Journal of Physics: Conference Series

PAPER • OPEN ACCESS

Anomaly detection based on temporal convolution Autoencoders

To cite this article: Lihao Wu and Jiahui Liang 2022 *J. Phys.: Conf. Ser.* **2366** 012041

Anomaly detection based on temporal convolution Autoencoders

Lihao Wu*[a], Jiahui Liang[a]

[a]Guangzhou City University of Technology, Huadu District, Guangzhou, China, 510800;

[*] Corresponding author: wulh@gcu.edu.cn

ABSTRACT: Due to the rapid growth of the number of sensors in modern industrial society, detecting outliers in time series has become very important. And unsupervised detection of outliers in time series is a very challenging task. For most detection tasks of time series outliers, the autoencoder is one of the main choices. And the recursive network is usually used in the self-coding network structure. However, recent studies have shown that the network structure using dilated causal convolution performs better than the recursive network in all kinds of sequence modeling. In this paper, the encoder and decoder are network structures based on extended causal convolution The time series are reconstructed and then compared with the original data to calculate the distance between them, so as to identify the outliers. We use the Temporal convolution autoencoders to evaluate anomaly data sets in multiple time series. Our results show that the Temporal convolution autoencoders has better anomaly detection ability. In addition, we learned that the combination of Feature Engineering and super parameters will also have a great impact on the results, so ablation experiments need to be carried out carefully.

1. Introduction

In the research of time series anomaly detection, autoencoders are usually used as the standard model. And in autoencoders, RNN [1], LSTM [2] and other cyclic structure networks are used as the standard components. RNN and LSTM are typical sequence modeling network structures, which are famous for their effectiveness in sequence modeling. However, recent research shows that convolution structure can also achieve better accuracy than traditional methods in many sequence modeling tasks, which shows that to obtain the time characteristics of time series, it is not necessary to use a sequence model, and convolution network structure can be considered.

We explore this problem through systematic experiments on various time series anomaly detection data sets. We design an encoder-decoder architecture based on temporal convolution network (TCN) [3]. This architecture reconstructs errors by reconstructing time series, so as to achieve the purpose of anomaly detection. In this process, it maintains simplicity. Moreover, compare with traditional sequence modeling networks such as RNN and LSTM, it combines the best parameter adjustment by the practice of modern deep learning. We believe that the self-encoder can reconstruct the input very well. It should also create a useful and meaningful potential representation.

The results show that the encoder-decoder architecture based on the temporal convolution network is suitable for many different time series anomaly detection tasks, and the results are better than many baseline models. In order to further optimize the self-encoder, we also try to introduce the attention mechanism into TCN.

2. Related work

2.1 AutoEncoder

The autoencoders model is a kind of neural network. Its purpose is to learn a function without supervision, reconstruct the input and compress the data in the process, so as to find a more effective hidden space representation. The network can be seen as consisting of two parts: an encoder function and a reconstructed decoder as shown in figure 1 [4]. The encoder network aims to convert the original high-dimensional input into a potential low-dimensional space, and its input size is larger than the output size. The goal of the decoder network is to reconstruct data from this low dimensional space, and its output layer may become larger and larger. Therefore, the function of encoder network is dimension reduction. In addition, the automatic encoder can optimize the data reconstruction in the code. A good intermediate representation can not only capture potential variables, but also optimize the whole decompression process.

The detection method based on reconstruction error is introduced from the spectral anomaly detection technology, and it is pointed out that the method based on PCA belongs to this method. From the perspective of dimension reduction, it is introduced from the autoencoders, and the advantages are explained: better feature extraction and higher hidden layer can obtain some abstract features.

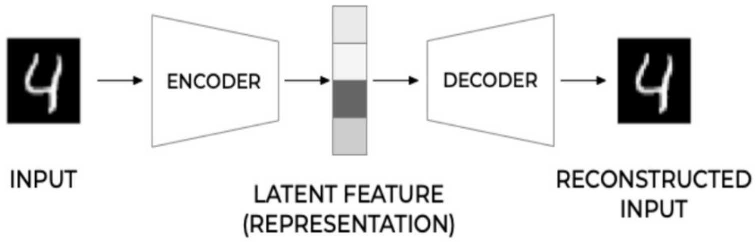

Figure 1. General structure of an autoencoders

2.2 Temporal convolution network

For time series, we can easily think of the following Markov model, where P represents the probability, y represents

$$P\left(y_k \middle| x_k, x_k - 1, \cdots, x_1\right) \tag{1}$$

the output value (label) at time k, and x represents the characteristic value at time k. RNN, LSTM, and other sequence models are designed to process this type of data. In fact, the convolution network can also operate time series data by using the extended causal convolution. TCN is basically a process of expanding causal convolution.

