IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 16, 2022, accepted March 7, 2022, date of publication March 10, 2022, date of current version March 15, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3158343

Least Squares Generative Adversarial Networks-Based Anomaly Detection

CHANG-KI LEE[1], YU-JEONG CHEON[2], AND WOOK-YEON HWANG[3]
[1]Quality Innovation Team, Global CS Center, Samsung Electronics, Suwon 16677, 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 Dong-A University Research Fund.

ABSTRACT Multivariate statistical process control (MSPC) is a technique for detecting anomalies by monitoring several quality characteristics simultaneously. For the MSPC problem, the Hotelling's T^2 control chart has been widely used as a typical method. Recently, researchers have converted the MSPC problem into a classification problem such as the artificial contrast (AC) and the one-class classification (OCC). Previous studies have shown that these methods outperform the Hotelling's T^2 chart when the data do not follow a multivariate normal distribution. However, unless the size of the process data is enough for the AC and the OCC, they cannot work properly. To tackle this problem, in this paper, we propose a novel anomaly detection (AD) approach. The proposed method adopts the least square generative adversarial network (LS-GAN) to estimate the probability distribution of the training data. It generates new training samples from the learned probability distribution. The classifiers such as the random forests (RF) and the one-class support vector machines (OC-SVM) are considered for tackling the AC and the OCC respectively. The numerical experiments demonstrate that the proposed approach outperforms the existing methods in terms of the area under the receiver operating characteristic (ROC) curve (AUC).

INDEX TERMS Anomaly detection, artificial contrast, one-class classification, least square generative adversarial network, Hotelling's control boundary, random forests, one-class support vector machines.

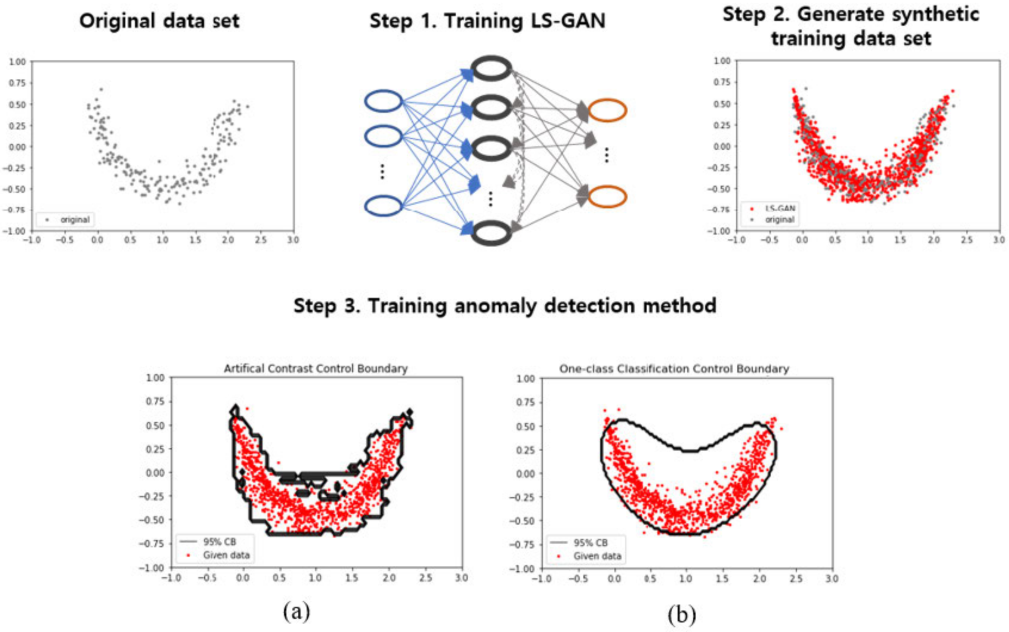
I. INTRODUCTION

Statistical process control (SPC) is widely used in various industries to monitor and improve output quality. Univariate control charts typically used in the SPC are the Shewhart control chart, the cumulative sum (CUSUM) chart, the exponentially weighted moving average (EWMA) chart, etc. However, they cannot consider correlation among two or more related quality characteristics. Applying two or more independent univariate control charts still fails to capture the correlation among the quality characteristics.

Hence, in order to monitor the correlation between the quality characteristics effectively, the multivariate statistical process control (MSPC) is suggested. The MSPC is a widely known technique for simultaneously monitoring several quality characteristics. In tradition, the Hotelling's T^2 control chart, the multivariate CUSUM chart and the multivariate EWMA chart have been used for many years for the MSPC. The Hotelling's T^2 control is defined as $T^2 = \left(\mathbf{X} - \bar{X}\right)^T \mathbf{S}^{-1}(\mathbf{X} - \bar{X})$, where $\bar{X}$ indicates the sample mean vector and $\mathbf{S}$ is the covariance matrix from the normal data [1]. Followed by the Hotelling's T^2 control chart, there have been many attempts to a comparative analysis of diverse multivariate control techniques [2]. However, although the Hotelling's T^2 control chart has been widely used for many years, it has a limitation in that a false alarm is frequently raised if the process data do not follow a multivariate normal distribution. In order to remedy the limitation, the artificial contrast (AC) and the one-class classification (OCC) were proposed.

The AC is one of the monitoring approaches for the MSPC problem. The key concept of the AC is to generate the artificial data from the uniform distributions and create labels (classes) to build the binary classification model. The numerous AC-based monitoring approaches have been proposed over the past few years. Hwang *et al.* [3] proposed the AC that converts the MSPC problem into the binary classification problem. Hwang and Lee [4] proposed a novel approach that can be applied to extremely imbalanced data with a system failure by shifting the artificial data.

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



(a)

(b)

FIGURE 1. An overview of the proposed approach.

The cluster-based AC was also proposed for monitoring the inhomogeneous multivariate processes without the normality assumption [5].

Besides, the OCC is also considered as an another method for the process monitoring that contains only target group (or normal group) and is used to determine the degree of abnormality of new samples [6]. First created by Moya and Hush [7], the OCC using only normal samples to train the model has been widely used until today.

The one-class support vector machine (OC-SVM) has been noted as a typical example for the OCC based on the SVM. The SVM [8] has been widely used for the binary classification problem, aiming to find the optimal control boundary that can maximize the generalization ability by enlarging the margin between the two classes. Similar to the SVM, the main purpose of the OC-SVM is to construct the control boundary around the positive samples in order to differentiate the outliers (non-positives) from the positive data. The SVM-based algorithms to deal with the OCC were proposed by Tax and Duin [9] and Schölkopf *et al.* [10]. The support vector data description (SVDD) that is the data description method using the kernel functions for solving the OCC was proposed [9]. This method differentiates between the two classes using a hyper-sphere, not a hyper-plane, around the positive class samples. As an extension of the classifiers for the OCC, the deep learning-based methods for solving the OCC have also been introduced in recent years. The one-class convolutional neural networks (CNN) which are inspired by the OC-SVM was suggested [11]. Also, the deep SVDD aiming to find the smallest hyper-sphere in the feature space of the data set learned from the CNN was proposed [12].

