IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 24, 2021, accepted May 5, 2021, date of publication May 10, 2021, date of current version May 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3078553

Studies on the GAN-Based Anomaly Detection Methods for the Time Series Data

CHANG-KI LEE[1], YU-JEONG CHEON[2], AND WOOK-YEON HWANG[3]
[1]Research Center for Industry-Academy Cooperation, Dong-A University, Busan 49315, South Korea
[2]Department of Management Information Systems, Dong-A University, Busan 49315, South Korea
[3]Department of Global Business, Dong-A University, Busan 49315, South Korea

Corresponding author: Wook-Yeon Hwang (wyhwang@dau.ac.kr)

This work was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education under Grant 2019R1I1A3A01040343.

ABSTRACT Anomaly detection (AD) for times series data using the generative adversarial network (GAN) has been proposed in recent years. According to the previous study, the GAN-based AD outperformed the cumulative sum (CUSUM) chart. However, no framework for comparison is provided in their works. So, we conduct new studies crucial for the GAN-based AD methods (the MAD-GAN and the TAnoGAN). First, we propose a new framework for fair and systematic comparisons for the prediction performance of the GAN-based AD methods as well as the cumulative sum (CUSUM) chart. So, we evaluate the three methods with four simulation data and secure water treatment system data. Under the proposed comparison framework, the CUSUM chart generally shows prediction performances better than the GAN-based AD methods. Our results imply that more follow-up studies are required before deploying the GAN-based AD methods. Second, we find that adjusting the number of backpropagation steps of the inverse mapping technique can improve the prediction performance of the GAN-based AD methods. Furthermore, we find that monitoring the residuals of the fitted model significantly improves the prediction performance of the GAN-based AD methods as well as the CUSUM chart.

INDEX TERMS Anomaly detection, time series data, comparison framework, generative adversarial network, cumulative sum chart, backpropagation.

I. INTRODUCTION

An anomaly can be defined as an unusual pattern that does not conform to the expected behavior. Anomaly detection (AD) refers to the automatic identification of unforeseen or abnormal phenomena embedded in a large amount of normal data [1]–[3]. The goal of AD is to determine which instances stand out as different from others. Since anomalies have values that deviate far from the average of other values, AD is also known as deviation detection. A lot of AD algorithms have been proposed for various data domains including high-dimensional, uncertain streaming, network, and time series data [4], [5]. In particular, with the advent of the Internet of Things (IoT), a significant amount of time series data is generated, and the demand for time series AD methods is increasing. In addition to IoT, AD algorithms have been employed in medical image, sensor networks, video surveillance, and industrial damage detection [6]–[9]. Traditional AD methods employ statistical approaches such as the cumulative sum (CUSUM) charts to detect changes in the underlying distribution [10], [11].

Recently, researchers have also proposed a lot of machine learning-based techniques for AD such as an autoencoder framework based on long short-term memory networks [12] and long short-term memory-based variational autoencoder (LSTM-VAE) for multimodal multivariate signal data [13]. Besides, a predictive approach to detect anomalies through the deep LSTM networks has been suggested [14]. It used the predicted error distribution of the deep LSTM model learned to determine whether electrocardiogram (ECG) signal data were normal or abnormal.

Moreover, the growing popularity of the generative adversarial networks (GAN) has been contributed to the advent of many GAN-based AD methods. An unsupervised GAN-based AD method called AnoGAN has been proposed to detect anomalies in medical image data [6]. However,

The associate editor coordinating the review of this manuscript and approving it for publication was Mehdi Hosseinzadeh.

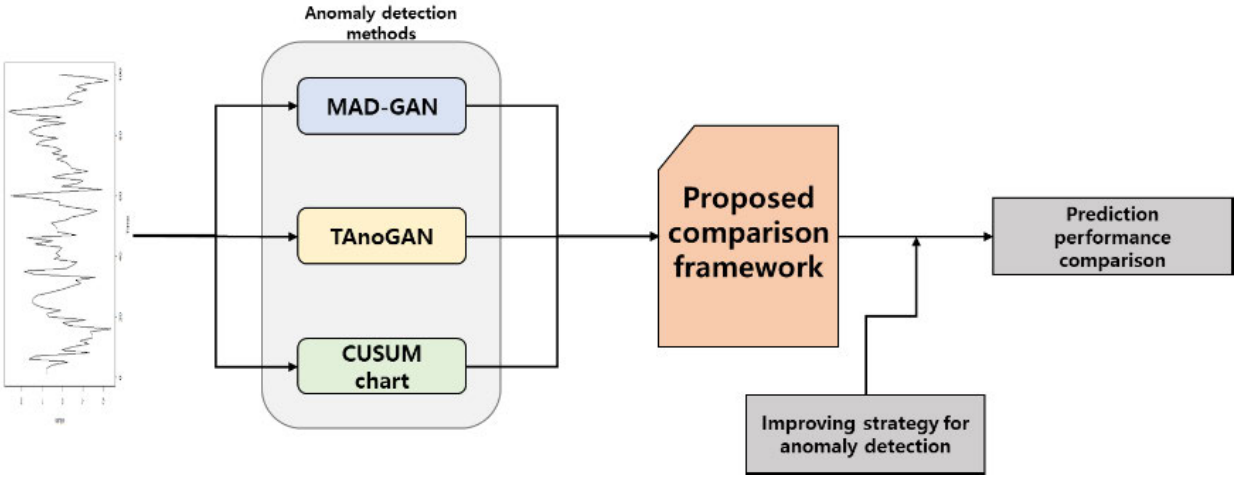

FIGURE 1. An overview of studies on the GAN-based anomaly detection methods for the time series data.

the AnoGAN is more suitable for image data than time series data. So, GAN-based AD methods for time series data have been proposed. Li *et al.* [15] proposed the multivariate AD method with GAN called MAD-GAN. Bashar and Nayak [16] also proposed a GAN-based AD method called TAnoGAN that can be applied when only a small number of instances are available. The GAN-based AD methods first calculate an instance's anomaly score and compare it with a threshold to detect anomalies. According to the previous work [17], the autoregressive integrated moving average (ARIMA) model performed better than the MAD-GAN. On the other hand, the GAN-based AD method yielded better prediction performance compared to the CUSUM chart [18]. Inspired by the two studies [17], [18], we perform new studies for the GAN-based AD methods (the MAD-GAN and the TAnoGAN). An overview of this paper is visualized in Fig. 1.

The contributions of this paper are as follows. First, our studies present a novel comparison framework and compare the prediction performance of the GAN-based AD methods with a CUSUM chart [18], not with the ARIMA model [19]. The CUSUM chart is one of the most popular methods of detecting anomalies in the statistical process control (SPC). It is effective for detecting small process shifts. According to the previous studies [18], [20], we employ the CUSUM chart as the baseline method. In order to differentiate our study from the previous works [15], [17], [21], we present a novel framework focusing on fair and systematic comparisons. The key idea of the framework is described as follows. The prediction performances of the three methods (the MAD-GAN, the TAnoGAN, and the CUSUM chart) are equalized using the training data, and then each of the prediction performances is compared using test data. Fig. 2 summarizes our framework.

Unlike the previous studies [15], [17] that use only real data, we evaluate three methods using simulation data as well as real data. To compare the GAN-based AD methods with the CUSUM chart, the secure water treatment (SWaT) system data and simulated time series data are considered. The simulated time series data include autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), and the beta-distributed multistage process. The AR considers a linear combination of past variables. Unlike the AR, the MA focuses on a linear combination of past noises. The ARMA is a combination of the AR and the MA. In this paper, we consider both univariate and multivariate time series data.

Second, we find out that the number of backpropagation steps affects the prediction performance of the MAD-GAN and the TAnoGAN. The number of backpropagation steps means the number of iterations of the inverse mapping, which is the process of obtaining an anomaly score [6], [15], [16], [18]. Although an appropriate number of backpropagation steps must be predefined to calculate anomaly scores, detailed discussions have been overlooked in previous studies [6], [15], [16], [18]. So, we examine the relationship between the number of backpropagation steps and the prediction performance of the GAN-based methods. As a result, in the autoregressive (AR) and the moving average (MA) data sets, we achieve the performance improvement of the GAN-based AD methods through early stopping. Moreover, the prediction performance of the three methods is improved by monitoring the residuals of the fitted model from the autoregressive moving average (ARMA) data.

The remainder of the paper is structured as follows. The existing AD methods are described in Section II. Section III details the GAN-based AD methods to be compared in this study. Section IV contains the comparison framework, simulation research, and improvement strategies for time series AD methods. Section V covers comparison results using the SWaT system data. Finally, concluding remarks are comprised in Section VI.