The temporal convolution network has the following advantages.

Firstly, TCN has many ways to change its acceptable field size. For example, stacking more dilated (causal) convolution layers, using a larger dilation factor, or increasing the size of the filter are all feasible options. Therefore, TCNs can better control the memory size of the model and easily adapt to different domains.

Secondly, it has a stable gradient. Different from the cyclic architecture, the anti-propagation path of TCN is different from the time direction of the sequence model. Therefore, TCN avoids the problem of explosion/disappearance gradient, which is a major problem of RNN.

Finally, it has Low memory requirements. Especially when the input sequence is long, LSTMs and GRU can easily occupy a large amount of memory to store some results of their multi-cell gates. However, in TCN, the filter is shared across layers, and the back propagation path only depends on the network depth. Therefore, it is found in practice that gated RNNs may use up to multiple times more memory than TCNs.

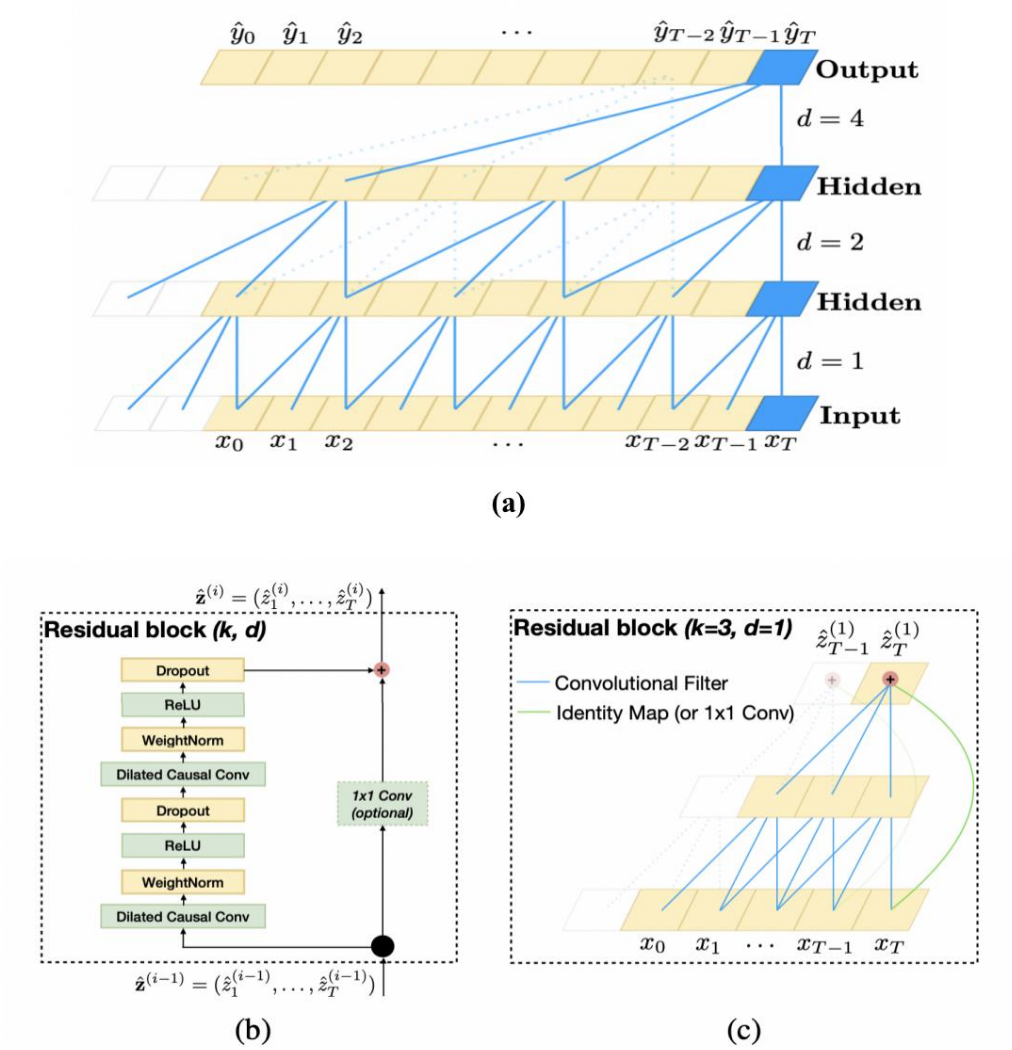

Figure 2. TCN Architectural . (a) All values in the input sequence can be sensed by the receptive field, and the expansion causal convolution, expansion factors d=1, 2, 4, and filter size k=3. (b) TCN residual block. When the remaining input and output dimensions are different, 1 x1 convolution kernel will be added (c) An example of residual connection in a TCN. The filter in the residual function is represented in blue, and the identification is mapped to green. [3]

