Anomaly detection for web traffic data

Xinli Gu
Center of Data Science
New York University
xg588@nyu.edu

He Di
Center of Data Science
New York University
dh3171@nyu.edu

Bo Zhang
Center of Data Science
New York University
bz854@nyu.edu

Abstract

Web traffic is the amount of data sent and received by visitors to a website. The web traffic data can be represented by a time series data to record activities on the website. Abnormal data points in this web traffic refers to abnormal changes of such traffic. Such abnormal change can be caused by network attacks. Thus, it is crucial to detect anomalies accurately and efficiently in the time series web traffic to further identity network attacks and prevent consequential economic and social losses. In this project, we experimented with both ARIMA and C-LSTM to perform anomaly detection on web traffic data, where ARIMA is a typical statistical approach and C-LSTM is an innovated deep learning structure applied on time series data. Experiments demonstrate that ARIMA performs differently on different types of anomalies as ARIMA focuses only on local data rather than a full picture. On the other hand, C-LSTM outperforms ARIMA and CNN alone, reaching a recall rate of 79.1%.

1 Introduction

As internet technology develops, computer networks are becoming more and more important. A large amount of information is exchanged through web servers and a variety of services are provided. However, with the increase in internet services, malicious attacks through networks are gradually becoming more advanced and diversified. Various network attacks can cause serious damage to web service operation, leading to both social and economic losses [Ahmed et al., 2016]. There are mostly 4 different methods of network attacks: Denial of Services, Probe Attack, User to Root Attack and Remote to User Attack [Kim and Cho, 2018]. In Table 1, we summarized their detailed description. Thus, it is crucial to take actions to prevent all different methods of network attacks that threaten network infrastructure.

The anomaly detection task in web traffic is a time-series classification problem because abnormal data points are usually having an irregular pattern than other data points around it. Such anomalies can be categorized into three different types. Figure 1 presents all three different categories of anomalies. Figure 1a contains point anomalies. They occur for data points that are considered abnormal when viewed against the whole dataset. Figure 1b contains contextual anomalies, which are cases where a data instance is abnormal in a specific context, but not otherwise. That is, the value of this data point is within normal range. However, it becomes anomaly when looked at together with its other context, such as the time it happens or the location it happens. Figure 1c contains collective anomalies, which is a collection of related data instances that is exceptional as a whole, even though individual values may be normal.

Different categories of anomalies can be mapped to different types of network attacks. Point anomalies are usually caused by User to Root Attack and Remote to User Attack. Contextual anomalies are usually caused by Probe Attack. Lastly, collective anomalies are usually caused by Denial of Services Attack.

[1]Link to the project git repo: https://github.com/zhangbo1997/DSGA1018_final_project.git

Table 1: Types of Network Attacks

Denial of Service	A cyber-attack in which the perpetrator seeks to make a machine or network resource unavailable to its intended users by temporarily or indefinitely disrupting services of a host connected to a network.
Probe Attack	An attack where the attacker attempts to gather information about the target machine or the network, to map out the network.
User to Root Attack	An attach taht tries to access normal user account and gains root access information of the system.
Remote to User Attack	An attack in which a user sends packets to a machine over the internet.

In the project, we proposed two different methods in detecting web traffic anomalies introduced above. The first one is the statistical approach: ARIMA. ARIMA is going to learn different distribution patterns of the time series from different types of anomalies. It classifies a data point as anomaly when the prediction error is beyond three stand deviation. The second method proposed is a neural network: C-LSTM. The C-LSTM network consist of Convolutional Neural Network (CNN) layers and Long Short-Term Memory Network (LSTM) layers. The CNN layer is used to extract spatial features while LSTM layer is used to extract temporal features. Detailed design of both methods will be discussed in Section 3.

