applied sciences

MDPI

Article

Time Series Anomaly Detection for KPIs Based on Correlation Analysis and HMM

Zijing Shang [1], Yingjun Zhang [2,*], Xiuguo Zhang [1,*], Yun Zhao [1], Zhiying Cao [1] and Xuejie Wang [1]

[1] School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China; shangzj@dlmu.edu.cn (Z.S.); zhao_yun@dlmu.edu.cn (Y.Z.); czysophy@dlmu.edu.cn (Z.C.); wxj@dlmu.edu.cn (X.W.)

[2] Navigation College, Dalian Maritime University, Dalian 116026, China

* Correspondence: zhangyj@dlmu.edu.cn (Y.Z.); zhangxg@dlmu.edu.cn (X.Z.)

Abstract: KPIs (Key Performance Indicators) in distributed systems may involve a variety of anomalies, which will lead to system failure and huge losses. Detecting KPI anomalies in the system is very important. This paper presents a time series anomaly detection method based on correlation analysis and HMM. Correlation analysis is used to obtain the correlation between abnormal KPIs in the system, thereby reducing the false alarm rate of anomaly detection. The HMM (Hidden Markov Model) is used for anomaly detection by finding the close relationship between abnormal KPIs. In our correlation analysis of abnormal KPIs, firstly, the time series prediction model (1D-CNN-TCN) is proposed. The residual sequence is obtained by calculating the residual between the predicted value and the actual value. The residual sequence can highlight the abnormal segment in each data point and improve the accuracy of anomaly screening. According to the obtained residual sequence, these abnormal KPIs are preliminarily screened out from the historical data. Next, KPI correlation analysis is performed, and the correlation score is obtained by adding a sliding window onto the obtained anomaly index residual sequence. The correlation analysis based on the residual sequence can eliminate the interference of the original data fluctuation itself. Then, a correlation matrix of abnormal KPIs is constructed using the obtained correlation scores. In anomaly detection, the constructed correlation matrix is processed to obtain the adaptive parameters of the HMM model, and the trained HMM is used to quickly discover the abnormal KPI that may cause a KPI anomaly. Experiments on public data sets show that the method obtains good results.

Keywords: convolutional neural network (CNN); temporal convolutional network (TCN); anomaly detection of KPIs; hidden Markov model (HMM); correlation analysis

Citation: Shang, Z.; Zhang, Y.; Zhang, X.; Zhao, Y.; Cao, Z.; Wang, X. Time Series Anomaly Detection for KPIs Based on Correlation Analysis and HMM. *Appl. Sci.* **2021**, *11*, 11353. https://doi.org/10.3390/app112311353

Academic Editor: Markus Goldstein

Received: 9 October 2021
Accepted: 29 November 2021
Published: 30 November 2021

1. Introduction

KPI (Key Performance Indicator) anomaly detection is a low-level core technology in intelligent operation and maintenance. It is mainly aimed at current events. By analyzing the KPI curve, the abnormal behaviors of KPIs (sudden increase, sudden drop, and jitter) imply that some potential faults have occurred in related applications, such as increased access latency, network failure, or sharp decreases in access users [1]. Due to the huge complexity of the system, KPIs of the monitoring system are numerous and various. When the system fails, the efficiency of manually searching for abnormal KPIs is extremely low, and consumes a lot of manpower and material resources. The manual analysis of system failure will also cause many misjudgments and produce certain economic losses.

In the field of intelligent operation and maintenance, KPI anomaly detection for a multi-index system is difficult. In KPIs of the system, there may be a temporal correlation between two indicators; that is, when a KPI is abnormal, it will cause similar fluctuations in the trends of other KPIs in a short time. Due to the presence of numerous monitoring indicators and complex structures in the distributed system, fluctuations will continue to spread to more KPIs. This causes more KPI anomalies in the entire system, making

anomaly detection and root cause analysis very complicated. The correlation analysis can give rise to multiple similar abnormal KPIs. For example, a CPU utilization rate increase in the system will cause a class of CPU-related KPIs (such as CPU utilization rate of a single machine, CPU idle state time, overall CPU load, etc.) to produce anomalies. Without using correlation analysis, it is very difficult for operation and maintenance personnel to analyze and infer the correlation between large numbers of complex KPI anomalies. When faults occur in large-scale distributed systems, many redundant alerts will arise that are relevant but distract operation and maintenance personnel. Correlation analysis can narrow the scope of the analysis of important abnormal KPIs and preclude the interference of redundant alerts; other KPIs unrelated to the fault can also be excluded. At the same time, correlation analysis can reduce the misstatement of abnormal KPIs after anomaly detection, and improve the accuracy of anomaly detection results. At present, most anomaly detection methods based on correlation analysis use baseline methods such as Pearson or Granger to calculate raw KPIs. However, these baseline methods can only obtain the correlation value. They do not take into account important factors such as the shift value between KPIs along the time axis and the order of influence, which often causes the misjudgment of abnormal KPIs, reducing the accuracy of the correlation analysis. Meanwhile, the volatility of raw KPIs will affect correlation analysis results at abnormal points between indicators [2].

At present, most of the time series anomaly detection methods for KPIs are concentrated in traditional machine learning and deep learning. The processing speed and accuracy of traditional machine learning and deep learning for KPI time series data are ideal. Some anomaly detection methods based on supervised learning [3,4] can perform fast and accurate anomaly detection by relying on a large number of types of anomaly-labeled data, but they are not suitable for the actual operation and maintenance environment, which contains fewer anomalies. The new unsupervised and semi-supervised learning anomaly detection methods [5–8] can better adapt to the actual operation and maintenance environment. Some models use RNN and LSTM to analyze time series data, but RNN and LSTM have problems, such as error accumulation and the need for a lot of training memory, which leads to some false positives and false negatives in anomaly detection.

Most existing anomaly detection methods are for large-scale anomaly detection in all the KPI timing data in the system. However, it is difficult to analyze all abnormal data and solve faults in a short time during intelligent operation and maintenance. In the actual maintenance process, it is necessary to find other KPIs that cause a KPI anomaly, so as to reduce the time of investigation and improve the processing efficiency. Therefore, it is necessary to quickly find the class of abnormal KPIs so as to repair and eliminate the corresponding fault. The Hidden Markov Model (HMM) [9] can capture the transition relationship between time-dependent data, and infer corresponding hidden states by establishing the transition relationship between observation states and hidden states, which fulfills the requirements of the system maintenance process.

There are some problems in the current KPI anomaly detection methods:

(1) Most of the current anomaly detection methods ignore the correlation between multiple indicators. Without considering this correlation, false negatives may arise. In addition, the factors considered by the correlation analysis methods are not comprehensive enough, and the effect of the volatility of the original data on the correlation analysis of abnormal trends is also ignored.
(2) Some time series-based anomaly detection methods are processed by RNN, LSTM, and other models, but these models themselves have the problem of error accumulation and require a lot of training memory.
(3) Most anomaly detection methods only focus on the detection of indicators with anomalies, and ignore the influence relationship between multiple indicators. A large number of abnormal indicators will make the operation and maintenance personnel's troubleshooting efficiency low.

To solve these problems, this paper first proposes a time series prediction model, 1D-CNN-TCN (1D Convolutional Neural Network and Temporal Convolutional Network),

to predict and obtain a residual sequence of KPIs, as well as to screen abnormal KPIs in a certain period. Then, a time series anomaly detection method for KPIs based on correlation analysis and HMM is proposed to detect exceptional KPIs and find other KPIs that may cause a KPI exception. The contributions of this paper are summarized as follows:

This paper considers such problems as the lack of consideration for the volatility of original KPI data interfering with correlation and correlation factors. In the KPI anomaly detection method, correlation analysis based on a KPI residual sequence is included. The residual sequence can highlight an abnormal section of the time series data, and exclude the influence of the original fluctuation trend of the time series data on the abnormal fluctuation. At the same time, correlation analysis of KPIs is carried out by comprehensively considering multiple factors, such as correlation value, shift value along the time axis, and the order of influence. Through the above methods, the accuracy of the correlation analysis is improved;

(1) This paper considers that RNN, LSTM and other methods have error accumulation problems and require a large amount of training memory. 1D CNN (One-Dimensional Convolutional Neural Network) and TCN (Temporal Convolutional Network) are combined, and a time series prediction model, 1D-CNN-TCN, is proposed to predict KPIs. The structural characteristics of 1D CNN and TCN make it possible to avoid the gradient explosion and error accumulation caused by insufficient memory during model training;
(2) In this paper, correlation analysis and HMM are combined to realize anomaly detection in KPI time series. Using the correlation matrix constructed by correlation analysis as the parameters, HMM can adaptively adjust the initial parameters and quickly infer abnormal KPIs that affect specific KPIs.

The rest of this paper is organized as follows: Section 2 introduces the background knowledge of KPI anomaly detection and the groundwork of correlation analysis. Section 3 elaborates the time series anomaly detection framework based on correlation analysis and HMM. Section 4 evaluates the performance of the proposed framework and compares it with related methods. Section 5 finally concludes the paper.

2. Related Work

2.1. KPI Time Series Anomaly Detection Method

With the development of artificial intelligence technology, some artificial intelligence applications, such as intelligent operation and maintenance [1–3,5] and intelligent transportation [10], have attracted researchers' attention. Time series anomaly detection for KPIs is a research hotspot in the field of intelligent operation and maintenance. At present, the most popular methods of time series anomaly detection for KPIs are mainly based on traditional statistics and machine learning models. The earliest KPI time series data anomaly detection methods are based on traditional statistics, and the main applications are anomaly detection based on ARIMA [11] and anomaly detection based on a Gaussian Mixture Model [12]. However, the statistical method is based on abnormal detection with certain assumptions. Although it has a degree of robustness, it is too dependent on assumptions. With the continuous development of machine learning, the KPI time series data anomaly detection method based on machine learning has increasing applicability. For example, EGADS [3] proposed by Yahoo and Opprentice [4] proposed by Tsinghua Laboratory both require a large number of anomaly label training models to achieve anomaly detection in multiple types of KPIs, but also require a large number of labels that are in line with the actual operation and maintenance situation. Li et al. [13] proposed the ROCKA method, which is based on a clustering algorithm. Although this method does not rely on a large number of labels for training, it has high computational complexity, and is too dependent on the model.

