The Hong Kong University of Science and Technology

MSBD5002 Data Mining and Knowledge Discovery

Group 29 Report

Supervised Anomaly Detection

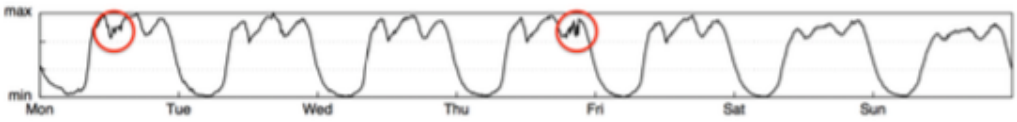
HE Shuo sheaw@connect.ust.hk 20797403

ZHU Zhenyi zzhubh@connect.ust.hk 20784183

FENG Geqin gfengac@connect.ust.hk 20787575

Background

To ensure that Internet-based services (such as search engines, online shopping, and social networks) are not disturbed, large Internet companies need to closely monitor various key performance indicators (KPIs, such as page views, online users, and orders) to be accurate find anomalies in time. The picture below is an example of 1 week PV, the red circles indicate some obvious anomalies.



PV picture

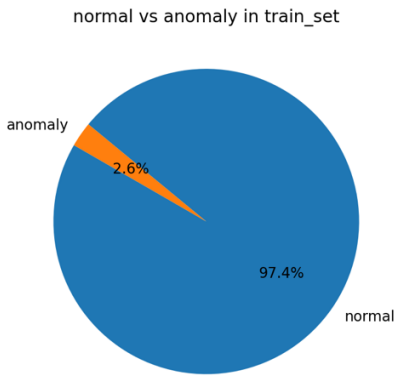
Data analysis in dataset

KPI ID number: 29

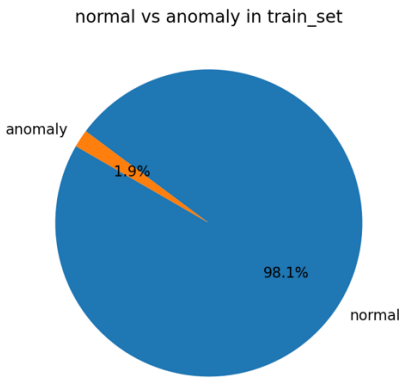
Label:

The normal data in train set is 2924512, the anomaly data in train set 79554

The normal data in train set is 2864287, the anomaly data in train set 54560



Anomalies in training set



Anomalies in test set

Feature engineering

Time series value. This means the exact value of each KPI at each moment. Obviously, it is a basic feature because it can directly reflect the current state of software or hardware.

Exponential smoothing method.

The exponential smoothing method is a time series analysis and forecasting method developed on the basis of the moving average method. By calculating the exponential smoothing value, with a certain time series forecasting model to predict the future of the phenomenon. Of course, we did not use exponential smoothing to predict KPIs, because we have KPIs at every moment. However, exponential smoothing can work well in forecasting tasks, which means that it can well grasp the trend of KPI changes, which is a useful feature. In general, we use plus exponential smoothing, simple exponential smoothing and Holt.[1]

Multiple Windows.

Multiple windows. The information contained in only one window at a time may not be enough. Since neural networks have strong feature learning capabilities, we try to derive other features from the above features. Specifically, we extract the above features on multiple windows of different sizes and calculate the difference between the features of different windows as another feature.

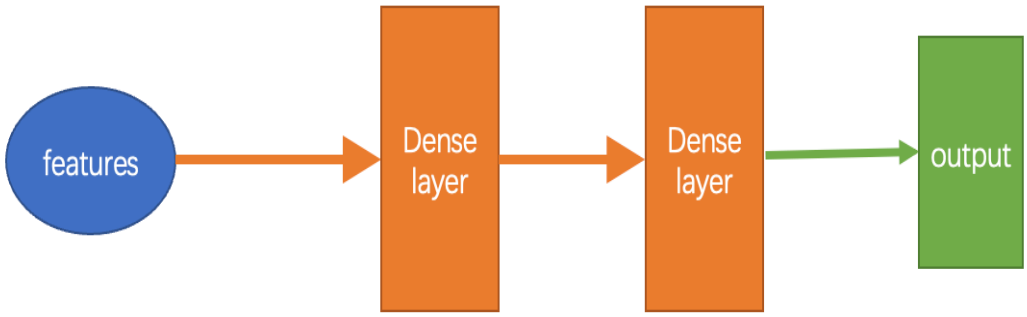
Preprocessing

Normalization

After feature extraction, we use sklearn.StandardScaler to normalize the features. It's worth noting that every different KPI time series has different StandardScaler, in other words, every different KPI time series has their own mean and variance. Every time series has different dimension, we can't normalize their features as a whole.Base on the previous procedure, we have 29 different KPI's normalizer(sklearn.StandardScaler). Then we use it to normalize test data's 29 KPI time series' features correspondingly.

