusions and detect energy consumption anomalies from. Lastly, we converted the time-related variables such as Revenue Month, Service Start Date and Service End Date to datetime type in python for processing and analysis. We will base off these variables to detect overlaps and gaps in billing windows.

2. Create and Validate Primary Key

Based on the metadata info of the NYCHA public dataset, we used the combination of 'TDS #' and 'Location' to create 'Building ID' as the unique identifier for each building. However, 'Building ID' alone is still not the primary key for each data entry, we are able to uniquely identify over 99.8% of the data entries by combining 'Building ID', 'Meter Number' and 'Revenue Month'. This aligns well with our understanding of the 2-level data granularity at building and account levels. The remaining 0.2% of the entries are caused by rebilling (two entries for the same billing window of the same meter) and invalid entries where all values associated with consumption and charges are zero.

3. Check validity of data entry logics and Meter Number Mapping

We found the percentages of rows with zero values in our metrics of interest were surprisingly high:

- percentage of rows - current charges of zero: 16.61%
- percentage of rows - kw charges of zero: 41.13%
- percentage of rows - kwh charges of zero: 33.03%

When aggregating the metrics based on meter numbers, it seems that some meters only record either the KWH charge or the KW charge. Following is a break-down of meters by type:

- percentage of kw_only meters: 27.66%
- percentage of kwh_only meters: 38.20%
- percentage of kwh_and_kw meters: 34.13%

52.79% of the buildings have both kw_only and kwh_only meters. For these buildings, we need to merge some of their meters so that both kw and kwh charges are represented in the same meter account.

By further exploring the dataset, we found there are many cases where under the same Building_ID, two Meter_Numbers share almost the same digits (except for the 1st digit) and billing windows of all years. Most of the time, one meter has zero values in all KW charge and one has zero values in all KWH Charges. It seems the two Meter_Numbers belong to the same account and therefore reasonable to combine them.

Furthermore, by comparing the billing months, we noticed around 13% of the meters have been replaced by newer meters over the year under the same building. Combining them further reduced the number of meters of invalid types.

In the end, our statistics improved as follows:

- percentage of rows - current charges of zero: 4.17%
- percentage of rows - kw charges of zero: 19.90%
- percentage of rows - kwh charges of zero: 8.47%
- percentage of kw_only meters: 2.16%
- percentage of kwh_only meters: 20.62%
- percentage of kwh_and_kw meters: 77.23%

The percentage of meters that do not have KW Charges is still quite high (21%), we need to further consult with our domain knowledge expert to figure out how to handle that. All other metrics appear reasonable.

4. Add new calculated metrics

After all the above data cleaning steps and with all the caveats of the edge cases, we aggregated the data to the two target levels:

- Building_ID + Revenue_Month
- Buildling_ID + Meter_Number + Revenue_Month

Per suggestion of our sponsor and domain knowledge expert Linnea Paton, we added two calculated fields for our predictive model and anomaly detection.

