# Power Laws: Detecting Anomalies in Usage

By PINGANAI

## Summary

The objective is to provide methods for detecting abnormal power overconsumption data in buildings. There are three sites with different kinds of meters, providing auxiliary information like weather, metadata, cos-phi, etc., and we create different models for each site respectively. We first tackle the problem as a supervised learning, and choose XGBoost [1] to build regression model; then we label anomalies using rules, such as 3-sigma, with the residuals. As we notice that anomalies detected in this way only include weekday anomalies, we adopt another unsupervised model Isolation Forest [2], which is very effective in detecting weekend anomalies. We also propose other abnormal patterns which focus on the shape of the daily usage curve itself instead of the power values.

# 1. Data Exploration

The three meters' data overview is listed as follows:

| meter_id | meter_type | period | # site_meters |
|---|---|---|---|
| 234_203 | virtual main | 2013/11-2018/01 | 19 |
| 334_61 | main meter | 2015/06-2017/09 | 2 |
| 38_9686 | main meter | 2010/08-2017/12 | 166 |

## 1.1 Dig into meter1（234_203）

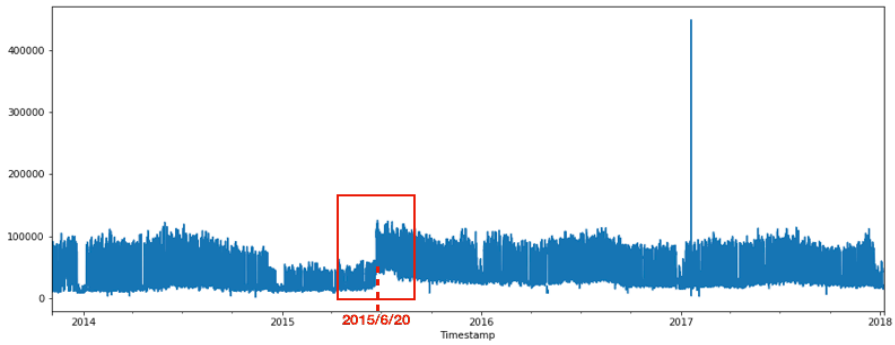



Figure1: Electricity consumption of meter 234_203 on an hourly basis

As Figure1 shows, there exists a sudden increase on the electricity consumption. By examining the other two main meters of site 234_203, we found that the virtual meter 234_203 is actually combined of meter 863 (the period before 2015/6/20) and meter 938 (the period 2015/6/20). So we decide to **detect anomalies in these two periods separately**.

## 1.2 Dig into meter2（334_61）

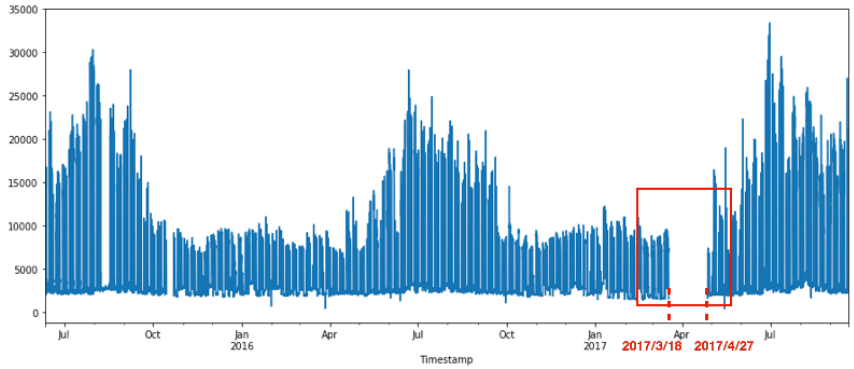



Figure 2: Electricity consumption of meter 334_61 on an hourly basis

As Figure 2 shows, there is about one-month data missing from 2017/3/17 to 2017/4/26. We will skip the one-month period in anomaly detection.

