Anomaly detection for time series using temporal convolutional networks and Gaussian mixture model

Jianwei Liu¹, Hongwei Zhu¹, Yongxia Liu², Haobo Wu³, Yunsheng Lan⁴ and Xinyu Zhang⁴

¹ College of Information Science & Electronic Engineering, Zhejiang University, Hangzhou, Zhejiang, China
² College of Electrical Engineering, Zhejiang University, Hangzhou, Zhejiang, China
³ School of Information Science and Engineering, Lanzhou University, Lanzhou, Gansu, China
⁴ Gacia Electrical Appliance Co., Ltd., Wenzhou, Zhejiang, China
Email: ljw0608.mail@gmail.com

Abstract. Anomaly detection, as an important research field in the analysis of time series, has practical and significant applications in many occasions, such as network security, medical health, Internet of Things (IoT), fault diagnosis and so on. However, due to the inherent characteristics of time series, such as tremendous data volumes, the imbalance of normal data and abnormal data, additional constraints and challenges are added for anomaly detection for time series. We present a novel anomaly detection framework, which applies temporal convolutional networks to extract features of time series and combined Gaussian mixture model with Bayesian inference to detect anomalies of systems. In order to evaluate the effectiveness of our approach, experiments are carried out on two typical time series datasets including EEG dataset and current dataset of electrical equipment. The experiments indicate that temporal convolutional network can contribute to extracting salient features of time series and Gaussian mixture model with Bayesian inference has good generalization and reliability for anomaly detection. Meanwhile, the designed architecture and analysis approach of anomaly detection reveal the method’s effectiveness and generalization in the feature extraction and anomaly detection for other time series.

1. Introduction

A time series is a sequence of data points changing over time, which is widely available in real life. Anomaly detection, as an important research field in the analysis of time series, has attracted the attention of many researchers. In order to ensure stable and safe operation of the system, anomaly detection plays a crucial role in many applications, such as network security, medical health, Internet of Things and system fault diagnosis [1].

Anomaly detection aims at detecting abnormal behaviors in time series, then triggering alerts or performing specific actions for abnormal behavior. Anomalies are usually classified into three categories: (1) point anomalies, which are the simplest type of anomalies. A single data point is abnormal relative to others in the sequence regardless of other factors. (2) contextual anomalies, which rely on the context, meaning that other factors can affect whether the value is abnormal or not. (3) collective anomalies, in which a single data point is not abnormal but the set of consecutive values...
deviates from a set of normal consecutive values [2]. As shown in figure 1, (a) is the EEG signal of the epileptic patient transition from seizure free to seizure, and (b) is the current waveform of the electrical equipment transition from normal operation to arc fault. Collective anomalies are usually difficult to detect in real data streams, but they can provide a lot of useful information and can be used as early warning of potential problems in practical applications. This paper will focus on collective anomaly detection.

![Figure 1. Examples of collective anomalies in time series. Blue lines (solid lines) and red lines (dotted lines) respectively denote normal and abnormal time series.](image)

Anomaly detection is an important research topic in the field of data science and machine learning, and has attracted the attention of many researchers. The existing anomaly detection methods can be divided into two categories: supervised and unsupervised methods. The former adopts supervised learning algorithms such as Support Vector Machine (SVM) and Artificial Neural Network (ANN) to anomaly detection [3]. The output of the model is generally defined in two types. The first is the predicted value of the current state. Whether the current state is abnormal depends on the differences between the observed value and the predicted value. The other way of model output is a label that represents whether the current state is abnormal. Despite of the effectiveness of supervised methods, there are still two problems in anomaly detection: (1) class imbalance in datasets, which means that the number of abnormal samples is far lower than that of normal samples in the dataset. (2) Obtaining high-quality labels can be costly.

