Luo *et al.*
EURASIP Journal on Advances in Signal Processing (2022) 2022:112
https://doi.org/10.1186/s13634-022-00943-7

EURASIP Journal on Advances in Signal Processing

RESEARCH Open Access

Anomaly detection by using a combination of generative adversarial networks and convolutional autoencoders

Xukang Luo, Ying Jiang*, Enqiang Wang and Xinlei Men

*Correspondence: 006907@xju.edu.cn

Department of Software Engineering, Xinjiang University, Ürümqi, China

Abstract

With the development of full digitalization, the amount of time series data generated by sensors is ever-increasing; thus, time series outlier detection has become crucial. Moreover, in practice, discovering and flagging anomalies is very time-consuming and expensive. To solve this problem, unsupervised anomaly detection methods have often been used in the past, in which the model is trained with normal data to learn its behavioral patterns. Generative adversarial networks (GANs) can simulate complex and high-dimensional distributions of data and can be used to learn the behavioral patterns of normal data for unsupervised anomaly detection. However, because of the problem of convergence, GANs are difficult to train. Thus, USADs (an unsupervised anomaly detection model) utilize an autoencoder (AE) to undertake the task of the generator and discriminator and enhance the stability during adversarial training by using the AE to alleviate the problem of non-convergence encountered in GANs. Therefore, in this study, we used the USAD's generative adversarial training architecture combined with convolutional AEs to improve the model's feature extraction capabilities. In addition, to reduce false-positive outcomes caused by the prominent sharp points in the reconstructed data, we used the exponential weighted moving average method to smooth the reconstruction error, thereby improving the anomaly detection accuracy of the model. Finally, we experimented with real-world time-series data (ECG and 2D gesture) and verified that our approach could improve accuracy. Compared to the best in the comparison method, our model improved by 0.028% in AUROC, 0.233% in AUPRC, and 0.187% in F1 on average.

Keywords: Autoencoder, Generative adversarial network, Time series, Anomaly detection

1 Introduction

With the continuous development of information technology, more and more sensors and monitoring devices are being used in various fields, such as industrial applications [1], aerospace [2], medicine [3], and financial transactions [4, 5]. Moreover, sensors generate large amounts of time series data in monitoring production processes [6, 7]; thus, the exploration and application of time series data generated in production practice have become an important research topic. Among them, anomaly detection has become one

© The Author(s) 2022. **Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

of the main tasks of time series data mining, and observations that do not conform to the expected behavior in the time series data are called outliers. Time-series anomaly detection is critical to ensure industrial equipment's availability, reliability, and safety [8]. Anomaly detection has been studied in various applications, such as credit card fraud detection, cyber-security intrusion detection, and troubleshooting of industrial processes.

However, detecting outliers in time series data is challenging, and time series data often have complex nonlinear, and high-dimensional dynamics that are difficult to model. Anomaly detection problems can be considered supervised binary classification problems in which the deep model can automatically extract features and learn hidden variables with sufficient labeled data; this method can achieve a high accuracy rate [9]. However, labeled data are often difficult to obtain in practice. Also, it is difficult to identify anomalies never seen before. There are numerous types of anomalies, and when new types of anomalies appear, the supervised detection accuracy does not work properly.

Finding and flagging anomalies in practice is very time-consuming and expensive. To alleviate this problem, unsupervised anomaly detection methods are often employed. Time series anomaly detection is often classified as a class of problems [10, 11] where the training set only contains normal samples. Such anomaly detection techniques can be broadly divided into prediction-based and reconstruction-based methods. Prediction-based methods [12, 13] predict the normal value of an indicator based on historical data and detect anomalies based on prediction errors. Common prediction models include the classical autoregressive moving average and autoregressive integrated moving average models [14], deep learning-based recurrent neural networks (RNNs) based on deep learning [15], and long short-term memory network predictor (LSTM). However, this approach is not suitable for predicting indicators in certain complex systems [12]. Reconstruction-based methods learn a compressed representation of the core statistical structure of normal data and then use it to reconstruct time series and detect anomalies based on reconstruction errors. Reconstruction-based approaches typically use auto-encoders (AEs) [13], representing more complex time series patterns by applying nonlinear functions for reconstruction and anomaly detection. However, such reconstructions can lead to overfitting without proper regularization, which results in low precision[10].

In addition, because generative adversarial networks (GANs) can generate quasi-real synthetic data through joint learning using generators and discriminators [16], complex and high-dimensional real-world data distributions can be simulated. This feature has been used successfully for anomaly detection. The detection of anomalies using GAN is the task of modeling normal behavior and detecting anomalies to measure anomaly scores using an adversarial training process [17]. However, due to problems such as mods and non-convergence [18], GANs are difficult to train. To solve the non-convergence problem of GANs, USADs [19] use two AEs to perform the tasks of the generator and discriminator, thus combining the advantages of AEs and adversarial training while compensating for the limitations of each technique. In this study, the USADs' adversarial generative training architecture was combined with a convolutional AE (CAE) to improve the ability to extract features from the model, thus improving the accuracy of anomaly detection. Furthermore, to reduce false positive outcomes caused by prominent spikes in the reconstructed data, we used the exponential weighted moving average

(EWMA) method to smooth the reconstruction error and further improve the model's accuracy. The main contributions of this study are as follows.