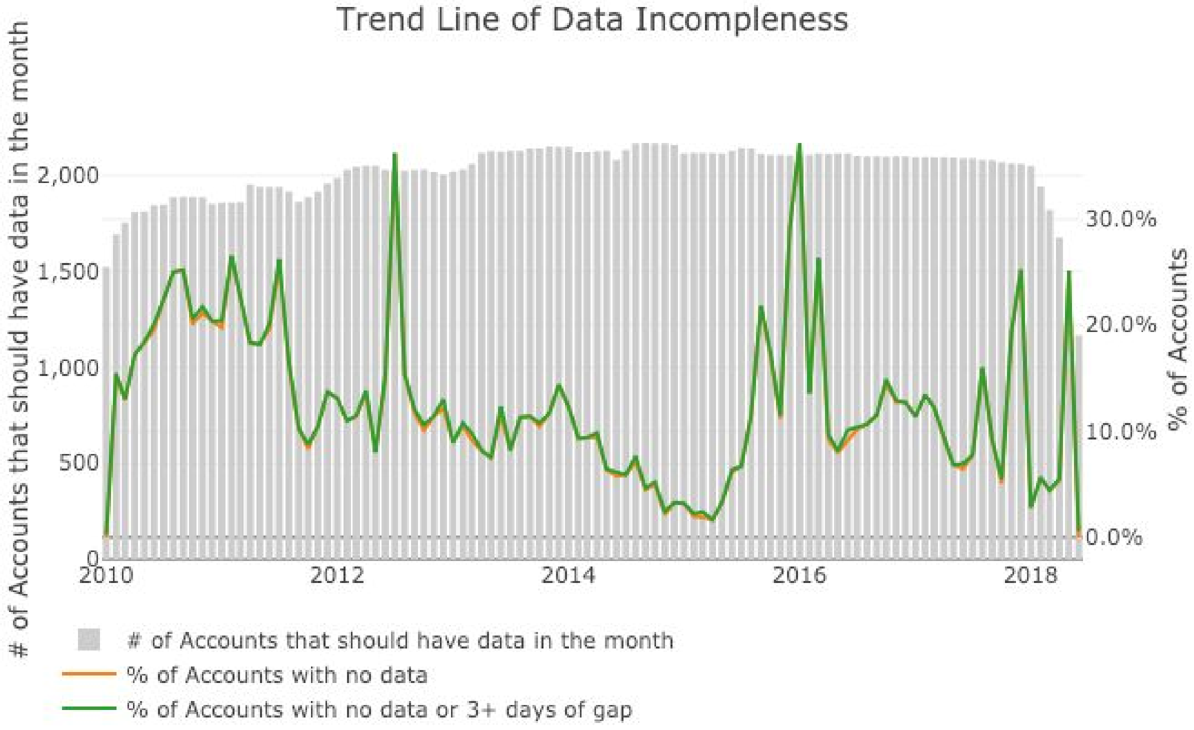
- 'Total Charges' is calculated by adding 'KW Charges' to 'KWH Charges', this field will provide a more comprehensive trend of total consumption cost.

- 'Total Energy Rate' is calculated by dividing 'Total Charges' by 'Consumption (KWH)', this field is the industry standard measure for evaluating consumption efficiency.

4.3. Qualification of Available Data

4.3.1. NYCHA electricity consumption and cost data

This public dataset from the New York City Housing Authority (NYCHA) is of good quality in terms of data cleanliness and consistency. After a few round of data cleaning and transformation, we reached a point where only a very low percentage of entries doesn't meet our assumptions of the logic of the dataset. In future iterations of analyses we plan to exclude those entries. The only concerning issue now is data completeness. We noticed majority of the accounts (83.3%) do not have full coverage of the billing months of interest (Jan 2010 - Jun 2018). 12.7% of them have gaps more than 3 days within a revenue month, as indicated in the green trend line the in chart below.

