A Comparison of GANs-Based Approaches for Combustor System Fault Detection

Rui Xu
Artificial Intelligence & Machine Learning Laboratory
GE Research
Niskayuna, NY, USA
xur@ge.com

Weizhong Yan
Artificial Intelligence & Machine Learning Laboratory
GE Research
Niskayuna, NY, USA
yan@ge.com

***Abstract*—In manufacturing industry, anomaly detection (AD) has been widely applied to monitor asset operation status and provide decision support for proactive maintenance. However, the extreme complexities of many industrial assets have raised many challenges to the classical anomaly detection approaches. For the past several years, generative adversarial networks (GANs) have achieved breakthrough in a variety of applications, such as image generation and video prediction. Some initial applications of GANs in image-related AD problems also show promising results. In this paper, we investigated the performance of three GANs-based anomaly detection approaches for a specific industrial use case – fault detection of the combustion system of gas turbines in power plants. The results show that under the framework of semi-supervised learning, the three approaches do not perform well on the one-year field data collected from a gas turbine. However, if a small portion of fault data is provided for training, we observe that the performance of GANs is significantly improved.**

Keywords—Generative adversarial networks (GANs); anomaly detection; semi-supervised learning; supervised learning; combustor system; gas turbine; manufacturing industry

I. Introduction

Anomaly detection (AD), a technique for finding patterns in data that do not conform to expected behavior, has been extensively used in a wide range of applications, such as fraud detection in credit card and insurance industries, intrusion detection in cyber-security industry, fault detection in industrial analytics, to name a few [1–6]. In manufacturing industry, AD is an important component of prognostics and system health management (PHM), a modern maintenance strategy designed for reducing operation and maintenance (O&M) costs by reducing unscheduled repairs and increasing availability of assets [7–11].

Gas turbines play a critical role in combined cycle power plants. The combustion system, or combustor, of a gas turbine, where fuel-air mixture is burned for generating power, is a critical component of gas turbines. During normal operations, gas turbine combustors undergo significant thermodynamic stresses due to their high temperature and high flow rate conditions. The thermodynamic stresses as well as many other factors, e.g., fuel distribution imbalance and combustion instabilities, result in different combustors' abnormalities over the life of the combustion system, including faults at the fuel nozzle, creaks in the liner, defects in the transition piece, and excessive vibration [12]. Those abnormalities, if not detected early, could potentially lead to catastrophic combustor failures, resulting in turbine trips. Reliably detecting combustors' abnormalities earlier, therefore, becomes crucial in ensuring that gas turbines operates efficiently. This prevents costly turbine trips and, ultimately, minimizes operation and maintenance (O&M) costs in power plant operations.

Developing a reliable anomaly detection system for gas turbine combustors has many challenges [13]. Firstly, as a system, gas turbine combustors are extremely complex and are operated under a harsh environment (high temperature and high flow rate) and various operating conditions/modes. Secondly, the number of relevant sensors available for fault detection is often limited. Lastly, sensor measurements are noisy and unreliable due to the faulty or failed sensors. As a result, most of existing combustor AD systems are inadequate at meeting the detection performance requirements. The power generation industry as well as the PHM community have been constantly seeking more advanced machine learning techniques that would enable them to improve the detection performance.

For the past several years, generative adversarial networks (GANs) have achieved breakthrough in a variety of applications, such as image generation and video prediction [14–15]. Under the framework of GANs, two players, a generator and a discriminator, play a game, in which the generator tries to generate data that are intended come from the same distribution of the training data, while the discriminator tries to distinguish the real and the generated data. The first application of GANs in AD, known as AnoGAN, was reported in [16], which aims to detect

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

anomalies in medical images. The anomaly score is defined as a combination of both the reconstruction error and the feature matching loss in the intermediate layer of the discriminator. Efficient GAN-based anomaly detection (EGAD) is then proposed to address the performance issue of AnoGAN, which requires additional and costly optimization steps to find the latent representation for every new input data point [17]. EGAD uses the BiGAN [18] model to include an encoder to learn the mappings between the real data and the latent representation, which is simultaneously trained with the generator. Adversarially learned anomaly detection (ALAD) further combines additional discriminators to improve the encoders, and ALAD also introduces spectral normalization to help stabilize the training [19]. Different from EGAD and ALAD, GANomaly addresses the computationally intensive run-time testing issue of AnoGAN by adding an autoencoder into the generator [20]. The anomaly scores of GANomaly depends on the encoder loss and feature scaling is applied to make the anomaly scores within the probabilistic range.

