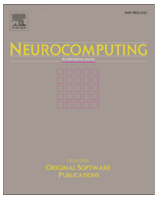
Neurocomputing 458 (2021) 349–363

ELSEVIER

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom



Self-adversarial variational autoencoder with spectral residual for time series anomaly detection

Yunxiao Liu[a,b,c], Youfang Lin[a,b,c], QinFeng Xiao[a,b,c], Ganghui Hu[a,b,c], Jing Wang[a,b,c,*]

[a] School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China
[b] Beijing Key Lab of Traffic Data Analysis and Mining, Beijing, China
[c] CAAC Key Laboratory of Intelligent Passenger Service of Civil Aviation, Beijing, China

ARTICLE INFO

Article history:
Received 7 October 2020
Revised 31 May 2021
Accepted 12 June 2021
Available online 16 June 2021
Communicated by Zidong Wang

Keywords:
Anomaly detection
Adversarial training
Spectral residual
Time series
Variational autoencoder

ABSTRACT

Detecting anomalies accurately in time series data has been receiving considerable attention due to its enormous potential for a wide array of applications. Numerous unsupervised anomaly detection methods for time series have been developed because of the difficulty of obtaining accurate labels. However, most existing unsupervised approaches suffer from the problem of anomaly contamination, which results in models that are unable to learn the normal pattern well and further deteriorate the performance of detection methods. To this end, a novel unsupervised method, called Self-adversarial Variational Autoencoder with Spectral Residual (SaVAE-SR), is introduced for time series anomaly detection in this paper. The SaVAE-SR first produces labels for unlabeled training data using the spectral residual technique to identify the most critical anomalies. A VAE model with a modified loss that can leverage label information to remove the influence of anomalous points is then trained in a self-adversarial manner, enabling the model to self-evaluate the learning of complex data distribution and improve itself accordingly. Specifically, the encoder acts as an encoder to approximate the posterior of latent variables and as a discriminator to evaluate the generative ability of the generator and improve itself accordingly. The generator is trained to capture the underlying data distribution and attempts to produce real samples to deceive the discriminator. The encoder and generator of the model compete with each other just like the behavior of GANs but work together under the theoretical framework of VAEs. As a result, the SaVAE-SR model combines the respective strengths of the VAE and adversarial training but does not require an additional discriminator, which makes the whole model very compact. Extensive experiments on five datasets demonstrate the superiority of the proposed method over the existing state-of-the-art methods.

1. Introduction

Time series anomaly detection refers to the automatic identification of abnormal behaviors from a large amount of time series data [1,2]. It is a fundamental but extraordinarily important task in data mining and has a series of application areas such as key performance indicator (KPI) monitoring [3–5], network intrusion detection [6], health monitoring [7,8], and fraudulent detection in financial transactions [9]. System failures or accidents can be reflected in their various output time series. Monitoring these time series closely and detecting anomalies within them accurately enables system failures to be discovered promptly and corresponding remedies performed. Currently, many leading companies monitor various metrics of their applications and services in real-time and build their monitoring systems [10,11] to ensure the steady operation of the whole system, further avoiding economic loss and maintaining corporate reputations. The performance of anomaly detection algorithms is of extreme importance for these monitoring systems. With the rapid growth in the scale of time series and the range of applications, efficient and effective time series anomaly detection algorithms are of significant concern.

A considerable number of efforts devoted to time series anomaly detection have been proposed in recent decades. They are roughly divided into three groups: statistical methods, supervised learning methods, and unsupervised learning methods. The first class of methods usually assumes that the normal data produced by systems conform to a family of models and only uses the normal data to learn the parameters of the model during the training stage, and then detects anomalies according to the degree to which the

[*] Corresponding author at: School of Computer and Information Technology, Beijing Jiaotong University, Beijing, China.

https://doi.org/10.1016/j.neucom.2021.06.030
0925-2312/© 2021 Elsevier B.V. All rights reserved.

incoming data conforms to the learned model. Supervised learning methods typically treat the time series anomaly detection as a classification problem. They extract various features using shallow models or deep models and subsequently perform a two-class or multi-class classification task. Although the above two kinds of methods provide impressive and promising results, unsupervised time series anomaly detection methods [3,4,12,11] have drawn a lot of attention in recent years for two main reasons. Large amounts of time series are readily available, but it is difficult to obtain the corresponding label information. This is because labels are typically annotated by domain human experts manually, which is a fairly tedious and costly process. Even with labels, the anomalies are sparse and cannot cover all types of anomalies. Both normal and anomalous samples are highly imbalanced, which makes supervised methods inefficient. The core idea of unsupervised detection methods is to learn the normal pattern of data using a large amount of available normal data. Although a large part of the training data is normal samples, there may be a few seriously abnormal samples, which greatly affect the learning of normal patterns. As shown in Fig.1, there are some significant anomalies that can affect the training of unsupervised models. Anomaly contamination still remains a significant challenge for unsupervised anomaly detection. Among various unsupervised methods, VAE-based models are preferable due to their solid theoretical background and stable performance. Nevertheless, these VAE-based methods still remain problematic. They are developed to aim at a particular type of data, such as seasonal smooth time series, and cannot generalize to the complex examples shown in Fig. 1. One possible reason is that the training principle of VAE makes it difficult to learn the underlying distribution of data very well. In contrast to VAE, GAN utilizing adversarial training, another type of popular generative model, can learn the complex data distribution well. Its disadvantage is that it is hard to train and existing GAN-based anomaly detection methods show unsatisfactory performance. As stated in [13], VAE and GAN seem to have complementary properties. Therefore, several hybrid models have been proposed to combine them [14,15] in order to better learn the true distribution of data. Nonetheless, these models either require an additional discriminator and are of complex model structure, or are developed for other problems, such as image synthesis, rather than time series anomaly detection.

To address the challenges mentioned above, we propose a new unsupervised method for time series anomaly detection, called Self-adversarial Variational Auto-Encoder with Spectral Residual (SaVAE-SR). In the whole model, the Spectral Residual (SR) technique is first adopted to produce pseudo-labels for unlabeled time series data. The main goal of this step is to identify the most significant anomalies. As illustrated in Fig. 2, after using SR to transform the raw time series (Fig. 2(a)) into the corresponding saliency map (Fig. 2(b)), we can find out the most significant anomalies by simply using a threshold denoted by a dotted blue line. A VAE model with a modified loss that can leverage the label information and remove the effect of anomalies is then trained in a self-adversarial manner. Specifically, the model consists of an encoder and a generator model similar to the structure of a standard VAE. The encoder is not only trained to minimize the divergence between the approximate posterior distribution of true samples from training data and the prior of the latent variables but is also trained to maximize the divergence between the posterior distribution of fake samples generated from the generator and the prior of the latent variables. The encoder plays the role of the discriminator as a discriminator in a common GAN. While the generator is trained to capture the underlying true distribution of data, it is also trained to deceive its encoder model. The objective of the encoder model and generator model corresponds to a min–max two-player game like that of GAN. At the same time, the theoretical background of the VAE with a modified loss builds a bridge between the encoder model and the generator model. As a result, the SaVAE-SR model not only combines the respective strengths of VAE and adversarial training but also is of a simple structure without any extra discriminators. Experiments on five real datasets illustrate the effectiveness of the proposed method.

The major contributions are summarized as follows:

- Self-adversarial Variational Auto-Encoder with Spectral Residual (SaVAE-SR), a novel unsupervised time series anomaly detection method is proposed.
- The spectral residual technique employed in the SaVAE-SR can identify the most significant anomalies in the time series and provide pseudo-labels for unlabeled data, which alleviates the problem of anomaly data contamination encountered in many unsupervised detection algorithms.

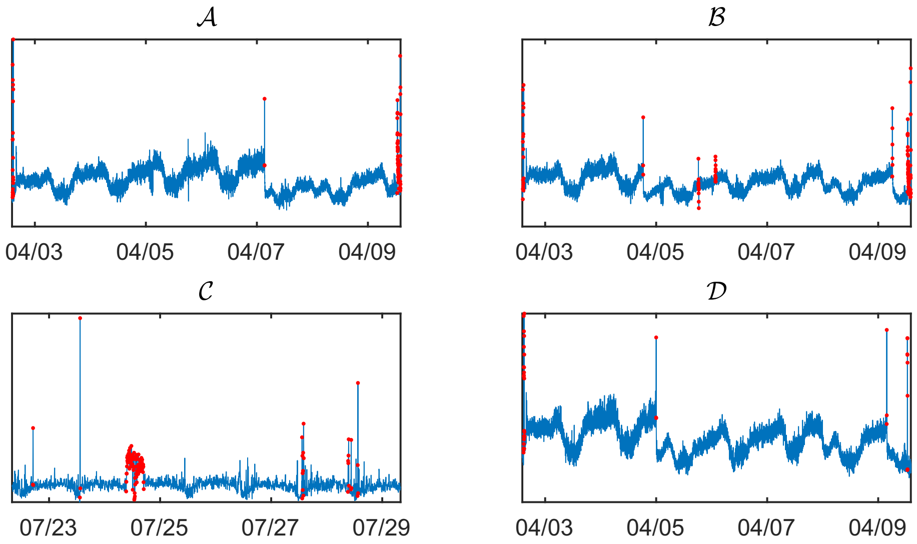


Fig. 1. Time series segments with a one-week duration of the KPI datasets used in the experiments, where red points indicate anomalies. These time series exhibit nonlinearity, nonstationarity, and non-Gaussian noise, and thus are of complex distributions that are difficult to model.

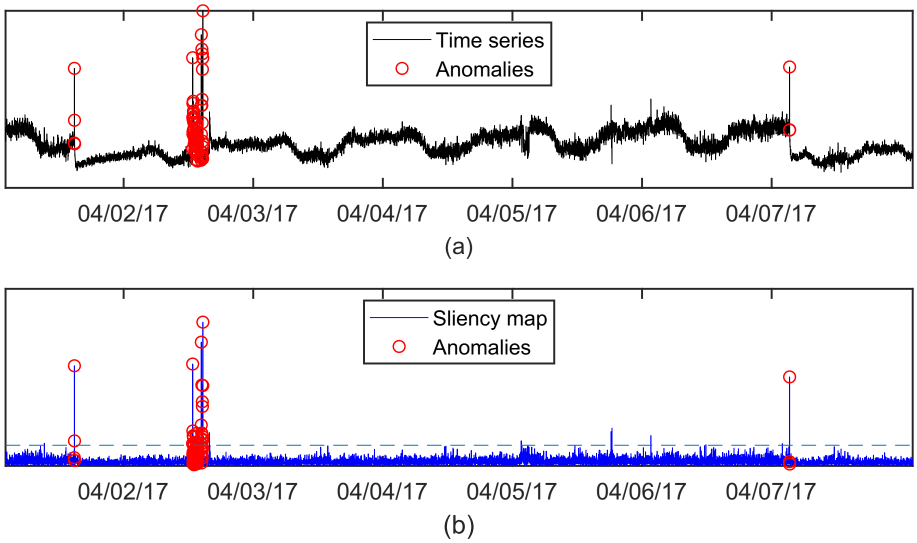


Fig. 2. Using the SR technique to ease the identification of the most significant anomalies in a time series. (a) The raw time series segment with a one-week duration and (b) the corresponding saliency map, where red circles represent anomalies.

