Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Generative adversarial network based novelty detection using minimized reconstruction error

Huan-gang WANG‡, Xin LI, Tao ZHANG

Department of Automation, School of Information Science and Technology, Tsinghua University, Beijing 100084, China

E-mail: hgwang@tsinghua.edu.cn; xin-li16@mails.tsinghua.edu.cn; taozhang@tsinghua.edu.cn

Received Nov. 24, 2017; Revision accepted Jan. 26, 2018; Crosschecked Jan. 26, 2018

Abstract: Generative adversarial network (GAN) is the most exciting machine learning breakthrough in recent years, and it trains the learning model by finding the Nash equilibrium of a two-player zero-sum game. GAN is composed of a generator and a discriminator, both trained with the adversarial learning mechanism. In this paper, we introduce and investigate the use of GAN for novelty detection. In training, GAN learns from ordinary data. Then, using previously unknown data, the generator and the discriminator with the designed decision boundaries can both be used to separate novel patterns from ordinary patterns. The proposed GAN-based novelty detection method demonstrates a competitive performance on the MNIST digit database and the Tennessee Eastman (TE) benchmark process compared with the PCA-based novelty detection methods using Hotelling's T^2 and squared prediction error statistics.

Key words: Generative adversarial network (GAN); Novelty detection; Tennessee Eastman (TE) process
https://doi.org/10.1631/FITEE.1700786 **CLC number:** TP391

1 Introduction

Novelty detection usually refers to recognizing abnormal samples in the test dataset when the training dataset contains only normal samples. Novelty detection has aroused great attention in the areas such as industry process fault detection (Ge et al., 2013), medical diagnosis (Clifton et al., 2011; Schlegl et al., 2017), drug discovery (Kadurin et al., 2017b), and fraud detection in the finance field (Patcha and Park, 2007). In these situations, normal working conditions are usually easily and cheaply observed, while abnormality is rarely observed because the abnormal states have a low frequency of occurrence or it is harmful to the system to do experiments in abnormal conditions. On the other hand, there is a significant variability of abnormal states, and the collected abnormal dataset can hardly represent all abnormal situations. These factors make conventional binary classification methods inapplicable. Novelty detection (or 'one-class classification') is a solution to this problem. In novelty detection, a model is taken according to historical data of a system in normal conditions to describe its normality, and a novelty score function is formulated to estimate the novelty of new data samples. When the novelty score of a sample is higher than a certain threshold, the sample is determined to be an abnormal sample.

In novelty detection, the training dataset $\boldsymbol{X}^{\text{train}}$ contains only normal samples, and the test dataset contains both normal and abnormal samples. A novelty detection model is trained on the training dataset. For samples $\boldsymbol{x}'$ in the test dataset, the novelty score $f(\boldsymbol{x}')$ is obtained from the trained model. The test sample is more likely to be abnormal with a higher novelty score. Pimentel et al. (2014) classified novelty detection methods into five categories:

‡ Corresponding author
ORCID: Huan-gang WANG, http://orcid.org/0000-0002-7322-3446

(1) Probabilistic methods like Gaussian mixture models (GMMs) (Yu and Qin, 2008, 2009; Yu, 2012) assume that low-density areas have a low probability of containing normal samples. Density estimation is made on the training dataset, and the estimated density is used as the novelty score. (2) Distance-based methods like the k-nearest neighbor (k-NN) (Hautamaki et al., 2004) assume that the normal samples are close to each other while the abnormal samples are far from their nearest neighbors. The distances to a sample's nearest neighbors are used to form the novelty score. (3) Reconstruction-based methods like principal component analysis (PCA) (Ge et al., 2009) and kernel PCA (Hoffmann, 2007) learn a map between the data space and the latent space, and the reconstruction error can be used as the novelty score. (4) Domain-based methods like support vector data description (SVDD) (Ge et al., 2011; Ge and Song, 2013) and one-class support vector machine (SVM) (Mahadevan and Shah, 2009) try to determine a decision boundary with normal samples inside the boundary and abnormal samples outside the boundary. (5) Information-theoretic techniques use information-theoretic measures such as entropy (He et al., 2005) or Kolmogorov (Keogh et al., 2004) complexity, assuming that the information content of the dataset is different when containing abnormal samples.

The generative adversarial network (GAN) is a new kind of generative model proposed by Goodfellow et al. (2014). Initially, GAN was used for image generation (Denton et al., 2015; Radford et al., 2015) to augment the dataset for deep learning. GAN has drawn great attention from researchers, and there have been achievements in a lot of image-related tasks such as image caption (Reed et al., 2016), image super-resolution (Ledig et al., 2016), image segmentation (Luc et al., 2016), image detection (Li J et al., 2017), image inpainting (Yeh et al., 2016; Li Y et al., 2017), and image de-occlusion (Zhao et al., 2018). Applications of the GAN model have been extended to video generation (Vondrick et al., 2016), encryption and decryption (Abadi and Andersen, 2016), 3D modeling (Wu et al., 2016), text generation (Yu et al., 2017), machine translation (Yang et al., 2017), and drug development (Kadurin et al., 2017a,b). There have also been theoretical studies on GAN like least squares GAN (Mao et al., 2016), energy-based GAN (Zhao et al., 2016), Wasserstein GAN (Arjovsky et al., 2017), and boundary equilibrium GAN (Berthelot et al., 2017). As a new kind of generative model, GAN also gains attention in dealing with classical machine learning problems such as clustering (Springenberg, 2015), unsupervised feature learning (Donahue et al., 2016; Dumoulin et al., 2016), classification (Ge et al., 2017), transfer learning (Kim et al., 2017; Yi et al., 2017; Zhu et al., 2017), ensemble learning (Grover and Ermon, 2017), and reinforcement learning (Yu et al., 2017).