The AC and the OCC are useful for building control boundaries by classifiers, but do not work properly for the limited number of samples. For the case, we assume that newly generated samples from a generative model can improve the prediction performance of the AC and the OCC. So, in this paper, to enhance the detection performance of the AC and the OCC with the limited number of samples, we generate new training samples using the GAN [13]. As a deep learning-based generative model, the GAN is an effective tool for generating new samples that did not exist in the original data set. Not to mention the image generator field, the GAN has been applied in time-series [14], [15] and has surged in popularity more recently.

The GAN consists of two adversarial networks (a generator and a discriminator) that compete with each other in order to generate realistic samples. Through adversarial training, the generator becomes capable of generating new samples by accurately learning the distribution of the input data set. Taking an advantage of the GAN's characteristics, Douzas and Bacao proposed a GAN-based oversampling method [16]. They adopted the conditional GAN (cGAN), a variant of the GAN, as an oversampling method and demonstrated that it shows better prediction performance than the other oversampling methods such as the synthetic minority oversampling technique (SMOTE). However, the cGAN is not appropriate for the AC and the OCC because it requires at least two classes.

Hence, we propose the least square generative adversarial network (LS-GAN)-based anomaly detection (AD) approach to handle data consisting only of positive classes. In other words, our proposed approach is based on the unsupervised learning method that the target class does not exist while the cGAN is based on the supervised learning method. The proposed method leverages the LS-GAN [17], a variant of the GAN, to improve the prediction performance of the AC and the OCC. According to the previous study [17], the LS-GAN can generate samples more similar to real samples than the regular GAN [13]. It also has improved learning stability over the regular GAN [17]. Fig. 1 shows an overview of the LS-GAN-based AD. The proposed method consists of three steps. The first step is the training of the LS-GAN. The goal in this step is to estimate the distribution of data within the input space. The second step is data augmentation to establish control boundaries. Additional samples generated in previous steps may contribute to improved anomaly detection performance. In this step, we use the LS-GAN trained in the previous step to generate new samples that do not exist in the training data. The third step is the construction of control boundaries. The control boundary is the boundary that separates normal from abnormal. Referring to the second row of Fig. 1, test samples located outside the control boundary are considered anomaly samples. The proposed method is the combination of off-line and on-line. It can be seen as a two-stage process phase I and phase II. The goal in the phase I is to establish a control boundary. Therefore, the proposed method is offline in phase I when the control boundary is not established. The goal of phase II is to monitor the on-line data and quickly detect anomalies in the process from the control boundary established in phase I. At this stage, the proposed method is the on-line. The AC control boundary looks better than the OCC control boundary because the AC control boundary can capture the quadratic pattern, while the OCC control boundary cannot. The reason why the AC control boundary outperforms the OCC control boundary is that the AC considers the process data as well as the artificial data. The definition of the AD encompasses the MSPC, the binary classification, and outlier detection. But, in this study we only consider the MSPC with the simulation data set, and the binary classification with the real data set. Both simulation and real data sets are considered for the performance comparison. For the simulation data sets, we give three cases of the shift (small, medium, and large) to the three different directions (x-axis, y-axis, and both of them). To compare their performances with the proposed method, the kernel density estimation (KDE) and the Gaussian mixture model (GMM) are also applied as the generative models. With the three generative models above (the GMM, the KDE, and the LS-GAN), the two, five, and ten times enlarged samples from the original samples are generated. Then, the OCC and the AC are used for detecting anomalies with the generated samples. As a result, we found that the AC with the LS-GAN demonstrates the best performances in terms of the AUC score.

The contribution of this study is as follows. First of all, we improve the prediction performance of the AC as well as the OCC using the LS-GAN in order to tackle the lack of training samples. In our experiment, we realize that the AD approaches with the LS-GAN are better than those with the other generative models. Second, we construct the LS-GAN model to mimic the real training data by setting appropriate parameters. Lastly, we find that the performance of the AC is better than those of the OCC in most of the results. Also, the experimental result shows that the AC with the LS-GAN which is trained with the largest training samples has the highest average ranking.

This paper is constructed with 7 sections. From Section II, we explain the details of the AC and the OCC. The existing generative models are illustrated in Section III. In Section IV, the proposed LS-GAN-based AD is explained. Section V elaborates the experimental settings with the 9 simulation and 5 real data sets. And the comparison results are specifically described in Section VI. We finalize this paper with the conclusion in Section VII.

II. THE AC AND THE OCC

A. THE AC

Proposed by Hwang *et al.* [3], the AC is one of the non-parametric monitoring approaches for the MSPC problem. Given that the MSPC is designed for simultaneously monitoring several process variables, the AC learns a classifier for defining a control boundary by generating the artificial data. The following is the overall procedure of how the AC works.

The first step is generating the artificial data. When the number of in-control process samples is n, the samples are denoted as $\{\mathbf{x}_1, \ \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. With the n number of samples and m number of process variables, $n \times m$ matrix is represented as $\mathbf{X} = \left(\mathbf{X}_1, \ldots, \mathbf{X}_j, \ldots, \mathbf{X}_m\right) = \left(x_{ij}\right), (i = 1, \ldots, n, j = 1, \ldots, m)$ where each process variable $\mathbf{X}_j$ is a column vector $\left(x_{1j}, \cdots, x_{nj}\right)^T$. In addition, a vector of the responses $\mathbf{Y}$ is represented as $\mathbf{Y} = (y_1, \cdots, y_n)^T$. Then, for each process variable $\mathbf{X}_j$, the artificial data set is generated with the sample size t. From a uniform distribution, the artificial data set is created in the range of subtracting the sample standard deviation of $\mathbf{X}_j$ from the minimum value of $\mathbf{X}_j$ to adding the sample standard deviation of $\mathbf{X}_j$ and the maximum value of $\mathbf{X}_j$. When the $\mathbf{x}_i$ is a sample from the process data, y_i is equal to 1, while y_i is 0 when the $\mathbf{x}_k$ is a sample from the artificial data ($k = 1, \ldots, t$). Finally, the training data can be obtained by combining the process data and the artificial data as illustrated in Fig.1 Step 3 (a).

After generating the artificial data, a classifier should be determined. According to Hwang *et al.* [3], two specific classifiers including the random forest (RF) and the regularized least square classifier are introduced. As a classifier V: $\{\mathbf{x}\} \ \rightarrow\in \ \{0, 1\}$ plays a role of a control boundary, the MSPC problem is converted to the inseparable binary classification problem. Here, the Type-I error is defined as the probability saying that the process is out of control even though the process is in control. By adjusting the cut-off value, the Type-I error can be manipulated.

Since the RF has been used to deal with the binary classification problem in order to determine the in-control boundaries [3]–[5], we follow the same approach. The RF represents an ensemble of individual decision trees based on a bootstrap technique [18]. The individual decision tree is a tree-like model that classifies samples. However, the problem of overfitting is a major limitation in the individual decision trees. In order to eradicate the problem, an ensemble of individual decision trees is suggested. The RF involves three steps as follows. 1) When n number of samples are taken from the training data set, an individual decision tree is constructed for the n samples. 2) As illustrated in the Fig. 2, each decision tree generates an output. 3) The final output is determined by the majority voting. In other words, by aggregating the votes from the individual decision trees, the final class of the test object is decided.