(1) The USAD's adversarial generative training architecture was combined with the CAE to improve the model's anomaly detection accuracy. The reconstruction error was smoothed to suppress the error spikes in the reconstruction data and reduce false positive results.
(2) The performance of the proposed model was compared with that of five anomaly detection models in terms of AURORAC, AUPRC, and F1. The results revealed that the proposed model is superior to the other anomaly detection models in terms of all three aforementioned indicators.
(3) Experiments with ECG and 2D gesture datasets were conducted to demonstrate the versatility of the proposed model.

2 Related work

The reconstruction-based approach focuses on reducing expected reconstruction errors and consists of two parts: refactoring model optimization and reconstruction-based anomaly scoring. The optimization goal of learning to rebuild the model can be expressed as follows.

$$L = \|X - G(X)\|_2 = \sum_x \|x - G(x)\|_2 \tag{1}$$

where X is the training data, and $G(X)$ is the model for reconstructing the results. The training goal is to narrow the gap between the reconstructed and the training data. X is made up of multiple x vectors, $A(x)$. The abnormal score of x can be calculated as follows:

$$A(x) = \|x - G(x)\|_2 \tag{2}$$

Many refactoring-based anomaly detection methods can be formalized as training targets [18]and use anomaly scoring [19]. According to the models used, the anomaly detection methods based on refactoring can be divided into the following two categories:

2.1 Autoencoders

AEs [20] are often used for anomaly detection by learning to reconstruct a given input. The model is trained using normal data; thus, once the input has been rebuilt, the instance is considered abnormal if the output does not match the input of normal data. The LSTM encoder-decoder model [21] is used to learn the time representation of time series through the LSTM network and uses reconstruction error to detect anomalies. Despite its effectiveness, LSTM cannot capture spatial features. The CAE [22] is an important method for video anomaly detection and can capture 2D image structures because its weights are shared between all locations in the input image. Convolutional LSTM (ConvLSTM) combines the characteristics of LSTM and the convolutional neural network (CNN) to simulate spatiotemporal correlation by using convolutional layers instead of fully connected layers. Thus, in the current study, the ConvLSTM layer

was added to the AE [23] to encode normal data more efficiently. Other AEs, such as variational AEs [24], denoising AEs [25], and deep faith networks [26], also show good performance.

2.2 Generative adversarial networks

Recently, the GAN framework was proposed to build a generative deep learning (DL) model through adversarial training [8]. While GANs are efficient in image-processing tasks, such as generating realistic images, with growing interest in GANs, researchers have proposed anomaly detection by using adversarial training. AnoGAN [27] and Ganomaly [28] have been proposed to detect anomalies in visual data. Furthermore, there are many methods for detecting anomalies in time series data, such as MAD-GAN [29], which uses the same LSTM as the generator and discriminator, CNN-based AEs, and BeatGAN [10].

3 Methods

In this study, the time-series data were split into independent samples to use DL models for anomaly detection. The adversarial generative architecture was then used to learn the high-dimensional distribution of the normal data. AEs were used to make adversarial training more stable, reconstruct the data, and smooth the anomaly detection results to reduce false positives. The process can be divided into training, anomaly detection, and smoothing.

3.1 Time-series anomaly detection

A time series $T = [s_1, s_2, \ldots, s_n]$ is an observation within n time steps, where each value $s_t \in R^d$ is a d-dimensional vector. If $d = 1$ then T is a univariate time series; if $d > 1$, then T is a multidimensional time series. For reconstruction-based anomaly detection methods, let $E = [e_1, e_2, \ldots, e_n]$ be the reconstruction error. e_t is a constant that represents the reconstruction error at time step t; the higher the value of e_t, the more likely s_t is to be an outlier. In practice, a threshold is usually set, and when e exceeds the threshold, the current value is judged as an outlier. Because time-series data are unbounded data, to facilitate model processing, we used a sliding window with length T_w and step T_s to sample the data and produce bounded data. The data sampled by the sliding window is called X; $X = [x_1, x_2 \ldots x_m]$, where x_t is a $d \times T_w$ matrix, representing a training sample. In reconstruction-based anomaly detection, reconstruction data from model output were recorded as $X' = [x'_1, x'_2, \ldots, x'_m]$. x'_2 is also a $d \times T_w$ dimensional matrix. As long as one of the vectors had a reconstruction error that exceeded the threshold, the sample was considered an anomaly.

