Hindawi
Scientific Programming
Volume 2021, Article ID 6636270, 11 pages
https://doi.org/10.1155/2021/6636270

Research Article

A Novel Deep Learning Approach for Anomaly Detection of Time Series Data

Zhiwei Ji,[1] **Jiaheng Gong**,[2] **and Jiarui Feng**[1]

[1]*College of Artificial Intelligence, Nanjing Agricultural University, No. 1 Weigang Road, Nanjing, Jiangsu 210095, China*
[2]*School of Information and Electronic Engineering, Zhejiang Gongshang University, 18 Xuezheng Street, Hangzhou 311300, China*

Correspondence should be addressed to Zhiwei Ji; zhiwei.ji@njau.edu.cn

Received 3 January 2021; Revised 7 May 2021; Accepted 10 July 2021; Published 21 July 2021

Academic Editor: Jiwei Huang

Anomalies in time series, also called "discord," are the abnormal subsequences. The occurrence of anomalies in time series may indicate that some faults or disease will occur soon. Therefore, development of novel computational approaches for anomaly detection (discord search) in time series is of great significance for state monitoring and early warning of real-time system. Previous studies show that many algorithms were successfully developed and were used for anomaly classification, e.g., health monitoring, traffic detection, and intrusion detection. However, the anomaly detection of time series was not well studied. In this paper, we proposed a long short-term memory- (LSTM-) based anomaly detection method (LSTMAD) for discord search from univariate time series data. LSTMAD learns the structural features from normal (nonanomalous) training data and then performs anomaly detection via a statistical strategy based on the prediction error for observed data. In our experimental evaluation using public ECG datasets and real-world datasets, LSTMAD detects anomalies more accurately than other existing approaches in comparison.

1. Introduction

Time series analysis is a hot research topic, which mainly includes two aspects: (1) identifying the nature of the phenomenon represented by the time series of observation [1] and (2) predicting future values of the time series variable based on historic data [2,3]. It was widely used in many areas in the real world, e.g., signal processing, pattern recognition [4], mathematical finance [5], weather forecasting [6], and control engineering [7]. Particularly, anomaly detection of time series is a more important direction, which promotes the development of outlier recognition techniques in real-time big data [8].

In the past years, many computational approaches were developed and used for anomaly detection in many applications, e.g., traffic detection or network intrusion detection. They can be categorized to three classes: (1) statistical modeling [9–14], (2) data mining-based techniques [15–21], and (3) machine learning-based approaches [22–29]. A lot of previous studies revealed that the above models have been successfully used for anomaly classification [10,17,18]; however, the computational frameworks focusing on abnormal subsequence detection in time series are still not well developed.

Recent studies show that some time series analysis approaches [30,31] can work well, particular to some well-known public time series, such as EEG and ECG datasets. However, they face the challenges to the generalization, robustness, and efficiency [32]. These approaches always failed when they were applied on the real-world problems [31]. Because the time series from the real world is always complicated, including missing values, high noise, and normalization issue, therefore new computational strategies are urgent to address the above problems.

As a branch of machine learning, deep learning (DL) methods offer a lot of promise rather than traditional machine learning, including higher accuracy, greater flexibility, stronger generalization, and less dependency on domain

knowledge [33–35]. Thus, it provides a new way to improve the area of anomaly classification of time series [36]. Different from various popular computational tools of anomaly classification, DL-based discord search in time series was not well studied. As a new type of DL model, long short-term memory (LSTM) provides great power in time series forecasting [37,38], which raises a question whether we can use LSTM to achieve discord search. In this study, we proposed a LSTM-based anomaly detection approach (LSTMAD) for identifying the abnormal subsequence from univariate time series. LSTMAD can learn the temporal structure of normal signal from the historic values so that it can easily identify the discords in the testing series. In the simulation experiments, we applied our LSTMAD model on various time series datasets and found that it can offer high accuracy. Moreover, LSTMAD also outperformed three other typical discord search algorithms. In summary, the developed LSTMAD provides a new pipeline to accurately capture abnormal sequences in the real-time systems.

The rest of the paper is structured as follows: in Sections 2 and 3, the related work and the proposed computational approach LSTMAD are presented. In Section 4, the datasets for validation and the experiment design are described in detail. In addition, this section describes the steps and parameter settings of the method in detail. In Section 5, the simulation results are shown and discussed, while in Section 6 conclusions are drawn and suggestions for future work are presented.

2. Related Works

According to the previous works reported in literature, the computational approaches for anomaly detection can be summarized as three categories: statistical approaches, data mining based techniques, and machine learning. We summarized these methods as follows.

2.1. Statistical Approaches. Yamanishi et al. proposed a Gaussian mixture model by scoring each data point and identifying the outlier with high scores [9]. Zhang and coworkers proposed a mathematical criterion to distinguish between normal and abnormal data using statistical algorithms [10]. Kosek et al. developed a regression model based method for anomaly detection [11]. Goldsein et al. proposed histogram-based outlier detection (HBOS) algorithm, which assumes independence of the features, making it much faster than multivariate anomaly detection approaches. It points out that the histogram is required if the results of outlier detection are available immediately [12]. The limitation of these approaches is that anomaly detection depends on the assumption that the data is generated in a particular statistical distribution [13].

2.2. Data Mining-Based Techniques. Solutions making anomaly detection more effective are by using data mining techniques, including clustering, or classification. Researchers have mostly used K-means clustering for grouping of similar data points [15, 16], so that the data points locating outside of these clusters were considered as anomalies. These approaches operate in an unsupervised mode; however, they may not offer accurate insights at the required level of detail in smaller datasets. Classification-based anomaly detection was also widely studied for real-world applications, e.g., traffic, intrusion, or network detection [17–20]. The goal of classification is to learn from labeled classes of training data for identifying classes of new or unknown instances [39]. However, the good performance requires that the training set must have well defined labels.

2.3. Machine Learning. In recent years, machine learning techniques were widely used for anomaly detection, including fuzzy logic [22–24], Bayesian approach [25,26], genetic algorithm [23,27], and neural network [28,29]. Nakano et al. proposed a fuzzy logic-based anomaly detection method for network anomaly detection [22]. Hamamto and coworkers developed a hybrid approach for network anomaly detection by using genetic algorithm and fuzzy logic [23]. Mascaro et al. explored the use of Bayesian networks for analyzing vessel behavior and detecting anomalies [26]. Combining the dynamic and static networks, they proved that their approach improved the detecting accuracy in vessel tracks. As the rapid progress of artificial intelligence, various neural network models, e.g., recurrent neural network (RNN) [29] and back propagation neural network (BPNN) [28], were developed to monitor the anomalies of a complicated system. These approaches work well in some special application areas; however, the generalization is still a big challenge.