3. Methodology

3.1 Encoder-Decoder

In most typical encoder and decoder structures, both encoder and decoder are neural networks [5]. The encoder proposed in this paper consists of three modules, each of which is composed of a time convolution network [6]. The decoder consists of two TCN modules, as shown in Figure 3. The encoder can be expressed by the equation (2).

$$h_i = g(X_i) \tag{2}$$

Where h represents the output from the encoder. Its function is to map the feature information into the implicit space, which can theoretically obtain more information in the time dimension. The decoder can be expressed by the equation (3).

$$z_i = f(h_i) = f\left(g(X_i)\right) \tag{3}$$

And the feature dimensions of Z and X are the same. The reconfiguration loss can be defined as equation (4).

$$LOSS = \frac{1}{N}\sum_{i=1}^{N}\left\|X_i - z_i\right\|_2^2 \tag{4}$$

3.2 Abnormal Score

We take the output reconstruction error as the final output, and then normalize it to judge the outliers. The smaller the value, the higher the degree of abnormality. In order to calculate this anomaly score, we use the cosinus similarity from the following the paper [7]， which is shown in equation (5).

$$Anomaly \quad score = 1 - \mathrm{Cos}\, Similarity(x, x) \tag{5}$$

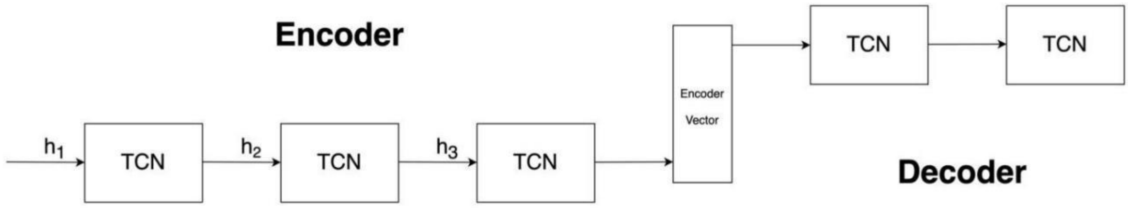

Figure 3. TCN AutoEncoder

4. Experiments

4.1 Experiment platform

The python library used is as follows.

NumPy: It is an open source code base for numerical calculation based on Python. This tool is mainly used for matrix operation. It is much more efficient than the built-in data structure of Python. It supports high-dimensional arrays and multiple matrix operations.

PyTorch: PyTorch is a python version of torch, and PyTorch is a deep learning framework developed by Facebook, which has been open source. Torch is a classic tensor library that operates on multidimensional matrix data, and is widely used in machine learning and other math-intensive applications.

The software and hardware configuration of the experimental environment is shown in the table 1.

Table 1. The software and hardware configuration

CPU	**graphics card**	**Memory**
AMD R9 5900X	3090Ti	32g
operating system	**Python**	**CUDA**
Ubuntu 20.04	3.8	11.3

4.2 Data Sources

We use the KPI exception data of Tsinghua NetMan lab, which is labeled with exception. Because the data set is sorted by time, we use the first 70% as the training set and the last 30% as the test set. And in the experiment, isolated forest, autoencoder based on RNN, autoencoder based on LSTM and autoencoder based on TCN were used to train and learn from data.

4.3 Experimental Design

This paper uses a single dimension of KPI traffic data, which has exception labels and timestamps, and is sorted according to the order of timestamps. The dataset used in this experiment is 3 million, of which outliers account for 2% of the total. And before the experiment, we first preprocess the data set. The preprocessing operations are as follows: (1) Scaling of the time series. (2) Extract the average value, minimum value, maximum value, peak value, offset value and other features.

We first adopt two methods to make the baseline model. The first one is to get a new time series by exponentially smoothing the data, and then use the medium absolute deviation (MAD) to detect anomalies. MAD is a method to detect outliers by calculating the sum of the distances between the observed values and the average values. The second is to use isolated forest to detect data anomalies. The effect of these two models is almost negligible, because the traditional methods consider the distance between data as the core idea, which will have great errors in time series data.

