Computers, Materials & Continua CMC, vol.61, no.2, pp.829-847, 2019

Long Short Term Memory Networks Based Anomaly Detection for KPIs

Haiqi Zhu[1], Fanzhi Meng[2, *], Seungmin Rho[3], Mohan Li[4, *], Jianyu Wang[1], Shaohui Liu[1] and Feng Jiang[1]

Abstract: In real-world many internet-based service companies need to closely monitor large amounts of data in order to ensure stable operation of their business. However, anomaly detection for these data with various patterns and data quality has been a great challenge, especially without labels. In this paper, we adopt an anomaly detection algorithm based on Long Short-Term Memory (LSTM) Network in terms of reconstructing KPIs and predicting KPIs. They use the reconstruction error and prediction error respectively as the criteria for judging anomalies, and we test our method with real data from a company in the insurance industry and achieved good performance.

Keywords: LSTM, anomaly detection, KPIs.

1 Introduction

In real-world many internet-based service companies that have online presences need to closely monitor large amounts of data in order to ensure stable operation of their business. We call these data KPIs (Key Performance Indicators, e.g., CPU usage or session logical reads per second) [Liu, Zhao, Xu et al. (2015)] of their applications and systems. Each company has thousands to millions of KPI curves from different server nodes or different applications. The KPI curves can show the current state of the server or applications data and reflect their state change to some extent. So we can infer whether the current system or application has anomalies based on the anomalies on the KPI curves, such as server down, servers overload, etc. Thus, anomaly detection techniques have been widely used to detect anomalous events timely to minimize the loss caused by such events. However, anomaly detection for these KPI curves with multiple shapes and patterns is a great challenge, especially for these unlabeled data **[**Xu, Feng, Chen et al. (2018)].

At present, a lot of literature exist about anomaly detection **[**Beal (2003); Bishop, Bishop and Bishop (2006); Chandola, Banerjee and Kumar (2009); Chen, Mahajan, Sridharan et

[1] School of Computer Science and Technology, Harbin Institute of Technology, Harbin, 150000, China.

[2] Institute of Computer Application, China Academy of Engineer Physics, Mianyang, 621900, China.

[3] Department of Software, Sejong University, Seoul, Korea.

[4] Cyberspace Institute of Advanced Technology, Guangzhou University, Guangzhou, 510006, China.

[*] Corresponding Authors: Fanzhi Meng. Email: mengfz@caep.cn;
Mohan Li. Email: limohan@gzhu.edu.cn

al. (2013)**]**. We will talk about the challenges encountered in anomaly detection in later chapters. Facing these challenges, we propose an efficient LSTM-based anomaly detection system that includes anomaly detection algorithm, KPI curves trend predict, etc. Due to the extreme imbalance of the data, we did not follow most current methods of anomaly detection that based classification models. They usually train a classification model by learning the distribution of normal and anomalous data and need labeled data that will take a lot of manpower and material resources. We find an abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment [Cheng, Xu, Tang et al. (2018)] We are inspired by that. In conclusion, we decide to solve the problem of anomaly detection with reconstruction normal data. It is also kind of a prediction-based perspective.

KPIs are streaming data aggregated at pre-defined time intervals (e.g., one minute or three seconds), thus are essentially time series. Therefore, we choose to solve the problem of time series anomaly detection from the perspective of prediction, which can also be converted into the problem of time series prediction. The amount of load on a server node or an application at a time may be unknown or change very frequently, for example, the volume of business for a particular holiday. In this case, it becomes difficult to predict the time-series, even a very short future, lead some prediction-based time-series anomaly detection models to be invalid or ineffective, such as ones based on exponentially weighted moving average (EWMA) **[**Michèle and Nikiforov (1993)**]**, support vector regression (SVR) **[**Ma and Perkins (2003)**]**, or recurrent neural network (RNN) **[**Yadav, Malhotra, Vig et al. (2016)].

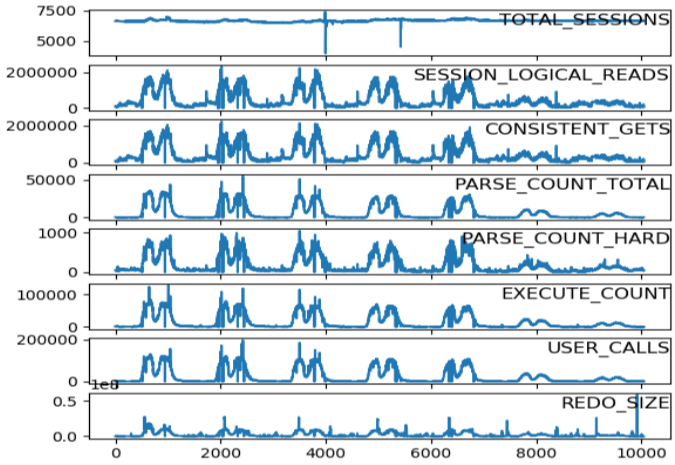


Figure 1: KPI curves in real world

In this paper, we adopt an anomaly detection algorithm based on Long Short-Term Memory (LSTM) Network in terms of reconstructing. Recurrent Neural Networks (RNNs), which are commonly used to process sequential data, have achieved great success and wide application in many Natural Language Processing (NLP). In the traditional neural network model, from the input layer to the hidden layer to the output layer, layers are fully connected, and the nodes between each layer are unconnected. This

kind of neural network structure has achieved great achievements in many yields, but due to its network structure problems, it fails to perform well when facing series problems or problems closely related to time. For example, if you want to predict what the next word in a sentence is, you usually need to use the previous word, because the words before and after in a sentence are not independent. RNNs are called cyclic neural networks, meaning that the current output of a sequence is also related to the previous output. The concrete manifestation is that the network will remember the previous information and apply it to the calculation of the current output, that is, the nodes between the hidden layer are no longer connected but connected, and the input of the hidden layer includes the output of the hidden layer at the previous time as well as the output of the hidden layer. In theory, RNNs can process any length of sequence data (not a variable length time series). In practice, however, to reduce complexity, it is often assumed that the current state is only related to the previous states. As a neural network model specially dealing with time series, LSTM solve the long-term dependence problem of RNNs to a certain extent and becomes the best choice of our method.

2 Related work

In this section, we will first introduce the background and the mainstream methods of anomaly detection. And then we will present the goals, challenges of our proposed method and some methods we tried.

2.1 Background

Traditional Statistic-based models. As a traditional method, quite a few statistic-based models have been proposed and applied. Traditional statistic-based models often need to assume that the data conform to a certain distribution. The statistical anomaly detection techniques are based on the key assumption: Normal data instances occur in high probability regions of a stochastic model, while anomalies occur in the low probability regions of the stochastic model [Chandola, Banerjee and Kumar (2009)]. Statistical techniques fit a statistical model (usually for normal behavior) to the given data and then apply a statistical inference test to determine if an unseen instance belongs to this model or not. This method can only fit the data distribution through normal data, and can identify other abnormal data besides normal data, but it is often very demanding on data. If the assumptions regarding the underlying data distribution hold true, statistical-based models provide a statistically solution for anomaly detection. But this assumption often does not hold true, especially for high dimensional real data sets. Even when the statistical assumption can be reasonably justified, there are several hypothesis test statistics that can be applied to detect anomalies, choosing the best statistic is not an easy task.

