

Letter

LSTM-Based VAE-GAN for Time-Series Anomaly Detection

Zijian Niu, Ke Yu * and Xiaofei Wu

School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, China; zijian@bupt.edu.cn (Z.N.); wuxf@bupt.edu.cn (X.W.)

* Correspondence: yuke@bupt.edu.cn

Received: 19 May 2020; Accepted: 29 June 2020; Published: 3 July 2020



Abstract: Time series anomaly detection is widely used to monitor the equipment states through the data collected in the form of time series. At present, the deep learning method based on generative adversarial networks (GAN) has emerged for time series anomaly detection. However, this method needs to find the best mapping from real-time space to the latent space at the anomaly detection stage, which brings new errors and takes a long time. In this paper, we propose a long short-term memory-based variational autoencoder generation adversarial networks (LSTM-based VAE-GAN) method for time series anomaly detection, which effectively solves the above problems. Our method jointly trains the encoder, the generator and the discriminator to take advantage of the mapping ability of the encoder and the discrimination ability of the discriminator simultaneously. The long short-term memory (LSTM) networks are used as the encoder, the generator and the discriminator. At the anomaly detection stage, anomalies are detected based on reconstruction difference and discrimination results. Experimental results show that the proposed method can quickly and accurately detect anomalies.

Keywords: anomaly detection; VAE-GAN; time series

1. Introduction

In recent years, with the development of the Industrial Internet, industrial big data has become an important research topic. Due to the complicated production process, large number of sensors and high sampling frequency, it is easy for industrial equipment to accumulate a large amount of time series data in a short time [1,2]. Some anomalies occurring in the production process will cause the industrial equipment to shut down. Detecting anomalies early can improve the equipment's overall equipment effectiveness by implementing early preventive maintenance. The industrial time series data has the characteristics of large scale and week periodicity. Designing an effective anomaly detection method for it is a very valuable subject and is also the work done in this paper.

Anomalies, also referred to as outliers, are defined as observations that deviate so much from the other observations as to arise suspicions that they were generated by different mechanisms [3,4]. Most scholars give the definitions of time series anomalies based on this and the actual application field. Anomaly detection has been studied in a variety of data domains including high-dimensional data, uncertain streaming data, network data, and time series data [5–12]. A significant amount of work has been performed in time series anomaly detection. In the statistics literature, several models were proposed, including autoregressive integrated moving average (ARIMA), cumulative sum statistics (CUSUM), exponentially weighted moving average (EWMA), etc [13–16]. However, in the face of industrial time series data, traditional time series anomaly detection methods cannot meet the expected requirements in efficiency and accuracy. In the past few years, a large number of unsupervised anomaly detection methods based on deep learning have been developed [17,18]. Many scholars use neural network to learn the unknown relationship in time series data, and then build a

prediction model to detect anomalies by the deviation of the predicted value from the actual value at each time point. For example, an anomaly detection method based on LSTM prediction model was modeled on normal time series data, and anomalies were identified by comparing the residual of the predicted value and the true value [19–21]. Malhotra et.al use stacked LSTM networks trained on non-anomalous data as a predictor over a number of time steps for anomaly detection in time series [22]. Other prediction models include multilayer perceptron (MLP) predictor and support vector regression. With the development of industrial systems, time series data become more and more complicated. In the industrial production process, the behavior of machines always changes based on usage and external factors that are difficult to capture [23]. Under such circumstances, it is difficult to predict the time series even in a few time steps, resulting in the time series anomaly detection method based on the prediction model being no longer applicable.

In order to solve such problems, some reconstruction-based models were proposed. Anomaly detection methods based on autoencoder (AE) appeared. An encoder learns a vector representation of the input time-series and the decoder uses this representation to reconstruct the time-series. The method based on AE performs anomaly detection through reconstruction difference [24–27]. AE is a representative reconstruction approach that is a connected network with an encoder and a decoder. It has also been applied for reconstructing time-series data using a sliding time-window [28]. Subsequently, some time series anomaly detection methods based on variational autoencoder (VAE) were proposed [23]. Unlike an AE, a VAE models the underlying probability distribution of observations using variational inference. At present, a novel time series anomaly detection method based on GAN has been proposed [29]. The LSTM networks are used as the generator and the discriminator to capture the distribution of the time series. However, the method based on GAN needs to find the best mapping from real-time space to the latent space at the anomaly stage. This optimization process of finding the best mapping brings new errors and takes a long time, so that the system cannot provide early warning in time.

In this paper, we propose a LSTM-based VAE-GAN for time series anomaly detection, which effectively solves the above problems. The encoder, the generator and the discriminator are jointly trained to take advantage of the mapping ability of the encoder and the discrimination ability of the discriminator simultaneously. In order to capture time dependence, LSTM networks are used as the encoder, the generator and the discriminator. The model is trained on the normal time series. The encoder maps the input time series to the latent space. The generator reconstructs the input time series. The characteristics of the discriminator make it possible to judge anomalies directly from the input time series [30]. Since the encoder, the generator and the discriminator are jointly trained at the training stage, it is not necessary to calculate the best mapping from real-time space to the latent space at the anomaly detection stage. The time required at the anomaly detection stage is greatly reduced, which can make our model detect anomalies more quickly. At the same time, due to the joint optimization of the various modules of our model, our model can detect anomalies more accurately.

In addition, our model design is inspired by a method applied in images of faces, which combines variational autoencoder with a generative adversarial network and shows that this method outperforms VAEs with element-wise similarity measures in terms of visual fidelity [31–34].