Contrary to the aforementioned approaches, whose generators aim to learn the real data distributions, it is shown in [21] that to make the discriminator to be a good classifier, the generator needs to be bad in terms of approximating the real data distributions. In other words, the objective of the generator is to create complementary samples to the true data, rather than just learn the true data distributions. Guided by this idea, two approaches have been proposed to distinguish normal and fault data, known as complementary GAN [22] and fence GAN [23]. Complementary GAN defines the complementary distribution of the true data and constructs the objective function by minimizing the KL divergence between the complementary distribution and the learned generative distribution. Fence GAN modifies the generator cost function to focus on the generation of data samples on the boundary of true data distribution. For both approaches, it is not necessary to modify the architecture of the original GANs and the anomalies can be detected based on the output of discriminator without the need to define the anomaly functions.

The GANs-based AD approaches are mostly used for image related applications, such as the benchmark MNIST and CIFAR-10 data sets [24] and medical images [16]. To the authors' knowledge, there are no results reported in the literature on the performance of these approaches on applications in the manufacturing industry, which calls for more advanced AD approaches urgently. In this paper, we investigate the performance of EGAD, complementary GAN, and fence GAN in a data set that contains one-year field data collected from a gas turbine of a power plant. As in practice, the fault data is usually very limited, it is very interesting to investigate whether complementary GAN and fence GAN could potentially generate data that provide some insights on the anomalies. However, according to our experimental results, under the framework of semi-supervised learning, i.e., only normal data is used for training, all three approaches, in their original AD design settings, cannot achieve satisfactory performance on the data set. We propose to inject certain amount of fault data into the training set to help the discriminator learn an effective decision boundary between the normal and fault. By applying complementary GAN under the supervised setting, we observe that even a small amount of fault data can significantly improve the performance of fault detection.

The remainder of this paper is organized as follows. Section II briefly review the three GANs-based anomaly detection approaches. In Section III, we present and compare empirical results of these methods on the gas turbine combustion system data. Section IV concludes this paper.

II. Methods

A. A Brief on GANs

GANs were originally proposed to model complicated and high-dimensional real-world (mostly image) data distribution [14–15], and were demonstrated to be successful in many image generation related applications, as well as in video prediction [25], music generation [26], and so on. The basic idea of GANs is to simultaneously train two networks, a generator *G* and a discriminator *D*, via an adversarial process, in which *G* aims to capture the data distribution while *D* learns to distinguish between real data and data generated by *G* (see Fig. 1). Such a framework can be formalized as a two-player min-max game with the following cost function *V*(*G*, *D*),

$$\min_{G}\max_{D} V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}[\log(D(\mathbf{x})] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}[\log(1-D(G(\mathbf{z}))], \quad (1)$$

where $p_{\mathbf{z}}(\mathbf{z})$ is a prior on a noise variable $\mathbf{z}$ and $p_{data}(\mathbf{x})$ is the real data distribution.

Theoretically, this min-max game has a global optimum when the generator's learned distribution p_G is equal to the real data distribution, i.e. $p_G = p_{data}$. In practice, *G* and *D* are typically built by using gradient descent alternatively on the weights of the two networks.

In the context of anomaly detection, the problem can usually be formulated under the framework of semi-supervised learning. In other words, the generator *G* tries to learn the distribution of the normal data (From now on, we assume that only normal data are available for GANs unless otherwise indicated). After the training process is complete, the normal data distribution can then be approximated by p_G.

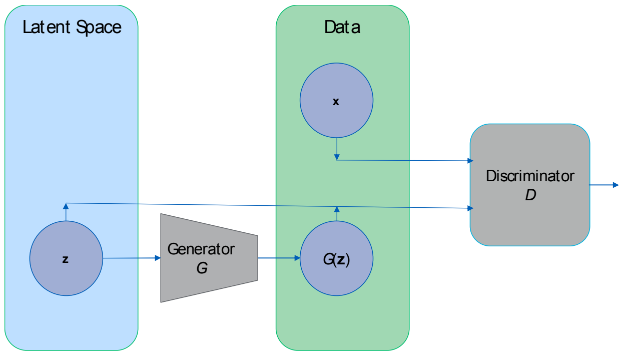

Fig. 1. The framework of GANs that inclues a discriminator *D* and a generator *G*.