## 1.3 Dig into meter3（38_9686）

The consumption curve of meter3 is quite weird compared to the other two meters. We spent a lot of time in understanding the electricity background information of site 38, and try to give out our insight here:

- There are four main meters in site 38: the **apparent power meter** (38_9686), the **demand power meter** (38_9687), the **reactive energy meter** (38_9688) and the **cosphi meter** (38_9689).
- The apparent (also reactive) power meter is an accumulative meter, its first-order difference represents the real electricity consumption, which is shown as below:

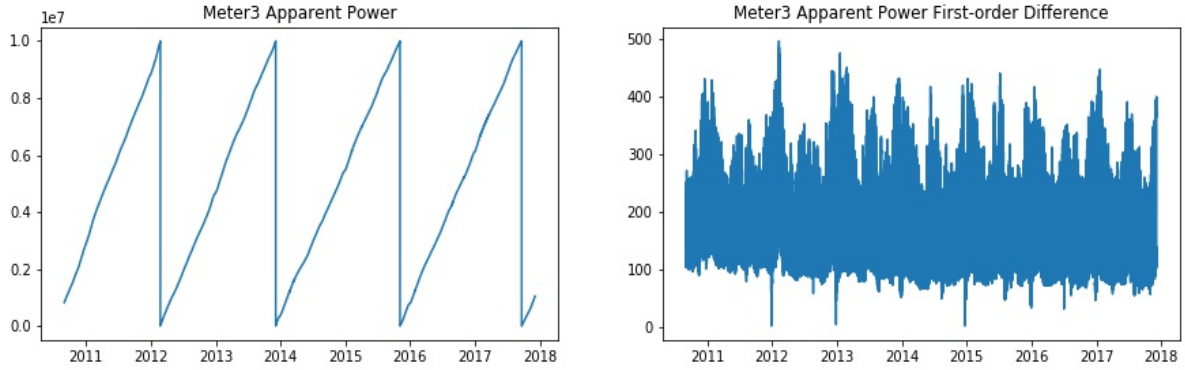



Figure 3: Electricity consumption of meter 38_9686 on an hourly basis

- There exists phase difference between the apparent power meter and the demand power meter (more obvious for the latter years, even one-day difference in 2017), and the reactive energy is consistent with the latter, as Figure 4 shows, thus **we decide to use the demand power meter data** as our model input.

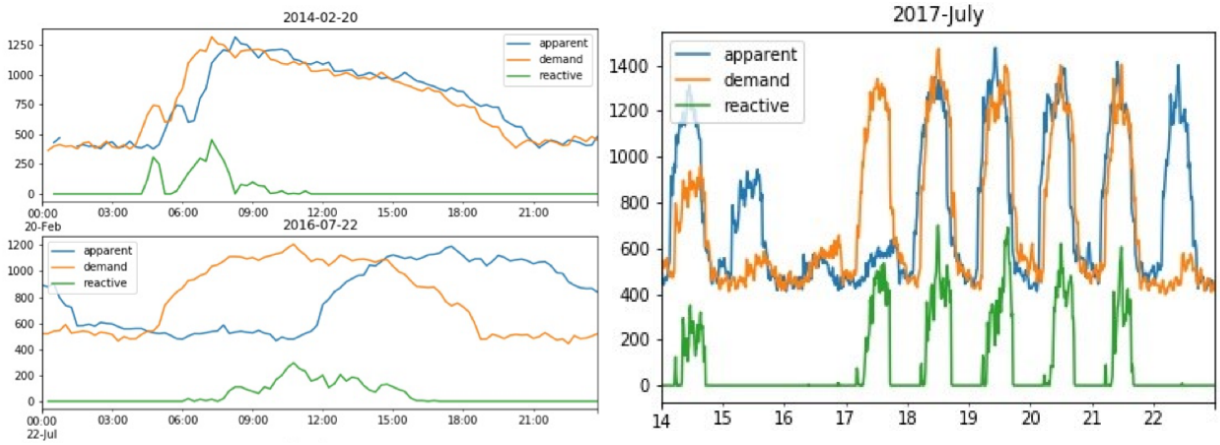


Figure 4: apparent, demand, reactive meter comparison of site 38

# 2. Feature Engineering and Modeling

As the electricity consumption data are time series, there is also other auxiliary information like temperature, holiday, and site metadata, we decide to model the problem in three directions:

- **Raw time series prediction**
  - Use ARIMA prediction to predict the future values based on past values, and extract features like rolling mean and moving average;
  - Then detect anomalies based on the error range of the predicted values and the real values.
- **Regression based on supervised model (XGB)**
  - Use the environmental and date-based features, like **month, week, hour, temperature, etc**., to build the supervised model for predicting electricity consumption values;
  - Detect anomalies based on the residuals of predicted values and real values.
- **Anomaly detection based on unsupervised model (Isolation Forest)**
  - Use a combination of features which can be extracted from the time series and the additional information, build unsupervised model to identify anomaly.