Comparing with traditional machine learning methods, deep learning (DL) has stronger learning ability and can achieve higher accuracy [40]. The most frequently deep learning methods are generative adversarial network (GAN) [41], autoencoder [42], convolutional neural network (CNN) [43], and Long Short-Term Memory (LSTM) [44]. Previous studies show that almost all of the above models were applied to anomaly classification [45–47]; however, the work focusing on DL-based abnormal subsequence detection in time series is rarely reported.

Despite this, there still have been many attempts to perform anomaly detection in time series using various statistical or SVM-based methods, including MFAD [31] and LRRDS [32]. However, few attempts have been made to accurately predict the abnormal subsequence in time series using LSTM. Therefore, a proper deep learning method is required to perform anomaly detection using LSTM.

3. Method

The flowchart of the proposed LSTMAD approach is shown in Figure 1(a). The whole framework consists of four modules, including noise reduction, normalization, LSTM model, and anomaly detection. The details of each module are described in the following sections.

3.1. Noise Reduction. Since the noisy signal might be involved in the processing of data collection, which will affect the accuracy of the computational results, therefore, it is

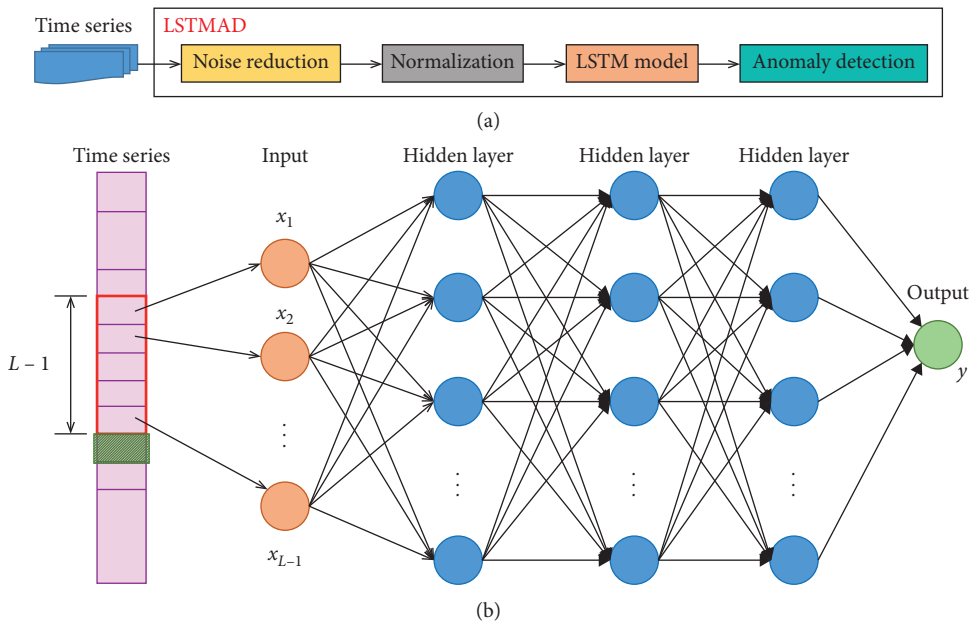

Figure 1: The flowchart of the proposed framework LSTMAD.

necessary to reduce the noise from the raw sequence before constructing anomaly detection model. In this study, we removed the noise information from time series by using S-G filter, which was proposed by Savitzky and Golay in 1964 [48]. It can be applied to a set of digital data points for the purpose of smoothing the data, to increase the precision of the data without much destroying its original properties. S-G algorithm is capable of not only removing the noise from raw data, but also ensuring the shape and width of the original signal [49, 50].

3.2. Normalization. Given a univariate time series $A = [a_1, a_2, \ldots, a_i, \ldots, a_N]$ with length N $(N > 1)$, the normalization was implemented as follows:

$$x_i = \frac{a_i - \overline{A}}{S_i},$$

$$\overline{A} = \frac{1}{N}\sum_{i=1}^{N} a_i, \tag{1}$$

$$S_i = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i - \overline{A})^2},$$

where $\overline{A}$ and S_i are the mean value and standard deviation of the original series A. The vector $X = [x_1, x_2, \ldots, x_i, \ldots, x_N]$ is the normalized sequence. After normalization, the series X will follow 0-1 normal distribution.

3.3. LSTM Model. The Long Short-Term Memory (LSTM) model was first developed by Horchreiter and Schmidhuber in 1997 [51]. Different from RNN's capability to process short term sequential data, LSTM can be used to represent the long-term dependencies in time series data [52]. A common LSTM unit is composed of a memory cell, an input gate, an output gate, and a forget gate (Figure 2). The cell remembers values over arbitrary time intervals and the three gates regulate the flow of data into and out of cell. The processing of state transition in the memory cell was implemented via formula (2)–(6). The input vector at time point t is x_t, and the hidden state vector at $t-1$ (h_1) is introduced to the LSTM unit, and then the hidden state h_t will be finally obtained. Equation (2) decides what information is going to be thrown away from the cell state via the forget gate (f_t). The input gate (i_t) decides which values to be updated, and (3) and (4) were used to update the old cell state (c_{t-1}) into the new cell state c_t. Equation (5) indicates that the output gate (o_t) decides what parts of the cell state are going to be produced as output. Finally, the cell state goes through tanh layer and multiply it by the o_t so that we get the hidden value h_t as the output of the LSTM unit (in (6)).

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \tag{2}$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right), \tag{3}$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tanh\left(W_{hc}x_t + W_{hc}h_{t-1} + b_c\right), \tag{4}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right), \tag{5}$$

$$h_t = o_t \cdot \tanh\left(c_t\right). \tag{6}$$

According to Figure 1(b), the LSTM model in our LSTMAD framework includes five layers. The input layer has $L-1$ nodes, indicating that a subseries with $L-1$ elements was used as input to a fully connected hidden layer. There are three hidden layers to process the information from input