With the continuous development of deep learning, many experts and scholars choose to use deep learning models such as RNN (Recurrent Neural Networks) and LSTM (Long Short-Term Memory) for anomaly detection [14,15]. For example, Hundman et al. [16] used

LSTM for anomaly detection in spatial telemetry time series data. LSTM alleviates the problem of error accumulation found in RNN, but LSTM itself needs a lot of training memory in the case of long input sequences [17]. To solve these problems, some methods [18,19] use a Convolutional Neural Network (CNN) to replace, or work in combination with, RNN, LSTM, and other models to analyze time series data. For example, Ren et al. [19] used SR and CNN for anomaly detection in time series data, but CNN could not extract the dependence between the time series. Then, Bai et al. [17] proposed a Temporal Convolutional Network (TCN) model for processing temporal data, which reduces the error accumulation via extracting temporal features. Li et al. [20] used CPA-TCN to detect anomalies in time series data, which overcame the gradient explosion caused by the presence of insufficient memory in deep learning models such as RNN, and obtained dependence between time series.

2.2. Anomaly Detection of KPIs Based on Correlation Analysis

Time series anomaly detection of KPIs based on correlation analysis can solve the problem of multi-index anomaly detection. However, at present, time series anomaly detection of KPIs based on correlation analysis ignores problems such as the volatility of KPIs itself, and the presence of insufficient factors of correlation analysis. Usually, KPIs of the same category are more correlated than those of different categories, and the trend in the original data will cause misjudgment of the final results [21]. Correlation between abnormal and normal KPIs of the same category is different due to the abnormal state, and universal correlation analysis methods cannot completely solve this problem. For example, Jiang et al. [11] improved the Pearson method, and integrated it into anomaly detection. Kao et al. [22] used Pearson correlation analysis to classify KPIs, and here, different models were used for anomaly detection in different kinds of KPIs. In one method, more models are used, but the influence of the fluctuation trend in the original data over the correlation analysis results is not considered. The partial correlation analysis method [23] analyzes the correlation between KPIs based only on correlation value, without considering the shift value, the order of influence, and other factors. To accurately distinguish the fluctuations in abnormal KPIs from normal fluctuations with different structures, Su et al. [2] proposed the Coflux method, which can determine whether two KPIs are related to abnormal fluctuations, and identify the relevant shift value and order of influence. However, due to the large amount of data required and the complex calculation methods, this method takes a lot of time.

At present, the anomaly detection method based on correlation analysis does not perform any processing after detecting all abnormal KPIs. However, for intelligent operation and maintenance personnel, it is still necessary to identify the causes of faults from a large number of abnormal indicators, and the efficiency is not high. The Hidden Markov Model can derive observation data from known data by analyzing the time transfer relationship. Some experts and scholars combine temporal data correlation analysis with HMM [24,25]. Correlation analysis results can be used as HMM parameters to adaptively adjust the HMM, which helps avoid the limitations and low accuracy of the inference results.

To solve the above problems, this paper proposes an anomaly detection method based on correlation analysis and HMM. The correlation analysis method in this paper can be used to accurately detect anomalies between multiple KPIs in a short time, and HMM can be used to infer the abnormal KPIs that affect a specific KPI from the abnormal KPIs caused by a certain fault. This method can improve the accuracy of anomaly detection.

3. Method

This paper presents an anomaly detection method for KPI time series based on correlation analysis and HMM. A time series prediction model, 1D-CNN-TCN, is proposed to obtain residual sequences of KPI time series. Residual sequences can highlight anomalies to improve the accuracy of anomaly screening and reduce the influence of the volatility in the original KPI data. The correlation of abnormal KPIs is analyzed, and this generates

a clustering matrix to realize alarm clustering. This correlation matrix is then treated as an HMM parameter, and HMM is used for KPI anomaly detection.

3.1. Method Flow

This paper presents an anomaly recognition method. The overall framework of the method is shown in Figure 1, which mainly includes three modules: abnormal KPI screening, alarm clustering, and abnormal detection. Firstly, we propose a 1D-CNN-TCN prediction model. The model is used to learn the characteristics of past data to predict future trends. The difference between a predicted sequence and an actual observation sequence at the corresponding time point is calculated to obtain the residual sequence. The 3-sigma method [26] is used to identify all KPI anomalies at a known time. Based on the residual sequences, a sliding window is added to the correlation analysis of abnormal KPIs to obtain correlation values, shift values, and the order of influences. Through comprehensive analysis of these three factors, final correlations are obtained, and the correlation matrix is constructed to further realize alarm clustering. All parameters of the HMM are determined, including the correlation matrix that has been converted to be suitable as an HMM parameter, and the HMM is trained. Then, other KPIs are found that may cause a certain KPI anomaly using the trained HMM model.

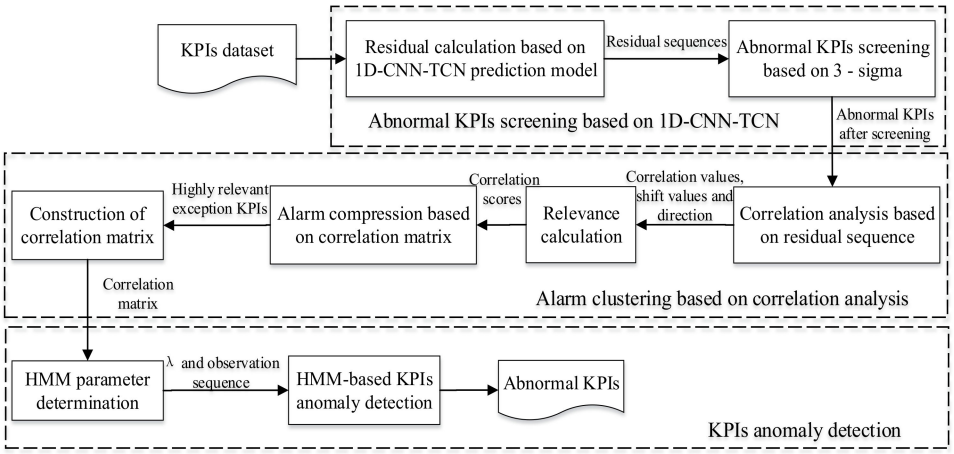


Figure 1. Method flow chart.

3.2. 1D-CNN-TCN Time Series Prediction Model

The Convolutional Neural Network (CNN) was first proposed by LeCun et al. [27] and is currently mainly used in the fields of image recognition and natural language processing. 1D CNN is suitable for time series with a one-dimensional structure and has characteristics of easy parallel, and does not induce gradient explosion or error accumulation. However, its ability to capture temporal dependency relations between time series data points is poor. After the continuous optimization and development of a CNN, the Time-Series Convolutional Neural Network (TCN) has appeared. The TCN mainly deals with time-series data, and uses dilated convolution [28] and residual connection [29] to increase the receptive field of a network and accelerate training speed. The general network structure of the TCN algorithm is shown in Figure 2. The TCN can make up for the 1D CNN's inability to capture temporal dependence, and also has other characteristics of the 1D CNN. The use of expanded convolution not only ensures that the network covers all effective information, but it also enables the deep network to obtain more effective information. The formula of expanded convolution is:

$$F(s) = (x *_d f)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \quad (1)$$

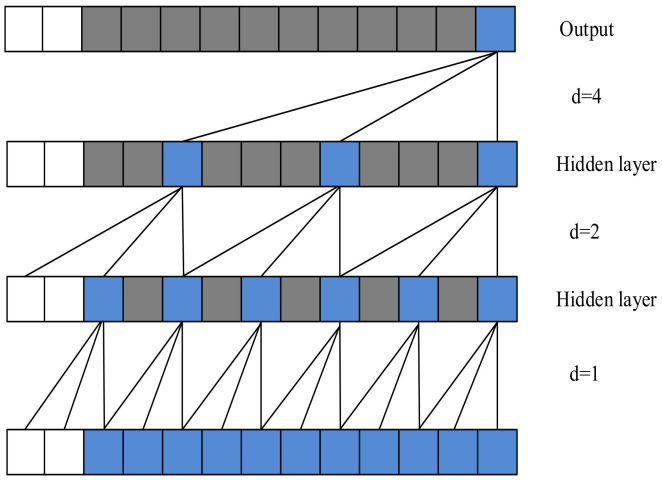


Figure 2. Basic TCN network structure.

Here, d represents expansion coefficient, and k represents convolution kernel size. When d is 1, the extended convolution degenerates into ordinary convolution. By controlling the size of d, the receptive field is broadened without changing the amount of calculation.

This paper combines 1D CNN with TCN. The first layer of the 1D-CNN-TCN model uses 1D CNN to perform convolution on local regions of the input data to obtain its local features. There are 128 convolution kernels in 1D CNN, and the convolution kernel size is 2. The ReLU activation function is used to realize nonlinear transformation such that the model is more convergent. After extracting local features with the 1D CNN, the data dimension will expand. Therefore, in the second layer, a MaxPooling layer is used to downsample the obtained feature sequence, so as to reduce the data dimension of the feature sequence. The pooling window size is 2. Based on the extracted local features, the third layer uses TCN to obtain temporal features, wherein the TCN has 64 convolution kernels, the convolution kernel size is 20, the activation function is LeakyReLU, and the dropout rate is 0.5. Finally, the final prediction time series is output through a full connection layer Dense. The specific network structure is shown in Figure 3. Compared with RNN and LSTM, 1D-CNN-TCN has a better prediction effect and calculation efficiency.

