GAN-Based Anomaly Detection for Multivariate Time Series Using Polluted Training Set

Bowen Du, Xuanxuan Sun, Junchen Ye, Ke Cheng, Jingyuan Wang, and Leilei Sun

Abstract—Multivariate time series anomaly detection has great potentials in many practical applications such as structural health monitoring, intelligent operation and maintenance, quantitative trading, *etc.* Extreme unbalanced training set and noise interference make it challenging to accurately capture the distribution of normal data and then detect anomalies. Recently, dozens of AutoEncoder (AE) and Generative Adversarial Network (GAN) based methods have been proposed to learn the latent representation of normal data and then detect anomalies based on reconstruction error. However, existing AE-based approaches are lack of effective regularization method specially designed for anomaly detection tasks thus easily overfitting while GAN-based approaches are mostly trained under the hypothesis of pollution-free training set, which means the training set is all composed of normal samples and that is hard to satisfy in practice. To tackle these problems, in this paper we propose a GAN based anomaly detection method for multivariate time series named FGANomaly (letter F is for Filter). The core idea is to filter possible anomalous samples with pseudo-labels before training the discriminator thus to capture the distribution of normal data as precise as possible. In addition, we design a novel training objective for the generator, which leads the generator to concentrate more on plausible normal data and ignore anomalies. We conducted comprehensive experiments on four public datasets, and the experimental results show the superiority of our method over baselines in both performance and robustness.

Index Terms—Anomaly Detection, Generative Adversarial Networks, Multivariate Time Series, Pseudo-label.

1 Introduction

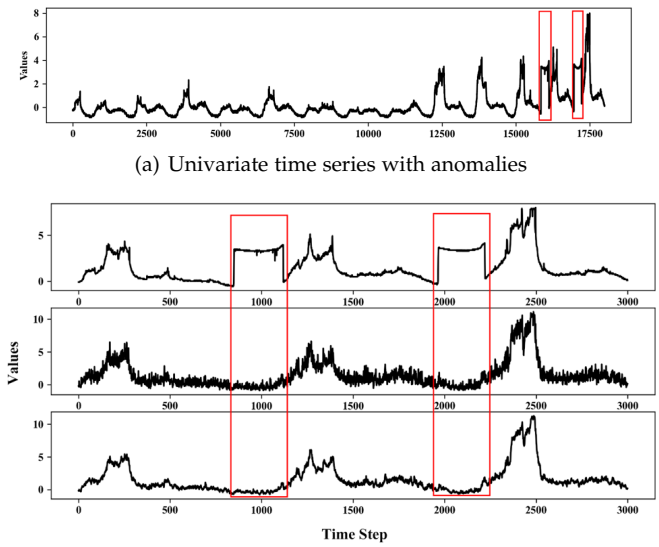
Multivariate time series is a type of real-world data of great volume, which is usually collected by monitoring system and contains significant run-time information of the monitored subject. Generally, anomalies in the measured data reflect abnormal states of the system such as structural failures in structural health monitoring [1], intrusions in intelligent operation and maintenance [2], frauds in finance [3], *etc*. These abnormal states could bring great economic losses even casualties. Therefore, the establishment of anomaly detection system which could detect anomalies in multivariate time series accurately and fast has drawn more and more interest from both of academia and industry.

In a time series, anomalies are defined as observations deviate severely from their expected values. Historically, the expected values are determined by system monitoring experts who have extensive experience thus clear about the boundaries of normal values [4], [5]. The experts firstly establish normal behavior thresholds for every feature to be measured, then measurements exceed their corresponding expert-defined thresholds will be considered as anomalies. However, because of the scale and complexity of the monitored subject, the number of features to be measured has increased dramatically over time, which makes it no longer simple and effective to detect anomalies with traditional expert-defined methods. As shown in Figure 1, compared with univariate time series, anomaly detection in multivariate time series is more complex for the reason of complicated correlations between different dimensions. Under this circumstance, automated and scalable anomaly detection methods are in urgent need.

The rapid development of data acquisition and storage technology provides abundant raw materials for machine learning algorithms, which leads to the boom of machine learning based anomaly detection methods. In the past years, many approaches have been proposed to address the problem of anomaly detection. Amongst them, the most commonly used methods are clustering based models and one class classification based models. Clustering based models such as K-means and density peak clustering consider the observations far away from any of the clustering centers as anomalies [6], [7]. One class classification methods such as one class SVM and isolation forest only model the distribution of normal data, then consider the observations that do not follow this distribution as anomalies [8], [9]. These data driven anomaly detection methods are more intelligent, but they are still not competent to large scale data due to the curse of dimensionality and sub-optimal performance. Most recently, the unsupervised learning methods based on reconstruction has received a lot of attention [10], [11], [12], [13].

Reconstruction based anomaly detection methods for multivariate time series usually have two stages: 1) data decomposition and reconstruction. 2) anomaly scores calculation and anomalies detection. In the first stage, a model is trained to decompose the input data and then reconstruct them precisely. During this process, the model learns the patterns of normal data thus is able to reconstruct them precisely but fail to reconstruct anomalous data well. The

- *B. Du, X. Sun, J. Ye, K. Cheng, J. Wang, and L. Sun are with the State Key Laboratory of Software Development Environment (SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing 100191, China. (corresponding author: J. Ye, E-mail: yjchen@buaa.edu.cn.)*
- *B. Du, J. Wang, and L. Sun are also with the Peng Cheng Laboratory, Shenzhen 518055, China.*



(a) Univariate time series with anomalies

(b) Multivariate time series with anomalies

Fig. 1. Compared with univariate time series, anomaly detection in multivariate time series is more complex for the reason of complicated correlations between different dimensions.

distance between each observation and its reconstructed counterpart is then calculated as anomaly score. Generally, the farther the distance is, the more likely the observation is to be abnormal. By setting a threshold, the abnormal samples can be detected. Matrix factorization is a kind of classic dimension reduction technologies which can be also used for data reconstruction and anomaly detection. Non-negative matrix factorization (NMF) [11] and principal component analysis (PCA) [14] are widely used for anomaly detection. However, they are linear models which are not suitable for complex non-linear data while deep learning based methods are more powerful and suitable. Among various deep learning methods, recurrent neural networks (RNNs) [13] are very popular for that they are proper to deal with sequential data like time series. However, RNNs incur in high costs associated to time for the reason of its structural characteristics. Other deep learning based models include autoencoder based and variational autoencoder based models such as Deep Autoencoding Gaussian Mixture Model (DAGMM) [15], Donut [12] and OmniAnomaly [16] which also have limitations. They are lack of effective regularization method specially designed for anomaly detection tasks, which makes them learn the patterns of abnormal data as well thus fail to detect anomalies precisely. For the reason of sample generation capability, GAN-based anomaly detection methods are rising quickly, such as GANomaly [17], MAD-GAN [18], LSTM-based VAE-GAN [19] and BeatGAN [20]. The architecture of GAN is able to capture the distribution of training data, which provides global regularization information during the training process thus to relieve overfitting [20]. But that only makes sense when the training set is pollution-free, which means there are only normal samples in the training set. When the training set is polluted with anomalous samples, the models would not only capture the distribution of normal data but also the distribution of abnormal data.

To address the problems aforementioned, in this paper we propose a novel GAN-based unsupervised anomaly detection method for multivariate time series named FGANomaly. The intuition behind FGANomaly is that before training the discriminator of GAN, filter the plausible abnormal samples at first using pseudo-labels. When training the generator of GAN which is in the form of encoder-decoder, replace MSE loss with a special designed loss as training objective. In summary, the main contributions of this paper are:

- To the best knowledge of us, this is the first research conducting multivariate time series anomaly detection with GAN but not under the hypothesis of pollution-free training set.
- We propose a pseudo-label generation method which enables the GAN architecture to filter possible anomalous samples thus accurately capture the distribution of normal data even the training set is polluted with abnormal data. We name the novel GAN architecture Filter GAN.
- We design a novel training objective for the generator named Adaptive Weighted Loss (AdaWL), which dynamically assigns weights to different points during the training process according to their reconstruction errors. With this training objective, the model could concentrate more on plausible normal data and thus relieve overfitting.