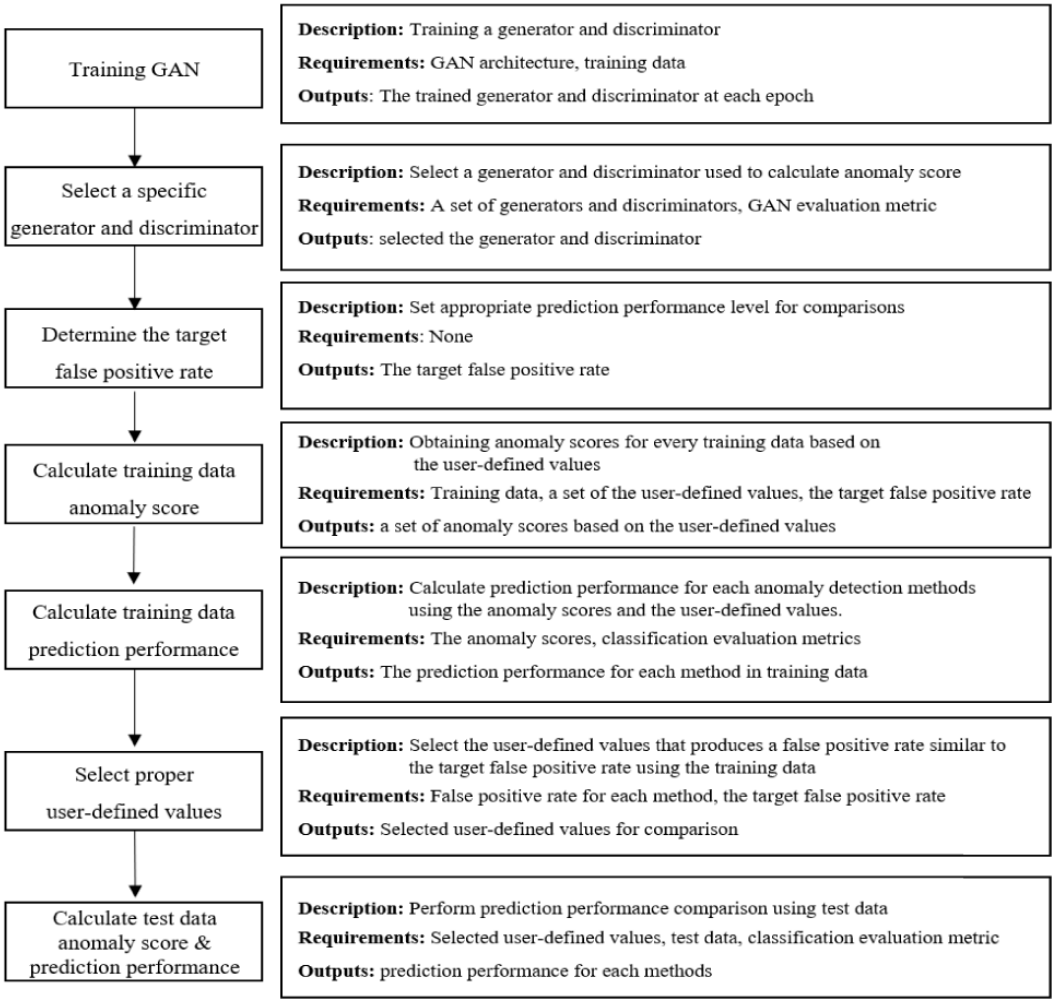

FIGURE 2. A framework for comparisons.

II. RELATED WORK

Control charts, which are based on statistical theory, are the most widely used AD methods in the SPC. Monitoring statistics and control limits are the two major components in the construction of a control chart. A monitoring statistic plotted on a control chart can be computed and analyzed as a function of the variables. Control limits are usually decided based on the probability distribution of the monitoring statistic with user-defined Type I error (false positive rate). If a monitoring statistic calculated on an instance exceeds the control limits, we assume that an anomaly is detected. Several AD methods based on statistics have been proposed, including the CUSUM chart [22]–[24]. Statistical approaches need to estimate the underlying probability distributions of the monitoring statistics. However, it is difficult to know what distribution monitoring statistics follow.

There are also many studies based on machine learning techniques. Machine learning-based methods can be classified into three categories: supervised, semi-supervised, and unsupervised learning [24]. The supervised learning method is to build a discriminative model to distinguish between normal and abnormal instances. When a new instance occurs, the trained discriminative model determines whether it is normal or abnormal. As the discriminative models, well-known supervised learning approaches such as decision trees, neural networks, and support vector machines are employed. Supervised learning models require a sufficient number of normal and abnormal instances. However, these approaches are not as general as the semi-supervised or the unsupervised methods, owing to the insufficiency of abnormal instances in training data [25], [26].

In practice, normal instances can be obtained more easily than abnormal instances. Thus, the semi-supervised methods are more widely employed than supervised methods. These methods leverage existing normal instances to separate abnormal instances. One popular way to use autoencoders (AE) is to train them in a semi-supervised process on data without anomalies. A sufficiently trained AE would yield low reconstruction errors for normal instances compared to abnormal instances [27]–[29].

The unsupervised methods are also applied to tackle the AD. The cases of local outlier factor (LOF) [30],

connectivity-based outlier factor (COF) [31] which is a developed version of the LOF, and cluster-based local outlier factor (CBLOF) [32] are good examples. The difference between the LOF and the COF is that the LOF utilizes distance while the COF uses density to detect anomalies. To determine density areas in normal data, the CBLOF uses clustering and then performs a density estimation for each cluster. The density areas are utilized for detecting anomalies.

With the recent success of deep learning, deep learning-based AD methods have been proposed [12]–[14]. Up-to-date approaches to deep learning-based models are based on the GAN framework. The AnoGAN [6] proposed to detect anomalies in medical image data. It maps test images from image space to latent space and reconstructs the image to calculate an anomaly score. The loss between the test image and the reconstructed image is used to compute the anomaly score. The AnoGAN used a convolutional neural network (CNN) specialized in image processing for the architecture of a generator and a discriminator. However, the CNN does not include mechanisms to handle time series characteristics, so it is not generally used in time series data [16]. Therefore, two variants of the AnoGAN have been proposed for time series data [15], [16]. Both GAN-based AD methods (the MAD-GAN and the TAnoGAN) use the LSTM in generators and discriminators instead of the CNN, but there are differences in calculating anomaly scores. The differences are explained in Section III.

III. ANOMALY DETECTION WITH GAN FOR TIME SERIES DATA

A description of the previous studies that we are considering is discussed in this Section. The GAN-based AD methods consist of two steps. The first step is to train the GAN using only normal time series data to learn the distribution of normal states. And then, the anomaly score is calculated using the trained generator and discriminator. This Section deals with three topics: (1) the GAN using the LSTM for time series data (LSTM-GAN), (2) the inverse mapping procedure for AD [6], and (3) the calculation of two anomaly scores [15], [16].

A. TRAINING LSTM-GAN

The basic concept of the GAN is to train two neural network models to improve each other using the minimax objective function. A generator (G) tries to produce fake samples that look real, while a discriminator (D) does to distinguish between generated samples and real data [31]. The GAN is defined as a minimax game with the following objective function.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\log D(\mathbf{x})\right] + \mathbb{E}_{\mathbf{x}\sim p_{\mathbf{x}}(\mathbf{z})}\left[\log\left(1 - D(G(\mathbf{z}))\right)\right]. \quad (1)$$

During adversarial training, the generator's ability to generate fake data is improved and the discriminator's ability to distinguish between real and fake data is also enhanced. Goodfellow *et al.* [33] used deep multi-layer perceptron to train the GAN. However, in the time series data, the LSTM is more appropriate than the multi-layer perceptron. It is a family of neural networks for processing sequential data [34]. The LSTM has special units known as memory blocks in the recurrent hidden layer. A memory block contains a memory cell, input gate, forget gate, and output gate [35], [36]. Each component interacts to control inputs that are not related to the final outputs. The LSTM used in conjunction with memory blocks can learn complex time series data. The LSTM is adopted in the MAD-GAN and the TAnoGAN due to its suitability for time series data for GAN training [15], [16].

B. INVERSE MAPPING PROCEDURES

To identify anomalies, the model is trained to recognize normal data variations based on the GAN. In other words, the generator learns the mapping from latent space $\mathbf{z}$ to real data $\mathbf{x}$, i.e., $G(\mathbf{z}) : \mathbf{z} \mapsto \mathbf{x}$. However, the generator does not learn the inverse mapping from $\mathbf{x}$ to $\mathbf{z}$, i.e., $G(\mathbf{x})^{-1} : \mathbf{x} \mapsto \mathbf{z}$. So, Schlegl *et al.* [6] presented the inverse mapping $(G(\mathbf{x})^{-1})$ function that maps real data to latent space using gradient descent. The concept of the inverse mapping function is to find the optimal vector $\mathbf{z}$ that can generate a fake sample that is most similar to the real data $\mathbf{x}$. To find the optimal vector $\mathbf{z}$, first, randomly selected vector $\mathbf{z}_1$ from latent space is fed into a trained generator to get a generated sample $G(\mathbf{z}_1)$. Then, $\mathbf{z}_1$ will be updated using the gradient of a loss function to get $\mathbf{z}_2$. The $G(\mathbf{z}_2)$ is more similar to $\mathbf{x}$ compared to $G(\mathbf{z}_1)$. The elements of vector $\mathbf{z}$ are iteratively updated through backpropagation steps to find the most similar fake sample $G(\mathbf{z}_\Gamma)$. The Γ is a predetermined number as the maximum number of iterations. Hereafter the Γ is called as the number of backpropagation steps.

C. ANOMALY SCORES

1) MAD-GAN ANOMALY SCORES

Li *et al.* [15] proposed an anomaly score to detect time series data known as discrimination and reconstruction anomaly score (DR-score). The DR-score is composed of two components. One is the discrimination-based score (discriminator loss) and the other is the reconstruction-based score (residual loss). The discrimination-based score uses the trained discriminator $D(\mathbf{x})$ within the GAN to distinguish whether the test data are normal or abnormal. The reconstruction-based score is a sum of residuals based on the inverse mapping from test data to latent space. The loss function of the MAD-GAN to get a gradient is defined as:

$$\mathcal{L}_{MAD}(\mathbf{z}_i) = 1 - similarity(\mathbf{x}, G(\mathbf{z}_i)), \quad (2)$$

where $similarity(\cdot,\cdot)$ is defined as covariance for simplicity [15], [18], $\mathbf{z}_i$ is the ith inverse mapped vector, $G(\mathbf{z}_i)$ is the generated time series sample from $\mathbf{z}_i$, and $i \in \{1, 2, \ldots, \Gamma\}$.

After $\mathbf{z}_i$ is sufficiently updated, the residual score is defined as:

$$\mathcal{R}(\mathbf{x}) = \sum_{j=1}^{w} \left|\mathbf{x}^j - G\left(\mathbf{z}_\Gamma^j\right)\right|, \quad (3)$$

where $G(\mathbf{z}_\Gamma)$ is the generated time series sample from the last updated vector $\mathbf{z}_\Gamma$ and w is window size.