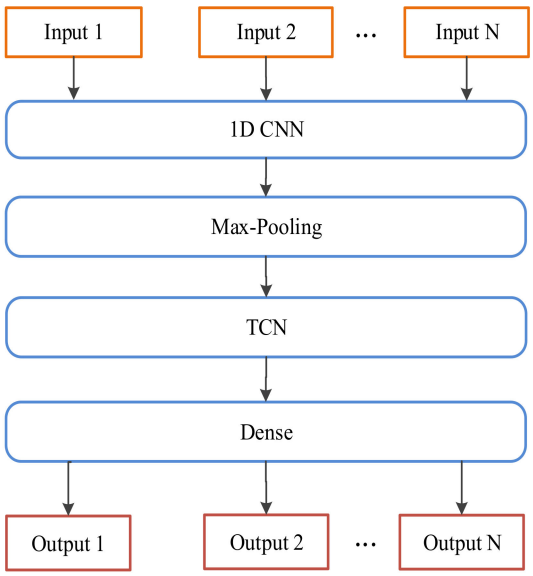


Figure 3. 1D-CNN-TCN network structure.

The specific steps are as follows:

(1) Firstly, time series data are processed in a sliding window, sliding one data point forward each time to predict the value of the next moment;
(2) Preprocessed time series are input into the 1D CNN, and the convolution kernel performs the convolution operation on the local region of the input sequence with a certain step length to extract the local features of the time series. Then, the Relu activation function is used to realize the nonlinear transformation, which can make the model converge better and assume a sparse representation so as to prevent the gradient from disappearing;
(3) After the convolution operation, the number of feature sequences extracted increases, resulting in the expansion of the data dimension and the increasing of the computational complexity. Through the pooling operation, the feature sequence is downsampled to reduce the dimension of the feature sequence data;
(4) Based on local features extracted by the 1D CNN, TCN is used to further extract time features;
(5) Finally, the time series predicted via the full connection layer Dense is output.

3.3. Abnormal KPI Screening Based on 1D-CNN-TCN Model

3.3.1. Data Preprocessing

It is necessary to preprocess data before using a model for prediction. The original data obtained via system monitoring will appear lacking, and will contain too much noise. Interference factors in these original data will affect the training of the following model, and thus ultimately affect the accuracy of the output results. Therefore, before model training, this paper first conducts a simple preprocessing of the original data. In this paper, the linear interpolation method is used to fill in the missing values, which reduces errors and their impact on the ensuing model training. After filling in missing values, the Z-score method is used to standardize the processed data. All data are scaled to remove the unit limitation, and to convert them into dimensionless pure values, which can improve the reliability of the results.

3.3.2. Acquisition of Residual Sequence

In this paper, the 1D-CNN-TCN time series prediction model in Section 3.2 is used to predict KPIs. The predicted value is compared with the actual observation data of the same period at the corresponding time point, and residual sequences are obtained. The formula is shown as Equation (2). When an original sequence is found to be abnormal at a given time, the corresponding position in the residual sequence will produce large fluctuations, different from the normal trend of the time series data. The obtained residual sequence can increase the fluctuation trend of the anomaly, reduce the influence of the original trend in the KPI itself, facilitate correlation calculation and anomaly detection, and make the results more accurate.

Suppose sequence $X = (x_1, x_2, \ldots, x_l)$ is the predicted time series, and $Y = (y_1, y_2, \ldots, y_l)$ is the real time series. The calculation formula for each element z_i in the residual sequence $Z = (z_1, z_2, \ldots, z_l)$ is as follows:

$$z_i = y_i - x_i, i \in (1, l) \quad (2)$$

3.3.3. Abnormal KPI Screening Based on 3-Sigma

Since the residual sequence values of each KPI are quite different, selecting a unified threshold can have a great impact on the accuracy of results. The 3-sigma method [26] conducts a separate analysis of each KPI data point, and considers that 99.7% of the probability of a given data point's value is concentrated in the interval $[\mu - 3\sigma, \mu + 3\sigma]$ (μ is the average, σ is standard deviation). The probability of exceeding this range is only 0.3%, which is very low. Therefore, values not in the $[\mu - 3\sigma, \mu + 3\sigma]$ range are anomalies.

In this paper, the 3-sigma method is used to filter all KPIs based on the residual sequence. All kinds of abnormal KPIs in the known time period are obtained and constitute a set. Finally, the time range of KPIs selected as outliers is recorded.

3.4. KPI Alarm Clustering Based on Correlation Analysis

On the basis of the previously detected abnormal KPI set, this section uses Coflux [2] to analyze the correlation of all abnormal KPIs. A correlation matrix between KPIs is obtained. Finally, alarm clustering is realized.

3.4.1. Correlation Analysis of Abnormal KPIs

A sliding window is added to analyze the correlation of abnormal KPIs based on Coflux [2]. The time series of two KPIs with length l after feature amplification using the Coflux method are $X = (x_1, x_2, \ldots, x_l)$ and $Y = (y_1, y_2, \ldots, y_l)$. For fixed sequence Y, we let the sequencing slide forward and backward, and calculate the inner products for each sliding step $|s|$ of sequence X. The process is shown in Equations (3) and (4):

$$X_s = \begin{cases} [\overbrace{0, \cdots, 0}^{|s|}, x_1, \cdots, x_{l-s}], for\ s \geq 0 \\ [x_{l-s}, \cdots, x_l, \underbrace{0, \cdots, 0}_{|s|}], for\ s < 0 \end{cases} \tag{3}$$

With sliding s, the cross-correlation values of sequence X and sequence Y are calculated as follows:

$$CC(X_s, Y) = \frac{\sum_{i=-l+1}^{l-1} X_s[i] \times Y[i]}{\sqrt{(\sum_{i=-l+1}^{l-1} X[i] \times X[i]) \times (\sum_{i=-l+1}^{l-1} Y[i] \times Y[i])}} \tag{4}$$

where CC refers to cross-correlation and FCC refers to flux-based cross-correlation. This enumerates all s that can calculate relevant row vectors of length $2l-1$. The final correlation value, FCC, between two sequences for all values of s is selected through the following calculation process, and the shift values of two sequences are obtained. The calculation process is shown in Equations (5)–(7):

$$\min CC = \min_s(CC(X_s, Y)), s_1 = \arg\min_s(CC(X_s, Y)) \tag{5}$$

$$\max CC = \max_s(CC(X_s, Y)), s_2 = \arg\max_s(CC(X_s, Y)) \tag{6}$$

$$FCC(X, Y) = \begin{cases} [\min CC, s_1], for |\max CC| < |\min CC| \\ [\max CC, s_2], for |\max CC| \geq |\min CC| \end{cases} \tag{7}$$

Through the above correlation calculation steps, we can obtain correlation value FCC and shift value $|s|$ between two KPIs, as well as the temporal direction of sequence X and sequence Y, that is, the order of influence can be obtained via the positive and negative values of s. When the FCC value is within $[-1, 1]$, the closer the value to –1 or 1, the stronger the correlation to sequence X or Y. The shift value $|s|$ indicates the possible time interval between sequences X and Y. In the actual monitoring of KPIs, if two are affected by the same anomaly, the shift time interval is short, and so it needs to be analyzed according to different system data. If the time interval is exceeded, this indicates that there is no correlation between two KPIs, even if the correlation value is high. By analyzing the above problems, this paper adds a sliding window to the Coflux method to improve the calculation speed. The length of the sliding window is set according to the shift values of KPI and the abnormal duration. Finally, the correlation value, shift value, and temporal order after adding the sliding window are obtained.

3.4.2. Correlation Matrix Construction and Alarm Clustering

This paper proposes a correlation analysis method called R-SWFCC, and constructs a correlation matrix to realize alarm clustering between KPIs. In this paper, we set the correlation score of the KPIs as *CCscore*, and the *CCscore* comprehensively considers the correlation value, offset, and influence sequence. Firstly, whether sequence Y affects sequence X is determined. If not, a *CCscore* = 0 is set directly; that is, the sequence of correlation influences is incorrect. If the temporal order is correct, the correlation scores will continue to be calculated, and the calculation formula is shown in Equation (8):

$$CCscore = \begin{cases} 0 & , if\ X \to Y \\ W_1 * |SWFCC| + W_2 * \frac{1}{|s|+1}, if\ Y \to X\ or\ X \leftrightarrow Y \end{cases} \tag{8}$$

Here, $SWFCC$ represents the correlation calculation results after the sliding window is added on the basis of Coflux. W_1 and W_2 in Equation (8) are the two weights corresponding to the correlation value and the shift value, respectively, obtained via the correlation calculation with the sliding window. After the analysis and test, they are set as $W_1 = 0.8$ and $W_2 = 0.2$. Shift value $|s|$ is inversely proportional to correlation, that is, the higher of shift value, the lower the similarity between two sequences. Therefore, the shift value is used in the form of reciprocal calculation in Equation (8). Given that $CCscore \in [0, 1]$ and the denominator in the fraction cannot be zero, the influence of additional values on the overall results needs to be minimized. After comprehensively analyzing the above problems, this paper adds 1 to the offset, and then takes its reciprocal to prevent the denominator from being zero. Therefore, the specific form of the offset is set as $\frac{1}{s+1}$.