GAN is motivated by the two-player zero-sum game theory. The two players are a generator G and a discriminator D. The generator tries to learn the distribution of the real dataset, and the discriminator judges whether a data sample is from the real dataset or is generated by generator G. The generator and the discriminator are optimized in turn to improve their generation and discrimination ability, and a description of the dataset is learned by the GAN model. The novelty detection problem can benefit from such ability to describe the data distribution. In a novelty detection problem, the training dataset contains only normal samples, and the description of the distribution of normal data can be learned by the GAN model. Abnormal samples have a different distribution. Therefore, the novelty score can be designed using the trained generator and discriminator, and a novelty detection method is achieved.

2 Generative adversarial networks

The structure of GAN is shown in Fig. 1. The generator and the discriminator can be represented by differentiable functions with latent variable $\boldsymbol{z}$ and data sample $\boldsymbol{x}$ as input, respectively. Input data from a real dataset are labeled as 1, and generated data $G(\boldsymbol{z})$ are labeled as 0. The optimization of GAN is a minimax problem:

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

where $V(D,G)$ represents how the discriminator correctly judges real data and generated data. Generator G tries to maximize $V(D,G)$, while discriminator D tries to minimize it. G and D are both differentiable functions, so problem (1) can be optimized using gradient base methods.

The goal of the discriminator is to judge real data and generated data correctly, so it is updated by ascending the gradient to maximize $V(D,G)$:

$$\nabla_{\theta_d}\frac{1}{m}\sum_{i=1}^{m}\left[\log D\left(\boldsymbol{x}^{(i)}\right)+\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]. \tag{2}$$

The goal of the generator is to generate realistic data, and it minimizes $V(D,G)$ by descending the gradient to reduce the accuracy of the discriminator:

$$\nabla_{\theta_g}\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right). \tag{3}$$

G and D are updated alternately in each iteration until $V(D,G)$ converges. At this point, the generator cannot be improved to generate more realistic data and the discriminator cannot enhance its discrimination ability, and generator G learns the distribution of the real data. If $\min_G \max_D V(D,G)$ reaches the global optimum, then $p_g = p_{\text{data}}$ and G will generate data with the same distribution as that of real data.

The generator and the discriminator are typically formed with neural networks, such as multilayer perceptrons (Goodfellow et al., 2014; Arjovsky et al., 2017), convolutional neural networks (Radford et al., 2015), and recurrent neural networks (Mogren, 2016). Fig. 2 shows the results of training a GAN model using multilayer perceptrons on 2D synthetic datasets. The generator and the discriminator are both multilayer perceptrons. The hidden layers use leaky ReLU activations and the output

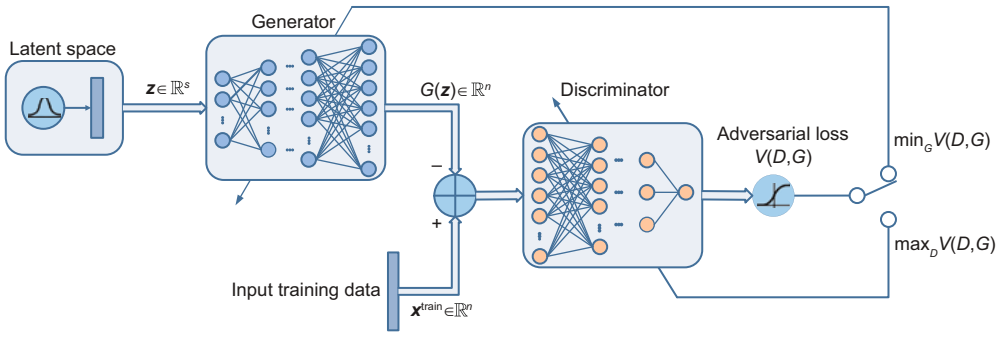

Fig. 1 Structure of the generative adversarial networks (GAN)

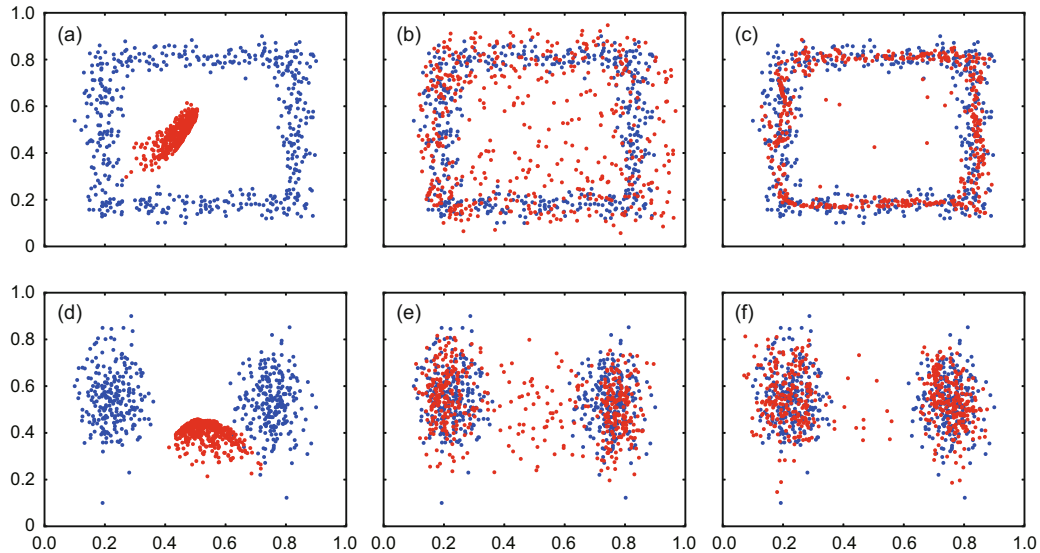

Fig. 2 Results of a GAN model using multilayer perceptrons on 2D synthetic datasets: (a)–(c) are the results of the 1st, the 300th, and the 3000th iteration, respectively, on a square distribution; (d)–(f) are the results of the 1st, the 300th, and the 3000th iteration, respectively, on a two-model distribution