The rest of this paper is organized as follows. Section 2 discusses related unsupervised methods for detecting anomalies in multivariate time series. Section 3 discusses some preliminaries for this task. And in section 4, the details of our method will be described. Section 5 and 6 describe the experiments we conducted and the conclusions we draw. Our code has been published on GitHub for better reproducibility[1].

2 Related Works

Tremendous effort has been devoted to unsupervised anomaly detection, the existing methods can be grouped up into three main categories, which are: 1) one class classification based methods, 2) clustering based methods, 3) reconstruction based methods.

One class classification based anomaly detection methods only modelling the distribution of normal data, the observations that do not follow this distribution are considered as anomalies. Under this framework, what the algorithms need to do is to learn a discriminative boundary that surrounds the normal samples. One class SVM [21] and isolation forest [9] are typical models. Considering that lots of deep learning based anomaly detection approaches are not trained on an anomaly detection based objective, Ruff et al. introduces a new anomaly detection method—Deep Support Vector Data Description which is trained on an anomaly detection based objective and it combines the advantages both of deep learning an one class classification [22]. Miao et al. propose a distributed online one class

1. https://github.com/sxxmason/FGANomaly

SVM for anomaly detection to address the problem of anomaly detection for distributed data [23]. To reduce the false alarm rate, Khreich et al. proposes a new anomaly detection system which combines the temporal information of system with an one class support vector machine detector [8]. Though this kind of methods are easy to understand and implement, they usually suffer from sub-optimal performance due to the curse of dimensionality as the number of dimensions grows higher.

Clustering based anomaly detection methods usually consider the observations far away from any of the clustering centers as anomalies. This kind of methods are essentially doing density estimation and samples with lower probability density than threshold are considered as anomalies. Multivariate Gaussian Model, Gaussian Mixture Model and K-means are typical models. Emadi et al. detect anomalies in wireless sensor networks using a density-based spatial clustering of applications with noise (DBSCAN) algorithm, this algorithm detects the points in regions with low density as anomalies [24]. Harish et al. presents a network anomaly detection method based on fuzzy clustering [25]. To address the problem of detecting anomalous multi-elemental geochemical footprints of mineral deposits, Ghezelbash et al. proposes genetic K-means clustering (GKMC) algorithm, which is a hybrid genetic algorithm-based technique [26]. To make use of the temporal and variable relationships between multivariate time series, Li et al. introduces an extended Fuzzy C-Means clustering to reveal the available structure within multivariate time series data by taking the amplitude and the shape of multivariate time series into consideration [27]. For the reason that the structures of clustering based models are too simple to deal with complex data, they have little potential to accommodate further anomaly detection tasks.

Reconstruction based methods assume that anomalies can not be effectively reconstructed from low dimensional representations when the ratio of abnormal samples in original data is quite small. Therefore, the anomaly score can be defined as reconstruction error, the larger the anomaly score is, the more likely the observation is to be abnormal. Classic method of this category is PCA [14]. Plain PCA only has the ability of linear projection while kernel PCA [28] can carry out non-linear projections. For more complex data, deep learning models have been proposed [12], [13], [15], [16]. However, deep autoencoder based models are short of effective regularization method for anomaly detection tasks while GAN-based models introduce adversarial regularization to relieve overfitting [17], [18], [19], [20], [29]. But when the training set is polluted with anomalous samples, the conventional GAN-based models would not only capture the distribution of normal data but also the distribution of abnormal data. To tackle the aforementioned problems, we have designed two special components with which FGANomaly can relieve overfitting greatly and detect anomalies precisely.

Existing reconstruction based methods provide a paradigm that can accurately identify anomalies even under unsupervised situations. Meanwhile, with the powerful representation learning ability of deep learning methods, it is possible to deal with the tremendous data and mining useful information to detect anomalies. However, it is still faced with some problems which are waiting to be solved such as easily overfitting with abnormal patterns and non-robustness when trained with polluted data.

3 PRELIMINARIES

In this section, the notations throughout the manuscript are introduced first, and then the task of multivariate time series anomaly detection is defined.

Definition 1. **Observation (or Sample/Instance).** An observation $\boldsymbol{x}$ is a data point collected by the monitoring system at a certain time. For univariate time series, $\boldsymbol{x}$ degrades into a scalar while it is a vector of M-dimension under the scenario of multivariate time series.

Definition 2. **Multivariate time series.** A type of sequential data of length T containing successive observations collected at equal-space intervals, noted as $\mathcal{T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. The sampling interval between two consecutive observations ranges from seconds to hours.

Definition 3. **Sliding window.** A sliding window is a consecutive subsequence of length W sampled from $\mathcal{T}$, noted as $\mathcal{W}$. A time series could be transformed into a sequence of sliding windows: $\mathbb{W} = \{\mathcal{W}^1, \mathcal{W}^2, ..., \mathcal{W}^J\}$, $\mathcal{W}^j = \{\boldsymbol{x}_1^j, \boldsymbol{x}_2^j, ..., \boldsymbol{x}_W^j\}$, j is the serial number of current sliding window. For the sake of simplicity, j will be omitted in the following sections. If the labels are provided, $\mathcal{W}^j$ could be further divided into two sets: $\mathcal{S}_n^j$ and $\mathcal{S}_{an}^j$. The former consists of all the normal observations in $\mathcal{W}^j$ while the latter consists of all the anomalous observations.

Definition 4. **Reconstruction.** Reconstruction based anomaly detection methods first decompose the input data into their low dimensional representations (or codings): $\mathbb{H} = \{H^1, H^2, ..., H^J\}$, here $H^j = \{\boldsymbol{h}_1^j, \boldsymbol{h}_2^j, .., \boldsymbol{h}_W^j\}$. Then, the input data will be reconstructed with these codings as: $\tilde{\mathcal{W}}_j = \{\tilde{\boldsymbol{x}}_1^j, \tilde{\boldsymbol{x}}_2^j, ..., \tilde{\boldsymbol{x}}_W^j\}$.

Definition 5. **Reconstruction error.** The distance between the original observation and its corresponding reconstruction. Generally defined as:

$$d_w^j = ||\boldsymbol{x}_w^j - \tilde{\boldsymbol{x}}_w^j||. \tag{1}$$

Problem statement. Given a multivariate time series $\mathcal{T} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_T\}$. The objective of anomaly detection is to assign a label $y \in \{0, 1\}$ (0 for normal, 1 for abnormal) for each observation $\boldsymbol{x}_t \in \mathcal{T}$. To learn the correlations between an observation and its surrounding ones, the original sequence is usually transformed into a sequence of sliding windows $\mathbb{W} = \{\mathcal{W}^1, \mathcal{W}^2, ..., \mathcal{W}^J\}$. In order to detect anomalies, reconstruction based anomaly detection methods first decompose and reconstruct the input data as: $\widetilde{\mathbb{W}} = \{\tilde{\mathcal{W}}^1, \tilde{\mathcal{W}}^2, ..., \tilde{\mathcal{W}}^J\}$, then the reconstruction errors are calculated with the input data and reconstructions, which will be taken as anomaly

scores for the input data. Eventually, the observations with d greater than a given threshold th will be considered as anomalies.

4 METHODOLOGY

In this section, we present the details and implementation of FGANomaly.

4.1 Overview of Proposed Model

The overall architecture of FGANomaly is depicted in Figure 2. As shown in the figure, FGANomaly consists of three components: *generator*, *filter* and *discriminator*. Note that the generator is under the architecture of encoder-decoder. Let $\mathcal{X}$, $\mathcal{H}$ denote the feature space and latent vector space respectively. What the generator G needs to do is mapping the observations into lower dimensional space first and then reconstructs them into feature space:

$$f_{enc} : \mathcal{X} \to \mathcal{H}, \qquad f_{dec} : \mathcal{H} \to \mathcal{X}. \tag{2}$$

We implement the encoder f_{enc} with a one-layer bidirectional LSTM network followed with a linear projection, and the decoder f_{dec} with a symmetrical network. Conventionally, the original samples and reconstructed ones are then labeled as real and fake (considering that the generator is in the form of autoencoder, fake means reconstructed and real means original here) respectively to train the discriminator D. The discrimination mapping f_{dis} is:

$$f_{dis} : \mathcal{X} \to [0, 1], \tag{3}$$

it receives temporal sequences and returns classification results. We implement the discriminator with a step-wise feedforward network. Different from conventional models, in FGANomaly the original samples and reconstructed ones must be screened by the filter before they are sent to the discriminator. The details of filter will be described later. With these screened samples, the discriminator will be trained to distinguish fake samples from the real ones.

