Feature Engineering and Classification in Time Series Data:

A Study of Abnormal Electricity Detection

Linxiao Bai, Li Hsin Cheng
University of Rochester
Rochester, NY

Abstract – This report briefly explains a road map for detect abnormal electricity usage using a series of methodology of feature engineering and classification. Major approaches include unsupervised learning techniques and some customized feature generation process onto time series data. The performance of features is tested on labelled dataset with standard bi-classification models. The goal of the study is to construct a clear road map for time series feature engineering. The specific result drawn from the dataset will apply to abnormal electricity usage detection in industrial.

Keywords – electricity stealing detection; time series analysis; feature engineering; change point detection; probabilistic model; time series analysis; MAP estimator

I. INTRODUCTION

A. General Roadmap

Time series data are particularly interesting the field of feature engineering. Depending on the particular problem, transforming a series of observation to consistent features that help solving the problem can be an arbitrary and exhausting process. Classic approaches include using time series indexes such as moving average, CUSUM and other statistics tests to evaluate the data. On the other hand, unsupervised learning techniques with iid assumption, such as clustering, probabilistic model fitting based on MAP decision can also be adopted. Further more, we designed a transition point detecting algorithm that uses MAP decision and EM algorithm to detect the model transition point of a time series data. The performance of the features is tested on different common classifier such as SVM, Naïve Bayes, Random Forest, and so on.

B. About Electricity Stealing

Electricity stealing/ Theft of electricity is the criminal practice of stealing electrical power. Such crime has caused great damage all over the global, and brings about huge economic losses. According to the annual Emerging Markets Smart Grid: Outlook 2015 study by the Northeast Group, LLC, the world loses US$89.3 billion annually to electricity stealing. Common ways of stealing electricity include tapping lines into electricity grid or by-passing electricity meters. The two stealing behavior will result in two different pattern of power meter readings. Specific features are designed to detect such patterns. The goal is to replace the detection techniques from traditional eye-balling and inspection to data driven algorithm.

C. Motivation, Support, and Data Acquisition

The project comes from a competition held by China Computer Federation. All data is provided by China National Grid after UID masking.

See web page for more:
http://www.wid.org.cn/data/science/player/competition/detail/description/241

The training data contains more than 9000 users with their historical electricity reading. Each user is labeled with 1 as stealing detected and 0 as non-detected. The positive and

negative samples are slightly skewed with a ratio of 1:5. Table 1 gives a simple visualization of the data.

Identity	Date	Electricity Consumption @ t1	Electricity Consumption @ t2	Diff
1	day1	E1	E2	(E2-E1)
	day3	E3	E4	(E4-E3)
	day7	E5	E6	(E6-E5)
	day10	E7	E8	(E8-E7)
	...	...	...	
2	day1			
	day3			
	day6			
	day8			
	day9			
	day15			
	...	...	...	

Table1, a visualization of the data

Column 4 and 3 of the data monitor absolute meter reading of the current day and the day before, where Date column keep track of the time of observation. However, Date is not continuous in time. Sudden date jump may occur due to missing of record. Missing value handling method will be discussed.

II. METHODOLOGY and Analysis

A. Data Preprocessing

Missing values of Date are handled with care, and case consideration:

- For iid models missing value are ignored because their absence will not influence the accuracy. Forcing to fill the missing values will result in bias.
- For time series models, two approaches will be implemented, including regression fitting and filling with uniform value of average value of the closest points.

To handle Skewness of the data, resampling is introduced before feeding data into classifier to avoid high type 2 error.

Observation with short length of records are removed from the training dataset, because both time series assumption and iid assumption rely on the prerequisite of long record to extract useful information:

B. iid Based Features

Common statistics are introduced to describe the data:

- Mean
- Standard Deviation
- Test of Normality
- Pearson's Skewness
- Length of Record
- Proportion of "zero" records

Clustering method are also introduced to iid Features, including Gaussian Mixture Model with two components and K-means. Explanation of such pattern designing is associated with the certain case analysis and stealing pattern we introduced before (tapping grid and bypassing power meter).

In the case of a user that is classified as "1", the user has certain normal usage amount, but has too many 0 meter reading through out the year. It is also unlikely such user is away from home, because certain wiring will still consume power. With such behavior, the user is likely to be a stealer that by-passing his meter. Figure 1 and 2 are visualization of user power consumption behavior.