To summarize, the main contributions of our work are:

- A novel anomaly detection method based on VAE-GAN is proposed to detect anomalies in times series data from sensors.
- Our method jointly trains the encoder, the generator and the discriminator, which takes advantage of the mapping ability of the encoder and the discrimination ability of the discriminator simultaneously.
- The anomaly score consists of the reconstruction difference of the VAE part and the discrimination results of the discriminator, which makes it more able to distinguish anomalies from normal data.

2. Materials and Methods

2.1. Time Series

A time series is a series of data points indexed in time order. Most commonly, it is a sequence taken at successive equally spaced points in time.

We use two time series datasets in our experiment. They are Yahoo and KPI commonly used for evaluating the performance of time-series anomaly detection. In these datasets, anomaly points are labeled as positive, and normal points are labeled as negative.

Yahoo is released by Yahoo Labs. It contains both real and synthetic time series with varying trend, noise and seasonality, representing the metrics of various Yahoo services [35]. The Yahoo dataset has four different parts, the first part A1Benchmark is real data, and the other three parts are synthetic data. The timestamps of the A1Benchmark are replaced by integers with the increment of 1, where each data-point represents 1 hour worth of data. The A1Benchmark has 94866 points in total, 1669 of which are anomalies, and the anomaly rate is 1.76%. We only use the real time series A1Benchmark to evaluate the anomaly detection methods, as shown in Figure 1a.

KPI is released by AIOPS Challenge [36]. It is collected from some Internet Companies, such as Sogo, Tencent, eBay, etc, which reflect the health status of machines (servers, routers, and switches) and quality of Web services. We take the first 10000 points and down sample it every 2 timestamps. After that, we obtain the dataset with 5000 timestamps length, 38 anomaly points, 0.76% anomaly rate, as shown in Figure 1b.

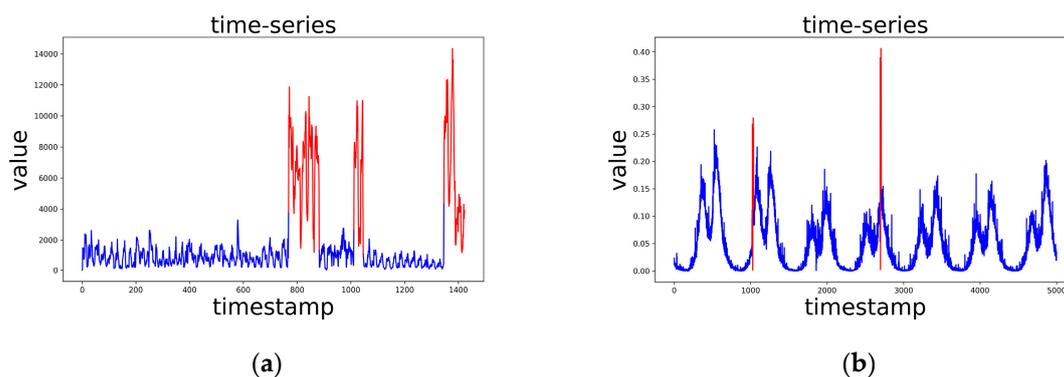


Figure 1. Time series used in our experiment, and the red parts are anomalies. (a) An example from Yahoo dataset. (b) Time series in KPI dataset.

In the data preprocessing, we use the min-max normalization to bring all values in each time series into the range $[0,1]$ and divide each time series into two halves according to the time. Since our model is aimed at learning the distribution of normal data at the model training stage, we remove the anomalies in the data of the first part to get the normal training data. The data in the second part is used for testing. In addition, we divide the time series into training data and testing data into sub-sequences by a sliding window with a size of 10 and a step-size of 3.

2.2. LSTM-Based VAE-GAN

This paper presents a LSTM-based VAE-GAN method for time series anomaly detection. The method has two stages, one is the model training stage and the other is the anomaly detection stage. Our model is trained on the normal time series data to learn the distribution of them at the model training stage and calculates the average anomaly score of each point in testing time series data by identifying whether the testing time series data conform to the normal time series data distribution at the anomaly stage. The architecture of LSTM-based VAE-GAN is shown in Figure 2.