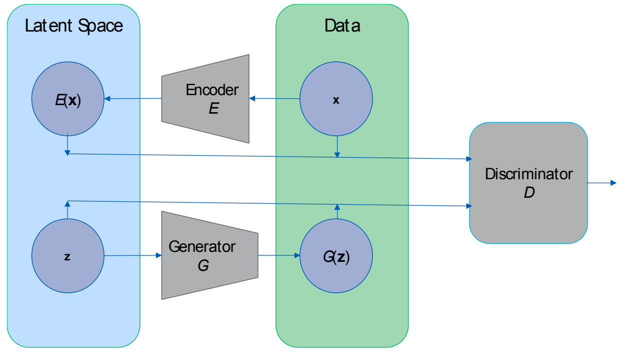

Fig. 2. The framework of EGAD. An encoder E is included to learning the mapping from data to the latent space.

An anomaly score can be defined based on the input reconstruction error and discrimination loss [16], which will be further discussed in the next section.

B. *Efficient GANs-based Anomaly Detection (EGAD)*

EGAD is based on the idea of BiGAN [18], which introduces an encoder into the framework so that the mapping between the normal data and latent space can be learned, along with G and D [17]. As shown in Fig. 2, D must learn to discriminate a pair of inputs ($\mathbf{x}$, $E(\mathbf{x})$) or ($G(\mathbf{z})$, $\mathbf{z}$), rather than just real or fake inputs.

Correspondingly, the cost function, $V(G, D, E)$, can be written as

$$\min_{G,E} \max_{D} V(D,G,E) = \mathbb{E}_{\mathbf{x}\sim p_{data}(\mathbf{x})}\left[\mathbb{E}_{\mathbf{z}\sim p_E(\mathbf{z}|\mathbf{x})}[\log(D(\mathbf{x},\mathbf{z})]\right] + \mathbb{E}_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\mathbb{E}_{\mathbf{x}\sim p_G(\mathbf{x}|\mathbf{z})}[\log(1-D(\mathbf{x},\mathbf{z})]\right], \qquad (2)$$

where $p_E(\mathbf{z}|\mathbf{x})$ is the distribution induced by the encoder and $p_G(\mathbf{x}|\mathbf{z})$ is the distribution induced by the generator. It is worthwhile to emphasize that E and G need to be learned jointly in practice.

To measure the anomality of a new input sample $\mathbf{x}$, an anomaly score function S(x) can then be defined as follows,

$$S(\mathbf{x}) = \lambda L_G(\mathbf{x}) + (1-\lambda)L_D(\mathbf{x}), \qquad (3)$$

where the reconstruction loss $L_G(\mathbf{x})$ is

$$L_G(\mathbf{x}) = \sum\left|\mathbf{x} - G\left(E(\mathbf{x})\right)\right|, \qquad (4)$$

and the discriminator-based loss $L_D(\mathbf{x})$ can be either calculated as the cross-entropy loss or defined as the feature matching loss,

$$L_D(\mathbf{x}) = \sum\left|f_D\left(\mathbf{x}, E(\mathbf{x})\right) - f_D\left(G\left(E(\mathbf{x})\right), E(\mathbf{x})\right)\right|, \qquad (5)$$

where f_D represents the output of the intermediate layer of the discriminator, which typically is the layer right preceding the logits. The feature mapping loss evaluates whether the generated sample from G has the similar features in D as the real input sample.