4.2 Pseudo-Label and Filter GAN

As deep generative models, GANs are able to capture the distribution of training set thus generate samples which follow this distribution. That is to say, the models will learn the knowledge about normal data and be able to generate normal samples if they are trained only with normal samples. Meanwhile, it is hard for them to generate abnormal samples since they have never met any of them. Under this scenario, the normal samples will be reconstructed at a relatively high accuracy while the abnormal ones cannot be reconstructed well and eventually be easily detected. But in practice it is hard and costly to build a training set only consists of normal samples. When the training set is polluted with abnormal samples the distribution captured by GAN is no longer accurate.

To address this problem, a plain idea is to abandon abnormal samples before training the discriminator. But labels are not available under the scenario of unsupervised learning, thus a pseudo-label generation method is needed. Considering that an observation with relatively larger reconstruction error will be considered as anomaly under the paradigm of reconstruction based anomaly detection method, we can generate pseudo-labels based on the reconstruction errors. Given input data $\mathcal{W} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_W\}$, the reconstruction errors $\boldsymbol{d} = \{d_1, d_2, ..., d_w\}$ can be obtained after the encoding and decoding processes. With denoting $\overline{d}$ and δ as the mean value and variance, the z-score z_w of d_w is calculated as:

$$z_w = \frac{d_w - \overline{d}}{\delta}. \tag{4}$$

Then, the abnormal probability $\hat{P}_{ab}$ of observation $\boldsymbol{x}_w$ is defined as:

$$\hat{P}_{ab}(\boldsymbol{x}_w) = \sigma(z_w) = \sigma(\frac{d_w - \overline{d}}{\delta}), \tag{5}$$

σ is the sigmoid function. As Equation 5 indicates, if the reconstruction error of an observation equals to $\overline{d}$, the probability of it to be abnormal is 0.5. But if the reconstruction error of an observation is larger than the mean value, it is more likely to be abnormal thus will be assigned a probability larger than 0.5. On the contrary, an observation whose reconstruction error is less than the average level is more likely to normal thus will be assigned a probability less than 0.5.

However, at the beginning of the training process, the model is not strong enough to learn efficient latent representations, so the reconstruction errors cannot reflect anomaly information precisely, a balance factor should be introduced. The probability of observation $\boldsymbol{x}_w$ to be abnormal is eventually defined as:

$$P_{ab}(\boldsymbol{x}_w) = \sigma[(\frac{d_w - \overline{d}}{\delta}) \cdot (1 - \frac{1}{f(n)})], \tag{6}$$

here, n denotes the number of current iteration. $f(n)$ is a function of n which meets the following requirements:

$$\begin{cases} f(n) = 1 & when \quad n = 1 \\ f(n) \to +\infty & when \quad n \to +\infty, \end{cases} \tag{7}$$

$f(n)$ could be $log(n - 1 + e)$, n, n^2. With this balance factor $P_{ab}(\boldsymbol{x}_w)$ would be adjusted dynamically according to current training iteration. The abnormal probabilities of all the observations are 0.5 when the training process just begins. As the training process goes on they are assigned with different abnormal probabilities. After $P_{ab}(\boldsymbol{x}_w)$ is obtained, observation $\boldsymbol{x}_w$ will be labeled as positive sample at the probability of $P_{ab}(\boldsymbol{x}_w)$ and negative sample at the probability of $1 - P_{ab}(\boldsymbol{x}_w)$.

With the generated pseudo-labels $\boldsymbol{y} = \{y_1, y_2, ..., y_W\}$, the training objective L_{dis} for the discriminator of Filter GAN is redefined as:

$$L_{dis} = -\frac{1}{W_n} \sum_{w=1}^{W} (1 - y_w) \cdot [logD(\boldsymbol{x}_w) + log[1 - D(G(\boldsymbol{x}_w))]], \tag{8}$$

W_n is the number of positive samples.

4.3 Adaptive Weighted Loss

Deep learning based anomaly detection methods take mean squared error as training objective to reconstruct the input data. It is a good choice when the training set is clean or slightly polluted with anomalous samples, because the

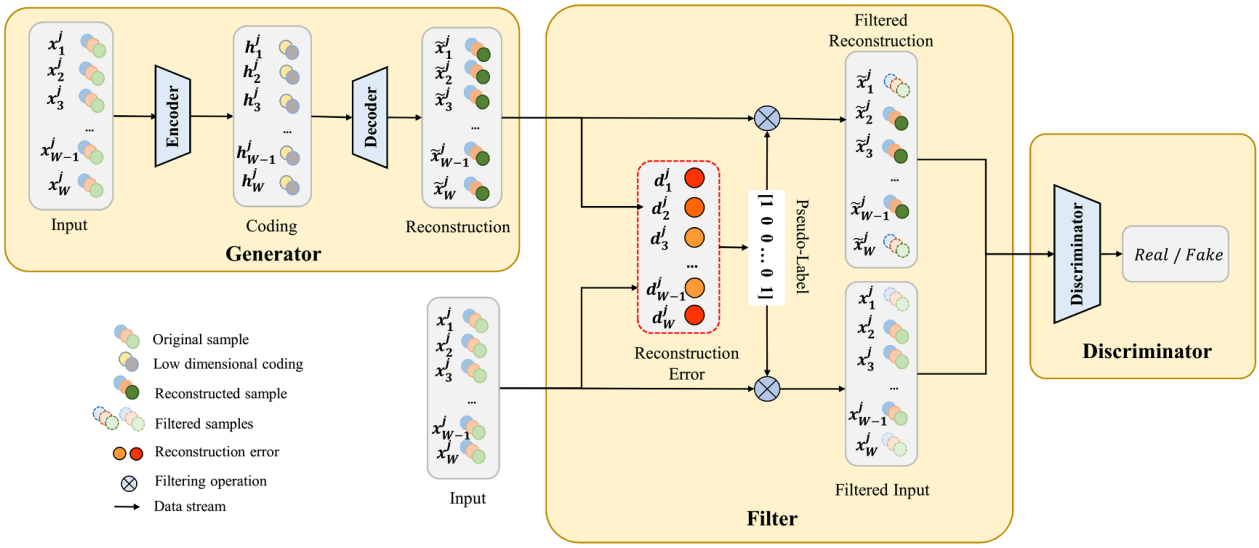
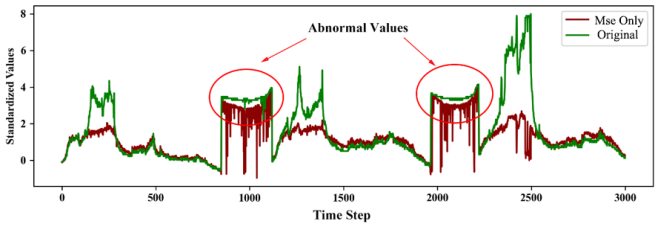
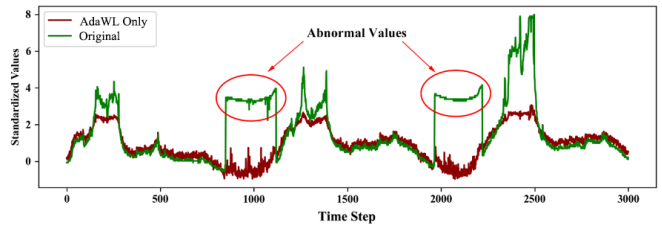


Fig. 2. Overview of the proposed model. It consists of three parts which are generator, filer and discriminator. The generator reconstructs the input data and the filer selects possible normal samples both from the input and reconstruction.The discriminator is then trained with these cleaned samples.



(a) Trained with MSE loss.



(b) Trained with AdaWL.