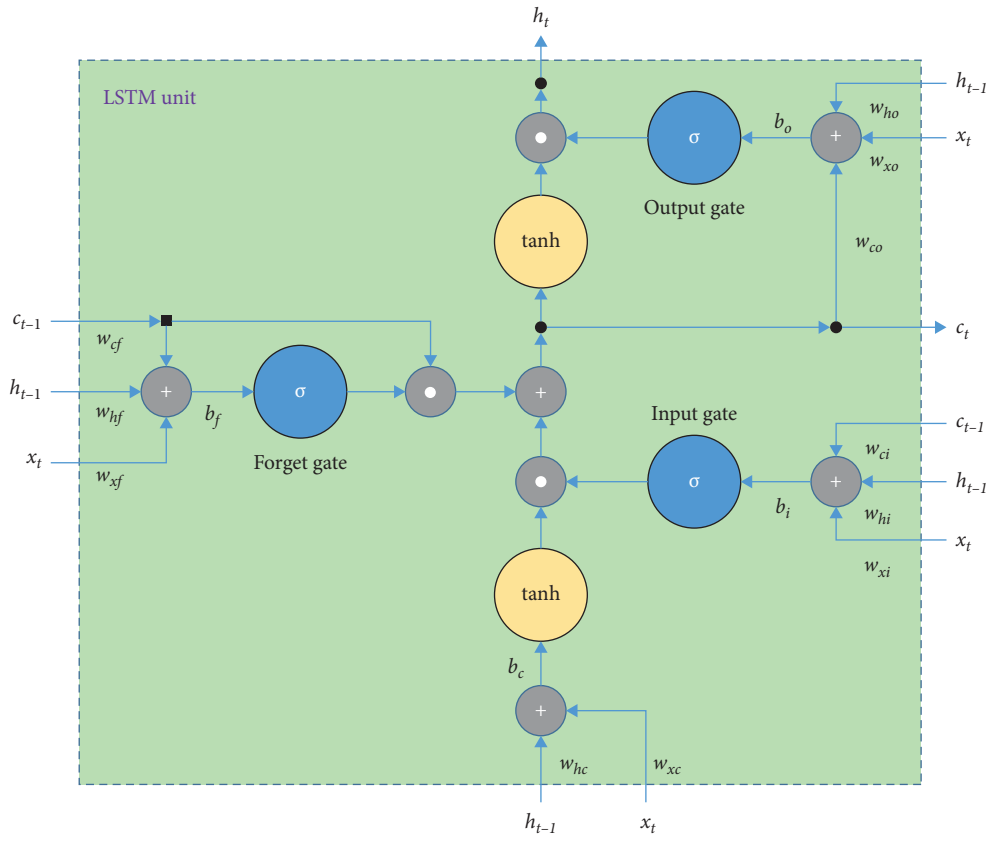

Figure 2: The basic cell unit of LSTM network.

layer. Each blue node shown in the hidden layer is a LSTM unit. The output layer only has one node (Y), which can be obtained from

$$y = \sum_i W_i^3 * h_i^3, \tag{7}$$

where W^3 is the weight of the i-th hidden node in layer 3 (last hidden layer) and output node Y. h^3 is the i-th hidden node connecting to node Y.

For a certain subseries $\mathrm{STS}_j = [x_j, x_{j+1}, \ldots, x_{j+L-1}]$, where $1 \le j \le N - L + 1$, the first elements were $L - 1$ sent to the input layer of the LSTM model simultaneously, and then the last element was considered as the expected result to be optimized. This also can be represented as $x_{j+L-1} \approx Y^* = F(\mathrm{STS}_j(1{:}\ L-1))$. Mathematically, the function $F(\cdot)$ is a trained LSTM model. Before training the LSTM model, the original time series was segmented to multiple subseries via a sliding window with length L (Figure 3). All these segmented subsequences were randomly ordered. To obtain enough training samples, each subseries was replicated k copies, where $1 \le k \le 10$. Finally, each row in the augmented matrix was input into LSTM network for model training.

According to the above description, our LSTM model can be considered as a supervised regression machine for predicting the upcoming values based on the historic data. Based on this rational, the LSTM module was firstly trained with the samples converted from the series without anomaly; hence, the model prediction would reflect the tendency of the normal signal.

3.4. Discord Search. As described above, a normal subsequence $X_{\mathrm{Trn}} = [x_1, \ldots, x_m]$ $(1 < m < N)$ was firstly extracted for LSTM training. In the meantime, a testing subsequence $X_{\mathrm{Tst}} = [x_{m+1}, \ldots, x_N]$, including discords (anomalies), will be selected. Our rationale is that a trained LSTM model "memorized" the characters of a dynamic system in normal state; hence, it can predict the future state of the system if it still normally works. Given a testing sequence that contains abnormal signals, the discord values can be easily identified by comparing the predicted values from LSTM with the observed values. The calculation for discord search includes the following steps.

3.4.1. Segmentation of the Testing Sequence X_{Tst}. Similar to the training sequence, the testing series also needs to be converted to multiple segments via sliding window (Figure 3). Here, we set the length of sliding window as L. In our experiments, we set $X_{\mathrm{Tst}} = X$ to simultaneously present the fitting error and prediction error.

3.4.2. Prediction of LSTM Model. For each segmented small piece of sequence $\mathrm{STST}_j = [x_{j+1}, \ldots, x_{j+L}]$, the element vector $\mathrm{STST}_j(1), \mathrm{STST}_j(2), \ldots, \mathrm{STST}_j(L-1)|$ would be used as input of trained LSTM model. We thus obtained the model outcome $\mathrm{Prd}_j = F(\mathrm{STST})_j$, which is the theoretical value of observation $\mathrm{STST}_j(L)$. For J testing samples (subsequences), we will obtain the prediction error vector:

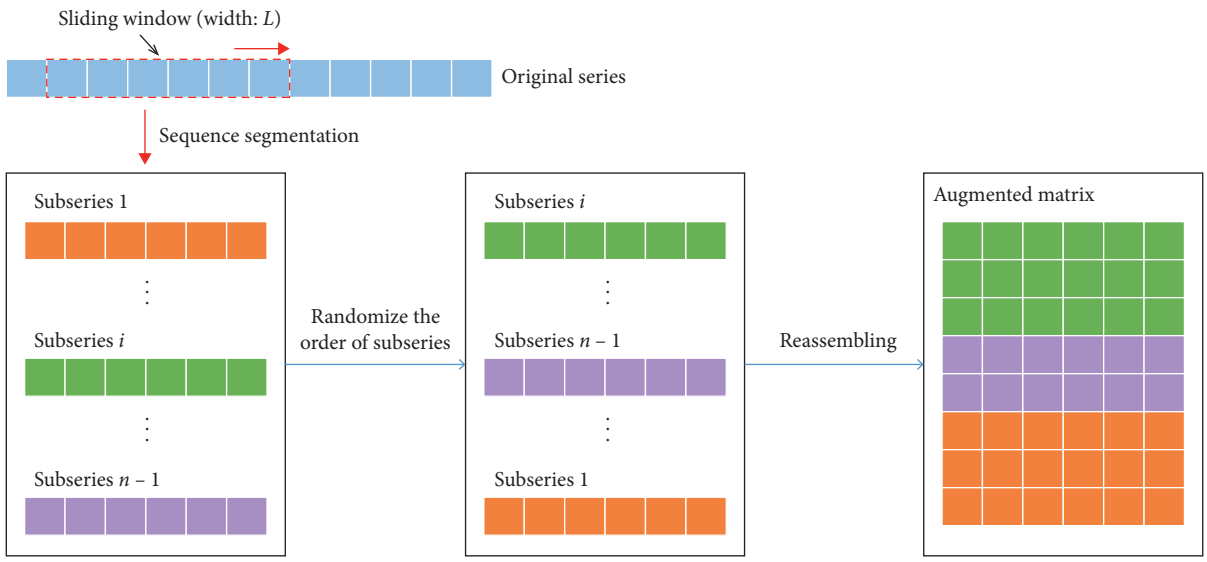

Figure 3: The basic idea of resampling in the data-preprocessing.