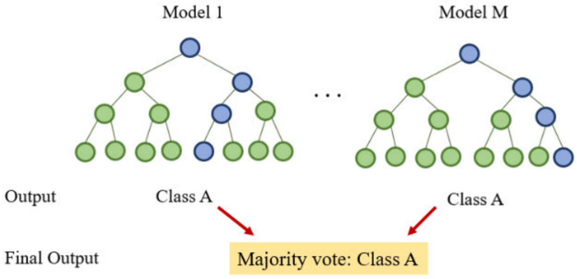


FIGURE 2. The RF approach.

B. THE OCC

The major idea of the OCC is to construct a control boundary that surrounds normal samples, assuming the samples outside of the control boundary as abnormal samples. The detail concept of this methodology can be easily grasped by comparing it with a multi-class classification and a binary class classification [19]. While the multi-class classification contains training data from several classes, the two classes (a positive class and a negative class) are included in the binary class classification. Unlike the multi-class classification and the binary classification, the OCC has only one training samples from the positive class. Therefore, the control boundary by the learned classifier in the OCC has the shape of encircling the positive samples [19]. Fig. 3 demystifies the differences among the three types of classification.

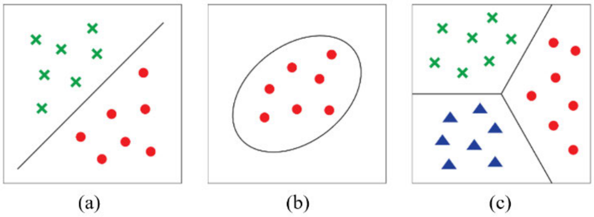


FIGURE 3. (a) Indicates the binary classification which has been generally used. The OCC with only one positive class is described in (b). The multi-class classification is illustrated in (c).

As shown in Fig. 3, the OCC is described as classifying properly-characterized positive samples with no negative samples [20]. Since the negative class of the binary classification is difficult to obtain in the real situations, the OCC has attracted many attentions for several decades. A number of the classifiers for the OCC for the SPC have been proposed [21], [22]. For example, a novel control chart for the OCC based on a k-nearest neighbor algorithm was introduced [6]. Another OCC-based control chart using the k-means data description (KMDD) algorithm, which is called the KM-chart, was proposed by Gani and Limam [23]. Moreover, the OC-SVM that adapts the SVM to the OCC was suggested [10].

The OC-SVM is to solve the OCC [7]. The OC-SVM aims to make positively-labeled samples classified by the hyperplane furthest away from the origin after mapping the original data to the feature space. For a better understanding, the OC-SVM method is illustrated in Fig. 4 [24], [25].

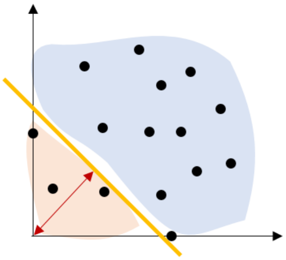


FIGURE 4. The OC-SVM approach.

As shown in the Fig. 4, the hyper-plane separates the original data set from the origin. The upper part of the hyper-plane is classified as the normal data and the lower part of the hyperplane is classified as the abnormal data. Finding the optimal hyper-plane that separates the normal data from the origin is required [24]. The optimization objective function can be defined as:

$$\begin{aligned} &\min_{\mathbf{w}} \ \frac{1}{2}\|\mathbf{w}\|^2 + \frac{1}{\nu n}\sum\nolimits_{i=1}^{n}(\xi_i - \rho) \\ &\text{s.t. } \ \mathbf{w}\cdot\Phi(\mathbf{x}_i) \geq \rho - \xi_i, \quad (\xi_i \geq 0), \end{aligned} \tag{1}$$

where the first term is for the regularization to decrease the variability. ρ is the distance between the origin and the hyperplane, ξ_i is the slack variable that is penalized in the objective function when the i^{th} training sample is located inside,n is the number of training samples, $(i = 1, 2, \ldots, n)$, and ν is a trade-off parameter ranging from 0 to 1 determining the proportion of a penalty. So, the second term is the summation of penalties given to the normal data which are located close to the origin than the distance ρ. Φ is a mapping function that maps original data $\mathbf{x}_i$ to a kernel space using a kernel function $K(\cdot, \cdot)$. The proper choice of a kernel function is dependent on the number of features (e.g. linear, sigmoid, polynomial and radial basis kernels) [24]. When the hyperplane is determined after the optimization problem is solved, the data can be classified as the normal data when it is above the hyper-plane, and considered as the abnormal data when it is below the hyper-plane using the condition sign $(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - \rho)$.

With the Lagrange multipliers $\alpha_i, \beta_i \geq 0$, the Equation (1) can be modified as:

$$\begin{aligned} L = \frac{1}{2}\|\mathbf{w}\|^2 &+ \frac{1}{\nu n}\sum\nolimits_{i=1}^{n}(\xi_i - \rho) \\ &- \sum\nolimits_{i=1}^{n}\alpha_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i) - \rho + \xi_i) \\ &- \sum\nolimits_{i=1}^{n}\beta_i\xi_i, \end{aligned} \tag{2}$$

where the column vectors $\boldsymbol{\alpha} = [\alpha_i, \alpha_{2,}, \ldots, \alpha_n]^T$ and $\boldsymbol{\beta} = [\beta_i, \beta_{2,}, \ldots, \beta_n]^T$. Since the derivatives of primal variables are equal to 0, the following formulas are valid ($\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i), \alpha_i = 1/\nu n - \beta_i, \sum_{i=1}^{n} \alpha_i = 1$). Then, with the formula $\mathbf{w} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i)$, the primal Lagrange problem can be converted to the dual optimization problem as follows.

$$\begin{aligned} L = {} & \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \Phi(\mathbf{x}_i)\, \Phi\left(\mathbf{x}_j\right) + \frac{1}{\nu n} \sum_{i=1}^{n} (\xi_i - \rho) \\ & - \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \Phi(\mathbf{x}_i)\, \Phi\left(\mathbf{x}_j\right) + \rho \sum_{i=1}^{n} \alpha_i \\ & - \sum_{i=1}^{n} \alpha_i \xi_i - \sum_{i=1}^{n} \beta_i \xi_i, \end{aligned} \tag{3}$$

The Equation (3) can be simplified to:

$$L = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j \Phi(\mathbf{x}_i)\, \Phi\left(\mathbf{x}_j\right). \tag{4}$$

Then, the maximization problem can be altered to the minimization problem by switching the sign. Using the kernel function $K(\cdot, \cdot)$, the mapping function Φ can be substituted with $K(\cdot, \cdot)$.

$$\begin{aligned} & \min\ L = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right), \\ & \text{s.t. } \sum_{i=1}^{n} \alpha_i = 1, \quad 0 \leq \alpha_i \leq \frac{1}{\nu n} \end{aligned} \tag{5}$$

III. THE EXISTING GENERATIVE MODELS

In general, increasing the number of training samples is one of the primary methods for enhancing the performance of the classification models. Not to mention the GAN, the KDE and the GMM that estimate a probability density function using the observed data are also used for the data augmentation.