The trained discriminator D can classify whether the test data come from the underlying distribution of the training data, so it can be defined as the discrimination score [15], [18]. An anomaly score of the MAD-GAN is defined as below:

$$\mathcal{A}_{MAD}(\mathbf{x}) = (1-\lambda)\cdot\mathcal{R}(\mathbf{x}) + \lambda\cdot D(\mathbf{x}), \tag{4}$$

where $\mathcal{R}(\mathbf{x})$ is the residual score, $D(\mathbf{x})$ is the discrimination score and the weighting factor λ is a positive value between 0 and 1. When the anomaly score is less than or equal to the threshold (τ), we assume that an anomaly is not detected. On the other hand, an anomaly is detected when the anomaly score is greater than the threshold (τ).

D. TAnoGAN ANOMALY SCORES

The TAnoGAN is the variant of the AnoGAN [6] for time series data. It is virtually the same procedure as the AnoGAN except to train GAN with the LSTM, not with the CNN. Its loss function is composed of two parts, a residual loss $\mathcal{L}_R$ and a discriminator loss $\mathcal{L}_D$. The residual loss $\mathcal{L}_R$ is calculated by the point-wise distance (e.g., values in timestamps) between real data $\mathbf{x}$ and generated sample $G(\mathbf{z}_i)$. It is defined as:

$$\mathcal{L}_R(\mathbf{z}_i) = \sum_{j=1}^{w}\left|\mathbf{x}^j - G\left(\mathbf{z}_i^j\right)\right|, \tag{5}$$

where $\mathbf{z}_i$ is the ith inverse mapped vector, $G(\mathbf{z}_i)$ is the generated time series sample from $\mathbf{z}_i$, $i \in \{1, 2, \ldots, \Gamma\}$, and w is window size.

The feature mapping technique [37] uses an output of an intermediate layer of the discriminator instead of a sigmoid output of the discriminator to calculate the discriminator loss. Therefore, the discriminator loss is defined as:

$$\mathcal{L}_D(\mathbf{z}_i) = \sum_{j=1}^{w}\left|f\left(\mathbf{x}^j\right) - f\left(G\left(\mathbf{z}_i^j\right)\right)\right|, \tag{6}$$

where $f(\cdot)$ is the output of the intermediate layer of the discriminator and w is window size. Thus, the loss function of the TAnoGAN is defined as a weighted average of both components:

$$\mathcal{L}_{TAno}(\mathbf{z}_i) = (1-\lambda)\cdot\mathcal{L}_R(\mathbf{z}_i) + \lambda\cdot\mathcal{L}_D(\mathbf{z}_i). \tag{7}$$

Only the elements of vector $\mathbf{z}$ are updated through back-propagation. After $\mathbf{z}_i$ is sufficiently updated, an anomaly score of the TAnoGAN is defined as below:

$$\mathcal{A}_{TAno}(\mathbf{x}) = (1-\lambda)\cdot\mathcal{L}_R(\mathbf{z}_\Gamma) + \lambda\cdot\mathcal{L}_D(\mathbf{z}_\Gamma), \tag{8}$$

where $\mathcal{L}_R(\mathbf{z}_\Gamma)$ and $\mathcal{L}_D(\mathbf{z}_\Gamma)$ are residual loss and the discrimination loss. With the formulation for the anomaly score above, an anomaly is detected when an $\mathcal{A}_{TAno}(\mathbf{x})$ is greater than a user-defined threshold (θ).

For both the MAD-GAN and the TAnoGAN, the λ value to calculate the anomaly score, and the thresholds τ and θ are needed to determine to detect the anomaly. In the following, these values are called user-defined values. There is no procedure for determining the user-defined values λ and τ in [15]. Similarly, Bashar and Nayak [16] implicitly described how to determine λ and θ, but did not explicitly demonstrate systematic procedures. According to Bashar and Nayak [16], λ is determined empirically and θ is determined by the TAnoGAN's prediction performance. The limitation of the two studies [15], [16] is that they did not provide a clear selection procedure of the user-defined. Therefore, a systematic selection of the user-defined values is necessary for a fair comparison of prediction performances.

IV. SIMULATION STUDY

In this Section, we present a novel framework by leveraging previous works [38], [39]. The comparison framework originates from the hypothesis testing and the SPC. Next, the experimental results for the univariate simulation data and the multivariate simulation data are provided.

A. COMPARISON FRAMEWORK

As already mentioned in Section I, the CUSUM chart becomes a baseline method. The CUSUM chart is one of the most popular methods of detecting anomalies in the SPC. The most frequent application of the CUSUM chart is to detect small changes in the mean of normally distributed data. Page [40] is the first to propose the CUSUM chart and a lot of variations have been proposed due to its usefulness. According to the previous studies [15], [18], we compare the GAN-based AD methods with the CUSUM chart under univariate time series data (AR, MA, ARMA, and multistage) in this Section.

First, in order to calculate the anomaly score, it is necessary to select a generator and a discriminator. Unfortunately, it is not clear to compare among generators except by visualizing fake samples. However, a quantitative evaluation metric is needed to select well-trained generators and discriminators. The maximum mean discrepancy (MMD) is employed to evaluate whether the generator successfully learned the distribution of the training data sets [15], [18], [41]. So, in this paper, the GAN models during the training are evaluated using the MMD.

Next, for fair comparisons, we control the prediction performance of the three methods (The MAD-GAN, the TAnoGAN, and the CUSUM chart) in the training data. Since there are no abnormal instances in the training data, we use a false positive rate (FPR) as an evaluation metric. Our proposed framework borrowed theoretical basis from the hypothesis testing and the SPC [39]. A control chart is used to detect changes from normal operating conditions. Usually, normal operating conditions are called in-control. The control chart uses control limits for making a decision. When a monitoring statistic falls outside of the control limits, we assume that the process is out-of-control. It can give out-of-control signals when the process is in-control. This is the so-called Type I error (or FPR). On the contrary, the control chart can fail to give out-of-control signals when the process is out-of-control. This is known as Type II error. These two types of error rates are traded off against each other. To handle

this issue, we usually fix the Type I error at a given level (e.g., 0.005) and try to make the Type II error as small as possible. Therefore, in our comparison framework, we fix the FPR at a target level for the training data and the prediction performances are compared using the test data. After determining the user-defined values, the prediction performances of the existing methods are compared through the test data. We measure the prediction performances of the three methods using five classification evaluation metrics: accuracy (Acc.), F1 score (F1), recall (Rec.), precision (Pre.), and FPR.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{9}$$

$$F1\ score = 2 \times \frac{Pre. \times Rec.}{Pre. + Rec.} \tag{10}$$

$$Recall = \frac{TP}{TP + FN} \tag{11}$$

$$Precision = \frac{TP}{TP + FP} \tag{12}$$

$$FPR = \frac{FP}{TN + FP}, \tag{13}$$

where TP (True positive) is the number of anomalies which are correctly detected, FP (False positive) is the number of anomalies erroneously assigned, TN (True negative) indicates the number of normal values correctly predicted, and FN (False negative) is the number of normal values erroneously assigned. Excluding FPR, the higher values of the evaluation metrics, the better. Fig. 2 summarizes our framework.

B. UNIVARIATE SIMULATION EXPERIMENTS

A description of the univariate simulation data sets, settings, and results is presented in this Section.

1) UNIVARIATE SIMULATION DATA SETS

To compare the prediction performance of the GAN-based AD methods and the baseline method in univariate time series data, we generate simulation data from three time series models: AR (p), MA (q), and ARMA (p, q) model. The ARMA (p, q) model is a combination of the AR (p) model and the MA (q) model and is defined as:

$$x_t = c + \phi_1 x_{t-1} + \cdots + \phi_p x_{t-p} + \varepsilon_t + \psi_1 \varepsilon_{t-1} + \cdots + \psi_q \varepsilon_{t-q}, \tag{14}$$

where c is a constant, and $\{\varepsilon_t\}$ represents a white noise process with a zero mean. In this univariate simulation, we only consider cases where $p = q = 1$ and $\phi_1 = \psi_1 = 0.5$. To generate abnormal data sets, the constant term is changed from 0 to 0.5 (small shifted), 1 (medium shifted), and 2 (large shifted). So, we produce three training data sets consisting of the only process and nine test data sets consisting of normal and abnormal processes. Each of the three training data sets has 50,000 instances generated from normal states. In each of the nine test data sets, there are 808 instances generated from the normal time series and 808 instances generated from the abnormal time series.

2) SIMULATION SETUP FOR UNIVARIATE DATA SETS

In this Section, we describe the preliminary work for AD. The three key steps remain before GAN training with the simulation data sets: data preprocessing, building the architecture of GAN and setting predetermined values.

a: DATA PREPROCESSING

Let $[\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^T]$ be a d-dimensional time sequence with length T, where $\mathbf{x}^i \in R^{d \times i}$ is d vector at time i. For example, in the case of the ARMA (1,1) training data sets, T is 50,000 and d becomes 1. For each data set, normalization is applied to a range between -1 and 1. To obtain training and test samples, we employ a sliding window with window size w and step size s to divide the original time sequence T into N sub-sequence $X = \{[x_1^1, \ldots, x_1^w], \ldots, [x_N^1, \ldots, x_N^w]\}$, where $N = (T - w)/s$. Following the previous study [18], we set the window size and step size differently for training and test samples. In this univariate simulation, the window size and the step size are set to 8 and 4 respectively. So, we obtain 12,498 training samples for training the GAN. Next, we set the window size to 8 and the step size to 8 to detect anomalies. So, the number of training and test samples is 6,249 and 100 respectively.

b: GAN ARCHITECTURE

Because the univariate simulation data sets are too simple, we apply a simple model rather than a complex model to avoid overfitting. So, we use 5 hidden units, 1 hidden layer, and 1 latent space dimension for training the GAN in the univariate simulation data. Both a generator and a discriminator have applied mini-batch optimization based on the Adam optimizer with a batch size of 20 and a learning rate of 0.001. If the learning rate is too large, the loss function may exceed the minimum value. Otherwise, it gets stuck at an undesirable local minimum. The learning rate is empirically determined because it depends on the complexity of the data. So, we heuristically adjust the learning rate. As a result, the small learning rate is more suitable for the simulation data. The gradient update steps are alternated between the generator and the discriminator. However, in practice, it is not performed alternately with the same number of iterations. Hence, the discriminator is updated 3 times per the generator update.