Blue points represent real data from the synthetic dataset, and red points represent the points generated by the GAN model. References to color refer to the online version of this figure

layers use sigmoid activations. Figs. 2a–2c are results of the 1st, the 300th, and the 3000th iteration on a square distribution, respectively. Figs. 2d–2f are results of the 1st, the 300th, and the 3000th iteration on a two-model distribution, respectively. Blue points represent real data from the synthetic dataset, and red points represent the generated points of $G(\boldsymbol{z})$ when $\boldsymbol{z}$ is randomly sampled from a Gaussian distribution. Fig. 2 shows that GAN can generate data with a distribution similar to that for the training dataset, and the description of the training dataset is learned by the model.

The model for novelty detection needs to learn a description of the training dataset which contains only normal samples, and to formulate a novelty score so that abnormal samples have higher scores than normal samples. When a GAN model is trained on the training dataset, the model learns not only the distribution of the training data but also the distribution of the normal data, because samples in the training dataset are all normal ones. In the testing dataset, the description of normal samples conforms with the GAN model's description of normal samples, but the description of abnormal samples deviates from it. Using the generator and the discriminator in the trained GAN model, a novelty score is formulated to achieve novelty detection.

3 Generative adversarial networks for novelty detection

In novelty detection, the training dataset contains only samples with normal status. A GAN model is trained on the training dataset to learn the description of the normal data. Then the novelty score of each test sample is obtained from the trained GAN model. Samples with high novelty scores are detected as novel.

3.1 Adversarial novelty score

Training the GAN model on the training dataset containing only normal samples, the trained generator G and discriminator D involve the description of the normal data. The trained G and D are used to formulate the novelty score to evaluate the novelty of a sample.

The generated samples $\boldsymbol{x}_g = G(\boldsymbol{z})$ using a trained generator G are similar to a normal sample in the training dataset for any latent variable $\boldsymbol{z}$ in the latent space. When $\boldsymbol{x}$ is a normal sample, there exists a corresponding $\boldsymbol{z}$ so that the generated sample $\boldsymbol{x}_g = G(\boldsymbol{z})$ is very similar to $\boldsymbol{x}$; i.e., sample $\boldsymbol{x}$ can be reconstructed perfectly by the generated G. However, when $\boldsymbol{x}$ is an abnormal sample, $G(\boldsymbol{z})$ will have a large reconstruction error with $\boldsymbol{x}$ for any $\boldsymbol{z}$. Fig. 3 illustrates the reconstruction error between test samples and generated samples. Blue points represent training data samples and red points represent samples generated by a trained generator. Point A represents an abnormal test sample, and point B a normal test sample. Points A' and B' are the nearest generated samples of A and B, respectively, which suggest the best reconstruction of test samples that generator G can achieve. The distance between a test sample and its nearest generated sample is the least reconstruction error. The least reconstruction error of abnormal test sample A is much larger than that of normal test samples, because generator G generates only normal samples.

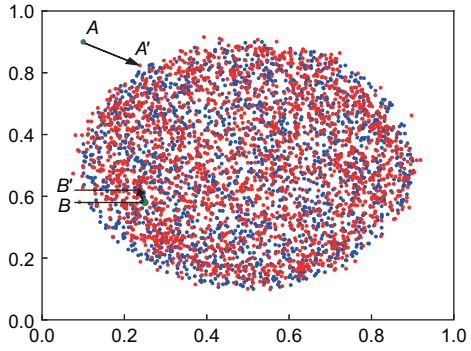

Fig. 3 Reconstruction error between test samples and generated samples

Blue points represent training data samples and red points represent samples generated by a trained generator. References to color refer to the online version of this figure

Therefore, we find the best latent variable $\boldsymbol{z}$ to minimize the reconstruction error of generator G for a given sample $\boldsymbol{x}$, and the minimized reconstruction error is formulated as the novelty score:

$$f_g(\boldsymbol{x}) = \min_{\boldsymbol{z}\in\mathbb{R}^s} \|\boldsymbol{x} - G(\boldsymbol{z})\|^2. \tag{4}$$

We call the novelty score formulated in Eq. (4) a G-score.

Fig. 4 illustrates the concept of G-score using the MNIST handwritten digits. Assume the '0' digits are normal and other kinds of digits abnormal. A training dataset is made up of part of the '0' digits and the GAN model is trained on it. The '0' and '1'

Fig. 4 Illustration of G-score: (a) and (b) are real '0' and '1' digits from the MNIST database not contained in the training dataset, respectively; (c) and (d) are '0' and '1' digits reconstructed by the generator, respectively

digits in Figs. 4a and 4b for test are real digits from the MNIST database that are not contained in the training dataset. Digits in Figs. 4c and 4d are those reconstructed by generator G. Fig. 4 shows that '0' digits can be well reconstructed while '1' digits are reconstructed with large reconstruction errors if GAN is trained on '0' digits. The reconstruction error in Eq. (4) can be used as a novelty score to distinguish normal and abnormal samples.