3.2 GAN model

GAN is a framework for establishing generative models through an adversarial process by using two models, namely discriminator D and generator G. The Generator G is designed to learn the distribution of the data, while the discriminator D is used to distinguish whether a sample is real data or data generated by G. To learn the distribution of the data x, the generator establishes a mapping from the noise distribution p_z to the data space $G(z; \theta_G)$, where θ_G is the generator parameter. The discriminator outputs

a single scalar that represents the probability x that the given sample is real data rather than generated data.

The original GAN framework [16] treats this problem as a minimum game in which two participants (G and D) compete against each other to play the following minimum zero-sum game:

$$\min_G \max_C V(D,G) = E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{3}$$

Training network D distinguishes between the training sample and the generated sample (maximized $\log D(x)$ and $\log(1 - D(G(z)))$), and the training network G minimizes $\log(1 - D(G(z)))$, that is, maximizes the loss of D. During the training process, one side is fixed, the parameters of the other network are updated, and iterations are alternated so that the error of the other party is maximized. Finally, G estimates the distribution of the sample data, that is, whether the generated sample is more realistic. However, due to the imbalance between the generator and the discriminator, GAN training is often difficult to converge. Therefore, the use of the AE and GAN combination scheme is more common [10, 19]. On the one hand, GANs, being able to learn the data distribution pattern, overcome the inherent flaws of AEs. On the other hand, AEs can enhance stability during adversarial training, and thus alleviate the non-convergence problem encountered in GANs.

3.3 Model structure

The model consists of three parts: an encoder network, encoder, and two decoder networks, Decoder1 and Decoder2. As shown in Fig. 1, these three elements make up two AEs, AE1 and AE2, and the two AEs use the same encoder. We used the same CNN with a one-dimensional convolutional kernel that slides along the time

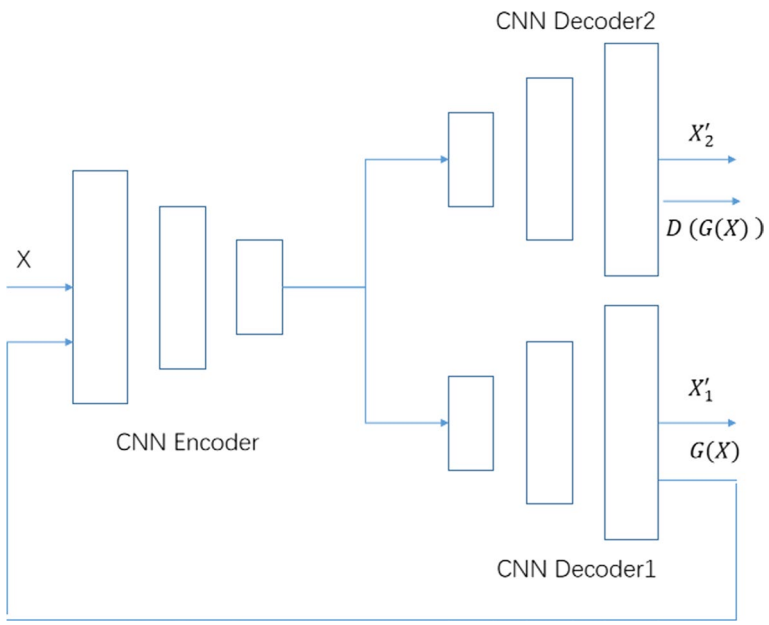

Fig. 1 Model structure

dimension for the encoder and decoder networks. For time series, CNNs are more robust than LSTMs [9]. In addition, by adjusting the sensory field of the CNN, the long-term correlation can be captured as in LSTMs.

3.4 Procedure

The process can be divided into three parts: training the model, using the model for anomaly detection, and smoothing out the anomaly detection results.

3.4.1 Training

Training can be divided into two phases. First, the two AEs, AE1 and AE2, are trained to learn to reconstruct normal input data. The two AEs are then trained in an adversarial manner so that AE2 cannot distinguish whether the input data are real data or data generated by AE1.

(i) *AE training* At this stage, the two AEs are trained to reconstruct the training data, and the input data is X mapped by the encoder to the potential space h. Then, the two decoders are used to reconstruct them into X_1' and X_2', respectively. To make the reconstructed data as consistent with the original data as possible, we minimized the loss functions (Eqs. (4) and (5)); $\left\|X - X_1'\right\|_2$ represents the distance between X and X_1'.