c: SET PREDETERMINED VALUES

Fig. 3 plots the MMD values calculated during the GAN training. As shown in Fig. 3, since the MMD value converges to 0 after 80 epoch, in this Section, the generator and the discriminator at 80 epoch are used to calculate the anomaly scores. For fair comparisons, we use the same generator and discriminator in the MAD-GAN and the TAnoGAN. After selecting the generator and the discriminator at 80 epoch, the number of backpropagation steps (Γ) has to be determined to calculate the anomaly score. In this univariate simulation, the number of backpropagation steps is variously changed to

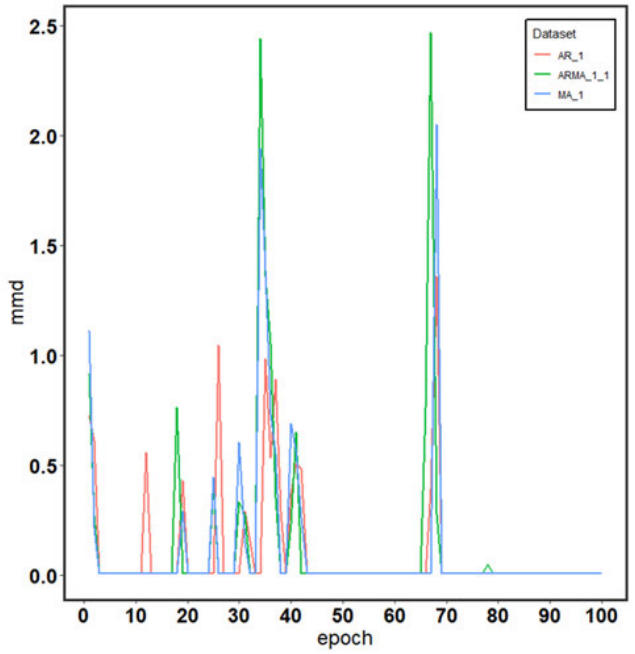

FIGURE 3. The MMD values during GAN training for univariate simulation data sets.

find the best prediction performance. A relationship between Γ and the prediction performance of the GAN-based AD methods will be described later.

3) UNIVARIATE SIMULATION RESULTS

The simulation results of prediction performance for the univariate data sets are summarized in this Section. The following results of each data set are obtained under the same procedure.

a: AUTOREGRESSIVE DATA SETS

To objectively compare the prediction performance of the GAN-based AD methods with the CUSUM chart in the test data, λ and τ must be determined in advance using the training data. So, we set three target FPRs: 10%, 5%, and 1%, and select a combination of λ and τ that produces similar prediction performance to the target FPR. Table 1 shows the FPR results of training data with varying λ and τ. When the target FPR is equal to 10 %, the values of λ and τ are 0.50 and 0.45 respectively. The most similar values to the target FPRs in Table 1 are bolded.

TABLE 1. The MAD-GAN's false positive rates corresponding to λ and τ for the AR (1).

λ \ τ	0.40	0.45	0.50	0.55	0.60	0.65	0.70
0.10	11.58	7.12	4.20	2.39	1.25	0.64	0.32
0.30	13.83	7.58	3.89	1.88	**0.78**	0.33	0.13
0.50	20.02	**9.69**	**4.45**	1.77	0.54	0.16	0.04
0.70	40.95	19.48	7.89	2.21	0.34	0.04	0.00
0.90	84.51	62.48	29.38	5.79	0.06	0.00	0.00

The same weighting factor λ is used for the MAD-GAN and the TAnoGAN. Next, the threshold θ of the TAnoGAN is determined by the FPR of the MAD-GAN. The quantile function value is employed to find an appropriate threshold θ corresponding to the FPR of the MAD-GAN. An appropriate threshold θ is found by the following procedure. First, the anomaly scores for all training samples are calculated using Equation (8). Second, they are sorted in ascending order. Third, θ is obtained from 100(1 - MAD-GAN's FPR)th percentile of a set of the TAnoGANs anomaly scores. For example, when the FPR of the MAD-GAN is 9.69%, the threshold θ is obtained from 90.31th percentile. Table 2 presents selected user-defined values corresponding to each target FPR.

TABLE 2. The user-defined values for the AR (1) data.

User-defined value	Target false positive rate		
	10 %	5 %	1 %
λ	0.50	0.50	0.30
τ	0.45	0.50	0.60
θ	0.20	0.28	0.44

Table 3 shows the prediction performance of the three methods in the AR (1) data sets. The best prediction performances for each simulation data sets are indicated in bold. Interestingly, the baseline model CUSUM chart performs the best in the AR (1) data sets. When comparing only to the GAN-based AD methods, the MAD-GAN performs better than the TAnoGAN. The higher the degree of constant shift, the better the prediction performance of all three methods.

b: MOVING AVERAGE DATA SETS

Following our presented framework, the prediction performance comparisons between the three methods are conducted in the MA (1) data sets. The FPR of the training data is calculated by changing λ and τ, and the results are summarized in Table 4. Boldface is the most similar value to the target FPR. The selected user-defined values are represented in Table 5. Table 6 contains the prediction performance using the MA (1) data sets. The best performances are shown in bold. In the MA (1) data sets, similar to the AR (1) data sets, the CUSUM chart outperforms other methods. When comparing only to the GAN-based AD methods, the TAnoGAN outperforms the MAD-GAN. As with the AR (1) data sets, all three methods improve the prediction performance as the shift size increases.

c: AUTOREGRESSIVE MOVING AVERAGE DATA SETS

In this Section, we carry out a prediction performance comparison using the ARMA (1,1) data sets. The contents of the user-defined value selection are summarized

TABLE 3. The prediction performances for AR (1) data.

Shift size	Target FPR	Method	F1	Acc.	Rec.	Pre.	FPR
Small ($c = 0.5$)	10%	MAD GAN	28.26	54.94	17.75	69.27	7.88
		TAno GAN	15.37	49.75	9.13	48.67	9.63
		CUSUM	**50.92**	**65.41**	**35.89**	**87.61**	**5.07**
	5%	MAD GAN	16.70	53.25	9.38	76.53	2.88
		TAno GAN	5.15	49.31	2.75	40.00	4.13
		CUSUM	**65.46**	**74.13**	**50.00**	**96.65**	**1.73**
	1%	MAD GAN	3.90	50.75	2.00	80.00	**0.50**
		TAno GAN	0.73	49.44	0.38	20.00	1.50
		CUSUM	**24.22**	**56.62**	**13.86**	**95.73**	0.62
Medium ($c = 1$)	10%	MAD GAN	40.77	59.88	27.62	77.82	7.88
		TAno GAN	18.80	50.88	11.38	54.17	9.63
		CUSUM	**95.12**	**95.11**	**95.30**	**94.94**	**5.07**
	5%	MAD GAN	28.69	57.12	17.25	85.19	3.00
		TAno GAN	6.05	49.56	3.25	44.07	4.13
		CUSUM	**97.71**	**98.21**	**98.14**	**98.27**	**1.73**
	1%	MAD GAN	10.82	52.62	5.75	92.00	**0.50**
		TAno GAN	1.22	49.56	0.63	29.41	1.50
		CUSUM	**91.31**	**91.96**	**84.53**	**99.27**	0.62
Large ($c = 2$)	10%	MAD GAN	68.94	74.44	56.75	87.81	7.88
		TAno GAN	40.98	59.31	28.31	74.59	9.63
		CUSUM	**97.34**	**97.28**	**99.63**	**95.15**	**5.07**
	5%	MAD GAN	60.22	70.69	44.37	93.67	3.00
		TAno GAN	21.44	54.19	12.50	75.19	4.13
		CUSUM	**98.33**	**98.82**	**99.38**	**98.29**	**1.73**
	1%	MAD GAN	34.23	60.12	20.75	97.65	**0.50**
		TAno GAN	9.39	51.75	5.00	76.92	1.50
		CUSUM	**99.19**	**99.20**	**99.01**	**99.38**	0.62

in Tables 7 and 8. The prediction performances are summarized and visualized in Table 9. In the ARMA (1,1) data sets, unlike the results of the AR (1) and the MA (1) data sets, the CUSUM chart does not always perform better than other methods. For example, if the target FPR is 1% and the shift size is small, the F1 score of the CUSUM chart is 0. In addition, the F1 score of the CUSUM chart is 0, when target FPR is 1% and the shift size is medium. This means that the CUSUM chart does not detect anomalies correctly. However, the CUSUM chart is superior to the others when the shift size is large. Except when the shift size is large, all three methods show prediction performance with about 50% accuracy. In particular, the GAN-based methods produce 50% or less than 50% accuracy even with a large shift size. This result means that the three methods do not work properly because the normal and abnormal proportions are the same. Thus, we carry out a study to improve the prediction performance of the GAN-based AD methods. Two aspects are considered for improvement: parameter tuning and data preprocessing. There are many parameters related to the GAN architecture, but here we only focus on the parameter Γ related to the detection of anomalies.

TABLE 4. The MAD-GAN's false positive rates corresponding to λ and τ for the MA (1).

λ \ τ	0.40	0.45	0.50	0.55	0.60	0.65	0.70
0.10	11.89	7.52	4.45	2.58	1.36	0.67	0.37
0.30	15.54	**8.72**	4.52	2.17	**0.93**	0.40	0.14
0.50	24.64	11.68	**4.81**	1.70	0.51	0.13	0.03
0.70	58.94	23.05	6.53	1.24	0.16	0.01	0.00
0.90	95.63	83.51	26.87	1.24	0.00	0.00	0.00

TABLE 5. The user-defined values for the MA (1) data.