2 Related work

The study of anomaly detection on time series can be traced back to 1972, where Fox experimented with auto-regressive forecasting model and developed several statistical tests on the scale of prediction error when detecting anomalies [Fox, 1972]. Since then, the time series anomaly detection problem has been treated with statistical methods and recently machine learning algorithms. [Burman and Otto, 1988]. used ARIMA to perform anomaly detection by classifying outliers in the time series. However, statistical models cannot properly classify abnormal traffic data with the same distribution as normal traffic data. More recently, with the growth of popularity of deep learning, LSTM based neural network attracts a lot of attention in the field of anomaly detection [Cheng et al., 2016]. Cheng et al. extracted temporal information by using a LSTM model in their process of anomaly detection. However, LSTM alone can only capture temporal features of sequence of data. If a pattern of web traffic data does not have a certain periodicity, the LSTM will not properly classify the web traffic data. Thus, CNN was introduced to extract spatial features together with LSTM to make the C-LSTM neural architecture [Kim and Cho, 2018]. In this project, we will experiment and compare how the statistical method ARIMA and neural network C-LSTM are going to perform differently on the anomaly detection task in time series.

3 Problem definition and algorithms

3.1 Task

The aim of this project is to perform anomaly detection on the time series data extracted from the Yahoo Webscope program. Detection of traffic anomalies to web servers is a binary time series classification problem. Two methods are proposed on this task. The first one is ARIMA, the statistical approach. The second one is C-LSTM, a combination of CNN layers and LSTM layers, aiming to capture both spatial information and temporal information from the time series data.

3.2 Algorithm

3.2.1 ARIMA

The ARIMA model is for a single time series and it is an offline method. Our proposed pipeline is shown at Figure 2.

- The first step is to remove the daily and weekly seasonality. Because our data is hourly and we know there are daily and weekly changes in web traffics. Removing seasonality will help to transform contextual anomaly into point anomaly and help to do the detection. Here, we assume seasonality is additive.
- The second step is to check the stationary and fit the ARIMA model on a de-seasonality time series. We use Augmented Dickey-Fuller(ADF) test to check the stationary and choose the ARIMA parameters range based on ACF and PACF plots. And finally use AIC to find the best ARIMA parameters.
- Third, get the prediction errors for ARIMA one-step ahead forecast.
- Last, use Box-Cox to transform prediction errors into a normal shape. And detect the anomaly timestamps where the transformed prediction error is beyond 3 standard deviation. Because the prediction errors could be skewed, we choose to make it normalized to prevent classifying fat tails into anomalies falsely.

3.2.2 C-LSTM

The C-LSTM model is composed of CNN layers and LSTM layers and they are linearly combined. As CNN has proven to have excellent performance in capturing high dimensional spatial features, the preprocessed time series signals are first passed into convolution and pooling layers to extract and create intermediate features. Then the LSTM layers help extract the temporal features followed by several fully-connected layers for classification. For easier model training and inference, we utilized the sliding window for input processing. The entire model structure for anomaly detection is presented in Figure 3.

CNN layer consists of convolution layers, which extract higher-level sequences of web traffic spatial features, and pooling layers, which lower the dimensionality. Let the input web traffic window be $x = (x_1, x_2, ..., x_n)$, where n is the window length and x_i is the normalized traffic values. The convolved result is given by

$$y_{ij}^{l} = \sigma(b_j^{l-1} + \sum_{m=1}^{M} W_{m,j}^{l-1} x_{i+m-1,j}^{l} - 1), \tag{1}$$

where i is the index of the feature value, j is the index of feature map for each traffic window, b represents the bias of the jth feature map, W is the weight of the kernel, M is the size of the filter and σ is the activation function, Tanh or ReLU, and l means lth convolution layer.

The convolution layer is then followed by a max pooling layer which decreases the dimensionality or the spatial size of the representation to reduce the number of parameters and computational costs. Max pooling is a type of pooling layer that selects the largest number in the feature map. Equation 2 shows the pooling operation,

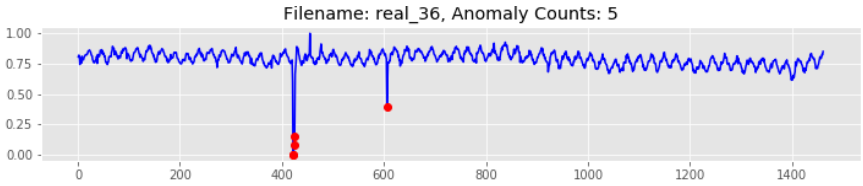
$$p_{ij}^{l} = \max_{r \in R} y_{i \times T + r, j}^{l-1}, \tag{2}$$

where R is a pooling size, y is the input and T is the stride of the pooling operation.