From our practice, the model based on raw time series prediction does not perform well. The reason may be that time series prediction usually lags behind 1 or 2 hours, and these lags are often detected as false anomalies. Thus we focus on the other two methods, and find that the supervised **XGBoost** regression **works well for weekday anomaly detection**, and the unsupervised **Isolation Forest works well for weekend anomaly detection**. Our final anomaly output will be **grouped by day**, meaning that days with abnormal patterns will be all marked as anomalies.

## 2.1 Weekday anomaly detection: XGBoost

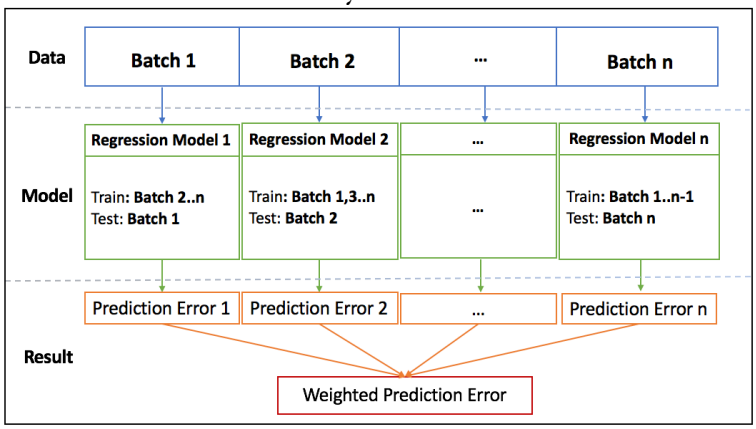
The features we use for training the regression model include **month, week, hour, day of the year, day of the month, day of the week, temperature, etc**. We drop time-series features like one-order lag as these features have big impacts on the next values, thus lead the model to focus more on local regions rather than the global pattern.

### 2.1.1 XGBoost for regression

XGBoost is an efficient library for training gradient boosting trees [3]. It produces a prediction model by combining an ensemble of weak predictors, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.

### 2.1.2 Detect anomalies by prediction error

We define an instance as an outlier if the prediction error exceeds a predefined threshold. To avoid over-fitting, we split the data into training and testing sets to ensure each instance occurs only once in the train set and test set, as Figure 5 shows.




i. Data are separated into n batches ;

ii. For every round, choose the i-th batch as test set and the remaining as train set, then train model;

iii. Calculate a weighted average for the n predicted results(1 in test, n-1 in train) as the final prediction

Figure 5: XGBoost train and predict mechanism illustration

The predictions are grouped by day and days with extreme large errors will be labeled as anomalies. We notice that anomalies we detect in this way are mostly weekday observations, as weekend instances usually have lower values and lower prediction errors. Therefore, for weekend instances we decide to try other methods and find out that Isolation Forest, an unsupervised method, to be very effective.

### 2.1.3 Visualization of detected anomalies

By applying the above features and models, we find out anomalies shown as Figure 6 shows (a few examples). We see that days with continuous (until late) high consumption values are more likely to be anomalies, which corresponds to our model logic and practical sense.

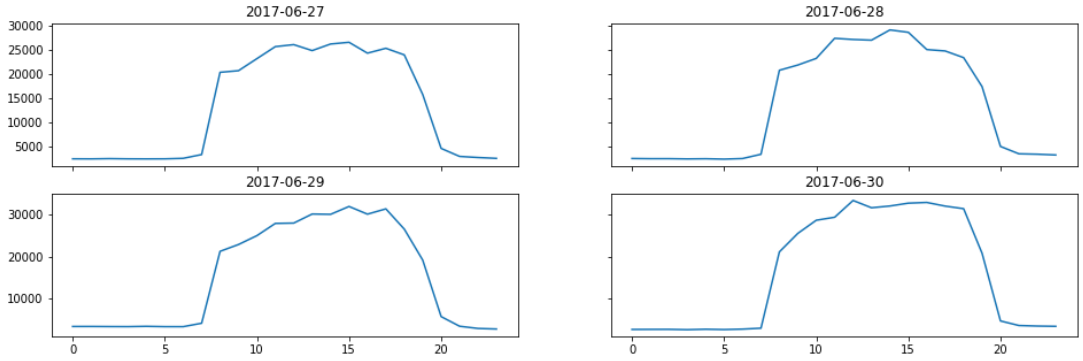



Figure 6: Abnormal daily patterns of meter 334_61

## 2.2 Weekend anomaly detection: Isolation Forest

As we would like to detect anomalies on a daily basis, and different from supervised model, we don't need large amount of training data, we decide to organize the features grouped by day. In this way, **each day is treated as an instance**, with daily features like temperature, 24-hour power values of the day, etc., and the output would be the anomaly scores of each day.