In contrast, unsupervised methods can be promising for anomaly detection, as they can be modeled using unlabeled training data. Unsupervised anomaly detection methods can be further divided into two categories, namely, probabilistic methods and unsupervised learning methods. Assuming that the normal data follows an underlying distribution (e.g., Gaussian distribution), statistical method can estimate the probability density of the current state. Yamanishi et al. [4] proposed an anomaly detection engine based on statistical theory to score the current state by the probability model using online learning. The higher the score, the more likely it is to be an anomaly. Recently, unsupervised learning methods have achieved remarkable performance in handling large and complicated datasets. Among them, clustering analysis is widely used in anomaly detection. Clustering analysis groups normal observations into one or several clusters, and each cluster has a cluster center. Some observations that do not belong to any cluster or are far away from all of cluster centers are defined as anomaly. Moreover, advanced clustering algorithms are well-designed for anomaly detection, for example, Gaussian mixture model for epilepsy detection, which proves the effectiveness of automatic epilepsy diagnosis [5]. Jacobs et al. [6] make use of the Gaussian mixture model for fault detection and location of gas turbines.

However, the properties of time series add additional constraints and limitations to the machine learning. Firstly, because time series generate quickly, and the amount of data volume is huge, it is impossible to label abnormal data manually and store all of data. Furthermore, anomalies rarely occur, and the time series contains a large number of normal data and few of abnormal data, which results in class imbalance in datasets, and it is difficult to acquire a satisfying anomaly detection model by supervised learning. Therefore, an unsupervised algorithm is needed to detect anomaly for time series.

What is more, collective anomaly detection aims at detecting anomaly sub-sequences in a time series. In general, sub-sequences can be mapped into another feature space in which anomaly detection
is performed. Therefore, feature extraction will be critical in anomaly detection. Traditionally, feature extraction for anomaly detection utilizes signal analysis tools, such as Fourier transform (FFT), Discrete wavelet transform (DWT) or chaos analysis to manually extract features. However, studies have shown that traditional methods of feature extraction are subjective in a certain sense, which limits the expression ability of original sequence [7]. In recent years, deep learning has achieved state of the art in many applications, such as face recognition [8], person re-identification [9], image semantic segmentation [10] and natural language processing [11]. It is well-known that the essence of deep learning is feature representation learning [12], which is a basic modeling tool for acquiring, representing and compressing large-scale signals. Convolutional neural network (CNN) [13] has been used for sequence modeling for several decades. Nowadays, CNN has been widely applied to machine translation [14], audio synthesis [15], and language modeling [16]. Therefore, in order to improve the performance and applicability of anomaly detection, it is necessary to apply the representation learning of CNNs to extract feature of time series.

In this paper, a novel framework is proposed to detect anomaly for time series. Temporal convolutional networks (TCN) are applied to extract feature of time series, mapping high-dimensional time series into low-dimensional feature space, in which anomaly detection are performed. Furthermore, based on the Gaussian mixture model (GMM), the Bayesian inference is introduced to effectively detect anomaly with low computational complexity, and it can be applied to online anomaly detection for time series.

2. Architecture

Considering the particularity of collective anomaly detection in time series, we propose a novel anomaly detection framework, which firstly maps the high-dimensional signals to low-dimensional feature space to extract salient features of time series. Inspired by Van et al. [15], this paper optimizes temporal convolutional network to make it more suitable for feature extraction of time series. Secondly, Gaussian mixture model and Bayesian inference together are used to detect anomalies. The details about anomaly detection framework is as follows.

As shown in figure 2, the basic architecture of the anomaly detection proposed in this paper consists of the following parts:

- **Data Generator**: this module mainly extracts input series from data stream and feed it to the next stage. In the analysis of time series, a time window is commonly used to extract sample by a fixed length (i.e., window width), and the window moves a fixed length (i.e., a stride size) over time. Assuming that the width of the time window is $W$ and the stride size is $S$.

- **Feature Extractor**: The module consists of a well-designed stack of temporal convolutional networks to extract features of time series. The input series $(x_1, x_2, \ldots, x_W)$ passes through the feature extractor to obtain a low-dimensional feature vector $y$.

- **Anomaly Detector**: The module consists of another temporal convolutional network to get the final feature vector $y$.
- **Anomaly Detector**: The module utilizes the Gaussian mixture model to detect whether the current system state is abnormal and output the detection result. GMM can calculates the confidence probability $p$ of current system belonging to the abnormal state, and compares and analyzes with the given confidence threshold $T$ to obtain the detection result.