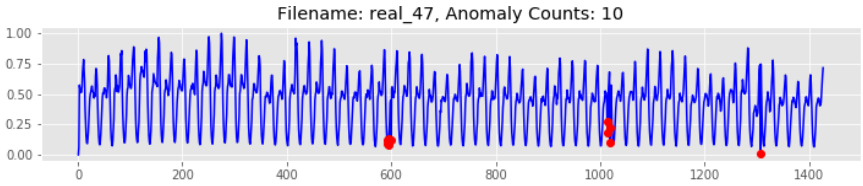
LSTM is one of the recurrent neural networks, which utilizes memory cells and gates to prevent vanishing gradient problem in vanilla RNNs. The output of the previous pooling layer is then passed to LSTM layers. During one LSTM layer, the hidden state h_t will pass through the forgetting gate (Equation 3), input gate (Equation 4) and output gate (Equation 5).

$$\boldsymbol{f}_t = \sigma\left(W_{pf}\boldsymbol{p}_t + W_{hf}\boldsymbol{h}_{t-1} + W_{cf} \circ \boldsymbol{c}_{t-1} + \boldsymbol{b}_f\right), \tag{3}$$

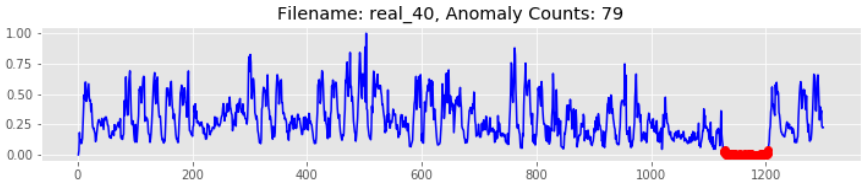
$$\boldsymbol{i}_t = \sigma\left(W_{pi}\boldsymbol{p}_t + W_{hi}\boldsymbol{h}_{t-1} + W_{ci} \circ \boldsymbol{c}_{t-1} + \boldsymbol{b}_i\right), \tag{4}$$



(a) Point Anomaly



(b) Contextual Anomaly



(c) Collective Anomaly

Figure 1: Three Categories of Anomalies

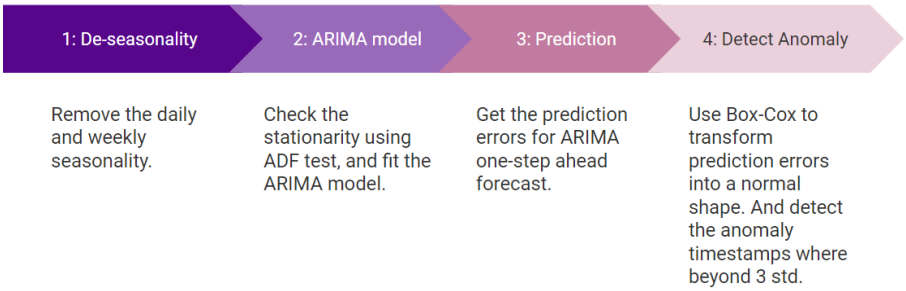


Figure 2: ARIMA Method Pipeline

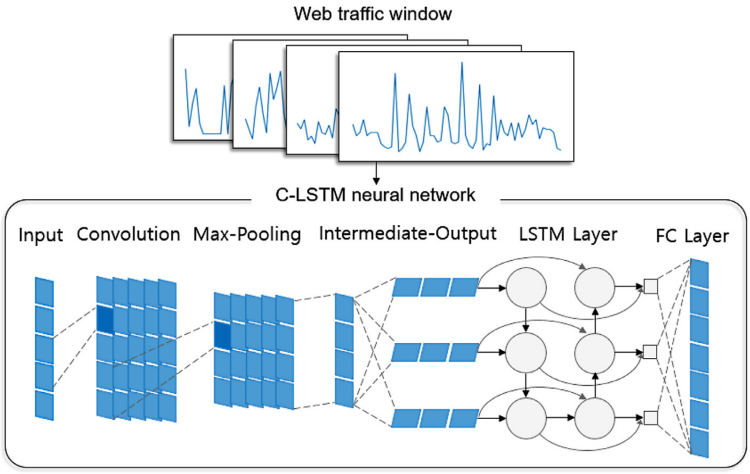


Figure 3: C-LSTM model structure

$$\boldsymbol{o}_t = \sigma\left(W_{po}\boldsymbol{p}_t + W_{ho}\boldsymbol{h}_{t-1} + W_{co} \circ \boldsymbol{c}_t + \boldsymbol{b}_0\right), \tag{5}$$