User-defined value	Target false positive rate		
	10 %	5 %	1 %
λ	0.30	0.50	0.30
τ	0.45	0.50	0.60
θ	0.14	0.16	0.30

4) IMPROVING STRATEGY FOR TIME SERIES ANOMALY DETECTION

We improve the detection performance with the following two approaches: tuning the number of backpropagation steps and monitoring the residuals of the fitted model. A detailed explanation of how the number of backpropagation steps and using residuals affect the detection performance is elaborated in this Section.

a: THE NUMBER OF BACKPROPAGATION STEPS

Both the MAD-GAN and the TAnoGAN adopt the inverse mapping function $(G(\mathbf{x})^{-1})$ using backpropagation. As backpropagation proceeds using the gradient obtained in Equation (2) or (7), the difference between the generated sample and the test sample continuously decreases. That is,

TABLE 6. The prediction performances for the MA (1) data.

Shift size	Target FPR	Method	F1	Acc.	Rec.	Pre.	FPR
Small ($c = 0.5$)	10%	MAD GAN	17.31	50.44	10.37	52.20	9.50
		TAno GAN	23.07	53.31	14.00	65.50	7.38
		CUSUM	**77.57**	**81.31**	**66.46**	**94.54**	**3.84**
	5%	MAD GAN	9.73	50.12	5.38	51.19	5.13
		TAno GAN	19.06	53.81	10.88	76.99	3.25
		CUSUM	**68.83**	**75.87**	**54.46**	**95.24**	**2.72**
	1%	MAD GAN	1.47	49.75	0.75	97.5	1.25
		TAno GAN	3.41	50.44	1.75	66.67	0.88
		CUSUM	**61.72**	**72.34**	**45.42**	**98.39**	**0.74**
Medium ($c = 1$)	10%	MAD GAN	23.81	52.00	15.00	57.69	11.00
		TAno GAN	39.29	59.44	26.25	78.07	7.38
		CUSUM	**96.87**	**97.34**	**98.51**	**96.25**	**3.84**
	5%	MAD GAN	11.50	50.00	6.50	50.00	6.50
		TAno GAN	35.13	59.38	22.00	87.13	3.25
		CUSUM	**97.41**	**97.90**	**98.51**	**97.31**	**2.72**
	1%	MAD GAN	5.75	50.81	3.00	68.57	1.38
		TAno GAN	11.89	52.75	6.38	87.93	0.88
		CUSUM	**98.07**	**98.58**	**97.90**	**99.25**	**0.74**
Large ($c = 2$)	10%	MAD GAN	47.25	61.62	34.37	75.55	11.12
		TAno GAN	71.98	76.50	60.38	89.11	7.38
		CUSUM	**97.37**	**97.83**	**99.5**	**96.29**	**3.84**
	5%	MAD GAN	32.62	56.88	20.87	74.55	7.13
		TAno GAN	69.41	75.81	54.88	94.41	3.25
		CUSUM	**97.91**	**98.39**	**99.50**	**97.34**	**2.72**
	1%	MAD GAN	21.76	55.50	12.37	90.00	1.38
		TAno GAN	51.24	66.94	34.75	97.54	0.88
		CUSUM	**98.76**	**99.26**	**99.26**	**99.26**	**0.74**

as the number of backpropagation steps Γ increases, the value of the loss function becomes smaller. A sufficiently large value of Γ can make the anomaly score of an abnormal instance close to the anomaly score of a normal instance. This is because the loss functions of both the MAD-GAN and the TAnoGAN have no way to avoid overfitting like regularization terms. So, we examine the relationship between Γ and the prediction performance of the GAN-based AD methods. To test the effect of Γ on performance, we use λ, τ, and θ corresponding to the target FPR of 5%. For example, in AR (1) data sets, λ, τ, and θ are 0.50, 0.50, and 0.28 respectively. The F1 score corresponding to the change in the Γ is summarized in Table 10. As can be represented in Table 10, the prediction performances decrease with increasing Γ except in one case.

TABLE 7. The MAD-GAN's false positive rates corresponding to λ and τ for the ARMA (1,1).

λ \ τ	0.40	0.45	0.50	0.55	0.60	0.65	0.70
0.10	16.19	**11.03**	7.37	**4.60**	2.81	1.66	**0.92**
0.30	20.03	12.46	7.49	4.14	2.15	1.08	0.48
0.50	27.86	15.27	7.77	3.52	1.45	0.49	0.16
0.70	53.51	23.45	8.54	2.43	0.52	0.10	0.00
0.90	99.40	80.43	14.60	0.86	0.03	0.00	0.00

TABLE 8. The user-defined values for the ARMA (1,1).

User-defined value	Target false positive rate		
	10 %	5 %	1 %
λ	0.10	0.10	0.10
τ	0.45	0.55	0.70
θ	0.15	0.20	0.28

b: RESIDUALS FOR ANOMALY DETECTION

Using residuals by removing time series effects from the original data is a widely applied approach to time series data [19], [20]. The residual can be defined as:

$$e_i = x_i - \hat{x_i}, \tag{15}$$

where $\hat{x_i}$ is the fitted value of x_i. The three methods show the poor prediction performance in the ARMA (1,1) data sets, so we apply the residuals only to the ARMA (1,1) data sets. The LSTM is used to obtain the residuals. The contents of the user-defined value selection are summarized in Tables 11 and 12. Table 13 shows the prediction performances using the residual data. Compared with Table 9, the prediction performances of the three methods in Table 13 are remarkably improved. Unlike the previous simulation results, the prediction performances are not always improved even if the shift size increases. In addition, in Table 9, the F1 scores of the three methods decrease as the target FPR decrease, but in Table 13, the F1 scores of the three methods do not differ significantly corresponding to the decrease of the target FPR.

C. MULTIVARIATE SIMULATION EXPERIMENTS

Similar to the previous section, the process and results of multivariate simulation experiments are discussed in this section.

TABLE 9. The prediction performances for the ARMA (1,1) data.

Shift size	Target FPR	Method	F1	Acc.	Rec.	Pre.	FPR
Small ($c = 0.5$)	10%	MAD GAN	20.70	50.19	13.00	50.73	12.62
		TAno GAN	**20.96**	49.56	**13.38**	48.42	14.25
		CUSUM	15.32	**52.17**	8.79	**66.35**	**4.46**
	5%	MAD GAN	**10.95**	50.19	**6.13**	51.25	5.75
		TAno GAN	10.09	49.88	5.63	48.91	5.88
		CUSUM	7.72	**50.56**	4.21	**57.63**	**3.09**
	1%	MAD GAN	1.23	49.81	0.63	38.46	1.00
		TAno GAN	**1.96**	**49.88**	**1.00**	**44.44**	1.25
		CUSUM	0.00	49.38	0.00	0.00	**1.24**
Medium ($c = 1$)	10%	MAD GAN	19.44	49.75	12.12	48.99	12.62
		TAno GAN	20.08	49.25	12.75	47.22	14.25
		CUSUM	**21.88**	**54.27**	**13.00**	**74.47**	**4.46**
	5%	MAD GAN	10.20	**51.56**	5.50	**69.84**	**2.38**
		TAno GAN	9.23	49.62	5.13	46.59	5.88
		CUSUM	**10.75**	51.42	**5.94**	65.75	3.09
	1%	MAD GAN	0.49	49.62	0.25	20.00	1.00
		TAno GAN	**2.68**	**50.06**	**1.38**	**52.38**	1.25
		CUSUM	0.00	49.38	0.00	0.00	**1.24**
Large ($c = 2$)	10%	MAD GAN	18.35	49.38	11.37	47.40	12.62
		TAno GAN	20.43	49.38	13.00	47.71	14.25
		CUSUM	**43.04**	**62.25**	**28.96**	**86.67**	**4.46**
	5%	MAD GAN	8.16	49.38	4.50	43.90	5.75
		TAno GAN	10.94	50.12	6.13	51.04	5.88
		CUSUM	**24.20**	**55.63**	**14.36**	**82.27**	**3.09**
	1%	MAD GAN	1.23	49.75	0.63	65.71	1.13
		TAno GAN	4.58	50.56	2.38	65.52	1.25
		CUSUM	**6.85**	**51.18**	**3.59**	**74.36**	**1.24**

TABLE 10. The F1 score of the GAN-based methods corresponding to the change for the Γ when the target false positive rate is 5 %.

Method	Data set	Shift size	Γ values				
			10	50	100	500	1000
MAD GAN	AR (1)	Small	**16.70**	16.14	16.36	16.16	16.16
		Medium	**28.69**	23.71	22.99	22.99	22.99
		Large	**60.22**	48.89	39.50	35.70	35.70
	MA (1)	Small	**9.73**	8.62	8.63	8.63	8.63
		Medium	**11.50**	9.32	9.31	9.53	9.53
		Large	**32.62**	20.90	13.07	12.02	12.02
	ARMA (1,1)	Small	**10.95**	8.45	7.78	7.78	7.77
		Medium	**9.24**	9.12	8.45	7.77	7.77
		Large	8.16	**10.20**	**10.20**	9.98	9.76
TAno GAN	AR (1)	Small	**5.15**	0.00	0.00	0.00	0.00
		Medium	**6.05**	0.25	0.25	0.25	0.25
		Large	**21.44**	5.82	5.58	5.82	5.82
	MA (1)	Small	**19.06**	1.98	1.73	1.98	1.98
		Medium	**35.13**	9.50	9.28	8.84	8.84
		Large	**69.41**	44.72	43.71	43.29	42.98
	ARMA (1,1)	Small	**10.09**	2.71	2.71	2.71	2.71
		Medium	**9.23**	3.44	3.19	3.19	3.19
		Large	**10.94**	5.82	6.05	6.05	6.05

TABLE 11. The MAD-GAN's false positive rates corresponding to λ and τ for the residual data.

