Capstone Report 1 - EDAV

Introduction

Our capstone aims at detecting anomaly pattern in general datasets in our real world, especially for time series data. For general purpose, we first explore the open source datasets. Because of the complexity of the anomaly definition and the lack of labeled data, we first manually label several kinds of anomaly pattern based on the open source datasets available for now. In general, we explore three datasets and come up with 3 types of general anomaly pattern as follows:

1. Outlier
2. Sudden change of distribution
3. Pattern Inconsistency

Outlier describes a sudden outlier value happens at a random time point. Sudden change of distribution describes the time series data follow some distribution before a time point and suddenly the distribution change at that point, including change of distribution function, such as from normal distribution to uniform distribution, or change of distribution parameter, such as the mean of the normal distribution change from 0 to 1, or change of the trend of distribution parameters, such as mean of normal is increasing linearly and suddenly change to decreasing linear. Pattern inconsistency describes the data follow some periodic pattern such as a jump up every 100 time steps and suddenly the cycle period changes from 100 to 50, or there is suddenly another periodic pattern happens the same time.

We formulate anomaly data into arbitrary combination of above 3 patterns and then generate virtual data so that we can have ground truth in our generated data for us to run some baseline to have a common sense of the difficulty of the problem and then explore some advanced algorithms try to improve the model performance. We'll finally applied our model on real world data and analysis the result to see if it goes in line with human knowledge.

Literature

From a recent survey [1] on anomaly detection, there are these kinds of anomaly detection algorithms. (the number of categories lists here is different from the original survey, because we have merged some of the categories and some may require specific structured data that is out of the scope of our project.)

- Distance-based methods

  The algorithms that view an outlier as data points that are away from its neighbor or criterion.

- Density or clustering based methods

  The algorithms that assume the outlier is not within or nearby large or dense clusters.

- Ensemble based algorithms.

  The algorithms that use the ensemble method to help detect outlier.

- Statistical-based methods

  The algorithms that have some form of distribution assumption upon data.

We will select one or several examples for each type list above to give the reader a better understanding. Of course, there exist algorithms that can not be simply classified and we will also give one example.

Distance-based example

The simplest distance-based method for anomaly detection is a KNN based algorithm [2]. They just rank each point on the basis of its distance to its kth nearest neighbor and declare the top n points in this ranking to be outlier.

There are also lots of algorithms try to reduce the dimension of the data (and remove noise in the meantime) before compare outlier with normal data, for instance, we can use autoencoder to encode data into low dimension feature vector and reconstruct data without noise.

Density and clustering based example

The idea of clustering-based algorithm is to use a common clustering algorithm and the data point that is not within the large cluster are regarded as outliers. And LOF [3] is a typical density-based algorithm, it defines the local density upon KNN to get a LOF score.

Ensemble based example

Isolation forest [4] is one of the most well-known anomaly detection algorithms. It is an unsupervised tree-based anomaly detection. The basic unit of an isolation forest is an isolation tree. The formation of an isolation tree is almost the same as a binary decision tree. The only difference is the approach to find split point. Because the data for isolation tree is unlabeled, we

could not use the maximum information gain as the metric. Instead, we would randomly select an attribute and random select the value of split point (between the maximum and minimum).

And isolation forest to isolation tree is the same as random forest to decision tree. It is using the same ensemble method. The anomaly data points would very likely be away from the normal data and they could be easily separated (meaning their path length from root is small). Therefore, the algorithm based on the path size of each data point to decide its probability of being anomaly.

Statistical-based method example

Assume we know the data following gaussian distribution, we could use a sliding window to calculate the p-value of recent data points within the window. And if the p-value is smaller than the certain threshold, we would reject the assumed distribution and claim there is an anomaly. This method is very easy to implement and would also work on stream data. Yunbai implements a basic version here https://github.com/DH-Diego/CapstoneAnormalyDetection/tree/master/Models.