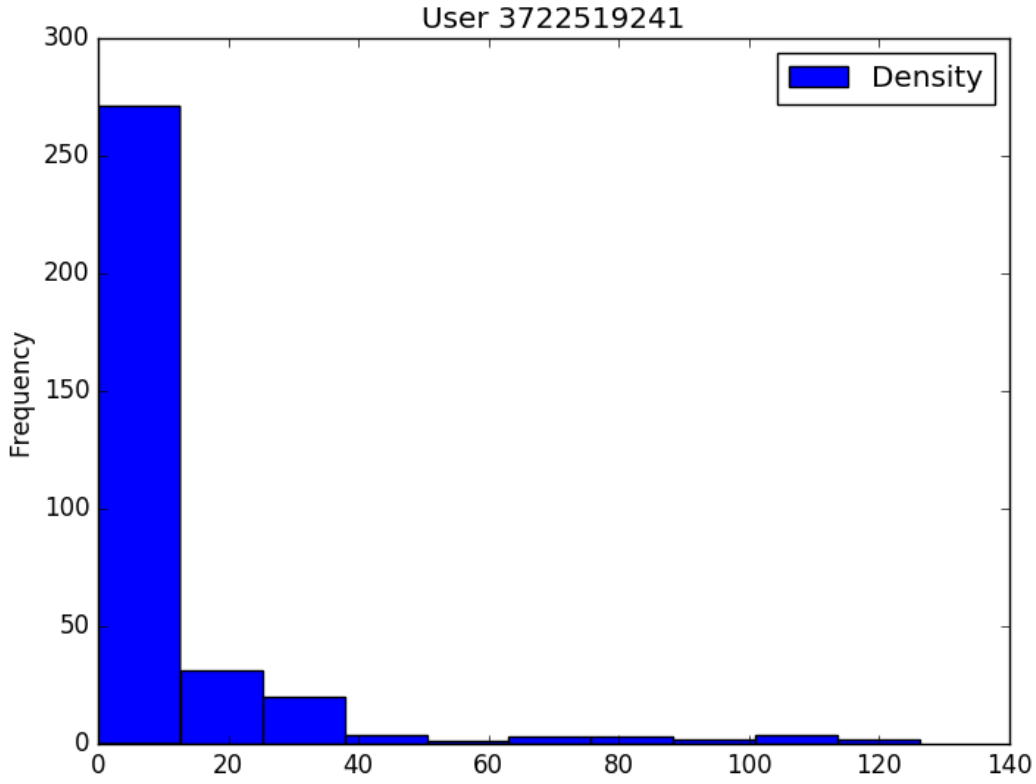


Figure 1, user 3722519241 has too many zero readings and high Skewness.

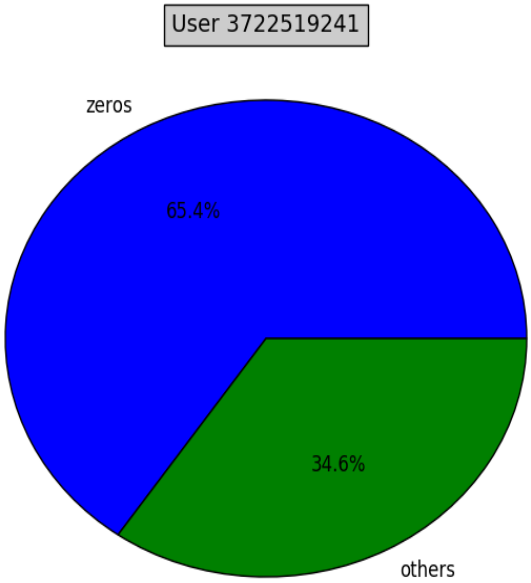


Figure 2, user 3722519241 has 65.4% of zeroes.

These kind of feature is successfully capture by Skewness and proportion of zeroes. Where proportion of zeroes is a feature that designed specially for this problem setting and background.

Another case analysis represents the other type of stealing behavior, that is tapping into the grid. Figure 3 shows the other type meter reading pattern.