Therefore, the anomaly detection framework designed in this paper can output an operational alarm signal that can be used in systems such as automatic fault diagnosis or medical diagnosis. Next, we will demonstrate the details about the anomaly detection framework.

3. Temporal convolutional networks

Temporal Convolutional Networks (TCN) based on a 1D convolutional network, is a generic network structure for sequence modeling. The following network structure is an optimized version of structure proposed by Van [15]. This revised network structure improves its performance to extract the salient features of the sequence.

3.1. Casual convolutions

TCN structure is mainly based on two principles: 1) For every layer, the numbers of inputs and outputs are equal; 2) The casual constraints, which means that the output $y_t$ at time $t$ only depends on the previous inputs $x_0, x_1, \ldots, x_t$, but not depends on the future inputs $x_{t+1}, x_{t+2}, \ldots, x_T$. According to the first principle, TCN uses a 1D fully convolutional networks structure. To satisfy the second principle, TCN utilizes casual convolutions, in which the output at time $t$ is convolved with elements of convolutional kernel and the current and previous inputs.

However, this basic network has an obvious drawback. In order to ensure the effectiveness of inputs in the very long history, very deep network or very large convolutional kernel is needed. In fact, experiments have revealed that neither of the methods is feasible. Therefore, modern convolutional network is supposed to be elaborated to optimize the performance of TCN.

![Figure 3. Architectural elements in a dilated-causal convolution network.](image)

![Figure 4. Architecture elements of TCN.](image)

3.2. Dilated convolutions

As is stated above, receptive field of TCN is limited if only casual convolutions are included, leading to the fact that output at time $t$ can't receive the inputs in the long history. Thus, dilated convolutions are necessary to be introduced to TCN to improve its receptive field without extra computation cost. For input sequence $x = (x_0, x_1, \ldots, x_t)$ and $k$-sized convolutional kernel $f$, the $s$-th output of dilated convolutions is shown in equation (1).
where $d$ denotes dilation rate. Obviously, dilated convolution degrades to ordinary convolution when $d=1$. And larger dilation rate means larger input range of top layer in the network, thus increasing the receptive field.

Figure 3 illustrates the architecture of dilated-causal convolutions with dilation rate $d=1, 2, 3, 4$. It can be seen that receptive field are enlarged layer by layer, as the dilation rate increase exponentially with the linear increase of network depth.

3.3. Residual connections

It can be seen from figure 3 that increasing the network depth or extending its width can improve the performance of network. However, with the increasing of network depth, it is challenging to train the network, and the performance of network begins to deteriorating gradually. Thus, residual networks can be applied to simplify the training progress. This method makes the optimization of deep network easier and network deeper, improving the network performance apparently.

Receptive field of TCN structure depends on the network depth $n$ and the size $k$ of convolution kernel, so the stabilization of network is very significant. As is illustrated in figure 4, residual connections are included in the dilated-causal convolutional network. Besides, dropout layer and layer normalization are added following the dilated-causal convolutional layer to improve the generalization performance of network.

To sum up, TCN structure has been optimized to adapt to time series. The network proposed in this paper contains input layer, three TCN blocks, a fully-connected layer and a softmax layer. After fully-connected layer, L2 normalization are added as feature embedding layer. We train the TCN model in the form of classification to improve the ability of feature representation.

3.4. Squeeze-and-excitation block

Convolutional network, based on the convolution operation, extracts features through integrating spatial information and channel information locally. Fortunately, the dependence between channels can be modeled explicitly by a new network called squeeze-excitation (SE) block, which can improve the feature extraction ability of TCN. SE block is shown in figure 5, $F_{tr}$ is traditional convolutional operation, while $X$ and $U$ are its input and output respectively. In the SE block, global average pooling follows the input $U$, outputting $1 \times C$ data, which is a procedure called squeeze ($F_{sq}$). Then, the data go through two fully-connected layers, which is a procedure called excitation ($F_{ex}$). The mechanism of SE block automatically obtains the significance of each channel, enhancing the major features and weakening the minor features to improve the discrimination of extracted feature.