The trained discriminator D can also formulate a novelty score. Theoretically, when the GAN reaches the global optimum, discriminator D cannot distinguish between generated data and normal data in the training dataset. In practice, the discriminator can hardly reach the global optimum. The discriminator is trained with both normal data labeled as '1' and generated data labeled as '1'. In the early stage of training, the generated samples are different from normal samples, and the discriminator can learn how to distinguish between normal and abnormal samples. The discriminator based novelty score is formulated as

$$f_d(\boldsymbol{x}) = -D(\boldsymbol{x}), \tag{5}$$

where $D(x)$ represents the output of D for data $\boldsymbol{x}$, and the minus is used to make abnormal samples have higher novelty scores than normal ones. The novelty score in Eq. (5) is called D-score.

3.2 Algorithm of GAN-based novelty detection

The GAN-based novelty detection system trains a GAN model on the training dataset first. When the training is finished, parameters in the generator and the discriminator formulate novelty scores $f_g(\boldsymbol{x})$ and $f_d(\boldsymbol{x})$, respectively. Novelty scores of training samples are computed and thresholds are determined with a certain confidence level. Then novelty scores of test samples are computed.

Let $\boldsymbol{X}^{\text{train}} = \left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\right\}$ be the set of training samples, where the dimensionality of each sample $\boldsymbol{x}^{(i)} \in \mathbb{R}^n$ is n. All training samples are labeled as 1. Let $G(\boldsymbol{z})$ be the generator whose input is the latent variable $\boldsymbol{z} \in \mathbb{R}^s$. Let $D(\boldsymbol{x})$ be the discriminator whose input $\boldsymbol{x} \in \mathbb{R}^n$ has the same dimensionality as data samples. Then GAN is trained on training dataset $\boldsymbol{X}^{\text{train}}$ following the steps described in Section 2. When converging, G-score $f_g(\boldsymbol{x})$ and D-score $f_d(\boldsymbol{x})$ are formulated according to Eqs. (4) and (5), respectively.

When the G-score is used for novelty detection, G-scores of training samples are computed and a threshold T_g is determined so that 95% of training samples have scores lower than the threshold:

$$T_g = 95 \text{ quantile of } \left\{f_g(\boldsymbol{x}) | \boldsymbol{x} \in \boldsymbol{X}^{\text{train}}\right\}. \tag{6}$$

The decision function on the test dataset is defined as

$$h_g(\boldsymbol{x}' | \boldsymbol{X}^{\text{train}}) = \text{sgn}(f_g(\boldsymbol{x}') - T_g), \tag{7}$$

where $\boldsymbol{x}' \in \boldsymbol{X}^{\text{test}}$ is the test sample. When the G-score of a sample is higher than threshold T_g, the sample is judged as an abnormal one; otherwise, it is considered a normal sample.

When the D-score is used for novelty detection, a threshold T_d is determined so that 95% of training samples have D-scores lower than T_d:

$$T_d = 95 \text{ quantile of } \left\{f_d(\boldsymbol{x}) | \boldsymbol{x} \in \boldsymbol{X}^{\text{train}}\right\}, \tag{8}$$

and the decision function on the test dataset is

$$h_d(\boldsymbol{x}' | \boldsymbol{X}^{\text{train}}) = \text{sgn}(f_d(\boldsymbol{x}') - T_d), \tag{9}$$

where $\boldsymbol{x}' \in \boldsymbol{X}^{\text{test}}$ is the test sample, and samples whose D-scores are higher than T_d are judged as abnormal samples.

The steps of GAN-based novelty detection are listed in Algorithm 1.

4 Experiments

The GAN-based novelty detection methods using the proposed G-score and D-score are evaluated on the MNIST handwritten digits dataset and Tennessee Eastman benchmark process, and the PCA-based novelty detection methods using Hotelling's T^2 and squared prediction error (SPE) statistics are used for comparison.

Algorithm 1 GAN-based novelty detection

Input: training dataset $\boldsymbol{X}^{\text{train}} = \{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(N)}\}$ and test dataset $\boldsymbol{X}^{\text{test}} = \{\boldsymbol{x}'^{(1)}, \boldsymbol{x}'^{(2)}, \ldots, \boldsymbol{x}'^{(N^{\text{test}})}\}$

Output: novelty detection decision $h_g(\boldsymbol{x}'|\boldsymbol{X}^{\text{train}})$ and $h_d(\boldsymbol{x}'|\boldsymbol{X}^{\text{train}})$ for each $\boldsymbol{x}'$ in $\boldsymbol{X}^{\text{test}}$

1: Train the GAN model on $\boldsymbol{X}^{\text{train}}$ and obtain generator $G(\boldsymbol{z})$ and discriminator $D(\boldsymbol{x})$

2: Obtain the G-score and D-score functions $f_g(\boldsymbol{x})$ and $f_d(\boldsymbol{x})$ according to Eqs. (4) and (5), respectively

3: Determine the G-score and D-score thresholds T_g and T_d following Eqs. (6) and (8), respectively

4: For each $\boldsymbol{x}' \in \boldsymbol{X}^{\text{test}}$, obtain the G-score and D-score decisions $h_g(\boldsymbol{x}'|\boldsymbol{X}^{\text{train}})$ and $h_d(\boldsymbol{x}'|\boldsymbol{X}^{\text{train}})$ according to Eqs. (7) and (9), respectively

4.1 MNIST data

MNIST is a handwritten digit database (http://yann.lecun.com/exdb/mnist/). It contains 60 000 training digit samples and 10 000 test digit samples. Each sample is one of the digits from '0' to '9' and has a label suggesting which number the digit is. Each sample is a grayscale image of size 28×28.