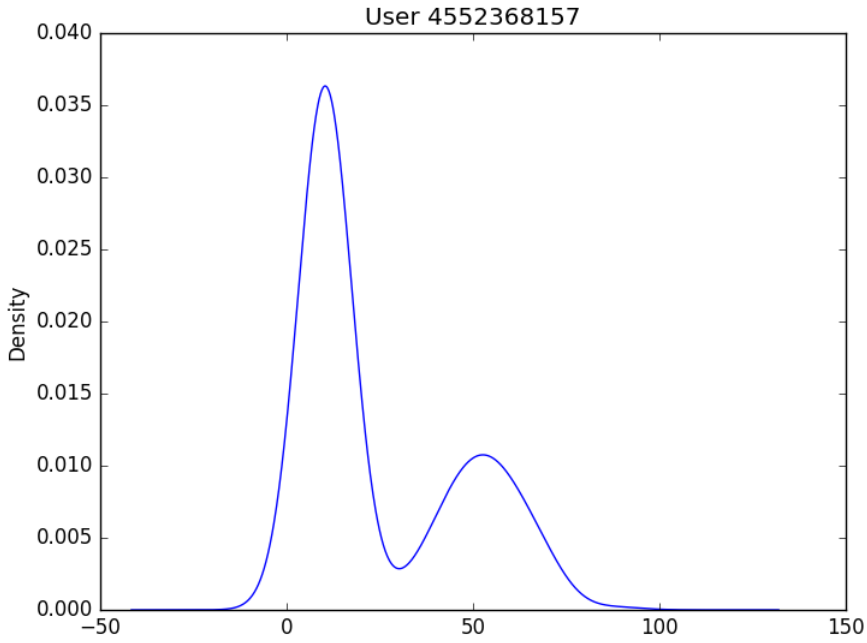


Figure 3, user 4552368157 is being detected as "1" because two reading patterns presents

In This case, two belt shaped curve presents in the Figure . In general, a reading of a single common household follows roughly one Gaussian distribution. Because of the wiring of the household is relatively stable, and using pattern is relatively unchanged, it is unlikely to observe two belt-shaped. However, because of stealer will constantly tapping into other girds to steal power, meter will record another type of consumption patter follows a different Gaussian distribution. Some times the Gaussian distribution with higher mean represents the additive pattern of two households. This pattern is successfully captured by the Gaussian Mixture Model. Figure 4 shows how Gaussian Mixture Model that capture the two belt shaped curve.

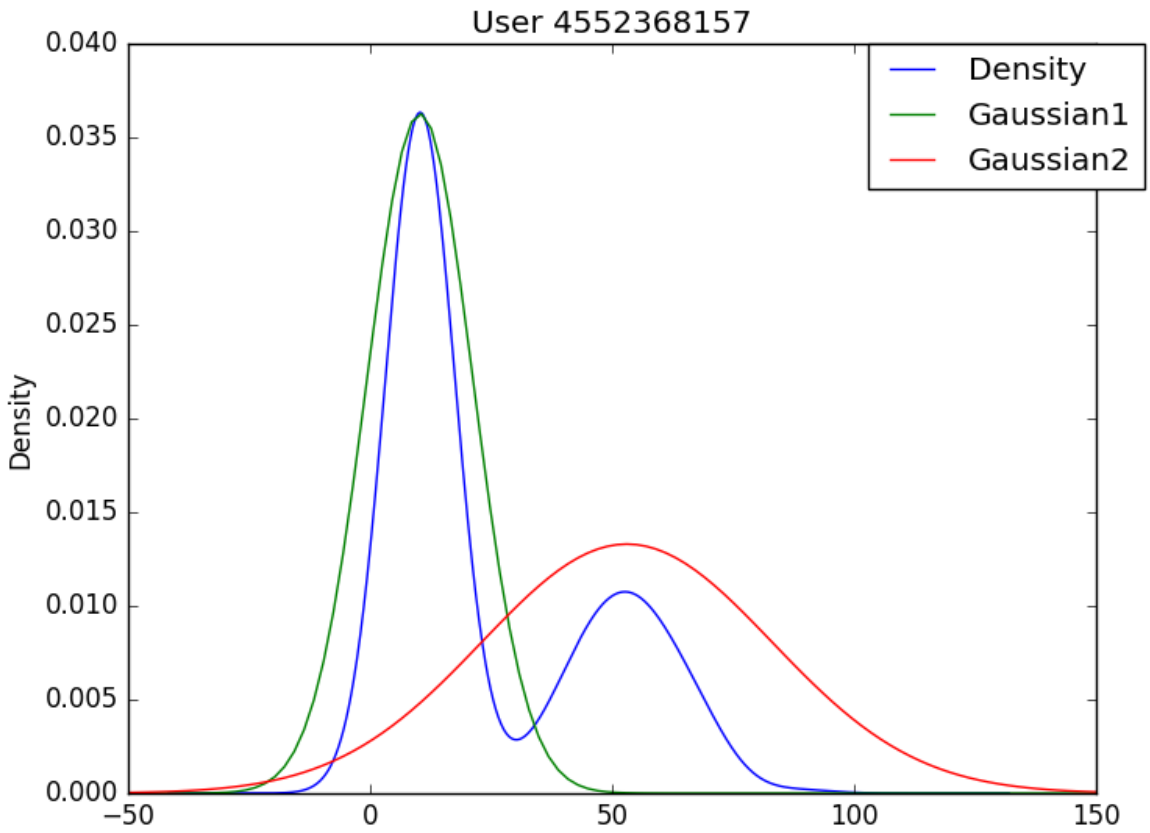


Figure 4, fitted Gaussian Mixture Model successfully captured the two patterns

It follows common sense the further two Gaussian models are separated, the stronger the alarm of stealing is. A mushy separation could be the result of forcing the data to split, and in fact is a signal from one using pattern.