C. *Complementary GAN*

In contrast to the generator of original GANs, the goal of the generator of complementary GANs is to approximate the complementary distribution p' of the real data, which is defined as [22],

$$p'(\mathbf{x}') = \begin{cases} \frac{1}{\tau \times p_{data}(\mathbf{x}')} & \text{if } p_{data}(\mathbf{x}') > \varepsilon \text{ and } \mathbf{x}' \in \mathcal{B}_{\mathbf{x}} \\ C & \text{if } p_{data}(\mathbf{x}') \leq \varepsilon \text{ and } \mathbf{x}' \in \mathcal{B}_{\mathbf{x}} \end{cases}, \qquad (6)$$

where $\mathbf{x}' = G(\mathbf{z})$ is the generated sample, τ is a normalization term, C is a small constant, ε is a threshold to decide whether the generated samples fall in the high-density regions, and $\mathcal{B}_{\mathbf{x}}$ represents the space of real data samples.

As such, the cost function of G can be formulated below, which aims to minimize the KL divergence between p' and p_G,

$$\min_{G} V(D,G) = -\mathcal{H}(p_G) + \mathbb{E}_{\mathbf{x}'\sim p_G} log\ p_{data}(\mathbf{x}')\mathbf{1}[\, p_{data}(\mathbf{x}') > \mathcal{E}] + \left\|\mathbb{E}_{\mathbf{x}'\sim p_G} f_D(\mathbf{x}') - \mathbb{E}_{\mathbf{x}\sim p_{data}} f_D(\mathbf{x})\right\|^2, \qquad (7)$$

where the first term $\mathcal{H}(\cdot)$ is the entropy, $\mathbf{1}[\cdot]$ is the indicator function, and the third term is the feature matching loss.

The goal of the discriminator D is to discriminate the generated complementary samples from G from the normal data, with its cost function defined as,

$$\max_{D} V(D,G) = \mathbb{E}_{\mathbf{x}\sim p_{data}}[\log(D(\mathbf{x})] + \mathbb{E}_{\mathbf{x}'\sim p_G}[\log(1 - D(\mathbf{x}')] + \mathbb{E}_{\mathbf{x}\sim p_{data}}[(D(\mathbf{x})\log(D(\mathbf{x})]. \qquad (8)$$

Compared to the cost function in the original GANs, the third term, which calculates the conditional entropy, is added to focus on the identification of normal data with high confidence.

To make complementary GAN work in a supervised way, we add a small portion of fault data into the training set and provide the data to the discriminator with a label different from the normal samples. As the fault data is very limited, we included all available fault samples for every batch of normal samples during training.

Different from EGAD, the output from the discriminator can be interpreted as the probability of being the normal samples, and therefore, no additional anomaly score function is needed.

D. *Fence GAN*

Like complementary GANs, the objective of Fence GAN is to not approximate the normal data distribution, but to

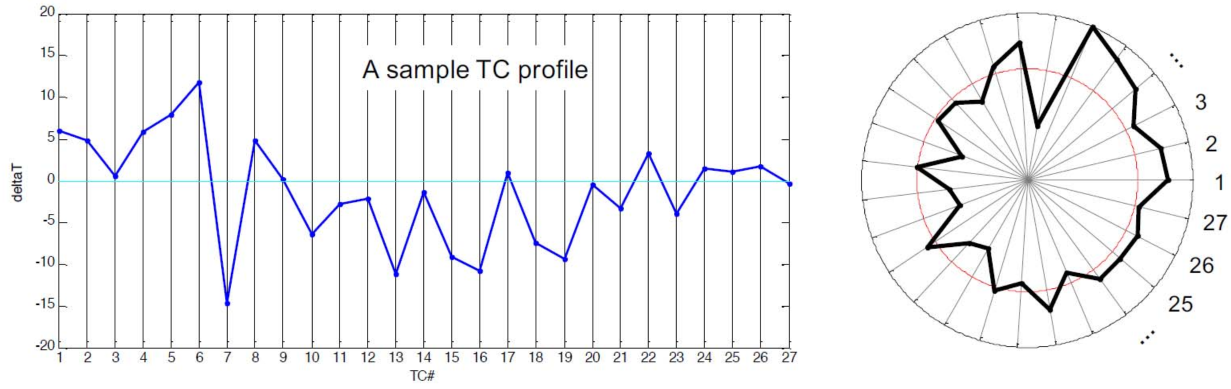

Fig. 3. A sample of TC profile.

generate samples that lie around the boundary of the normal data distribution [23].