Fig. 3. A comparison between MSE loss and AdaWL. (a) The reconstruction result of a fully connected network autoencoder trained with MSE loss. (b) The reconstruction result of the same autoencoder trained with AdaWL. Compared two subfigures, we can find that the model is severely overfitting and reconstruct abnormal values precisely when trained with MSE loss. But when the training objective is changed to AdaWL, the overfitting phenomenon is relieved.

models could learn effective low dimensional representations of the training data. But when the training set contains relatively more anomalies, they would not only learn the representations of normal samples but also those of anomalous samples, which means the models are overfitting. Here we have a little case study shown in Figure 3. We trained two simple models on a multivariate time series with conventional MSE loss and AdaWL respectively. From the visualization of reconstruction results we can tell that it is easy for models to be overfitted with abnormal patterns when trained with MSE loss. For mere reconstruction tasks, the result shown in figure 3(a) may be better because its reconstruction error is less, but it is not good for anomaly detection tasks. In the task of anomaly detection, *a good model is supposed to reconstruct normal samples as precise as possible but fail to reconstruct abnormal samples well*. Inspired by this idea, we propose a novel reconstruction objective specially designed for anomaly detection tasks.

As previously discussed, given input data $\mathcal{W} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_W\}$, conventional training objective L_{rec} for reconstruction is:

$$L_{rec} = \frac{1}{W}\sum_{w=1}^{W} ||\boldsymbol{x}_w - \tilde{\boldsymbol{x}}_w||. \tag{9}$$

In Equation 9, each observation contributes to the loss equally. However, the ideal training objective in anomaly detection tasks is supposed to be:

$$L_{rec} = \frac{1}{|\mathcal{S}_n|}\sum_{w=1}^{W} ||\boldsymbol{x}_w - \tilde{\boldsymbol{x}}_w|| \cdot (1 - y_w), \tag{10}$$

where $|\mathcal{S}_n|$ is the number of elements in set $\mathcal{S}_n$. In Equation 10, only normal observations have contribution to the loss. When the model is trained with this objective by gradient descent, it can only learn the knowledge about normal patterns, thus avoid overfitting. But as aforementioned, labels are not available under the scenario of unsupervised learning, normal samples cannot be separated from abnormal ones. To address this problem, we propose an approximate function for Equation 10.

Considering that the larger the reconstruction error is, the more likely the observation is to be abnormal, so we can assign a weight for each observation before calculating the loss. The more likely the observation is to be normal, the larger the weight assigned to it is. Before back-propagation at each iteration, we get z-scores for all the observations.

Then, the weight $\hat{\alpha}_w$ for observation $\boldsymbol{x}_w$ is defined as:

$$\hat{\alpha}_w = \frac{e^{-z_w}}{\sum_k e^{-z_k}}, \tag{11}$$

z-score larger than 0 means the corresponding reconstruction error is larger than the average level, then the weight will be less than $\frac{1}{W}$. The larger the reconstruction error is, the less the weight will be. z-score less than 0 means the corresponding reconstruction error is less than the average level, then the weight will be larger than $\frac{1}{W}$. The less the reconstruction error is, the larger the weight will be. Similarly, a balance factor should be introduced. Thus, the weight of $\boldsymbol{x}_w$ is finally defined as:

$$\alpha_w = \frac{1}{Z} \cdot \frac{e^{-z_w}}{\sum_k e^{-z_k}} \cdot \frac{\sum_{k \neq w} e^{-z_k} + f(n) \cdot e^{-z_w}}{f(n) \cdot e^{-z_w}}, \tag{12}$$

Z is the normalization factor. When the training process just begins, $n = 1$ and $\alpha_w = \frac{1}{W}$, which means every observation contributes equally to the loss. As n increases to $+\infty$, α_w gets closer to Equation 11, each observation is assigned with different weights. Now, Equation 10 is approximated by:

$$L_{rec} = \sum_{w=1}^{W} ||\boldsymbol{x}_w - \tilde{\boldsymbol{x}}_w|| \cdot \alpha_w. \tag{13}$$

With this function as reconstruction objective, the model could concentrate more on plausible normal data and thus relieve overfitting.

4.4 Model Training

Given the training set $\mathbb{W} = \{\mathcal{W}^1, \mathcal{W}^2, ..., \mathcal{W}^J\}$, the generator and discriminator of FGANomaly are trained alternately. For discriminator, the training objective is defined in Equation 8. Under the framework of conventional GANs, the generator is trained only with adversarial loss:

$$L_{adv} = -\frac{1}{W} \sum_{w=1}^{W} logD[G(\boldsymbol{x}_w)], \tag{14}$$

which is not adequate for the generator to attend to the temporal correlations unique to time-series data. We trained the generator with both of L_{adv} and L_{rec}. Therefore, the training objective of the generator is:

$$L_{gen} = L_{rec} + \gamma \cdot L_{adv}, \tag{15}$$

γ is used to parameterize the trade-off between L_{rec} and L_{adv}. The training process is shown in algorithm 1.

Algorithm 1 Training algorithm

Input: Sequence of fragmented data $W = \{\mathcal{W}^1, ..., \mathcal{W}^J\}$, max epochs M, the rate of adversarial loss γ
Output: Trained G and D
Initialization: initialize all the parameters of G and D
$n \leftarrow 1, m \leftarrow 1$
repeat
  for $j \leftarrow 0$ **to** J **do**
    $\mathcal{W} \leftarrow G(\mathcal{W})$
    Get $\boldsymbol{d}$ based on Equation 1
    for all $\boldsymbol{x}_w$ **do**
      Get $P_{ab}(\boldsymbol{x}_w)$ based on Equation 6
      $\eta \leftarrow rand(0, 1)$
      if $\eta < P_{ab}(\boldsymbol{x}_w)$ **then**
        $y_w \leftarrow 1$
      else
        $y_w \leftarrow 0$
      end if
    end for
    $\boldsymbol{y} \leftarrow \{y_1, y_2, ..., y_w\}$
    Get L_{dis} based on Equation 8
    Update the parameters of D using L_{dis}
    for all $\boldsymbol{x}_w$ **do**
      Get α_w based on Equation 12
    end for
    $\boldsymbol{\alpha} \leftarrow \{\alpha_1, \alpha_2, ..., \alpha_w\}$
    Get L_{adv} and L_{rec} based on Equation 14 and 13
    $L_{gen} \leftarrow L_{rec} + \gamma \cdot L_{adv}$
    Update the parameters of G using L_{gen}
    $n \leftarrow n + 1$
  end for
  $m \leftarrow m + 1$
until m=M

5 Experiments

To evaluate the effectiveness and robustness of proposed method, comprehensive experiments were conducted. In this section, experimental setups are described firstly. Then, we compare the overview performance of FGANomaly with other classic anomaly detection methods and state-of-art methods. We also conducted ablation study to investigate how different components influence the performance of FGANomaly. The robustness of FGANomaly under varying noise ratio was verified by injecting noise into the training set. At last, we introduce a case study to get insights into the model.

5.1 Experimental Setup

5.1.1 Public Datasets

Four public datasets are used in our experiments. The main characteristics of the datasets are summarized in table 2, and here is a brief introduction for them.

Mars Science Laboratory (MSL) rover dataset and Soil Moisture Active Passive (SMAP) satellite dataset. Both of MSL and SMAP were collected from NASA [13]. Each dataset has been divided into two parts of training and testing subsets, and anomalies in testing subsets are expert-labeled.

Secure Water Treatment (SWaT) dataset. The data was collected from an operational test-bed for water treatment [31]. The original dataset was systematically generated from the test-bed over 7 days under normal situation and 4 days with varying attack scenarios. Since we want to verify the robustness of all the models when trained with polluted training set, the first 7 days data are abandoned and then the remaining data are divided into training set and testing set evenly, at which we conducted all the experiments.

The Water Distribution (WADI) dataset. WADI test-bed is an extension of the SWaT system. Similarly, WADI was collected over 16 days of continuous operation, of which 14 days were collected under normal operation and 2 days with attack scenarios. For the same reason, we also abandon

TABLE 1
Overview performance of all methods. Following the setting of [30]. Top: Precision(P), Recall(R) and F1 Score(F1) of MSL and SMAP with point-adjust. Middle: Metrics of SWaT subset both with and without point-adjust. Bottom: Metrics of WADI subset both with and without point-adjust.