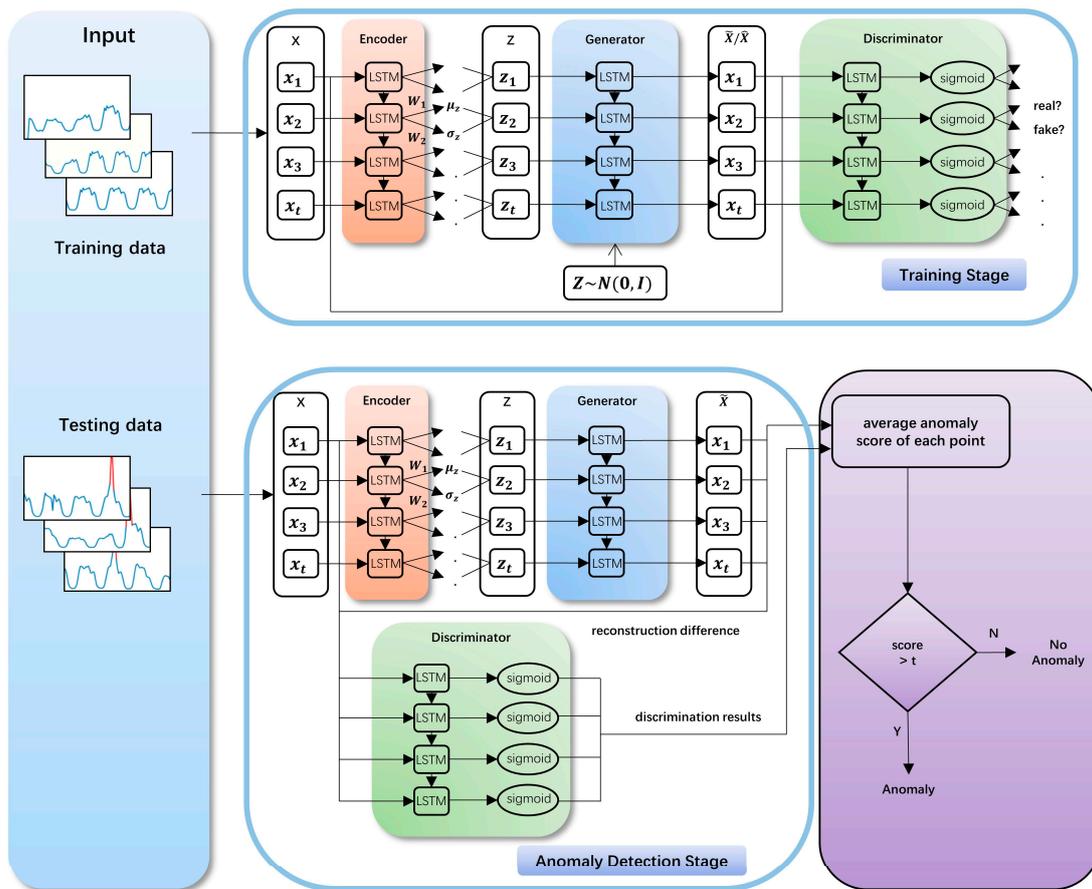


Figure 2. LSTM-based VAE-GAN architecture. μ_z and σ_z are obtained by the linear transformation of the encoder output. W_1 and W_2 are the coefficients of the linear transformation. $z = \mu_z + \sigma_z \odot \epsilon$, where $\epsilon \sim N(0, I)$ and \odot signify an element-wise product.

Our model is trained on normal data and learns the distribution of normal data. To make the VAE-GAN learn the temporal dependence of time series, we combine the VAE-GAN with LSTM by using LSTM as the encoder, the generator and the discriminator of VAE-GAN. We divide the time series into sub-sequences by a sliding window in a certain step size, which corresponds to the input variables. Each input sample to the encoder is a vector of a certain size denoting the sub-sequence which is encoded to the vector in the latent space. The generator uses the vector in the latent space to generate the vector in the real-time space. The discriminator outputs a vector which denotes whether the vector in the real-time space obeys the distribution of the normal training data.

At the training stage, the encoder approximates the posterior distribution $q(z|x)$ and encodes a data sample x to a latent representation z . The generator reconstructs \tilde{x} by decoding the latent representation back to data space. As the same time, a random variable \hat{z} is sampled from $p(z) = N(0, I)$, which is a standard normal distribution used for the prior and sent to the generator to generate \hat{x} . The LSTM of the discriminator is followed by a full connection with activation function sigmoid. With x , \tilde{x} and \hat{x} as inputs, the discriminator will learn to discriminate data by making \tilde{x} , \hat{x} close to 0, and x close to 1.

$$z \sim Enc(x) = q(z|x), \quad (1)$$

$$\tilde{x} \sim Gen(z) = p(x|z), \quad (2)$$

For time series, instead of VAE reconstruction error, we use a reconstruction error expressed in the discriminator and obtain better results [34].

The loss of VAE-GAN consists of three parts. For the encoder,

$$L_{enc} = L_{kl} + L_{re}, \quad (3)$$

with

$$L_{kl} = KL(q(z|x)||p(z)), \quad (4)$$

$$L_{re} = -E_{q(z|x)}[\log(p(Dis'(x)|z))], \quad (5)$$

where KL is the Kullback–Leibler divergence, L_{re} is the reconstruction of x by maximizing the log-likelihood $\log(Dis'(x)|z)$ with sampling from $q(z|x)$, and $Dis'(x)$ is the representation of the hidden LSTM in discriminator.

For the generator,

$$L_{gen} = -\log(Dis(\hat{x})) - \log(Dis(\tilde{x})) + L_{re}, \quad (6)$$

For the discriminator,

$$L_{dis} = -\log(1 - Dis(\hat{x})) - \log(1 - Dis(\tilde{x})) - \log(Dis(x)), \quad (7)$$

We trained the encoder, the generator and the discriminator in LSTM-based VAE-GAN using Adam optimizers with a 0.001 learning rate.

2.3. Anomaly Score

At the anomaly detection stage, the time series for anomaly detection is also divided into sub-sequences by a sliding window in the same step size as the training stage, which are input into the encoder. The encoder maps inputs into the latent space and sends their latent representation to the generator. The generator outputs reconstructed sub-sequences X_{test} . The discriminator outputs the possibility of inputs being normal.

The anomaly score utilizes the encoder, the generator, and the discriminator simultaneously trained in the model training phase, which is composed of reconstruction difference and discrimination results. Since anomalies do not conform to the distribution of normal data, their anomaly scores will be relatively high.

$$anomaly_{score} = (1 - \alpha)|X_{test} - \hat{X}_{test}| - \alpha Dis(X_{test}), \quad (8)$$

Due to the moving window mechanism, some points' anomaly scores are calculated many times, and some points' anomaly scores are calculated only once. For each point in time series, the anomaly detector needs to calculate its average anomaly score.