A. THE KDE

The KDE is a non-parametric technique for estimating the probability density function of a random variable with a kernel function [26]. In order to alleviate the disadvantage of a traditional histogram (e.g., discontinuity between each bin of a histogram, and fluctuation by the size of bins, etc.), a kernel function is adopted and is able to produce the smooth estimate of the probability density function. The formula for the KDE method is as follows.

$$P_h(\mathbf{x}) = \frac{1}{n} \sum_{i=1}^{n} K_h(\mathbf{x} - \mathbf{x}_i) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right), \tag{6}$$

where $K_h(\mathbf{x}) = \frac{1}{h^d} K\left(\frac{\mathbf{x}}{h}\right)$, n is the number of training samples and K is a kernel function that is generally symmetric function such as a Gaussian [26]. And we define K_h as a kernel function with the size transformation since the result is dependent on the bandwidth (h). d indicates the number of dimensions of feature vectors. Since the scarcity of data in the high-dimensional feature space is the main challenge of the KDE, it is generally used with the low-dimensional data.

B. THE GMM

While the KDE is a non-parametric technique, the GMM is a parametric method, where the data are assumed to come from prescribed models that are determined by parameters. As one of the probabilistic models, the GMM estimates a parametric probability density function, where samples are generated from a mixture of the multiple Gaussian distribution. The GMM is able to represent the feature of distribution precisely that is not capable of with only one normal distribution function. The GMM can be defined as $\mathrm{p}(\mathbf{x} \mid \lambda) = \sum_{i=1}^{n} \omega_i g(\mathbf{x} \mid \mu_i, \Sigma_i)$, where $\mathbf{x}$ is a m-dimensional continuous-valued data vector, $\omega_i (i = 1, \ldots, n)$ is the mixture weights, and $g(\mathbf{x} \mid \mu_i, \Sigma_i)$, $i = 1, \ldots, n$, is the component Gaussian densities [27]. Since the GMM is a mixture of the multiple Gaussian distribution, a parametric estimation problem can occur by calculating the various mixture components including weight, mean, and variance. In general, the maximum likelihood estimation (MLE) is used for the parametric estimation. However, in the mixture model, the expectation maximization (EM) algorithm is the key to solve the parametric estimation problem [28]. Mainly used in incomplete data or data with missing values, the EM algorithm provides the MLE of the parameters of an underlying distribution from the given data set.

C. THE GAN AND LS-GAN

The GAN is a generative model that generates samples which are close to the real samples using the two adversarial networks. One is a generator which generates fake samples and the other is a discriminator which distinguishes the original samples from the generated samples. An adversarial training improves the generator over time, until the discriminator can no longer distinguish between the real and the fake. The GAN shows a powerful ability to learn the high-dimensional and complex distribution of data without any parametric assumptions.

Formally, the generator randomly takes the noise samples from a noise distribution, such as Gaussian and uniform distribution. It maps them to a data space same as input real data $G(\mathbf{z}) : \mathbb{R}^d \to \mathbb{R}^m$. The discriminator $D(\mathbf{x})$ calculates a probability that $\mathbf{x}$ is a real sample, not a generated sample from the generator $D(\mathbf{x}) : \mathbb{R}^m \to [0, 1]$. The GAN is based on a non-cooperative game and training the GAN means optimizing the following minmax objective function.

$$\begin{aligned} \min_{G} \max_{D} V(D, G) = {} & \mathbb{E}_{\mathbf{x} \sim p_{data}(\mathbf{x})} \left[\log D(\mathbf{x})\right] \\ & + \mathbb{E}_{\mathbf{x} \sim p_{\mathbf{x}}(\mathbf{z})} \left[\log\left(1 - D(G(\mathbf{z}))\right)\right]. \end{aligned} \tag{7}$$

If the generator learns the distribution of data successfully, the generated samples become so close that they are indistinguishable from the real data. Also, the discriminator emits 0.5 everywhere [13].

Although the GAN can learn the distribution followed by the real data without any assumptions, it has some unsolved problems, such as the gradient vanishing and the gradient explosion. When it comes to tackling the problems, the

LS-GAN that uses the least squares loss function rather than the sigmoid cross-entropy for the discriminator has been proposed [17]. The objective function of the LS-GAN is expressed as:

$$\begin{cases} \min_{w_D} \frac{1}{2}\mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[(D(\mathbf{x})-1)^2\right] \\ \qquad + \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[(D(G(\mathbf{z})))^2\right], \\ \min_{w_G} \frac{1}{2}\mathbb{E}_{\mathbf{z}\sim p(\mathbf{z})}\left[(D(G(\mathbf{z}))-1)^2\right] \end{cases} \tag{8}$$

where w_G and w_D are the parameters of the generator and the discriminator, respectively.

IV. THE LS-GAN APPLICATION TO ANOMALY DETECTION

The aim of this paper is to improve the prediction performance with the combination of the LS-GAN and the AC and the OCC. The proposed LS-GAN-based AD proceeds as follows. 1) The first step is the training of the LS-GAN to learn the distribution of the training data set. 2) Generating new samples using the trained LS-GAN is the second step. 3) The AC and the OCC are trained with both existing and generated samples. Fig. 1 depicts the procedure of the proposed method. We conduct an experiment of the comparison between the regular GAN and the LS-GAN in order to verify their performances. We consider the number of outliers a comparison measure because even the very small number of outliers can distort the control boundary. For the performance comparison between the regular GAN and the LS-GAN, we use the banana-shaped simulation data set that follows the non-normal distribution [3]. Fig. 5 shows the performance results of the regular GAN and the LS-GAN. As shown in Fig. 5 (a), the extreme outliers are marked in the dashed box.

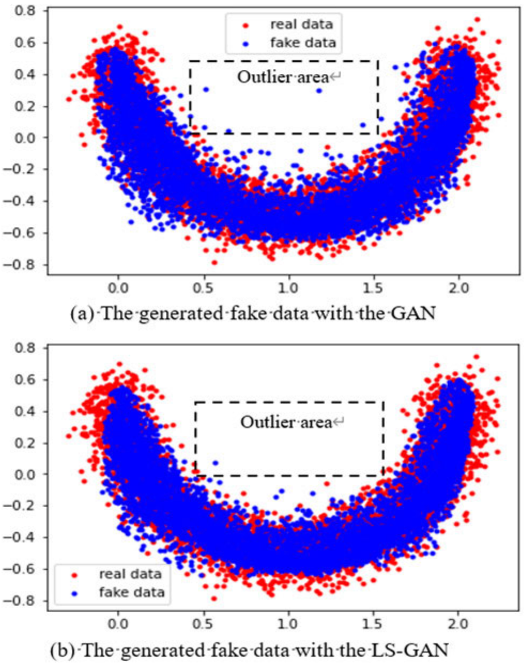
Therefore, in this study, since the LS-GAN generates the extreme outliers less than the regular GAN, we select the LS-GAN as a generative model instead of the regular GAN for the stable training. The hyper-parameters of the LS-GAN are determined by leveraging on Douzas and Bacao's approaches [16] as shown in Table 1. Both the generator and the discriminator adopt the rectified linear units as the activation function for the five hidden layers. The last activation of the generator is the hyperbolic tangent function and the last activation of the discriminator is the sigmoid function. Each model is trained by the Adam optimizer with the different learning rates. The learning rates of the generators and the discriminators are 0.0001 and 0.001, respectively. The training epoch is set to 10,000 and the mini-batch size is 20.