To achieve that, the cost function of the generator is designed to consist of two terms, the encirclement loss *EL* and the dispersion loss *DL*, written as

$$\underset{G}{\mathrm{Min}} V(D,G) = EL + \beta \times DL$$
$$= \frac{1}{N}\sum_{i=1}^{N}[\log(|\alpha - D(G(\mathbf{z}_i))|)] + {}^{\beta}\Big/_{\frac{1}{N}\sum_{i=1}^{N}\|G(\mathbf{z}_i) - \boldsymbol{\mu}\|_2} \tag{9}$$

where N is the number of normal data samples, β is the dispersion hyperparameter, $\alpha \in (0,1)$ is a hyperparameter that functions as a discriminator score for data samples lying on the boundary of normal data, and $\boldsymbol{\mu}$ is the center of mass of the generated samples.

As shown in Eq. 9, the encirclement loss will penalize G when the generated points fall either inside or far away from the normal data distribution, and the dispersion loss tries to make the generated points cover the entire boundary to avoid model collapse.

Similarly, the cost function for the discriminator can be defined as,

$$\max_{D} V(D,G) = \frac{1}{N}\sum_{i=1}^{N}\left[-\log\left(D(\mathbf{x}_i)\right) - \gamma \times \log(1 - D(G(\mathbf{z}_i)))\right] \tag{10}$$

where $\gamma \in (0,1]$ is the anomaly hyperparameter that provides the tradeoff to focus more on the correct classification of real data samples or the generated samples.

Similar to complementary GAN, the output of discriminator lies in the probability range of [0, 1], and can be directly used to decide whether the given sample is an anomaly or not.

III. Experiment results

A. Data Set and Experimental Design

Our case study is concerned with anomaly detection of a combustor system used in industrial gas turbines of combined cycle power plants. The combustor system concerned has 27 combustor chambers circumferentially located around the core section of the turbine, through which the expanded hot gas is guided through turbine stages to derive work. There are 27 thermal couples (TC) installed at the turbine exhaust section to measure the exhaust gas temperatures. All of the TC temperature measurements at a given time form a so-called TC profile; such TC profile and its change over times can be used for assessing the thermal performance of the gas turbine and its combustors. In fact, using the TC profile to infer combustor health condition is an industry-wide practice [27]. Fig. 3 shows a typical TC profile (after mean normalization) of the 27 combustor chambers.

Our company's database has many years of field data collected on fleets of gas turbines. For case study in this paper, we retrieved approximately a year's worth of data for one turbine from the database. The retrieved data is in CSV format, where columns are variables or attributes and rows are time stamps. The variables include the 27 TC temperature measurements, operation conditions (megawatt, speed, etc.), and ambient conditions (temperature, pressure, and humidity). The data sampling rate is once-per-minute. During the year of operation, 10 true events occurred, which correspond to combustor anomalies or faults. Samples collected at the time when the 10 events occurred are treated as abnormal samples. Considering the fact that combustor abnormalies or faults happen gradually, we also take 30 samples before the time when each of the events occurred and treat these 30 samples as abnormal as well, which gives us a total of 300 abnormal samples. The rest of the samples are considered as normal samples. After filtering out bad data points and those data points corresponding to different load conditions (e.g., part-load), we end up with 13,791 normal samples before the events, and 47,528 normal samples after the events.

In this study, all normal samples are combined and reshuffled to be a single normal data set. Under the framework of semi-supervised learning, a five-fold cross validation is