- The VAE model with a modified loss is trained in a self-adversarial manner to learn the normal pattern of data, which makes use of the label information provided by the spectral residual technique and combines the respective strength of VAE and the adversarial training, and further improves the performance and robustness of the model.
- Extensive experiments on five real-world datasets demonstrate that SaVAE-SR obtains good results and consistently outperforms several state-of-the-art baselines.

The remainder of this paper is organized as follows. We provide an overview of related work and some preliminaries in Sections II and III, respectively. We then introduce the self-adversarial variational autoencoder with spectral residual in detail in Section IV, followed by experimental results given by the SaVAE-SR and other baseline methods. Finally, we conclude with the directions for future research in Section VI.

2. Related work

Anomaly detection in time series has gained an ever-increasing interest in academia and industry. Plenty of existing methods can be roughly grouped into three categories: traditional statistical models, supervised learning methods, and unsupervised learning methods. The first category of algorithms [16–22,7,23–27] is based primarily on statistical models such as MA, ARIMA, and Holter-Winter so that they usually make strong assumptions about the studied time series. Choosing different algorithms and fine-tuning parameters are required for different types of time series. Therefore, they are not suitable for anomaly detection in complex time series which are typically encountered in real applications and it is difficult to deploy them on real-world applications.

Supervised learning approaches consider the anomaly detection problem as a classification task [28,29,10,30,9,5,31,32]. They employ classical anomaly detectors based on statistical models as various feature extractors, and the outputs of these extractors are served as features. Subsequently, either the aggregation functions [28,29,10,30] are used to output the detection results or powerful supervised machine learning methods like Random Forest [33] are used to train a classification model [9,5,31,32]. Although supervised learning methods are powerful and exhibit promising results, they rely heavily on labeled training data, which is difficult to satisfy in the context of time series anomaly detection. Even when abnormal data are collected, they are incomplete and may not contain all types of anomalies. As a result, supervised detection methods are constrained to limited uses.

Unsupervised learning methods have attracted extensive attention and popularity due to the absence of label information. A large amount of efforts [1,2,34–41,3,42,12,4] have been proposed and have demonstrated that unsupervised methods obtain state-of-the-art performance and promising results. The core idea behind unsupervised methods is to model the normal pattern of data and those that do not conform to the normal pattern are considered as anomalies. Some earlier efforts [1,2,34] convert the raw time series into the form of feature vectors and then use the traditional unsupervised anomaly detection like One-Class Support Vector Machine (OCSVM) [43] and Local Outlier Factor (LOF) [44]. They construct low-dimensional feature vectors and cannot scale to large-scale datasets. LSTM-based methods [35,36] learn to predict the future values of the time series and detect anomalies using the difference between predicted values and true values, which assumes the time series is predictable. However, the time series derived from real-world applications are typically nonstationary and nonlinear and thus may be unpredictable. GAN-based methods [37–40] train a generator and a discriminator to detect anomalies by the output of the discriminator or the reconstructed error. But GAN-based methods typically are difficult to train and suffer from the mode collapse problem [45]. The recently proposed TadGAN [46] is built on GAN which employs the cycle consistency loss and Wasserstein loss for effective time series reconstruction so that these issues are greatly alleviated. Several autoencoder-based methods [47–49] have been proposed for time series anomaly detection. The reconstruction error is used as the anomaly score and only normal data are employed to train the model. VAE-based methods [41,3,42,12,4,50,51] are particularly popular due to their high performance and a solid theoretical background. However, the problem of anomaly contamination can affect the performance of unsupervised methods, especially some significant anomalies contained in training data. Donut [4] is an unsupervised method that introduces a modified evidence lower bound of the VAE model, enabling the model to use the label information when this information is available. But it is developed for seasonal time series data and does not perform well on complex time series and large-scale labels are typically unavailable.

3. Background and preliminaries

In this section, we briefly expound on the problem studied and provide preliminaries including spectral residual, variational autoencoder, and adversarial training. Finally, we describe the overall framework of the proposed model.

3.1. Problem statement

A time series $\mathbf{x} = \{x_t\}_{t=1}^{N}$ is a sequence of real-valued observations which are usually recorded at regular time intervals, where N represents the length of $\mathbf{x}$, $t = 1, 2, \cdots, N$ represents the index of the observation and $x_t \in \mathbb{R}$ is the t-th observation. The problem of anomaly detection in time series is to determine whether an observation x_t is abnormal given the historical observations. Considering whether an observation is abnormal or not depends on the historical information, we employ sliding windows of length W to obtain a vector $\mathbf{x}_t = \{x_{t-W+1}, x_{t-W+2}, \cdots, x_t\}$ as the representation of the observation x_t, where W represents the length of the historical observation. When we apply sliding windows over the whole time series $\mathbf{x}$, we could obtain a dataset with $N - W + 1$ samples for training the model, the resulting dataset $X \in \mathbb{R}^{(N-W+1)\times W}$:

$$X = \begin{bmatrix} \mathbf{x}_W \\ \mathbf{x}_{W+1} \\ \vdots \\ \mathbf{x}_N \end{bmatrix} = \begin{bmatrix} x_1 & x_2 & \dots & x_W \\ x_2 & x_3 & \dots & x_{W+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-W+1} & x_{N-W+2} & \dots & x_N \end{bmatrix}.$$

If the label sequence $\mathbf{y} = \{y_t\}_{t=1}^{N}$ of $\mathbf{x}$ is available, we can construct the corresponding label vector $\mathbf{y}_t = \{y_{t-W+1}, y_{t-W+2}, \cdots, y_t\}$ of $\mathbf{x}_t$ and the label matrix Y in the same way as follows:

$$Y = \begin{bmatrix} \mathbf{y}_W \\ \mathbf{y}_{W+1} \\ \vdots \\ \mathbf{y}_N \end{bmatrix} = \begin{bmatrix} y_1 & y_2 & \dots & y_W \\ y_2 & y_3 & \dots & y_{W+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{N-W+1} & y_{N-W+2} & \dots & y_N \end{bmatrix}.$$

For clarity, we will omit the subscript t of $\mathbf{x}_t$ and $\mathbf{y}_t$ without causing confusion. Although the label information is typically absent in the unsupervised anomaly detection setting, we will use the spectral residual technique to provide the pseudo-labels at the training stage in our proposed model to address the problem of lack of label information. Since our algorithm finally outputs an anomaly score for each observation point indicating a degree of being an anomaly, we use $\alpha_t = 1$ to represent the observation x_t is abnormal when the corresponding anomaly score is greater than a chosen threshold r, otherwise, $\alpha_t = 0$ indicates the observation x_t is normal.

3.2. Spectral residual

The Spectral Residual (SR) algorithm is first proposed for visual saliency detection [52] and has been recently adopted in time series anomaly detection [11] due to its simplicity and efficiency. Given a time series $\mathbf{x}$, the procedure for computing its SR is as follows,

$$A(f) = Amplitude(\mathfrak{F}(\mathbf{x})), \tag{1}$$

$$P(f) = Phase(\mathfrak{F}(\mathbf{x})), \tag{2}$$

$$L(f) = \log(A(f)), \tag{3}$$

$$AL(f) = h_q(f) \cdot L(f), \tag{4}$$

$$R(f) = L(f) - AL(f), \tag{5}$$

$$S(\mathbf{x}) = \|\mathfrak{F}^{-1}(\exp(R(f) + iP(f)))\|, \tag{6}$$

where $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ denote Fourier Transform and Inverse Fourier Transform, respectively. $A(f)$ and $P(f)$ represent the amplitude spectrum and phase spectrum of $\mathbf{x}$. $L(f)$ is the log representation of $A(f)$. $AL(f)$ is the average of $L(f)$ which can be approximated by convoluting the input series by $h_q(f)$, where $h_q(f)$ is a $q \times q$ matrix defined as:

$$h_q(f) = \frac{1}{q^2} \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix},$$

where $h_q(f)$ plays a role of the average filter, q represents the size of the filter and is typically set to 3 [52,11]. $R(f)$ is the spectral residual and serves as a compressed representation of the series where the innovation part of the original series becomes more significant. Finally, the Inverse Fourier Transform is used to transform the sequence back to the time domain. The resulting $S(\mathbf{x})$ is called the saliency map of $\mathbf{x}$. The anomaly score $O(x_i)$ [11] is defined by:

$$O(x_i) = \begin{cases} 1, & if \quad \frac{S(x_i) - \overline{S(x_i)}}{\overline{S(x_i)}} > \eta, \\ 0, & otherwise, \end{cases}$$

where $S(x_i)$ represents the saliency map of an arbitary point x_i in $\mathbf{x}$, and $\overline{S(x_i)}$ is the local average of $S(x_i)$, η is a predefined threshold.

3.3. Variational Autoencoder

Variational Autoencoder (VAE) [53,54] is a probabilistic graphical model and is mainly used to model the relationship between the observed variables $\mathbf{x}$ and the hidden variables $\mathbf{z}$. A VAE model is usually specified by a parametric generative model $p_\theta(\mathbf{x}|\mathbf{z})$ of the observed variables $\mathbf{x}$ given the latent variables $\mathbf{z}$, a prior $p(\mathbf{z})$ over the latent variables, and an approximate inference model $q_\phi(\mathbf{z}|\mathbf{x})$. The prior $p(\mathbf{z})$ is usually a multivariate unit Gaussian $\mathcal{N}(\mathbf{0}, \mathbf{I})$, the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ is a multivariate diagonal Gaussian $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2 I)$ parameterized by the encoder model. The specific form of $p_\theta(\mathbf{x}|\mathbf{z})$ is determined by the task at hand. In the context of time series anomaly detection, it is a multivariate diagonal Gaussian $\mathcal{N}(\boldsymbol{\mu}_{\mathbf{x}}, \boldsymbol{\sigma}_{\mathbf{x}}^2 I)$ parameterized by the generator model. The training objective of the VAE model is to maximize the evidence lower bound (EBLO) which is a lower bound of data log-likelihood, denoted as $L(\mathbf{x})$, its formulation is as follows,

$$\begin{aligned} L(\mathbf{x}) &= E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})] + E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\ &= L_{AE}(\mathbf{x}) + L_{REG}(\mathbf{z}). \end{aligned} \tag{7}$$

The right hand side of Eq. (7) consists of two terms, L_{AE} and L_{REG}. The first term L_{AE} is the log-likelihood term. The second term L_{REG} is the regularized term by encouraging the approximate posterior $q_\phi(\mathbf{z}|\mathbf{x})$ to match the prior $p(\mathbf{z})$.