V. EXPERIMENTAL SETTINGS

A. DATA SETS

In this paper, we consider both simulation and real data sets for evaluating the effectiveness of the proposed approach. To generate the simulated data, we consider a bivariate banana-shaped distribution. For testing, normal samples generated under the same conditions as the training data set are combined with the abnormal samples. Here, the samples with no shift, $\delta_0 = 0$ are considered as normal samples. On the other hand, we consider not only a shift size but also the shift types for the abnormal samples. The shift sizes are small ($\delta_1 = 0.1$), medium ($\delta_2 = 0.3$), and large ($\delta_3 = 0.5$). And the three types of a shift are considered: x-only, y-only, and both x and y. So, 9 simulation data sets (Simulated 1 ∼ Simulated 9) are represented in Table 2 and shown in Fig. 6.

Additionally, in order to recheck the effectiveness of the proposed method, the five real data sets are borrowed from the UCI Machine Learning Repository. The AC and the OCC



(a) The generated fake data with the GAN

(b) The generated fake data with the LS-GAN

FIGURE 5. The visualization of generated fake data with the regular GAN and the LS-GAN respectively.

TABLE 1. The hyper-parameter settings of the LS-GAN.

Data set	The number of layers	The number of hidden nodes	
		Generator	Discriminator
Pima	5	32	32
Breast cancer		90	45
Heart diseases		80	40
Wine		16	16
Haberman		4	8
Simulated		16	8

TABLE 2. The description of 9 simulation data sets.

Shift size	Shift direction		
	x-only	y-only	both x and y
Small	Simulated 1	Simulated 2	Simulated 3
Medium	Simulated 4	Simulated 5	Simulated 6
Large	Simulated 7	Simulated 8	Simulated 9

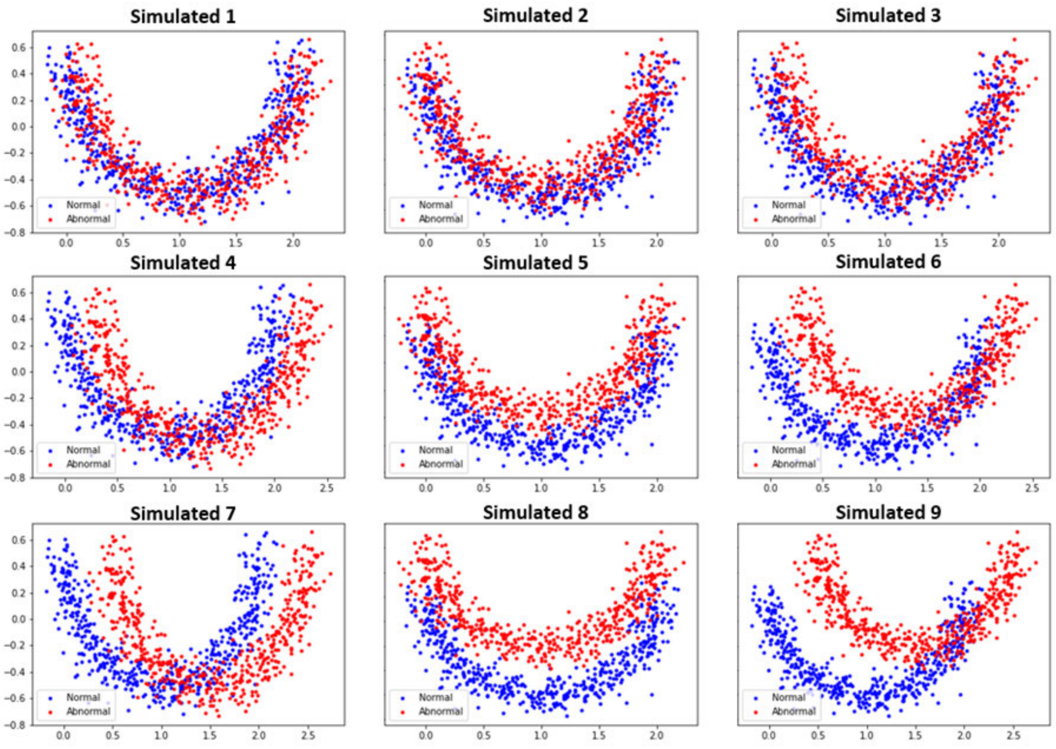


FIGURE 6. The visualized 9 simulation data sets.

TABLE 3. The description of the real and simulation data sets.

Data set	The number of variables	First *d* principal components	The number of samples	The number of abnormal samples
Pima	9	6	768	268
Breast cancer	30	8	569	212
Heart diseases	14	10	303	138
Wine	13	8	178	48
Haberman	4	3	306	81
Simulated	2	-	500	500

may not be appropriate for the binary classification problems of the five data sets because they are the alternatives of the Hotelling's control boundary. Only the Wine data set consists of three classes, and the rest of the real data sets consist of binary classes. Samples belonging to the majority class are considered as the normal samples in this paper. For the Wine data set, we consider the two major classes as the normal and the remaining minor class as the abnormal. 80% of the randomly chosen normal samples are reserved for training while the remaining 20% of the normal samples are combined with the abnormal samples for the testing. Finally, the principal component analysis (PCA) is applied to all the real data sets for the dimensionality reduction. In this paper, the first *d* principal components taking more than 90% of the total variance are used. Besides, all of data sets are normalized between −1 and 1. Table 3 represents the description of the data sets.

TABLE 4. The experimental settings for the AC.

The number of process data	Generated samples by the generative models	The aggregate amount of training samples	The number of artificial data
500	1000	1500	3000
	2500	3000	6000
	5000	5500	11000

B. THE SETTINGS FOR THE GENERATIVE MODELS

The generative models (the KDE, the GMM, and the LS-GAN) are used to increase the amount of training samples. The number of the training samples of each data set is respectively increased by 2, 5, or 10 times of the original samples. So, with the three different generative models, a total of 9 increased training data sets are generated for the one original data set. The illustrative examples of each original data set and the two times-increased fake data set made by the generative models are shown in the appendix Figure 8. The appendix Figure 8 shows that the proposed method can

TABLE 5. The average ranking according to the AD approaches with each generative model.

Data set	AD approaches	Baseline	KDE			GMM			LS-GAN		
			X2	X5	X10	X2	X5	X10	X2	X5	X10
Average ranking	OCC	15.8	12.9	12.5	14.5	17.4	15.1	14.5	12.0	11.5	11.2
	AC	7.6	10.6	7.8	10.1	8.4	6.1	5.4	6.4	5.9	**4.2**

generate new samples that are not observed in the training data set. Gray dots denote observed samples in the training data, and colored dots denote generated samples by the proposed model. For the real data sets the first principal component is on the x-axis and the second principal component is on the y-axis.