λ \ τ	0.30	0.35	0.40	0.45	0.50	0.55	0.60
0.10	**10.79**	6.09	3.35	1.92	1.08	0.62	0.34
0.30	20.20	9.77	**4.65**	2.21	1.04	0.48	0.21
0.50	57.11	23.14	8.19	2.91	**1.02**	0.32	0.09
0.70	100.00	96.32	29.90	5.87	1.04	0.11	0.02
0.90	100.00	100.00	99.97	75.94	2.24	0.13	0.05

1) MULTIVARIATE SIMULATION DATA SETS

When it comes to multivariate time series data for a comparison of prediction performances among the three methods, multistage data sets are used for AD. In this paper, we leverage the previous work [42] to generate the multistage data. Hwang *et al.* [42] considered a three-stage process for monitoring, which is adapted with the new autoregressive factor in this study. The generated multistage data sets is converted to the deviance residuals according to the previous works [43].

Two input variables, denoted by x_{i1}, x_{i2}, and five output variables, denoted by $y_{i1}, y_{i2}, y_{i3}, y_{i4}, y_{i5}$, are generated at the

TABLE 12. The user-defined values for the residual data.

User-defined value	Target false positive rate		
	10 %	5 %	1 %
λ	0.10	0.30	0.50
τ	0.30	0.40	0.50
θ	0.15	0.19	0.31

ith stage ($i = 1, 2, 3$). Each of the two input variables follows the standard normal distribution. The five output variables each follow the beta distribution with logit link. The mean of the beta distribution for the first stage is described as:

$$\mu_{y_{1j(t)}(x_{11(t)},x_{12(t)},y_{1j(t-1)})} = \frac{\exp\left(1 + 0.5x_{11(t)} + 0.25x_{12(t)} + 3y_{1j(t-1)}\right)}{1 + \exp\left(1 + 0.5x_{11(t)} + 0.25x_{12(t)} + 3y_{1j(t-1)}\right)}. \quad (16)$$

The mean of the beta distribution for the second stage as

$$\mu_{y_{2j(t)}(x_{21(t)},x_{22(t)},y_{2j(t-1)})} = \frac{\exp\left(1+0.5x_{21(t)}+0.25x_{22(t)}+y_{11(t-1)}+2y_{2j(t-1)}\right)}{1+\exp\left(1+0.5x_{21(t)}+0.25x_{22(t)}+y_{11(t-1)}+2y_{2j(t-1)}\right)}, \quad (17)$$

and the mean of the beta distribution for the third stage as

$$\mu_{y_{3j(t)}(x_{31(t)},x_{32(t)},y_{3j(t-1)})} = \frac{\exp\left(1+0.5x_{31(t)}+0.25x_{32(t)}+y_{21(t-1)}+2y_{3j(t-1)}\right)}{1+\exp\left(1+0.5x_{31(t)}+0.25x_{32(t)}+y_{21(t-1)}+2y_{3j(t-1)}\right)}, \quad (18)$$

where j is from 1 to 5, t is the time sequence, and the shape parameter of the beta distribution is 100. Equation (16) represents the first stage with no the preceding stage. On the other hand, the preceding stage outputs, $y_{11(t-1)}$ and $y_{21(t-1)}$ are respectively considered as an input variable in Equations (17)-(18).

4,999 training instances at each stage are generated from Equations (16)-(18). Similarly, 4,999 instances at each stage are given as a test data set but the characteristics of each test data set are different for each stage. For stage 1, 2,500 instances among the 4,999 instances are collected under the abnormal condition where the intercept coefficient in Equation (16) increases from 1.0 to 1.2, whereas 2,499 instances under the normal condition. Unlike stage 1, test data sets for stage 2 and 3 are composed of only normal instances.

2) SIMULATION SETUP FOR MULTIVARIATE DATA SETS

Along with univariate data sets, the same procedure for simulation setup is applied to the multivariate data sets: data preprocessing, building the architecture of GAN and setting predetermined values.

TABLE 13. The prediction performance for the residual data.

Shift size	Target FPR	Method	F1	Acc.	Rec.	Pre.	FPR
Small ($c = 0.5$)	10%	MAD GAN	87.92	88.06	86.87	88.99	10.75
		TAno GAN	89.57	89.56	89.62	89.51	10.50
		CUSUM	**97.17**	**97.09**	**99.75**	**94.71**	**5.57**
	5%	MAD GAN	87.36	88.19	81.62	93.96	5.25
		TAno GAN	91.14	91.44	88.12	94.38	5.25
		CUSUM	**97.82**	**97.77**	**99.75**	**95.95**	**4.21**
	1%	MAD GAN	84.14	86.12	73.62	98.17	**1.38**
		TAno GAN	89.13	90.06	81.50	**98.34**	**1.38**
		CUSUM	**98.89**	**98.89**	**99.63**	98.17	1.86
Medium ($c = 1$)	10%	MAD GAN	88.06	88.19	87.12	89.09	10.75
		TAno GAN	89.36	89.38	89.25	89.47	10.50
		CUSUM	**97.10**	**97.03**	**99.63**	**94.71**	**5.57**
	5%	MAD GAN	88.09	88.81	82.75	94.17	5.13
		TAno GAN	91.35	91.62	88.50	94.40	5.25
		CUSUM	**97.63**	**97.59**	**99.38**	**95.94**	**4.21**
	1%	MAD GAN	83.98	86.00	73.37	98.16	**1.38**
		TAno GAN	90.03	90.81	83.00	**98.37**	**1.38**
		CUSUM	**98.27**	**98.27**	**98.39**	98.15	1.86
Large ($c = 2$)	10%	MAD GAN	88.62	88.69	88.12	89.13	10.75
		TAno GAN	90.05	90.00	90.50	89.60	10.50
		CUSUM	**96.42**	**96.35**	**98.27**	**64.64**	**5.57**
	5%	MAD GAN	87.94	88.69	82.50	94.15	5.13
		TAno GAN	91.07	91.38	88.00	94.37	5.25
		CUSUM	**96.75**	**96.72**	**97.65**	**95.87**	**4.21**
	1%	MAD GAN	85.05	86.81	75.00	98.20	**1.38**
		TAno GAN	90.33	91.06	83.50	**98.38**	**1.38**
		CUSUM	**97.89**	**97.90**	**97.65**	98.13	1.86

a: DATA PREPROCESSING

We first apply normalization to the multistage data sets. After that, the original sequence is divided into subsequences through the sliding window. For multistage data sets, window size $w = 60$, step size $s = 10$ for training GAN and $w = 60$,

$s = 60$ for an AD are chosen. In other words, 493 samples are extracted for training GAN and 82 samples are used to calculate the anomaly score. Although 5 variables are generated from the multistage data sets, the CUSUM chart is only available for univariate, so, this Section only considers univariate for fair comparisons (the multivariate GAN-based AD methods will be considered in Section V). In order to have the same dimensions, we perform principal component analysis (PCA), and then, the 1st principal component (PC) is monitored to detect anomalies.

b: GAN ARCHITECTURE

The followings are the architecture for both a generator and a discriminator. The multistage data sets are relatively complicated compared to the univariate simulation data sets. Thus, the number of hidden nodes is increased to 100 and the number of latent space is 15. The GAN is trained with the Adam optimizer with a learning rate of 0.10. The generator's parameters are updated three times, whereas the discriminator's parameters one time. To achieve reasonable performance, we tried sufficient iterations by setting 100 epoch.

c: SET PREDETERMINED VALUES

The process for comparing three methods is the same as that of the univariate simulation. Since the MMD values converge to 0 after 70 epoch, the generator and discriminator of 70 epoch are selected and used to calculate anomaly scores. Fig. 4 shows the MMD values corresponding to each epoch. Table 14 shows the FPR values in the multistage training data sets with varying λ and τ. The most similar values to the target FPRs are also marked in bold. Table 15 presents the selected user-defined values concerning the target FPRs.

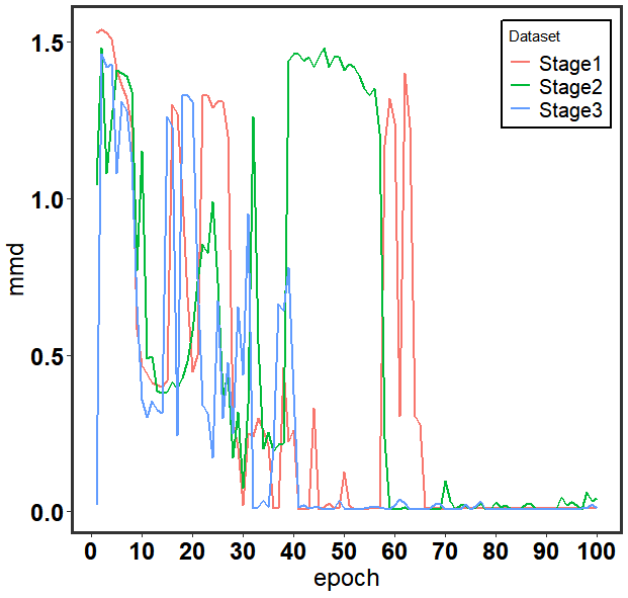

FIGURE 4. The MMD values during GAN training with the multistage dataset for each stage.

TABLE 14. The MAD-GAN's false positive rates corresponding to λ and τ for the multistage dataset for each stage.

Stage 1					
λ \ τ	0.10	0.30	0.50	0.70	0.90
0.10	77.60	24.65	**5.22**	**1.00**	0.08
0.30	77.74	21.18	3.46	0.35	0.00
0.50	76.65	14.39	5.26	0.59	0.00
0.70	73.31	**10.35**	5.45	3.66	0.00
0.90	60.57	9.39	5.63	4.53	1.59
Stage 2					
λ \ τ	0.50	0.60	0.70	0.80	0.90
0.40	26.00	**9.92**	3.50	**1.06**	0.20
0.50	42.07	15.65	**5.12**	1.38	0.22
0.60	66.14	28.76	8.94	2.20	0.28
0.70	78.58	52.34	17.28	3.88	0.49
0.80	83.78	68.62	39.53	9.02	0.96
Stage 3					
λ \ τ	0.50	0.60	0.70	0.80	0.90
0.10	**10.12**	**4.92**	2.18	0.85	0.20
0.20	10.22	4.61	1.85	0.57	0.12
0.70	13.62	2.62	0.57	0.14	0.00
0.80	15.65	2.50	0.67	0.14	0.00
0.90	19.94	3.15	**0.87**	0.20	0.00

TABLE 15. Predetermined user-defined values with respect to target FPRs for multistage data for each stage.