To verify the performance of GAN-based novelty detection, '0' digits are assumed normal and other digits are assumed abnormal. The training dataset is made up of 4096 randomly chosen '0' digits. The test dataset is made up of 2048 '0' digits and 2304 other digits, where '0' digits are different from those in the training dataset and '1'–'9' digits each have 256 samples in the test dataset. The training dataset is shuffled before training the model.

When training the GAN model on the training dataset containing only '0' digits, the dimension of the latent variable $\boldsymbol{z}$ is set at $s = 100$. After training the model, the G-score and D-score are computed, and the results on the test dataset are shown in Fig. 5. The first 2048 samples in the test dataset are normal ones ('0' digits) and the last 2304 samples are abnormal ones ('1'–'9' digits).

PCA-based novelty detection is also applied in the experiment, using the same training and test datasets. The number of principal components is set at $p = 100$. Hotelling's T^2 and the SPE statistics are used as the novelty scores. T^2 measures a sample's deviation from the distribution center and SPE measures the reconstruction error of the principal component space.

Fig. 6 shows the reveiver operating characteristic (ROC) curves of the four novelty scores on the testing dataset. The horizontal axis represents the fraction of abnormal samples that are falsely judged as normal samples, and the vertical axis represents the fraction of normal samples that are correctly judged as normal samples. Fig. 6 also shows the area under curve (AUC) values of the four novelty scores. A larger AUC indicates a better performance of the model. On the MNIST dataset, the D-score has a larger AUC value than T^2 and SPE statistics, and the G-score has a lower AUC value. Fig. 4 shows that the reconstruction errors on normal samples in the text dataset are far from zero. This suggests that the GAN model may not be trained well and that the

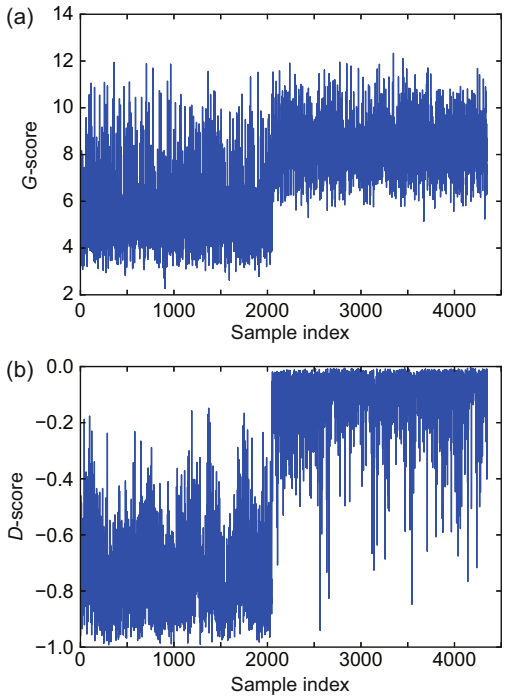

Fig. 5 ***G*-score (a) and *D*-score (b) on the test dataset in the MNIST experiment**

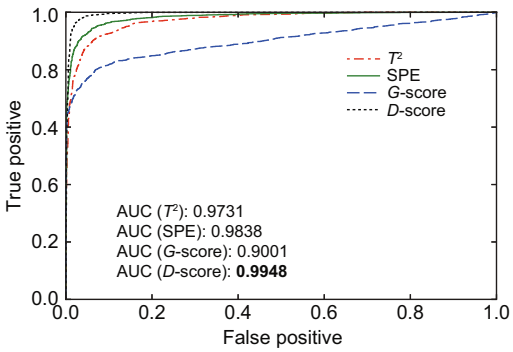

Fig. 6 Receiver operating characteristic (ROC) curves on the test dataset and the area under curve (AUC) value of each score

optimization is far from the global optimum. This may result in a better D-score performance but a worse G-score performance.

4.2 Tennessee Eastman process data

The Tennessee Eastman (TE) process is a simulation system based on a real chemical industry process proposed by Downs and Vogel (1993). It has been widely used as a benchmark for comparing fault detection methods (Mahadevan and Shah, 2009; Ge et al., 2011; Li and Maguire, 2011; Xiao et al., 2016). The flowchart of the TE process is shown in Fig. 7. The TE process consists of five unit operations (a reactor, a condenser, a compressor, a separator, and a stripper) and eight components (A, B, ..., H). Each sample has 52 variables, including 22 process variables, 19 composition variables, and 11 manipulated variables, The TE process contains one normal status and 21 faults [IDV(1), IDV(2), ..., IDV(21)]. The faults are described in Table 1.

In this study, 33 variables in the TE process are used to form the data samples. These variables are 22 process variables and 11 manipulated variables. The training dataset consists of 500 samples with normal status. There are 21 test datasets corresponding to the 21 faults. In each test dataset, the first 160 samples are normal, and the last 800 samples are abnormal with the fault introduced. When training the GAN model, the training dataset is first scaled to range (0, 1) and the dimension of the GAN latent variable $\boldsymbol{z}$ is set as $s = 2$. When training the PCA model, the training dataset is standardized to have zero mean and unit standard deviation, and the number of principal components p is set to 9.