	MSL			SMAP		
Metrics	P	R	F1	P	R	F1
PCA	0.93655	0.84007	0.88569	0.90691	0.57600	0.70453
NMF	0.85891	0.89209	0.87519	0.97293	0.55324	0.70538
AE	0.87627	0.85359	0.86478	0.59024	0.91579	0.71780
BeatGAN	0.88445	0.86930	0.87681	0.75878	0.91162	0.82821
USAD	0.89396	0.87619	0.88499	0.83925	0.88052	0.85939
FGANomaly	0.90052	0.93600	**0.91792**	0.76093	0.99925	**0.86395**

	$SWaT_{subset}$					
	With Point-Adjust			Without Point-Adjust		
Metrics	P	R	F1	P	R	F1
PCA	0.98554	0.92723	0.95550	0.38665	0.32885	0.35542
NMF	0.94916	0.98782	0.96810	0.46683	0.56875	0.51278
AE	0.99509	0.88869	0.93889	0.92842	0.75351	0.83187
BeatGAN	0.99277	0.91601	0.95285	0.97172	0.79201	0.87271
USAD	0.98392	0.96666	0.97522	0.97516	0.76784	0.85917
FGANomaly	0.97735	0.98862	**0.98295**	0.98506	0.79538	**0.88011**

	$WADI_{subset}$					
	With Point-Adjust			Without Point-Adjust		
Metrics	P	R	F1	P	R	F1
PCA	0.56650	0.49581	0.52880	0.06900	0.21664	0.10467
NMF	0.63266	0.83124	0.71848	0.16543	0.29952	0.21314
AE	0.57798	0.72412	0.64285	0.54799	0.32286	0.40633
BeatGAN	0.72808	0.61610	0.66742	0.40050	0.43058	0.41500
USAD	0.85543	0.72412	**0.78431**	0.33980	0.47187	0.39509
FGANomaly	0.96350	0.61610	0.75160	0.43536	0.50180	**0.46622**

TABLE 2
Main characteristics of four benchmark datasets.

Dataset	Training set size	Testing set size	No. of Dimension	Anomaly ratio (%)
MSL	58317	73729	55	10.72
SMAP	135183	427617	25	13.13
$SWaT_{subset}$	224959	224960	51	19.05
$WADI_{subset}$	86401	86401	123	3.87

the the data collected under normal situation and divide the remaining 2 days data into training set and testing set.

5.1.2 Baselines

We take 5 unsupervised anomaly detection methods as baselines which are

- *Principal Component Analysis (PCA) and Non-negative Matrix Factorization (NMF)*. These are two classic matrix decomposition based anomaly detection algorithms. This kind of methods first decompose the input matrix into two different matrices, then reconstruct the input matrix with them. The anomaly score is defined as the reconstruction error.
- *Multi-Layer Perceptron Autoencoder (AE)*. A simple deep autoencoder based on multi-layer perceptron. Compared with PCA and NMF, AE could capture nonlinear relationship between data and thus reconstruct normal data better at most time.
- *BeatGAN* [20]. A GAN based multivariate time series anomaly detection method, which adopt autoencoder as main architecture and discriminator as adversarial regularization.
- *UnSupervised Anomaly Detection (USAD)* [30]. USAD is a anomaly detection model based on adversely trained autoencoders. By jointly training two autoencoders adversely and combining the reconstruction errors of them, USAD could detect anomalies in multivariate time series fast and precisely.

Since scikit-learn has excellent implementations of PCA and NMF, we directly conducted anomaly detection based on the implementations of scikit-learn. The number of components for PCA and NMF is 10. Both of the encoder and decoder of AE are single-hidden-layer fully connection network, the number of hidden units is 100 and the dimension of coding is 10. The architecture of BeatGAN is as same as that described by Zhou et al. [20], but the dimension of coding is changed to 10. The architecture of USAD is totally as same as that described by Audibert et al. [30].

5.1.3 Evaluation Metrics

Considering the imbalance problem, we adopt Precision (P), Recall (R), and F1 Score (F1) to evaluate anomaly detection

performance, which are defined as:

$$P = \frac{TP}{TP + FP}, \tag{16}$$

$$R = \frac{TP}{TP + FN}, \tag{17}$$

$$F1 = 2 \cdot \frac{P \cdot R}{P + R}, \tag{18}$$

where TP is for True Positives, FP is for False Positives, and FN is for False Negatives.

We assign a label for each data point. The performance of each model is assessed after a post-processing operation named *point-adjust* proposed by Xu et al. [12], which is inspired by the fact that anomalous observations usually occur in contiguous anomaly segments for the reason of the continuity of time series. Therefore, we consider all points in an anomaly segment as detected if any point in it is detected while points outside the ground truth anomaly segment are treated as usual. In order to compare the performance of our model with the one proposed by Audibert et al. [30], we also assess performance on SWaT and WADI without *point-adjust*.

5.1.4 Task Setting

In order to search the best hyper-parameters for models, each training set is further divided into two parts: training set and validation set at the ratio of 7:3. Since it is costly to build an expert-labeled training set in practice, we consider searching optimal hyper-parameters based on the reconstruction error on validation set rather than F1 score directly, which means we consider hyper-parameters with whom the model has smallest reconstruction error on validation set as optimal hyper-parameters.

Since the original data are in the form of long-range continuous time series, which is not suitable for neural network-based models, thus a sliding window is needed to turn the original data into fragments. In all of our experiments, the sliding window size and stride of neural network based models except USAD are 120 and 1 respectively. The sliding window size of USAD for different datasets are as same as described by Audibert et al. [30].

All of the methods will assign an anomaly score to each observation but not all of them have a specific threshold selection theory, we simply search all of the possible thresholds to decide whether an observation is normal or anomalous, then take the best F1 scores as their performance. Specifically, following the setting from Y. Su et al. [16], when anomaly scores of all the samples in the testing set are obtained, we set each score one by one as the threshold to evaluate the models and then choose the best F1 scores from them.

5.2 Overview Performance

The key hyper-parameters for training FGNomaly are $f(n)$ and γ, we search $f(n)$ in $[log(n-1+e), n, n^2]$ and γ in $[0.001, 0.005, 0.01, 0.05, 0.1, 0.5]$. The performance of all the methods is demonstrated in Table 1, from which we can tell that at most time FGANomaly achieves better performance than the baselines.

TABLE 3
Models and corresponding descriptions of ablation study.

Model Name	Description
LSTM-AE	Plain LSTM based autoencoder, trained only with MSE loss
LSTM-AE+AdaWL	LSTM based autoencoder, trained only with AdaWL
LSTM-AE+FGAN	LSTM based FGAN, the generator is trained with MSE loss.

On the top of table 1 lists the results of all the methods on MSL and SMAP with point-adjust. As demonstrated in the table, the difference between performance of different models on MSL is not so significant, which could be explained by the characteristics of dataset MSL and point-adjust: observations of MSL have little variance which makes it hard to learn normal patterns from abnormal ones, and after point-adjust, the difference is further reduced. Despite such tough circumstance, FGANomaly improves F1 score significantly compared with baselines. In addition, FGANomaly also has the best recall which is very important under the background of anomaly detection. As for dataset SMAP, due to the extreme large data volume, PCA and NMF obtain worst results. It is hard to learn important patterns in such a large matrix by decomposing it directly. AE dose not perform well on this dataset either, which is because AE is faced with overfitting problem during training process. Since both of BeatGAN and USAD have special mechanisms to relieve overfitting, they obtain better results. FGANomaly achieves best performance whether evaluated with F1 Score or recall, the latter is almost reach 1.0.

In the middle of table 1 lists the results of all the methods on SWaT subset both with and without point-adjust. As demonstrated in the table, FGANomaly achieves best performance under both of the two situations. Due to the characteristics of point-adjust, the difference between different models after adjustment is not so significant, the results before point-adjust could provide more precise information. From the results before adjustment, we can see that NMF and PCA obtain the worst results for the same reason of large data volume while BeatGAN and USAD perform better than AE, FGANomaly achieves the best performance both evaluated with F1 and recall. On the bottom of the table lists the results on WADI subset both with and without point-adjust. Since the data dimension of WADI is relatively higher, it is harder for the models to precisely capture the complex correlations between different dimensions, all the methods perform worse on this dataset. USAD achieves the best performance after point-adjust, but FGANomaly obtain the best performance before point-adjust.