In addition, we used a small part of the test set containing anomalies to select the optimal threshold. This threshold can accurately distinguish the anomalies in this part, and then generalize to the entire test set.

2.4. Anomaly Detection Algorithm

Our method is divided into two stages, namely the model training stage and the anomaly detection stage. After the data preprocessing described above, we can obtain the normal training data and the testing data. The detailed algorithm flow is described in Algorithm 1.

Algorithm 1. Anomaly detection algorithm used the LSTM-based VAE-GAN

Input: training data X_{train} , testing data X_{test}

Output: anomaly or no anomaly

At training model stage:

Initialize Enc, Gen, Dis

In each iteration:

Generate random mini-batch X from training data X_{train}

Generate Z from encoder $Z = Enc(X)$

Generate \tilde{X} from generator $\tilde{X} = Gen(Z)$

Sample \hat{Z} from prior $p(Z) = N(0, I)$

Generate \hat{X} from generator $\hat{X} = Gen(\hat{Z})$

Update parameters of encoder according to gradient

$$\theta_{Enc} \leftarrow -_{Enc} \left\{ KL(q(Z|X)|p(Z)) - E_{q(Z|X)} \left[\log(p(Dis'(X)|Z)) \right] \right\}$$

Update parameters of generator according to gradient

$$\theta_{Gen} \leftarrow -_{Gen} \left\{ -\log(Dis(\hat{X})) - \log(Dis(\tilde{X})) - E_{q(Z|X)} \left[\log(p(Dis'(X)|Z)) \right] \right\}$$

Update parameters of discriminator according to gradient

$$\theta_{Dis} \leftarrow -_{Dis} \left\{ -\log(1 - Dis(\hat{X})) - \log(1 - Dis(\tilde{X})) - \log(Dis(X)) \right\}$$

At anomaly detection stage:

Calculate reconstruction difference: $Re = |X_{test} - Gen(Enc(X_{test}))|$

Calculate discrimination results: $Dis = Dis(X_{test})$

Calculate anomaly score: $score = |-\alpha Dis + (1 - \alpha) Re|$

Calculate average anomaly score for each point of time series corresponding to the testing data X_{test}

if (score > threshold):

return anomaly

else:

return no anomaly

3. Results

3.1. Comparison with Other Reconstruction Models in F1 Score

In the LSTM-based VAE-GAN, the LSTM networks for the encoder, the generator and the discriminator have the same size with depth 1 and 60 hidden units. In addition, we set the dimension of latent space as 10.

We use the Precision, Recall and F1 score to evaluate the anomaly detection performance of our model.

$$Precision = \frac{TP}{TP + FP} \quad (9)$$

$$Recall = \frac{TP}{TP + FN} \quad (10)$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (11)$$

where TP is the number of anomaly points correctly detected, FP is the number of normal points incorrectly identified as anomaly points, and FN is the number of anomaly points incorrectly identified as normal points.

To evaluate the performance of the proposed method, we implemented three baseline methods which are the representative time series anomaly detection methods based on sample reconstruction. They all perform anomaly detection through reconstruction difference.

- LSTM-AE: An anomaly detection method using an LSTM-based autoencoder [28].

- LSTM-VAE: A anomaly detector using a variational autoencoder. Unlike an AE, a VAE models the underlying probability distribution of observations using variational inference. The LSTM networks are used as the encoder and decoder [23].
- MAD-GAN: An anomaly detection method based on Generative Adversarial Networks which uses the LSTM networks as the generator and the discriminator [29].

Table 1 shows the best results of our method LSTM-based VAE-GAN and those representative time series anomaly detection methods based on sample reconstructions. LSTM-AE, LSTM-VAE, and MAD-GAN all use LSTM networks as the basic modules and their basic parameters are the same as those in LSTM-based VAE-GAN. In order to focus on comparing the ability of the model to distinguish between anomalies and normal points, we use the same threshold selection strategy described in this paper for all methods. As shown in Table 1, our method consistently outperforms the other time series anomaly detection methods based on sample reconstruction in F1 score.

Table 1. Precision, Recall and F1 score of representative time series anomaly detection methods based on sample reconstruction and our method LSTM-based VAE-GAN.

Dataset	Method	Precision	Recall	F1
Yahoo	LSTM-AE	0.4353	0.848	0.5753
	LSTM-VAE	0.8464	0.8516	0.849
	MAD-GAN	0.6007	0.8509	0.7042
	LSTM-based VAE-GAN	<u>0.8752</u>	<u>0.9067</u>	<u>0.8907</u>
KPI	LSTM-AE	0.9474	0.4737	0.6316
	LSTM-VAE	0.76	<u>0.5</u>	0.6032
	MAD-GAN	0.9444	0.4474	0.6071
	LSTM-based VAE-GAN	<u>0.95</u>	<u>0.5</u>	<u>0.6552</u>

3.2. Time Spent in the Anomaly Detection Stage

Compared with the time series anomaly detection method based on GAN, since LSTM-based VAE-GAN jointly trains the encoder, the generator and the discriminator at the training stage, it does not need to calculate the best mapping from real-time space to the latent space at the anomaly detection stage. The time required at the anomaly detection stage is greatly reduced, which can make the model detect anomalies quicker. We do the time loss experiment on the hardware environment of 2.10 GHz CPU (24 cores, x86 64 architecture), Ubuntu OS and RAM with 128 G. Figure 3 shows the time spent by four methods in Yahoo at the anomaly detection stage, respectively. As Figure 3 shows, during the anomaly detection stage, the time required by our model at each step size is much shorter than the time required by the method based on GAN. Compared with LSTM-AE and LSTM-VAE, LSTM-based VAE-GAN needs to calculate the discrimination results of input, so it takes a little longer than LSTM-AE and LSTM-VAE. In addition, because the number of samples decreases with increasing step size, the time required for both methods decreases as the step size increases.