Stage 1			
User-defined value	Target false positive rate		
	10 %	5 %	1 %
λ	0.70	0.90	0.90
τ	0.30	0.70	0.90
θ	1.00	1.60	1.80
Stage 2			
λ	0.40	0.50	0.40
τ	0.60	0.70	0.80
θ	0.02	0.04	0.09
Stage 3			
λ	0.90	0.10	0.10
τ	0.70	0.60	0.50
θ	0.02	0.04	0.09

3) MULTIVARIATE SIMULATION RESULTS

As mentioned in the previous Section, the number of backpropagation steps (Γ) affects the prediction performance of the GAN-based AD methods. The F1 scores corresponding to the change in the Γ are summarized in Table 16. Table 16 shows that the smaller Γ, the higher the F1 score of the MAD-GAN. In addition, the F1 score of the MAD-GAN is not affected when the number of backpropagation steps is more than 500. In terms of the TAnoGAN, the proper

TABLE 16. The F1 score of the GAN-based AD methods corresponding to the change for the multistage data.

Method	Target FPR	Γ values				
		10	50	100	500	1000
MAD GAN	10%	**21.69**	21.25	21.16	19.86	19.86
	5%	**14.47**	12.24	9.91	9.72	9.72
	1%	**6.69**	5.04	2.43	2.43	2.43
TAno GAN	10%	16.92	**21.63**	19.28	19.61	20.86
	5%	9.05	6.45	10.73	**11.21**	10.32
	1%	2.27	0.15	2.49	**2.96**	2.50

number of backpropagation steps depends on the target FPR. We select the Γ that produced the best F1 score for comparisons in Table 16.

Since there are no abnormal instances in stage 2 and stage 3 data sets, the evaluation metrics are only computed for stage 1 data sets. Abnormal instances that occurred in stage 1 can propagate to other stages because the input and output variables of stage 1 affect stage 2 and stage 3. However, the three methods of stage 2 and stage 3 do not detect anomalies because the abnormal instances occur only in the stage 1. The results of the prediction performance of the three methods using the multistage data are presented in Table 17.

TABLE 17. The prediction performance for the multistage data.

Target FPR	Method	F1	Acc.	Rec.	Pre.	FPR
10%	MAD GAN	**21.69**	50.53	**13.48**	55.43	11.20
	TAno GAN	21.63	51.24	13.24	59.00	9.50
	CUSUM	21.56	**52.35**	13.32	**60.77**	**8.60**
5%	MAD GAN	14.47	49.78	8.36	53.73	7.44
	TAno GAN	11.21	49.45	6.28	52.16	5.95
	CUSUM	**15.17**	**51.99**	**8.72**	**64.88**	**4.72**
1%	MAD GAN	6.69	49.51	3.56	54.94	3.02
	TAno GAN	2.96	49.27	1.52	52.78	**1.41**
	CUSUM	**7.13**	**51.17**	**3.80**	**72.52**	1.44

Similar to the results of the ARMA (1,1) data sets, all three methods show the prediction performance with about 50% accuracy. Also, when the target FPR is 10%, none of the three methods overwhelms the others. In order words, with the target FPR of 10%, similar prediction performances in the F1 score are observed in all methods. However, when the target FPR is equal to 5% and 1%, the CUSUM chart demonstrate prediction performances better than the GAN-based methods.

V. CASE STUDY

In this Section, we consider whether the prediction performance of the GAN-based AD methods varies under the proposed comparison framework. Therefore, the prediction performance of the GAN-based AD methods and the CUSUM chart is compared according to the proposed comparison framework.

A. TRAINING GAN

In this Section, we consider the multivariate GAN-based AD methods using real data. The SWaT data were collected for a total of 11-days under the non-stop 24 hours operating system. During this period, any attacks had not been detected for the first seven days, while certain cyber and physical attacks were launched for the remaining four days [44]. It contains 51 variables with 496,800 instances in the training data sets and 449,919 instances in the test data sets. According to the previous studies [15], [18], we eliminate the first 21,600 instances from the training data sets and the number of instances in the training data sets becomes 475,200. Test data sets contain 395,298 normal instances (about 88%) and 54,621 abnormal instances (about 12%). Same as the previous work [18], we set up window size $w = 120$ and step size $s = 10$ for training GAN and $w = 120$, $s = 120$ to detect anomalies. Consequently, we obtain 47,508 training samples and 3,293 test samples.

The PCA is applied to the SWaT system data to reduce dimensions and computational costs. We plot the explained variance in Fig. 5. As shown in Fig. 5, the 1st PC explains more than 50 % of the variance in the SWaT system data. The PCs after the 5th PC rarely contribute to explaining the overall variance (close to zero). Thus, we only consider up to the 5th PCs for the GAN-based AD methods. In this case study, we follow the GAN architecture from the previous work [18]. So, we use 100 hidden units, 1 hidden layer, and 15 latent space dimensions for training GAN. The mini-batch optimization based on the Adam optimizer is applied to both generator and discriminator with a batch size of 500 and a learning rate of 0.1.

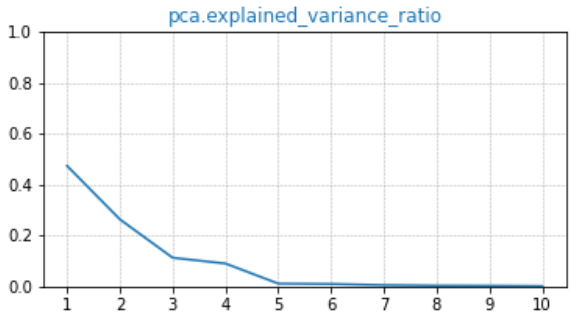

FIGURE 5. The explained variance ratio for the SWaT system data.

B. ANOMALY DETECTION SETUP

Li *et al.* [18] indicated that the MMD values converge to small values after 30 epoch. So, the generator and discriminator at 30 epoch are used to calculate the anomaly scores. When the target FPR is 5%, the user-defined values λ, τ, and θ are 0.30, 1.60, and 1.52, respectively. And the number of backpropagations steps Γ is 100.

Since the baseline model is the univariate CUSUM chart, modifications are required to handle multivariate data. In the previous study [18], the MAD-GAN was compared with the SPE-based (squared predicted error) CUSUM chart. The SPE chart is suggested to detect when anomalies appear on excluded PCs [45]. For example, assuming that we use up to the 5th PC for dimensionality reduction, the SPE chart attempts to detect anomalies occurring in the remaining 46 PCs. However, as can be seen in Fig. 4, the PCs after the 5th one hardly contribute to account for the total variance (close to zero). In contrast, 1st PC accounts for more than 50% of the total variance. Therefore, instead of utilizing the SPE-based CUSUM chart, we only apply the 1st PC to the CUSUM chart.

C. ANOMALY DETECTION RESULTS

Table 18 reveals the result of prediction performance in the SWaT system data. As shown in Table 18, in terms of F1 score and accuracy, the baseline method performs the best for the SWaT data. In our study, the MAD-GAN shows prediction ability lower than previous studies [15], [18]. There are two main reasons for this. First, the generator and the discriminator used to detect anomalies are different. In their works, it is not clear at which epoch they used a generator and a discriminator. On the other hand, in this paper, we use a generator and a discriminator at 30 epoch for fair comparisons. Second, the target FPR was not considered in previous work [15]. In Section IV, the simulation results indicate that the prediction performance varies depending on the target FPR, but no information related to the target FPR is provided in the previous works [15], [18]. In contrast, the target FPR is set to 5% in this Section V. The above two reasons may cause a different prediction performance.

TABLE 18. Prediction performance for the SWaT system data.

	MAD-GAN	TAnoGAN	CUSUM
F1 Score	36.77	3.83	**61.68**
Accuracy	69.68	83.67	**90.67**
Recall	**72.60**	2.68	63.19
Precision	24.62	6.73	**61.20**
FPR	30.72	**5.13**	5.54

VI. CONCLUSION

In this paper, the following studies are conducted on the GAN-based AD methods. First, the framework for comparisons is presented and comparative studies are conducted under the proposed comparison framework. Not to mention the real data set, a total of 11 simulation data sets are also considered for testing. In the simulation study, the traditional time series models (AR, MA, and ARMA) and the residuals are monitored. A lot of comparisons have been conducted under the proposed comparison framework. We believe that the framework is still valid for other real data sets because it is theoretically based on the hypothesis testing and the SPC. In our experiments, the CUSUM chart shows the prediction performance better than the others except for the ARMA (1,1) data and the multistage data. In the ARMA (1,1) and the multistage data sets, none of the three methods perform better than the others. Also, as shown in Table 9 and 17, the three methods show prediction performances with approximately 50% accuracy. Besides, in the SWaT system data, the CUSUM chart using only the 1st PC produces the highest F1 score and accuracy. Although the GAN-based methods do not always perform well, they can detect anomalies for the big shift size ($c = 2$), as shown in Tables 3 and 6. Our experimental results demonstrate that the prediction performances of the GAN-based AD methods depend on the comparison framework. Therefore, more follow-up studies on the GAN-based AD methods are required.

Second, compared with previous studies [15], [16], [18], the parameter Γ that affects the prediction performance is dealt with in detail. As a result, we find that a small Γ is more appropriate for simple data such as the AR (1) and the MA (1). Finally, as shown in Table 13, monitoring the residuals to detect anomalies can improve the prediction performance.

There can be interesting directions for future research. One such direction is an additional term such as L1 regularization to the loss function to obtain the anomaly scores. Our paper has demonstrated that overfitting can occur during the inverse mapping process. It will be a very interesting study to improve the prediction performance by adding a regularization term to the loss function to avoid overfitting.