where $\boldsymbol{p}_t$ is the output of pooling layer or the input of LSTM layers, $\boldsymbol{c}_t$ is the cell state, $\boldsymbol{h}_t$ is the hidden state and $\boldsymbol{b}$ is the bias vector. In order to get the $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$, we need to perform Equation 6 and Equation 7,

$$\boldsymbol{c}_t = \boldsymbol{f}_t \circ \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \circ \sigma\left(W_{pc}\boldsymbol{p}_t + W_{hc}\boldsymbol{h}_{t-1} + \boldsymbol{b}_c\right), \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{o}_t \circ \sigma\left(\boldsymbol{c}_t\right), \tag{7}$$

Finally we make the prediction using a fully connected layer given by Equation 8, followed by a softmax classifier given by Equation 9 to detect anomalies in the web traffic data.

$$d_i^l = \sum_i \sigma\left(W_{ji}^{l-1}\left(h_i^{l-1}\right) + b_i^{l-1}\right), \tag{8}$$

where d_i is the output of the fully connected layer and h_i is the input unit.

$$P(c \mid d) = \text{argmax}_{c \in c} \frac{\exp\left(d^{L-1}w^L\right)}{\sum_{k=1}^{N_c} \exp\left(d^{L-1}w_k\right)}, \tag{9}$$

where C is the activity class, L is the last layer index and N_c is the total number of activity classes.

The C-LSTM architecture and layer specification is presented in Table 2.

Type	Filter	Kernel size	Stride	Param
Conv1d	64	5	1	384
Activation	–	–	–	0
MaxPooling	–	2	2	0
Conv1d	64	5	1	20,544
Activation	–	–	–	0
MaxPooling	–	2	2	0
LSTM (64)	–	–	–	262,400
Dense (32)	–	–	–	2080
Activation	–	–	–	0
Dense (2)	–	–	–	66
Softmax	–	–	–	0
Total number of parameters				285,474

Table 2: C-LSTM model architecture

4 Experimental evaluation

4.1 Data

The dataset that we use is extracted from the Yahoo Webscope program. The dataset consists of four benchmarks: A1Benchmark, A2Benchmark, A3Benchmark and A4Benchmark. We choose to use A1Benchmark because it is based on real production of traffic data to some of the Yahoo web servers. The class A1 contains 67 files and each file has a different distribution of traffic. And there is exactly 94,866 datapoints in A1 file and 1669 of them are anomalies, which occupy 1.76%. Note that the timestamps of the A1Benchmark are replaced by integers with the increment of 1, where each datapoint represents one-hour worth of data. Even though an exact timestamp is not available, it is still possible to identify the daily and weekly seasonality given that each datapoint represents one-hour worth of data. Figure 4 shows the number of outliers and anomalies calculated in each file. The outliers are defined to be out of the range of three standard deviations of the mean. From the figure, we can see that outliers don't follow the same pattern as anomalies. As both anomalies and outliers distributions are non-Gaussian, the Spearman Correlation was chosen to test if they share a monotonic relationship. However, the Spearman Correlation score is 0.39, which is considered to be not notable. Thus, it is hard to detect anomalies by simply classifying statistical outliers. Therefore, it is very difficult to perform anomaly detection using statistical approach on files with different pattern distributions.