3.5. Loss function

3.5.1. Focal loss. Due to the class imbalance in datasets, trained TCN model by cross-entropy loss function will be more inclined to the class with more samples, which results in an unsatisfying model.
To solve this problem, focal loss revises cross entropy loss function by introducing adjustment factor $((1-p))$ and balance factor $\alpha$ to improve the classification performance of the model.

$$p_i = \begin{cases} p, & \text{if } y = 1 \\ 1-p, & \text{otherwise} \end{cases}$$  \hspace{1cm} (2)

$$FL(p_i) = -\alpha((1-p))\log(p_i)$$  \hspace{1cm} (3)

where $p$ denotes the possibility that the sample $x$ belonging to the positive class, and $\alpha$ is focal parameter.

3.5.2. Center loss. The features extracted by the trained TCN networks with focal loss are separated but not discriminated. To solve this problem, center loss function is supposed to be added to train the TCN network. The basic mechanism of center loss function is that increasing variance among different classes and minimizing variance within the same class can extract discriminated features. The center loss function is shown in equation (4).

$$L_c(x) = \frac{1}{2} \sum_{i=1}^{n} \|x_i - c_i\|^2$$  \hspace{1cm} (4)

where $x$ denotes input feature vector and $c_i$ represents the center of $x$ corresponding to class $i$.

For time series classification, there are only two centers representing positive and negative samples respectively. During the training process, center loss and focal loss are supposed to be combined, and the centers $c_i$ are updated in each iteration. Focal loss can increase the feature variance between different classes while center loss can decrease the feature variance within the same class. Thus, the training method combines focal loss and center loss can help obtain the more discriminated features.

4. Gaussian mixture model for anomaly detection

The Gaussian mixture model (GMM) is a weighted linear combination of multiple Gaussian components, which is often used to solve the problem that data contains multiple different distributions. GMM can smoothly approximate the density distribution of arbitrary shapes and can be applied to complicated object modeling.

Let $x \in \mathbb{D}^D$ be a $D$-dimension sample from time series, its probability density function can be formulated as following:

$$\Pr(x) = \sum_{k=1}^{K} \omega_k \psi(x; \mu_k, \Sigma_k)$$  \hspace{1cm} (5)

$$\psi(x; \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{D/2} |\Sigma_k|^{1/2}} \exp \left[ -\frac{1}{2} (x - \mu_k)^T \Sigma_k^{-1} (x - \mu_k) \right]$$  \hspace{1cm} (6)

where $\omega_k$ is the weight of the $k$-th Gaussian component, $\sum_{k=1}^{K} \omega_k = 1$, $0 \leq \omega_k \leq 1$. $\mu_k \in \mathbb{D}^D$, $\Sigma_k \in \mathbb{D}^{D \times D}$ respectively represents the mean vector and covariance matrix of the $k$-th Gaussian component. We can utilize expectation maximum (EM) algorithm to estimate the parameters $\theta = (\omega, \mu, \Sigma)$ of GMM.

Based on GMM, the Bayesian inference probability (BIP) index \cite{17} is introduced to detect anomaly. The BIP is formulated as follows:

$$BIP = \sum_{k=1}^{K} P(C_k | x_j) \cdot P_k^{(k)}(x_j)$$  \hspace{1cm} (7)

$$P_k^{(k)}(x_j) = \Pr \{ D(x, C_k) | x \in C_k \leq D(x_j, C_k) | x_j \in C_k \}$$  \hspace{1cm} (8)

$$D((x_j, C_k) | x_j \in C_k) = (x_j - \mu_k)^T \Sigma_k^{-1} (x_j - \mu_k)$$  \hspace{1cm} (9)
where \( p_{ki}(x) \) indicates the local Mahalanobis distance-based probability index of testing sample \( x \).

\[ D(x_j, C_k) \]

is the Mahalanobis distance of testing sample \( x_j \) to the center of \( k \)-th Gaussian component.

Given the threshold \( T \), the BIP index is calculated by equation (7) and compared with the given the threshold \( T \) to determine whether the current state is abnormal.

5. Experiment
In this section, experiments are carried out on two typical time series datasets to evaluate the anomaly detection for time series proposed in this paper, and the experimental results are analysed.