### 2.2.1 Extracted features

- **24-hour power values**

The power value of each hour (0 to 23) will be extracted as features as below:

| Date | Hour0 | Hour1 | Hour2 | Hour3 | … | Hour23 |
|---|---|---|---|---|---|---|
| **2015/6/13** | 2329.75 | 2275.50 | 2308.50 | 2235.25 | | 2497.25 |
| **2015/6/14** | 2463.00 | 2362.00 | 2323.75 | 2362.50 | | 2653.75 |
| **2015/6/20** | 2171.50 | 2190.50 | 2169.25 | 2169.25 | | 2421.75 |

- **Temperature**

As there exist several temperature meters for each site, we aggregate their values by averaging on a daily basis.

- **Power value area after 9:00am**

We notice that most daily power curves in weekends are steady, some may have higher values in the early morning due to the influence of Friday. Therefore, high consumption values after 9:00am may be a strong signal for anomalies. We calculate the area as the sum of difference between the power value with the minimum value.

- **KL (Kullback–Leibler) divergence**

The KL divergence is a measure of how one probability distribution diverges from the empirical distribution. We calculate the mean value of power usage of each hour as the empirical distribution, then calculate the KL divergence of each weekend 24-hour power usage with the empirical distribution.

- **The hour of daily power values' peak**

If the peak appears late in the day, it's more likely to be an abnormal signal.

For meter 3, as we have more kinds of meters, more features will be extracted, such as **power value area of the demand power**, **power value area of the reactive power**, **the hour of demand power values' peak, etc.**

### 2.2.2 Isolation forest

Isolation forest [1] is a very effective unsupervised algorithm to detect anomalies, with a low linear time complexity and a small memory requirement. The way the algorithm works is that, first build isolation trees by randomly selecting features and splitting the data instances based on the features' break points, then anomalies are detected as instances which have short average path lengths on the isolation trees.

### 2.2.3 Visualization of detected anomalies

As our target is to detect overconsumption anomalies, the power values also mean a lot. Therefore, we use the original power values instead of normalizing them to the same scale. By applying the above features and Isolation forest algorithm, we get the top normal patterns and top abnormal patterns of daily electricity consumption distributions: (Take meter 234_203 as example)

We see from figure 7 that, there do exist big differences between the 24-hour electricity consumption patterns. Normal days have low consumption values and relatively steady patterns, abnormal days have long-period high consumption values, and sudden shifts.

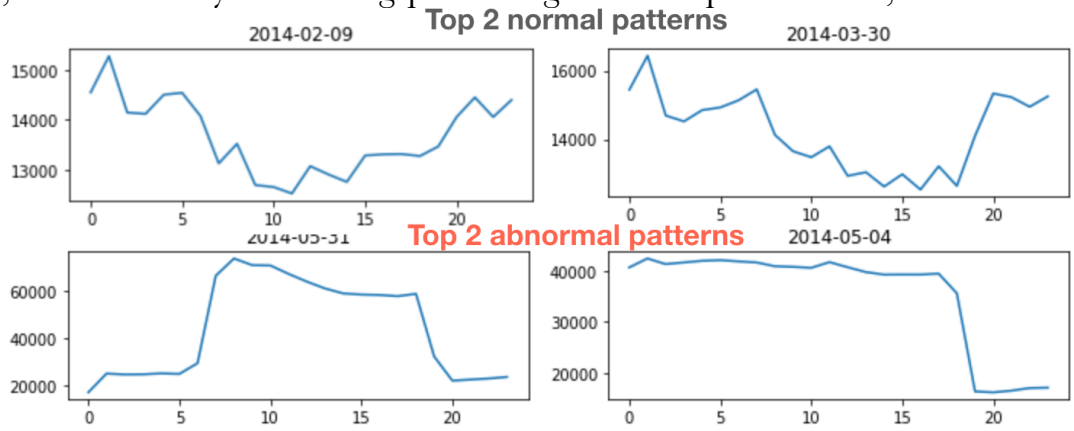



Figure 7: Top 2 normal/abnormal weekend patterns of meter 234_203

# 3. Other Potential Anomaly Patterns

Our main focus before is on detecting overconsumption anomaly patterns, therefore we don't normalize the original power values. But sometimes, the normalized daily consumption patterns also signify some kind of anomalies, although they don't have big differences in values. We then applied the Isolation forest on the **normalized 24-hour values**, and got the results as below (**weekday and weekend separately**):