Through Equation (8), a correlation between two KPIs can be calculated, and a correlation matrix can be formed. Since there may be some abnormal KPIs with very low correlation in the correlation matrix, said correlation matrix can be classified and clustered through alarm clustering for all abnormal KPIs in a known period. Alarm clustering is used to classify and merge similar, associated, or identical KPIs through correlation analysis. Alarm clustering can reduce the impact of some irrelevant alarm KPIs. After the correlation calculation, a correlation matrix M, with high correlation among multiple sets of KPIs, can be obtained according to the threshold of alarm clustering, and this threshold can be obtained through experimental analysis. There are multiple sets of KPIs in the correlation matrix, and the correlation between KPIs in each group is high, while the correlation between groups is low. Aiming at the abnormal KPIs in a certain period, KPIs with high correlation are selected, and their correlation is determined by the final correlation matrix. The calculation formulas for the matrix and each element in the matrix are shown in Equations (9) and (10):

$$M = \begin{pmatrix} m_{11} & \cdots & m_{1N} \\ \vdots & \ddots & \vdots \\ m_{N1} & \cdots & m_{NN} \end{pmatrix} \tag{9}$$

$$m_{ij} = CCscore_{ij}, i \in (1, N), j \in (1, N) \tag{10}$$

There are N different KPIs in the matrix, and $m_{ij} \in [0, 1]$ represents the similarity of the transition probability between ith and jth KPIs.

3.5. HMM-Based KPI Anomaly Detection

The Hidden Markov Model is a time series probability model that includes a hidden state and an observation state. The implicit state is the actual state within the system, and the observation state refers to the state that can be directly observed, and which has a correlation with the implicit state. The HMM can be described by five elements, including two state sets and three probability matrices. These five elements are the hidden state set S, the observation state set V, the initial observation probability π, the hidden

state transition probability matrix A, and the observation state transition probability matrix B. Generally, HMM is represented as $\lambda = (A, B, \pi)$.

The Baum–Welch algorithm [30] is often used to update the parameters of the HMM. Viterbi [31] inferred the most likely explicit state in HMM. In this paper, the Baum–Welch algorithm is used to update the initial parameters of the HMM, and then the HMM is trained to obtain the most relevant and abnormal of the current KPIs. Finally, the anomaly detection of KPIs is realized. This paper uses HMM combined with the KPI correlation matrix obtained in the previous section to realize the detection of abnormal KPIs.

3.5.1. Construction of Hidden and Observed States and State Transition Matrix

In the distributed system, when an abnormal situation occurs, each KPI may be a direct impact indicator of another KPI; that is, each KPI may be an implicit state or an observation state. Therefore, this paper takes KPIs as being either in the hidden state or the explicit state, while the hidden state and the observation state are the same. However, in a real system, there are exceptions related to user timeout and data reading, and other abnormal states. The hidden state set S and observation state V in HMM that are applicable to a distributed system are represented by $S = V = (s_1, s_2, \ldots, s_N)$, where N is the number of hidden states and observer states in the HMM. The structure of this is shown in Figure 4, where element a_{ij} denotes the probability at time t that the hidden state of the system transfers to state s_j at time $t + 1$, and $b_i(v_k)$ denotes the probability of the observed state v_k being emitted by the system under state i at time t.

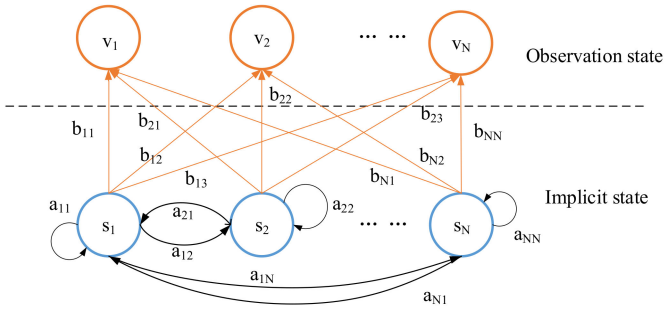


Figure 4. Hidden Markov State Model.

After determining the observed state set and hidden state set of the model, it is necessary to construct the state space. In this paper, the KPI data in the system are analyzed, and the observation state is the same as the hidden state, the number of which is N. The state transition matrix is constructed as follows:

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{NN} \end{pmatrix} \tag{11}$$

Here, the elements in the final correlation score matrix obtained in the previous section are transformed according to the requirements of $\sum_{j=1}^{N} a_{ij} = 1$ in the state transition matrix of HMM. The transformation formula for each transition probability in the matrix is as follows:

$$a_{ij} = \frac{m_{ij}}{m_{i1} + \cdots + m_{iN}}, i \in (1, N), j \in (1, N) \tag{12}$$

In this paper, according to the data characteristics of distributed systems, the observation emission matrix is the same as the state transition matrix of hidden states. The internal

transition probability of the matrix is the same as that of the state transition matrix. The observation emission matrix is obtained as formula (13) shows:

$$B = A = \begin{pmatrix} b_{11} & \cdots & b_{1N} \\ \vdots & \ddots & \vdots \\ b_{N1} & \cdots & b_{NN} \end{pmatrix} \quad (13)$$

In the state transition matrix and observation emission matrix, horizontal represents the observable state, and vertical represents the hidden state. b_{ij} represents the probability that the jth hidden state generates the ith observable state.

3.5.2. HMM Training

This section will train HMM $\lambda = \{A, B, \pi\}$. HMM training first estimates the maximum likelihood of model parameters based on the observed state sequence, but the time complexity of this direct calculation dependent on the probability formula is very high. The Baum–Welch algorithm [30] is usually used to update transition probability a_{ij}, observation probability $b_i(v_k)$, and initial observation probability π_i, from which the updated hidden state probability matrix $\widetilde{A}$, observation state probability matrix $\widetilde{B}$, and initial observation probability $\widetilde{\pi}$ are obtained. After obtaining the updated parameters, the Viterbi algorithm [31] is used to predict the state, and the final anomaly prediction result is obtained; that is, the abnormal KPIs that are most likely to affect this KPI. The Viterbi algorithm uses dynamic programming to find the maximum probability path and finally realize a prediction of the HMM, using a class of KPIs that cause anomalies related to a certain fault. Finally, other KPIs that cause a KPI exception are obtained.

4. Experiment

In this section, we conduct a large number of experiments to evaluate the efficiency and effectiveness of the proposed model. We select sample data of the 2021 International AIOps Challenge [32] to evaluate the method. Firstly, the abnormal KPIs are screened, and then a correlation matrix of the screened abnormal KPIs is constructed to realize alarm clustering. Then, we process the correlation matrix and train the HMM to find other anomaly KPIs that cause a certain anomaly. Finally, the obtained anomaly detection results are evaluated using the F1-score index. The experimental environment of this paper is the Windows 10 (64 bit) operating system. The hardware configuration is Intel(R) Core(TM) i7-1065G7 CPU@ 1.30 GHZ 1.50 GHZ, 16 GB RBM. The development language python3.7 is used. The Development tool is PyCharm 2021. The development framework is Keras, and the back-end engine is Tensorflow.

4.1. Abnormal Screening of KPIs Based on 1D-CNN-TCN

4.1.1. Data Set

The data were provided by the 2021 International AIOps Challenge. The number of anomalies in the real system environment is very small, and there is a lack of abnormal samples available to evaluate the anomaly detection method. Therefore, to evaluate the relevant model in detail, the competition data are injected into the real environment of large commercial banks by replaying the real fault type and injecting it in batches. This paper uses the first and second batches of data in the pre-match stage. The first batch of data has no anomalies, and the second batch contains abnormal KPIs that inject faults. In this experiment, the KPI data that do not contain anomalies for two days are used as the input of the 1D-CNN-TCN model, and the corresponding prediction data are obtained as the output. The residuals of the abnormal KPI and prediction data injected in the next two days are calculated to obtain the corresponding residual sequence, and this residual sequence is screened for abnormal KPIs. This experiment selects some KPIs from the metric.csv file provided by the competition for illustration, with the KPI names shown in Table 1. In Table 1, the "system" section shows that the experimental data are derived

from the b system in the AIOps competition. "kpi_name" shows the specific name of the KPI used in the experiment. "cmbdid" shows the object of the KPI index, namely, the service node.

Table 1. Information about KPIs used in experiments. "System" represents the system wherein each KPI is located, "kpi_name" shows the KPI name, and "cmbdid" shows the service node whereat each KPI is located.

System	kpi_Name	cmbdid
system-b	OSLinux-OSLinux_NETWORK_NETWORK_TCP-FIN-WAIT	Tomcat02
system-b	OSLinux-OSLinux_NETWORK_NETWORK_TCP-CLOSE-WAIT	Tomcat02
system-b	OSLinux-CPU_CPU-0_SingleCpuUtil	MG02
system-b	OSLinux-CPU_CPU-0_SingleCpuidle	MG02
system-b	OSLinux-CPU_CPU-1_ SingleCpuidle	MG02
system-b	OSLinux-CPU_CPU-1_SingleCpuUtil	MG02
system-b	OSLinux-CPU_CPU_CPUUserTime	MG02
system-b	OSLinux-CPU_CPU_CPULoad	MG02
system-b	OSLinux-CPU_CPU_CPUidleutil	MG02
system-b	OSLinux-CPU_CPU_CPUCpuUtil	MG02
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKTps	Tomcat03
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKPercentBusy	Tomcat03
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWTps	Tomcat03
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKReadWrite	Tomcat03
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWrite	Tomcat03
system-b	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKBps	Tomcat03
system-b	OSLinux-CPU_CPU_CPUidleutil	Tomcat03
system-b	OSLinux-OSLinux_MEMORY_MEMORY_MEMUsedMemPerc	Tomcat03

The 2021 AIOps competition data were generated based on a simulation of large commercial banks, and include the metric data file, the trace data file, the log data file, and the specific description of the injected fault. The KPI data used in this paper are from the metric file in the competition data set. In the metric file, the time stamp, service node name, KPI name, and the specific values of each KPI under different timestamps are given. An example of the format of the monitoring index data is shown in Table 2. In the fault label file, the specific fault time, duration, fault category, fault content, fault service node, and specific root cause index of the injected fault are described in detail. Due to the massive content of the fault label file, the relevant information on fault time, duration, service node, and specific root cause index of the fault label of the two periods addressed in this paper are selected for display. These details can be seen in Table 3.

Table 2. Example of monitoring indicator data format.