$\mathrm{PEV} = [\mathrm{AE}_1, \ldots, \mathrm{AE}_j, \ldots, \mathrm{AE}_J]$, where $\mathrm{AE}_j = |\mathrm{Prd}_j - \mathrm{STST}_j(L)|$.

3.4.3. Discord Search. The vector PEV reflects the difference between prediction and observation. If the value AE_j is significantly higher ($\mathrm{AE}_j \geq 0.33\,M$), the corresponding point at time $j + L$ can be considered as the peak of discord. We then use Gaussian model to fit each candidate point with significant higher value AE_j, and the abnormal sequence was finally selected from the region $[\mu - 3\sigma, \mu + 3\sigma]$ (μ and σ are the mean value and standard deviation, respectively).

4. Simulation Experiments

4.1. Data Collection and Preprocessing. To examine the performance, we applied the LSTMAD approach on 6 datasets, including four well-known public datasets and two industrial time series from the real world. The details of these datasets are described as follows.

(1) Chf01 [30] and Chf13 [53], ECG (electrocardiogram) related data, are collected from BIDMC Congestive Heart Failure Database [53,54]. The length of both datasets is 3751 and 3750, respectively. Each of them includes two series. In our experiments, we selected the 1st series from Chf01 and the 2nd series from Chf13 to test our algorithm.

(2) Ltstdb_20221 [30], an ECG dataset, is selected from Long Term ST Database. Its length is also 3750. We used the 1st series in our experiment.

(3) Xmitdb_x108 [30,55], an ECG dataset with length 5400, is selected from MIT-BIH Arrhythmia Database. The first sequence was used in our simulation.

(4) SLD1 and SLD2, two sequences related with "soil pressure" in shield tunneling machine [56], were collected from a project of shield tunnel construction in the real world. The real-time construction state was collected at each 10 seconds by local sensors. Totally, over 400 features were observed during the whole process of construction. In our experiments, we focused on the time series related with "soil pressure" because abnormal pressure is a typical fault in tunneling construction. The lengths of SLD1 and SLD2 are 18,087 and 210,907, respectively.

Before the implementation of anomaly detection, the performance of S-G filters on both categories (original signal and processed signal) of data sample was evaluated in terms of PSNR (peak signal-to-noise ratio), SNR (signal-to-noise ratio), MSE (mean square error), and PRD (root mean square difference) values [57].

4.2. Experiment Design. First, we applied the proposed LSTMAD approach on the above six datasets to prove its outstanding performance. Second, we further compared LSTMAD with three well-known algorithms, including HOT SAX [30], Robust Random Cut Forest (RRCF) [58], and Telemanom [59]. To evaluate the accuracy of anomaly detection, two statistical metrics, MCC and F_0, are defined as follows:

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}}, \tag{8}$$

$$F_0 = \frac{1}{N}\sum_{i=1}^{N} f_i, \tag{9}$$

$$f_i = \frac{\mathrm{length}(\mathrm{Pre}_i - \mathrm{Ref}_i)}{\mathrm{length}(\mathrm{Ref}_i)}. \tag{10}$$

As reported in previous studies, MCC produces a more informative and truthful score in evaluating binary classifications, particularly for the imbalanced data [60, 61]. In (8),

TP, FP, TN, and FN define the number of normal subsequences correctly detected as normal (true positive), the number of abnormal subsequences that are detected as normal (false positive), the number of abnormal subsequences that are predicted as abnormal (true negative), and the number of normal states that are recognized as abnormal (false negative). The above four variables were counted if a predicted anomaly overlapped with the observed anomaly. In addition, F_0 is defined as the global overlapping degree between predicted and observed anomalies. f_i denotes the overlapping degree of i-th abnormal subsequence between prediction (Pre_i) and observation (Ref_i).

4.3. Experiment Parameters. All the simulations were performed under the environment of Keras 2.2.4 [62] and Python 3.5.4 with Intel Core i7-7700HQ Processor and 8 G RAM (2.8 GHz). For the S-G filter, the size of sliding window is 11, and the order is 3-4. The LSTM network was constructed with five layers. The input layer includes 49 neurons, and the output layer has only one neuron. The size of the three hidden layers is 64, 256, and 100 neurons, respectively. Default parameters were set as set batch size = 500 and dropout = 0.2. Loss function is MSE (mean square error). Optimizer was set as "rmsprop" [63].

5. Results

5.1. Evaluation of Noise Reduction. Firstly, we evaluated the quality of each time series processed by S-G filter. The performance of S-G filter of data sample was compared in terms of PSNR, SNR, MSE, and PRD values. Table 1 shows that S-G filter works well on the six datasets. The anomaly detections implemented on the processed datasets are reliable.

5.2. Validation on Univariate Time Series. According to the description in Section 4.2, the proposed LSTMAD approach was tested on six time series datasets shown in the above section. The simulation results were presented as follows. The reference (observed) and predicted anomalies were highlighted with red color.

Figure 4 shows the simulation results of LSMAD on chfdb_chf01. Figure 4(a) presents a reference anomaly, which locates in the range [2182, 2392]. Figure 4(b) shows an abnormal subsequence from 2252 to 2392 identified by LSTMAD.

Comparing with Figure 4(a), we found that the predicted result is very close to the reference.

Similarly, Figure 5 shows the simulation results of LSTMAD on the dataset chfdb_chf13. In Figure 5(a), we found that the normal signal is a periodic sequence, which is repeated many times. Moreover, there is a discord located in the range [2758, 2967]. The outcomes of LSTMAD revealed that the predicted anomaly occurred in the range from 2758 to 2874 (Figure 5(b)). It indicates that the prediction of LSTMAD fit the observation well.

Different from the above two sequences (Figures 4 and 5), the series ltstdb_20221 is not easily identified because the abnormal subsequence is very similar to the normal signal. In Figure 6(a), the discord is determined in the range [583, 783]. After calculating with LSTMAD, we predicted the subsequence locating at [583, 857] as a discord (Figure 6(b)).

In addition, we examined the performance of LSTMAD on the last ECG dataset xmitdb_x108 (Figure 7). The reference and predicted anomaly locate at [3995, 4207] and [3899, 4207], respectively. Taken the above together, we found that the proposed algorithm works well on four well-known ECG datasets.

Furthermore, we applied the LSTMAD framework on two real datasets, which were generated from a shield tunnel construction project. For SLD1, the log file recorded that there was a fault ("soil pressure continues to decrease" that occurred in the region from 11,940 to 12,160. The reference discord also can be obviously identified in Figure 8(a).