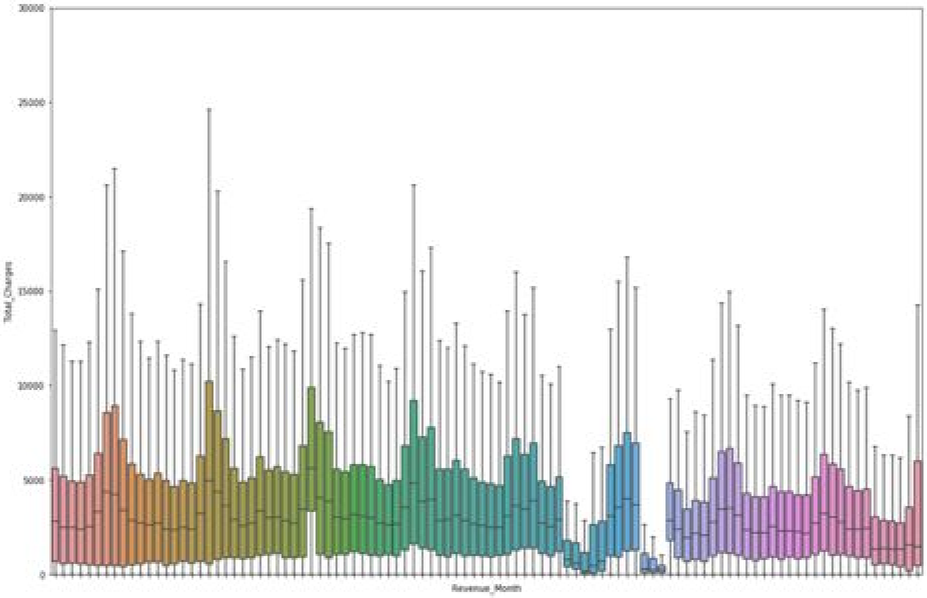


4.3.2. Other Data Sources

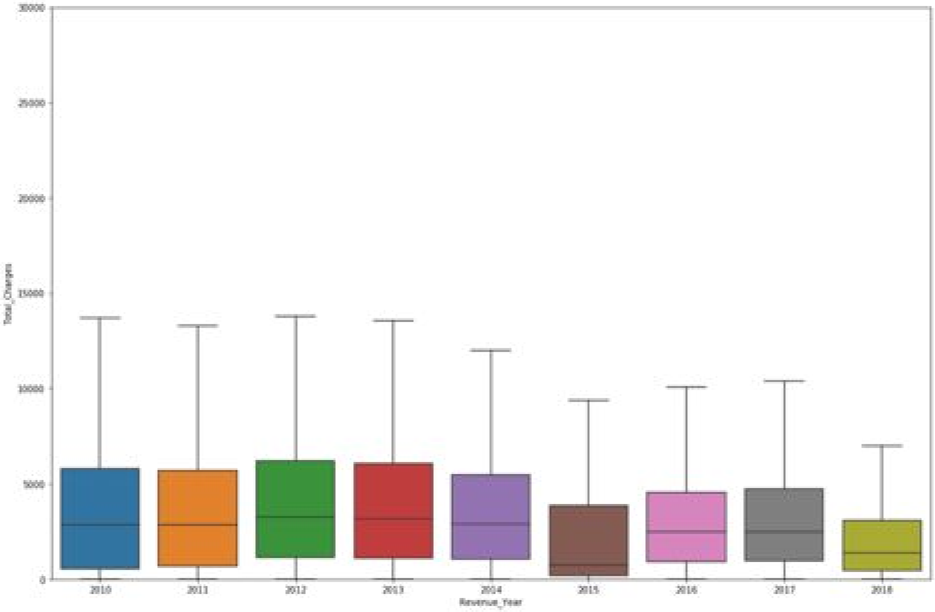
We also plan to leverage other data sources such as the monthly utility billing data of JLL clients. The data quality is expected to be slightly worse than the NYCHA dataset as they are collected from various utility companies whereas the NYCHA dataset is solely from one utility and thus cleaner. Although having got access to the JLL clients datasets, we expect them to contain more energy-related attributes such as building occupancy and sq. ft, and other types of energy consumptions such as gas, fuel oil and water.

4.4. Graphical and Statistical Analysis of the Data

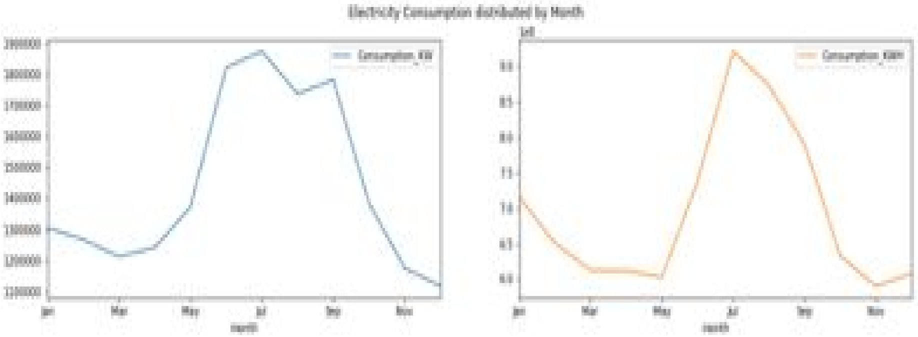
To aid in our understanding of the data we created multiple graphs along the way. To start with we wanted to get an idea of how the total charges varied according to the revenue month. We created a time series graph that plots the total charges on the y axis and the revenue month on the x axis, for all 8 years of data. It is easy to see the variation in the electricity consumption, and how it changes based on the season.