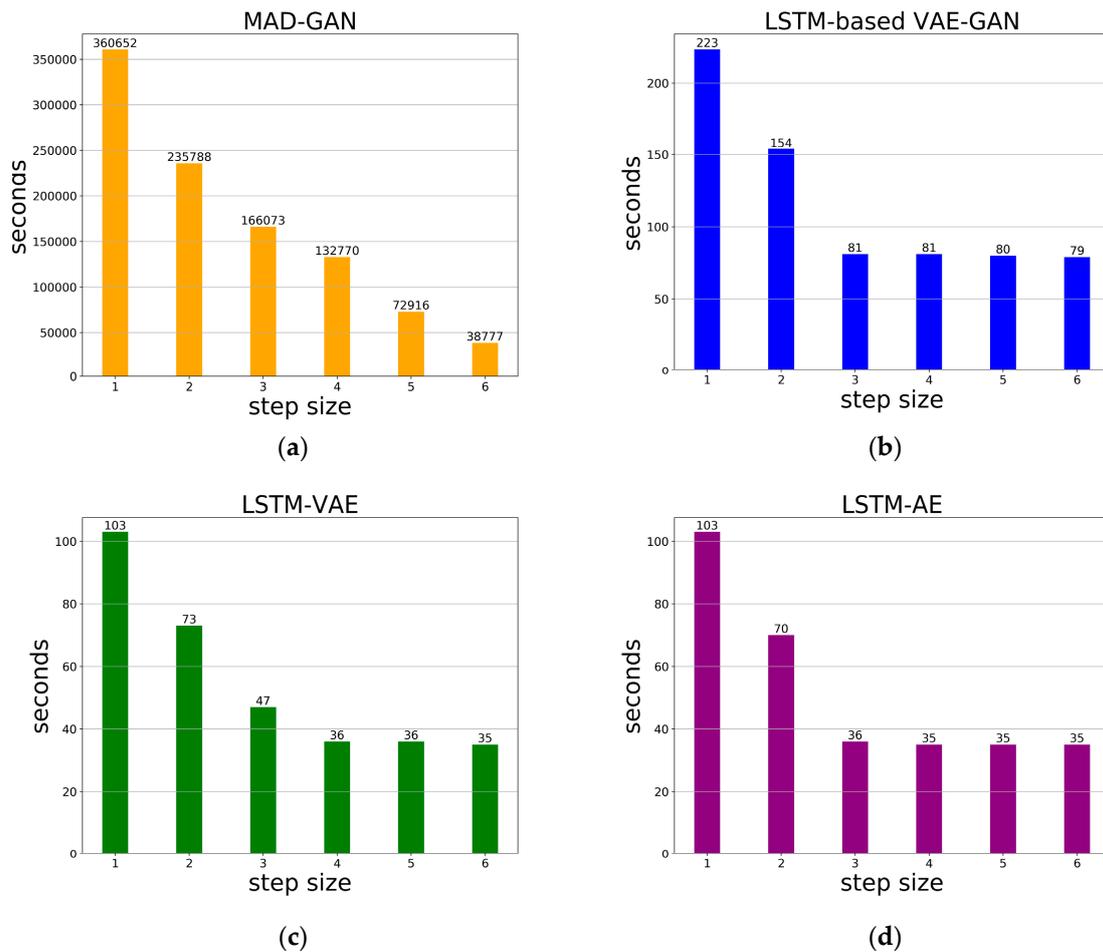


Figure 3. With different step sizes, the time spent by four methods at the anomaly detection stage in Yahoo dataset. (a) MAD-GAN, (b) LSTM-based VAE-GAN, (c) LSTM-VAE, (d) LSTM-AE.

3.3. The Impact of Latent Space's Dimensions

The latent space representation of our data contains all the important information needed to represent our original data point. This representation must then represent the features of the original data. The representation capability of latent space varies with the dimensions of latent space. We observe the effect of latent space's dimensions on the performance of the reconstruction-based models in time series anomaly detection. We set the dimensions of latent space to 5, 10, and 15, respectively. Table 2 describes the performance of LSTM-AE, LSTM-VAE, MAD-GAN, and LSTM-based VAE-GAN in different latent space's dimensions in Yahoo dataset.

Table 2. The experiment results in different latent space’s dimensions in Yahoo dataset.

Method	Latent Dim	Precision	Recall	F1
LSTM-AE	5	0.6095	0.7171	0.6589
	10	0.4353	0.848	0.5753
	15	0.4861	0.855	0.6198
LSTM-VAE	5	0.7513	0.872	0.8072
	10	0.8464	0.8516	0.849
	15	0.8281	0.8822	0.8543
MAD-GAN	5	0.6071	0.8434	0.706
	10	0.6007	0.8509	0.7042
	15	0.795	0.887	0.8385
LSTM-based VAE-GAN	5	0.9	0.8577	0.8784
	10	0.8752	0.9067	0.8907
	15	0.8698	0.9054	0.8873

3.4. Visual Analysis

The LSTM-based VAE-GAN was trained on normal data and learns the distribution of normal data. Since the anomaly samples do not obey the distribution of normal data, the generator cannot reconstruct them well when inputting anomaly samples to the encoder. In order to observe this intuitively, we draw the input time sub-sequences and the reconstructed time sub-sequences in Figure 4. It shows that the normal samples and reconstructed samples of them are roughly the same. When the input sample contains anomaly points as the red part in the figure, the reconstructed sample does not reproduce abnormal points, which provides the possibility for anomaly detection.