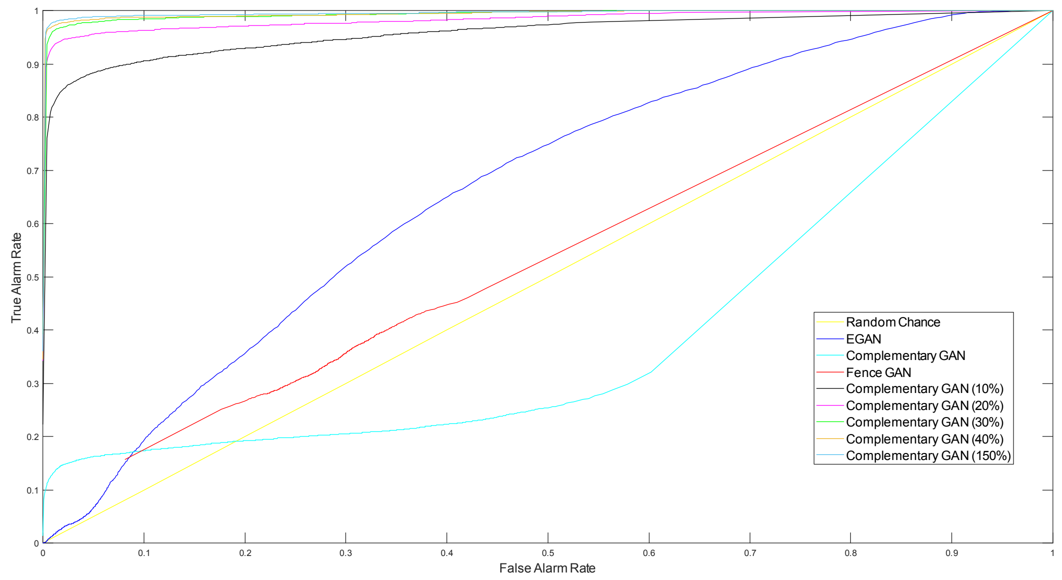

Fig. 4. Comparion of ROCs of three GAN-based approacehes, together with ROCs of complementary GAN under supervised setting.

performed on the normal data set with four folds of data used for training and one for test each time. The 300 abnormal samples are only used for testing. For complementary GAN, we also investigate the impact on the performance if some abnormal samples are used for training. For this supervised setting, we set aside 10%, 20%, 30%, 40%, and 50% of fault data for training, respectively, and use the remaining fault data for testing.

For all 3 types of GANs, the dimensions of the latent variables are all set at 12, and the discriminators and generators are feedforward neural networks that have the same architecture. For the discriminator, there are two hidden layers with 128 nodes for each. For the generator, the two hidden layers have 16 and 20 nodes, respectively. For EGAD, the encoder is also a feedforward neural network with 2 hidden layers that have 20 and 16 nodes, respectively. For all the hidden layers, LeakyReLU is used as the activation function, and dropout is applied with the probability rate set at 0.2. For Fence GAN, we set both α and γ at 0.5, and set β at 15. The number of training epochs is set at 5,000 for all GANs.

B. *Results*

Figs. 4 and 5 depict the receiver operating characteristic (ROC) curves and the recall-precison curves for EGAD, complentary GAN, and fence GAN on the gasturbine combustor system data set. The performance of complementary GAN under the supervised learning setting is also reported in these plots. The curves are generated based on the means of five different random runs. The corresponding area-under-curve (AUC), in terms of mean ± standard deviation, and the best F1-score for each approch are summarized in Table I. As shown in the plots and the table, EGAN has the largest AUC, while complementary GAN achieves the best F1-score. However, for all three approaches under the semi-supervised learning setting, the performances are not satisfactory with the best F1-score being only 0.2079. In constrast, under the setting of supervised learning, the performance of complementary GAN is significantly improved with mean AUC reaching 0.9624 and the best F1-score being 0.8921 even when only 10% fault data is used for training. The peformance continues to improve as more fault data are used for training. When 50% of fault data are included in the training set, the mean AUC is 0.9950 with the best F1-score increasing to 0.9447. Such results suggest that without any fault samples for training, complementary GAN and Fence GAN have some difficulty in training the discriminators to find a good decision boundary to separate the normal samples from the fault ones. For EGAD, the discriminator is not explicitly trained to build the boundary for normal and fault samples. With the regular minmax loss functions, the constructed anomaly score is not effective in distinguishing fault samples from normal points.

Fig. 6 illustrates the output probabilities of being normal samples by the discriminator of complementary GAN as the training process progresses. When the algorithm is only trained with normal samples (left plot), the output probabilities (cyan line) for the fault samples increases sharply in the beginning, then vibrates around the green line, which is the probabilities for the normal samples. The discriminator, however, still can separate well between the normal samples and the created samples from the generator. In this sense, the generator learns to generate data from low density regions, but fails to learn the representation of these fault samples. On the other hand, if certain amount of fault samples is provided for training (20% for the right plot), we can see that the