$$l_{AE1} = \left\|X - X_1'\right\|_2 \tag{4}$$

$$l_{AE2} = \left\|X - X_2'\right\|_2 \tag{5}$$

(ii) *Adversarial training* AE2 is trained to distinguish between real data and data generated by AE1, and AE1 is trained so that reconstructed data cannot be differentiated by AE2. The data reconstructed by AE1 is encoded again by the encoder; then, the output of the encoder is reconstructed by AE2. The goal of AE1 is to minimize the difference between X and the outputs of AE2. The goal of AE2 is to maximize this difference. In Eq. (6), $G(X)$ represents the output of AE1 at this stage, and $D(G(X))$ represents the output of AE1:

$$\min_{AE1} \max_{AE2} = \|X - D(G(X))\|_2 \tag{6}$$

3.4.2 Anomaly detection

During the anomaly detection phase, the anomaly score is defined as a linear combination of the reconstruction errors of the two AEs:

$$E(X) = \alpha l_{AE1} + \beta l_{AE2} = \alpha\left\|X - X_1'\right\|_2 + \beta\left\|X - X_2'\right\|_2 \tag{7}$$

The parameters we used in the GAN and AEs are shown in the following tables (Table 1 and Table 2):

Table 1 Parameters and modules of the encoder

Encoder
Conv1d input:2, output:32, kernel_size:4, stride:2, padding:1, bias:False
LeakyReLU negative_slope:0.2, inplace:True
Conv1d input:32, output:64, kernel_size:4, stride:2, padding:1, bias:False
BatchNorm1d
LeakyReLU negative_slope:0.2, inplace:True
Conv1d input:64, output:128, kernel_size:4, stride:2, padding:1, bias:False
BatchNorm1d
LeakyReLU negative_slope:0.2, inplace:True

Table 2 Parameters and modules of the decoder

Decoder
BatchNorm1d
ReLU
ConvTranspose1d input:128, output:64, kernel_size:4, stride:2, padding:1, bias:False
BatchNorm1d
ReLU
ConvTranspose1d input:64, output:32, kernel_size:4, stride:2, padding:1, bias:False
BatchNorm1d
ReLU
ConvTranspose1d input:32, output:2, kernel_size:4, stride:2, padding:1, bias:False
Tanh

3.4.3 Error smoothing

Smoothing out the anomaly detection results: Scores $E(X)$ are smoothed to suppress error spikes in the reconstructed data. Some values in the normal data differ greatly from the surrounding data, leading to sharp spikes in the abnormal score [30]. We use the EWMA method to smooth such errors, and the smoothed anomaly score was noted as $E_s(X)$. To assess whether the reconstructed values are normal, we set a threshold for the smoothing error, and values with smoothing errors that exceeded the threshold were classified as abnormal.

4 Datasets and indicators

4.1 Datasets

We performed experiments with three commonly used real-world time-series datasets ECG and 2D gesture [31]:

1. ECG: This is a collection of data sets that contain abnormal heartbeats detected from ECG readings. We selected two datasets from it.
2. 2D gesture: This contains the time series of the X-Y coordinates of the actor's right hand. The data were extracted from a video in which the actor took a gun from the holster, moved it to the target position, and then put it back in. The abnormal area is the area where the actor has not put his gun back in the holster.

Both the ECG and the 2D gesture are two-dimensional time series data ($d = 2$). Common datasets include training sets (which contain only normal data) and test sets. We used 30% of the training set for validation and the rest for actual training. The model with the lowest rebuild loss in the validation set was evaluated. The time series was divided into sequences of length T_w according to the sliding window. The sliding window length was set at 320 and 80 on the ECG and 2D gesture datasets, respectively.

4.2 Evaluation indicators

Performance metrics such as precision and recall depend on the threshold of the abnormal score. To avoid setting this threshold, we used the following metrics that are widely used in anomaly detection:

(1) Area under the receiver operating characteristic curve (AUROC): As shown in Fig. 2, AUROC is a metric used to measure the performance of a classifier. AUROC has a value between 0 and 1. When the AUROC value is close to 1, the classifier can better classify positive and negative samples.

(2) Area under the precision recall curve (AUPRC) and the precision recall curve (Fig. 3), with recall on the x-axis and precision on the y-axis: Precision indicates the proportion of the actual positive sample to the predicted positive sample, while recall indicates the proportion of the original positive samples that were correctly predicted to be positive. The calculation formulas are as follows:

$$Recall = \frac{TP}{TP + FN} \tag{8}$$

$$Precision = \frac{TP}{TP + FP} \tag{9}$$

where TP indicates true positive (the sample prediction is positive, and the actual is also positive), FP indicates false positive (the sample prediction is positive, and the actual is negative), and FN indicates false negative (the sample prediction is negative, and the actual is positive).