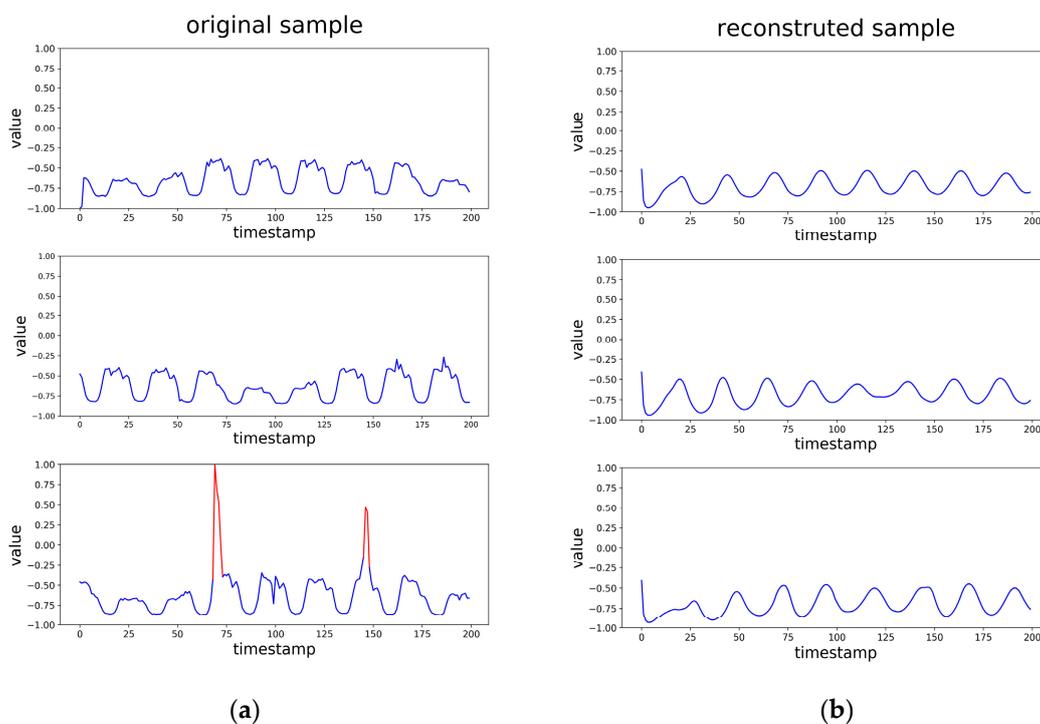


Figure 4. (a) The time subsequence needed to detect anomalies, and the red parts are anomalies. (b) The reconstructed time subsequence corresponding to the time subsequence in part a.

Figure 5 shows the anomaly score of the time series, which were outputted by our model. The red dotted line is the optimal threshold. It can be seen that the scores of normal points are mostly below the optimal threshold, and the scores of anomaly points are mostly above the optimal threshold. Since our reconstructed samples are relatively smooth as shown in Figure 4, the reconstruction differences that are part of the anomaly score make the anomaly score curves and the time series shape approximately the same.