Sample ratio method
As shown in Figure above, we can find that in the dataset, anomaly is much less than normal data. So we use the sample ratio parameter to help us tackle this problem.

Model chosen
We choose to use XGBoost model, random forest model first, but the result is not so good. The recall in test set is both around 85%, so we choose to build a DNN model. The structue of DNN is presented as follows:

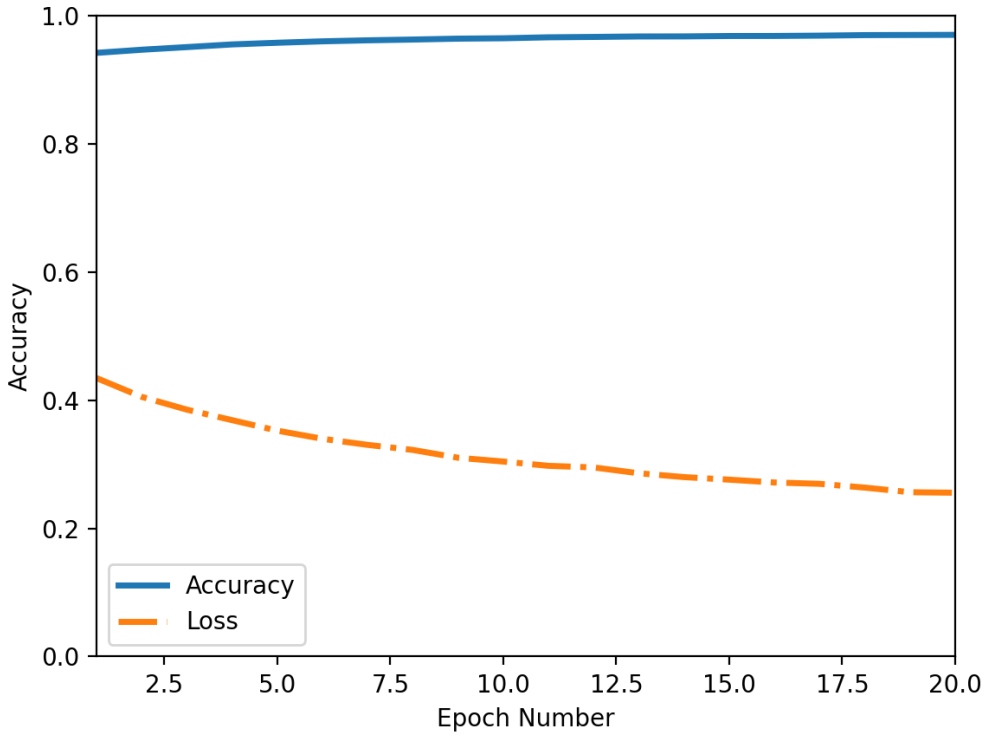


DNN structure

We have done a lot of experiments to find the best parameter in this model. For example, we implement two layers of densely connected neural networks with ReLU activation. The output layer is activated with the sigmoid function.

Note that all hyperparameters above are set by a series of empirical results, but without carefully finetuning. We believe that if we use AutoML techniques, we can improve the performance of DNN model.

The training process for first 20 epochs, later after we train 1000 epochs, we get the classification report in next part.



Training curve

Classification report

The anomaly detection algorithm need give a predict label (0 or 1) for every observation in testing set. We use classification report on both train set, and test set as the final performance metric in the ranking. Where the recall is more important than other parameters.

0(normal)

1(anomaly)

In training

`precision_score:` `0.9949978865307466`

`recall_score:` `0.9912560938871207`

`f1_score:` `0.9923412116062924`

	precision	recall	f1-score
0	0.99	0.99	0.99
1	0.88	0.88	0.88

In testing

`precision_score:` `0.9591524232429808`

`recall_score:` `0.9582312846720674`

`f1_score:` `0.9567294792462332`

	precision	recall	f1-score
0	0.96	0.96	0.96
1	0.84	0.84	0.84

References

[1] GitHub. https://github.com/x-bin,2021.