The original VAE is an unsupervised method and cannot use the label information. For this purpose, Xu et al. modified the ELBO to use the available label information and proposed an anomaly detection model called Donut [4]. Donut [4] can remove the contribution of anomalies and missing observations to the learning of the model when this information is available. The formulation of the modified ELBO (MELBO) is as follows,

$$\begin{aligned} \tilde{L}(\mathbf{x},\mathbf{y}) &= E_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\sum_{w=1}^{W}(1-y_w)\log p_\theta(x_w|\mathbf{z}) + \beta \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})\right] \\ &= E_{q_\phi(\mathbf{z}|\mathbf{x})}\left[\sum_{w=1}^{W}(1-y_w)\log p_\theta(x_w|\mathbf{z})\right] + E_{q_\phi(\mathbf{z}|\mathbf{x})}[\beta \log p_\theta(\mathbf{z}) - \log q_\phi(\mathbf{z}|\mathbf{x})] \\ &= \tilde{L}_{AE}(\mathbf{x},\mathbf{y}) + \tilde{L}_{REG}(\mathbf{z},\mathbf{y}), \end{aligned} \tag{8}$$

where $\mathbf{y} = \{y_1, y_2, \ldots, y_W\}$ is the label vector corresponding to the sample $\mathbf{x} = \{x_1, x_2, \ldots, x_W\} = 1,\ y_w = 1$ indicates the observation x_w is abnormal or missing, and $y_w = 0$ represents x_w is a normal observation. β is defined as $1 - \sum_{w=1}^{W} y_w / W$ and represents the portion of the normal points in a window. We use $\widetilde{L}_{AE}$ and $\widetilde{L}_{REG}$ to denote the modified log-likelihood term and modified regularized term, respectively. The MELBO is reasonably useful and helpful in the context of time series anomaly detection due to the partly available labeled and missing data. In this work, we employ the MELBO as the objective function in the proposed SaVAE-SR.

3.4. Adversarial training

Adversarial training is one of the core ideas of GAN [55] and its numerous variants. Typically, the structure of a GAN contains a discriminator D and a generator G. The discriminator is trained to distinguish real samples from fake samples generated by the generator, while the generator is trained to produce fake samples to deceive the discriminator. The discriminative ability of the discriminator and the generative ability of the generator can be improved during the process of adversarial training. Given a sample $\mathbf{x}$, the objective of GAN is as follows:

$$\max_{D}\min_{G} E_{\mathbf{x}\sim p_{data}} \log D(\mathbf{x}) + E_{\mathbf{z}\sim p(\mathbf{z})}(1 - D(G(\mathbf{z}))). \tag{9}$$

The merit of GAN-based models is that generated samples look very realistic, but they are difficult to train stably and suffer from the mode collapse problem. To this end, several efforts [14,15] have been introduced aiming at combining the VAE and adversarial training. They require an extra discriminator in the latent space or data space to distinguish true data derived from the training data from the fake data generated by the generator, which increases the complexity of the model. However, EBGAN [56], one of the various variants of GAN, views the discriminator as an energy function that assigns low energies to the regions near the data manifold and higher energies to other regions. Viewing the discriminator as an energy function allows us to utilize more flexible architectures and loss functions besides the frequently-used binary cross-entropy loss. Its objective is formulated as:

$$\max_{D}\min_{G} E_{\mathbf{x}\sim p_{data}} \log D(\mathbf{x}) + E_{\mathbf{z}\sim p(\mathbf{z})}[m - D(G(\mathbf{z}))]^{+}, \tag{10}$$

where $[\cdot]^{+} = \max(\cdot, 0)$ and m is a positive margin that needs to be determined empirically. The IntroVAE [57] adopts this idea and utilizes the adversarial training of the encoder and generator in a VAE model to combine the VAE model and adversarial learning cleverly. But it is proposed as a method for image synthesis rather than anomaly detection. Inspired by IntroVAE, we train the encoder and generator of our VAE model in a self-adversarial manner to improve the generative ability of our model.

3.5. The overall framework

The overall framework of the SaVAE-SR model consists of four parts: Pre-labelling, Model Training, Anomaly Detection, and Performance Evaluation as shown in Fig. 3. In the Pre-labeling module (Fig. 3(a)), the missing value in the training set is first filled using the linear interpolation technique. The spectral residual method [52,11] is then adopted to generate the point-wise pseudo-labels for unlabeled training data. The goal of this operation is to find out the most significant anomalies in unlabeled training data as much as possible and therefore alleviate the problem of anomaly data contamination. In the Model Training module (Fig. 3(b)), the time series with pseudo-labels is normalized and passes through a sliding window to form the training dataset and is then sent to train a VAE model with modified ELBO capturing the normal pattern of data in an adversarial training manner, this model is called Self-adversarial Variational Autoencoder (SaVAE). Subsequently, the well-trained model is applied for anomaly detection in testing data as shown in Fig. 3(c). The testing data may contain the missing data, therefore, it is first normalized and then through the MCMC imputation technique [4] to fill the missing values. The MCMC imputation technique takes advantage of the learned model to reconstruct the input sample including missing values and replaces the missing values with the corresponding part of the reconstructed one. More specifically, a testing sample $\mathbf{x} = [\mathbf{x}_o, \mathbf{x}_m]$, where $\mathbf{x}_o$ is the observed part and $\mathbf{x}_m$ is the missing part, is passed the model to obtain the reconstruction $\mathbf{x}_r = [\mathbf{x}_o^1, \mathbf{x}_m^1]$ for the first time.

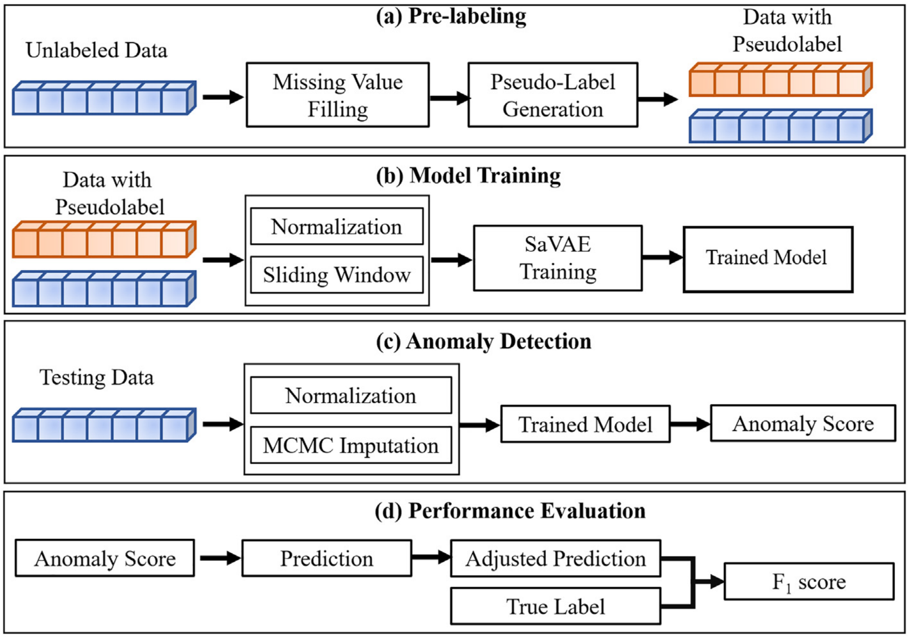


Fig. 3. The overall framework consists of four modules: (a) Pre-labeling, (b) Model Training, (c) Anomaly Detection, and (d) Performance Evaluation.

Subsequently, $\mathbf{x}_m$ is replaced by the $\mathbf{x}_m^1$ and regarding $\left[\mathbf{x}_o, \mathbf{x}_m^1\right]$ as the input to feed into the model to obtain $\left[\mathbf{x}_o^2, \mathbf{x}_m^2\right]$ again. Repeat this procedure for M times, the ultimately obtained $\mathbf{x}_m^M$ is used to replace the missing part and $\left[\mathbf{x}_o, \mathbf{x}_m^M\right]$ is used for computing the anomaly score of the sample $\mathbf{x}$. Typically, a predefined threshold is provided r, if the anomaly score of a sample is larger than r, this sample is regarded as an anomaly. We do not focus on the choice of threshold in this study. Therefore, in the Performance Evaluation module (Fig. 3(d)), we use every possible threshold to obtain the prediction and adjust the obtained prediction. Why and how to adjust the prediction is descripted specifically in Section 5.1. And then we use the adjusted prediction and true label to compute F_1 score and finally report the best one.

4. Proposed method

In this section, an unsupervised anomaly detection method, **S**elf-**a**dversarial **V**ariational **A**utoencoder with **S**pectral **R**esidual (SaVAE-SR), is introduced for the time series. The first step of the proposed method is to produce pseudo-labels for unlabelled time series data by simply employing the spectral residual with the goal of identifying the most significant anomalies. Subsequently, a VAE model consisting of an encoder network and a decoder network combined with a modified loss function that can utilize the label information is trained in a self-adversarial manner. The structure of the SaVAE-SR and its adversarial training steps is illustrated in Fig. 4. The process of producing labels and adversarial training of the model is detailed below.

4.1. Spectral residual for pre-labeling

In the proposed method, the first step after missing values filling and normalizing is to provide pseudo-labels for unlabeled training data to alleviate the problem of data contamination encountered in many unsupervised methods. For this purpose, the spectral residual (SR) technique is adopted to output the point-wise pseudo-label for training data. The goal of using the SR algorithm is not to create accurate labels for unlabeled training data, but rather to find the most significant anomalies contained in training data as much as possible. Given a time series $\mathbf{x} = \{x_1, x_2, \cdots, x_N\}$ of N points, its SR is $S(\mathbf{x}) = \{S(x_1), S(x_2), \cdots, S(x_N)\}$ that can be calculated through the Eqs. (1) to (6). The pseudo-label y_i of the observation $x_i, i = 1, 2, \cdots, N$ is given by the following formulation:

$$y_i = \begin{cases} 1, & S(x_i) \geqslant r, \\ 0, & S(x_i) < r, \end{cases}$$

where r is a threshold. Although there are many complicated methods used for selecting a better threshold, we simply select the 95th quantile of $S(\mathbf{x})$ as the value of r. As demonstrated in the following experiments, this simple method provides good performance. It is worth noting that the corresponding labels of the missing observations are also 1. In the model training stage, the contribution of both anomalies and missing points to model training would be removed.