Mixed Method Example (Loda)

Figure 1: The training algorithm for Loda

Algorithm 1: Loda's training (update) routine.

input: data samples $\{x_i \in \mathbb{R}^d\}_{i=1}^n$;
output: histograms $\{h_1, \ldots, h_n\}$, projection vectors $\{w_i\}_{i=1}^k$. ;
initialize projection vectors with $\left[d^{-\frac{1}{2}}\right]$ non zero elements $\{w_i\}_{i=1}^k$;
initialize histograms $\{h_i\}_{i=1}^k$;
for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** k **do**
    $z_i = x_j^{\mathrm{T}} w_i$;
    update histogram h_i by z_i;
  end
end
return $\{h_i\}_{i=1}^k$ and $\{w_i\}_{i=1}^k$.

Algorithm 2: Loda's classification routine on sample x.

input: sample x, set of histograms $\{h_i\}_{i=1}^k$ and projection vectores $\{w_i\}_{i=1}^k$.;
output: anomaly value $f(x)$;
for $i \leftarrow 1$ **to** k **do**
  $z_i = x^{\mathrm{T}} w_i$;
  obtain $\hat{p}_i = \hat{p}_i(z_i)$ from h_i ;
end
return $f(x) = -\frac{1}{k} \sum_{i=1}^k \log \hat{p}_i(z_i)$;

Loda is an example of containing different detection methods above. Loda contains a collection of k one-dimensional histograms {hi}, and each of them approximates the probability density of the data after we mapped them onto the projection vector wi, which is similar to the process of Principal Component Analysis (PCA). The algorithm is that after initialize the set of histograms {hi}, we mapped the data sample onto the projection vector {wi}, where we assume that wi is independent of wj for i not equal to j. Then we calculate the log joint density of each transformed data sample and find the average log of them. From the Loda's algorithm, the smaller the output scores are, the higher likely the data would be anomalous.

The advantage of Loda method is that when dealing with a large number of samples in real-time or domains where the data stream if subject to concept drift and the detector needs to be updated on-line, Loda method has low time and space complexity property.

Neural Network Based Method Example

For the time series dataset, we can use traditional statistical approach to anomalous data detection - analyzing the residuals. Based on the residuals, we can do supervised or unsupervised learning approaches.

First, the residual = Time Series - Median - Seasonality. Here we use median instead of trend to avoid spurious anomalies. Then we can analyze the residual by density-based, clustering-based or ensembled-based methods above.

Also, instead of deriving the residuals, we can use Long Short-Term Memory(LSTM) method to detect anomalies. Compared to other neural network method, such as Deep Neural Networks and early RNN, LSTMs have been shown to improve the ability to maintain memory of long-term dependencies by introducing a weighted self-loop which depend on the context that allows them to forget part of past information while accumulating it. [6] As a result, because of the property to learn the long term correlations in a sequence, when using LSTM method, we can detect deviations from normal behaviour without a pre-specified time window adopted in traditional statistical model such as finding the p-value or performing the Kolmogorov-Smirnov test.

The algorithm is that we need to learn a prediction model by using stacked LSTM networks and computing the error distribution, which is the difference between xi at time t and t-j. Then we use the error vectors ***e*** to fit a multivariate Gaussian distribution and calculate the likelihood from this distribution given the value **e**. Finally, we compare the likelihood with ***alpha*** which is learned from the validation set by maximizing F-score. If the likelihood is less than alpha, we can say that the observation is classified as 'anomalous'. [7]