C. THE SETTINGS FOR THE CLASSIFIERS

The appropriate settings for the two classifiers are also required. For the RF, the number of artificial data is twice of the training samples for each case as shown in the Table 4. Plus, so as to enhance the performance of the OC-SVM, the hyper-parameter ν is set to 0.05.

We use the area under the curve (AUC) as the performance measures [29]. The AUC is the area under the receiver operating characteristic (ROC) curve generated by plotting the false positive rate (FPR) against true positive rate (TPR) at various threshold values. The definitions of the FPR and TPR are presented in Equation (9) and (10).

$$FPR = \frac{FP}{FP + TN} \tag{9}$$

$$TPR = \frac{TP}{TP + FN}, \tag{10}$$

where FP (false positive) is the number of normal samples which is falsely predicted, TP (true positive) is the number of abnormal samples which is correctly assigned, TN (true negative) is the number of normal samples which is correctly predicted, and FN is the number of normal samples which is falsely predicted.

VI. EXPERIMENTAL RESULTS

As mentioned in the previous section, we increase the training samples using the generative models. Each generative model increases the number of the training samples to the multiples of the number of the original training data by 2, 5, and 10 respectively. As a result, the AUC scores for the 20 classification models for each data set are calculated. The detail results of the experiments are summarized in the appendix TABLE 6. The AUC rankings for each data set are in parentheses, and the boldface indicates the best-performed model for each corresponding data set.

Table 5 shows the average rankings according to the AD approaches with each generative model based one the appendix Table 6. The average ranking values close to 1 indicate the best overall performance, while values close to 20 indicate the worst overall performance.

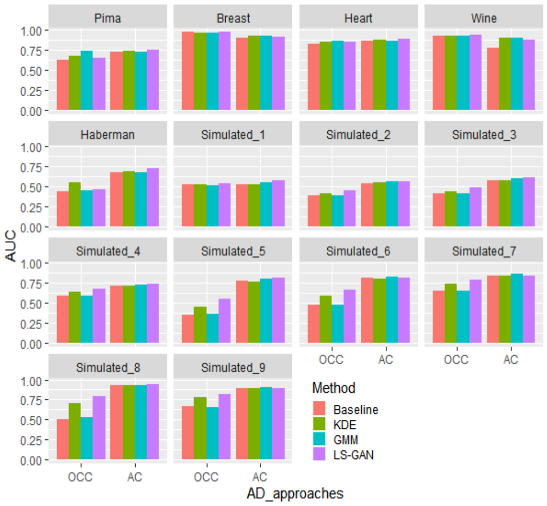


FIGURE 7. The comparisons of the three generative models with their best performances respectively.

The experimental results demonstrate that the average rankings of the LS-GAN as a generative model are higher than those of the baselines regardless of the AD approaches. Especially, the AC performs the best when the LS-GAN increases the number of training samples to ten times. In the perspective of the average rankings, the AC performs better than the OCC for every result of generative models' performances. Regardless the AD approaches, the LS-GAN ranks the first place in all the real data sets.

Only the best performances of the three generative models are compared for the respective real and the simulation data sets in Fig. 7. As shown in the Fig. 7, the LS-GAN-based AD performs the best with the majority of the simulation data sets.

VII. CONCLUSION

In this study, we propose the LS-GAN-based AD to increase the prediction performance of the AC as well as the OCC under the circumstances of the limited training samples. The AD procedures are as follows. The first step is the training of the LS-GAN to learn the distribution of the training data set. Second, in order to increase the number of the training samples, the LS-GAN generates new training samples and then combines the generated samples with the original samples.

Third, the new training data sets are used to detect anomalies with the AC and the OCC. Finally, we evaluate the prediction

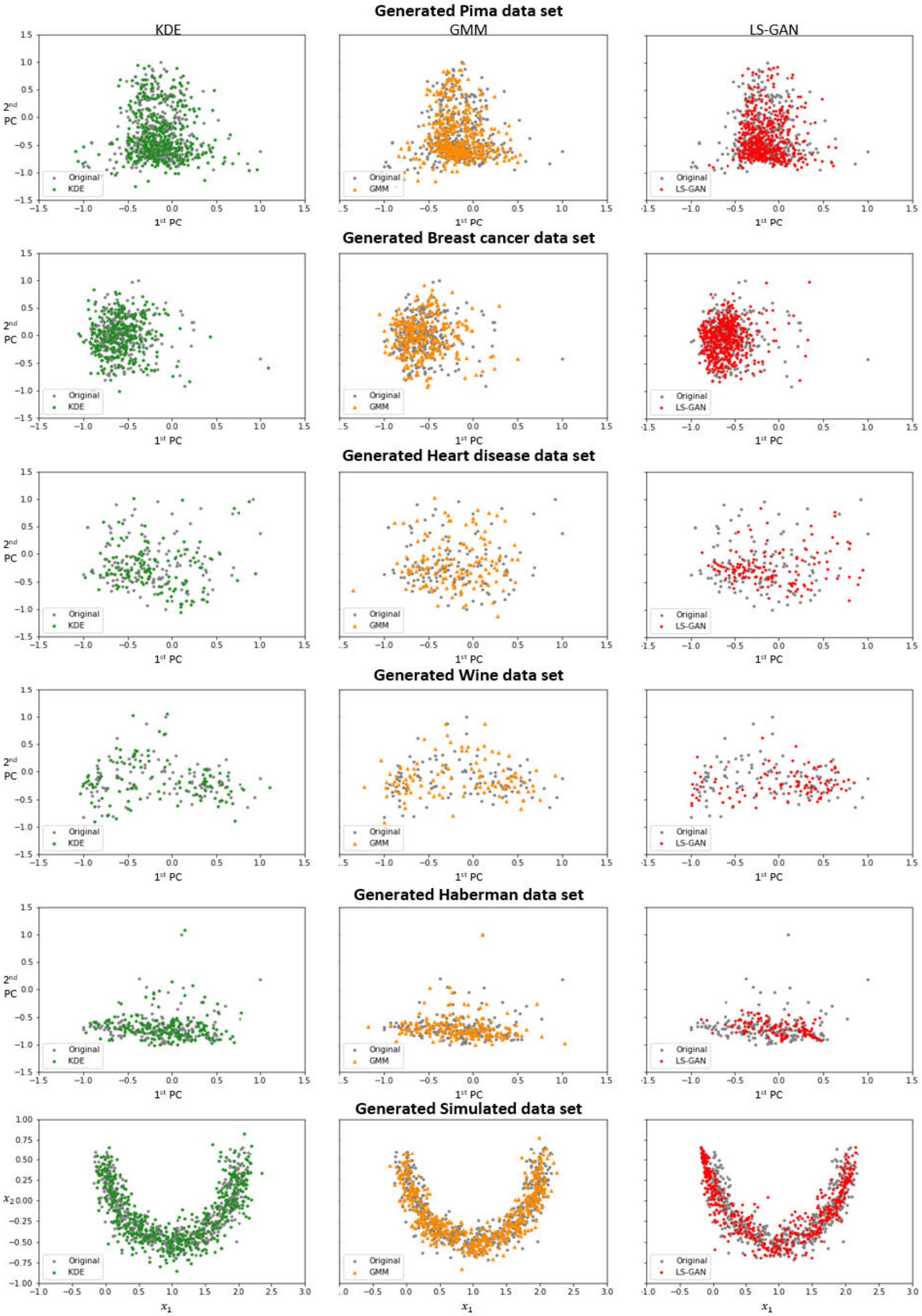


FIGURE 8. Each of original data set and the synthetic data set generated by the generative models.