4.2. Self-adversarial Variational Autoencoder for Model Training

The basic structure of Self-adversarial Variational Autoencoder (SaVAE) consists of an encoder E and a generator G just like a standard VAE model. The encoder E is trained to learn the parameters of the approximate posterior distribution of the latent variables $\mathbf{z}$. The generator G is trained to learn the generative process of the data $\mathbf{x}$ given the latent variables $\mathbf{z}$. To improve their respective ability, we train them in a self-adversarial manner. Namely, in the process of model training, the encoder E also plays the role of the discriminator to judge the realness of the reconstruction ones. Thus, it is trained to minimize the KL-divergence of the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x})$ of the real data to match the prior $p(\mathbf{z})$ and simultaneously to maximize the KL-divergence of the posterior distribution of fake samples generated by the generator G to deviate from the prior $p(\mathbf{z})$. Fake samples generated by the generator have two sources. As shown in Fig. 4, one is derived from the construction samples, another is derived from the generated samples through the prior. To take advantage of the generated labels to remove the effect of some significant anomalies on the model training, thus we employ the $\widetilde{L}_{AE}$ and $\widetilde{L}_{REG}$ mentioned in MELBO to compute the construction error and the regularization term. As a result, the training objective of the encoder E is formulated as follows:

$$L_E = \widetilde{L}_{AE}(\mathbf{x}, \mathbf{y}) + \widetilde{L}_{REG}(\mathbf{z}, \mathbf{y}) + \left[m - \widetilde{L}_{REG}(\mathbf{z_r}, \mathbf{y})\right]^+ + \left[m - \widetilde{L}_{REG}(\mathbf{z_{pp}}, \mathbf{y})\right]^+, \quad (11)$$

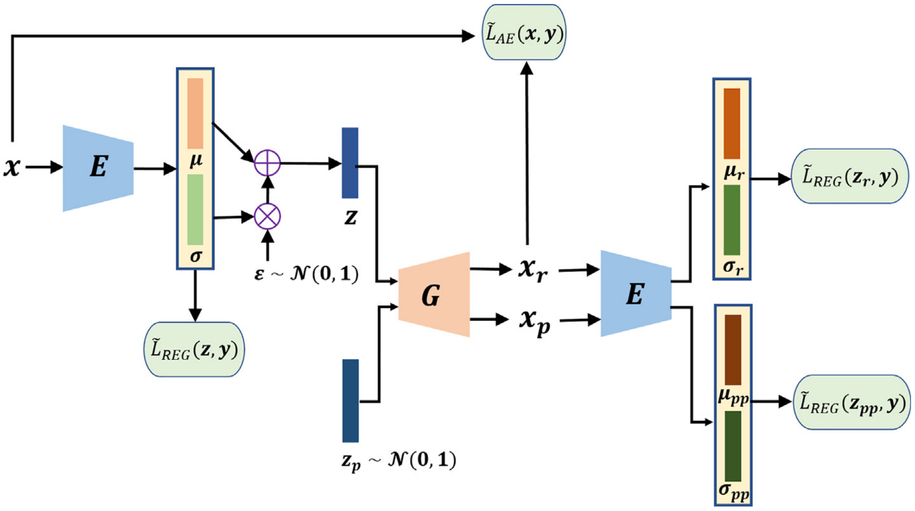

$\widetilde{L}_{AE}(\boldsymbol{x}, \boldsymbol{y})$
$\boldsymbol{x}$
E
$\boldsymbol{\mu}$
$\boldsymbol{\sigma}$
$\boldsymbol{z}$
$\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
$\widetilde{L}_{REG}(\boldsymbol{z}, \boldsymbol{y})$
G
$\boldsymbol{x_r}$
$\boldsymbol{x_p}$
$\boldsymbol{z_p} \sim \mathcal{N}(\mathbf{0}, \mathbf{1})$
E
$\boldsymbol{\mu_r}$
$\boldsymbol{\sigma_r}$
$\widetilde{L}_{REG}(\boldsymbol{z_r}, \boldsymbol{y})$
$\boldsymbol{\mu_{pp}}$
$\boldsymbol{\sigma_{pp}}$
$\widetilde{L}_{REG}(\boldsymbol{z_{pp}}, \boldsymbol{y})$

Fig. 4. Illustration of a training step of SaVAE. The SaVAE model is composed of an encoder model E and a generative model G. The training data passes forward the E and produces the true latent codes $\mathbf{z}$, and $\mathbf{z}$ is passing the G and generate the reconstructed data $\mathbf{x}_r$. The reconstructed data $\mathbf{x}_r$ is passing the E and produce the fake latent code $\mathbf{z}_r$. The objective of training E is to minimize the divergence between the latent code and the prior, and to maximize the divergence between the fake latent code and the prior. The objective of training G is to deceive the discriminator.

where $\mathbf{z}$ is from the posterior distribution of training data, both $\mathbf{z_r}$ and $\mathbf{z_{pp}}$ are from the posterior distribution of fake data produced by the generator model G. The $\mathbf{z_r}$ is the reconstructed fake sample, and $\mathbf{z_{pp}}$ is the generated fake sample drawn from the prior distribution $p(\mathbf{z})$. Meanwhile, the generator G attempts to let the E believe that its produced samples are from the training data. Therefore, its training objective is to minimize the KL-divergence of the posterior distribution of fake samples generated by the generator G to deviate from the prior $p(\mathbf{z})$ and formulated as follows,

$$L_G = \widetilde{L}_{AE}(\mathbf{x},\mathbf{y}) + \widetilde{L}_{REG}(\mathbf{z_r},\mathbf{y}) + \widetilde{L}_{REG}(\mathbf{z_{pp}},\mathbf{y}). \tag{12}$$

We train the VAE model in a self-adversarial manner such that the model can self-evaluate the differences between the generated data and the real data and improve itself accordingly. Therefore, the objectives of the G and E form a minimax two-play game. Note that the log-likelihood term builds naturally a bridge between the encoder and the generator, such that our model remains the stable training of VAE and at the same time keeps the advantage of the GAN. Algorithm 1 illustrates the training process of the encoder and generator in the SaVAE model. There is only a hyper-parameter m need to determine in advance. We empirically set it to 15 and find it can work very well.

4.3. Anomaly detection

In the anomaly detection module, we have completed the model training and learned the pattern of the normal samples. We adopt $E_{q_\phi(\mathbf{z}|\mathbf{x})}[\log p_\theta(\mathbf{x}|\mathbf{z})]$ as the anomaly score of the observation $\mathbf{x}$, which is a probabilistic measure and has been widely used. There are other choices for VAE-based models and the detained discussion about them is referred to as Ref. [4]. In addition, we also employ the MCMC imputation proposed in [4] for missing values filling in testing data. In experiments, we apply the MCMC imputation technique in all VAE-based baseline methods to ensure the fairness of comparison. After obtaining the anomaly scores of testing data, we feed them into the Performance Evaluation module to evaluate the performance of our method.

5. Experiments

In this section, we first describe the experimental setup, including the datasets, evaluation metrics, and baseline methods. Then, we conduct a series of systematic experiments to evaluate the performance of the proposed model and the effect of several important parameters. Finally, we report and discuss experimental results.

Algorithm 1 Training SaVAE model

```
$\theta_G$, $\phi_E$ ← Initialize network parameters.
while not converged
    $X, Y$ ← Random mini-batch from dataset
    $Z \leftarrow E(X)$
    $Z_p$ ← Samples from prior $N(0, I)$
    $X_r \leftarrow G(Z)$, $X_p \leftarrow G(Z_p)$
    $Z_r \leftarrow E(X_r)$, $Z_{pp} \leftarrow E(X_p)$
    $L_E \leftarrow \widetilde{L}_{AE}(X,Y) + \widetilde{L}_{REG}(Z,Y)$
      $+\left[m - \widetilde{L}_{REG}(Z_r,Y)\right]^+ + \left[m - \widetilde{L}_{REG}(Z_{pp},Y)\right]^+$
    $\phi_E \leftarrow \phi_E - \eta\Delta_{\phi_E}(L_E)$
    $Z_r \leftarrow E(X_r)$, $Z_{pp} \leftarrow E(X_p)$
    $L_G \leftarrow \widetilde{L}_{AE}(X,Y) + \widetilde{L}_{REG}(Z_r,Y) + \widetilde{L}_{REG}(Z_{pp},Y)$
    $\theta_G \leftarrow \phi_G - \eta\Delta_{\theta_G}(L_G)$
end while
```

5.1. Experimental setup

Datasets. To evaluate the performance of SaVAE-SR, we carry out extensive experiments on five publicly available datasets. The datasets $\mathscr{A}$, $\mathscr{B}$, $\mathscr{C}$ and $\mathscr{D}$ are KPI datasets which are collected from large Internet companies and released by the AIOps competition[1], where datasets $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{D}$ have an interval of 1 min while dataset $\mathscr{C}$ has an interval of 5 min. The last dataset used is Yahoo S5[2] which is composed of four different sub-datasets. The A1 dataset is based on real production traffic of Yahoo computing systems. The other three are based on synthetic time series. Each signal in the Yahoo S5 data has no missing values. Their statistical information is summarized in Tables 1 and 2, respectively. Time series across all datasets have manual labels. Our proposed method is performed in an unsupervised fashion. Thus, our model utilizes the pseudo-labels generated by the SR algorithm instead of the original labels during the model training phase. The original labels are used only during the validation phase to evaluate the performance of the model.

Network architecture. Both the encoder model and generator model in SaVAE-SR are consisted of a fully-connected neural network with two hidden layers, and each layer has one hundred units. The SGD algorithm with a fixed learning rate of 0.0002 for the encoder model and 0.0005 for the generative model is used to update iteratively the parameters of the model. The sliding window size W, namely the dimension of the input, is set to 120 empirically, the dimension of the latent variables $\mathbf{z}$ is empirically set to 3, and the positive margin m is set to 15. During the model training stage, the batch size is 256 and the model is trained for 100 epochs. An implementation based on the PyTorch [58] of the proposed SaVAE-SR model is available on Github repository[3].

Evaluation metrics. Precision, Recall, and F1-score are adopted to evaluate the performance of the proposed model and baseline methods. Their formulations are expressed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{13}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{14}$$

Table 1
The detailed information of the KPI datasets.

Datasets	Length	Anomaly ratio	Missing ratio
$\mathscr{A}$	299053	0.761%	1.227%
$\mathscr{B}$	299053	0.639%	1.238%
$\mathscr{C}$	17856	2.666%	7.924%
$\mathscr{D}$	299053	0.528%	1.217%

Table 2
The detailed information of the Yahoo S5 dataset.

Dataset	signal	Length	Total of points	Anomaly ratio
A1	67	1420	92300	1.58%
A2	100	1421	142100	0.32%
A3	100	1680	168000	0.56%
A4	100	1680	168000	0.62%

[1] http://iops.ai/competition_detail/?competition_id=5&flag=1.

[2] https://webscope.sandbox.yahoo.com/catalog.php?datatype=s&did=70.

[3] https://github.com/lyxiao15/SaVAE-SR.

$$\mathrm{F1} = \frac{2 \times \mathrm{Precision} \times \mathrm{Recall}}{\mathrm{Precision} + \mathrm{Recall}}, \tag{15}$$

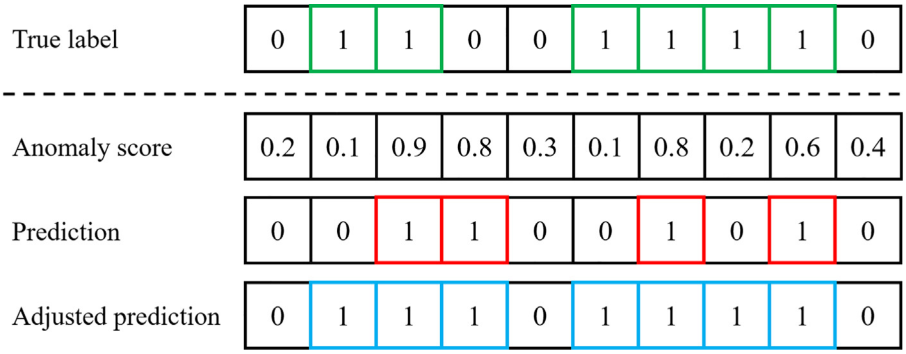
where TP, FP, and FN represent the true positive, the false positive, and the false negative respectively. In practice applications, operators usually concern about whether abnormal observations could be detected not too late instead of at extract time. Therefore, several modified metrics for anomaly detection in time series have been introduced to accommodate this preference, such as [59,4]. This adjusted metric in [4] has been widely accepted and used [41,3], therefore, we use this adjusted metric for evaluating our model and several baseline methods. Specifically, if any point in an anomaly segment in the ground truth is detected within a given time delay τ, then all points in this segment are thought to detect correctly. This adjusted process with $\tau = 1$ is illustrated in Fig. 5. In our experiments, the value of τ is set to 7 which is a commonly used value [4,3,41,11] and the mean and standard deviation of the adjusted best F1-score of 10 independent experiments is reported.