Supervised ensemble methods. Typical supervised ensemble methods, EGADS [Laptev, Amizadeh and Flint (2015)] and Opprentice [Liu, Zhao, Xu et al. (2015)], have been proposed in recent years. The approach also chooses train anomaly classifiers using the user feedbacks as labels and using anomaly scores output by traditional detectors as features to solve the problem for anomaly detection. These methods shown promising results, but they all rely heavily on good labels. Moreover, running multiple traditional

classification detectors is bound to cause a lot of computing overhead, which is a practical concern.

Unsupervised and deep learning methods. Recently, due to the lack of labels in data, there is an increasing trend to adopt unsupervised machine learning algorithm to solve the problem for anomaly detection, e.g., one-class SVM [Amer, Goldstein and Abdennadher (2013); Erfani, Rajasegarar, Karunasekera et al. (2016)], clustering based methods [Fu, Hu and Tan (2005)] like K-Means [Münz, Li and Carle (2012)] and GMM [Laxhammar, Falkman and Sviestins (2009)], KDE [Heywood, Mcdermott, Castelli et al. (2016)], and VAE [An and Cho (2015)] and VRNN [Sölch, Bayer, Lundersdorfer et al. (2016)]. These technologies focus on normal data rather than abnormal data, since KPI usually contain a large amount of normal data and lack abnormal data (unlabeled), and the model can be trained without labels. In general, we make the model recognize normal sequence patterns or their latent representations, and then calculate the anomaly score by measuring the distance between an observation and the normal pattern.

2.2 Problem and goal

Anomaly detection is an important issue in different research fields and application fields. Many anomaly detection techniques have been specially developed to deal with some problems in professional application fields. Anomaly detection refers to finding problems in the data that do not conform to the expected behavior pattern. These nonconforming patterns are called outliers or anomalous value in different application areas. Anomaly detection is widely used in fraud detection of credit card, insurance or medical insurance, network security intrusion detection, failure detection of security critical systems, and military surveillance of enemy activities. The key of anomaly detection is to transform the exception in data into the key information in the wide application field. For example, abnormal flow patterns in computer networks may mean that the hacked computer is sending sensitive data to unauthorized destinations, abnormal medical images may indicate the presence of malignant tumors, or abnormal readings of spacecraft sensors may indicate some missing fault on the spacecraft.

In some sense, anomaly is defined as patterns that do not conform to expected normal behavior. Therefore, the intuitive anomaly detection method is to define a regional representation of normal behavior and declare any observation in the data that does not belong to the normal region as abnormal. But there are several factors that make this deceptively simple approach very challenging.

In general, anomaly detection can be regarded as a classification problem of data imbalance. Therefore, if data conditions allow, the use of supervised anomaly detection is preferred. In the case of only a few labels, the semi-supervised anomaly detection model can also be adopted. For example, unsupervised learning is used as a feature extraction method to assist the supervised learning. This method can also be understood as feeding the supervised classification model after preprocessing the data through unsupervised feature engineering. However, in reality, anomaly detection problems are often unlabeled,

and the data does not indicate which ones are abnormal. Therefore, more attention is currently focused on unsupervised learning.

In addition to the problem of data imbalance, time series anomaly detection still has several great challenges. It is very difficult to define a region containing all possible normal behaviors. The boundary between normal and abnormal is usually not clear. In many areas, normal behavior is evolving, and the current concept or scope of normal behavior may not be representative in the future. And for different application areas, the concept of exception is different. Small fluctuations in medical temperature, for example, may be abnormal, but small fluctuations in stock prices in the stock market may be artificially normal. Therefore, applying mature methods from one domain to another is not an easy task.

In view of the shortcomings of current time series anomaly detection algorithms, this paper proposes a time series anomaly detection method based on LSTM Encoder-Decoder, which is trained to reconstruct instances of normal data with the predict time-series being the input time-series itself. And the reconstruct error is used as a standard for anomaly detection. Our method avoids the manual annotation of data that requires a lot of manpower and material resources. The anomaly detection and prediction of KPI curves can be realized only by using normal time-series to ensure the stable operation of the business.

2.3 Some methods

In the actual problem of the KPI curves anomaly detection, we can be only provided with normal data, and there is no abnormal data or even any abnormal data that can be provided us. Therefore, we consider and try different approaches from multiple perspectives. Most anomaly detection problems are still solved by classification, which can not only detect anomalous, but also detect different types of anomalous. However, due to the particularity of KPI curves, this classification method can't solve the problem of abnormal detection of KPI curves. Therefore, we choose the method based on the prediction to solve the problem. Firstly, we choose a traditional ARIMA model suitable for time series analysis and SVR regression method.

There are four commonly used time series models: Autoregressive model AR(p), Moving average model MA(q), Autoregressive moving average model ARMA(p,q) and ARIMA(p, d, q). It can be said that the first three are all special forms of ARIMA(p, d, q) models. ARIMA model, Auto Regressive Integrated Moving Average model, is built on the basis of stable time series, so the stability of time series is an important prerequisite for modeling. Generally, ADF unit root test model is used to test the stability of time series model. Of course, if the time sequence is not stable, but can be by some action to make the stable time series (such as the exponential difference), then the ARIMA model prediction, the stability of time series prediction results, and then the predicted results are