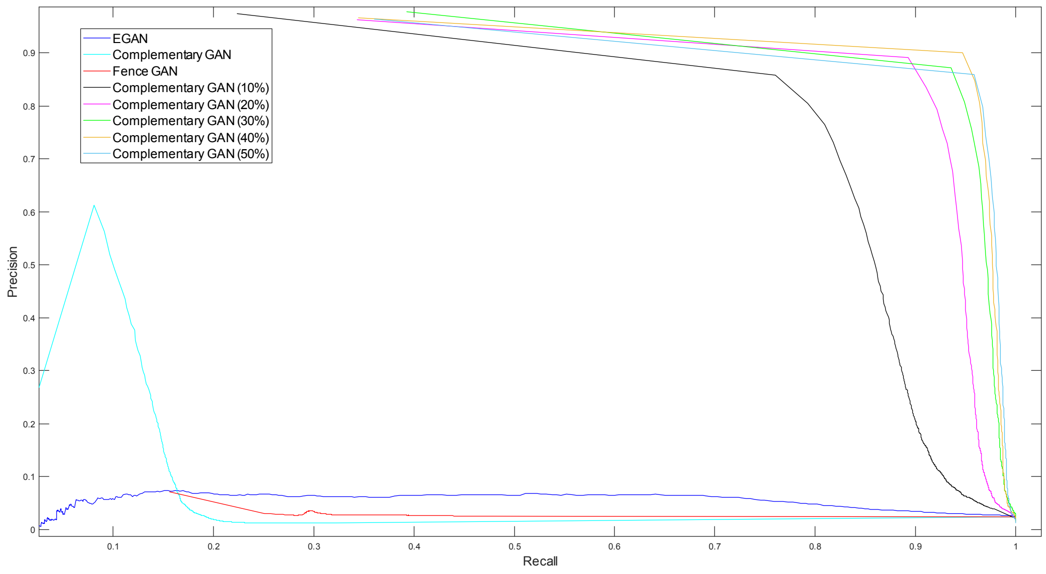

Fig. 5. Rceall-Precison curves of three GAN-based approacehes, together with curves of complementary GAN in supervised settings.

probabilities for being normal are very low for the fault samples. The probability line for the generated samples lies between the lines for the normal and fault samples, which is what we expect.

TABLE I. AUC (MEAN±STANDARD DEVIATION) AND BEST F1-SCORE COMPARISON OF EGAD, COMPLEMENTARY GAN, AND FENCE GAN, TOGETHER WITH COMPLEMENTARY GAN UNDER SUPERVISED SETTING

Model	**AUC**	**F1-score**
EGAN	0.7312±0.0824	0.1727
Complementary GAN	0.3666±0.1183	0.2079
Fence GAN	0.5417±0.1679	0.1835
Complementary GAN (10%)	0.9624±0.0215	0.8921
Complementary GAN (20%)	0.9845±0.0135	0.9222
Complementary GAN (30%)	0.9941±0.0053	0.9339
Complementary GAN (40%)	0.9950±0.0048	0.9466
Complementary GAN (50%)	0.9964±0.0039	0.9447

Fig. 7 shows the projections of the normal, fault, and generated data from the generator of fence GAN into 2 dimensions by using the t-distributed stochastic neighbor embedding (t-SNE) [28], which provides some insights why the algorithm does not perform well on this data set (we observe similar results for the complimentary GAN). Particularly, the normal data distributions are multimodal and the number of samples of those modes varies significantly, which make it very difficult for the generator to learn the distributions of those modes with sparse samples. As a result, only a few major modes of data are generated, which is known as the problem of mode collapse [15]. From Fig. 7, one can also observe that the margins between some fault samples and normal samples are very small, which requires further tuning of the hyperparameters of fence GAN. However, the performance of fence GAN is sensitive to the three hyperparameters and the parameter tuning is not trivial without some effective guidance.

IV. CONCLUSIONS

This paper focuses on the investigation of GANs-based anomaly detection approaches in applications from manufacturing industry, rather than commonly image or video related problems. Here, we compare the performance of EGAD, complementary GAN, and Fence GAN on a real industrial data set, which is collected from combustor systems of gas turbines of combined cycle power plants. The results show that under the setting of semi-supervised learning, the detection performances of these algorithms are not satisfactory. However, the inclusion of some fault data in the training set seems to be very effective in improving the algorithm performance. The future work includes how to design more effective loss functions to build the boundaries of normal data, particularly for data with high dimensionality and multi-modals with unbalanced data samples.