5.3 Ablation Study

As discussed in the previous section, FGANomaly has achieved significant improvement over most of the baselines. In this section, we investigate the improvement gained by different components of FGANomaly. We compare FGANomaly with other three models which are LSTM-AE, LSTM-AE+AdaWL and LSTM-AE+FGAN, they are described in table 3.

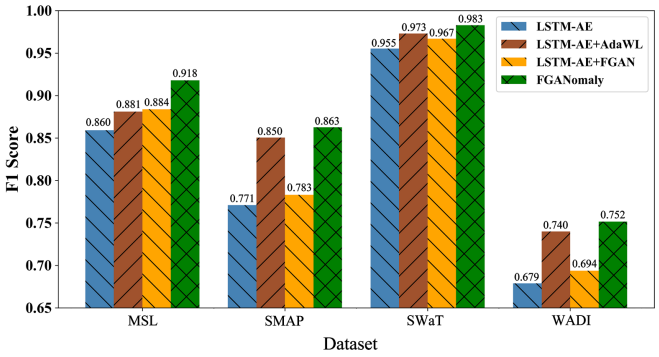


Fig. 4. Ablation study result with point-adjust. Both of AdaWL and FGAN could bring performance improvement but the gains from AdaWL are more significant on SMAP, SWaT and WADI while AdaWL and FGAN bring almost the same improvement on MSL.

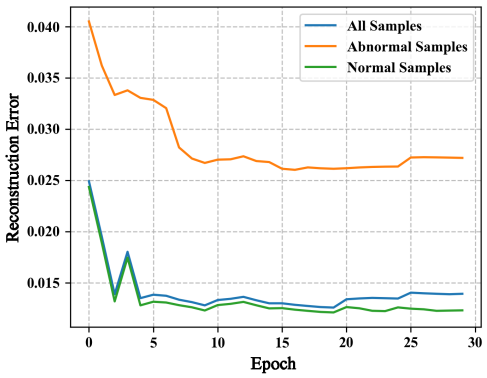


Fig. 5. The training curve and the changes in the reconstruction loss both of normal samples and abnormal samples.

The result of ablation study is illustrated in figure 4. From figure 4, we can tell that both of AdaWL and FGAN can bring improvement when evaluated with F1 score. Particularly, LSTM-AE+AdaWL achieves significant improvement compared with plain LSTM-AE on all datasets, which indicates AdaWL performs better than MSE loss when it comes to anomaly detection problems. FGAN also brings improvement though not as much as AdaWL does. AdaWL fails to perform better than FGAN on MSL but obtains better results on the remaining datasets. This is because as a pairwise loss, AdaWL provides local and direct guidance to learn normal patterns and ignore abnormal ones, it plays a leading role in the training process while as a regularization, FGAN provides global guidance to capture the distribution of normal samples, it plays a secondary role in the training process.

5.4 Robustness Verification

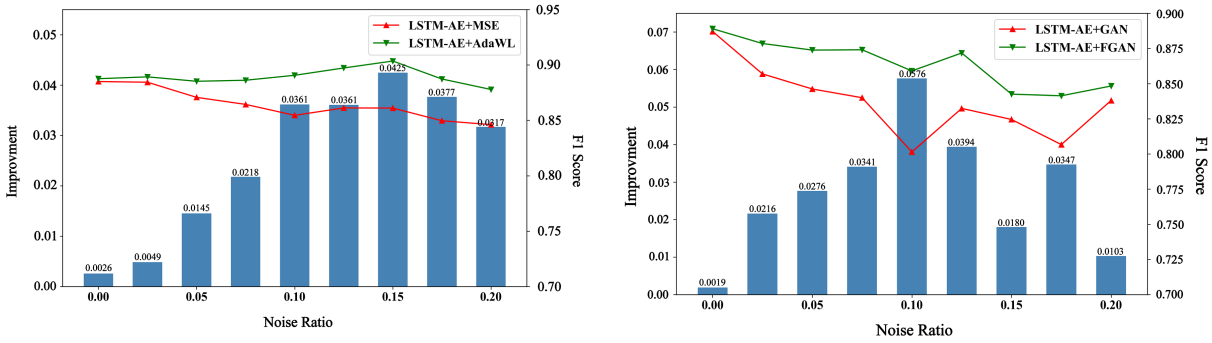
In order to probe into the stability of the training process and the effectiveness of our method, we visualize the training curve and the changes in the reconstruction loss both of the normal samples and the abnormal ones. As shown in Figure 5, despite that we introduce items related to training epochs in both the generation of pseudo-label and the adaptive weighted loss, the training process is quite stable. With the training process going on, the reconstruction loss of the normal samples keeps slight decrease while the reconstruction loss of the abnormal samples remains the same, even slightly increases.

We randomly inject noise into the training set at a certain percentage to verify the robustness of FGANomaly. We first investigate how the noise influences AdaWL and FGAN on MSL, the noise we inject is limited to Gaussian distribution (μ = 0, δ = 1) and the noise ratio ranges from 0.0% to 20%. The results are shown in Figure 6. Figure 6(a) shows the performance of LSTM-AE trained with MSE loss and AdaWL respectively. Line charts demonstrate the change of F1 scores while histogram shows the disparity. As indicated in figure 6(a), the performance of the model trained with MSE loss gets worse rapidly as the noise ratio increases while the performance of model trained with AdaWL remains relatively stable when the noise ratio is under 15% but starts to decline as it exceeds 15%. The gap between MSE loss and AdaWL almost increases continually when the noise ratio is under 15% while it starts to decline as noise ratio exceeds 15%. Figure 6(b) shows the performance of LSTM-AE trained under FGAN architecture and conventional GAN architecture. From figure 6(b) we can tell that as the noise ratio increases, both of GAN and FGAN perform worse. But FGAN has better robustness because it declines slower as indicated in the histogram. As demonstrated in the line charts, sometimes the performance of models get better as noise ratio increases, that is because the noise we inject into the training set can be regarded as adversarial examples which sometimes increase the robustness of models. As the experiment results indicate, the components we propose for anomaly detection have better characteristics compared with their counterparts.

Then, we compare the robustness of FGANomaly with two state-of-art methods: BeatGAN and USAD on four of the datasets, and the noise we inject is limited to Gaussian distribution (μ = 0, δ = 1) and the noise ratio ranges from 0.0% to 30%. The results are illustrated in figure 7. As shown in figure 7, BeatGAN has the worst robustness, the F1 score of BeatGAN declines rapidly as noise ratio increases on all of the datasets, especially on WADI, the F1 score changes from about 0.66 to 0.3. That is because BeatGAN is designed to be trained only on clean data to capture the distribution of normal data. But as noise ratio increases, conventional GAN architecture captures not only the distribution of normal data but also the one of abnormal data, thus fails to detect anomalous samples from normal ones. USAD has better robustness compared with BeatGAN due to the design of adversarial training, but the effect is not so significant. Due to the special design of AdaWL and FGAN, FGANomaly shows the best robustness. When the noise ratio is under 20%, the F1 scores of FGANomaly remain relatively constant and high which indicates FGANomaly has the ability to precisely learn the normal patterns even if the training set has been contaminated partially.

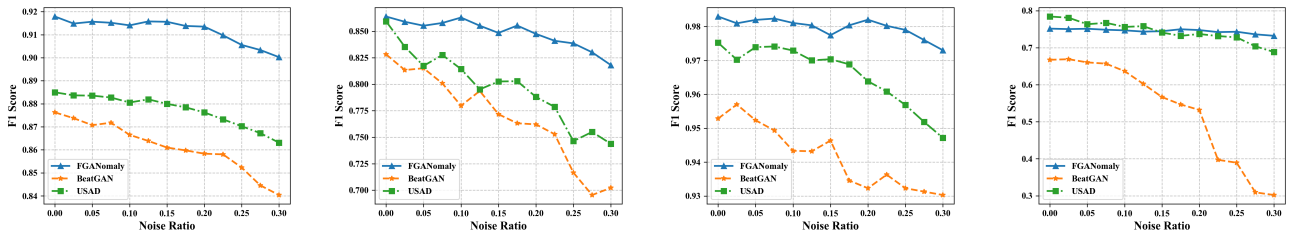
5.5 Case Study

We apply FGANomaly to The 1^{st} International Project Competition for Structural Health Monitoring (IPC-SHM, 2020).