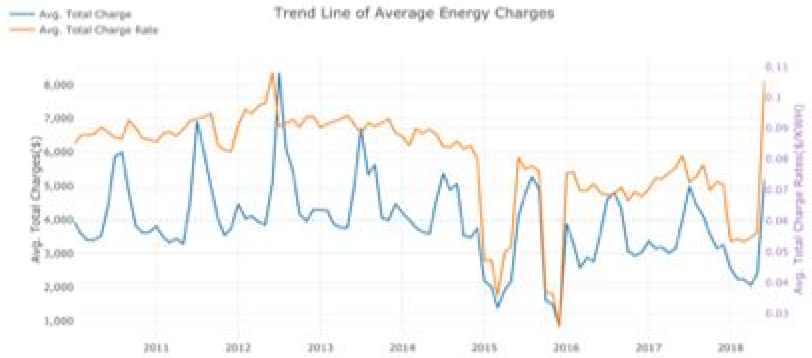
The boxplot below plots the same total charge on the y axis, but instead of the month, we now consider the entire year. If we look at the trend, there is a slight decrease in the overall electricity consumption, especially post 2013, with a significant drop in 2015. This could be an anomaly that warrants further investigation.



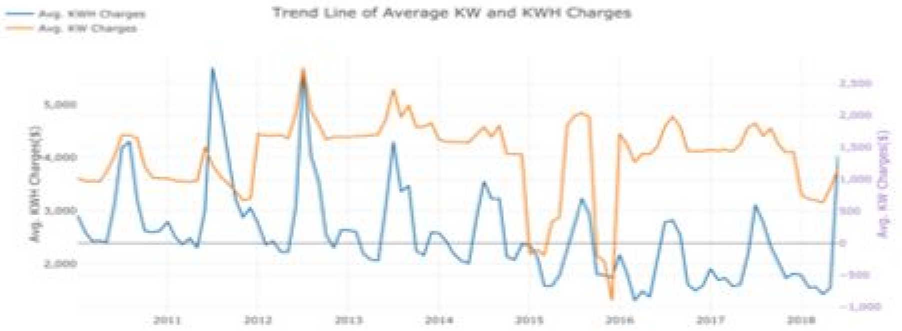
Another seasonal analysis that we did, was to segregate the months out from the cleaned dataset, aggregate all the values for electricity consumption (KW and KWH), and visualize the trend of how the consumption varies on a month by month basis. The plot on the left charts the KW consumption and the one on the right KWH. The graphs are similar in their peaks and lows, and it is easy to spot a trend of higher consumption during the summer months.



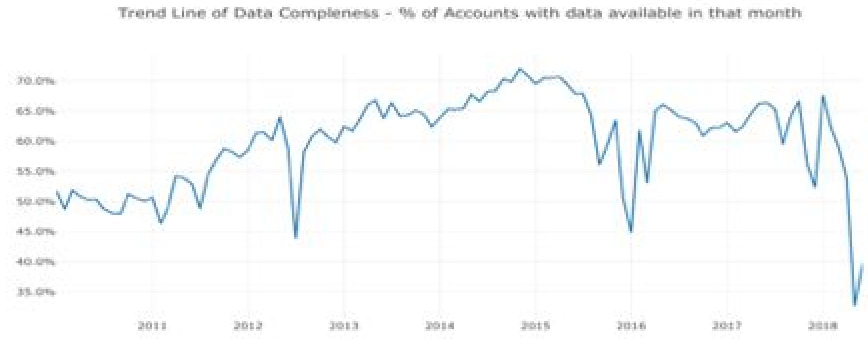
We also wanted to look at the trend lines for total energy charge (KWH + KW charge) and the total energy rate (total energy charge/KWH consumption) and how they varied over all the months. These 2 metrics follow a similar pattern within each year, with higher values during the summer and smaller peaks in the winter. There is again a drop in 2015 and 2016 that seems to be slightly more than what we expected.



The graph below highlights the trend of how the average KW charges($) and KWH charges($) for all accounts has varied over the years. Slightly concerning is the drop in the average KWH charges post 2015, where for a significant portion of the year, it is in negative territory.



For each meter number, we concatenated all the data entries whose service data ranges are subsequent to each other and there is no gap in between. With that we found that only 0.1% of the accounts have a gap of more than 5 days in a single revenue month. However, when we consider all the months across years, many accounts have missing data for multiple months. 83.2% of accounts have missing data that amounts to a gap of over 5 days, at some point or the other over the course of 8 years (see chart below). As far as overlaps are concerned, only 0.71% of the accounts have that issue, mostly due to incorrect logging of service dates.