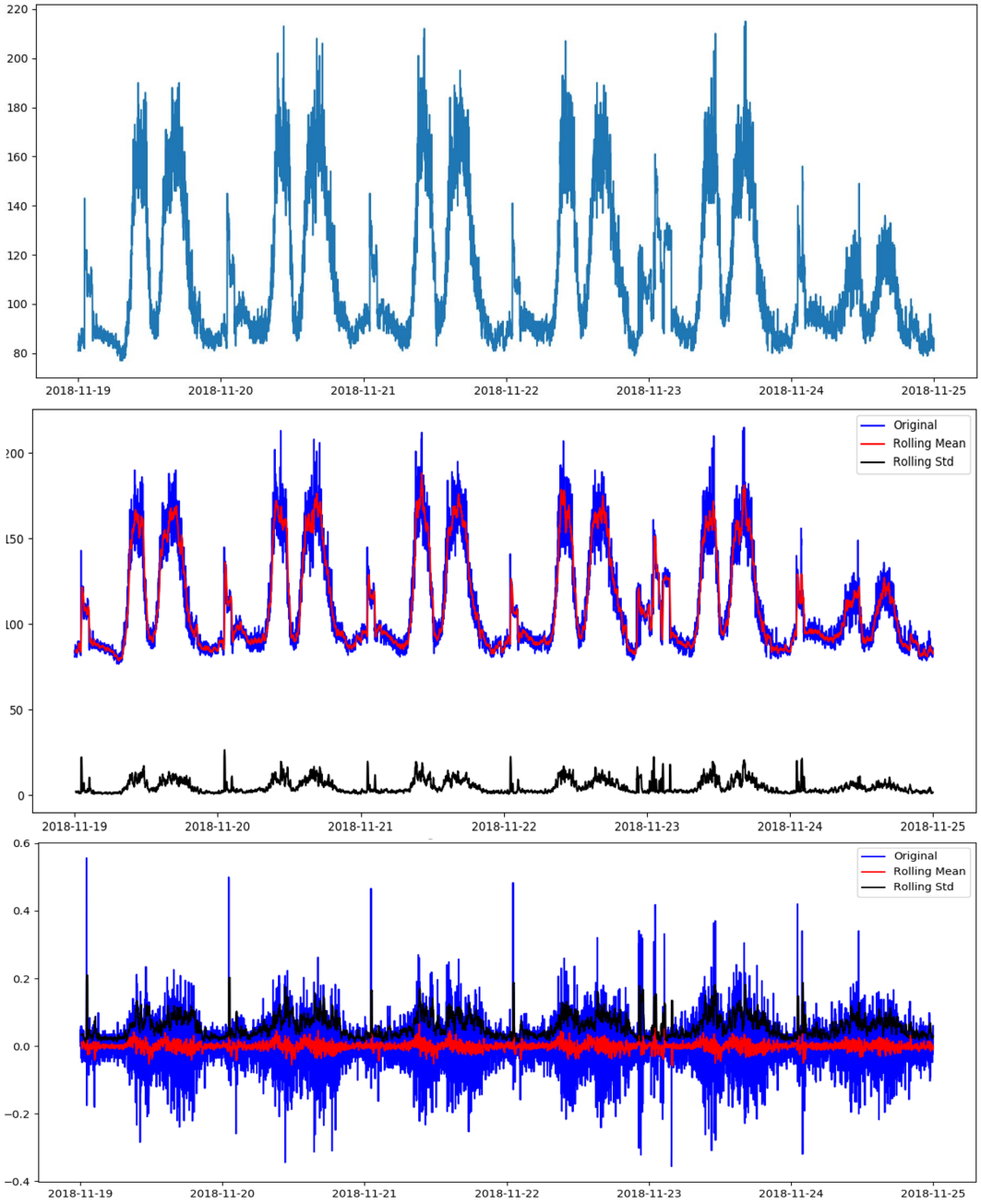


Figure 2: The first row is the original KPI curves, the second row is the comparison between the original and the rolling mean and the rolling std, the last row is the data after the difference before make the sequence of the stable operation of the inverse operation (index, differential inverse operation), can get the original data of predicted results

In the ARIMA model method, we selected a continuous time series with a length of 6 days and a frequency of 1 minute. The ARIMA model requires a stationary time series. In our observation, the KPI curves is not a stationary time series. So, the first thing we need to do is to make the difference of the time series until we get a stationary time series. We made one difference on the KPI curves to obtain a relatively stable sequence, and then determined the d (difference times) of the parameters in the ARIMA model.

After obtaining the stationary sequence, we need to obtain the autocorrelation coefficient ACF and partial autocorrelation coefficient PACF for the stationary sequence respectively to determine the parameters p and q of the model.

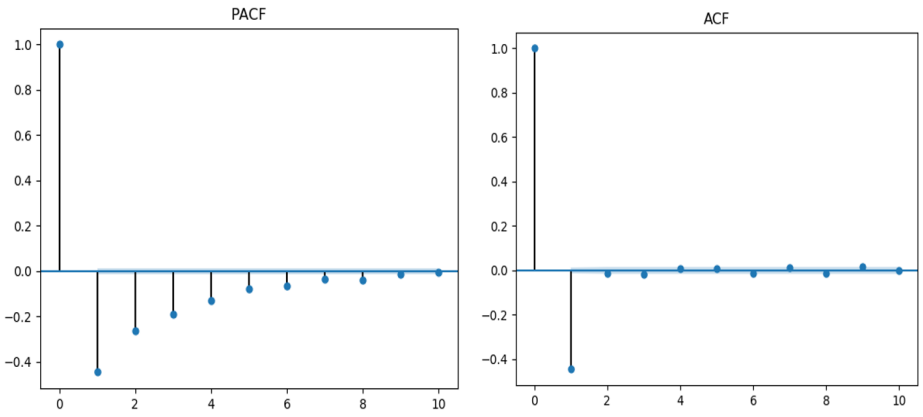


Figure 3: the autocorrelation coefficient ACF and partial autocorrelation coefficient PACF for the stationary sequence

After the differential processing of the data and the selection of other parameters, we get a more appropriate ARIMA model. And we also got a relatively good result that can well predict the normal trend of KPI curves (refer Fig. 4). But as we can see from the figure, the results predicted by this method are the same almost every day, and they cannot adopt to the changes in the normal pattern of data. For example, in the position circled in the figure, these two spikes are caused by a large number of people logging into the system when the working hours begin in the morning, but on the rest day, no one or a small number of people logging in, this spike will not occur, which is not an abnormal situation. But this method results suggest that there will be a spike here.

As the same time, we found that not all kinds of KPI curves can be processed into a stable time series. However, the ARIMA model can only deal with stable time series or those that are stable after differentiation. Moreover, the ARIMA model is inherently intelligent in capturing linear relationships, but not non-linear ones (in KPI curves, many trends are non-linear). Many KPI curves are not only influenced by themselves, but also by many external factors that we can't collect. They will fluctuate randomly with these influences.