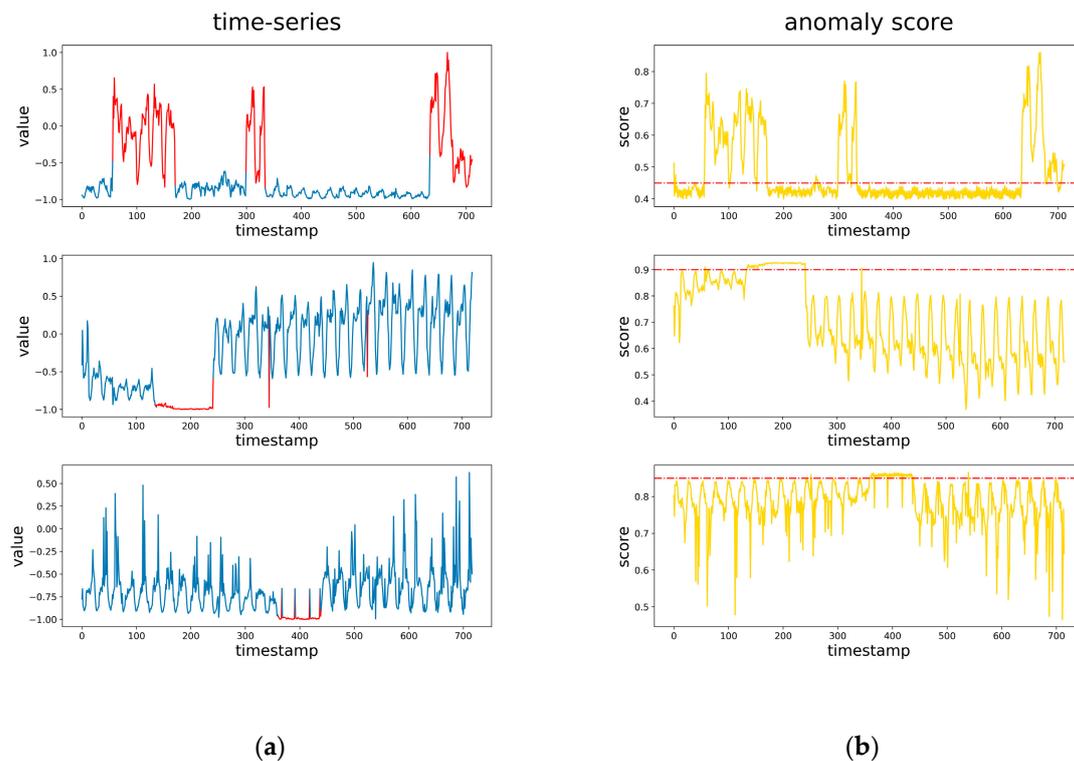


Figure 5. (a) The original time series containing anomalies, and the red parts are anomalies. (b) The anomaly score corresponding the time series in part a, and the dotted line is the optimal threshold.

4. Discussion

In this paper, a LSTM-based VAE-GAN anomaly detection method for time series is proposed. The method is designed to monitor the equipment states through the data collected in the form of time series.

The time series anomaly detection method based on sample reconstruction can be divided into two stages. One is the model training stage, where the model learns the distribution of normal data. The other is the anomaly detection stage, where the anomaly score of the time series is calculated to identify anomaly. The LSTM-based VAE-GAN jointly trains the encoder, the generator and the discriminator to take advantage of the mapping ability of the encoder and the discriminatory ability of the discriminator simultaneously. The optimization process at the anomaly detection stage is avoided so that anomalies can be detected more quickly and more accurately. In experiments based on Yahoo and KPI time series data, our method has a higher F1 value than several classic sample-reconstruction based time series anomaly detection methods. In the time loss comparison with GAN, our method is shown to spend less time due to avoiding the optimization process at the anomaly detection stage. Due to the moving window mechanisms, some points' anomaly scores are calculated many times, the others are calculated only once. The accuracy is not influenced by the number of calculations of the anomaly scores at the anomaly detection stage. In fact, the moving window mechanism is not essential in the data preprocessing. It depends on the length of the time series. For increasing the number of subsequences used to train the model at the training stage, we set the step size smaller than

subsequence length. If the length of the time series is long enough, the time series can be divided at the same interval.

Although our method can accurately and quickly detect anomalies in time series, there are still some limitations. In our paper, anomalies in time series refer to anomaly points, and the anomaly score module is designed for this background. In some application scenarios where anomalies in time series may be successive anomaly subsequences, anomaly subsequence can be detected if some points in it are detected by the model. A new design of the anomaly score module is needed to meet the application scenarios.

Our research has room for further development. In the current situation, our method needs to accumulate certain data to adjust the threshold of the anomaly score. The next enhancement of this method is to provide an adaptive threshold adjustment method for quick use.

Author Contributions: Conceptualization, design, and experiments, Z.N., K.Y.; administrative support: K.Y., X.W.; manuscript writing and final approval of manuscript, all authors. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61601046 and Grant 61171098, in part by the 111 Project of China under Grant B08004, in part by the EU FP7 IRSES Mobile Cloud Project under Grant 270 12212.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2233–2243.
2. Marjani, M.; Nasaruddin, F.; Gani, A.; Karim, A.; Hashem, I.A.T.; Siddiqua, A.; Yaqoob, I. Big IoT data analytics: Architecture, opportunities, and open research challenges. *IEEE Access* **2017**, *5*, 5247–5261.
3. Martí, L.; Sanchez-Pi, N.; Molina, J.M.; Garcia, A.C.B. Anomaly detection based on sensor data in petroleum industry applications. *Sensors* **2015**, *15*, 2774–2797. [[CrossRef](#)] [[PubMed](#)]
4. Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. *Acm J.* **2009**, *41*, 1–58. [[CrossRef](#)]
5. Gupta, M.; Gao, J.; Aggarwal, C.C.; Han, J.; Engineering, D. Outlier detection for temporal data: A survey. *IEEE Trans. Knowl. Data Eng.* **2014**, *26*, 2250–2267. [[CrossRef](#)]
6. Pham, D.-S.; Venkatesh, S.; Lazarescu, M.; Budhaditya, S.J.D.M.; Discovery, K. Anomaly detection in large-scale data stream networks. *Data Min. Knowl. Discov.* **2014**, *28*, 145–189. [[CrossRef](#)]
7. Aggarwal, C.C.; Subbian, K. Event detection in social streams. In Proceedings of the 2012 SIAM International Conference on Data Mining, Anaheim, CA, USA, 26–28 April 2012; pp. 624–635.
8. Aggarwal, C.C.; Yu, P.S. Outlier detection for high dimensional data. In Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, CA, USA, May 2001; pp. 37–46.
9. Aggarwal, C.C.; Yu, P.S. Outlier detection with uncertain data. In Proceedings of the 2008 SIAM International Conference on Data Mining, Atlanta, GA, USA, 24–26 April 2008; pp. 483–493.
10. Aggarwal, C.C. On abnormality detection in spuriously populated data streams. In Proceedings of the 2005 Siam International Conference on Data Mining, Newport Beach, CA, USA, 21–23 April 2005; pp. 80–91.
11. Aggarwal, C.C.; Zhao, Y.; Philip, S.Y. Outlier detection in graph streams. In Proceedings of the 2011 IEEE 27th International Conference on Data Engineering, Hannover, Germany, 11–16 April 2011; pp. 399–409.
12. Gao, J.; Liang, F.; Fan, W.; Wang, C.; Sun, Y.; Han, J. On community outliers and their efficient detection in information networks. In Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, July 2010; pp. 813–822. [[CrossRef](#)]
13. Chalapathy, R.; Chawla, S. Deep learning for anomaly detection: A survey. *arXiv* **2019**, arXiv:1901.03407.
14. Pincus, R.J.B.J.; Barnett, V.; Lewis, T. *Outliers in Statistical Data*; J. Wiley & Sons: Hoboken, NJ, USA, 1994; pp. 199–256.
15. Contreras, J.; Espinola, R.; Nogales, F.J.; Conejo, A.J. ARIMA models to predict next-day electricity prices. *IEEE Trans. Power Syst.* **2003**, *18*, 1014–1020. [[CrossRef](#)]