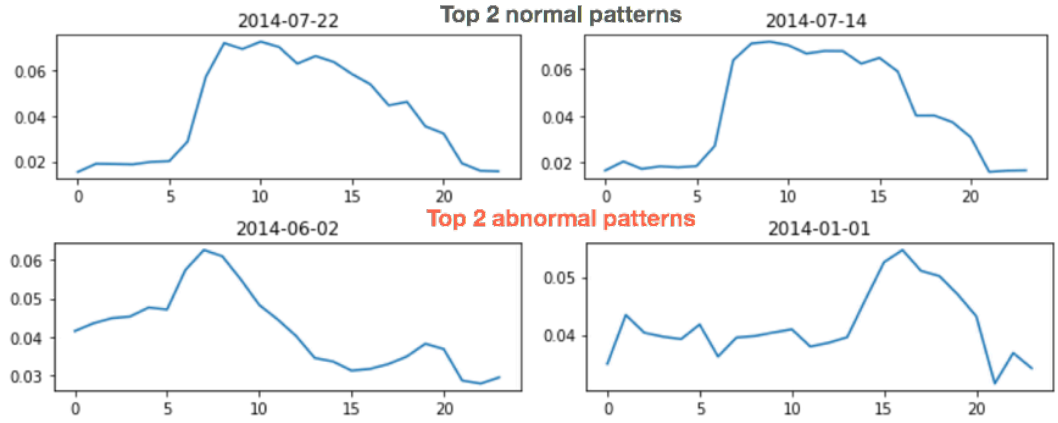



Figure 8: Top 2 normal/abnormal **weekday** patterns of meter 234_203

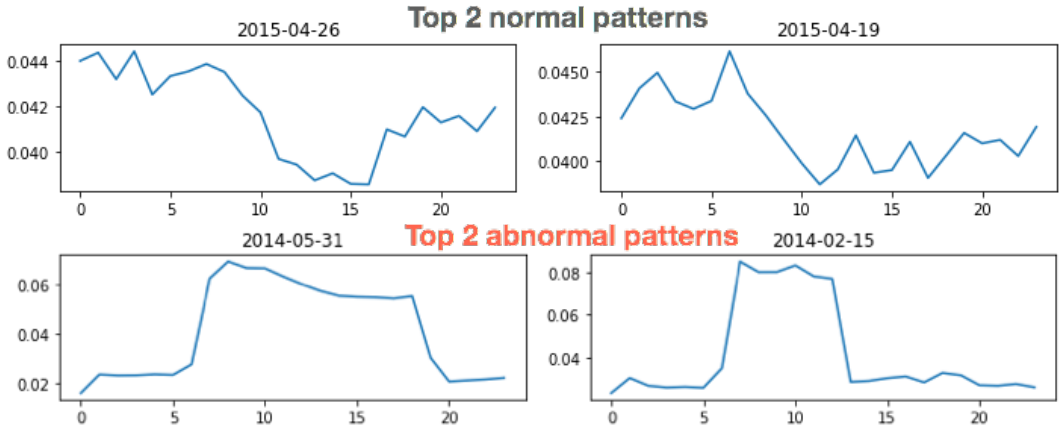



Figure 9: Top 2 normal/abnormal **weekend** patterns of meter 234_203

# 4. Energy-saving Suggestions

Based on the anomalies we detected, such as overconsumption caused by temperature variations and abnormal consumptions in nighttime and off days, our suggestions are:

- Deploy tools to detect and report abnormal consumptions in real time (such as methods stated above), and introduce the Automatic Power Off mechanism for electric appliances like air conditioners, lights, etc.) to avoid overconsumption during off days;
- Optimize the energy consumption supervision mechanism, promote to the public on energy-saving principles, and introduce incentive policies to encourage energy-saving behaviors.

# References

[1] Tianqi Chen. Introduction to Boosted Trees
[2] Zhi-Hua Zhou, Fei Tony Liu, Kai Ming Ting, Isolation Forest
[3] https://en.wikipedia.org/wiki/Gradient_boosting