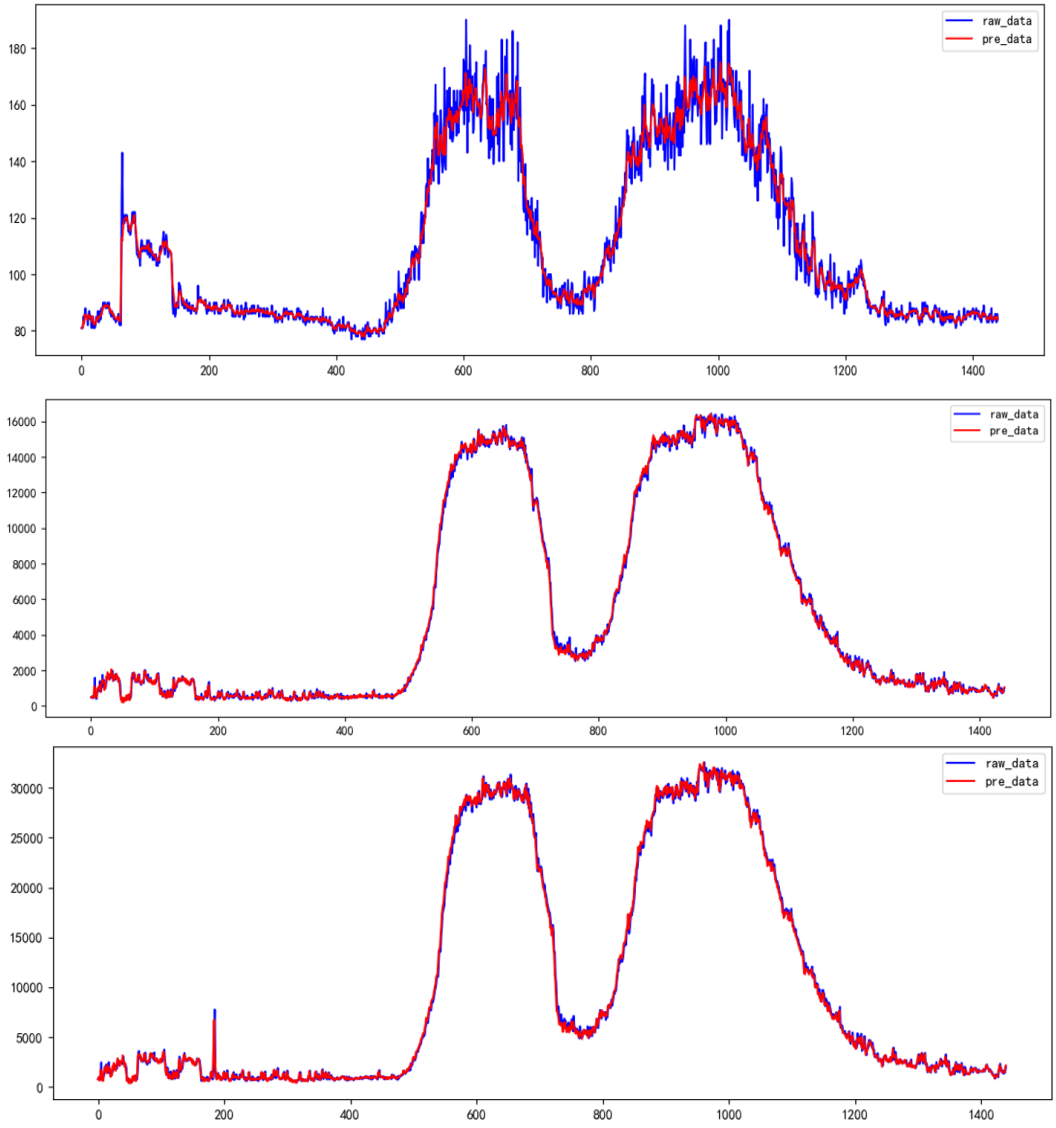


Figure 4: The prediction results of three KPI curves obtained by the ARIMA model

According to the observation of historical data, KPI curves has certain regularity to follow. It is usually related to the business of the company. Every day, there will be two big peaks in the morning and afternoon working hours, one trough in the lunch break, and a stable minimum state in the midnight. Therefore, it is reasonable to think that such a possibility, there are some ways to find out what the pattern is and to describe it in a complex function, exists. So, we choose the method of regression to try to find out whether there are some fixed rules in the historical data for us to predict the data of the next time point, and describe the trend and slight changes of the whole time series.

In the traditional regression method, the prediction is only considered correct if $f(x)$ is completely equal to y. For example, in linear regression, the mean square error is often used to calculate the loss. However, SVR (Support Vector Regression) believes that as long as the deviation between $f(x)$ and y is not too large, it can be considered that the prediction is correct and the loss is not calculated. Specifically, the threshold value a, is set to calculate the loss of data points that meet the requirements, such as $|f(x) - y| > a$. We all believe that model is accurate in predicting the data points in the shaded part, and only calculate the loss of data points outside the shaded part (refer Fig. 5).

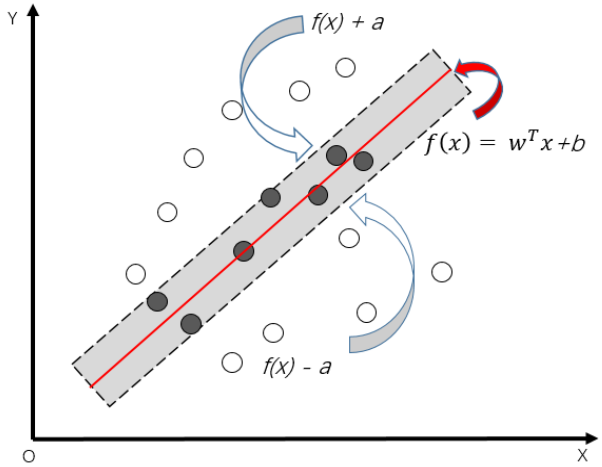


Figure 5: Principle of SVR

In the regression method, we choose three kernel functions which are commonly used: 'RBF', 'Linear', 'Poly'. The results of three different kernel functions are shown in the Fig. 6. Among them, the RBF and Linear kernel functions can only describe the basic trend of KPI curves and the results of Poly function deviate greatly from the real data. But it cannot be used as the basis of anomaly detection. And when making predictions about one of the KPI curves, you need as much information about all the other dimensions as possible. However, in the process of data processing, we find that there is no direct correlation between each curve, that is to say, it is not a simple relationship between variables and independent variables, and there are also many variables that we cannot collect and will have an impact on the result variable. In addition, when there is abnormality in one of the dimensions, the results we calculated will also produce abnormality, which violates the rule that we describe the normal data pattern and use the mutation of indicators as the detection condition.

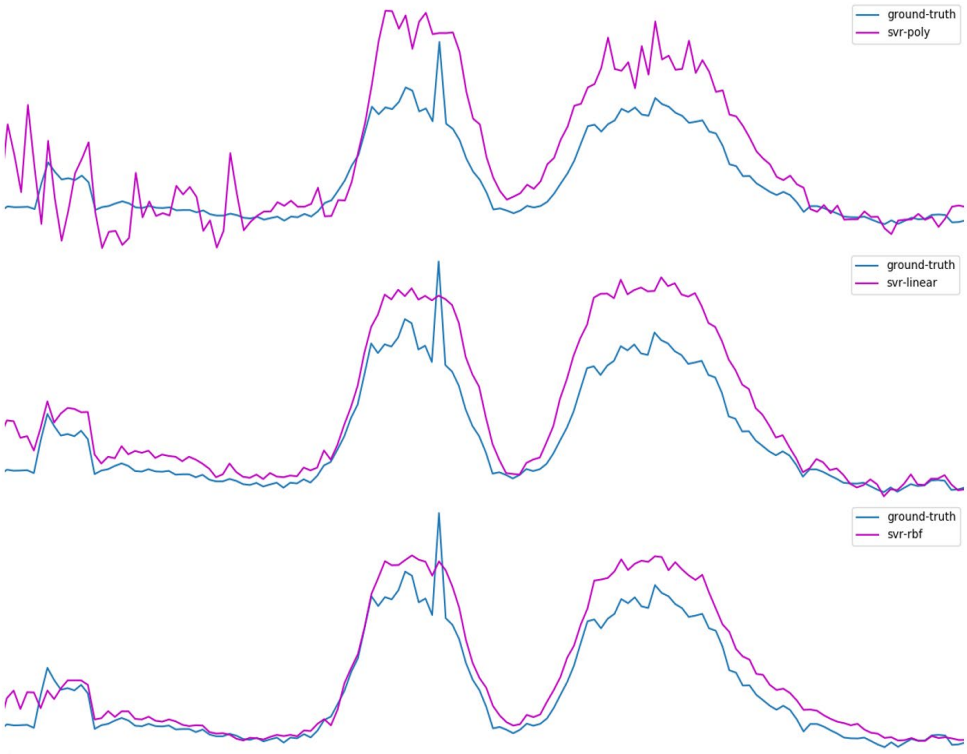


Figure 6: Prediction results of three KPI curves obtained by SVR

3 LSTM-based anomaly detection

In this section, we will first present the architecture of LSTM, and then describe our design details.