We selected Leaky ReLU, ReLU and Tanh as activation functions. The super parameters of encoder and decoder are selected as Table 2 and Table 3.

Table 2. Encoder parameters

	RNN-AE	**LSTM-AE**	**TCN-AE**
activation	Tanh, Relu	Tanh, Relu	Leaky ReLU, Tanh
hidden_size	300	300	0

Table 3. Decoder parameters

	RNN-AE	**LSTM-AE**	**TCN-AE**
activation	Tanh, Relu	Tanh, Relu	Leaky ReLU, Tanh
hidden_size	300	300	0

In the experimental comparison, we selected the autoencoders based on RNN and the autoencoders based on LSTM for comparison. The encoder part of the RNN based autoencoders is divided into three layers. The first layer is the RNN, the second layer is the full connection layer, the decoder part is symmetrical with the encoder part, and the same is true of the LSTM based autoencoder. The TCN based model is shown in the previous section.

In the training process, we used five-fold cross validation, which refers to dividing the data into five equal parts on average, taking one part of each experiment for testing, and the rest for training. The average value is obtained after five experiments.

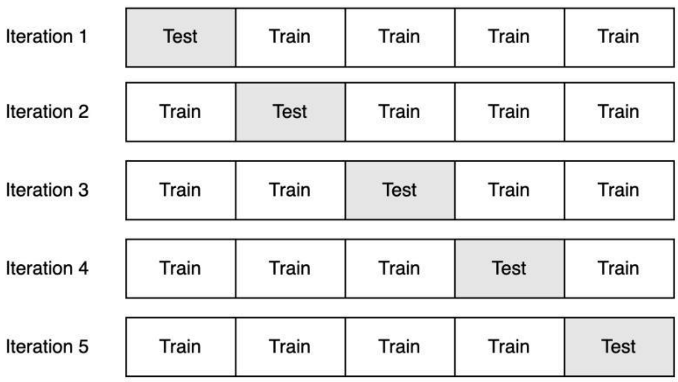

Figure 4. Five-fold cross validation design

Table 4. Comparison test results.

	RNN-AE	**LSTM-AE**	**TCN-AE**
precision	61.22%	67.63%	70.14%
recall	67.98%	73.29%	75.98%
F1	64.4%	70.34%	73%

The results show that TCN based autoencoders has better results than RNN or LSTM, especially in terms of precision, which indicates that recurrent neural network is not necessarily the first choice for processing time series data.

5. Conclusion

The topic of this paper is how to use time data more effectively for anomaly detection. The time series network structure is embedded into the encoder and decoder system to reconstruct the data, and then the reconstructed data is compared with the original data to calculate the reconstruction error, so the abnormal points are identified.

Firstly, this paper introduces some concepts of the temporary revolution network, focusing on the advantages of the temporary revolution network in temporal data. Then the general encoder and decoder architecture is introduced.

In order to make the experimental results more robust, we used five fold cross validation. In addition, we use the distance based on cosine to calculate the abnormal score, so as to avoid the error caused by Euclidean distance. After comparing the model based on the temporary revolution network proposed in this paper with the autoencoders based on RNN and LSTM, it is concluded that the model based on the temporary revolution network can better perform on time series data. In addition, we learned from the process of selecting parameters that the combination of Feature Engineering and super parameters will also have a great impact on the results, so ablation experiments need to be carried out carefully.

Acknowledgements

This work is supported by the Open Foundation of the Guangdong Provincial Key Laboratory of Electronic Information Products Reliability Technology, Climbing Program Special Funds(2021b0675), and Basic and Applied Basic Research Foundation of Guangdong Province (202002030228).

Jiahui Liang contributed equally to this study

References

[1] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
[2] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.
[3] Bai S, Kolter J Z, Koltun V. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling[J]. arXiv preprint arXiv:1803.01271, 2018.
[4] Michelucci U. An Introduction to Autoencoders[J]. arXiv preprint arXiv:2201.03898, 2022.
[5] Bao W, Yue J, Rao Y. A deep learning framework for financial time series using stacked autoencoders and long-short term memory[J]. PloS one, 2017, 12(7): e0180944.
[6] Kieu T, Yang B, Guo C, et al. Outlier Detection for Time Series with Recurrent Autoencoder Ensembles[C]//IJCAI. 2019: 2725-2732.
[7] Habler E, Shabtai A. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages[J]. Computers & Security, 2017, 78(sep.):155-173.