TABLE 6. The prediction performances according to the data sets.

Data set	AD approaches	Baseline	KDE			GMM			LS-GAN		
			X2	X5	X10	X2	X5	X10	X2	X5	X10
Pima	OCC	0.6229 (18)	0.5956 (19)	0.6746 (14)	0.6491 (15)	0.6978 (13)	0.7192 (9)	0.7330 (4)	0.5921 (20)	0.6475 (16)	0.6471 (17)
	AC	0.7281 (7)	0.7169 (12)	0.7257 (8)	0.7329 (5)	0.7315 (6)	0.7186 (10)	0.7173 (11)	0.7365 (3)	0.7440 (2)	**0.7490** **(1)**
Breast cancer	OCC	0.9711 (2)	0.9700 (3)	0.9695 (4)	0.9692 (6)	0.9674 (8)	0.9639 (10)	0.9693 (5)	**0.9724** **(1)**	0.9686 (7)	0.9655 (9)
	AC	0.8977 (20)	0.9251 (12)	0.9133 (16)	0.9220 (14)	0.9246 (13)	0.9165 (15)	0.9328 (11)	0.9081 (18)	0.9002 (19)	0.9090 (17)
Heart disease	OCC	0.8270 (18)	0.8483 (10)	0.8382 (14)	0.8360 (16)	0.8344 (17)	0.8643 (6)	0.8592 (8)	0.8459 (12)	0.8408 (13)	0.8248 (19)
	AC	0.8646 (5)	0.8460 (11)	0.8744 (2)	0.8200 (20)	0.8364 (15)	0.8691 (3)	0.8524 (9)	0.8625 (7)	**0.8879** **(1)**	0.8690 (4)
Wine	OCC	0.9223 (6)	0.9207 (7)	0.9287 (4)	0.9255 (5)	0.9175 (10)	0.9303 (3)	0.9183 (9)	0.9199 (8)	0.9359 (2)	**0.9423** **(1)**
	AC	0.7720 (19)	0.8281 (18)	0.8994 (12)	0.8938 (13)	0.8674 (16)	0.9010 (11)	0.8906 (14)	0.7720 (19)	0.8534 (17)	0.8726 (15)
Haberman	OCC	0.4405 (17)	0.5562 (11)	0.5406 (12)	0.5167 (13)	0.4125 (16)	0.4536 (11)	0.4089 (14)	0.4229 (19)	0.4536 (17)	0.4613 (15)
	AC	0.6719 (6)	0.6335 (9)	0.6909 (3)	0.6412 (8)	0.6291 (10)	0.6772 (5)	0.6782 (4)	0.6471 (7)	**0.7226** **(1)**	0.7193 (2)
Simulated 1	OCC	0.5200 (16)	0.5204 (14)	0.5235 (12)	0.5150 (20)	0.5166 (17)	0.5162 (19)	0.5166 (18)	0.5335 (6)	0.5237 (11)	0.5297 (9)
	AC	0.5315 (7)	0.5201 (8)	0.5222 (5)	0.5309 (9)	0.5241 (4)	0.5450 (10)	0.5443 (3)	0.5658 (6)	0.5416 (2)	**0.5710** **(1)**
Simulated 2	OCC	0.3909 (19)	0.4092 (15)	0.4176 (14)	0.4062 (16)	0.3897 (20)	0.3918 (18)	0.3920 (17)	0.4289 (13)	0.4465 (12)	0.4533 (11)
	AC	0.5397 (7)	0.5378 (8)	0.5566 (5)	0.5304 (9)	0.5569 (4)	0.5291 (10)	0.5613 (3)	0.5556 (6)	0.5633 (2)	**0.5667** **(1)**
Simulated 3	OCC	0.4178 (17)	0.4375 (15)	0.4401 (14)	0.4253 (16)	0.4076 (20)	0.4136 (19)	0.4136 (18)	0.4648 (13)	0.4685 (12)	0.4822 (11)
	AC	0.5795 (5)	0.5786 (6)	0.5587 (10)	0.5595 (9)	0.5762 (8)	0.5771 (7)	0.5999 (4)	0.6042 (3)	0.6073 (2)	**0.6202** **(1)**
Simulated 4	OCC	0.5928 (17)	0.6276 (15)	0.6411 (14)	0.6070 (16)	0.5880 (19)	0.5854 (20)	0.5911 (18)	0.6618 (12)	0.6554 (13)	0.6729 (11)
	AC	0.7199 (7)	0.7081 (9)	0.7105 (8)	0.7043 (10)	0.7331 (3)	0.7203 (6)	0.7330 (4)	0.7402 (2)	0.7278 (5)	**0.7455** **(1)**
Simulated 5	OCC	0.3483 (19)	0.4462 (15)	0.4527 (14)	0.4136 (16)	0.3441 (20)	0.3562 (18)	0.3587 (17)	0.4884 (13)	0.5330 (12)	0.5519 (11)
	AC	0.7758 (7)	0.7411 (10)	0.7689 (8)	0.7431 (9)	0.8039 (5)	0.7897 (6)	0.8084 (3)	**0.8165** **(1)**	0.8078 (4)	0.8100 (2)
Simulated 6	OCC	0.4833 (17)	0.5880 (14)	0.5715 (15)	0.5371 (16)	0.4618 (20)	0.4748 (19)	0.4811 (18)	0.6284 (13)	0.6387 (12)	0.6642 (11)
	AC	0.8153 (4)	0.7813 (10)	0.8014 (8)	0.7816 (9)	0.8024 (7)	0.8200 (2)	**0.8241** **(1)**	0.8036 (6)	0.8082 (5)	0.8164 (3)
Simulated 7	OCC	0.6557 (18)	0.7351 (15)	0.7410 (14)	0.7013 (16)	0.6477 (20)	0.6478 (19)	0.6588 (17)	0.7610 (13)	0.7643 (12)	0.7866 (11)
	AC	0.8435 (3)	0.8277 (9)	0.8424 (4)	0.8262 (10)	0.8361 (7)	0.8440 (2)	**0.8627** **(1)**	0.8373 (6)	0.8355 (8)	0.8407 (5)
Simulated 8	OCC	0.4980 (19)	0.7097 (14)	0.7047 (15)	0.6469 (16)	0.4894 (20)	0.5074 (18)	0.5291 (17)	0.7404 (13)	0.7787 (12)	0.7957 (11)
	AC	0.9274 (7)	0.9017 (10)	0.9249 (8)	0.9157 (9)	0.9280 (6)	0.9363 (3)	0.9283 (5)	0.9357 (4)	0.9368 (2)	**0.9415** **(1)**
Simulated 9	OCC	0.6676 (18)	0.7757 (14)	0.7510 (15)	0.7410 (16)	0.6427 (20)	0.6573 (19)	0.6735 (17)	0.8028 (13)	0.8093 (12)	0.8241 (11)
	AC	0.8955 (2)	0.8703 (10)	0.8927 (4)	0.8813 (8)	0.8840 (7)	0.8929 (3)	**0.9040** **(1)**	0.8877 (6)	0.8806 (9)	0.8891 (5)