(3) The F1 score considers both the precision and the recall of the classification model. The F1 score can be seen as a weighted average of the accuracy and recall of the model. The highest F1 score was chosen from all samples using 1000 thresholds, as

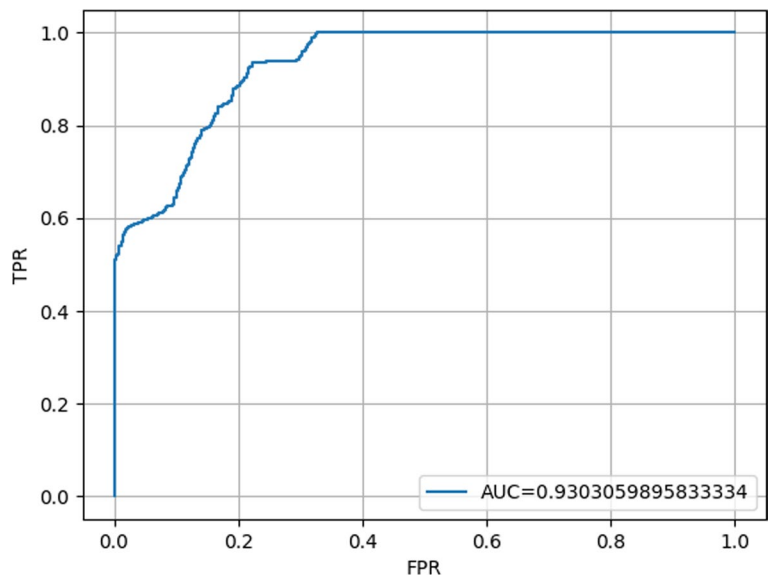

Fig. 2 ROC curve

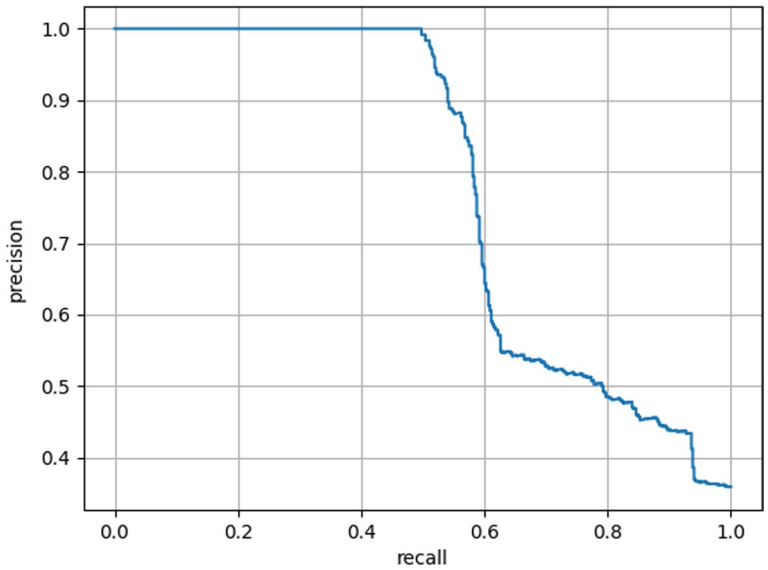

Fig. 3 Precision recall curve

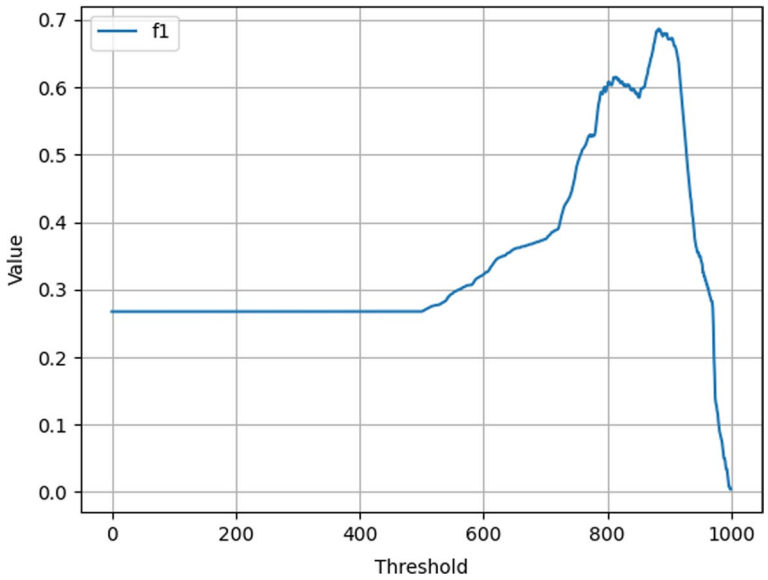

Fig. 4 F1-threshold curve

shown in Fig. 4, for all samples in the test set. The anomaly score was chosen evenly from 0 to the maximum value.

$$F1 = \frac{2 * Recall * Precision}{Recall + Precision} \tag{10}$$

5 Results and discussion

We compared our model with five recent anomaly detection algorithms: (1) recursive AE (RAE) [32]; (2) recurrent reconstructive network (RRN) [33], which combines attention, jump conversion, and force regularization; (3) recursive AE integration (RAEensemble) [34], which uses RNN integration with sparse hop connections as encoders and decoders; and (4) BeatGAN [10], a recent CNN AE-based GAN developed for time-series anomaly detection; (5)USAD [19]. The experimental results are shown in Table 3.