To quantify mushiness of Gaussian separation, two measurements are implemented.

- Overlapping area
- *Z* score of two sample mean test statistics:

  Where $\boldsymbol{Z} = \frac{\mu 1 - \mu 2}{\sqrt{\sigma 1^2 + \sigma 2^2}}$

Figure 5 shows how mushiness interfere with our decision making about the separation of two Gaussian, the two mushiness index we designed will be included in the model and tested.

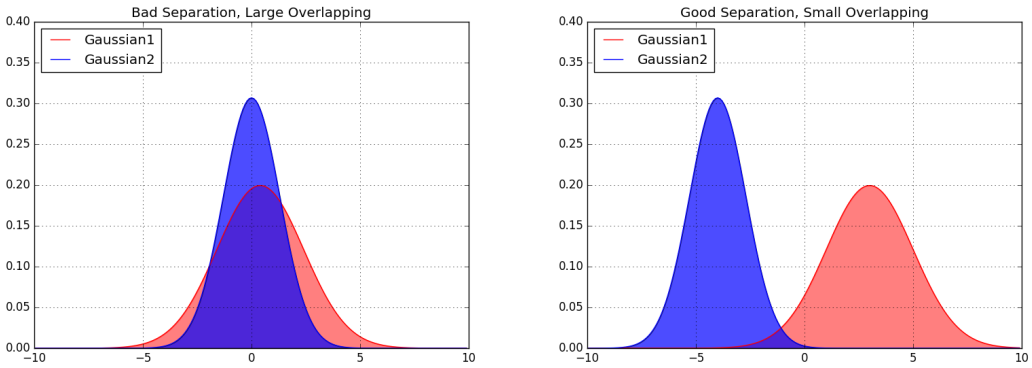


Figure 5, the alert of detection success increases as mushiness decreases.

C. Simi-time series Based Features:

This approach is related to recent study of time series model change point detection. This algorithm is specifically designed to overcome the challenge that model change at certain time. Thus result in two different looking patter before and after the change point. Similar to the grid-tapping situation, but this model handles the situation that stealing behavior starts at the middle point and continuous until the end.

Given a change point that stealing occur, and assume the using patter follows a Gaussian distribution, and the later signal follows another. The parameter of the two Gaussian models can be decided by MAP decision:

$\boldsymbol{\theta_1 = arg\ max_{\theta 1}\ P(\theta_1 \mid X1{:}Xt)}$

$\boldsymbol{\theta_2 = arg\ max_{\theta 2}\ P(\theta_2 \mid Xt{:}Xn)}$

Given the two Gaussian Model before and after the Splitting point, the probabilistic favorable splitting point can also be decided using ML estimation:

$$t' = \arg\max_{t'} P(X_1{:}Xt \mid \boldsymbol{\theta_1}) \times P(Xt{:}Xn \mid \boldsymbol{\theta_2})$$

As shown in Figure 6, this approach quickly falls into the realm of EM algorithm where the slitting point and $\boldsymbol{\theta_1, \theta_2}$ the parameter of Gaussians will surely converge to a local optimum.

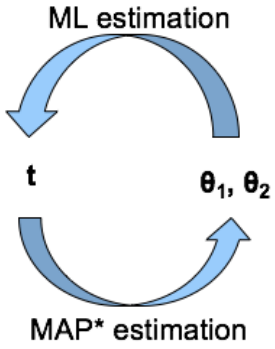


Figure 6, EM algorithm to decide model change point.

After obtaining the change point and two Gaussian distributions, mushiness defined as before are applied. Change point as well as mushiness of two model are selected as features.

This change point detection approach is first used and implemented by us. The implementation and mathematics are further discussed in the project of ECE 443, Probabilistic Model and Inference, where we will include as reference. Here we are just using the algorithm.

Because splitting point is a time series feature and probabilistic model we used is an iid model, it is hard to determine which assumption this feature belongs to. For now lets call this feature simi-time-series feature.

Finally, classification result of iid and simi-time-series features are tried out with different classifiers with different parameters. Shown in table 2, the best result is generated from Random Forest Model. Notice that without time series assumption, iid models are doing absolutely great job, with 88.9% over all accuracy and class 1 precision of 68% and recall 30%. However, we will further improve features by adding time series features.