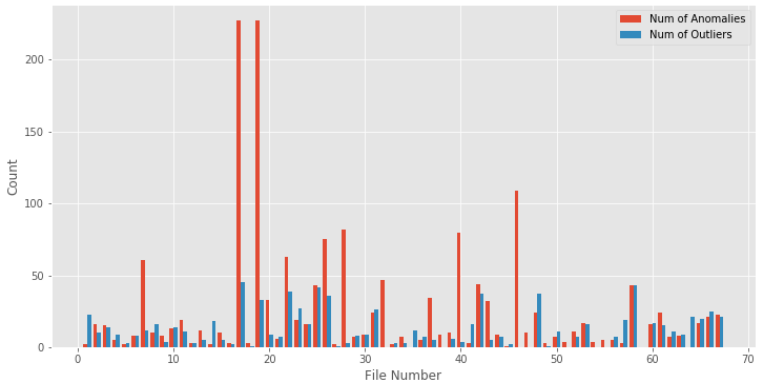


Figure 4: Distribution of Anomalies and Outliers

4.2 Methodology

We use classification metrics for evaluation metrics for both the ARIMA and C-LSTM model. Precision in Equation 10 is the percentage of relevant instances among all retrieved instances and recall in Equation 11 is the percentage of relevant instances retrieved among all relevant instances. Recall is particularly important for anomaly detection because it represents the ratio of the number of anomalies found over the total number of abnormal instances and false negatives has larger cost than false positives. Also, we measure F1 score, as shown in Equation 12 because F1 score is base rate invariant and is able to evaluate the model performance for imbalanced dataset.

$$\text{precision} = \frac{TP}{TP + FP} \tag{10}$$

$$\text{recall} = \frac{TP}{TP + FN} \tag{11}$$

$$\text{F1} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{12}$$

4.3 Results

ARIMA Figure 5 shows ARIMA's performance on detecting point and contextual anomalies. The red point stands for real anomaly, while the bigger orange point represents detected anomaly. It shows that ARIMA successfully detected the contextual anomaly at the first red orange overlapping dot. And it also detected some point anomalies at other red orange overlapping dots. We also observed small contextual changes were detected at timestamps 311, 433, 695, 962, which is not labeled as anomaly but still has abnormal changes.

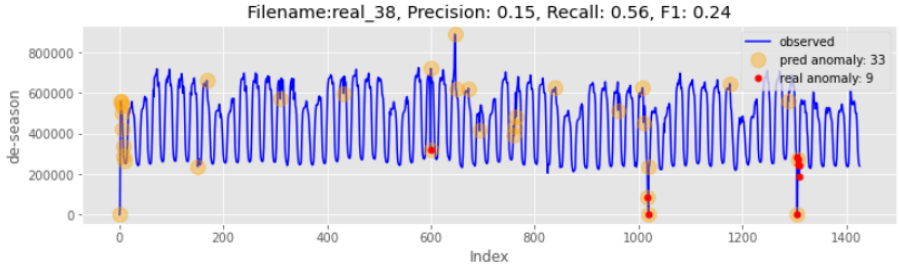


Figure 5: ARIMA Detected Anomaly on File 38

The Figure 6 shows ARIMA could hardly detect the collective anomaly. We think the reason is that ARIMA focuses on local data rather than a full picture. Therefore, it treats the collective anomaly as a new normal stage rather than an abnormal set. To check this idea, we used only De-seasonally data without fitting ARIMA to detect anomalies. Figure 7 shows that this method could successfully detect more collective anomalies but it also falsely classified some peaks as anomalies. The reason is that it didn't consider auto-correlations and local information. Therefore those points different from the whole picture will be detected as anomalies.

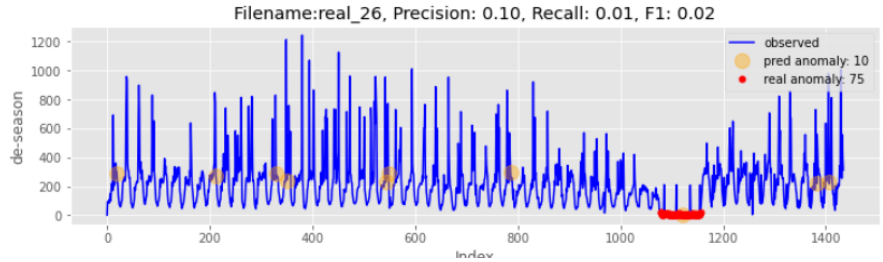


Figure 6: ARIMA Detected Anomaly on File 26

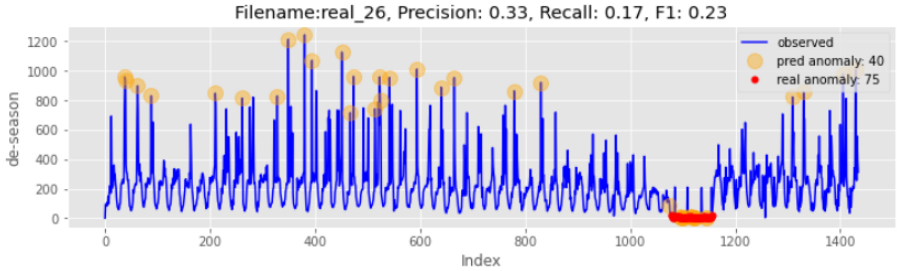


Figure 7: De-seasonality Detected Anomaly on File 26

C-LSTM Table 3 shows the performance results on Yahoo S5 web traffic dataset. From the results