Experimental comparison analysis revealed that the proposed model produces better detection performance than recent anomaly detection algorithms, and the model achieved higher AUROC, AUPRC, and F1 scores for all three datasets, indicating that the model can achieve superior detection performance.

Finally, to suppress error spikes in the reconstructed data, we used the EWMA method to smooth out these errors. We used the ECG1 dataset to illustrate the effect of the smoothing error, as shown in Fig. 5. The upper panel shows the unshared reconstruction error, and the lower panel shows the smoothing effect. Before the smoothing process, the normal data area exhibits a higher error score, resulting in false positives. The performance metrics before and after smoothing are presented in Table 4, and the AURORAC, AUPRC, and F1 scores of the EWMA-processed model are shown; both are higher than those that are not smoothed.

Our method can achieve better results, but the threshold value to achieve the best detection effect still needs to be set manually. In practical, especially in real-time anomaly detection scenarios, we can't update the threshold automatically. This means that the system can't always be in top shape. And this is the next problem we need to solve.

Table 3 Comparative experiments

Metric	Method	ECG1	ECG2	2D gesture
AUROC	BeatGAN	0.7056	0.7329	0.7256
	RAE	0.7502	0.8289	0.7601
	RRN	0.7623	0.7405	0.7530
	RAE-ensemble	0.7788	0.8570	0.7808
	USAD	0.7223	0.4845	0.4852
	Our method	0.8058	0.8904	0.8048
	BeatGAN	0.4101	0.2254	0.4952
	RAE	0.4249	0.4996	0.4979
AUPRC	RRN	0.5653	0.4139	0.4866
	RAE-ensemble	0.4769	0.5256	0.5287
	USAD	0.3470	0.2289	0.2281
	Our method	0.8669	0.5942	0.7697
	BeatGAN	0.4204	0.2931	0.4941
	RAE	0.4736	0.5046	0.5300
F1	RRN	0.5502	0.4537	0.5240
	RAE-ensemble	0.5016	0.5333	0.5511
	USAD	0.3633	0.4124	0.4119
	Our method	0.7635	0.6539	0.7314

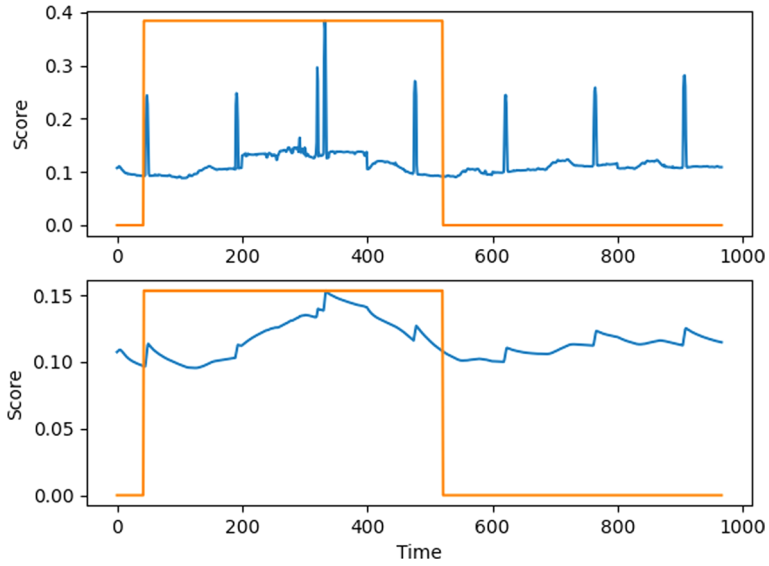

Fig. 5 Anomaly score smoothing

Table 4 Experiments to compare smoothing performance

	AUROC	AUPRC	F1
Without EWMA	0.70139	0.74361	0.75755
With EWMA	0.80576	0.86694	0.76345

6 Conclusions

To improve the accuracy of anomaly detection, in this study, a detection model was proposed that combines the USAD generative adversarial training architecture and CAE to enhance stability during adversarial training by generating the distribution of normal data for adversarial training and improving the ability to extract characteristics of the model. Finally, the EWMA method was used to suppress the error score spikes of the reconstructed data smoothly. The experimental results revealed that the proposed model is superior to other methods in terms of detection accuracy and does not exhibit widely differing results on different datasets, thus indicating that the proposed model is highly versatile.