Data Visualization and Exploration

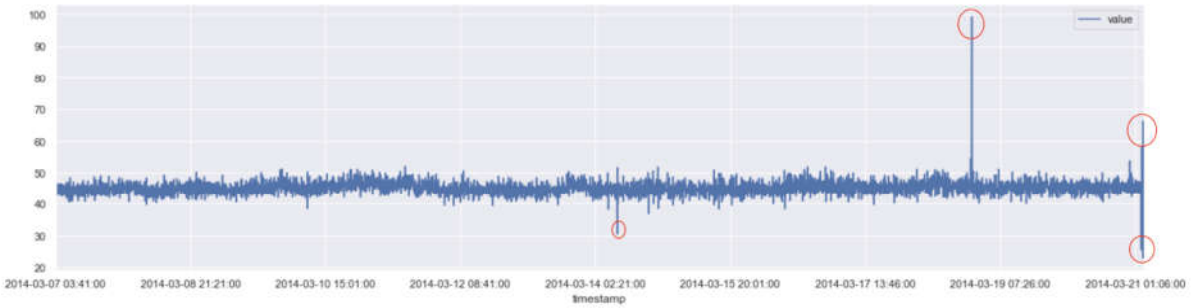
We visualize dataset within 3 areas: ec2 cpu utilization, advertisement, and anomaly data with known cause.
The open source data can be found at https://github.com/numenta/NAB/tree/master/data.
We then labeled all anomaly point of 3 types of anomaly pattern we mentioned before and label them with different color. We label all outliers with red circle, and we label the distribution changing point with yellow rectangle and we label the different pattern points with orange rectangle.

1.Anomaly Data with Known Cause

This dataset consists of the data which we know the anomaly reasons. We choose two representative datasets from it to analyze and classify the anomaly.

1.1 CPU usage data from a server in Amazon's East Coast datacenter Dataset

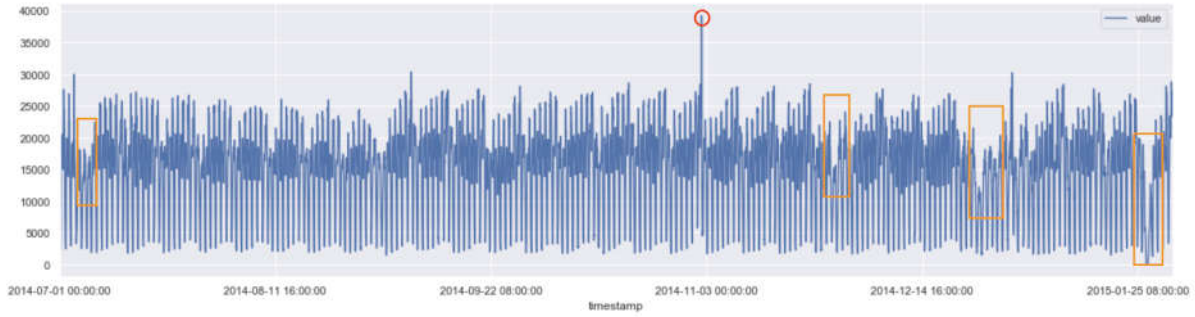
Figure 2: Time series plot for CPU usage data



The dataset ends with complete system failure resulting from a documented failure of AWS API servers. From the general picture, the dataset is obviously stationary with several outliers marked by red circles. The dataset could be regarded as a normal distribution with mean around 45 and relatively small variance with less noise. Sudden change of distribution and pattern inconsistency do not happen in this dataset. If we encounter dataset similar to this, we only need to detect the outliers.

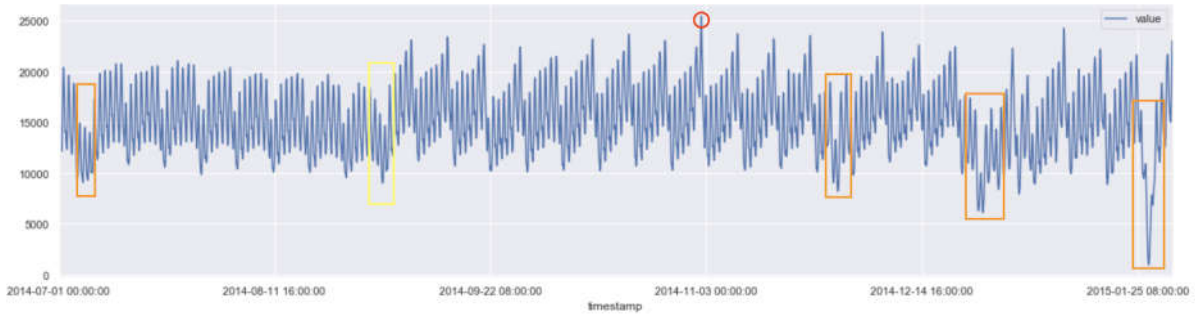
1.2 New York City Taxi Passengers Dataset

Figure 3: Time series plot for New York City Taxi Passengers Dataset



The data file included here consists of aggregating the total number of taxi passengers into 30 minute buckets. There are two kinds of anomalies existing in the dataset. There is one obvious outlier marked by red circle. The dataset has a cycle pattern and there are four periods of time do not follow the pattern, which is also : referred as pattern inconsistency.