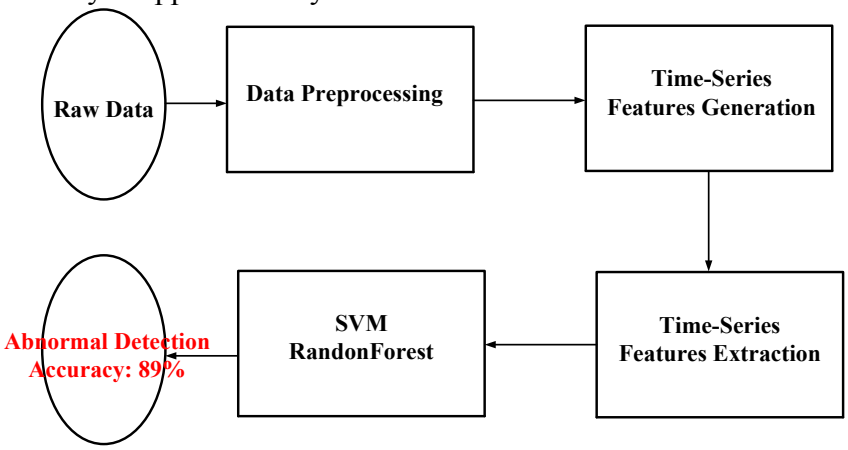
```
=== Summary ===

Correctly Classified Instances        2653               88.9075 %
Incorrectly Classified Instances       331               11.0925 %
Kappa statistic                          0.3631
Mean absolute error                      0.1807
Root mean squared error                  0.2951
Relative absolute error                 76.0946 %
Root relative squared error             87.2142 %
Total Number of Instances             2984

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.979    0.702    0.902      0.979    0.939      0.401    0.814     0.955     0
                 0.298    0.021    0.680      0.298    0.414      0.401    0.814     0.499     1
Weighted Avg.    0.889    0.613    0.873      0.889    0.870      0.401    0.814     0.895

=== Confusion Matrix ===

    a    b   <-- classified as
 2536   55 |    a = 0
  276  117 |    b = 1
```

Table 2, classification result generated from Random Forest Model

D. Time Series Based Features:

The electricity abnormal usage can also be considered as a time series sequence problem, except that there are missing value or some irrelevant value in the time series data. The time series abnormal analysis of more than 9000 identities electricity usage can be categorized as 4 major step: Data preprocessing, Time series feature generation, Time series feature extraction and use some classifier (like Random Forest or SVM) to classified the abnormal

electricity usage based on selected and reduced relevant time-related features. The corresponding accuracy is approximately 89%.



- **Data Preprocessing:**

The raw data is not shaped to be analyzed, we need to preprocess the data before we can do the consequential time-related feature generation. Firstly, we have the raw data containing identities that contain no or little time-related information. The most common form that contains no time-related information in our raw data is the identities containing too many 0s or missing values. There are methods to deal with unevenly-spaced time series data [2], but in our analysis, we discard identities that contains too many 0s and empty values (NULL) because it provides no information in time series analysis, and focus on the domain of evenly spaced time series data. With respect to identities having missing values, we use linear fitting to generate the missing data-points. Linear fitting assumes the identity uses the same amount of electricity between the first and last day of the missing interval. At last, we should remove data points less than 10 to ensure that there is sufficient amount of information to make confidence decisions.

Another crucial step in data preprocessing is normalization of the data. If we want to track the trend of the time-series data, we must normalize to avoid the influence of order. For example, if the two identities are normal electricity user with steady electricity usage every day, one is an institution that uses huge amount of electricity per day, whereas the other is a normal family; both of them have the same trend of electricity usage but due to its scale, one might classify incorrectly.

- **Time-Series Feature Generation:**

We use the package *tsfresh* to extract the time related features including entropy, Mean, kurtosis, Fourier transform coefficients, longest strike above mean, etc. Figure 7 illustrates the extracted feature in Matrix form. The row of the matrix represents approximately 9200 identities whereas the columns contains 200 time-related featrues.