Table 1 Fault types in the Tennessee Eastman (TE) process

Fault index	Description
1	A/C feed ratio, B composition constant (stream 4)
2	B composition, A/C ratio constant (stream 4)
3	D feed temperature (stream 2)
4	Reactor cooling water inlet temperature
5	Condenser cooling water inlet temperature
6	A feed loss (stream 1)
7	C header pressure loss-reduced availability (stream 4)
8	A, B, C feed composition (stream 4)
9	D feed temperature (stream 2)
10	C feed temperature (stream 4)
11	Reactor cooling water inlet temperature
12	Condenser cooling water inlet temperature
13	Reaction kinetics
14	Reactor cooling water valve
15	Condenser cooling water valve
16–20	Unknown
21	The valve for stream 4 fixed at the steady-state position

Fig. 8 shows the G-score and D-score on the test dataset of fault IDV(1). In Fig. 8a, G-score values

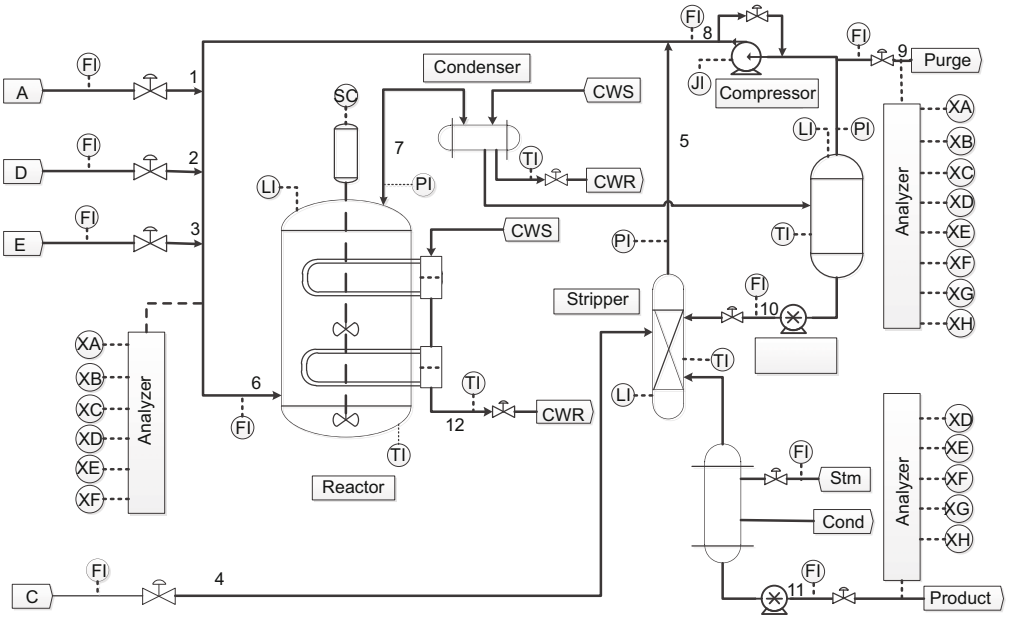

Fig. 7 Structure diagram of the Tennessee Eastman (TE) process

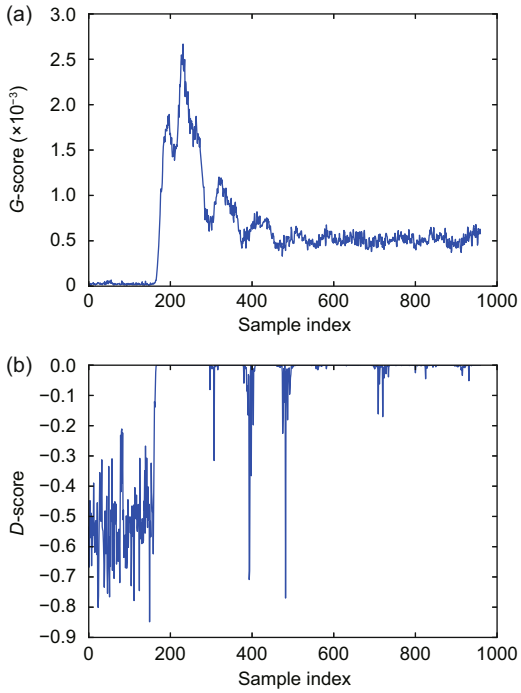

Fig. 8 *G*-score (a) and *D*-score (b) on the test dataset of the Tennessee Eastman (TE) process IDV(1)

of normal samples are very close to zero, suggesting that the GAN is trained better on the TE process than on the MNIST dataset.

The AUC values on 21 faults of the four novelty scores are shown in Table 2. For each fault, the AUC values of the four novelty scores are close to each other, showing similar performances. On the other hand, the *G*-score has higher AUC values than the *D*-score on almost all 21 faults, suggesting that the GAN is trained well on the TE process. Compared with the experimental results on MNIST, the *G*-score and *D*-score appear to be complementary. Such a property could be exploited to make GAN-based novelty detection less sensitive to hyperparameter selection.

5 Conclusions

In this paper, a generative adversarial network (GAN) based novelty detection method was proposed. In novelty detection, the training dataset contains only normal samples. GAN can generate new samples similar to the training data. This demonstrates its ability to describe the training data. Such an implicit data description of normal data was transformed to a novelty score for novelty detection by formulating the *G*-score and *D*-score. Experiments on MNIST and TE benchmark process showed a competitive performance compared with conventional methods like PCA.