Figure 4: Rolling window mean for New York City Taxi Passengers Dataset



If we use a rolling window on the new york city taxi passengers dataset, we will discover a sudden change of distribution marked by yellow rectangular.

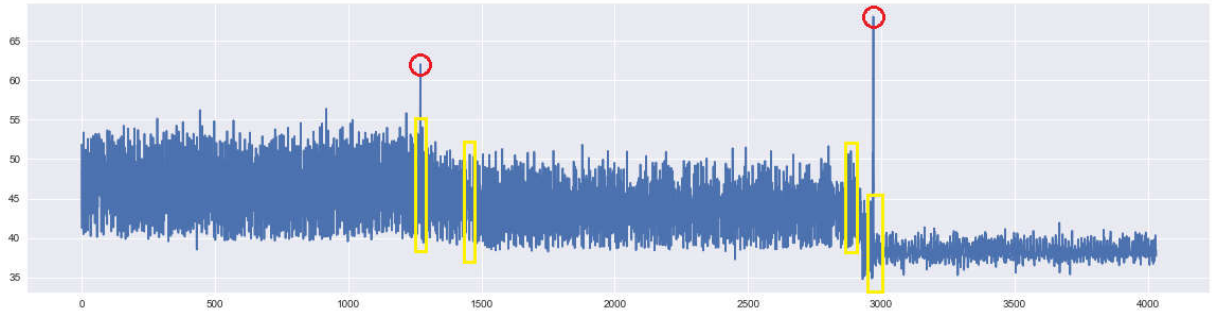
According to the known cause, there are five anomalies occurring during the NYC marathon, Thanksgiving, Christmas, New Years day, and a snow storm. The anomalies we discover correspond to most of the known anomalies.

2 . EC2 CPU Utilization

Below are 3 typical cpu utilization datasets of AWS EC2.

2.1 CPU usage data from a server in Amazon's East Coast datacenter Dataset

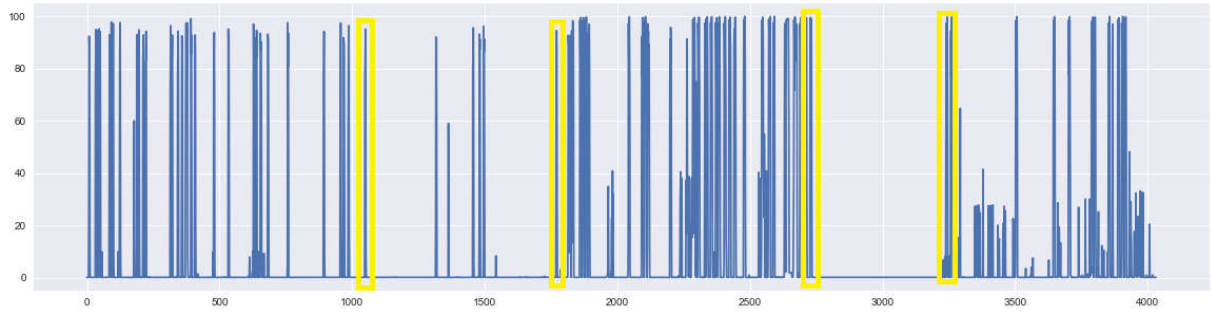
Figure 5: CPU - 5f5533



For the first dataset there are several changes of distribution and outliers. Including change of normal distribution parameters and two obvious outliers.