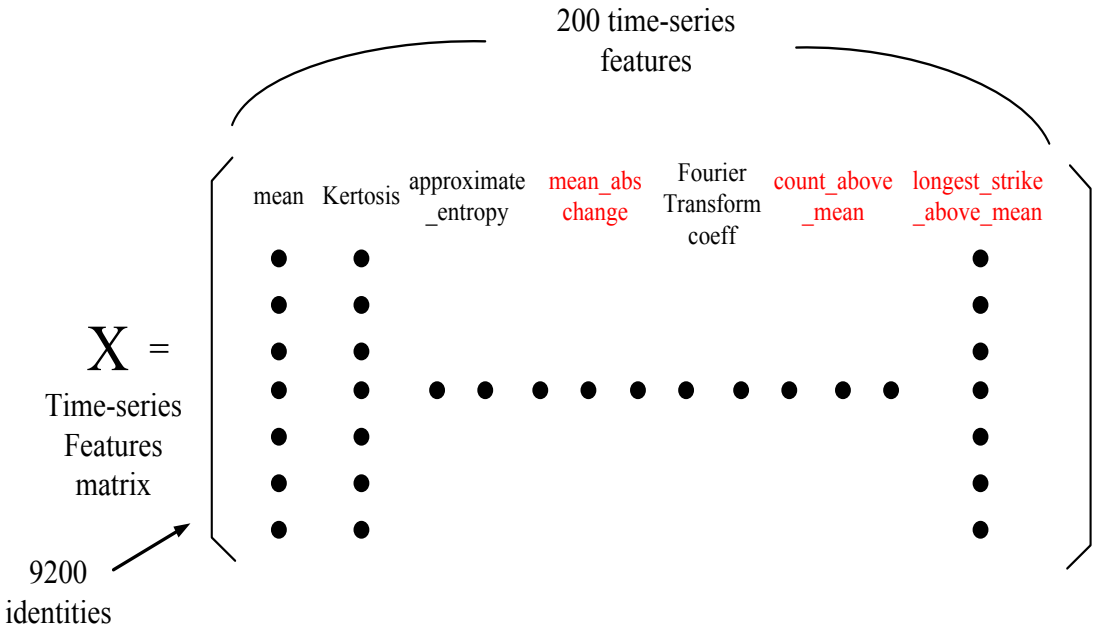


Figure 7

- **Time-Series Feature Extraction:**

We use the function in the package to do the hypothesis testing and base on p-value to accept or reject the feature in the matrix. The multiple test is based on Benjamini, Y. and Yekutieli, D. procedure [3]. The False Rejection Ratio$FDR = E[\frac{false\ rejection}{all\ rejection}]$, in our test it is set to almost 0 to eliminate unwanted features. The null hypothesis is the feature is not relevant and cannot be added, setting $\alpha = 0.05$, we use p-value significance test to see if we gave enough evidence to reject the null hypothesis.

After reducing to approximately 10 features, we can use RandomForest Algorithm to test the importance of each of these selected 10 features, the importance of each feature is as listed in figure 8. We can see from figure 8 that the most important two features: means absolute change from quantile 0.4 to 1 and means absolute change of quantiles from 0.2 to 1. We can also inspect from the other 8 feature and find the similarities between them. Can have an idea of what is the time series trend that happens to detect the abnormality usage of electricity?

We can see that most of the features listed are related to sum, changes or frequency of changes of the sequential electricity usage. Figure 9 shows the randomly sampled 10 identities with normalized time series electricity usage from positive (abnormal electricity usage) and negative (not abnormal) identities respectively. The randomly sampled positive identities somewhat shows the tendency of rapid growth after some certain time point. This is identical to the most important two features: mean absolute change from quantile 0.4~1 and quantile 0.2~1. These two features simply mean that if we discard the first 0.4 or 0.2 quantile, the rest of the data shows how rapid the change might be in the remaining last 0.6 or 0.8 quantile. This matches our intuition for electricity theft.

Features	Importance		Features_added	Importance
mean_abs_change_quantiles__qh_1.0__ql_0.4	0.19113586		sum_values	0.14025606
mean_abs_change_quantiles__qh_1.0__ql_0.2	0.18798775		mean_abs_change_quantiles__qh_1.0__ql_0.2	0.1379033
sum_values	0.17082261		average_unnormalized(added)	0.12056456
count_above_mean	0.09462988		sum_unnormalized(added)	0.11704146
longest_strike_above_mean	0.08898925		mean_abs_change_quantiles__qh_1.0__ql_0.4	0.09871492
mean_change	0.06356284		longest_strike_above_mean	0.0788439
last_location_of_minimum	0.06215619		mean_abs_change	0.07207339
mean_abs_change_quantiles__qh_1.0__ql_0.0	0.05641584		count_above_mean	0.06981881
mean_abs_change	0.04687143		mean_change	0.04855427
feature__length	0.03742837		last_location_of_minimum	0.04426189
			feature__length	0.03674651
			mean_abs_change_quantiles__qh_1.0__ql_0.0	0.03522092