4.5. Concerns and Limitations of the data

Since we started cleaning and analyzing the public dataset from NYCHA, we have been removing and combining records that do not meet our data cleaning rules such as gaps in recording periods or insufficient data after certain time period. We removed and combined almost 50% (from 313,147 data entries to 180,020 data entries) of the data points due to data validity concerns. We were only able to perform this rigorous data validity checking because of the sufficient amount of data points from the public dataset collected since 2010. However, the public dataset is long in historical data points but not wide in features. After combining columns to create primary key for the dataset and creating calculated fields, we have 17 variables in the dataset as features.

On the other hand, we expect our actual data from the sponsor to be wider in features with limited historical data points. As communicated by our sponsor, we should be expecting two datasets, one with 50 buildings with minimum 12 months data and the other one with 150 buildings with the same limitation on time frame. The actual data will have more descriptive features on the building for us to categorize different building types. However, we may not have enough data points to build time series trend if we only have 12 months of entry. Our potential work around this limitation will be leveraging more of the NYCHA dataset for time series and using actual data for feature generations.

5. Design and Modeling Approach

5.1. Introduction of General Approach

5.1.1. Predict confidence interval band based on past data

One of the existing rules drafted by the client, and which we think will be a key step to identify anomalies, is comparing the prorated data - which includes the ELF and baseline adjustments, for each account per month, to the data from the previous year for the same period. If the difference is beyond a certain threshold, the account can be flagged as an anomaly.

ELF and baseline adjustments refer to the manual additions made by the data analysts for cases where the electricity consumption is expected to be very high or very low. For instance, if the client building is unoccupied for a couple of months, it can be assumed that there will be a significant drop in the consumption values for these months. To avoid this account from being flagged as an anomaly, the analysts manually make an adjustment according to what they expect the consumption to be.

Another comparison that can be made is between the actual prorated data and the value calculated by the PEERs energy model, which is a multivariate linear regression model, that accepts certain parameters as input and makes a prediction of what the consumption should be based on these inputs. If there is a significant difference between these 2 values, it warrants further investigation. As of now, we are not sure if this estimated value generated by the model will be part of our actual dataset, as an additional field.

5.1.2. Analyze residuals of actual data after SLT decomposition

A potent method to detect outliers is splitting a time series signal into 3 components seasonal, trend and residuals. A seasonal component could be associated with any cyclical pattern that causes variation in the electricity consumption, for example weather or even if the building has some unique characteristic that causes ups and downs. The trend component, just as the name suggests helps us identify known trends in the data. After these 2 components have been accounted for, we are left with the residuals which can then be analyzed for outliers. There are packages available in Python that can help with the implementation.

5.2. Statistical Model

5.2.1. Detailed steps for analysis

The data would need to be cleaned and prepped for analysis using Python, following which it can be aggregated based on account numbers. The first step to identify anomalies is to prorate the electricity consumption data, for every calendar month in the year and for all accounts/meters in the data. By this we mean that even if the data for a consumer is not complete, or missing a few days in the year, the estimate we generate will ensure that we have a rough idea of what the consumption should be in lieu of that missing data. We could look at multiple strategies for estimation, with mean or average probably being the most straightforward and effective in this scenario. So, for a given month, if we can find the average per day consumption, which includes values from all the accounts for that month, and multiply that by the number of days in the month, it should give us a reasonable estimate. This can then be compared to previous years data, and any variation beyond the stipulated threshold can be flagged as a potential anomaly.

The second step involves identifying accounts with incomplete data. The criterion for this evaluation, is any account that has data missing for more than 3 days in a calendar month for the current fiscal year. This helps us get an overall picture of how the data is distributed and how many accounts have data that is missing. In other words, it also helps us establish how much of the comparisons are due to our estimations completing the data.

5.2.2. Detailed technology tools and packages

We intend to perform most of the analysis using Python. For statistical tests and splitting the data into STL components, the stldecompose package, and additive models like Prophet have been found to be extremely useful. Another implementation is twitter's Anomaly detection library. Even though this was originally implemented in R, there are open source ports available that have implemented this in Python using the open source pyloess library.
The Statsmodels library can be used for ARIMA forecasting in python, and Scikit learn has elaborate packages for implementing classification and regression tree models.