The prediction of LSTMAD shows that our method is capable of capturing the abnormal subsequence (Figure 8(b)). However, the predicted discord is located at the region [11,255, 12,219], where there exists a little bias.

Finally, we tested the performance of LSTMAD on the time series SLD2. It seems that there are two peaks in the reference sequence (Figure 9(a)); however, only one fault was reported in the log file. The reference anomaly, from 173,982 to 174,002, was shown in Figure 9(a) with red color. Our algorithm successfully identified the anomaly in the range [173,982, 174,002] (Figure 9(b)). In summary, the developed LSTMAD approach not only works well on some public time series, but also works on real-world sequence.

5.3. Comparison with Other Algorithms. To further prove the effectiveness of the proposed algorithm, we tested all the above datasets on three classic anomaly detection methods: Hot SAX [30], Robust Random Cut Forest (RRCF) [58], and Telemanom [59]. Table 2 shows that LSTMAD outperformed the three other methods for anomaly detection in univariate time series. The values of MCC on six datasets show that LSTMAD can capture the abnormal subsequences in most of the time series. However, the performance of RRCF and Telemanom is obviously lower than that of others. Moreover, the measurements of F_0 on six datasets indicate that the predicted anomalies obtained from LSTMAD match the references very well. In summary, the accuracy of our approach is significantly superior to existing methods.

6. Discussion and Conclusion

In this study, we proposed a novel LSTM-based approach (LSTMAD) for anomaly detection in time series data. LSTMAD was developed by combing LSTM network with a statistical strategy. There is no need to depend on prior knowledge; our method is capable of learning the context of sequence data from the normal signal and then identifying the abnormal regions based on the prediction error for observed data. To verify the performance, we applied LSTMAD on several time series datasets, including well-known public data and real-world data. The simulation results revealed that LSTMAD can identify the discords from

Table 1: The MSE, PNSR, PRD, and SNR values of S-G filters on six datasets.

	MSE		PSNR		PRD		SNR	
	Original	Filter	Original	Filter	Original	Filter	Original	Filter
Chfdb_chf01	0.077	0.061	22.62	22.38	0.059	0.052	7.45	7.12
Chfdb_chf13	0.061	0.053	24.74	21.86	0.052	0.046	8.36	6.08
Ltstdb_20221	0.047	0.038	21.47	18.78	0.037	0.028	4.52	3.18
Xmitdb_x108	0.094	0.079	27.55	24.64	0.081	0.069	8.51	7.96
SLD1	0.026	0.019	37.65	32.59	0.018	0.014	12.76	11.97
SLD2	0.012	0.009	98.27	90.23	0.008	0.006	35.12	32.57

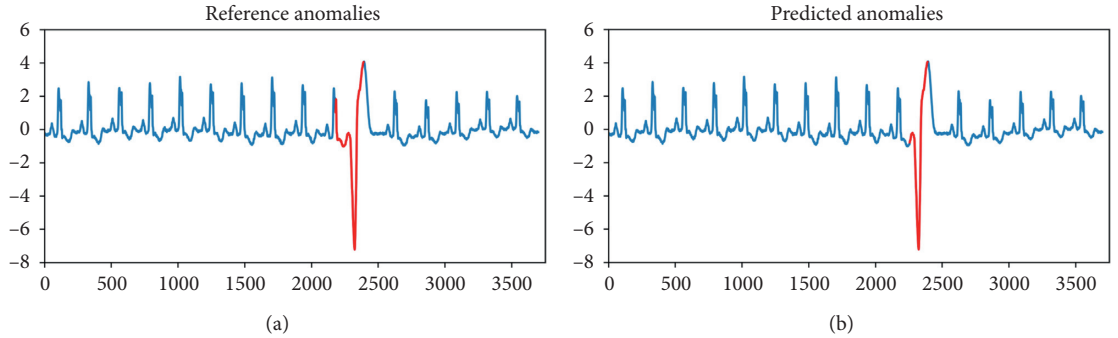

Figure 4: The reference (a) and predicted (b) anomalies in the time series chfdb_chf01.

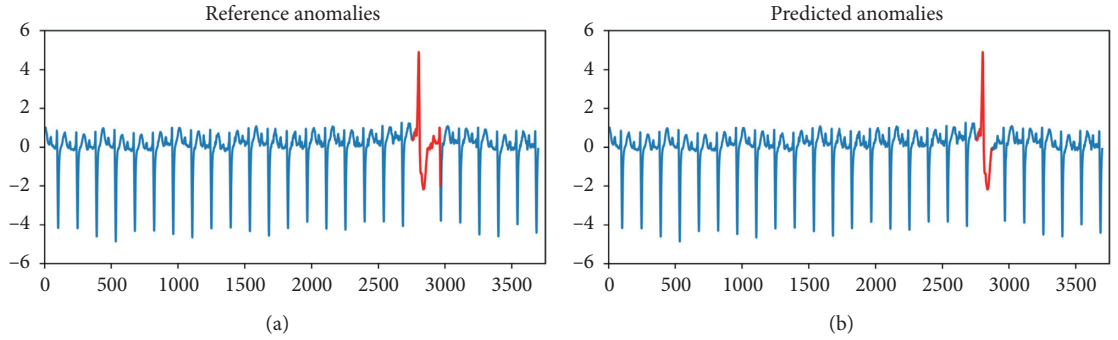

Figure 5: The reference (a) and predicted (b) anomalies in the time series chfdb_chf13.

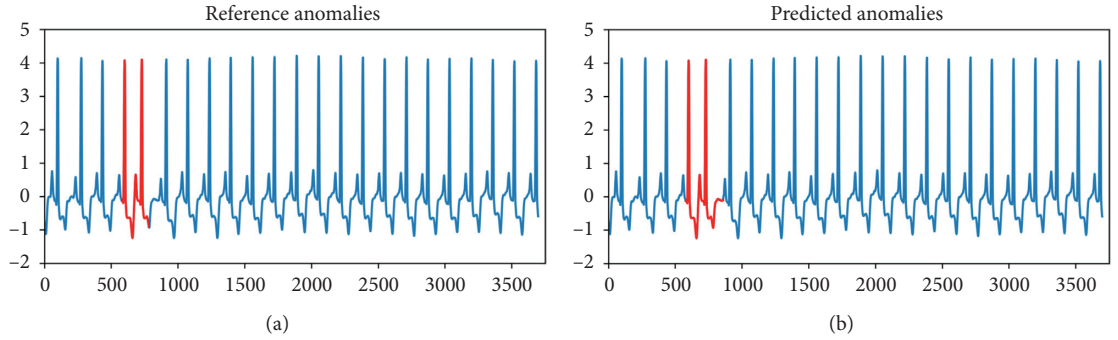

Figure 6: The reference (a) and predicted (b) anomalies in the time series ltstdb_20221.

a whole sequence with high accuracy. Moreover, LSTMAD outperformed the other golden standard approaches on all the testing datasets.