Baselines. One supervised anomaly detection method: Opprentice [5] and seven unsupervised anomaly detection methods: SR [11], Vanilla VAE [12], Donut [4], Bagel [3], USAD [49], adVAE [51], and TadGAN [46] are served as baseline methods to demonstrate the performance of SaVAE-SR. Opprentice [5] is a state-of-the-art supervised method that extracts various statistical features as well as transform-based features and performs anomaly detection using the random forest classifier. SR [11] is a Fourier Transformation-based method that attempts to detect anomalies by using the saliency map of time series. The vanilla VAE [12] directly utilizes the data obtained using the sliding window to train a standard VAE model and employs the reconstructed probability to perform anomaly detection. Donut [4] proposes a modified ELBO to replace the original objective function of a VAE model, which removes the consequence of missing values and accessible anomalies for the learning of data distribution. Bagel [3] in an improvement version of Donut, which incorporates external time code information and applies the conditional variational autoencoder to learn the normal pattern of data. USAD [49] couples autoencoder with adversarial training and leverages the adversarial training between the encoder and decoder to learn how to amplify the reconstruction error of the abnormal inputs. adVAE [51] assumes that in the latent space, the corresponding latent of anomalies also has a Gaussian prior distribution. And thus it introduces a Gaussian transformer net to synthesize the abnormal latent variables and trains the encoder and generator of a VAE model in an adversarial manner to improve on both the encoder and decoder. TadGAN [46] couples the autoencoder with the adversarial generative network and employs the cycle consistency loss to ensure that maps between the latent variate and the data are inverse. Both vanilla VAE, Donut, Bagel, are VAE-based methods without employing the adversarial training, their encoder and decoder have the same network structure as that of our model, except with a different learning rate of 0.001. Since SaVAE-SR is trained in an adversarial manner, therefore, we select to use a smaller learning rate of 0.0002 for the encoder and 0.0005 for the generator. The encoder and decoder of adVAE also have the same network as us but with a different learning rate of 0.0001.

5.2. Results

5.2.1. Overall performance

We illustrate the best F1-score of SaVAE-SR and baseline methods on studied datasets in Tables 3 and 4. It is observed that the performance of our method is consistently superior to those of several state-of-the-art baselines on the KPI datasets and A1 dataset in which signals are drawn from real-world applications. On dataset A2, the performance of the proposed method is on par with several state-of-the-art baselines. On the datasets A3 and A4, both Bagel and SR obtain better results. The main reason is those signals in



True label	0	1	1	0	0	1	1	1	1	0
Anomaly score	0.2	0.1	0.9	0.8	0.3	0.1	0.8	0.2	0.6	0.4
Prediction	0	0	1	1	0	0	1	0	1	0
Adjusted prediction	0	1	1	1	0	1	1	1	1	0

Fig. 5. The process of adjusting prediction. The first row is the true labels of data, there are two anomaly segments marked by green boxes. The second row is the anomaly score produced by the trained model. The third row is the prediction by thresholding a value of 0.5, datapoints highlighted by red boxes are detected anomalies. The last row shows the adjusted prediction, where the time delay is set to 1. The blue boxes indicate the detected anomalies after adjusting.

Table 3
The best F1-scores of eight baselines and SaVAE-SR method on KPI dataset.

Methods	$\mathscr{A}$	$\mathscr{B}$	$\mathscr{C}$	$\mathscr{D}$
Opprentice [5]	0.803 ± 0.028	0.711 ± 0.0428	0.808 ± 0.0324	0.830 ± 0.0211
SR [11]	0.814	0.700	0.175	0.676
VAE [12]	0.902 ± 0.0221	0.808 ± 0.0446	0.850 ± 0.0466	0.841 ± 0.0328
Donut [4]	0.897 ± 0.0174	0.809 ± 0.0439	0.898 ± 0.0453	0.841 ± 0.0369
Bagel [3]	0.882 ± 0.0238	0.835 ± 0.0322	0.856 ± 0.0508	0.876 ± 0.0508
USAD [49]	0.841 ± 0.0007	0.834 ± 0.0009	0.853 ± 0.0031	0.835 ± 0.0005
adVAE [51]	0.948 ± 0.0188	0.879 ± 0.0384	0.872 ± 0.0035	0.775 ± 0.0674
TadGAN [46]	0.395 ± 0.0146	0.433 ± 0.1015	0.519 ± 0.0915	0.456 ± 0.0008
SaVAE-SR	$\mathbf{0.977 \pm 0.0056}$	$\mathbf{0.951 \pm 0.0050}$	$\mathbf{0.984 \pm 0.0023}$	$\mathbf{0.966 \pm 0.0054}$

Table 4
The best F1-scores of eight baselines and SaVAE-SR method on the Yahoo dataset.

Methods	A1	A2	A3	A4
SR [11]	0.283	0.569	0.844	0.782
VAE [12]	0.816 ± 0.0073	0.717 ± 0.0018	0.816 ± 0.0129	0.692 ± 0.0180
Donut [4]	0.817 ± 0.0061	0.715 ± 0.0015	0.805 ± 0.0088	0.687 ± 0.0274
Bagel [3]	0.829 ± 0.0038	0.718 ± 0.0015	**0.911 ± 0.0055**	**0.811 ± 0.0104**
USAD [49]	0.807 ± 0.0002	0.468 ± 0.0002	0.689 ± 0.0003	0.669 ± 0.0002
adVAE [51]	0.832 ± 0.0090	**0.738 ± 0.0036**	0.803 ± 0.0124	0.653 ± 0.0039
TadGAN [46]	0.611 ± 0.0651	0.593 ± 0.0309	0.638 ± 0.0017	0.493 ± 0.005
SaVAE-SR	**0.842 ± 0.0068**	**0.711 ± 0.0040**	**0.791 ± 0.0158**	**0.705 ± 0.0198**

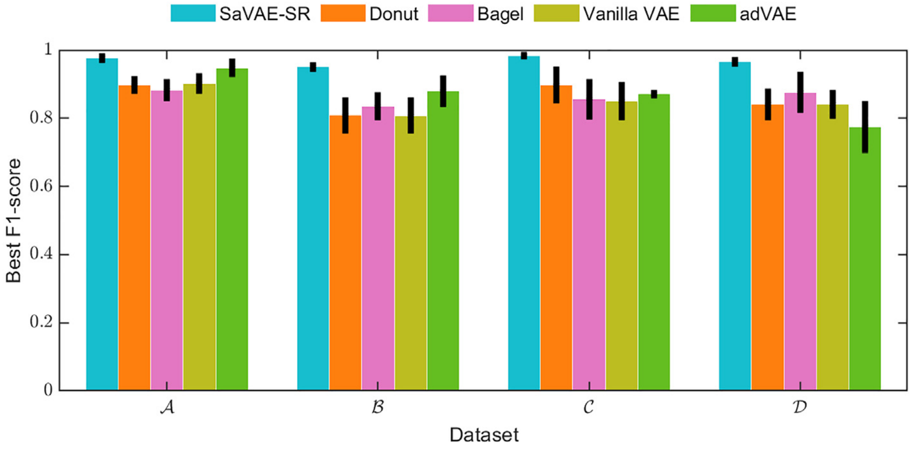


Fig. 6. Best F1-score of four VAE-based baselines and SaVAE-SR.

datasets A3 and A4 have obvious periodicity. Bagel can code the time information, while the SR algorithm is based on the Fourier transform, and they are all suitable for this type of signal. TadGAN is a GAN-based method that illustrates poor results on almost all datasets. In addition, several VAE-based baselines show good results on four KPI datasets, indicating the effectiveness of VAE-based methods. We also observe that the supervised method, Opprentice, illustrates good performance on the datasets $\mathscr{B}$ and $\mathscr{D}$, but it relies on accurate label information. Further, we illustrate the performance of several VAE-based baselines and SaVAE-SR on the KPI datasets in Fig. 6. It can be observed that our method not only obtains the best performance but also has a considerably small standard deviation, which indicates that our method is very stable and robust over other VAE-based baselines. The adVAE, another method combining the VAE model and adversarial training, also presents relatively good performance, which implies the effectiveness of combing adversarial training. We visualize the detected results of state-of-the-art baselines (except for TadGAN that obtains the worst results) and the proposed method on a segment of time series in dataset $\mathscr{A}$ in Fig. 7. Our SaVAE-SR detects all anomalies, while other baselines either provide fake alarms or miss some anomalies.

5.2.2. Effect of parameters

In this section, we study the effect of different parameters on the performance of the SaVAE-SR. All experiments are conducted on the first four datasets for which there are sufficient data points.

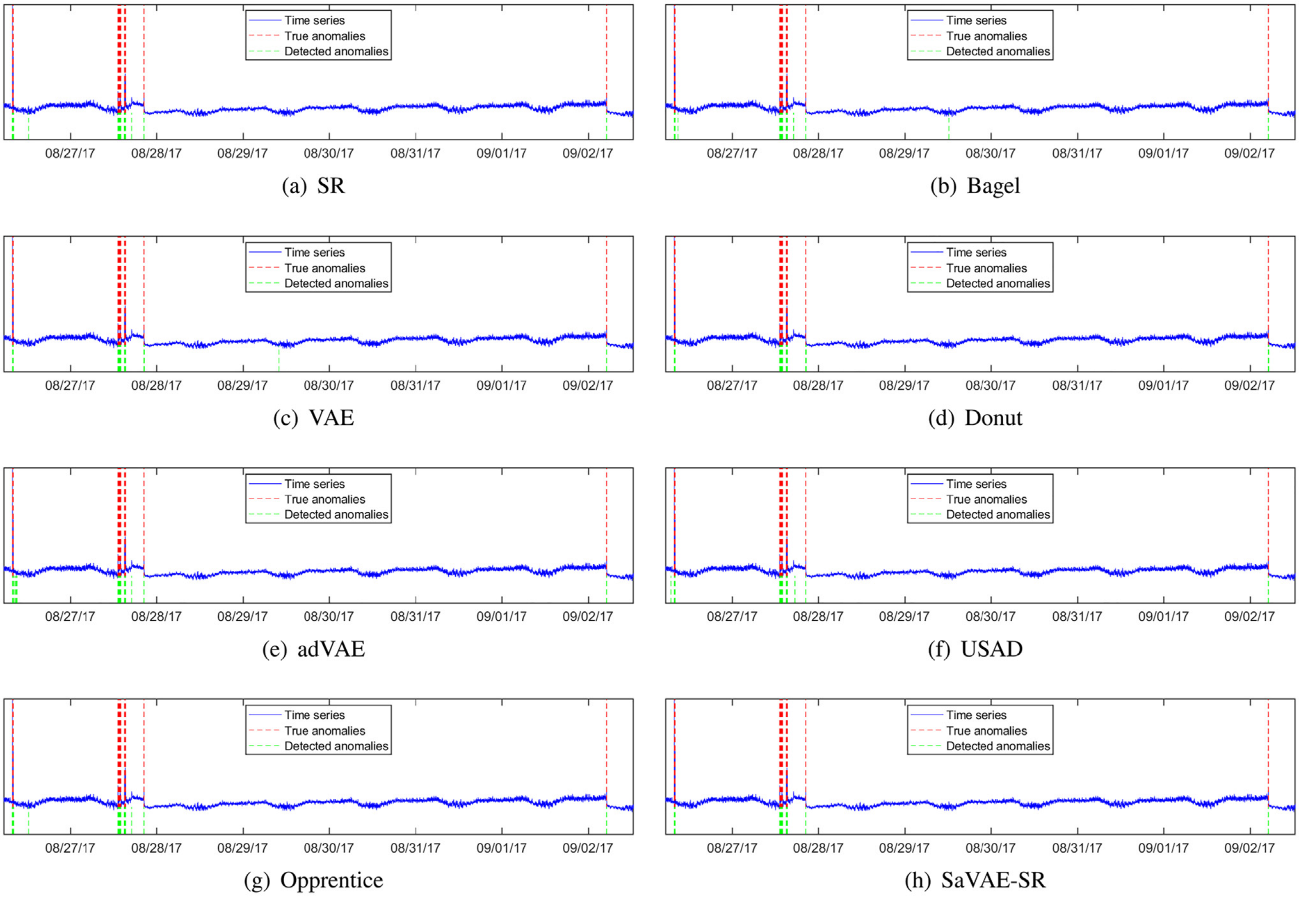
The positive margin m. The parameter m is a positive margin, we simply vary its value m from 1 to 30 with an interval of 2 and show the curve of the best F1-scores over the values of m in Fig. 8. It is observed that the performance of the SaVAE-SR model is suboptimal when the value of m is less than 15. When $m \geqslant 15$, the performance reaches the best one, and it is insensitive to the value of the parameter m in a fairly wide range. Therefore, we empirically set the value of it to 15 in the experiments of this work and find it can obtain fairly good performance. Even though, it is possible to obtain a better performance via fine-tuning the value of m for the specific time series.