2.2 CPU usage data from a server in Amazon's East Coast datacenter Dataset

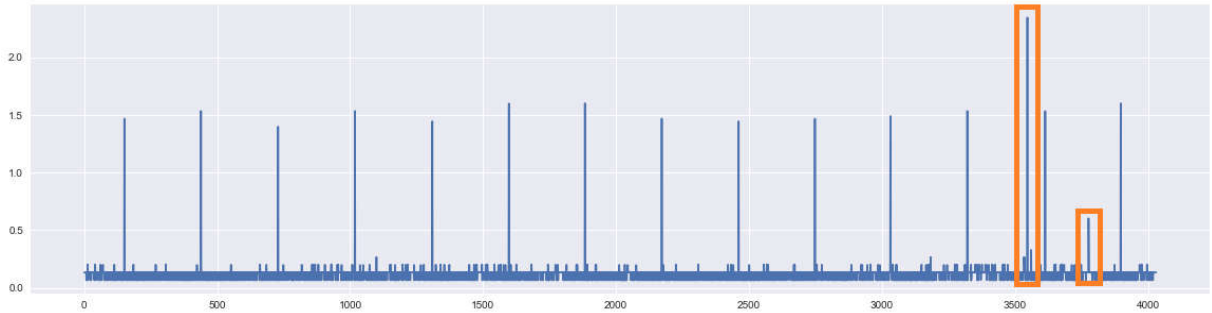
Figure 6: CPU - 77c1ca



For the second dataset, we take it as a Bernoulli distribution indicating if there is anything happening within a specific time window. We define all the value larger than 20 as something is happening (Y = 1 in Bernoulli context) and all others as nothing happen (Y = 0 in Bernoulli context). And we can see there is some changing point of the ‘average density’ which is an estimator of p in Bernoulli.

2.3 CPU usage data from a server in Amazon's East Coast datacenter Dataset

Figure 7: CPU - 825cc2



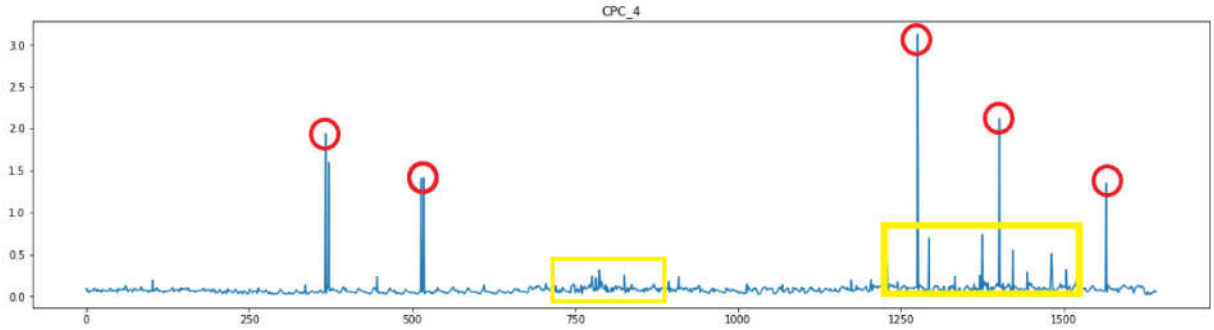
Although it seem the value later than 1.0 are outliers, they are not. All value larger than 1.0 has a fixed period of cycle until time step 3500 where suddenly come two peak not following the previous pattern. We label this as pattern inconsistency.

3. Advertisement

Advertisement dataset has three parts with the first part is normal, the second and the third part is anomaly. Each part consists of two kinds of data, one is cost per click (CPC) and another one is cost per thousand impressions (CPM). After visualizing both kinds, we found the graph of CPC is similar to the graph of CPM. So in the following analysis, we just show the results of CPC of anomaly parts.