In previous studies, LSTM was widely used for time series classification or forecasting [2,3,17,44]. However, it was rarely reported for discord search in time series. We are

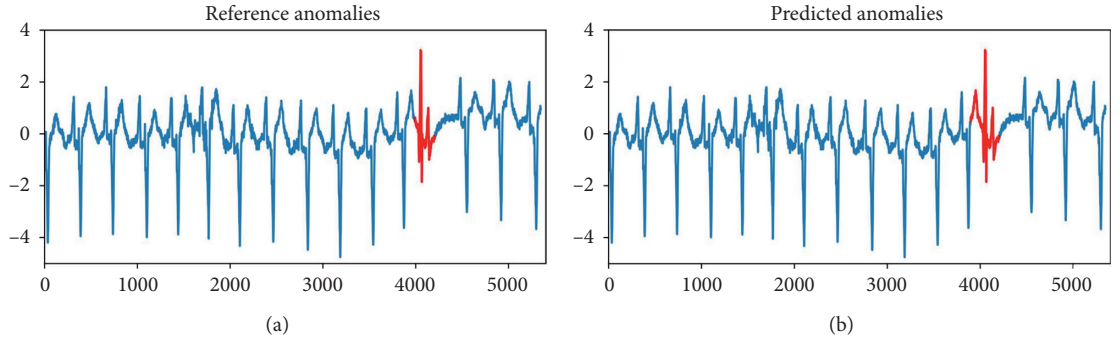

Figure 7: The reference (a) and predicted (b) anomalies in the time series xmitdb_x108.

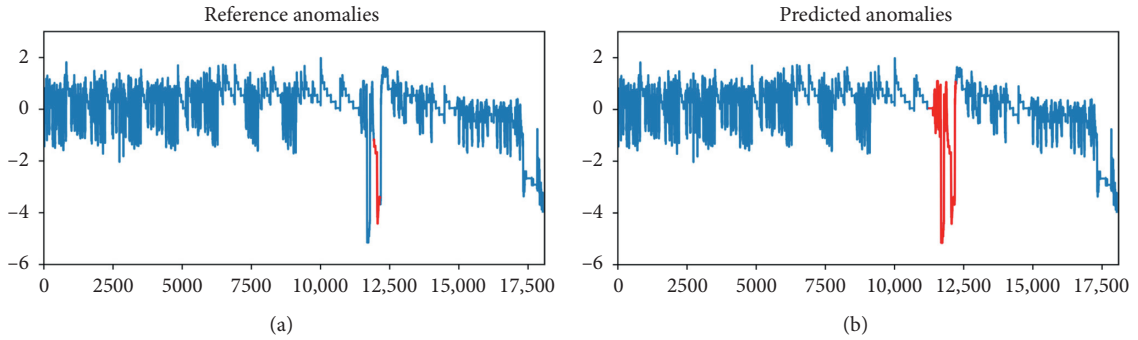

Figure 8: The reference (a) and predicted (b) anomalies in the time series SLD1.

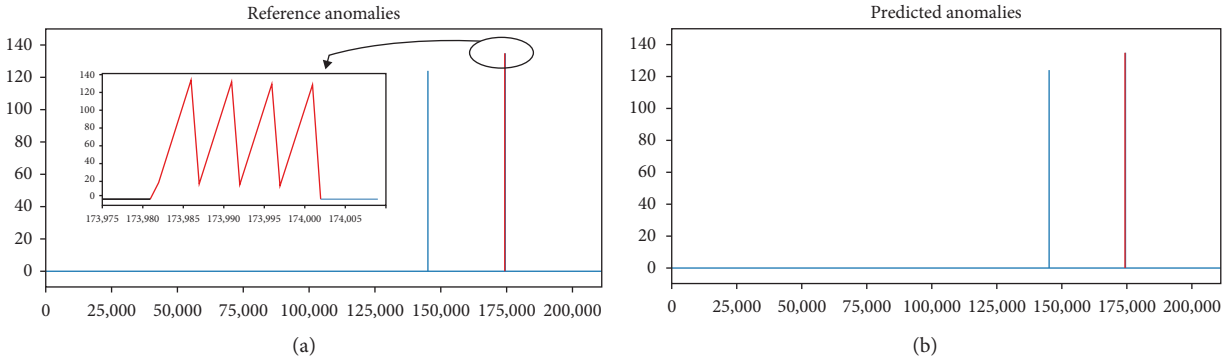

Figure 9: The reference (a) and predicted (b) anomalies in the time series SLD1.

Table 2: The comparison of LSTMAD and other algorithms.

Datasets	MCC				F_0			
	LSTMAD	RRCF	Telemanom	SAX	LSTMAD	RRCF	Telemanom	SAX
Chfdb_chf01	0.58	0.33	0.25	0.25	0.67	0.18	0.53	0.63
Chfdb_chf13	0.58	0.06	0.09	0.25	0.56	0.48	0.45	0.54
Ltstdb_20221	0.82	0.58	0.58	0.58	1.0	0.45	0.13	0.39
Xmitdb_x108	0.82	0.33	0.58	0.82	1.0	0.44	0.33	0.37
SLD1	0.41	0.25	0.06	0.35	1.0	0.12	0.08	0
SLD2	0.58	0.09	0.06	0.58	1.0	1.0	0.85	1.0

the first to build a predictive model from nonanomalous training data and then perform anomaly detection based on the prediction error for observed data. Moreover, the experimental evaluations also indicate that both the performance and generalization of LSTMAD are strong.

Our method is suitable for real-time anomaly prediction, especially when the underlying physical process is less fully understood and characterized. It does not rely on prior knowledge and is not sensitive to the length of sliding window; it thus will be a scalable algorithm for future application.

Limitations exist in the proposed LSTMAD approach. First, the current version is mainly developed for univariate time series so that it cannot directly address multivariate sequences. Second, bias also exists in the selection of public data because anomalies in periodic sequences are often more easily detected [30]. Third, enough evidence is still lacking to mathematically prove that the structure of current LSTM network is optimal. To refine the LSTMAD approach, there are four aspects that need to be considered in the future work: (1) new component will be included into the current framework to transform the multivariate sequence to univariate; (2) the rationality of the LSTM network needs further argumentation; (3) we will further design a reasonable strategy for parameter search in the future to improve the performance of our model; (4) various golden-standard time sequences need to be tested.

Data Availability

All the data used in this study are available at GitHub: https://github.com/Lostinparadise1981/LSTMAD.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this manuscript.

Authors' Contributions

Z. J. conceived and designed the algorithms. J. G. performed the simulations and processed and analyzed the data. Z. J. and J. G. wrote the paper and provided ideas to improve the computational approach. J. F. advised on the description of some analyses.