Timestamp	cmdb_id	kpi_Name	Value
1611224141	os_001	cpu_idle	80

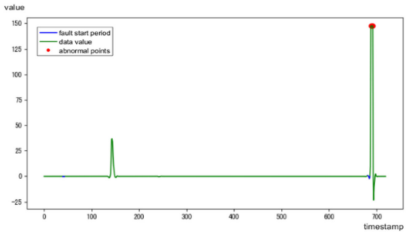
Table 3. Part of the fault tag file. The table contains two abnormal periods, each of which contains multiple abnormal indicators suspected as root cause indicators.

Time	Duration	cmbdid	Anomalous Indicator
8:39:00-04-03-2021	300	MG02	OSLinux-CPU_CPU-3_SingleCpuUtil; OSLinux-CPU_CPU-0_SingleCpuidle; OSLinux-CPU_CPU-0_SingleCpuUtil; OSLinux-CPU_CPU-1_SingleCpuidle; OSLinux-CPU_CPU-1_SingleCpuUtil; OSLinux-CPU_CPU-3_SingleCpuidle; OSLinux-CPU_CPU_CPUCpuUtil; OSLinux-CPU_CPU_CPUidleutil; OSLinux-CPU_CPU_CPULoad; OSLinux-CPU_CPU_CPUUserTime;
19:49:00-04-03-2021	300	Tomcat03	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKTps; OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWTps; OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWrite; OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKReadWrite; OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKBps; OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKPercentBusy;

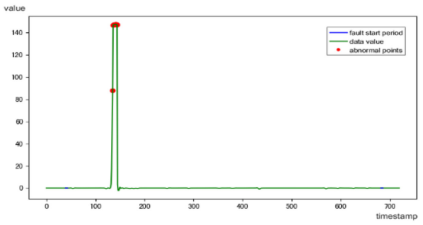
4.1.2. Results of Anomaly Screening Based on 1D-CNN-TCN

The KPI indexes in Table 1 are filtered to derive KPI indicators with overall anomalies at known times. The anomalies for each KPI are shown in Figure 5. The red annotation in the image shows the suspected anomaly point, and the blue line segment is the starting time of the injection anomalies. In the anomaly detection diagram in Figure 5, the x-axis represents the time in seconds, and the y-axis represents the value of each indicator at each time point. After using the method of this paper to screen the indicators in Table 1, the results are compared with the abnormal time points of specific KPIs given by the competition. If 3-sigma screens out abnormal KPI points, the KPI anomaly is considered in the whole period.

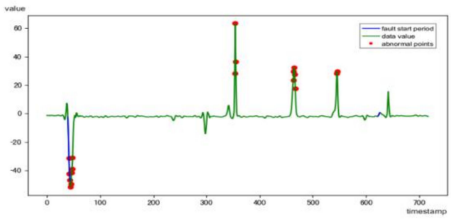
Figure 5a–s correspond to the abnormal screening results of the 18 KPIs given in Table 1. In general, abnormal KPIs with the same cause usually show anomalies within the same short period. The red node represents some of the abnormal points in the KPI abnormal period, and the two blue lines represent the starting periods of the two different faults given in the game fault label file. The abnormal times selected in Figure 5c–h,j are closer to the starting period of the first fault (the first blue line), indicating that these seven KPIs are anomalies caused by the first fault. The abnormal times selected in Figure 5a,k–p are closer to the starting time of the second fault (the second blue line), indicating that these seven KPIs are anomalies caused by the second fault. The abnormal periods of the 14 selected abnormal KPIs are near to the abnormal periods of corresponding KPIs in the fault label file given by the 2021 AIOps competition, so the abnormality screening is accurate. In Figure 5b,i, the time corresponding to the anomalies is too far from the starting time of the two faults, and so it cannot be considered that these two KPIs are caused by the faults marked by the blue line in the figure. The KPIs corresponding to Figure 5r,s do not filter out outliers, and this conforms to the instructions given by the fault tag file. Figure 5 shows that the abnormal period detected in Figure 5i does not conform to the description in the fault label file. The abnormal KPI results obtained by abnormal screening are not accurate enough. It is necessary to further analyze the abnormal KPIs screened via correlation analysis to improve the accuracy of detection. At the same time, the abnormal influence sequence between multiple indicators can be obtained.



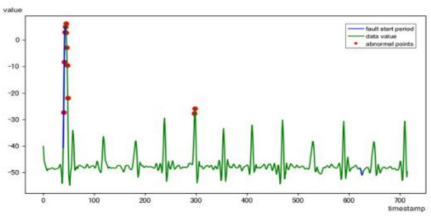
(**a**) OSLinux-OSLinux_NETWORK_NETWORK_ TCP-FIN-WAIT



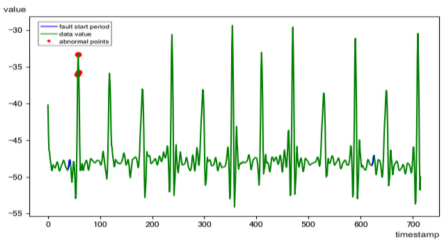
(**b**) OSLinux-OSLinux_NETWORK_NETWORK_ TCP-CLOSE-WAIT



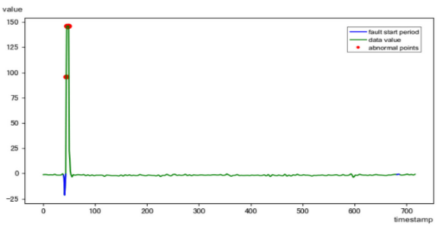
(**c**) OSLinux-CPU_CPU-0_SingleCpuUtil



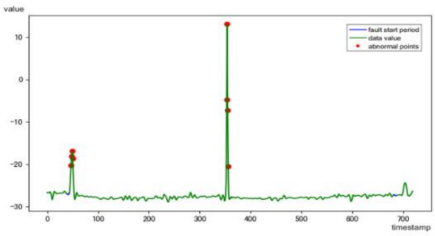
(**d**) OSLinux-CPU_CPU-0_SingleCpuidle



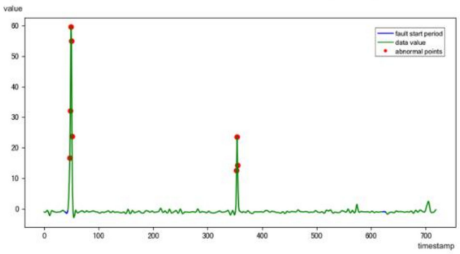
(**e**) OSLinux-CPU_CPU-1_SingleCpuidle



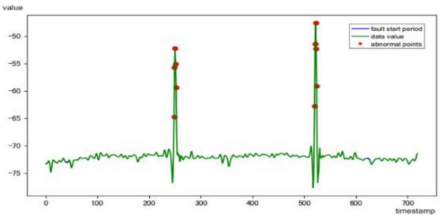
(**f**) OSLinux-CPU_CPU-1_SingleCpuUtil



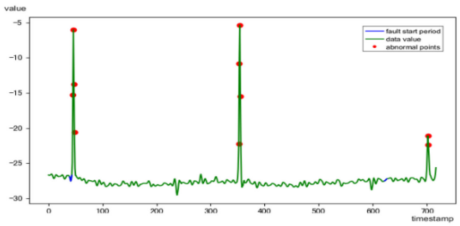
(**g**) OSLinux-CPU_CPU_CPUUserTime



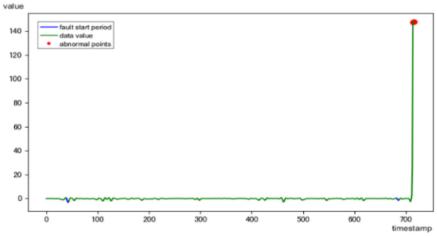
(**h**) OSLinux-CPU_CPU_CPULoad



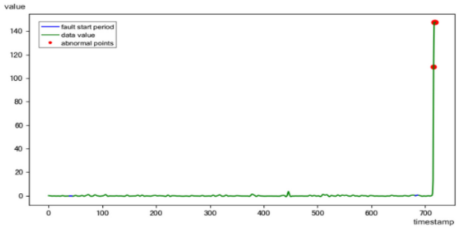
(**i**) OSLinux-CPU_CPU_CPUidleutil



(**j**) OSLinux-CPU_CPU_CPUCpuUtil

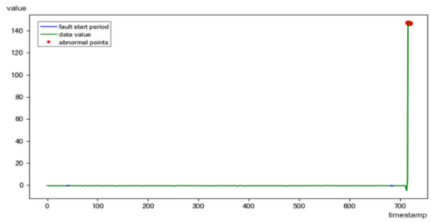


(**k**) OSLinux-OSLinux_LOCALDISK_LOCALDISK- sdb_DSKTps

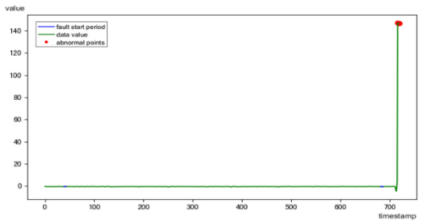


(**l**) OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKPercent Bus

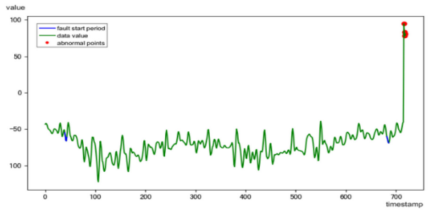
Figure 5. *Cont.*



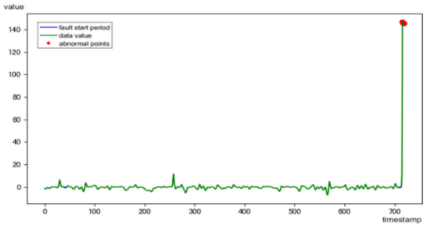
(**m**) OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWTps



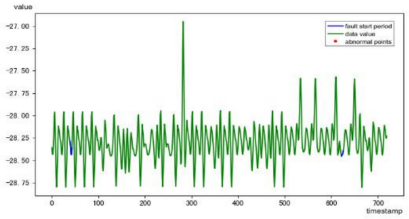
(**n**) OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKReadWrite



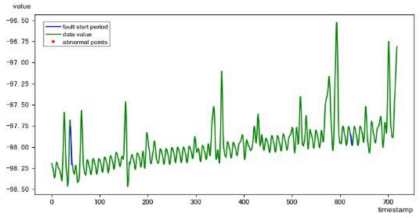
(**o**) OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWrite



(**p**) OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKBps



(**q**) OSLinux-CPU_CPU_CPUidleutil (Tomcat03)



(**r**) OSLINUX-OSLINUX_MEMORY_MEMORY_ MEMUsed-MemPerc

Figure 5. KPI curve of abnormal nodes detected during fault period. (**a**–**r**) The anomaly detection results of 18 KPIs in two periods. The red nodes represent the anomaly point, and the two blue lines represent the starting times of two different faults.