16. Rousseeuw, P.J.; Leroy, A.M. *Robust Regression and Outlier Detection*; John Wiley & Sons: Hoboken, NJ, USA, 2005.
17. Qin, Y.; Song, D.; Chen, H.; Cheng, W.; Jiang, G.; Cottrell, G. A dual-stage attention-based recurrent neural network for time series prediction. In Proceedings of the 26th International Joint Conference on Artificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 2627–2633.
18. Wu, X.; Shi, B.; Dong, Y.; Huang, C.; Faust, L.; Chawla, N.V. RESTful: Resolution-Aware Forecasting of Behavioral Time Series Data. In Proceedings of the Conference on Information and Knowledge Management, Torino, Italy, 22–26 October 2018; pp. 1073–1082.
19. Bontemps, L.; McDermott, J.; Le-Khac, N.-A. Collective anomaly detection based on long short-term memory recurrent neural networks. In Proceedings of the International Conference on Future Data and Security Engineering, Can Tho City, Vietnam, 23–25 November 2016; pp. 141–152.
20. 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 International Conference on Knowledge Discovery & Data Mining, London, UK, August 2018; pp. 387–395. [[CrossRef](#)]
21. Chauhan, S.; Vig, L. Anomaly detection in ECG time signals via deep long short-term memory networks. In Proceedings of the 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA), Paris, France, 19–21 October 2015; pp. 1–7.
22. Malhotra, P.; Vig, L.; Shroff, G.; Agarwal, P. Long short term memory networks for anomaly detection in time series. In Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Bruges, Belgium, 22–24 April 2015; pp. 89–94.
23. Park, D.; Hoshi, Y.; Kemp, C.C. A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder. *IEEE Robot. Autom. Lett.* **2018**, *3*, 1544–1551. [[CrossRef](#)]
24. Zhang, C.; Song, D.; Chen, Y.; Feng, X.; Lumezanu, C.; Cheng, W.; Ni, J.; Zong, B.; Chen, H.; Chawla, N.V. A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. In Proceedings of the AAAI Conference on Artificial Intelligence, Palo Alto, CA, USA, 27 January–1 February 2019; pp. 1409–1416.
25. Zong, B.; Song, Q.; Min, M.R.; Cheng, W.; Lumezanu, C.; Cho, D.; Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In Proceedings of the 6th International Conference on Learning Representations, Vancouver, BC, Canada, 30 April–3 May 2018.
26. Guo, Y.; Liao, W.; Wang, Q.; Yu, L.; Ji, T.; Li, P. Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach. In Proceedings of the Asian Conference on Machine Learning, Beijing, China, 14–16 November 2018; pp. 97–112.
27. Chen, R.-Q.; Shi, G.-H.; Zhao, W.-L.; Liang, C.-H. Sequential VAE-LSTM for Anomaly Detection on Time Series. *arXiv* **2019**, arXiv:1910.03818.
28. Malhotra, P.; Ramakrishnan, A.; Anand, G.; Vig, L.; Agarwal, P.; Shroff, G. LSTM-based encoder-decoder for multi-sensor anomaly detection. *arXiv* **2016**, arXiv:1607.00148.
29. Li, D.; Chen, D.; Goh, J.; Ng, S.-K. MAD-GAN: Multivariate Anomaly Detection for Time Series Data with Generative Adversarial Networks. In Proceedings of the 28th International Conference on Artificial Neural Networks, Munich, Germany, 17–19 September 2019; pp. 703–716.
30. Goodfellow, I. NIPS 2016 tutorial: Generative adversarial networks. *arXiv* **2016**, arXiv:1701.00160.
31. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. In Proceedings of the 2nd International Conference on Learning Representations, Banff, AB, Canada, 14–16 April 2014.
32. Rezende, D.J.; Mohamed, S.; Wierstra, D. Stochastic backpropagation and approximate inference in deep generative models. In Proceedings of the 31st International Conference on Machine Learning, Beijing, China, 21–26 June 2014; pp. 3057–3070.
33. Goodfellow, I. On distinguishability criteria for estimating generative models. In Proceedings of the 3rd International Conference on Learning Representations, San Diego, CA, USA, 7–9 May 2015.
34. Larsen, A.B.L.; Sønderby, S.K.; Larochelle, H.; Winther, O. Autoencoding beyond pixels using a learned similarity metric. In Proceedings of the 33rd International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 2341–2349.

35. Yahoo Webscope Dataset S5-A Labeled Anomaly Detection Dataset. Available online: <https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70> (accessed on 8 December 2019).
36. AIOps Challenge. Available online: http://iops.ai/competition_detail/?competition_id=5&flag=1 (accessed on 4 June 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).