Acknowledgments

This work was supported by the National Science Foundation of Zhejiang Province (no. LY20F020003). This work was partially supported by the Startup Award of New Professor at Nanjing Agricultural University (no. 106/804001).

References

[1] E. Keogh, J. Lin, and A. D. Fu, "Hot SAX: efficiently finding the most unusual time series subsequence," in *Proceedings of the Fifth IEEE International Conference on Data Mining*, pp. 226–233, Houston,TX, USA, November 2005.

[2] Z. Ji, B. Wang, S. Deng, and Z. You, "Predicting dynamic deformation of retaining structure by LSSVR-based time series method," *Neurocomputing*, vol. 137, pp. 165–172, 2014.

[3] M. Hu, W. Li, K. Yan, Z. Ji, and H. Hu, "Modern machine learning techniques for univariate tunnel settlement forecasting: a comparative study," *Mathematical Problems in Engineering*, vol. 2019, Article ID 7057612, 12 pages, 2019.

[4] E. A. Maharaj and P. D'Urso, "A coherence-based approach for the pattern recognition of time series," *Physica A: Statistical Mechanics and Its Applications*, vol. 389, no. 17, pp. 3516–3537, 2010.

[5] G.-W. Weber, O. Defterli, S. Z. Alparslan Gök, and E. Kropat, "Modeling, inference and optimization of regulatory networks based on time series data," *European Journal of Operational Research*, vol. 211, no. 1, pp. 1–14, 2011.

[6] D. Domańska and M. Wojtylak, "Application of fuzzy time series models for forecasting pollution concentrations," *Expert Systems with Applications*, vol. 39, no. 9, pp. 7673–7679, 2012.

[7] G. Yang, H. Yang, and L. Dai, "Time-series prediction modelling based on an efficient self-organization learning neural network," *IFAC-Papers On Line*, vol. 48, no. 8, pp. 248–253, 2015.

[8] J. Bi, T. Feng, and H. Yuan, "Real-time and short-term anomaly detection for GWAC light curves," *Computers in Industry*, vol. 97, pp. 76–84, 2018.

[9] K. Yamanishi, J.-i. Takeuchi, G. Williams, and P. Milne, "Online unsupervised outlier detection using finite mixtures with discounting learning algorithms," *Data Mining and Knowledge Discovery*, vol. 8, no. 3, pp. 275–300, 2004.

[10] J. Zhang and M. Zulkernine, "Anomaly based network intrusion detection with unsupervised outlier detection," in *Proceedings of the Ieee International Conference Communication*, pp. 2388–2393, Istanbul, Turkey, June 2006.

[11] A. M. Kosek and O. Gehrke, "Ensemble regression model-based anomaly detection for cyber-physical intrusion detection in smart grids," in *Proceedings of the Ieee Electronic Power Energy Conference*, Urbana,IL, USA, February 2016.

[12] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *Plos One*, vol. 11, no. 4, Article ID e0152173, 2016.

[13] T. J. Veasey and S. J. Dodson, "Anomaly detection in application performance monitoring data," *International Journal of Machine Learning and Computing*, vol. 4, no. 2, pp. 120–126, 2014.

[14] M. Hu, Z. JiK. Yan et al., Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.

[15] U. Ravale, N. Marathe, and P. Padiya, "Feature selection based hybrid anomaly intrusion detection system using K means and RBF kernel function," *Procedia Computer Science*, vol. 45, pp. 428–435, 2015.

[16] A. P. Muniyandi, R. Rajeswari, and R. Rajaram, "Network anomaly detection by cascading *K*-means clustering and C4.5 decision tree algorithm," *Procedia Engineering*, vol. 30, pp. 174–182, 2012.

[17] S. M. A. Al Mamun and J. Valimaki, "Anomaly detection and classification in cellular networks using automatic labeling technique for applying supervised learning," *Procedia Computer Science*, vol. 140, pp. 186–195, 2018.

[18] J. A. Barria and S. Thajchayapong, "Detection and classification of traffic anomalies using microscopic traffic variables," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 695–704, 2011.

[19] F. Iglesias and T. Zseby, "Analysis of network traffic features for anomaly detection," *Machine Learning*, vol. 101, no. 1-3, pp. 59–84, 2015.

[20] X. L. Tao, Y. Peng, F. Zhao, P. C. Zhao, and Y. Wang, "A parallel algorithm for network traffic anomaly detection based on isolation forest," *International Journal of Distributed Sensor Networks*, vol. 14, no. 11, 2018.

[21] M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, A novel computational approach for discord search with local recurrence rates in multivariate time series," *Information Sciences*, vol. 477, pp. 220–233, 2019.

[22] M. Nakano, A. Takahashi, and S. Takahashi, "Fuzzy logic-based portfolio selection with particle filtering and anomaly detection," *Knowledge-Based Systems*, vol. 131, pp. 113–124, 2017.

[23] A. H. Hamamoto, L. F. Carvalho, L. D. H. Sampaio, T. Abrão, and M. L. Proença, "Network anomaly detection system using genetic algorithm and fuzzy logic," *Expert Systems with Applications*, vol. 92, pp. 390–402, 2018.

[24] Q. Li, R. Sun, H. Wu, and Q. Zhang, "Parallel distributed computing based wireless sensor network anomaly data detection in IoT framework," *Cognitive Systems Research*, vol. 52, pp. 342–350, 2018.

[25] H. N. Akouemo and R. J. Povinelli, "Probabilistic anomaly detection in natural gas time series data," *International Journal of Forecasting*, vol. 32, no. 3, pp. 948–956, 2016.

[26] S. Mascaro, A. E. Nicholso, and K. B. Korb, "Anomaly detection in vessel tracks using Bayesian networks," *International Journal of Approximate Reasoning*, vol. 55, no. 1, pp. 84–98, 2014.

[27] Y. Li, L. Guo, Z.-H. Tian, and T.-B. Lu, "A lightweight web server anomaly detection method based on transductive scheme and genetic algorithms," *Computer Communications*, vol. 31, no. 17, pp. 4018–4025, 2008.

[28] Z. Chiba, N. Abghour, K. Moussaid, A. El Omri, and M. Rida, "A novel architecture combined with optimal parameters for back propagation neural networks applied to anomaly network intrusion detection," *Computers & Security*, vol. 75, pp. 36–58, 2018.

[29] T. Kubota and W. Yamamoto, "Anomaly detection from online monitoring of system operations using recurrent neural network," *Procedia Manufacturing*, vol. 30, pp. 83–89, 2019.

[30] E. Keogh, J. Lin, and A. Fu, "Hot SAX: efficiently finding the most unusual time series subsequence," in *Proceedings of the Fifth IEEE International Conference on Data Mining (ICDM'05)*, pp. 226–233, Houston, TX, USA, November 2005.