In the experiment, after repeated tests, under the premise of ensuring a good balance between time consumption and detection effect, the final hyperparameters are determined as shown in Table 4. For example, the process of kernel size parameter adjustment is shown in Figure 6. We evaluated different kernel sizes through MAE, and finally determined the kernel size to be 20.

Table 4. The hyperparameters and value of the 1D-CNN-TCN.

Hyperparameter Name	Value
kernels of CNN	128
kernel sizes of CNN	2
kernels of TCN	64
kernel sizes of TCN	20
number of iterations	100
dropout rate	0.5

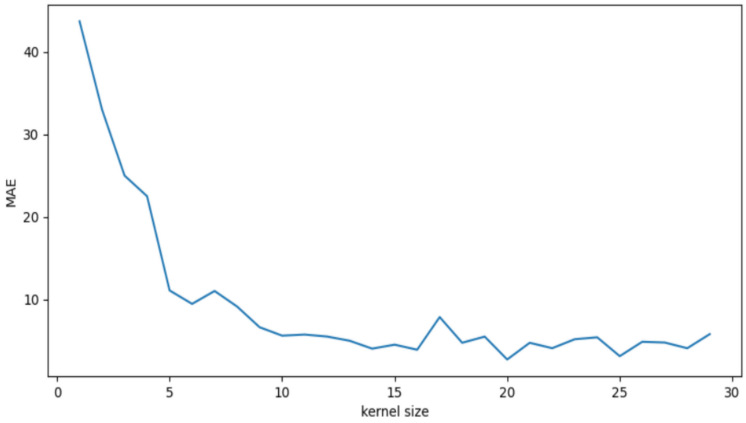


Figure 6. MAE under different kernel sizes.

4.1.3. Experimental Comparison

(1) Ablation experiment

In this paper, 1D-CNN-TCN is compared with models using only 1D CNN and only TCN for ablation experiments. The evaluation results using the 2021 AIOps Challenge data show that 1D-CNN-TCN had a better effect than 1D CNN and TCN. We used RMSE, MAE, and MAPE to evaluate these three methods. The performance comparison of the three prediction methods used on this dataset is shown in Table 5, and the best results are marked in bold.

Table 5. The proposed 1D-CNN-TCN prediction model is compared with related models. The table shows the evaluation results of the 1D-CNN-TCN, 1D CNN, and TCN ablation experiments using RMSE, MAE, and MAPE, and the evaluation results of 1D-CNN-TCN compared with a novel LSTM [33] and CNN-LSTM [34] using RMSE, MAE, and MAPE.

	RMSE	**MAE**	**MAPE**
1D CNN only	3.8277	3.0511	12.1083
TCN only	3.0243	2.2045	8.2456
A novel LSTM [33]	3.5280	2.5353	10.5672
CNN-LSTM [34]	2.5629	2.6272	8.1959
1D-CNN-TCN	2.2100	2.2004	7.9307

In Table 5, we can see that the 1D-CNN-TCN prediction method achieves the best performance among the three methods. 1D CNN alone cannot effectively capture the dependencies between time series, and only TCN can derive all the time series features, but its range is too large. Therefore, 1D-CNN-TCN first uses 1D CNN to obtain the local features of the input data, and then it uses TCN to obtain the time-dependent features, meaning the prediction results will be more accurate.

1D CNN is commonly used in 2D image data processing, and its time complexity is expressed as $O(k \cdot n \cdot d^2)$, where k is the convolution kernel size, n is the sequence length, and d is the dimension [35]. Since 1D CNN cannot effectively obtain temporal dependencies, its time series prediction results are not ideal. In contrast, TCN combines one-dimensional full convolution and causal convolution, and also uses residual network and dilated convolution, so its time complexity will be higher. However, TCN can derive the long-term temporal dependencies of time series data, and there is no problem of error accumulation. Its performance is significantly improved compared with LSTM and 1D CNN. In addition, this paper adds a layer of 1D CNN to obtain local features based on TCN, which can reduce the analysis of some secondary features and improve the efficiency of TCN processing. At the same time, in this paper, 1D-CNN-TCN achieves the fastest convergence rate compared with 1D CNN, which can make up for the problem of its high time complexity.

(2) Results and Analysis

We also compare 1D-CNN-TCN with novel LSTM [34] and CNN-LSTM [35] models, and the results are shown in Table 5. The novel LSTM consists of an LSTM network and a fully connected layer. The ability of LSTM to obtain longer time dependence is poor, so its experimental results are poor. CNN-LSTM uses LSTM instead of the TCN used in the 1D-CNN-TCN method. From the experiments, we can see that the TCN achieves better results for longer time series data prediction, while 1D CNN can capture local features, which is beneficial to the prediction effect of 1D-CNN-TCN.

Through experiments, we found that when only using CNN to predict, the time is the shortest, but the effectivity of CNN is far less than that of the novel LSTM and other time series prediction methods. The running time of 1D-CNN-TCN in this paper mainly depends on the running time of TCN. In general, the running time of 1D-CNN-TCN is longer than that of the novel LSTM, but there is no significant difference. This is because LSTM requires much training memory to increase the running time. However, the convergence rate of 1D-CNN-TCN in this paper is the fastest, and the ideal prediction effect can be achieved with fewer iterations.

4.2. Construction of Correlation Matrix

The correlation score of the KPI residual sequence is derived after abnormal KPI screening, and then the correlation matrix is constructed to realize alarm clustering. In this experiment, we divided the KPIs screened in the previous section into two groups for correlation calculation and correlation matrix construction. The specific KPIs of each group of experiments are shown in Tables 6 and 7. In Tables 6 and 7, the KPI numbers in the first column show each of the KPI indexes in the experiment. The meanings of "kpi_name" and "cmdbid" in the second and third columns are the same as those in the second and third columns in Table 1. For each group of KPIs, the F1-score (see Formula (14)–(16)) is used to evaluate the effectiveness of R-SWFCC in this paper as well as in those of Pearson [22] and Coflux [2]. Table 8 describes FP (false positive) and FN (false negative) in the correlation analysis.

Table 6. KPIs used in the first set of correlation analysis.

KPI Number	kpi_Name	cmdbid
KPI 1	OSLINUX-CPU_CPU_CPUUSERTIME	MG02
KPI 2	OSLinux-CPU_CPU-1_SingleCpuUtil	MG02
KPI 3	OSLinux-CPU_CPU-0_SingleCpuUtil	MG02
KPI 4	OSLinux-CPU_CPU-0_SingleCpuidle	MG02
KPI 5	OSLinux-CPU_CPU_CPULoad	MG02
KPI 7	OSLinux-CPU_CPU-1_SingleCpuidle	MG02
KPI 8	OSLinux-CPU_CPU_CPUCpuUtil	MG02
KPI 9	OSLinux-CPU_CPU_CPUidleutil	MG02
KPI 10	OSLinux-OSLinux_NETWORK_NETWORK_TCP-FIN-WAIT	Tomcat02
KPI 11	OSLinux-OSLinux_NETWORK_NETWORK_TCP-CLOSE-WAIT	Tomcat02
KPI 12	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWTps	Tomcat03
KPI 13	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKReadWrite	Tomcat03

Table 7. KPIs used in the second set of correlation analysis.

KPI Number	kpi_Name	cmdbid
KPI 1	OSLINUX-OSLINUX_NETWORK_NETWORK_TCP-FIN-WAIT	Tomcat02
KPI 2	OSLinux-OSLinux_NETWORK_NETWORK_TCP-CLOSE-WAIT	Tomcat02
KPI 3	OSLinux-CPU_CPU_CPUUserTime	MG02
KPI 4	OSLinux-CPU_CPU-1_SingleCpuUtil	MG02
KPI 5	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKTps	Tomcat03
KPI 7	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKPercentBusy	Tomcat03
KPI 8	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWTps	Tomcat03
KPI 9	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKReadWrite	Tomcat03
KPI 10	OSLinux-OSLinux_LOCALDISK_LOCALDISK-sdb_DSKWrite	Tomcat03

Table 8. Descriptions of FP and FN in correlation.

	FP/FN	Ground Truth	Output
Existence	FP	$X \sim Y$	$X \sim Y$
	FN	$X \sim Y$	$X \sim Y$

In Table 8, if the final correlation between X and Y is higher than the threshold, then $X \sim Y$. Otherwise, $X \not\sim Y$. We use the F1-score to evaluate the effectiveness of our method, as well as those of Coflux and Pearson. The calculation of the F1-score is shown in Equations (14)–(16):

$$Precision = \frac{TP}{TP + FP} \tag{14}$$

$$Recall = \frac{TP}{TP + FN} \tag{15}$$

$$F1 - score = \frac{2 \times Precision \times \mathrm{Recall}}{Precision + Recall} \tag{16}$$

Table 9 shows the best F1-scores of the three methods employed on two datasets. In these two datasets, the performance of proposed method is better than that of the other two methods. In the first set, the highest F1-score is 0.9162, which is derived from the method in this paper, and in the second set, the highest F1-score is 0.9020, again derived from the method in this paper.