Figure 8

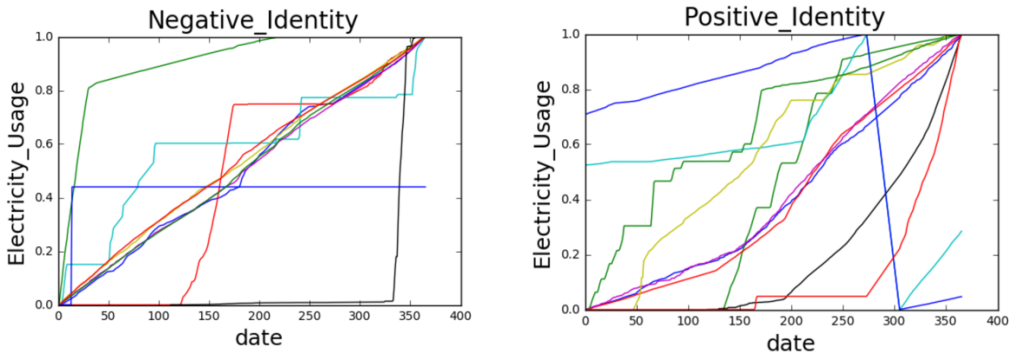


Figure 9

Figure 9 clearly shows some trend of absolute mean change in quantile 0.4~1 of the time series data, however, the diagonally rising lines in negative identities in figure 9 might have very close value with the positive identities. Figure 9 shows that although 9 out of 10 randomly sampled identities can be classified positive or negative by *mean_abs_change_quantile_0.2~1* and *mean_abs_change_quantile_0.4~1*, error still occurs.

Finally, we want to make sure that if the diagonally upward trend in negative identities and positive identities in figure 9 can be classified negative or positive with their real scale rather than normalized one. We thus add two features, namely *unnormalized_average_value* and *unnormalized_sum* to test if the scale is a feature affecting the classification of positive and negative as in figure 8. Unfortunately, it seems that these two feature does not add additional independent factor into the classification process.

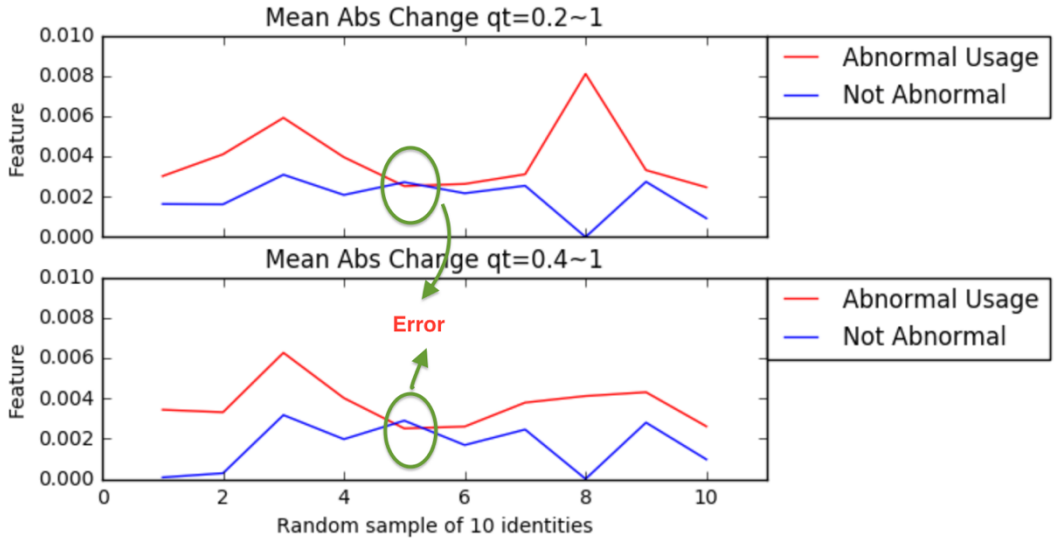


Figure 10

Finally, the 10 time-series features are used to classify if an identity is positive or negative. We use the RandomForest algorithm to test the accuracy with 30% training. The summary result and contingency table are show in figure 11.

```
Correctly Classified Instances        2433               88.1522 %
Incorrectly Classified Instances       327               11.8478 %
Kappa statistic                          0.4044
Mean absolute error                      0.1789
Root mean squared error                  0.306
Relative absolute error                 71.919  %
Root relative squared error             87.4678 %
Total Number of Instances             2760

   a    b   <-- classified as
2291   75 |    a = 0
 252  142 |    b = 1
```

Figure 11

E. Regression Result of all Features Together

After obtaining all the Features, all known common classifier are tired with different parameters tuning. Table 3-6 shows the performance of common classifier at an accuracy decreasing order.