Table 2 The area under curve (AUC) values on 21 faults of the Tennessee Eastman (TE) process

Fault	AUC			
	T^2	SPE	*G*-score	*D*-score
IDV(1)	0.9967	**0.9999**	0.9993	0.9962
IDV(2)	**0.9962**	0.9957	0.9956	0.9888
IDV(3)	0.5704	0.5234	0.4980	**0.6388**
IDV(4)	0.8314	**1.0000**	0.9012	0.5086
IDV(5)	**0.7540**	0.7301	0.7065	0.6812
IDV(6)	0.9994	**1.0000**	**1.0000**	**1.0000**
IDV(7)	0.9111	**1.0000**	**1.0000**	0.9843
IDV(8)	**0.9920**	0.9919	0.9918	0.9771
IDV(9)	0.4524	0.4907	**0.5111**	0.4535
IDV(10)	0.8238	0.8466	**0.8829**	0.7749
IDV(11)	0.7865	**0.9399**	0.8636	0.6730
IDV(12)	0.9951	0.9943	**0.9960**	0.9826
IDV(13)	0.9847	0.9764	**0.9918**	0.9646
IDV(14)	0.9981	**1.0000**	0.9999	0.8067
IDV(15)	0.6741	0.5645	**0.6864**	0.6131
IDV(16)	0.6047	**0.7465**	0.6983	0.6671
IDV(17)	0.9489	**0.9819**	0.9429	0.8878
IDV(18)	**0.9672**	0.9595	0.9652	0.9179
IDV(19)	0.6062	**0.8958**	0.6762	0.5449
IDV(20)	0.8601	**0.8868**	0.8545	0.8057
IDV(21)	0.7328	0.7418	**0.7938**	0.7086

The complementary properties of *G*-score and *D*-score are also discovered. On high-dimensional datasets like MNIST, GAN is likely to be trained less well, and *D*-score performs better than *G*-score. However, on low-dimensional datasets like those in the TE process, the generator is more ideally trained for the reconstruction error on normal samples close to zero. The two GAN-based novelty scores may be integrated for hyperparameter insensitivity.

Typically, generator G has no inverse function, so computing the *G*-score may entail a large time cost during minimizing the reconstruction error. Further study is required to find a new structure to directly map the data to the latent space to reduce this time.

As demonstrated by the experiments on the Tennessee Eastman process benchmark, the two novelty scores proposed in this study can be applied in industrial process monitoring and fault detection when the process variables and manipulated variables are measured to form the data samples. The scores can also be used in other areas like medical diagnosis when trained on medical measurements or images and used in drug discovery when molecules are represented in a feature space.

References

Abadi M, Andersen D, 2016. Learning to protect communications with adversarial neural cryptography. https://arxiv.org/abs/1610.06918

Arjovsky M, Chintala S, Bottou L, 2017. Wasserstein generative adversarial networks. Int Conf on Machine Learning, p.214-223.

Berthelot D, Schumm T, Metz L, 2017. BEGAN: boundary equilibrium generative adversarial networks. https://arxiv.org/abs/1703.10717

Clifton L, Clifton D, Watkinson P, et al., 2011. Identification of patient deterioration in vital-sign data using one-class support vector machines. Federated Conf on Computer Science and Information Systems, p.125-131.

Denton E, Chintala S, Fergus R, et al., 2015. Deep generative image models using a Laplacian pyramid of adversarial networks. Advances in Neural Information Processing Systems, p.1486-1494.

Donahue J, Krähenbühl P, Darrell T, 2016. Adversarial feature learning. https://arxiv.org/abs/1605.09782

Downs J, Vogel E, 1993. A plant-wide industrial process control problem. *Comput Chem Eng*, 17(3):245-255. https://doi.org/10.1016/0098-1354(93)80018-I

Dumoulin V, Belghazi I, Poole B, et al., 2016. Adversarially learned inference. https://arxiv.org/abs/1606.00704

Ge Z, Song Z, 2013. Bagging support vector data description model for batch process monitoring. *J Proc Contr*, 23(8):1090-1096. https://doi.org/10.1016/j.jprocont.2013.06.010

Ge Z, Yang C, Song Z, 2009. Improved kernel PCA-based monitoring approach for nonlinear processes. *Chem Eng Sci*, 64(9):2245-2255. https://doi.org/10.1016/j.ces.2009.01.050

Ge Z, Gao F, Song Z, 2011. Batch process monitoring based on support vector data description method. *J Proc Contr*, 21(6):949-959. https://doi.org/10.1016/j.jprocont.2011.02.004

Ge Z, Song Z, Gao F, 2013. Review of recent research on data-based process monitoring. *Ind Eng Chem Res*, 52(10):3543-3562. https://doi.org/10.1021/ie302069q

Ge Z, Demyanov S, Chen Z, et al., 2017. Generative OpenMax for multi-class open set classification. https://arxiv.org/abs/1707.07418

Goodfellow I, Pouget-Abadie J, Mirza M, et al., 2014. Generative adversarial nets. Advances in Neural Information Processing Systems, p.2672-2680.