The dimension of $\mathbf{z}$. The second key parameter is the dimension of latent variable $\mathbf{z}$ which plays a critical role in the whole model. The too large or too small value of the dimension may harm the performance of the model. For this purpose, we range its values from 1 to 30 with a step size 2 and show the result in Fig. 9. Obviously, our method is robust against this parameter except for the dataset $\mathscr{D}$. When the dimension of $\mathbf{z}$ is larger than 3, the performance of SaVAE-SR keeps stable on all datasets, therefore, it is empirically set to 3.

The window size W. The parameter W indicates the length of the historical information used to represent an observation. We vary the value of W from 120 to 300 (2 h to 6 h for datasets $\mathscr{A}$, $\mathscr{B}$ and $\mathscr{D}$, 6 h to 25 h for datasets $\mathscr{C}$) with an interval 30 and illustrate the results in Fig. 10. The larger W is, the more historical information is considered. However, as shown in Fig. 10, the large W does not bring the improvement of the performance of our method. For the datasets $\mathscr{A}$ and $\mathscr{B}$, the performance of our method approximately remains stable. While for the datasets $\mathscr{C}$ and $\mathscr{D}$, the performance of SaVAE-SR degrades with the increase of W. Thus, the value of W is empirically set to 120.

5.2.3. Effect of different proportions of anomalies

In this section, we investigate the effect of different proportions of anomalies on the performance of the proposed SaVAE-SR method. Because the type of data studied is time series, simply deleting anomalous points can break down the continuity of the time series, while adding anomalous points or changing certain points to anomalies also has some issues, such as how to control the anomalous level of added points. Nevertheless, the input data are obtained through the sliding window we carry out this investi-



(a) SR

(b) Bagel

(c) VAE

(d) Donut

(e) adVAE

(f) USAD

(g) Opprentice

(h) SaVAE-SR

Fig. 7. Anomalies detected using seven state-of-the-art baselines and our SaVAE-SR on an exemplary time series in dataset $\mathscr{A}$. Blue solid lines are the time series, red dashed lines represent true anomalies, and short green lines indicate detected anomalies.

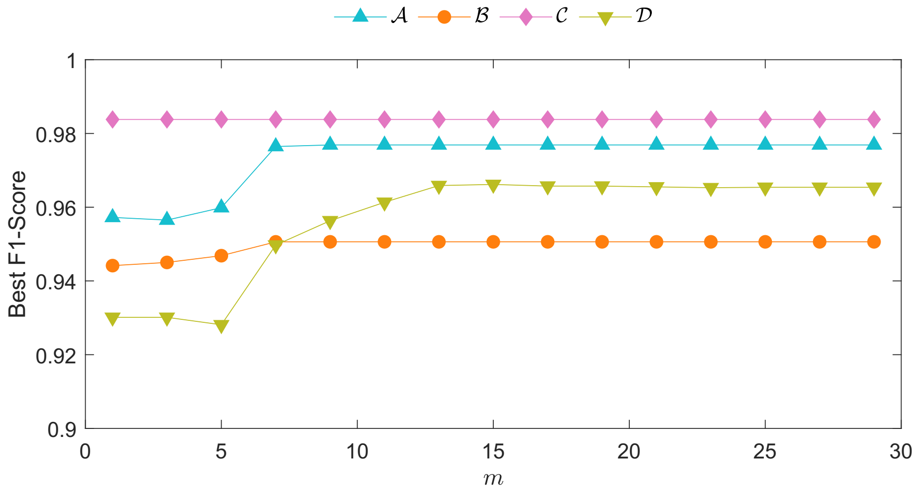


Fig. 8. Effect of the parameter m.

gation by varying the proportion of anomalous windows in training data. In this context, if a window contains anomalies, it is considered as an anomalous input. We vary the proportion of anomalous windows in training data from 1% to 10% and report the results in Fig. 11. Our method is relatively insensitive to the proportion of anomalous windows, which is explainable. The proposed SaVAE-SR method does not assume that the training data is clean, which is implied by many existing unsupervised anomaly detection approaches. The proposed SaVAE-SR employs the SR algorithm to label the most significant anomalies at first. The SR algorithm may mislabel some normal points as anomalies, but this is trivial. There is plenty of normal data which is sufficient for the normal pattern of the model.

5.2.4. Ablation study

Our model consists of three key modules: SR and two adversarial terms, denoted as $\mathbf{z}_{\mathbf{pp}}$ and $\mathbf{z}_{\mathbf{r}}$ respectively. The SR is employed to provide pseudo-labels for unlabeled training data,

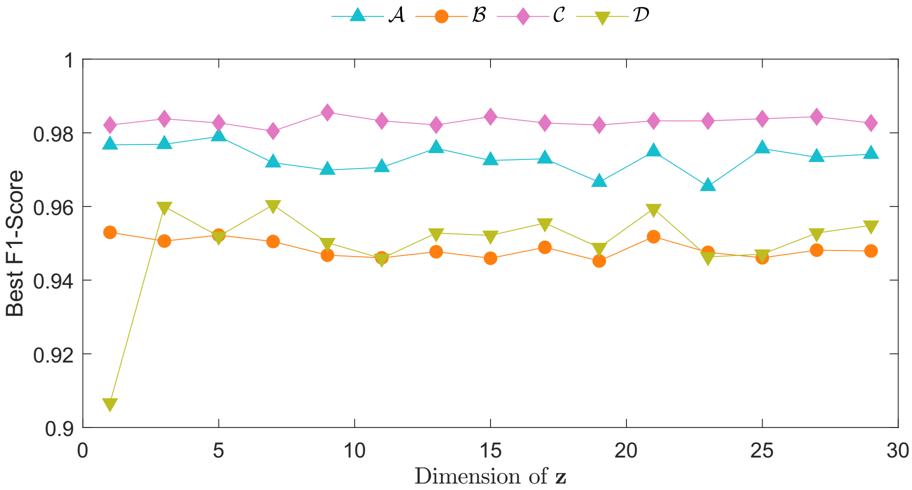


Fig. 9. Effect of the dimension of the latent variable **z**.

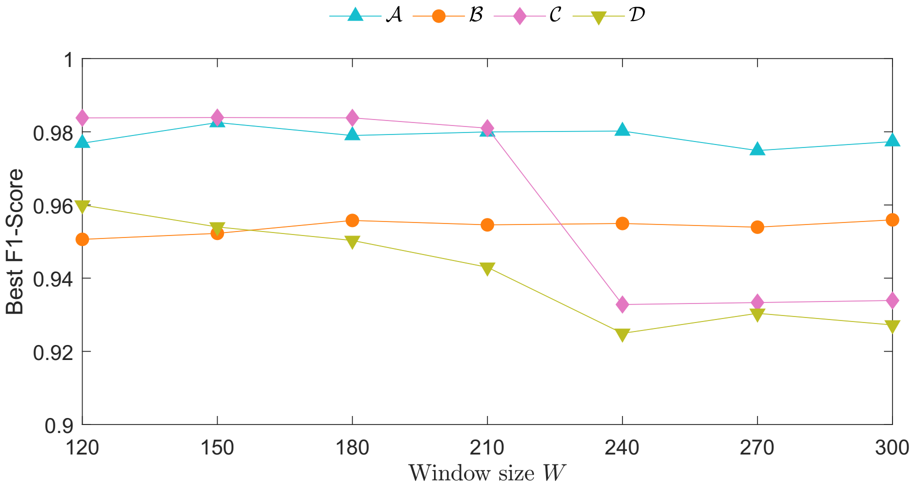


Fig. 10. Effect of the window size W.