5.1. Performance metrics
In order to make the experiment results readable, some objective evaluation metrics are introduced to evaluate the model performance. Formally, \( TP \), \( FP \), \( TN \) and \( FN \) denote respectively the number of normal time series correctly detected as normal (True Positives), the number of anomalies wrongly detected as normal (False Positives), the number of anomalies correctly detected as abnormal (True Negatives), and the number of normal time series wrongly detected as abnormal (False Negatives).

Accuracy, recall, FPR (False Positive Rate) and F1-score are defined as following:

\[
\text{Accuracy} = \frac{TP + TN}{TP + FP + TN + FN}
\]

\[
\text{recall} = \frac{TP}{TP + FN}
\]

\[
FPR = \frac{FP}{TN + FP}
\]

\[
F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

where \( \text{Precision} = \frac{TP}{(TP + FP)} \).

5.2. Implement details
The samples of datasets are collected from time series by using a sliding window, and the dataset is divided into training dataset and testing dataset by the ratio of 1:4. Then we use focal loss and center loss to conduct supervised training for TCN. During the process of training, a simple cross-validation method is used to verify the performance of the model.

Then the trained TCN model was applied to extract the features of normal time series. Determined the number of Gaussian components \( K \), the GMM parameters \( \theta = (\omega, \mu, \Sigma) \) were iteratively optimized by EM algorithm. Given a testing sample, the corresponding BIP index was calculated by equation (7). If the BIP index exceeds the given confidence threshold \( T \), the observation is considered to be abnormal, otherwise it is marked as a normal sample.

5.3. Results and analysis
This section conducts experiments on the available EEG datasets and private current datasets of electrical equipment to analyse and evaluate the performance of the anomaly detection techniques proposed in this paper.

5.3.1. Epilepsy signal detection. This experiment carried out on the available EEG dataset, whose detailed information can be found in [18]. The EEG dataset contains five subsets, each of which contains 100 fragments from single channel EEG signal. Subsets A and B were acquired from patients with open and closed eyes, and subsets C and D were collected from patients during epilepsy free intervals. Subset E was collected from patience during epilepsy. The length of all EEG signal fragments was 4096, and the sampling rate was 173.61Hz. As shown in figure 6 (a), EEG data without
epilepsy were taken as normal samples while the EEG data during epilepsy as abnormal samples for experiments.

In this paper, the sliding window size $W$ of EEG data was set to 150. The TCN model was trained on the EEG dataset, and the trained TCN was used as feature extractor. As shown in figure 7 (a), 1000 samples were mapped into 2D feature space, it can be seen that the abnormal and the normal samples are respectively clustered in different spatial regions. This distribution characteristic reflects that the TCN model can help to extract salient and discriminated features of the time series.

![Figure 6. Normal and abnormal samples of time series.](image)

![Figure 7. The distribution map of high-dimension time series mapping into 2D feature space.](image)
We extracted the features of normal EEG data by trained TCN model, and optimized Gaussian mixture model by EM algorithm. As is shown in the figure 8 (a), the number of Gauss components $K = 4$, they were marked as G1, G2, G3 and G4 respectively, and each ellipse represented a Gaussian component.

The 10-fold cross validation was utilized to evaluated the performance of the GMM model on testing dataset, and the evaluation results of GMM on EEG dataset was shown in table 1. Specifically, when the FPR was 1‰, the recall could reach 93.52%.

| Datasets     | Accuracy | recall  | FPR   | F1     |
|--------------|----------|---------|-------|--------|
| EEG          | 99.45%   | 99.67%  | 0.796%| 99.54% |
| Arc Fault    | 97.10%   | 97.096% | 0.498%| 95.66% |

5.3.2. Arc Fault detection. The current dataset was collected from the AC current signals of the electrical equipment. In the experiment, we collected the current signal of the air conditioner during normal operation and the arc fault at a sampling rate of 96 Hz. As shown in figure 6 (b), the current signals of the electrical equipment working normally were taken as normal samples, while the current signals during arc fault were taken as abnormal samples. The size $W$ of sliding window 96, and we trained the TCN model on the current dataset, and extracted the features of current series. As shown in figure 7(b), the high-dimensional time series were mapped into the 2D feature space. Obviously, the normal and abnormal samples were respectively clustered in different spatial regions, which further showed that the TCN model can help to extract the salient and discriminated features of the time series.