REFERENCES

[1] D. Dasgupta and S. Forrest, "Novelty detection in time series data using ideas from immunology," in *Proc. Int. Conf. Intell. Syst.*, Jun. 1996, pp. 82–87.
[2] V. Guralnik and J. Srivastava, "Event detection from time series data," in *Proc. 5th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, 1999, pp. 33–42.
[3] J. Ma and S. Perkins, "Time-series novelty detection using one-class support vector machines," in *Proc. Int. Joint Conf. Neural Netw.*, Portland, OR, USA, 2003, pp. 1741–1745, doi: 10.1109/IJCNN.2003.1223670.
[4] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han, "Outlier detection for temporal data: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, Sep. 2014, doi: 10.1109/TKDE.2013.184.
[5] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, Oct. 2019.
[6] T. Schlegl, P. Seeböck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," in *Proc. Int. Conf. Inf. Process. Med. Imag.* Cham, Switzerland: Springer, 2017, pp. 146–157.
[7] M. Mohammadi, A. Al-Fuqaha, S. Sorour, and M. Guizani, "Deep learning for IoT big data and streaming analytics: A survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2923–2960, 4th Quart., 2018.
[8] J. E. Ball, D. T. Anderson, and C. S. Chan, "Comprehensive survey of deep learning in remote sensing: Theories, tools, and challenges for the community," *J. Appl. Remote Sens.*, vol. 11, no. 04, p. 1, Sep. 2017.
[9] B. Kiran, D. Thomas, and R. Parakkal, "An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos," *J. Imag.*, vol. 4, no. 2, p. 36, Feb. 2018.
[10] M. Basseville and Nikiforov, "Introduction to Part II," in *Detection of Abrupt Changes: Theory and Application*, vol. 104. Englewood Cliffs, NJ, USA: Prentice-Hall, 1993, pp. 200–201.
[11] D. C. Montgomery and C. M. Mastrangelo, "Some statistical process control methods for autocorrelated data," *J. Qual. Technol.*, vol. 23, no. 3, pp. 179–193, Jul. 1991, doi: 10.1080/00224065.1991.11979321.

[12] P. Malhotra, A. Ramakrishnan, G. Anand, L. Vig, P. Agarwal, and G. Shroff, "LSTM-based encoder-decoder for multi-sensor anomaly detection," 2016, *arXiv:1607.00148*. [Online]. Available: http://arxiv.org/abs/1607.00148

[13] D. Park, Y. Hoshi, and C. C. Kemp, "A multimodal anomaly detector for robot-assisted feeding using an LSTM-based variational autoencoder," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1544–1551, Jul. 2018, doi: 10.1109/LRA.2018.2801475.

[14] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. Eur. Symp. Artif. Neural Netw.*, vol. 89. Ottignies-Louvain-la-Neuve, Belgium: Presses universitaires de Louvain, Apr. 2015, pp. 89–94.

[15] 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 *Proc. Int. Conf. Artif. Neural Netw.* Cham, Switzerland: Springer, Sep. 2019, pp. 703–716.

[16] M. A. Bashar and R. Nayak, "TAnoGAN: Time series anomaly detection with generative adversarial networks," in *Proc. IEEE Symp. Comput. Intell. (SSCI)*, Canberra, ACT, Australia, Dec. 2020, pp. 1778–1785, doi: 10.1109/SSCI47803.2020.9308512.

[17] A. Geiger, D. Liu, S. Alnegheimish, A. Cuesta-Infante, and K. Veeramachaneni, "TadGAN: Time series anomaly detection using generative adversarial networks," 2020, *arXiv:2009.07769*. [Online]. Available: http://arxiv.org/abs/2009.07769

[18] D. Li, D. Chen, J. Goh, and S.-k. Ng, "Anomaly detection with generative adversarial networks for multivariate time series," 2018, *arXiv:1809.04758*. [Online]. Available: http://arxiv.org/abs/1809.04758

[19] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, "Aspects of process control," in *Time Series Analysis: Forecasting and Control*, 5th ed. Hoboken, NJ, USA: Wiley, 2015, pp. 565–566.

[20] D. C. Montgomery, "Cumulative sum and exponentially weighted moving average control charts," in *Introduction to Statistical Quality Control: A Modern Introduction*, 6th ed. New York, NY, USA: Wiley, 2009, ch. 9, pp. 404–408.

[21] J. Goh, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *Proc. IEEE 18th Int. Symp. High Assurance Syst. Eng. (HASE)*, Singapore, 2017, pp. 140–145, doi: 10.1109/HASE.2017.36.

[22] D. C. Montgomery, "Other univariate statistical process monitoring and control techniques," in *Introduction to Statistical Quality Control: A Modern Introduction*, 6th ed. New York, NY, USA: Wiley, 2009, ch. 10, pp. 450–451.

[23] P. Qiu, "Univariate CUSUM chart," in *Introduction to Statistical Process Control*, 1st ed. Boca Raton, FL, USA: CRC Press, 2013, ch. 4, pp. 139–141.

[24] B. Mesnil and P. Petitgas, "Detection of changes in time-series of indicators using CUSUM control charts," *Aquatic Living Resour.*, vol. 22, no. 2, pp. 187–192, Apr. 2009.

[25] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," 2019, *arXiv:1901.03407*. [Online]. Available: http://arxiv.org/abs/1901.03407

[26] P.-N. Tan, M. Steinbach, and V. Kumar, "Anomaly detection," in *Introduction to Data Mining*. London, U.K.: Pearson, 2016, ch 10, pp. 655–656.

[27] D. Wulsin, J. Blanco, R. Mani, and B. Litt, "Semi-supervised anomaly detection for EEG waveforms using deep belief nets," in *Proc. 9th Int. Conf. Mach. Learn. Appl.*, Washington, DC, USA, Dec. 2010, pp. 436–441, doi: 10.1109/ICMLA.2010.71.

[28] M. Nadeem, O. Marshall, S. Singh, X. Fang, and X. Yuan, "Semisupervised deep neural network for network intrusion detection," in *Proc. KSU Conf. Cybersecur. Educ. Res. Pract.*, Oct. 2016, pp. 1–13.

[29] H. Song, Z. Jiang, A. Men, and B. Yang, "A hybrid semi-supervised anomaly detection model for high-dimensional data," *Comput. Intell. Neurosci.*, vol. 2017, pp. 1–9, 2017, doi: 10.1155/2017/8501683.

[30] M. M. Breunig, H. P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. conference Manage. Data*, Dallas, TX, USA, 2000, pp. 93–104.

[31] J. Tang, Z. Chen, A. W. C. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Proc. Pacific-Asia Conf. Knowl. Discovery Data Mining*. Berlin, Germany: Springer, May 2002, pp. 535–548.

[32] Z. He, X. Xu, and S. Deng, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, nos. 9–10, pp. 1641–1650, Jun. 2003.

[33] I. Goodfellow, J. Pouget-Abadie, M.Mirza, B. Xu, D. Warde-Farley, S. Ozair, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

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

[35] F. A. Gers, "Learning to forget: Continual prediction with LSTM," in *Proc. 9th Int. Conf. Artif. Neural Networks: ICANN*, vol. 2. Edinburgh, U.K., 1999, pp. 850–855, doi: 10.1049/cp:19991218.

[36] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber, "Learning precise timing with LSTM recurrent networks," *J Mach. Learn. Res*, vol. 3, pp. 115–143, Aug. 2002.

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*. [Online]. Available: http://arxiv.org/abs/1606.03498

[38] P. Qiu, "Univariate shewhart charts and process capability," in *Introduction to Statistical Process Control*, 1st ed. Boca Raton, FL, USA: CRC Press, 2013, ch. 4, pp. 79–80.

[39] W. Hwang, G. Runger, and E. Tuv, "Multivariate statistical process control with artificial contrasts," *IIE Trans.*, vol. 39, no. 6, pp. 659–669, Mar. 2007.

[40] E. S. Page, "Continuous inspection schemes," *Biometrika*, vol. 41, nos. 1–2, pp. 100–115, 1954, doi: 10.1093/biomet/41.1-2.100.

[41] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (Medical) time series generation with recurrent conditional GANs," 2017, *arXiv:1706.02633*. [Online]. Available: http://arxiv.org/abs/1706.02633

[42] W. Hwang, K. N. Al-Khalifa, A. M. S. Hamouda, M. K. Jeong, and E. A. Elsayed, "Multistage statistical process control for parametric variables," *WIT Trans. Eng. Sci.*, vol. 108, pp. 3–12, Mar. 2015, doi: 10.2495/QR2MSE140011.

[43] S. Kim, J. Kim, M. K. Jeong, K. N. Al-Khalifa, A. M. S. Hamouda, and E. A. Elsayed, "Monitoring and control of beta-distributed multistage production processes," *Qual. Technol. Quant. Manage.*, vol. 16, no. 1, pp. 1–18, Jan. 2019.

[44] G. Jonathan, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *Proc. Int. Conf. Crit. Inf. Infrastruct. Secur.*, 2016, pp. 88–99.

[45] J. E. Jackson and G. S. Mudholkar, "Control procedures for residuals associated with principal component analysis," *Technometrics*, vol. 21, no. 3, pp. 341–349, Aug. 1979.

CHANG-KI LEE received the Ph.D. degree in operations management from Dongguk University, in 2019. He is currently a Postdoctoral Researcher with the Research Center for Industry-Academy Cooperation, Dong-A University, Busan, South Korea. His current research interests include business analytics and quality management.

YU-JEONG CHEON received the bachelor's degree in global business from Dong-A University, in 2021, where she is currently pursuing the degree in management information. Her research interests include deep learning, intrusion detection, and text data mining.

WOOK-YEON HWANG received the M.S. degree in industrial engineering from Arizona State University, in 2004, and the Ph.D. degree in statistics with North Carolina State University, in 2009. He is currently an Associate Professor with the Department of Global Business, Dong-A University, Busan, South Korea. His current research interests include business analytics and quality engineering. In 2008, he received the Best Application Paper Award in quality and reliability engineering from the *IIE Transactions*.