performance of the proposed method using 5 real and 9 simulated data sets. Although the LS-GAN-based AC as well as the LS-GAN-based OCC shows the most successful performance, training the LS-GAN requires more effort and time compared to the other existing generative models. Also, whether the proposed method is applicable to time series data remains unresolved. Therefore, a combination of generative and anomaly detection models for time series data [30], [31] can be considered in future studies. In additions, systematic methods such as cross-validation using evaluation measures to train LS-GANs in the context of AC can be considered.

APPENDIX

See Fig. 8 and Table 6.

REFERENCES

[1] H. Hotelling, "Multivariate quality control-illustrated by air testing of sample bombsights," in *Techniques of Statistical Analysis*, C. Eisenhart, M. W. Hastay, and W. A. Wallis, Eds. New York, NY, USA: McGraw-Hill, 1947, pp. 111–184.

[2] C. A. Lowry and D. C. Montgomery, "A review of multivariate control chart," *IIE Trans.*, vol. 27, no. 6, pp. 800–810, Dec. 1995.

[3] 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.

[4] W.-Y. Hwang and J.-S. Lee, "Shifting artificial data to detect system failures," *Int. Trans. Oper. Res.*, vol. 22, no. 2, pp. 363–378, Mar. 2015.

[5] W.-Y. Hwang, "Cluster-based artificial contrasts for inhomogeneously distributed data with an indicator variable," *Int. J. Prod. Res.*, vol. 54, no. 17, pp. 5045–5055, Sep. 2016.

[6] T. Sukchotrat, S. B. Kim, and F. Tsung, "One-class classification-based control charts for multivariate process monitoring," *IIE Trans.*, vol. 42, no. 2, pp. 107–120, Nov. 2009.

[7] M. M. Moya and D. R. Hush, "Network constraints and multi-objective optimization for one-class classification," *Neural Netw.*, vol. 9, no. 3, pp. 463–474, Apr. 1996.

[8] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995, doi: 10.1007/BF00994018.

[9] D. M. J. Tax and R. P. W. Duin, "Support vector data description," *Mach. Learn.*, vol. 54, no. 1, pp. 45–66, Jan. 2004.

[10] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2014.

[11] P. Oza and V. M. Patel, "One-class convolutional neural network," *IEEE Signal Process. Lett.*, vol. 26, no. 2, pp. 277–281, Feb. 2019, doi: 10.1109/LSP.2018.2889273.

[12] L. Ruff, R. Vandermeulen, N. Goernitz, L. Deecke, S. A. Siddiqui, A. Binder, E. Müller, and M. Kloft, "Deep one-class classification," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 4393–4402.

[13] 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.

[14] C.-K. Lee, Y.-J. Cheon, and W.-Y. Hwang, "Studies on the GAN-based anomaly detection methods for the time series data," *IEEE Access*, vol. 9, pp. 73201–73215, 2021, doi: 10.1109/ACCESS.2021.3078553.

[15] D. Li, D. Chen, L. Shi, B. Jin, J. Goh, and S.-K. Ng, "MAD-GAN: Multivariate anomaly detection for time series data with generative adversarial networks," 2019, *arXiv:1901.04997*.

[16] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, Jan. 2018.

[17] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley, "Least squares generative adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2813–2821, doi: 10.1109/ICCV.2017.304.

[18] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[19] P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," 2021, *arXiv:2101.03064*.

[20] S. S. Khan and M. G. Madden, "A survey of recent trends in one class classification," in *Proc. Irish Conf. Artif. Intell. Cognit. Sci.* Berlin, Germany: Springer, 2009, pp. 188–197.

[21] S. Kittiwachana, D. L. S. Ferreira, G. R. Lloyd, L. A. Fido, D. R. Thompson, R. E. A. Escott, and R. G. Brereton, "One class classifiers for process monitoring illustrated by the application to online HPLC of a continuous process," *J. Chemometrics*, vol. 24, nos. 3–4, pp. 96–110, Feb. 2010.

[22] T. Sukchotrat, S. B. Kim, K.-L. Tsui, and V. C. P. Chen, "Integration of classification algorithms and control chart techniques for monitoring multivariate processes," *J. Stat. Comput. Simul.*, vol. 81, no. 12, pp. 1897–1911, Dec. 2011.

[23] W. Gani and M. Limam, "A one-class classification-based control chart using the k-means data description algorithm," *J. Quality Rel. Eng.*, vol. 2014, Jun. 2014, Art. no. 239861.

[24] L. Manevitz and M. Yousef, "One-class SVMs for document classification," *J. Mach. Learn. Res.*, vol. 2, pp. 139–154, Dec. 2002.

[25] M. Amer, M. Goldstein, and S. Abdennadher, "Enhancing one-class support vector machines for unsupervised anomaly detection," in *Proc. ACM SIGKDD Workshop Outlier Detection Description*, 2013, pp. 8–15.

[26] S. Węglarczyk, "Kernel density estimation and its application," in *Proc. ITM Web Conf.*, vol. 23, 2018, p. 00037.

[27] D. Reynolds, "Gaussian mixture models," in *Encyclopedia of Biometrics*. Boston, MA, USA: Springer, 2015, pp. 827–832.

[28] R. Singh, B. C. Pal, and R. A. Jabr, "Statistical representation of distribution system loads using Gaussian mixture model," *IEEE Trans. Power Syst.*, vol. 25, no. 1, pp. 29–37, Oct. 2010.

[29] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, pp. 1145–1159, Jul. 1997.

[30] J. Yoon, D. Jarrett, and M. V. D. Schaar, "Time-series generative adversarial networks," in *Proc. 33rd Conf. Neural Inf. Process. Syst.*, 2019, pp. 1–11.

[31] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (medical) time series generation with recurrent conditional GANs," 2017, *arXiv:1706.02633*.

CHANG-KI LEE received the Ph.D. degree in operations management from Dongguk University, in 2019. He worked as a Postdoctoral Researcher at Dong-A University Research Foundation for Industry-Academy Cooperation for one year. He joined Samsung Electronics, where he has been working with the Quality Innovation Team, Global CS Center, Suwon, South Korea. His current research areas include business analytics and quality management.

YU-JEONG CHEON received the B.A. 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, anomaly 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 from 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 areas include deep learning, anomaly detection, business analytics, and quality engineering. In addition to academic work, he worked as a Senior Researcher at the government research agencies and corporations. He received the Best Application Paper Award in quality and reliability engineering from the *IIE Transactions*, in 2008.