Language:

- ○ Python (Jupyter Notebook) for data cleaning and model building
- ○ SQL for Microsoft SQL Server pipeline implementation of the rules in sponsor's system, if time permits.

If the number of data points in the actual data, is found to be insufficient to generate stable models, we will use publicly available data (for example NYCHA data), with similar fields as our actual data, to generate the anomaly detection rules. These rules will then be tested on the actual data to verify its effectiveness.

5.3. Techniques and Specs

5.3.1. Potential techniques from Literature Review[4]

5.3.1.1. ARIMA Time series modeling on longitudinal data from NYCHA

ARIMA or Autoregressive Integrated Moving Average is a time series forecasting method for univariate data, that is data that has no external factors affecting it. It is mainly used for making predictions about the future using past or present data. The parameters can again be split into 3 components, seasonality, trend and noise. While ARIMA works well with autoregressive and moving average elements, it does not support seasonal data, that is a time series with a repeating cycle. For that, we can use SARIMA, which unsurprisingly stands for Seasonal ARIMA. The SARIMA time series forecasting is supported in python using the Statsmodels library.

5.3.1.2. Classification and Regression Trees (CART)

Classification and Regression Trees or CART for short is a term used to describe decision tree algorithms that can be used for classification or regression predictive modeling problems. This is a simple binary tree representation of the data, where we have root nodes with input variables, and leaf nodes with output variables. Some ensemble algorithms developed from the same foundation include random forest, bagged and boosted decision trees. Implementation should be fairly straightforward using the Scikit Learn library and its inbuilt functions.

5.3.1.3. Exponential Smoothing (Holt Winters seasonal method)

This is an exponential smoothing algorithm to implement forecasting with trends in the data. The way it works is by assigning a weight to historical data, based on how recent it is. So data that is 1 month old will be assigned a higher weight as compared to data from a year ago. The Holt Winters method is suitable for data with both trends and seasonalities and includes a smoothing parameter. The seasonal component can be expressed using 2 methods, the first being additive where the variations are roughly constant, and the other is multiplicative, where the variations change proportionally to the level of the series. Implementation in python is again supported by the Statsmodels library.

5.3.1.4. Extreme Studentized Deviate (ESD method)

The generalized extreme studentized test is used to detect outliers in univariate data that resembles a Gaussian distribution. The test is an improvement over the Grubbs test and the Tietjen Moore test, where the number of outliers must be specified beforehand, whereas with the ESD, we only need to mention the upper bound on the number of potential outliers. It then tests the null hypothesis that the data has no outliers vs the alternative, where we have n outliers, which is the upper bound specified by the user.