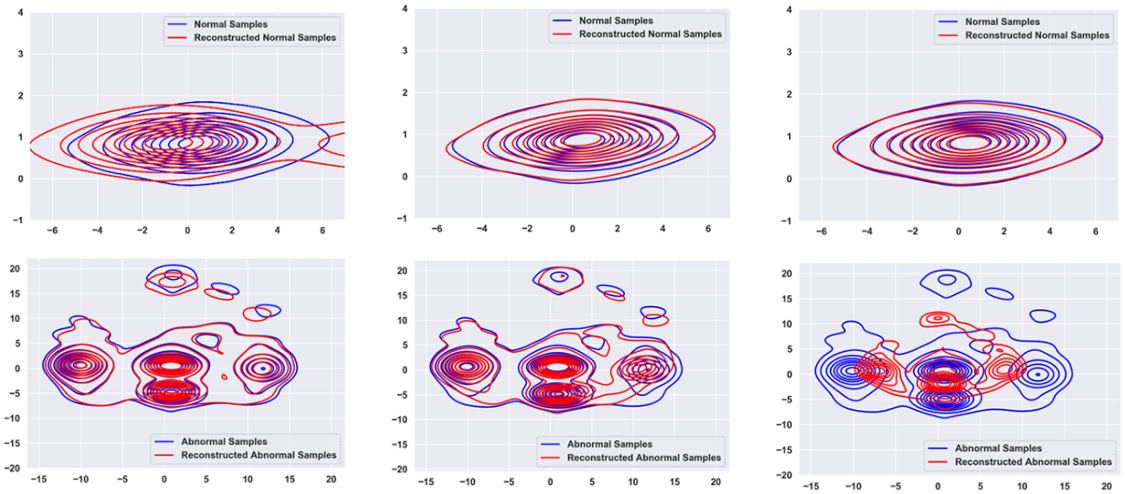
(a) Comparison between AdaWL and MSE loss (b) Comparison between GAN and FGAN

Fig. 6. Robustness verification results of AdaWL and FGANomly. (a): We compare the robustness of AdaWL and MSE loss. (b): We compare the robustness of FGANomaly and conventional GAN architecture. Line charts depict the change of F1 score while histograms depict the improvement.



(a) (b) (c) (d)

Fig. 7. Robustness verification results.(a) result on MSL. (b) result on SMAP. (c) result on SWaT subset. (d) result on WADI subset.



(a) BeatGAN (b) USAD (c) FGANomaly

Fig. 8. Visualization of the distributions of original data and reconstructed data. Pictures on the top are the kde plots of normal distributions after dimension reduction while on the bottom are the results of abnormal distributions. Red lines stand for reconstructed distribution and blue lines stand for original distribution.

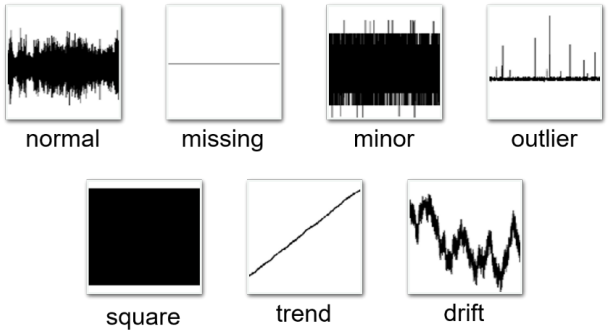


Fig. 9. Normal data and six classes of anomalies in project 2, IPC-SHM, 2020 [32]

It is a competition aiming at combining artificial intelligence (AI) technology with Structural Health Monitoring (SHM) to better ensure the infrastructures safe and reliable[2]. The competition consists of three projects, each incorporating data drawn from a full-scale bridge. We are interested in the second projects: data anomaly detection for SHM.

In this project, a dataset that consists of one-month of acceleration data for a long-span cable-stayed bridge in China is provided. There are 38 sensors thus the dimension of data is 38. The sampling frequency of all the sensors is 20Hz. As shown in Figure 9, there are six classes of anomalies in the dataset. For the sake of simplicity, we merge the six type of anomalies into one class: anomaly. Since the sampling frequency is high, the data of each 10 seconds are aggregated into single value by mean operation. The dataset is divided into two subsets: a training set corresponding to the first 15 days data and a testing set corresponding to the remaining data. In this competition, we take BeatGAN and USAD as baselines, the result is listed in Table 4, form which we can tell that FGANomaly obtains the best results on this project on all of the metrics.

TABLE 4
Anomaly detection result on project 2, IPC-SHM, 2020.

Models	Precision	Recall	F1 Score
BeatGAN	0.74962	0.80322	0.77549
USAD	0.90502	0.87132	0.88785
FGANomaly	0.93508	0.92134	**0.92816**

To compare the performance of the models more directly, anomaly scores of the normal samples and abnormal samples are visualized in Figure 10. As shown in the figure, there are three pairs of box plots which correspond to BeatGAN, USAD and FGANomaly respectively. For each model, the left box plot depicts the anomaly score distribution of normal samples while the right one is the anomaly score distribution of abnormal samples. As the figure shows, FGANomaly reconstructs the distribution of normal samples best, because the anomaly scores for normal samples obtained by FGANomaly are smallest while BeatGAN has largest anomaly scores for normal samples. The distance between anomaly scores of normal samples and abnormal samples obtained by FGANomaly is the largest, which means FGANomaly learns clearer boundary between the normal distribution and abnormal distribution. On the contrary, the boundary learned by BeatGAN is not so clear thus it obtains worse result to detect anomalies.

2. http://sstl.cee.illinois.edu/ipc-shm2020/

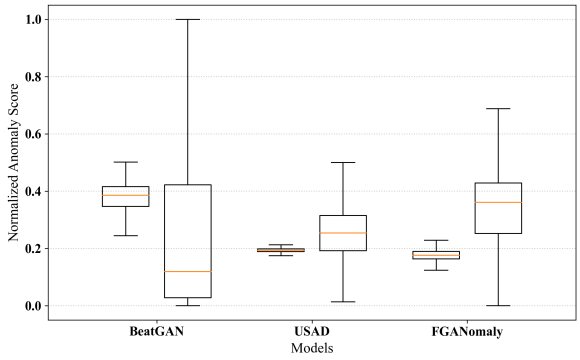


Fig. 10. Anomaly score visualization. For each model, there are two box plots: on the left depicts the anomaly score distribution of normal samples while on the right is the anomaly score distribution of abnormal samples.

The distributions of original data and reconstructed data are also visualized, which are shown in Figure 8. There are tow pictures for each model, the one on the top is a two-dimensional kde plot of the normal distribution after dimension reduction while the one on the bottom belongs to the abnormal distribution. As the figure shows, FGANomaly reconstructs the normal samples at a very high precision while reconstructed abnormal samples badly, which explains the good performance of FGANomly well.

6 Conclusion

In this paper, we proposed a novel GAN-based anomaly detection method for multivariate time series. To address the problem of overfitting in conventional AE-based and GAN-based anomaly detection methods, we proposed Filter GAN which screens the possible abnormal samples before training the discriminator thus makes the model to capture the distribution of normal data precisely, and we also designed a special objective named Adaptive Weight Loss for the generator which dynamically assigns weights to different points during the training process according to their reconstruction errors. To evaluate the proposed method, we conducted comprehensive experiments on four public datasets, the experimental results showed that our model has superiority both in performance and robustness. For the purpose of having some insights into the model, we did a case study to visualize the results, which showed that our model learns clear boundary between the normal distribution and abnormal distribution thus is able to detect anomalies well. Moreover, there are also some points to be improved, such as the fusion of information between different dimensions in multivariate time series, which is an important research direction for the future work.