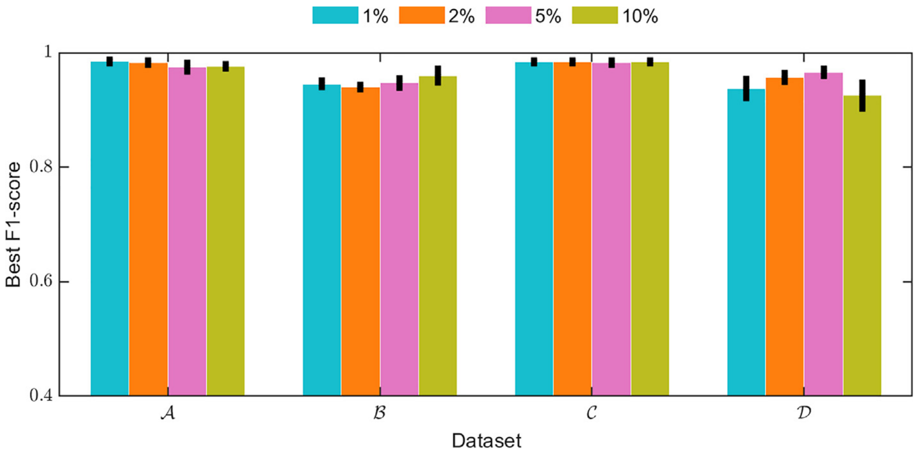


Fig. 11. Effect of the proportion of anomalous windows in training data.

its goal is alleviating the problem of anomaly data contamination to some extent. The $\mathbf{z}_r$ and $\mathbf{z}_{pp}$ are applied to train the encoder and generator of the model in an adversarial manner and further improve the generative ability of the model. In order to study the effectiveness of each module, seven possible variants are considered in the evaluation:

- VAE (None): None of three modules, which is equivalent to train the encoder and the generator individually with the original reconstructed term and regularized term;
- SaVAE ($\mathbf{z_r}$ only): This variant only uses the adversarial term $\mathbf{z_r}$ to train the model with the original reconstructed term and regularized term;
- SaVAE ($\mathbf{z_{pp}}$ only): This variant is similar to the variant above and only applies the adversarial term $\mathbf{z_{pp}}$ to train the model;
- SaVAE ($\mathbf{z_{pp}}$ & $\mathbf{z_r}$): This variant contains both $\mathbf{z_{pp}}$ and $\mathbf{z_r}$ modules. Similarly, the training data has no corresponding labels. In the model training, we employ the original reconstructed term and regularized term in ELBO of the standard VAE;
- VAE-SR (SR only): This variant only has the SR module, without any adversarial term. Therefore, this variant is equally to train the encoder and generator of the model with the modified reconstructed and regularized terms, separately;
- SaVAE-SR (SR & $\mathbf{z_r}$): This variant has the SR module and an adversarial module $\mathbf{z_r}$;
- SaVAE-SR (SR & $\mathbf{z_{pp}}$): This variant has the SR and another adversarial module $\mathbf{z_{pp}}$;

We present the best F1-score of SaVAE and seven possible variants in Fig. 12. It is apparent that the SR contributes most of the improvement over other modules. The contributions of the two adversarial terms are relatively small. The goal of employing the SR is to find out the most significant anomalies and remove the effect of them on model learning.

To study the effect of anomaly data contamination for the performance of the model and demonstrate the effectiveness of labels produced by the SR algorithm, we also train the model with 0%, 10%, 50%, and 100% labeled anomalies respectively, where different portions of labeled anomalies are obtained by sampling randomly in all anomalous points. The corresponding best F1-scores is illustrated in Fig. 13. Obviously, more labels are applied to train the model and higher results are obtained, which indicates the importance of labels for the learning of the model. However, our SaVAE-SR method can obtain approximately equal results with that of using 100% true labels to train the model, which demonstrates the effectiveness of the proposed SaVAE-SR. However, there is an exception for dataset $\mathscr{C}$. When using the 100% true labels to train the model, the obtained results are worse and unstable. The possible reason is that the dataset $\mathscr{C}$ is a relatively stable time series as shown in Fig.1 so that excessively clean data instead makes the model learns the relatively simple pattern and cannot generalize very well for the unobservable testing data. Instead, the SR technique used in our model can find the most significant anomalies so that the SaVAE-SR model provides superior and robust performance.

5.3. Discussion

We have verified the effectiveness of the proposed SaVAE-SR through extensive experiments conducted on five publicly available datasets and also demonstrated that the method is relatively

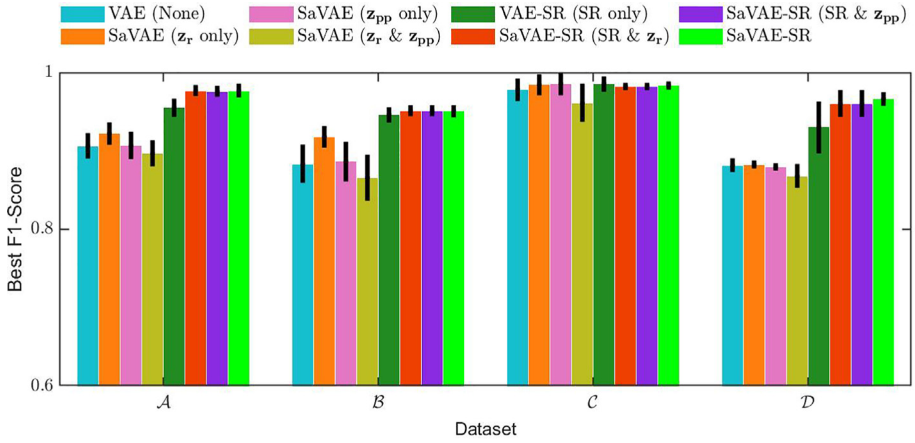


Fig. 12. The best F1-score of SaVAE-SR and seven possible variants.

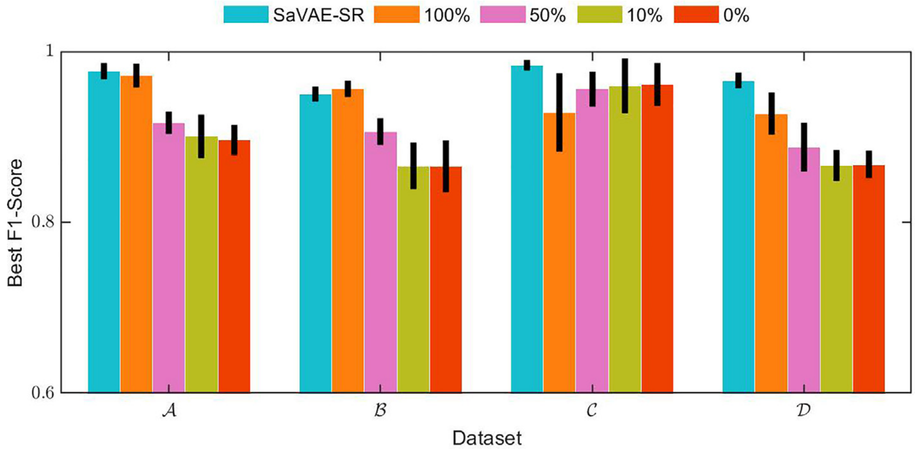


Fig. 13. The best F1-score of SaVAE-SR using the labels provided by the SR and the 0%, 10%, 50%, 100% true label, respectively.

insensitive to the variation of several critical parameters. The proposed SaVAE-SR far surpasses several state-of-the-art baselines on datasets derived from real applications, which benefits from the following two factors. The first is adopting the SR technique to identify the most significant anomalies and produce pseudo-labels for unlabeled training data. The VAE model with a modified loss function can leverage label information and remove the influence of these most significant anomalies on modeling the normal data distribution. The second is that self-adversarially training the encoder and generator of the VAE model combines the respective strengths of the VAE model and adversarial training, which greatly improves the learning ability of both the encoder and generator so that the model learns the complex data distribution better. Nevertheless, the performance of our method on synthetic data is not the best. We think there are two possible directions for improvement. As illustrated in the experiments, due to explicitly coding the time information, the performance of Bagel is outstanding on synthetic datasets A3 and A4 which consist mostly of seasonal signals. This means it is necessary to fully consider the time information, especially for signals with periodicity. The second possible direction is to leverage prior knowledge whenever possible. Although the SaVAE-SR method rarely requires prior knowledge like whether there are periodicity and trend, etc., if some prior knowledge is available, considering it when building the framework of the model or inputs of the model may further improve our model.

6. Conclusion

In this work, we have developed a novel unsupervised approach called Self-adversarial Variational Autoencoder with Spectral Residual (SaVAE-SR) for time series anomaly detection. To alleviate the problem of anomaly data contamination encountered in many previous unsupervised anomaly detection techniques, we employ the spectral residual technique to find the most significant anomalies and provide pseudo-labels for unlabeled training data. As shown in the experiments, this step provides a significant performance improvement. And then we have combined the VAE with the modified ELBO to leverage label information and adversarial training in a very simple yet efficient manner. The overall model consists of an encoder and a generator. The encoder is not only trained to model the approximate posterior of the latent variables but also plays the role of a discriminator for distinguishing real samples in training data from fake samples produced by the generator in latent space. The generator is trained to model the generative process of the data, and at the same time is trained to generate samples to deceive the encoder. The learning objective of the encoder and the generator forms a minimax two-player game, while the reconstruction term of the standard VAE model builds the bridge between the encoder and the generator. Therefore, the proposed method retains the strengths of VAE and adversarial training. Extensive experiments on five publicly available datasets demonstrate that our method is significantly superior to five state-of-the-art baselines.

CRediT authorship contribution statement

Yunxiao Liu: Conceptualization, Methodology, Software, Visualization, Writing - original draft, Writing - review & editing. **Youfang Lin:** Supervision, Writing - review & editing, Validation. **QinFeng Xiao:** Visualization, Software, Data curation. **Ganghui Hu:** Writing - review & editing. **Jing Wang:** Supervision, Investigation, Writing - review & editing.

Declaration of Competing Interest

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

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (2021JBM007) and the National Natural Science Foundation of China under Grant 61603029.

References

[1] Junshui Ma, Simon Perkins, Time-series novelty detection using one-class support vector machines, in: Proceedings of the International Joint Conference on Neural Networks, 2003, vol. 3, IEEE, 2003, pp. 1741–1745..

[2] Rui Zhang, Shaoyan Zhang, Sethuraman Muthuraman, Jianmin Jiang, One class support vector machine for anomaly detection in the communication network performance data, in: Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications, Citeseer, 2007, pp. 31–37..

[3] Zeyan Li, Wenxiao Chen, Dan Pei, Robust and unsupervised kpi anomaly detection based on conditional variational autoencoder, in: 2018 IEEE 37th International Performance Computing and Communications Conference (IPCCC), IEEE, 2018, pp. 1–9.

[4] Haowen Xu, Wenxiao Chen, Nengwen Zhao, Zeyan Li, Jiahao Bu, Zhihan Li, Ying Liu, Youjian Zhao, Dan Pei, Yang Feng, et al., Unsupervised anomaly detection via variational auto-encoder for seasonal kpis in web applications, in: Proceedings of the 2018 World Wide Web Conference, pages 187–196. International World Wide Web Conferences Steering Committee, 2018..

[5] Dapeng Liu, Youjian Zhao, Haowen Xu, Yongqian Sun, Dan Pei, Jiao Luo, Xiaowei Jing, Mei Feng, Opprentice: Towards practical and automatic anomaly detection through machine learning, in: Proceedings of the 2015 Internet Measurement Conference, ACM, 2015, pp. 211–224..

[6] Leonid Portnoy, Intrusion detection with unlabeled data using clustering (Ph.D. thesis), Columbia University, 2000..

[7] Luong Ha Nguyen, James-A. Goulet, Anomaly detection with the switching kalman filter for structural health monitoring, Struct. Control Health Monit. 25 (4) (2018), e2136.

[8] Bin Zhou, Shenghua Liu, Bryan Hooi, Xueqi Cheng, Jing Ye, Beatgan: anomalous rhythm detection using adversarially generated time series, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 4433–4439.

[9] Johannes Jurgovsky, Michael Granitzer, Konstantin Ziegler, Sylvie Calabretto, Pierre-Edouard Portier, Liyun He-Guelton, Olivier Caelen, Sequence classification for credit-card fraud detection, Expert Syst. Appl. 100 (2018) 234–245.

[10] Nikolay Laptev, Saeed Amizadeh, Ian Flint, Generic and scalable framework for automated time-series anomaly detection, in: Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 1939–1947.