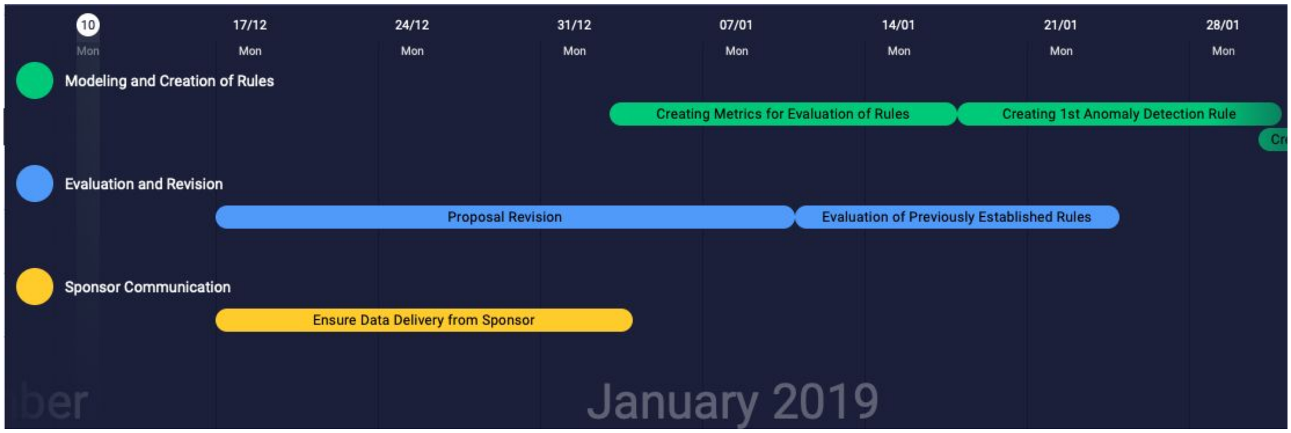
6. Project Schedule and Milestones

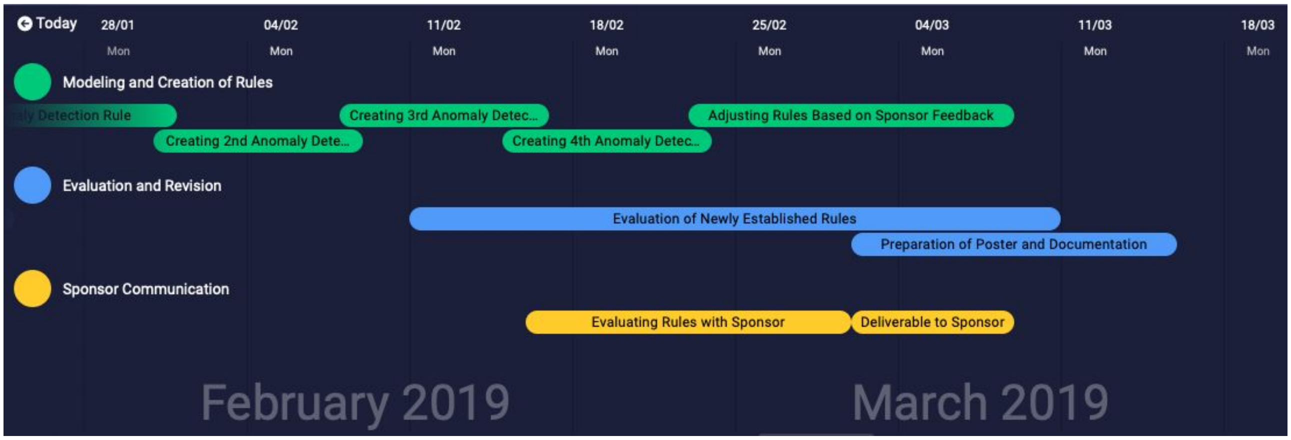
6.1. List of milestones

1. Identify the gaps in billing windows and months of missing values, both at individual account level and as a whole
2. Design rules to detect abnormal KWH consumptions and charges
3. Design rules to detect abnormal KW consumptions and charges

6.2. Project Schedule

We have created a Gantt Chart using Monday.com to track our progress in three main categories: Modeling and Creation of Rules, Evaluation and Revision and Sponsor Communication. The project calendar tracker will reset from December 17th, 2018 and last through March 14th, 2019, with deliverable due to sponsor by March 7th and a week after that set aside for poster and documentation. We will also be making adjustments to the progress tracker based on the workload.





7. Team Qualifications

Fei is pursuing his Master's Degree in Data Science at the University of Washington and has previously worked at Microsoft on their data related projects. Fei is looking forward to learning about anomaly detection methods and how to deploy them in industrial context through this project. His main responsibilities for this project will be exploratory data analysis, algorithms considerations and project coordination.

Yumeng is pursuing her Master's Degree in Data Science at the University of Washington and currently interning as a Data Scientist at T-Mobile. Yumeng is looking forward to learning about anomaly detection methods and gaining domain knowledge in energy management industry through this project. Her main responsibilities for this project will be exploratory data analysis, documentation and problem/scope definition.

Gautam is also a Masters student in the Data Science program at University of Washington, and has multiple years of work experience in the tech industry. His most recent role was working as a technology consultant with Infosys, with a focus on back end infrastructure and databases. Having never worked on Anomaly detection or Energy data previously, this is a good opportunity to gain experience in this sector. He intends to help with all facets of the project, with an emphasis on data engineering, wranging and model building.

8. References

[1] New York City Housing Authority Dataset:
https://opendata.cityofnewyork.us
[2] Time Series Anomaly Detection Algorithms
https://blog.statsbot.co/time-series-anomaly-detection-algorithms-1cef5519aef2
[3] A closer look at time series anomaly detection
https://www.anodot.com/blog/closer-look-time-series-anomaly-detection/
[4] Anomaly Detection: A Survey
http://www.cs.umn.edu/sites/cs.umn.edu/files/tech_reports/07-017.pdf