3.1

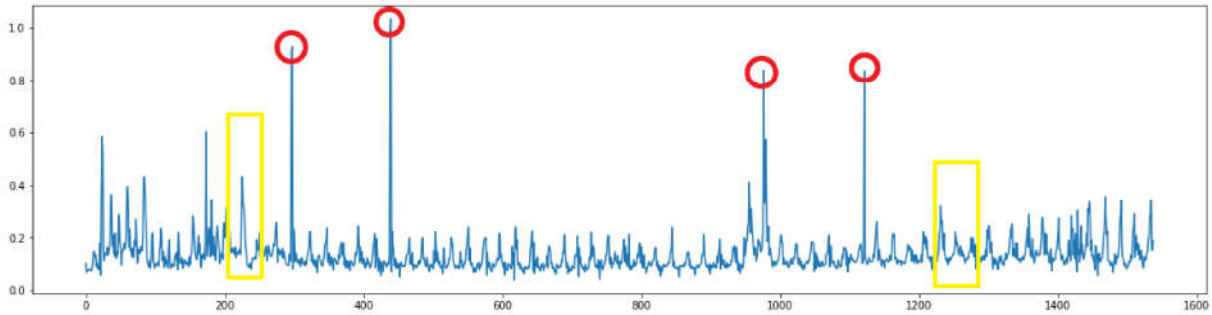
Figure 8: Cost Per Click of second part which has anomalous data



For the first part, we can see there are several outliers marked with red circles and changing of distribution marked with yellow rectangles. In the first rectangle, the vibration frequency increases. And in the second rectangle, the mean of the distribution is larger.

3.2

Figure 9: Cost Per Click of third part which has anomalous data



The second part is similar to the first part. First, there are several outliers. And then data distribution changes twice in the yellow rectangle parts. Mean of data decreases first and increase again. And variance in the first part is larger than the other parts.

Data Generation

Because most time series data for anomaly detection are not labeled, our team decided to write a simple library to automatically generate labeled data. The link toward this library is here: https://github.com/DH-Diego/CapstoneAnormalyDetection/tree/master/data_generated_library

Now the library supports mixture of different distribution, global trend, season and ARMA. Given the limited space for this report, please check the above link to view the README and code for this library.

Future Work

For next steps, we may apply several well developed methods we discussed in literature section on our generated dataset at the beginning. And for generating the dataset, we may come up with several anomalies ground truth of each anomaly type we discussed in the data visualization and exploration section, including one dataset for each type and one combined dataset mixed several types of anomaly together. And we will then apply these algorithms on these dataset aiming at evaluating and comparing the common and difference among all algorithms. And finally we will apply the tested method on real dataset to have a sense of the performance of the method on real world data. Also we explored one labeled real dataset, we will also try our method on it.

Contributions

In our project, our main body work can be divided into two parts: the literature part and the data exploration and visualization part. Zilin Zhu and Yunbai Zhang focus on the literature part. Han Ding, Zichen Pan and Feihong Liu focus on the EDA part. Introduction and future work is finished by Han and data generation part is finished by Zilin

Reference

[1] Wang H, Bah M J, Hammad M. Progress in Outlier Detection Techniques: A Survey[J]. IEEE Access, 2019, 7: 107964-108000.
[2] Ramaswamy S, Rastogi R, Shim K. Efficient algorithms for mining outliers from large data sets[C]//ACM Sigmod Record. ACM, 2000, 29(2): 427-438.
[3] Breunig M M, Kriegel H P, Ng R T, et al. LOF: identifying density-based local outliers[C]//ACM sigmod record. ACM, 2000, 29(2): 93-104.
[4] Liu F T, Ting K M, Zhou Z H. Isolation forest[C]//2008 Eighth IEEE International Conference on Data Mining. IEEE, 2008: 413-422.
[5] Pevný, Tomás; Loda: Lightweight on-line detector of anomalies. Machine Learning, July 2015
[6] K. Hundman, V.Constantinou, C.Laporte, I.Colwell, T.Soderstrom. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding; Machine Learning, Jun 2018
[7] P. Malhotra, L.Vig, G.Shroff, P.Agarwal; Long Short Term Memory Networks for Anomaly Detection in Time Series; Computational Intelligence and Machine Learning, April 2015