In order to solve the problem of long-term dependence, LSTM units have been developed to replace hidden layer neurons in traditional RNN. A typical LSTM cell contains one or more memory cells with internal state, an input gate i_t, a forgetting gate f_t, and an output gate o_t, as shown in Fig. 7. Assuming that s_t is the state of the memory cell at time t, the calculation process of this LSTM unit at time t is as follows:

$$\begin{cases} s_t = s_{t-1} f_t + i_t g_t \\ o_t = sigmoid\left(W^{xo} x_t + W^{ho} h_{t-1} + b_o\right) \\ h_t = o_t \tanh(s_t) \end{cases} \tag{1}$$

where g_t is the input extrusion unit, W^{xo} and W^{ho} respectively represent the weight matrix between x_t、 $h_{t-1}$and the output gate unit, b_o is the bias of the output gate unit, and *tanh* is the activation function. At this point, by changing the state of the forgotten gate unit at time t (0 or 1), the effect of controlling the network's hidden layer output h_t, can be achieved to remember the sequence's long-term reliance on information.

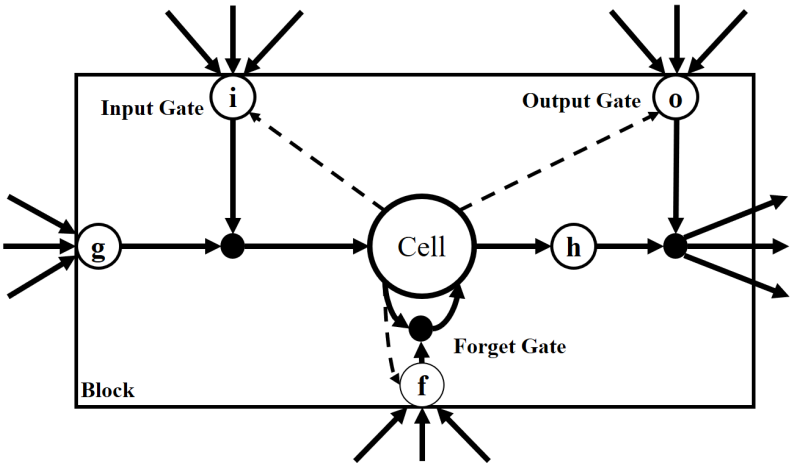


Figure 7: LSTM cell structure

We collected data from the company's server nodes and application nodes with a frequency of 60 Hz through the mature Zabbix software currently on the market. The KPI curves is different from other time-series. They usually reflect the running state of the server nodes or application nodes and the business level of the company, which have some special laws, and they have some kind of periodicity. So simple ways to set thresholds are inaccurate. Consider a time-series $\mathbf{X} = \left\{x^{(1)}, x^{(2)}, \ldots, x^{(L)}\right\}$ of length L, where each point $x^{(i)} \in R^m$ is an m-dimensional vector of readings for m variables at time-instance t_i. We consider that such time-series are available and can be obtained by taking a window of length L over a larger time-series. We considered two experimental schemes. First, we train the basic LSTM to learn the normal time-series and predict the next time point. The predict error are used to determine abnormal. Second, we choose LSTM Encoder-Decoder as the basic model, train the model to reconstruct the normal time-series. The reconstruction errors are then used to be the anomaly score of a point. A higher error indicates a higher likelihood of the point being anomalous.

3.1 LSTM-based predict model

In this method, we consider a prediction model learns to predict the next values for the input variable. We first learn a prediction model using LSTM networks, and then compute the prediction error using which we detect anomalies. We establish an LSTM prediction model for each KPI curve. We use stacked LSTM architecture to predict our KPI curves. We consider the following LSTM network architecture: We take one unit in the input layer for each of dimensions, one unit in the output layer. The LSTM units in a hidden layer are fully connected through recurrent connections. We stack LSTM layers s.t. each unit in a lower LSTM hidden layer is fully connected to each unit in the LSTM hidden layer above it through feedforward connections (refer Fig. 8). The prediction model is learned using the normal time-series.

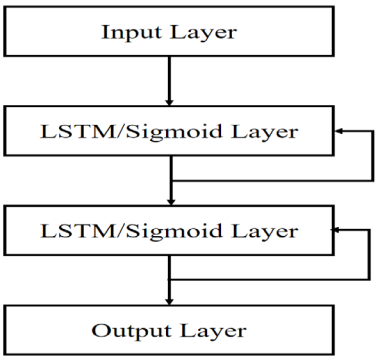


Figure 8: The Stacked Architecture

3.2 LSTM Encoder-Decoder based reconstruct model

On the basic of the LSTM-based predict model, we train an LSTM Encoder-Decoder to reconstruct the normal time-series. The encoder is used to learn a stable length vector representation of the input normal time-series that is used to reconstruct the time-series using the current hidden state and the value predicted at the previous time-step. Given X, $h_E^{(i)}$ is the hidden state of encoder at time t_i for each $i \in \{1,2,\ldots,L\}$, where $h_E^{(i)} \in R^c$, c is the number of LSTM units in the hidden layer of the encoder. We train the encoder and decoder jointly to reconstruct the time-series in reverse order, i.e., the target time-series is $\{x^{(L)}, x^{(L-1)}, \ldots, x^{(1)}\}$. We use a linear layer on the top of the LSTM decoder layer to predict the target, while the final state $h_E^{(L)}$ of encoder is used as the initial state for the decoder. During the state of training, we use $x^{(i)}$ as the input to obtain the decoder's previous hidden state$h_D^{(i-1)}$, and then predict $x'^{(i-1)}$ corresponding to the target$x^{(i-1)}$. Similarly, in the derivation, the predict value $x'^{(i)}$ is used as input to obtain the decoder's hidden state $h_D^{(i-1)}$ and predict $x'^{(i-1)}$. Our model's objection is minimizing $\sum_{X \in s_N} \sum_{i=1}^{L} \left\| x^{(i)} - x'^{(i)} \right\|^2$, while the s_N is the set of normal training time-series.

In the Fig. 9, we depict the inference steps in the LSTM Encoder-Decoder reconstruction model for a sequence with $L = 3$. At time t_i, the value $x^{(i)}$ and the hidden state $h_E^{(i-1)}$ of time t_{i-1} are used to obtain the hidden state $h_E^{(i)}$of encoder. The hidden state $h_E^{(i)}$ of the encoder at the end of the input sequence is used as the initial state $h_D^{(i)}$ of the decoder s.t. $h_E^{(3)} = h_D^{(3)}$. We compute the prediction $x'^{(i)} = w^T h_D^i + b$. In the decoder, we use $h_D^{(i)}$ and the prediction $x'^{(i)}$ to obtain the next hidden state $h_D^{(i-1)}$.

In the two methods mentioned above, reconstruction error and prediction error are used as the criteria for anomaly detection.

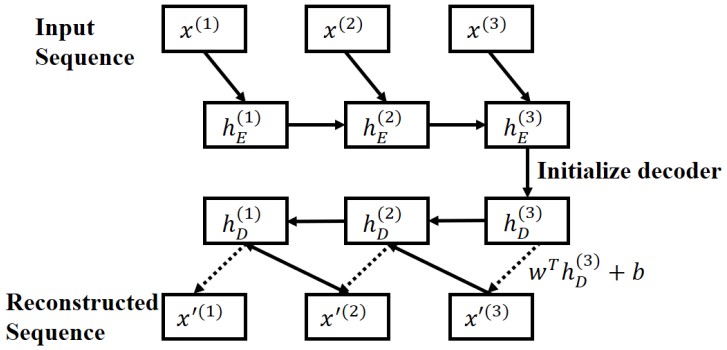


Figure 9: LSTM Encoder-Decoder inference steps for a sequence with L=3

4 Experiments

Our data comes from an insurance industry company, which has thousands of KPI curves and lacks abnormal data. We selected four KPI curves as our experimental data according to the needs of the current company's business. The time span is three months continuous time series.