ACKNOWLEDGMENTS

This work is supported by the National Key R&D Program of China (2018YFB2101003), the National Natural Science Foundation of China (51991395, 71901011, U1811463, 51822802, 72171013, 82161148011).

REFERENCES

[1] J. Mao, H. Wang, and B. F. Spencer Jr, "Toward data anomaly detection for automated structural health monitoring: Exploiting generative adversarial nets and autoencoders," *Structural Health Monitoring*, p. 1475921720924601, 2020.

[2] M. M. Hassan, A. Gumaei, A. Alsanad, M. Alrubaian, and G. Fortino, "A hybrid deep learning model for efficient intrusion detection in big data environment," *Information Sciences*, vol. 513, pp. 386–396, 2020.

[3] D. Cheng, S. Xiang, C. Shang, Y. Zhang, F. Yang, and L. Zhang, "Spatio-temporal attention-based neural network for credit card fraud detection," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 362–369.

[4] C. Horn and R. Willett, "Online anomaly detection with expert system feedback in social networks," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 1936–1939.

[5] J. Roy, "Rule-based expert system for maritime anomaly detection," in *Sensors, and Command, Control, Communications, and Intelligence (C3I) Technologies for Homeland Security and Homeland Defense IX*, vol. 7666. International Society for Optics and Photonics, 2010, p. 76662N.

[6] A. Candelieri, "Clustering and support vector regression for water demand forecasting and anomaly detection," *Water*, vol. 9, no. 3, p. 224, 2017.

[7] R. Sharma and S. Chaurasia, "An enhanced approach to fuzzy c-means clustering for anomaly detection," in *Proceedings of first international conference on smart system, innovations and computing*. Springer, 2018, pp. 623–636.

[8] W. Khreich, B. Khosravifar, A. Hamou-Lhadj, and C. Talhi, "An anomaly detection system based on variable n-gram features and one-class svm," *Information and Software Technology*, vol. 91, pp. 186–197, 2017.

[9] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 eighth ieee international conference on data mining*. IEEE, 2008, pp. 413–422.

[10] J. Ludeña-Choez, J. J. Choquehuanca-Zevallos, and E. Mayhua-López, "Sensor nodes fault detection for agricultural wireless sensor networks based on nmf," *Computers and Electronics in Agriculture*, vol. 161, pp. 214–224, 2019.

[11] H. Alshammari, O. Ghorbel, M. Aseeri, and M. Abid, "Non-negative matrix factorization (nmf) for outlier detection in wireless sensor networks," in *2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC)*. IEEE, 2018, pp. 506–511.

[12] H. Xu, W. Chen, N. Zhao, Z. Li, J. Bu, Z. Li, Y. Liu, Y. Zhao, D. Pei, Y. 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*, 2018, pp. 187–196.

[13] K. Hundman, V. Constantinou, C. Laporte, I. Colwell, and T. 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.

[14] D. H. Hoang and H. D. Nguyen, "A pca-based method for iot network traffic anomaly detection," in *2018 20th International conference on advanced communication technology (ICACT)*. IEEE, 2018, pp. 381–386.

[15] B. Zong, Q. Song, M. R. Min, W. Cheng, C. Lumezanu, D. Cho, and H. Chen, "Deep autoencoding gaussian mixture model for unsupervised anomaly detection," in *International Conference on Learning Representations*, 2018.

[16] Y. Su, Y. Zhao, C. Niu, R. Liu, W. Sun, and D. 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.

[17] S. Akcay, A. Atapour-Abarghouei, and T. P. Breckon, "Ganomaly: Semi-supervised anomaly detection via adversarial training," in *Asian conference on computer vision*. Springer, 2018, pp. 622–637.

[18] D. Li, D. Chen, B. Jin, L. Shi, J. Goh, and S.-K. 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.

[19] Z. Niu, K. Yu, and X. Wu, "Lstm-based vae-gan for time-series anomaly detection," *Sensors*, vol. 20, no. 13, p. 3738, 2020.

[20] B. Zhou, S. Liu, B. Hooi, X. Cheng, and J. Ye, "Beatgan: Anomalous rhythm detection using adversarially generated time series." in *IJCAI*, 2019, pp. 4433–4439.

[21] K.-L. Li, H.-K. Huang, S.-F. Tian, and W. Xu, "Improving one-class svm for anomaly detection," in *Proceedings of the 2003 International Conference on Machine Learning and Cybernetics (IEEE Cat. No. 03EX693)*, vol. 5. IEEE, 2003, pp. 3077–3081.

[22] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *International conference on machine learning*. PMLR, 2018, pp. 4393–4402.

[23] X. Miao, Y. Liu, H. Zhao, and C. Li, "Distributed online one-class support vector machine for anomaly detection over networks," *IEEE transactions on cybernetics*, vol. 49, no. 4, pp. 1475–1488, 2018.

[24] H. S. Emadi and S. M. Mazinani, "A novel anomaly detection algorithm using dbscan and svm in wireless sensor networks," *Wireless Personal Communications*, vol. 98, no. 2, pp. 2025–2035, 2018.

[25] B. Harish and S. A. Kumar, "Anomaly based intrusion detection using modified fuzzy clustering." *IJIMAI*, vol. 4, no. 6, pp. 54–59, 2017.

[26] R. Ghezelbash, A. Maghsoudi, and E. J. M. Carranza, "Optimization of geochemical anomaly detection using a novel genetic k-means clustering (gkmc) algorithm," *Computers & Geosciences*, vol. 134, p. 104335, 2020.

[27] J. Li, H. Izakian, W. Pedrycz, and I. Jamal, "Clustering-based anomaly detection in multivariate time series data," *Applied Soft Computing*, vol. 100, p. 106919, 2021.

[28] N. Merrill and C. C. Olson, "Unsupervised ensemble-kernel principal component analysis for hyperspectral anomaly detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 112–113.

[29] J. Wang, J. Jiang, W. Jiang, C. Li, and W. X. Zhao, "Libcity: An open library for traffic prediction," in *Proceedings of the 29th International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 145–148. [Online]. Available: https://doi.org/10.1145/3474717.3483923

[30] J. Audibert, P. Michiardi, F. Guyard, S. Marti, and M. 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.

[31] J. Goh, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *International conference on critical information infrastructures security*. Springer, 2016, pp. 88–99.

[32] Y. Bao, Z. Tang, H. Li, and Y. Zhang, "Computer vision and deep learning–based data anomaly detection method for structural health monitoring," *Structural Health Monitoring*, vol. 18, no. 2, pp. 401–421, 2019.

Bowen Du received the Ph.D. degree in computer science and engineering from Beihang University, Beijing, China, in 2013. He is currently an Assistant Professor with the State Key Laboratory of Software Development Environment, Beihang University. His research interests include smart city technology, multi-source data fusion, and traffic data mining.

Xuanxuan Sun received the B.S. in College of Computer Science and Engineering, South China University of Technology, China, in 2019. His research interests include multivariate time series, deep learning and data mining.

Junchen Ye is an Ph.D. candidate of software engineering in Beihang University, China. He received his B.S. degree from Beihang University of computer science in 2018. His research interests include intelligent transportation, deep learning and graph neural network.

Ke Cheng received the Master degree in computer science and technology from Dalian University of Technology, Dalian, China, in 2021. He is currently pursuing the Ph.D. degree in computer science and technology with Beihang University, Beijing, China. His current research interests include dynamic graph learning.

Jingyuan Wang received the Ph.D. degree from the Department of Computer Science and Technology, Tsinghua University, China. He is currently an Associate Professor of School of Computer Science and Engineering, Beihang University, China. His is also the head of the BIGSCity lab, and Vice Director of the Beijing City Lab (BCL). He published more than 50 papers on top journals and conferences, such as SIGKDD, AAAI, ICDM, IEEE TKDE, IEEE TMC, ACM TOIS, etc. His general area of research is data mining and machine learning, with special interests in smart cities.

Leilei Sun is with the State Key Laboratory of Software Development Environment(SKLSDE), School of Computer Science and Engineering, Beihang University, Beijing, China. His research interests include machine learning and data mining. He has published a series papers on IEEE Transactions on Data and Knowledge Engineering (TKDE), ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), etc.