Method	Accuracy	Precision	Recall	F1 score
C-LSTM + Tanh	**91.2**	55.9	**79.1**	65.5
C-LSTM + ReLU	89.6	48.9	72.9	58.5
CNN	89.2	47.7	71.1	57.1

Table 3: Performance comparison of C-LSTM and CNN

we can see that C-LSTM outperforms CNN model, and using tanh activation function is better than using ReLU as the activation function. Furthermore, C-LSTM achieves high performance for recall rate, which is desired for anomaly detection.

4.4 Discussion

In our experiments, we found domain knowledge is very important for the anomaly detection problem. For example, the standard to label an timestamp as anomaly. Daily and weekly web traffic changes will be treated as seasonality. Small contextual changes may not be labeled as anomaly, like Figure 5 shows. Sometimes a continues set of changes will be labeled as collective anomaly. However, it seems when this change happened for a period of time, it will not be labeled as anomaly anymore, like the Figure 8 shows. Those standard is related with particular industry and business context. Understanding those knowledge exactly will help use to choose model and set threshold.

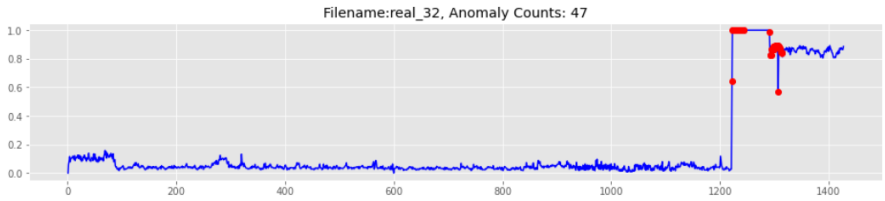


Figure 8: Labeled Anomaly on File 32

5 Conclusions and Future work

In conclusion, we proposed ARIMA, the statistical approach, and C-LSTM, an novel neural network architecture for anomaly detection in web traffic data. We've demonstrated that ARIMA performed differently on different categories of anomalies and C-LSTM could capture both spatial and temporal features from the time series data. Furthermore, the ARIMA model can function as an offline method because we can fit ARIMA over the entire static dataset and once p,d,q parameters are fixed, the model can predict next label sequentially. On the other hand, the C-LSTM can function as an online method. We can keep fine-tuning weights in C-LSTM when new dataset comes in and make use of any future data.

As for now, the anomaly detection is a binary classification problem on detecting whether the datapoint at this timestamp is an anomaly or not. In the future, we cam extend the problem to further classify the anomaly to different categories. As we've discussed in the Introduction section, different categories of anomalies can be mapped to different types of network attacks. With the knowledge of which category the anomaly belongs to, the model can further help with management over network attacks.

6 References

References

Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. A survey of network anomaly detection techniques. *Journal of Network and Computer Applications*, 60:19–31, 2016. doi: 10.1016/j.jnca.2015.11.016.

Peter Burman and Mark Otto. Census bureau research project: Outliers in time series. *Bureau of the Census*, 1988.

Min Cheng, Qian Xu, Jianming L.V., and et al. Ms-lstm: A multi-scale lstm model for bgp anomaly detection. *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, 2016. doi: 10.1109/icnp.2016.7785326.

Abram Jacob Fox. Outliers in time series. *Journal of the royal statistical society series b-methodological*, 34:350–363, 1972.

Tae-Young Kim and Sung-Bae Cho. Web traffic anomaly detection using c-lstm neural networks. *Expert Systems with Applications*, 106:66–76, 2018. doi: 10.1016/j.eswa.2018.04.004.

7 Student contributions

Xinli Gu: implemented C-LSTM model and CNN model, made presentation slides and wrote report
Di He: implemented ARIMA model, made presentation slides and wrote report
Bo Zhang: performed Exploratory Data Analysis and data pre-processing, made presentation slides and wrote report