Abbreviations

GAN	Generative adversarial networks
USADs	Unsupervised anomaly detection
AE	Autoencoder
ECG	Electrocardiogram
RNN	Recurrent neural networks
LSTM	Long short-term memory network
CAE	Convolutional autoencoder
EWMA	Exponential weighted moving average
CNN	Convolutional neural network
DL	Deep learning
AUPRC	Area under the precision-recall curve
AUROC	Area under the receiver operating characteristic curve
RAE	Recursive autoencoder

Acknowledgements
Not applicable

Author contributions
X.Luo is the experimental designer and executor of this work and was involved in completing data analysis and writing the first draft of the paper; E.Wang and X.Men participated in experimental design. E.Wang was involved in analysis of experimental results and revision of papers; Y.Jiang was person in charge of the project and was involved in guiding the design of experiments and revision of papers. All authors read and approved the final manuscript.

Funding
Xinjiang University Graduate Student Case Bank Construction

Availability of data and materials
ECG and 2D-gesture dataset are open source and from http://www. cs.ucr.edu/~eamonn/discords/

Declarations

Ethics approval and consent to participate
Not applicable.

Consent for publication
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Received: 12 June 2022 Accepted: 1 November 2022
Published online: 22 November 2022

References

1. D.Y. Oh, I. Yun, Residual error based anomaly detection using auto-encoder in smd machine sound. Sensors **18**, 1308 (2018). https://doi.org/10.3390/s18051308
2. S. Fuertes, G. Picart, J.-Y. Tourneret, L. Chaâri, A. Ferrari, C. Richard, Improving spacecraft health monitoring with automatic anomaly detection techniques. (2016)
3. Y. Hagiwara, H. Fujita, S.L. Oh, J.H. Tan, R.S. Tan, E.J. Ciaccio, U.R. Acharya, Computer-aided diagnosis of atrial fibrillation based on ECG signals: a review. Inf. Sci. **467**, 99–114 (2018). https://doi.org/10.1016/j.ins.2018.07.063
4. A. Grane, H. Veiga, Wavelet-based detection of outliers in financial time series. Comput. Stat. Data Anal. **54**, 2580 (2010). https://doi.org/10.1016/j.csda.2009.12.010
5. A. Siffer, P.-A. Fouque, A. Termier, C. Largouet, Anomaly detection in streams with extreme value theory. in Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining. KDD '17, (Association for Computing Machinery, New York 2017), pp. 1067–1075. https://doi.org/10.1145/3097983.3098144
6. R.G. Cirstea, T. Kieu, C. Guo, B. Yang, S. Pan, Enhancenet: plugin neural networks for enhancing correlated time series forecasting, (2021) pp. 1739–1750. https://doi.org/10.1109/ICDE51399.2021.00153
7. J. Hu, B. Yang, C. Guo, C.S. Jensen, Risk-aware path selection with time-varying, uncertain travel costs: a time series approach. VLDB J. **27**(2), 179–200 (2018). https://doi.org/10.1007/s00778-018-0494-9
8. J.C.M. Oliveira, K.V. Pontes, I. Sartori, M. Embiruçu, Fault detection and diagnosis in dynamic systems using weightless neural networks. Expert Syst. Appl. **84**, 200–219 (2017). https://doi.org/10.1016/j.eswa.2017.05.020
9. P. Rajpurkar, A. Hannun, M. Haghpanahi, C. Bourn, A. Ng, Cardiologist-level arrhythmia detection with convolutional neural networks (2017)
10. B. Zhou, S. Liu, B. Hooi, X. Cheng, J. Ye, Beatgan: Anomalous rhythm detection using adversarially generated time series, (2019) pp. 4433–4439. https://doi.org/10.24963/ijcai.2019/616
11. L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S.A. Siddiqui, A. Binder, E. Müller, M. Kloft, Deep one-class classification. ed. by J. Dy, A. Krause (eds.) in *Proceedings of the 35th international conference on machine learning*. Proceedings of Machine Learning Research, vol. 80, (PMLR, 2018) pp. 4393–4402. https://proceedings.mlr.press/v80/ruff18a.html
12. Z. Li, Y. Zhao, J. Han, Y. Su, R. Jiao, X. Wen, D. Pei, Multivariate time series anomaly detection and interpretation using hierarchical inter-metric and temporal embedding. in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*. KDD '21, (Association for Computing Machinery, New York, 2021) pp. 3220–3230. https://doi.org/10.1145/3447548.3467075
13. J. An, S. Cho, Variational autoencoder based anomaly detection using reconstruction probability. (2015)
14. H. Wold, A study in analysis of stationary time series. J. R. Stat. Soc. **102**(2), 295–298 (1938)
15. C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N.V. Chawla, N.V, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. in *Proceedings of the thirty-third AAAI conference on artificial intelligence and thirty-first innovative applications of artificial intelligence conference and ninth AAAI symposium on educational advances in artificial intelligence*. AAAI'19/IAAI'19/EAAI'19. (AAAI Press, 2019). https://doi.org/10.1609/aaai.v33i01.33011409
16. I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets. in *Proceedings of the 27th international conference on neural information processing systems*, Vol. 2. NIPS'14, (MIT Press, Cambridge, MA, 2014) pp. 2672–2680