Table 9. The best F1-scores of three algorithms for two sets of data.

Data Set	Algorithms	F1-Score
The first set	R-SWFCC	0.9162
	Coflux	0.9043
	Pearson	0.8848
The second set	R-SWFCC	0.9020
	Coflux	0.8909
	Pearson	0.7209

Figure 7 shows the times required for the correlation analyses of SWFCC, Coflux, and Pearson for different lengths of KPIs. It can be seen from the figure that the calculation speed of SWFCC is in the mid-range, and the calculation time increases linearly with the increase in KPI data length. The Pearson method is faster and the Coflux method is slower.

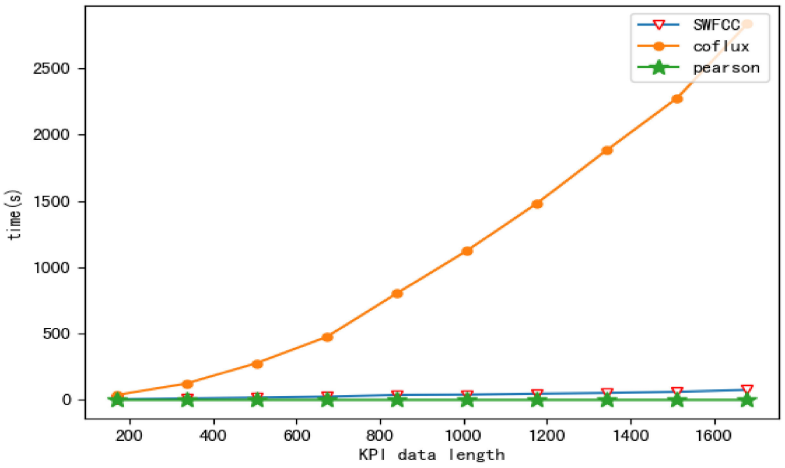
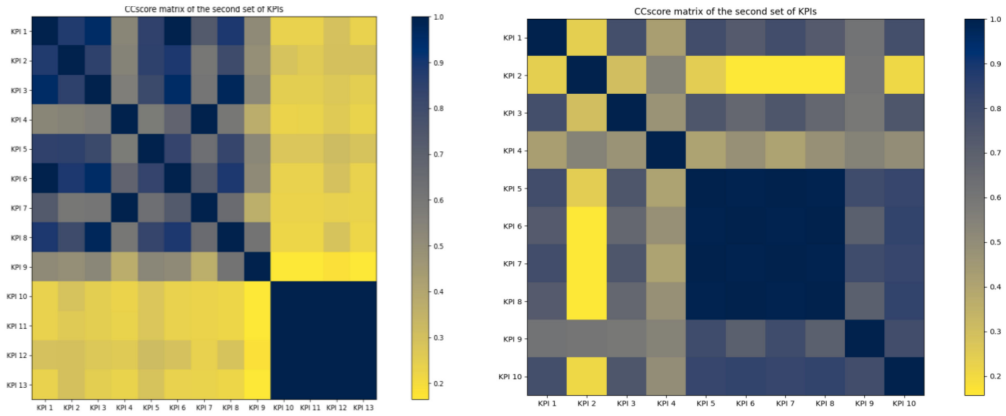


Figure 7. Calculation time of three of the methods under different KPI lengths. In the figure, the calculation speed of the SWFCC method proposed in this paper is close to that of Pearson, and the calculation speed of the Coflux method is slower.

A thermal diagram is used to show the specific correlation, as in Figure 8. A thermal diagram can show experimental results more clearly and intuitively. The matrix in Figure 8a is a heat map of the correlation matrix calculated by the first set of experimental KPIs in Table 6, and that in Figure 8b is a heat map of the correlation matrix calculated by the second set of experimental KPIs in Table 7. The value at (x, y) in the graph represents the absolute value of correlation between KPI x and KPI y. The correlation between each KPI can be more clearly shown using a thermal diagram. The higher the correlation between two KPIs, the deeper the blue in the corresponding thermal diagram; conversely, the lighter yellow in the corresponding thermal diagram means a lower correlation. It can be seen from the figure that, in the first group of experiments, the correlations between KPI 1, KPI 2, KPI 3, KPI 4, KPI 5, KPI 6, KPI 7, KPI 8, KPI 9, and KPI 10 are high, shown by the deep blue rectangle. In the second group of experiments, the correlations between KPI 5, KPI 6, KPI 7, KPI 8, KPI 9, and KPI 10 are high, again presented by a deep blue rectangle. Therefore, we selected KPIs with exceptions and high correlations in this period for the next round of anomaly detection.



(**a**) (**b**)

Figure 8. Heat map of KPI correlation matrix: (**a**) correlation matrix for the first set of KPIs; (**b**) correlation matrix for the second set of KPIs.

4.3. Performance Evaluation of Anomaly Detection

After the construction of the KPI correlation matrix, alarm clustering is employed to filter out KPIs with low similarity or no exception in the known period. We then train the HMM, and finally infer the most likely implicit state corresponding to a KPI. In order to verify the effectiveness of the anomaly screening model proposed in this paper, the detection effects of Isolation Forest, One-Class SVM, and this model are tested on the 2018 AIOps [36] Challenge dataset and the 2021 AIOps Challenge dataset, named Dataset 1 and Dataset 2, respectively. We use precision, recall, and F1-score to evaluate the performance of anomaly screening. TP is the number of abnormal points correctly detected, and FP is the number of normal points wrongly identified as abnormal points. FN is the number of abnormal points wrongly identified as normal points.

Table 10 shows the F1-scores of the three methods. Compared with the other two methods, our evaluation results are the best, indicating that our method achieves better results. However, the experimental results show that there are still false positives in this method. Moreover, because Dataset 2 contains a large number of indicator types and quantities, its F1-score will be lower than that of Dataset 1. Without the addition of a correlation analysis method, a wide variety of indicators in the system will affect the results. This shows the importance of correlation analysis when confronted with a large quantity of diverse data.

Table 10. F1-score evaluation results of different methods.

	Isolation Forest	One-Class SVM	1D-CNN-TCN
Dataset 1	0.8432	0.8064	0.8602
Dataset 2	0.8356	0.7807	0.8534

In order to verify the generalization ability of 1D-CNN-TCN, the model is tested by 5-fold crossover validation, and the evaluation index is accuracy. In the experiment, we use the 2018 AIOps challenge data, and divide the KPI data into five equal scores. One of these is taken as the test set each time, without repetition, and the other four are used as the training set to train the model. Finally, the specific accuracy values are obtained; please see Table 11.

Table 11. Accuracy of 5-fold crossover validation.

	1	2	3	4	5
accuracy	0.9799	0.9862	0.9803	0.9656	0.9825

It can be seen from Table 11, after 5-fold crossover validation, that the accuracy of the model in this paper is in the range of 0.97–0.99, and the average value is 0.9824. It shows that the model has good generalization ability. However, we can see from the comparison between F1-score and accuracy that there are still some false positives in the anomaly screening model in this paper. Therefore, correlation analysis is needed to reduce the number of false positives of abnormal KPIs.

In Table 12, if the final correlation between X and Y is higher than the threshold, then $X \sim Y$; otherwise, $X \nsim Y$. If $X \sim Y$, we continue to judge the temporal order. If X affects Y, that is, X shows an anomaly before Y, then $X \to Y$; otherwise, $Y \to X$. The F1-score (see Formulas (14)–(16)) is used to evaluate the performance of anomaly detection in this paper. Table 12 describes FP (false positive) and FN (false negative) in related issues of anomaly recognition. The experimental results of anomaly detection in this paper are shown in Table 13.

Table 12. Descriptions of FP and FN in anomaly detection.

	FP/FN	Ground Truth	Output
Existence	FP	$X \nsim Y$	$X \sim Y$
	FN	$X \sim Y$	$X \nsim Y$
Temporal direction	FP	$X \to Y$	$Y \to X$
	FN	$Y \to X$	$X \to Y$

Table 13. Best F1-scores for anomaly detection using two sets of correlation matrices.

Correlation Matrix	Best F1-Score	
	Existence	Temporal Order
The first set	0.94	0.90
The second set	0.92	0.86

According to the evaluation results of F1-score shown in Table 13, the optimal F1-scores of anomaly detection and temporal order (based on the two groups of correlation matrices) are 0.94 and 0.90, respectively, indicating that this method has a good effect on anomaly detection.

5. Conclusions

This paper proposes an anomaly detection method for KPIs based on correlation analysis and HMM. It can identify abnormal KPIs within a set period from a large number of KPIs in a system and the transfer state between them. In the anomaly detection method addressed in this paper, we first propose a 1D-CNN-TCN prediction model to predict the KPIs and obtain the residual sequence for screening the possible abnormal KPIs. This model combines CNN's local feature acquisition ability with TCN's temporally dependent feature acquisition ability to improve the prediction accuracy. The residual sequence of abnormal KPIs can highlight the abnormal segment in each KPI, such that the correlation analysis is not disturbed by the original fluctuation of KPIs, and thus the accuracy of the correlation analysis is improved. The experimental results show that the F1-score of the correlation analysis method in this paper is also the best. HMM parameters are confirmed according to the correlation matrix. After training the HMM, other KPIs that may cause a KPI anomaly are found, which reduces the time required for operation and maintenance staff to find a large number of abnormal KPIs. The results of the ablation experiment in this paper compared with the baseline method are relatively good, showing that it has certain advantages in analyzing the relationship between multiple abnormal indexes.