[11] Hansheng Ren, Bixiong Xu, Yujing Wang, Chao Yi, Congrui Huang, Xiaoyu Kou, Tony Xing, Mao Yang, Jie Tong, Qi Zhang, Time-series anomaly detection service at microsoft, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 3009–3017..

[12] Jinwon An, Sungzoon Cho, Variational autoencoder based anomaly detection using reconstruction probability, Special Lecture on IE 2 (1) (2015).

[13] Diederik P Kingma, Max Welling, An introduction to variational autoencoders, Found. Trends Mach. Learn. 12 (4) (2019) 307–392.

[14] Mihaela Rosca, Balaji Lakshminarayanan, Shakir Mohamed, Distribution matching in variational inference. arXiv preprint arXiv:1802.06847, 2018..

[15] Aditya Grover, Manik Dhar, Stefano Ermon, Flow-gan: Combining maximum likelihood and adversarial learning in generative models, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018..

[16] Anthony J. Fox, Outliers in time series, J. Roy. Stat. Soc.: Ser. B (Methodol.) 34 (3) (1972) 350–363.

[17] Owen Vallis, Jordan Hochenbaum, Arun Kejariwal, A novel technique for long-term anomaly detection in the cloud, in: 6th {*USENIX*} Workshop on Hot Topics in Cloud Computing (HotCloud 14), 2014..

[18] Suk-Bok Lee, Dan Pei, MohammadTaghi Hajiaghayi, Ioannis Pefkianakis, Songwu Lu, He Yan, Zihui Ge, Jennifer Yates, Mario Kosseifi, Threshold compression for 3g scalable monitoring, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 1350–1358..

[19] Yingying Chen, Ratul Mahajan, Baskar Sridharan, Zhi-Li Zhang, A provider-side view of web search response time, in: ACM SIGCOMM Computer Communication Review, vol. 43, ACM, 2013, pp. 243–254..

[20] David R. Choffnes, Fabián E. Bustamante, Zihui Ge, Crowdsourcing service-level network event monitoring, in: Proceedings of the ACM SIGCOMM 2010 Conference, 2010, pp. 387–398.

[21] Balachander Krishnamurthy, Subhabrata Sen, Yin Zhang, Yan Chen, Sketch-based change detection: methods, evaluation, and applications, in: Proceedings of the 3rd ACM SIGCOMM conference on Internet measurement, 2003, pp. 234–247.

[22] He Yan, Ashley Flavel, Zihui Ge, Alexandre Gerber, Dan Massey, Christos Papadopoulos, Hiren Shah, Jennifer Yates, Argus: end-to-end service anomaly detection and localization from an isp's point of view, in: 2012 Proceedings IEEE INFOCOM, IEEE, 2012, pp. 2756–2760..

[23] Eduardo H.M. Pena, Marcos V.O. de Assis, Mario Lemes Proença, Anomaly detection using forecasting methods arima and hwds, in: 2013 32nd International Conference of the Chilean Computer Science Society (SCCC), IEEE, 2013, pp. 63–66.

[24] Jingxiang Qi, Yanjie Chu, Liang He, Iterative anomaly detection algorithm based on time series analysis, in: 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), IEEE, 2018, pp. 548–552.

[25] Alban Siffer, Pierre-Alain Fouque, Alexandre Termier, Christine Largouet, Anomaly detection in streams with extreme value theory, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 1067–1075.

[26] Jordan Hochenbaum, Owen S. Vallis, Arun Kejariwal, Automatic anomaly detection in the cloud via statistical learning. arXiv preprint arXiv:1704.07706, 2017..

[27] Sunav Choudhary, Gaurush Hiranandani, Shiv Kumar Saini, Sparse decomposition for time series forecasting and anomaly detection, in: Proceedings of the 2018 SIAM International Conference on Data Mining, SIAM, 2018, pp. 522–530.

[28] Romain Fontugne, Pierre Borgnat, Patrice Abry, Kensuke Fukuda, Mawilab: combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking, in: Proceedings of the 6th International COnference, 2010, pp. 1–12.

[29] Shashank Shanbhag, Tilman Wolf, Accurate anomaly detection through parallelism, IEEE Network 23 (1) (2009) 22–28.

[30] Nga Nguyen Thi, Van Loi Cao, Nhien-An Le-Khac, One-class collective anomaly detection based on long short-term memory recurrent neural networks. arXiv preprint arXiv:1802.00324, 2018..

[31] Jingyu Wang, Yuhan Jing, Qi Qi, Tongtong Feng, Jianxin Liao, Alsr: an adaptive label screening and relearning approach for interval-oriented anomaly detection, Expert Syst. Appl. 136 (2019) 94–104.

[32] Sudipto Guha, Nina Mishra, Gourav Roy, Okke Schrijvers, Robust random cut forest based anomaly detection on streams, in: Proceedings of The 33rd International Conference on Machine Learning, vol. 48, 2016, pp. 2712–2721..

[33] Leo Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.

[34] Mingyan Teng, Anomaly detection on time series, in: 2010 IEEE International Conference on Progress in Informatics and Computing, vol. 1, IEEE, 2010, pp. 603–608..

[35] Tolga Ergen, Ali Hassan Mirza, and Suleyman Serdar Kozat. Unsupervised and semi-supervised anomaly detection with lstm neural networks. arXiv preprint arXiv:1710.09207, 2017..

[36] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, Tom Soderstrom, Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding, in: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining, 2018, pp. 387–395.

[37] Shuyu Lin, Ronald Clark, Robert Birke, Sandro Schönborn, Niki Trigoni, Stephen Roberts, Anomaly detection for time series using vae-lstm hybrid model, in: ICASSP 2020–2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2020, pp. 4322–4326.

[38] Md Abul Bashar, Richi Nayak, Tanogan: time series anomaly detection with generative adversarial networks. In 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 1778–1785..

[39] Dan Li, Dacheng Chen, Baihong Jin, Lei Shi, Jonathan Goh, See-Kiong Ng, Mad-gan: multivariate anomaly detection for time series data with generative adversarial networks, in: International Conference on Artificial Neural Networks, Springer, 2019, pp. 703–716..

[40] Dan Li, Dacheng Chen, Jonathan Goh, See-kiong Ng, Anomaly detection with generative adversarial networks for multivariate time series. arXiv preprint arXiv:1809.04758, 2018..

[41] Su. Ya, Youjian Zhao, Chenhao Niu, Rong Liu, Wei Sun, Dan Pei, Robust anomaly detection for multivariate time series through stochastic recurrent neural network, in: Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2019, pp. 2828–2837.

[42] Yifan Guo, Weixian Liao, Qianlong Wang, Yu. Lixing, Tianxi Ji, Pan Li, Multidimensional time series anomaly detection: a gru-based gaussian mixture variational autoencoder approach, Asian Conference on Machine Learning 95 (2018) 97–112.

[43] Bernhard Schölkopf, Robert C. Williamson, Alex J. Smola, John Shawe-Taylor, John C. Platt, Support vector method for novelty detection, in: Advances in Neural Information Processing Systems, 2000, pp. 582–588..

[44] Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander, Lof: Identifying density-based local outliers, in: Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, 2000, pp. 93–104.

[45] Ian Goodfellow, Nips 2016 tutorial: Generative adversarial networks. arXiv preprint arXiv:1701.00160, 2016..

[46] Alexander Geiger, Dongyu Liu, Sarah Alnegheimish, Alfredo Cuesta-Infante, Kalyan Veeramachaneni, Tadgan: time series anomaly detection using generative adversarial networks, in: 2020 IEEE International Conference on Big Data (IEEE BigData), IEEE, 2020..

[47] Tung Kieu, Bin Yang, Chenjuan Guo, Christian S. Jensen, Outlier detection for time series with recurrent autoencoder ensembles, in: Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, 2019, pp. 2725–2732.

[48] Chunyong Yin, Sun Zhang, Jin Wang, Neal N. Xiong, Anomaly detection based on convolutional recurrent autoencoder for iot time series, IEEE Trans. Syst. Man Cybern.: Syst. (2020) 1–11..

[49] Julien Audibert, Pietro Michiardi, Frédéric Guyard, Sébastien Marti, Maria A. Zuluaga, Usad: unsupervised anomaly detection on multivariate time series, in: Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2020, pp. 3395–3404.

[50] Longyuan Li, Junchi Yan, Haiyang Wang, Yaohui Jin, Anomaly detection of time series with smoothness-inducing sequential variational auto-encoder, IEEE Trans. Neural Networks Learn. Syst. 32 (3) (2021) 1177–1191.

[51] Xuhong Wang, Ying Du, Shijie Lin, Ping Cui, Yuntian Shen, Yupu Yang, advae: a self-adversarial variational autoencoder with gaussian anomaly prior knowledge for anomaly detection, Knowl.-Based Syst. 190 (2020) 105187.

[52] Xiaodi Hou, Liqing Zhang, Saliency detection: a spectral residual approach, in: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2007, pp. 1–8..

[53] Diederik P. Kingma, Max Welling, Auto-encoding variational bayes. arXiv preprint arXiv:1312.6114, 2013..

[54] Danilo Jimenez Rezende, Shakir Mohamed, Daan Wierstra, Stochastic backpropagation and approximate inference in deep generative models. arXiv preprint arXiv:1401.4082, 2014..

[55] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680..

[56] Junbo Zhao, Michael Mathieu, Yann LeCun, Energy-based generative adversarial networks, in: 5th International Conference on Learning Representations, ICLR 2017, 2017..

[57] Huaibo Huang, Ran He, Zhenan Sun, Tieniu Tan, et al., Introvae: introspective variational autoencoders for photographic image synthesis, in: Advances in Neural Information Processing Systems, 2018, pp. 52–63..

[58] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, Soumith Chintala, Pytorch: an imperative style, high-performance deep learning library, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, Advances in Neural Information Processing Systems, vol. 32. Curran Associates Inc, 2019..

[59] Alexander Lavin, Subutai Ahmad, Evaluating real-time anomaly detection algorithms–the numenta anomaly benchmark, in: 2015 IEEE 14th International Conference on Machine Learning and Applications (ICMLA), IEEE, 2015, pp. 38–44.

Yunxiao Liu received the B.S. degree in information and computing science from Beijing Jiaotong University, Beijing, China, in 2015, where she is currently pursuing the Ph.D. degree with the School of Computer and Information Technology. Her current research interests include time series analysis and anomaly detection.

Youfang Lin received the Ph.D. degree in singnal and information processing from Beijing Jiaotong University, Beijing, China, in 2003. He is a Professor with the School of Computer and Information Technology, Beijing Jiaotong University. His main fields of expertise and current research interests include intelligent systems, complex networks, and time series data mining.

Qinfeng Xiao is currently pursuing the master's degree at Institute of Network Science and Intelligent Systems, Beijing Jiaotong University, China. His research interests include anomaly detection and self-supervised learning.

Ganghui Hu is currently pursuing the master's degree at Institute of Network Science and Intellgent Systems in School of Computer Science and Technology, Beijing Jiaotong University, China. His research interests include time series anomaly detection.

Jing Wang received the Ph.D. degree in statistics from Beijing Jiaotong University, Beijing, China, in 2015. She is an Associate Professor with the School of Computer and Information Technology, Beijing Jiaotong University. Her current research interests focus on time series analysis and mining, machine learning and its applications and anomaly detection.