17. T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in *Information Processing in Medical Imaging*. ed. by M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, D. Shen (Springer, Cham, 2017), pp.146–157
18. M. Arjovsky, L. Bottou, Towards principled methods for training generative adversarial networks. stat **1050** (2017)
19. J. Audibert, P. Michiardi, F. Guyard, S. Marti, M.A. Zuluaga, Usad: unsupervised anomaly detection on multivariate time series. KDD '20, (Association for Computing Machinery, New York, 2020) pp. 3395–3404. https://doi.org/10.1145/3394486.3403392
20. M. Sakurada, T. Yairi, Anomaly detection using autoencoders with nonlinear dimensionality reduction. MLSDA'14, (Association for Computing Machinery, New York, 2014) pp. 4–11. https://doi.org/10.1145/2689746.2689747
21. P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection (2016)
22. M. Gutoski, M. Romero Aquino, M. Ribeiro, A. Lazzaretti, H. Lopes, Detection of video anomalies using convolutional autoencoders and one-class support vector machines. (2017). https://doi.org/10.21528/CBIC2017-49
23. C. Zhang, D. Song, Y. Chen, X. Feng, C. Lumezanu, W. Cheng, J. Ni, B. Zong, H. Chen, N.V. Chawla, A deep neural network for unsupervised anomaly detection and diagnosis in multivariate time series data. in *Proceedings of the thirty-third AAAI conference on artificial intelligence and thirty-first innovative applications of artificial intelligence conference and ninth AAAI symposium on educational advances in artificial intelligence*. AAAI'19/IAAI'19/EAAI'19. (AAAI Press, 2019). https://doi.org/10.1609/aaai.v33i01.33011409
24. H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. Feng, J. Chen, Z. Wang, H. Qiao, Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications. in *Proceedings of the 2018 world wide web conference*. WWW '18, (International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, CHE, 2018) pp. 187–196. https://doi.org/10.1145/3178876.3185996
25. D. Xu, Y. Yan, E. Ricci, N. Sebe, Detecting anomalous events in videos by learning deep representations of appearance and motion. Comput. Vis. Image Understand. **156**, 117–127 (2017). https://doi.org/10.1016/j.cviu.2016.10.010. Image and Video Understanding in Big Data
26. D. Wulsin, J. Blanco, R. Mani, B. Litt, Semi-supervised anomaly detection for EEG waveforms using deep belief nets. in *2010 Ninth international conference on machine learning and applications*, (2010) pp. 436–441. https://doi.org/10.1109/ICMLA.2010.71
27. T. Schlegl, P. Seeböck, S.M. Waldstein, U. Schmidt-Erfurth, G. Langs, Unsupervised anomaly detection with generative adversarial networks to guide marker discovery, in *Information processing in medical imaging*. ed. by M. Niethammer, M. Styner, S. Aylward, H. Zhu, I. Oguz, P.-T. Yap, D. Shen (Springer, Cham, 2017), pp.146–157
28. S. Akcay, A. Atapour-Abarghouei, T.P. Breckon, Ganomaly: semi-supervised anomaly detection via adversarial training, in *Computer vision - ACCV 2018*. ed. by C.V. Jawahar, H. Li, G. Mori, K. Schindler (Springer, Cham, 2019), pp.622–637
29. D. Li, D. Chen, B. Jin, L. Shi, J. Goh, S.-K. Ng, Mad-Gan: multivariate anomaly detection for time series data with generative adversarial networks, in *Artificial neural networks and machine learning–ICANN 2019: text and time series*. ed. by I.V. Tetko, V. Kůrková, P. Karpov, F. Theis (Springer, Cham, 2019), pp.703–716
30. D. Shipmon, J. Gurevitch, P. Piselli, S. Edwards, Time series anomaly detection; detection of anomalous drops with limited features and sparse examples in noisy highly periodic data (2017)
31. E. Keogh, J. Lin, A. Fu, ECG and 2d gesture dataset. (2005). https://www.cs.ucr.edu/~eamonn/discords/
32. P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, G. Shroff, Lstm-based encoder-decoder for multi-sensor anomaly detection (2016)
33. Y.-H. Yoo, U.-H. Kim, J.-H. Kim, Recurrent reconstructive network for sequential anomaly detection. IEEE Trans. Cybern **51**(3), 1704–1715 (2021). https://doi.org/10.1109/TCYB.2019.2933548
34. T. Kieu, B. Yang, C.S. Jensen, Outlier detection for multidimensional time series using deep neural networks. in *2018 19th IEEE international conference on mobile data management (MDM)*, (2018) pp. 125–134. https://doi.org/10.1109/MDM.2018.00029

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen journal and benefit from:

- ► Convenient online submission
- ► Rigorous peer review
- ► Open access: articles freely available online
- ► High visibility within the field
- ► Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com