First, we preprocess our data. It is common that KPIs have missing value. However, according to our observation, the percentage of missing value of each KPI curve is very small. We simply use linear interpolation to fill them based on their adjacent data point. Another important preprocessing step is data normalization: $\hat{x} = \frac{x_t - min_x}{max_x - min_x}$, where x_t is the raw data, and max_x, min_x is the maximum and minimum of x_t. There is another way we can do this with data standardization :$\hat{x} = \frac{x_t - \mu_x}{std_x}$, where x_t is the raw data, μ_x is the mean of data, and std_x is the standard deviation of x_t. As discussed in Erfani et al. [Erfani, Rajasegarar, Karunasekera et al. (2016)], time-series must be normalized in order to make meaningful comparison between them. KPI curves are mostly concentrated in a small range. If we do not normalize the data, the result we get in the prediction-based model may be a fixed value, which is not what we expect.

We divided the data of three months into one day and one week respectively. We can observe from the data of one day and one week that the KPI curve really has this kind of cyclical rule. (As the Fig. 10) And these indicators are in line with people's normal schedule, they have a high value at work time and a low value at midnight when the business is not running. We select four curves in the KPI to conduct our experiment in the stack LSTM-based model. We calculate the average of predict error and the mean of predict error. The point with high prediction error is considered abnormal. We can see the results in the Fig. 11.

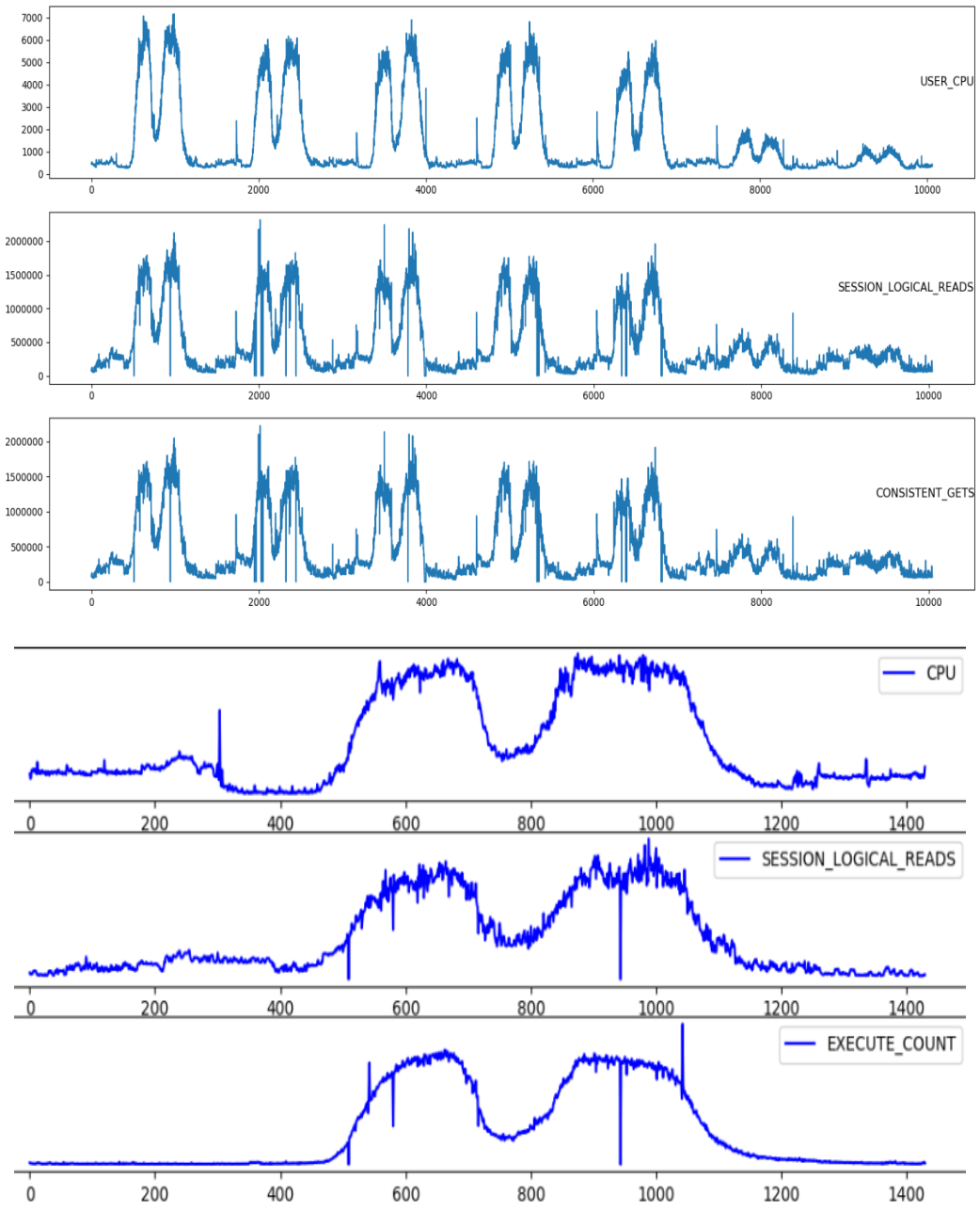


Figure 10: Compared data per day with data per week. The top is data per week, and the below is data per day