➢

```
=== Summary ===

Correctly Classified Instances        1886               91.7315 %
Incorrectly Classified Instances       170                8.2685 %
Kappa statistic                          0.5401
Mean absolute error                      0.1487
Root mean squared error                  0.2635
Relative absolute error                 63.8982 %
Root relative squared error             80.2791 %
Total Number of Instances             2056

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.979    0.526    0.930      0.979    0.954      0.558    0.862     0.969     0
                 0.474    0.021    0.758      0.474    0.583      0.558    0.862     0.605     1
Weighted Avg.    0.917    0.464    0.909      0.917    0.909      0.558    0.862     0.925

=== Confusion Matrix ===

    a    b   <-- classified as
 1767   38 |    a = 0
  132  119 |    b = 1
```

Table 3, Random Forest Model, 91.7% accuracy.

➢

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.987    0.781    0.901      0.987    0.942      0.354    0.798     0.956     0
                 0.219    0.013    0.705      0.219    0.334      0.354    0.798     0.460     1
Weighted Avg.    0.893    0.687    0.877      0.893    0.868      0.354    0.798     0.895

=== Confusion Matrix ===

    a    b   <-- classified as
 1782   23 |    a = 0
  196   55 |    b = 1
```

Table 4, Logistic Regression, 89.3% accuracy.

➢

```
=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall   F-Measure  MCC      ROC Area  PRC Area  Class
                 0.965    0.725    0.905      0.965    0.934      0.321    0.778     0.949     0
                 0.275    0.035    0.523      0.275    0.360      0.321    0.778     0.412     1
Weighted Avg.    0.881    0.641    0.859      0.881    0.864      0.321    0.778     0.884

=== Confusion Matrix ===

    a    b   <-- classified as
 1742   63 |    a = 0
  182   69 |    b = 1
```

Table 5, Naïve Bayes, 88.1% accuracy.

It is clear that our classifier has preference towards certain classifier. That is a common observation. However, SVM and DNN, who usually have good performance perform badly on our features. While parameter tuning is a problem, it is also possible that rule based

feature and case analysis results in some case sensitive feature that fits in tree-like structure better. So that Random Forest gives the best result.

IV. Conclusion

In the case of abnormal electricity usage detection, multiple algorithm and indexes are designed. 14 features from iid, 5 features from simi-timer-series and 10 features from time series are selected to the classify of the problems. The best result generates 91.7% accuracy with class 1 precision 75.8% and recall 47.4%. This is a good result suggest that with 75.8% of confidence, the algorithm will capture 47.4% of the electricity stealers. Considering the low efficiency detecting method in real-world, relying on field work, the result has real-world application.

Comparing among the different classifiers, this set of features contains sensitivity to cases.

For features such as GMM and EM Change point detection algorithm we proposed, the usage of certain features depends on the particular situation, the nature of the dataset, and most importantly the goal of study. It is possible that these features are good for abnormal detection, and fraud detection.

Other features such as time series indexes, general statistics, and tests results categorize the general shape of the data, and significant mathematics behavior. This type of features are particularly suitable for general time series features.

VI. References:

1. https://github.com/blue-yonder/tsfresh
2. Andreas Eckner*, "A Framework for the Analysis of Unevenly Spaced Time Series Data", Stochastic Environmental Research and Risk Assessment.
3. Benjamini, Y. and Yekutieli, D. (2001). The control of the false discovery rate in multiple testing under dependency. Annals of statistics, 1165–1188.
4. Christ, M., Kempa-Liehr, A.W. and Feindt, M. (2016). *Distributed and parallel time series feature extraction for industrial big data applications.*
5. *An Efficient GMM Classification Post-Processing Method for Structural Gaussian Mixture Model Based Speaker Verification R. Saeidi; H. R. Sadegh Mohammadi; M. K. Amirhosseini 2006 IEEE International Conference on Acoustics Speech and Signal Processing Proceedings.*
6. *ADempster,N.Laird,andD.Rubin."Maximumlikelihood from incomplete data via the EM algorithm," Journal of the Royal Statistical Society, 39 (Series B):1-38, 1977.*
7. *Graph-Based Hyperspectral Image Classification Using Outliers Detection Based on Spatial Information and Estimating of the Number of GMM Mixtures Mahboubeh Lak; Ahmad Keshavarz; Hossein Pourghassem 2013 International Conference on Communication Systems and Network Technologies.*
8. *Comparison of GMM and fuzzy-GMM applied to phoneme classification Kacem Abida; Fakhri Karray; Jiping Sun 2009 3rd International Conference on Signals, Circuits and Systems (SCS).*
9. *Li Hsing Cheng, Linxiao Bai, Zhongda Su, Probabilistic Model and Change Point Detection in Time Series Data, ECE443 CourseStudy.*