[31] M. Hu, Z. Ji, K. Yan et al., "Detecting anomalies in time series data via a meta-feature based approach," *IEEE Access*, vol. 6, pp. 27760–27776, 2018.

[32] M. Hu, X. Feng, Z. Ji, K. Yan, and S. Zhou, "A novel computational approach for discord search with local recurrence rates in multivariate time series," *Information Sciences*, vol. 477, 2018.

[33] A. B. Levine, C. Schlosser, J. Grewal, R. Coope, S. J. M. Jones, and S. Yip, "Rise of the machines: advances in deep learning for cancer diagnosis," *Trends in Cancer*, vol. 5, no. 3, pp. 157–169, 2019.

[34] H. Chen, O. Engkvist, Y. Wang, M. Olivecrona, and T. Blaschke, "The rise of deep learning in drug discovery," *Drug Discovery Today*, vol. 23, no. 6, pp. 1241–1250, 2018.

[35] M. P. McBee, O. A. Awan, A. T. Colucci et al., "Deep learning in radiology," *Academic Radiology*, vol. 25, no. 11, 2018.

[36] M. Munir, S. A. Siddiqui, A. Dengel, and S. Ahmed, "DeepAnT: a deep learning approach for unsupervised anomaly detection in time series," *IEEE Access*, vol. 7, pp. 1991–2005, 2019.

[37] A. Sagheer and M. Kotb, "Time series forecasting of petroleum production using deep LSTM recurrent networks," *Neurocomputing*, vol. 323, pp. 203–213, 2019.

[38] N. Jin, Y. Zeng, K. Yan, and Z. Ji, "Multivariate air quality forecasting with nested LSTM neural network," *IEEE Transactions on Industrial Informatics*, 2021.

[39] A. Jain, B. Verma, and J. L. Rana, "Anomaly intrusion detection techniques: a brief review," *International Journal of Scientific & Engineering Research*, vol. 5, no. 7, pp. 1372–1383, 2014.

[40] F. Feng, X. Liu, B. Yong, R. Zhou, and Q. Zhou, "Anomaly detection in ad-hoc networks based on deep learning model: a plug and play device," *Ad Hoc Networks*, vol. 84, pp. 82–89, 2019.

[41] Y. Lin, L. Li, H. Jing, B. Ran, and D. Sun, "Automated traffic incident detection with a smaller dataset based on generative adversarial networks," *Accident Analysis & Prevention*, vol. 144, Article ID 105628, 2020.

[42] W. Xie, J. Lei, B. Liu, Y. Li, and X. Jia, "Spectral constraint adversarial autoencoders approach to feature representation in hyperspectral anomaly detection," *Neural Networks*, vol. 119, pp. 222–234, 2019.

[43] N. Chouhan, A. Khan, and H.-u.-R. Khan, "Network anomaly detection using channel boosted and residual learning based deep convolutional neural network," *Applied Soft Computing*, vol. 83, Article ID 105612, 2019.

[44] T.-Y. Kim and S.-B. Cho, "Web traffic anomaly detection using C-LSTM neural networks," *Expert Systems with Applications*, vol. 106, pp. 66–76, 2018.

[45] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class SVM with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[46] L. Li, B. Du, Y. Wang, L. Qin, and H. Tan, "Estimation of missing values in heterogeneous traffic data: application of multimodal deep learning model," *Knowledge-Based Systems*, vol. 194, Article ID 105592, 2020.

[47] L. Li, Y. Lin, B. Du, F. Yang, and B. Ran, "Real-time traffic incident detection based on a hybrid deep learning model," *Transportmetrica A: Transport Science*, vol. 1813214, 2020.

[48] K. Baba, L. Bahi, and L. Ouadif, "Enhancing geophysical signals through the use of Savitzky-Golay Filtering method," *Geofísica Internacional*, vol. 53, no. 4, pp. 399–409, 2014.

[49] M. J. Zimoń, J. M. Reese, and D. R. Emerson, "A novel coupling of noise reduction algorithms for particle flow simulations," *Journal of Computational Physics*, vol. 321, pp. 169–190, 2016.

[50] R. Cao, Y. Chen, M. Shen et al., "A simple method to improve the quality of NDVI time-series data by integrating spatiotemporal information with the Savitzky-Golay filter," *Remote Sensing of Environment*, vol. 217, pp. 244–257, 2018.

[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[52] K. Greff, R. K. Srivastava, K. Jan, B. R. Steunebrink, and S. Jürgen, "LSTM: a search space odyssey," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 28, no. 10, pp. 2222–2232, 2017.

[53] D. S. Baim, W. S. Colucci, E. S. Monrad et al., "Survival of patients with severe congestive heart failure treated with oral

milrinone," *Journal of the American College of Cardiology*, vol. 7, no. 3, pp. 661–670, 1986.

[54] A. L. Goldberger, L. A. N. Amaral, L. Glass et al., "PhysioBank, PhysioToolkit, and PhysioNet," *Circulation*, vol. 101, no. 23, pp. E215–E20, 2000.

[55] G. Li, O. Bräysy, L. Jiang, Z. Wu, and Y. Wang, "Finding time series discord based on bit representation clustering," *Knowledge-Based Systems*, vol. 54, pp. 243–254, 2013.

[56] C. Zhou, T. Kong, Y. Zhou, H. T. Zhane, and L. Y. Ding, "Unsupervised spectral clustering for shield tunneling machine monitoring data with complex network theory," *Automation Construction*, vol. 107, 2019.

[57] T. N. Nagabhushan, V. N. Aradhya, P. Jagadeesh, and S. Shukla, "Communications in computer and information science," in *Proceedings of the Third International Conference CCIP*, Bengaluru, India, December 2017.

[58] M. Bartos, A. Mullapudi, and S. T. rrcf, "Implementation of the Robust random Cut forest algorithm for anomaly detection on streams," *The Journal of Open Source Software*, vol. 4, no. 35, 2019.

[59] K. Hundman, V. Constantinou, and C. Laporte, "Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 387–395, London, UK, August 2018.

[60] S. Boughorbel, F. Jarray, and M. El-Anbari, "Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric," *PLoS One*, vol. 12, no. 6, Article ID e0177678, 2017.

[61] D. Chicco and G. Jurman, "The advantages of the Matthews correlation coefficient (MCC) over *F*1 score and accuracy in binary classification evaluation," *BMC Genomics*, vol. 21, no. 1, p. 6, 2020.

[62] Y. Li, C. Huang, L. Ding, Z. Li, Y. Pan, and X. Gao, "Deep learning in bioinformatics: introduction, application, and perspective in the big data era," *Methods*, vol. 166, pp. 4–21, 2019.

[63] S. Roy, N. Das, M. Kundu, and M. Nasipuri, "Handwritten isolated Bangla compound character recognition: a new benchmark using a novel deep learning approach," *Pattern Recognition Letters*, vol. 90, pp. 15–21, 2017.