Similarly, the feature vectors of normal time series were extracted using TCN model, and optimized Gaussian mixture model by EM algorithm. As shown in figure 8 (b), the number of Gaussian components $K = 4$, they were marked as G1, G2, G3 and G4 respectively, and each ellipse represented a Gaussian component. Table 1 also showed the evaluation results of GMM on arc fault datasets.

To summarize, we can draw the following conclusions through the analysis of the experiments: (1) the TCN model can help to extract the salient and discriminated features of the time series. (2) In the case of class imbalance in datasets, GMM with Bayesian inference for anomaly detection can obtain high recall rate while keeping a low FPR, and it can be applied to detect anomaly for time series.
6. Conclusion
This paper proposed an online anomaly detection framework for time series based on temporal convolution networks and Gaussian mixture model, which can map high-dimensional time series into low-dimensional feature space for anomaly detection. For the sequence modeling, temporal convolutional networks were proposed for feature extraction of time series. Based on the GMM, Bayesian inference was introduced in this paper to obtain better performance in anomaly detection.

Experiments were conducted to train and evaluate the TCN and GMM on EEG dataset and arc fault dataset. The experiment results indicate the optimized temporal convolutional network can help to extract the salient and discriminated features of the time series, and the Gaussian mixture model has good generalization and reliability in anomaly detection. At the same time, the design architecture and analysis method of anomaly detection proposed in this paper reveal its effectiveness and generalization in feature extraction and anomaly detection of other time series.

References
[1] Chandola V, Banerjee A and Kumar V 2009 CSUR 41 15
[2] Saarinen I 2017 Adaptive real-time anomaly detection for multi-dimensional streaming data (Espoo: Aalto University)
[3] Batta P, Singh M, Li Z, Ding Q and Trajkovic L 2018 Proc. Of the IEEE Int. Symposium on Circuits and Systems (Florence) pp 1-4
[4] Yamanishi K, Takeuchi J I, Williams G and Milne P 2004 Data Mining and Knowledge Discovery 8 275-300
[5] Li Y, Cui W, Luo M, Li K and Wang L 2018 International Journal of Neural Systems 28 7
[6] Jacobs W R, Edwards H, Li P, Kadirkamanathan V and Mills A R 2018 International Journal of Prognostics and Health Management 9 26
[7] Wang Y, Zhang F and Zhang S 2018 IEEE Transactions on Instrumentation and Measurement 67 2526-37
[8] Schroff F, Kalenichenko D and Philbin J 2015 Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (Boston) pp 815-823
[9] Chen S Z, Guo C C and Lai J H 2016 IEEE Transactions on Image Processing 25 2353-67
[10] Girshick R, Donahue J, Darrell T and Malik J 2014 Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition (Columbus) pp 580-587
[11] Zhang Y, Lease M, Wallace B C 2017 31st AAAI Conference on Artificial Intelligence (San Francisco) pp 3386-3392
[12] Bengio Y, Courville A and Vincent P 2013 IEEE Transactions on Pattern Analysis and Machine Intelligence 35 1798-1828
[13] LeCun Y, Boser B, Denker J S and Henderson D 1989 Neural Computation 1 541-551
[14] Kalchbrenner N, Espeholt L, Simonyan K, Oord A V D, Graves A and Kavukcuoglu K 2016 Neural machine translation in linear time preprint arXiv/1610.10099
[15] Van Den Oord A, Dieleman S, Zen H, Simonyan, K, Vinyals O, Graves A, Kalchbrenner N, Senior A and Kavukcuoglu K 2016 Wavenet: A generative model for raw audio preprint arXiv/1609.03499
[16] Dauphin Y N, Fan A, Auli M and Grangier D 2016 Language modeling with gated convolutional networks preprint arXiv/1612.08083
[17] Yu J and Qin S J 2008 AIChE Journal 54 1811-29
[18] Andrzejak R G, Lehnertz K, Mormann F, Rieke C, David P and Elger C E 2001 Phys. Rev. E 64 061907