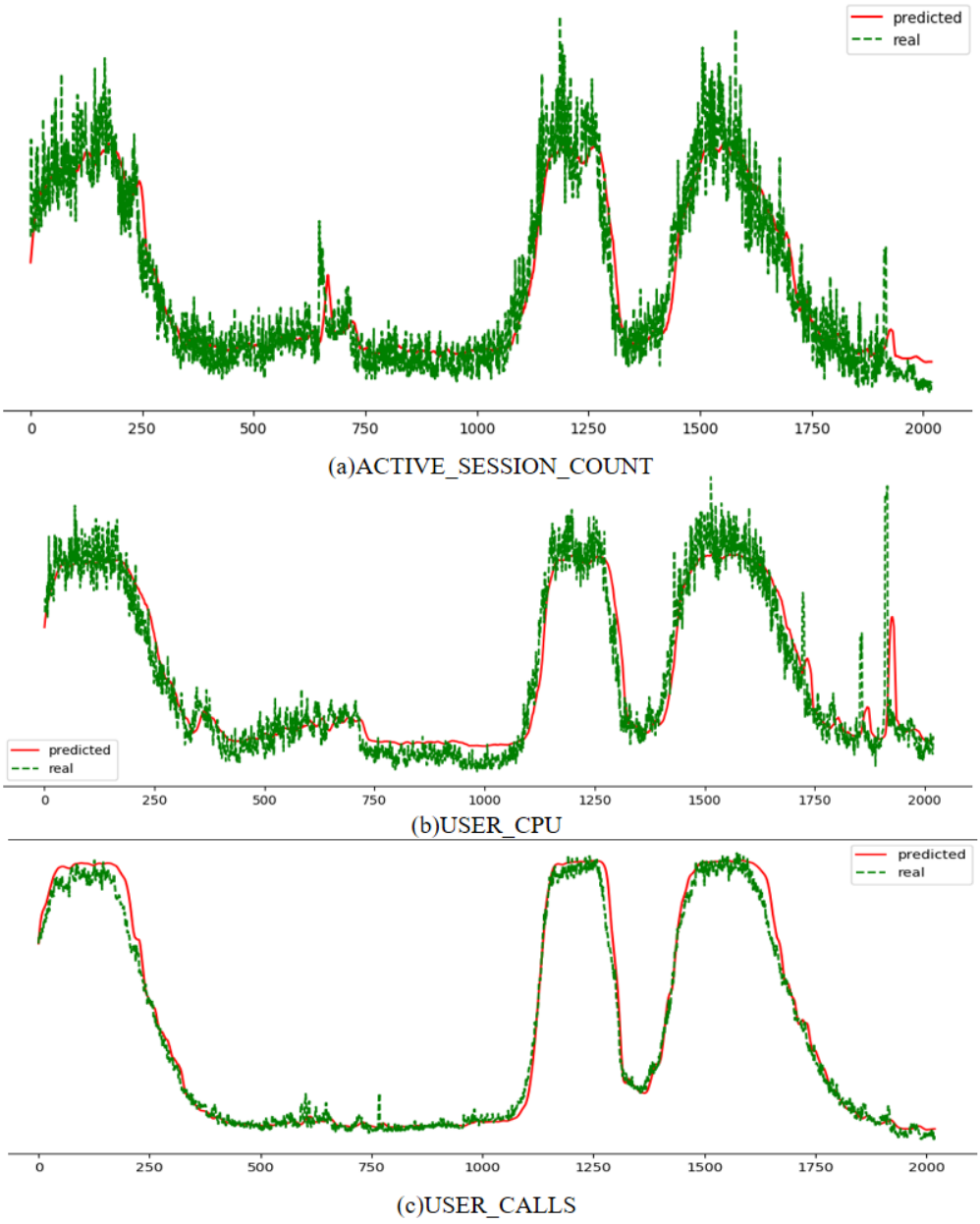
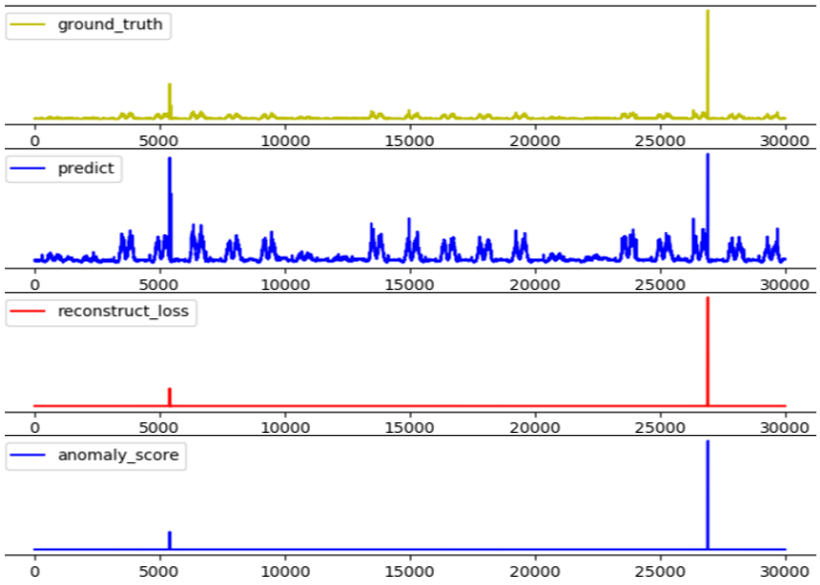
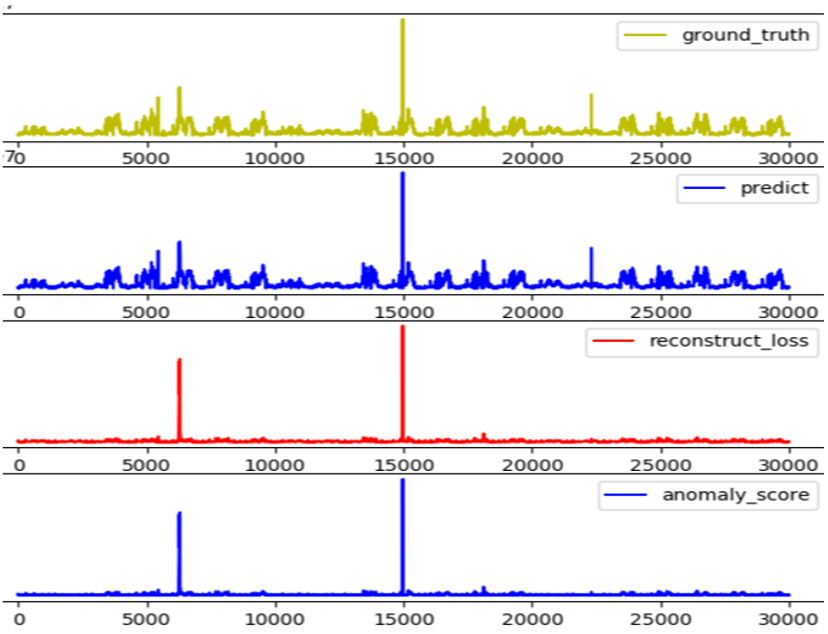


Figure 11: Stack LSTM-based predict model results. On the three figures, green line is the predict value and red line is the raw value

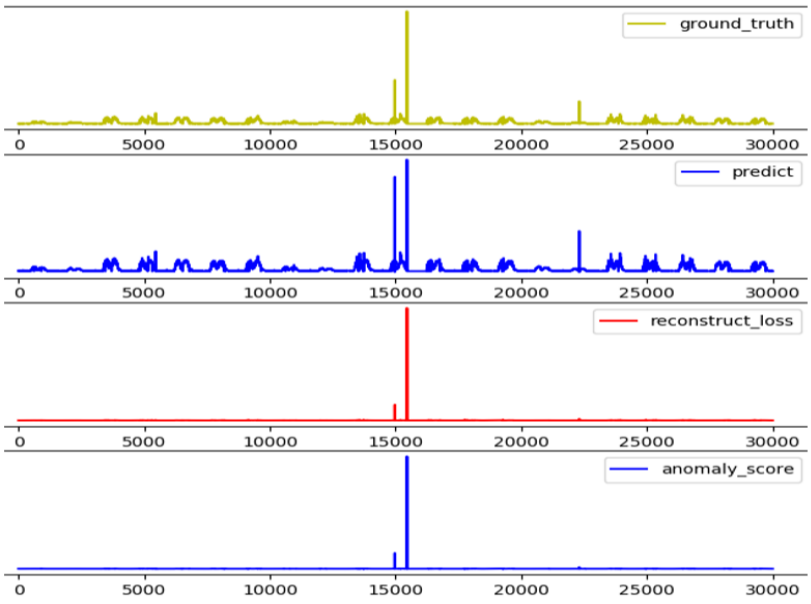
We chose the same four KPI curves to verify our experiment in the LSTM Encoder-Decoder reconstruction model. This time, we calculated the reconstruction error and anomaly score (according to the distribution of reconstruction error, the high anomaly score represents the high possibility of the anomaly of this point). We can see the results in Fig. 12.