Grover A, Ermon S, 2017. Boosted generative models. https://arxiv.org/abs/1702.08484

Hautamaki V, Karkkainen I, Franti P, 2004. Outlier detection using k-nearest neighbour graph. Proc 17th Int Conf on Pattern Recognition, p.430-433. https://doi.org/10.1109/ICPR.2004.1334558

He Z, Deng S, Xu X, 2005. An optimization model for outlier detection in categorical data. *LNCS*, 3644:400-409. https://doi.org/10.1007/11538059_42

Hoffmann H, 2007. Kernel PCA for novelty detection. *Patt Recogn*, 40(3):863-874. https://doi.org/10.1016/j.patcog.2006.07.009

Kadurin A, Aliper A, Kazennov A, et al., 2017a. The cornucopia of meaningful leads: applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8(7):10883. https://doi.org/10.18632/oncotarget.14073

Kadurin A, Nikolenko S, Khrabrov K, et al., 2017b. druGAN: an advanced generative adversarial autoencoder model for de novo generation of new molecules with desired molecular properties in silico. *Mol Pharmaceut*, 14(9):3098-3104. https://doi.org/10.1021/acs.molpharmaceut.7b00346

Keogh E, Lonardi S, Ratanamahatana C, 2004. Towards parameter-free data mining. Proc 10th ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.206-215. https://doi.org/10.1145/1014052.1014077

Kim T, Cha M, Kim H, et al., 2017. Learning to discover cross-domain relations with generative adversarial networks. https://arxiv.org/abs/1703.05192

Ledig C, Theis L, Huszár F, et al., 2016. Photo-realistic single image super-resolution using a generative adversarial network. https://arxiv.org/abs/1609.04802

Li J, Liang X, Wei Y, et al., 2017. Perceptual generative adversarial networks for small object detection. CVPR, p.1951-1959. https://doi.org/10.1109/CVPR.2017.211

Li Y, Maguire L, 2011. Selecting critical patterns based on local geometrical and statistical information. *IEEE Trans Patt Anal Mach Intell*, 33(6):1189-1201. https://doi.org/10.1109/TPAMI.2010.188

Li Y, Liu S, Yang J, et al., 2017. Generative face completion. CVPR, p.5892-5900. https://doi.org/10.1109/CVPR.2017.624

Luc P, Couprie C, Chintala S, et al., 2016. Semantic segmentation using adversarial networks. https://arxiv.org/abs/1611.08408

Mahadevan S, Shah S, 2009. Fault detection and diagnosis in process data using one-class support vector machines. *J Proc Contr*, 19(10):1627-1639. https://doi.org/10.1016/j.jprocont.2009.07.011

Mao X, Li Q, Xie H, et al., 2016. Least squares generative adversarial networks. https://arxiv.org/abs/1611.04076

Mogren O, 2016. C-RNN-GAN: continuous recurrent neural networks with adversarial training. https://arxiv.org/abs/1611.09904

Patcha A, Park J, 2007. An overview of anomaly detection techniques: existing solutions and latest technological trends. *Comput Netw*, 51(12):3448-3470. https://doi.org/10.1016/j.comnet.2007.02.001

Pimentel M, Clifton D, Clifton L, et al., 2014. A review of novelty detection. *Signal Process*, 99:215-249. https://doi.org/10.1016/j.sigpro.2013.12.026

Radford A, Metz L, Chintala S, 2015. Unsupervised representation learning with deep convolutional generative adversarial networks. https://arxiv.org/abs/1511.06434

Reed S, Akata Z, Yan X, et al., 2016. Generative adversarial text to image synthesis. Proc 33rd Int Conf on Machine Learning, p.1060-1069.

Schlegl T, Seeböck P, Waldstein S, et al., 2017. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. Int Conf on Information Processing in Medical Imaging, p.146-157. https://doi.org/10.1007/978-3-319-59050-9_12

Springenberg J, 2015. Unsupervised and semi-supervised learning with categorical generative adversarial networks. https://arxiv.org/abs/1511.06390

Vondrick C, Pirsiavash H, Torralba A, 2016. Generating videos with scene dynamics. Advances in Neural Information Processing Systems, p.613-621.

Wu J, Zhang C, Xue T, et al., 2016. Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. Advances in Neural Information Processing Systems, p.82-90.

Xiao Y, Wang H, Xu W, et al., 2016. Robust one-class SVM for fault detection. *Chemometr Intell Lab Syst*, 151: 15-25. https://doi.org/10.1016/j.chemolab.2015.11.010

Yang Z, Chen W, Wang F, et al., 2017. Improving neural machine translation with conditional sequence generative adversarial nets. https://arxiv.org/abs/1703.04887

Yeh R, Chen C, Lim T, et al., 2016. Semantic image inpainting with perceptual and contextual losses. https://arxiv.org/abs/1607.07539

Yi Z, Zhang H, Gong P, et al., 2017. DualGAN: unsupervised dual learning for image-to-image translation. https://arxiv.org/abs/1704.02510

Yu J, 2012. Semiconductor manufacturing process monitoring using Gaussian mixture model and Bayesian method with local and nonlocal information. *IEEE Trans Semicond Manuf*, 25(3):480-493. https://doi.org/10.1109/TSM.2012.2192945

Yu J, Qin S, 2008. Multimode process monitoring with Bayesian inference-based finite Gaussian mixture models. *AIChE J*, 54(7):1811-1829. https://doi.org/10.1002/aic.11515

Yu J, Qin S, 2009. Multiway Gaussian mixture model based multiphase batch process monitoring. *Ind Eng Chem Res*, 48(18):8585-8594. https://doi.org/10.1021/ie900479g

Yu L, Zhang W, Wang J, et al., 2017. SeqGAN: sequence generative adversarial nets with policy gradient. 31st AAAI Conf on Artificial Intelligence, p.2852-2858.

Zhao F, Feng J, Zhao J, et al., 2018. Robust LSTM-autoencoders for face de-occlusion in the wild. *IEEE Trans Image Process*, 27(2):778-790. https://doi.org/10.1109/TIP.2017.2771408

Zhao J, Mathieu M, LeCun Y, 2016. Energy-based generative adversarial network. https://arxiv.org/abs/1609.03126

Zhu J, Park T, Isola P, et al., 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. https://arxiv.org/abs/1703.10593