The method in this paper can further determine the relationship of influence between KPIs by obtaining abnormal KPIs, which can help build a fault propagation diagram and help the operation and maintenance personnel to perform rapid troubleshooting. However, the method in this paper still has some limitations, and the specific threshold of the correlation analysis method still needs to be adjusted according to different environments. The HMM can only obtain the influence relationship of KPIs for adjacent times, and cannot determine the continuous influence relationship of multiple KPIs over a long time.

There is still room for improvement in our research. Next, we will attempt to adjust the threshold adaptively, and optimize the acquisition of the relationship between KPIs.

Author Contributions: Z.S., conceptualization, methodology, investigation, validation, formal analysis, writing—original draft, writing—review and editing; Y.Z. (Yingjun Zhang), funding acquisition, supervision, validation, formal analysis, writing—review and editing; X.Z., supervision, validation, formal analysis, writing—review and editing; Y.Z. (Yun Zhao), investigation, validation, writing—original draft, writing—review and editing; Z.C., validation, formal analysis, writing—review and editing; X.W., investigation, formal analysis. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Key R&D Program of China (Grant No. 2018YFB1601502), the Liao Ning Revitalization Talents Program (Grant No. XLYC1902071), and the Fundamental Research Funds for the Central Universities (Grant No. 3132019313).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

Acknowledgments: The authors would like to thank all anonymous reviewers and editors for their helpful suggestions for the improvement of this paper.

Conflicts of Interest: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. Pei, D.; Zhang, S.; Pei, C. Intelligent operation and maintenance based on machine learning. *Commun. CCF* **2017**, *13*, 68–72.
2. Su, Y.; Zhao, Y.; Xia, W.; Liu, R.; Bu, J.; Zhu, J.; Cao, Y.; Li, H.; Niu, C.; Zhang, Y.; et al. CoFlux: Robustly correlating KPIs by fluctuations for service troubleshooting. In Proceedings of the International Symposium on Quality of Service (IWQoS), Phoenix, AZ, USA, 24–25 June 2019; pp. 1–10.

3. Laptev, N.; Amizadeh, S.; Flint, I. Generic and Scalable Framework for Automated Time-series Anomaly Detection. In Proceedings of the 21st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Sydney, Australia, 10–13 August 2015; pp. 1939–1947.
4. Liu, D.P.; Zhao, Y.J.; Xu, H.W.; Sun, Y.Q.; Pei, D.; Luo, J.; Jing, X.W.; Feng, M. Opprentice: Towards Practical and Automatic Anomaly Detection Through Machine Learning. In Proceedings of the ACM Internet Measurement Conference(IMC), Tokyo, Japan, 28–30 October 2015; pp. 211–224.
5. Niu, Z.; Yu, K.; Wu, X. LSTM-Based VAE-GAN for Time-Series Anomaly Detection. *Sensors* **2020**, *20*, 3738. [CrossRef] [PubMed]
6. Zhang, L.; Zhang, W.; McNeil, M.J.; Chengwang, N.; Matteson, D.S.; Bogdanov, P. AURORA: A Unified fRamework fOR Anomaly detection on multivariate time series. *Data Min. Knowl. Discov.* **2021**, *35*, 1882–1905. [CrossRef] [PubMed]
7. Xu, H.; Chen, W.; Zhao, N.; Li, Z.; Bu, J.; Li, Z.; Liu, Y.; Zhao, Y.; Pei, D.; Feng, Y.; et al. Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications. In Proceedings of the 2018 World Wide Web Conference (WWW), Lyon, France, 23–27 April 2018.
8. Li, T.; Comer, M.L.; Delp, E.J.; Desai, S.R.; Mathieson, J.L.; Foster, R.H.; Chan, M.W. Anomaly Scoring for Prediction-Based Anomaly Detection in Time Series. In Proceedings of the 2020 IEEE Aerospace Conference (AeroConf), Yellowstone Conference Center, Big Sky, MT, USA, 7–14 March 2020; pp. 1–7.
9. Wu, J.; Zeng, W.; Chen, H.; Tang, X. Approach of measuring and predicting software system state based on hidden Markov model. *J. Softw.* **2016**, *27*, 3208–3222.
10. Zhou, Z.; Zhang, Y.; Wang, S. A Coordination System between Decision Making and Controlling for Autonomous Collision Avoidance of Large Intelligent Ships. *J. Mar. Sci. Eng.* **2021**, *9*, 1202. [CrossRef]
11. Jiang, J.R.; Kao, J.B.; Li, Y.L. Semi-Supervised Time Series Anomaly Detection Based on Statistics and Deep Learning. *Appl. Sci* **2021**, *11*, 6698. [CrossRef]
12. Yang, X.; Latecki, L.J.; Pokrajac, D. Outlier Detection with Globally Optimal Exemplar-Based GMM. In Proceedings of the International Conference on Data Mining (SDM), Sparks, NV, USA, 30 April–2 May 2009; pp. 145–154.
13. Li, Z.; Zhao, Y.; Liu, R.; Pei, D. Robust and Rapid Clustering of KPIs for Large-Scale Anomaly Detection. In Proceedings of the 26th IEEE/ACM International Symposium on Quality of Service (IWQoS), Banff, AB, Canada, 4–6 June 2018; pp. 1–10.
14. Qu, Z.; Lun, S.; Wang, X.; Zheng, S.; Song, X. A Unsupervised Learning Method of Anomaly Detection Using GRU. In Proceedings of the IEEE International Conference on Big Data and Smart Computing (BigComp), Shanghai, China, 15–17 January 2018; pp. 685–688.
15. Provotar, O.I.; Linder, Y.M.; Veres, M.M. Unsupervised Anomaly Detection in Time Series Using LSTM-Based Autoencoders. In Proceedings of the IEEE International Conference on Advanced Trends in Information Theory (ATIT), Kyiv, Ukraine, 18–20 December 2019; pp. 513–517.
16. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. Detecting Spacecraft Anomalies Using LSTMs and Nonparametric Dynamic Thresholding. In Proceedings of the 24th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), London, UK, 19–23 August 2018; pp. 387–395.
17. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. *arXiv* **2018**, arXiv:1803.01271.
18. Tang, Z.; Chen, Z.; Bao, Y.; Li, H. Convolutional neural network-based data anomaly detection method using multiple information for structural health monitoring. *Struct. Control. Health Monit.* **2019**, *26*, e2296.1–e2296.22. [CrossRef]
19. Ren, H.S.; Xu, B.X.; Wang, Y.J.; Yi, C.; Huang, C.R.; Kou, X.Y.; Xing, T.; Yang, M.; Tong, J.; Zhang, Q. Time-Series Anomaly Detection Service at Microsoft. In Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), Anchorage, AK, USA, 4–8 August 2019; pp. 3009–3017.
20. Li, Z.H.; Xiang, Z.J.; Gong, W.J.; Wang, H. Unified model for collective and point anomaly detection using stacked temporal convolution networks. *Appl. Intell.* **2021**, *4*, 1–14. [CrossRef]
21. Weng, J.; Wang, J.H.; Yang, J.; Yang, Y. Root cause analysis of anomalies of multitier services in public clouds. In Proceedings of the 25th IEEE/ACM International Symposium on Quality of Service (IWQoS), Vilanova, Spain, 14–16 June 2017; pp. 1–6.
22. Kao, J.B.; Jiang, J.R. Anomaly Detection for Univariate Time Series with Statistics and Deep Learning. In Proceedings of the IEEE Eurasia Conference on IOT, Communication and Engineering (IEEE ECICE), Yunlin, Taiwan, 3–6 October 2019; pp. 404–407.
23. Luo, C.; Lou, J.; Lin, Q.; Fu, Q.; Ding, R.; Zhang, D.; Wang, Z. Correlating events with time series for incident diagnosis. In Proceedings of the 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), New York, NY, USA, 24–27August 2014; pp. 1583–1592.
24. Deng, S.; Zhang, N.; Zhang, W.; Chen, J.; Pan, J.Z.; Chen, H. Knowledge-Driven Stock Trend Prediction and Explanation via Temporal Convolutional Network. In Proceedings of the World Wide Web Conference (WWW), San Francisco, CA, USA, 13–17 May 2019; pp. 678–685.
25. Li, J.; Wu, B.; Sun, X.; Wang, Y. Causal Hidden Markov Model for Time Series Disease Forecasting. In Proceedings of the Computer Vision and Pattern Recognition (CVPR), Virtual. Nashville, TN, USA, 19–25 June 2021; pp. 12105–12114.
26. Pukelsheim, F. The three sigma rule. *Am. Stat.* **1994**, *48*, 88–91.
27. LeCun, Y.; Boser, B.; Denker, J.; Henderson, D.; Howard, R.; Hubbard, W.; Jackel, L. Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Comput.* **1989**, *1*, 541–551. [CrossRef]
28. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the 4th International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, 2–4 May 2016.

29. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
30. Mikls, I.; Meyer, I.M. A linear memory algorithm for baum. *BMC Bioinform.* **2005**, *6*, 1–8.
31. Rabiner, L.R. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. IEEE* **1989**, *77*, 257–286. [CrossRef]
32. 2021 International AIOps Challenge. Available online: http://iops.ai/competition_detail/?competition_id=17&flag=1 (accessed on 8 October 2021).
33. Miao, K.; Han, T.; Yao, Y.; Lu, H.; Chen, P.; Wang, B.; Zhang, J. Application of LSTM for short term fog forecasting based on meteorological elements. *Neurocomputing* **2020**, *408*, 285–291. [CrossRef]
34. Lu, W.; Li, J.; Li, Y.; Sun, A.; Wang, J. A CNN-LSTM-Based Model to Forecast Stock Prices. *Complexity* **2020**, *2020*, 6622927. [CrossRef]
35. Vaswani, A.; Shazeer, N.M.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is All you Need. In Proceedings of the Neural Information Processing Systems (NIPS), Long Beach, CA, USA, 4–9 December 2017.
36. 2018 International AIOps Challenge. Available online: http://iops.ai/competition_detail/?competition_id=5&flag=1 (accessed on 8 October 2021).