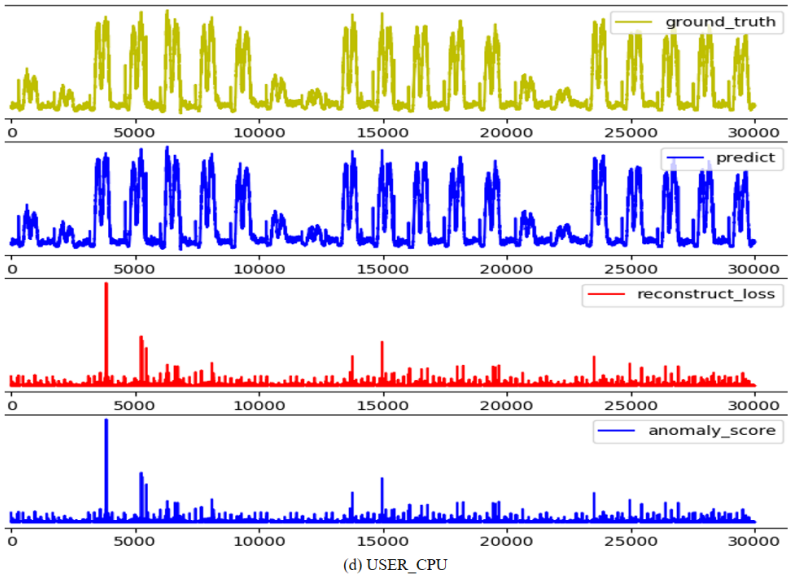
(a) ACTIVE_SESSION_COUNT



(b) SESSION_LOGICAL_READS



(c) EXECUTE_COUNT



(d) USER_CPU

Figure 12: LSTM Encoder-Decoder based reconstruction model results. In the first line of Fig. (a), we show the ground truth, and the second is the reconstruction time-series. The third and fourth lines present the reconstruction error and anomaly score. The figure (b) (c) (d) is similar

From the perspective of test results alone, both methods can achieve the desired effect to some extent. We are satisfied with the result of prediction model and reconstruction model. We also compared mean squared error, mean absolute error and R2-score of two methods as criteria for evaluating the two models (Since the methods approach the

problem from the perspective of prediction, the three standards can reflect the quality of the two kind of model. We calculate these standards using normal data as a test set). Show in Tab. 1 (We use the normalized data, because the actual value are millions, and the calculated results will be very large).

Table 1: Compare model

	Stack LSTM	LSTM Encoder-Decoder
MAE	0.0219	0.0126
MSE	0.0016	0.0019
R^2-SCORE	0.8512	0.8781

5 Conclusion

In this paper, we propose a LSTM-based anomaly detection method for KPI curves. We show that stack LSTM based prediction model predicted the normal KPI curves and LSTM Encoder-Decoder based reconstruction model learnt over normal KPI curves can be the viable approaches to detect anomalies in KPI curves. However, many existing models for anomaly detection rely on the fact that the work need favorable label, effective assumption for data, a clear understanding of the data and so on. Our method is shown to detect anomalies any KPI curves, and hence may be more rapid and robust compared to exiting models.

At present our approach is aimed at a KPI curve and train a model. We first test version contains only 12 KPI curves. The calculation level also in our hardware to withstand range, but the fact that any one server node of an enterprise has hundreds or even thousands of KPI curves, and the number of enterprise's application nodes and sever nodes is immeasurable. Such calculation amount is far beyond our estimation, which is also beyond our current calculation level. For such problem, we need to further study the general model to reduce calculation amount and cost. It is a good choice to greatly reduce the number of models by classifying KPI curves.

Acknowledgement: This work was supported by Harbin Institute of Technology, China Academy of Engineer Physics, College of Software and Convergence Technology of Sejong University and Guangzhou University.

References

An, J. W.; Cho, S. Z. (2015): Variational autoencoder based anomaly detection using reconstruction probability. *Technical Report. SNU Data Mining Center*, pp. 1-18.

Amer, M.; Goldstein, M.; Abdennadher, S. (2013): Enhancing one-class support vector machines for unsupervised anomaly detection. *Acm Sigkdd Workshop on Outlier Detection & Description*, pp. 8-15.

Basseville, M.; Nikiforov, I. (1993): Detection of abrupt changes-theory and application. *Prentice Hall*.

Beal, M. J. (2003): *Variational algorithms for approximate bayesian inference* (*Ph.D. Thesis)*. University of London.

Bishop, C. M.; Bishop, C.; Bishop, C. (2006): *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc.

Chandola, V.; Banerjee, A.; Kumar, V. (2009): Anomaly detection: a survey. *ACM Computing Surveys*, vol. 41, no.3, pp. 15.

Chen, Y.; Mahajan, R.; Sridharan, B.; Zhang, Z. L. (2013): A provider-side view of web search response time. *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 243-254.

Erfani, S. M.; Rajasegarar, S.; Karunasekera, S.; Leckie, C. (2016): High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning. *Pattern Recognition*, vol. 58, pp. 121-134.

Fu, Z.; Hu, W.; Tan, T. (2005): Similarity based vehicle trajectory clustering and anomaly detection. *IEEE International Conference on Image Processing*, vol. 2.

Heywood, M. I.; Mcdermott, J.; Castelli, M.; Costa, E.;Sim, K. (2016): One-Class classification for anomaly detection with kernel density estimation and genetic programming. *Springer International Publishing*, vol. 10, pp. 3-18.

Cheng, J. R.; Xu, R. M.; Tang, X. Y.; Victor, S.; Cai, C. T. (2018): An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. *Computers, Materials & Continua*, vol. 55, no. 1, pp. 95-119.

Laptev, N.; Amizadeh, S.; Flint, I. (2015): Generic and scalable framework for automated time-series anomaly detection. *Proceeding of the 21th ACM Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 1939-1947.

Laxhammar, R.; Falkman, G.; Sviestins, E. (2009): Anomaly detection in sea traffic-a comparison of the gaussian mixture model and the kernel density estimator. *International Conference on Information Fusion*, pp. 756-763.

Liu, D.; Zhao, Y.; Xu, H.; Sun, Y.; Pei, D. et al. (2015): Opprentice: towards practical and automatic anomaly detection through machine learning. *Internet Measurement Conference*, pp. 211-224.

Ma, J.; Perkins, S. (2003): Online novelty detection on temporal sequences. *ACM Sigkdd International Conference on Knowledge Discovery & Data Mining*, pp. 613.

Münz, Gerhard; Li, S.; Carle, G. (2012): Traffic anomaly detection using k-means clustering. *GI/ITG Workshop MMBnet*, pp. 13-14

Sölch, M.; Bayer, J.; Ludersdorfer, M.; Patrick, V. D. S. (2016): Variational inference for on-line anomaly detection in high-dimensional time series. arXiv:1602.07109.

Xu, H.; Feng, Y.; Chen, J.; Wang, Z.; Qiao, H. et al. (2018): Unsupervised anomaly detection via variational auto-encoder for seasonal KPIs in web applications. *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pp. 187-196.

Yadav, M.; Malhotra, P.; Vig, L.; Sriram, K.; Shroff, G. (2016): Ode-augmented training improves anomaly detection in sensor data from machines. arXiv:1605.01534.