REFERENCES

[1] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A Survey,” ACM Computing Surveys, vol. 41, no. 3, article no. 15, 2009.

[2] M. Pimentel, D. Clifton, L. Clifton, and L. Tarassenko, “A review of novelty detection,” Signal Processing, vol.99, pp. 215–249, 2014.

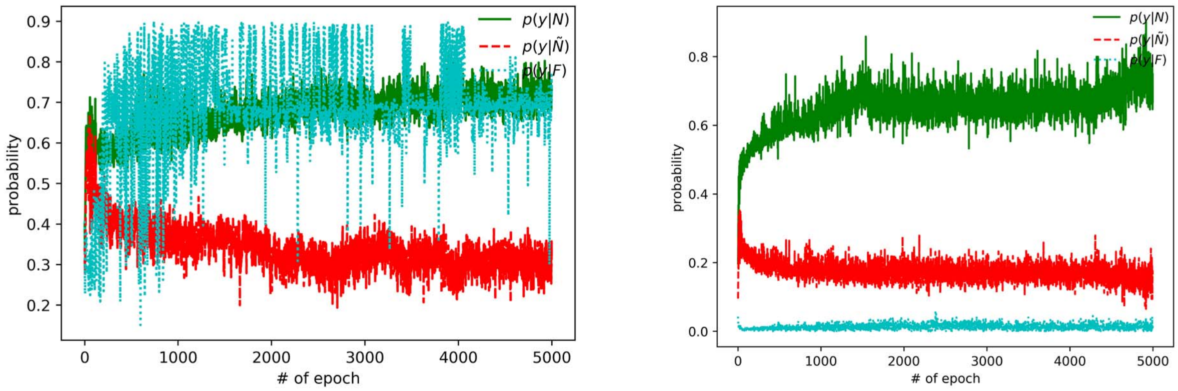

Fig. 6. Probabilites of being normal samples (output from complementary GAN) for real normal samples (green line), fault samples (cyan line), and generated samples (red line). For the right plot, 20% fault samples are used in the training set, while for the left plot, none of the fault samples is used in the training set.

[3] A. Zimek, E. Schubert, and H-P. Kriegel, “A survey on unsupervised outlier detection in high-dimensional numerical data,” Statistical Analysis and Data Mining, vol. 5, no. 5, pp. 363–387, 2012.

[4] L. Akoglu, H. Tong, and D. Koutra, “Graph-based anomaly detection and description: A survey,” Data Mining and Knowledge Discovery, vol. 29, no. 3, pp. 626–688, 2015.

[5] R. Chalapathy and S. Chawla “Deep learning for anomaly detection: A survey,” in arXiv preprint arXiv.1901.03407, 2019.

[6] V. Hodge and J.Austin, “A survey of outlier detection methodologies,” Artificial Intelligence Review, vol. 22, no. 2, pp. 85–126, 2004.

[7] A. Zaher, S. McArthur, D. Infield, and Y. Patel, “Online wind turbine fault detection through automated SCADA data analysis,” Wind Energy, vol. 12, no. 6, pp. 574–93, 2009.

[8] D. Tolani, M. Yasar, A. Ray, and V. Yang, “Anomaly detection in aircraft gas turbine engines,” Journal of Aerospace Computing, Information, and Communication, vol. 3, no. 2, pp. 44–51, 2006.

[9] F. Xue and W. Yan, ”Parametric model-based anomaly detection for locomotive subsystems,” in Proceedings of the 2007 International Joint Conference on Neural Networks (IJCNN), Orlando, FL, August 12-17, 2007.

[10] E. Ogbonnaya, H. Ugwu, and J. Theophilus-Johnson, “Gas turbine engine anomaly detection through Computer simulation technique of statistical correlation,” IOSR Journal of Engineering, vol. 2, no. 4, pp. 544–554, 2012.

[11] A. Arranz, A. Cruz, M. Sanz-Bobi, P. Riuz, and J. Coutino, “DADICC: Intelligent system for anomaly detection in a combined cycle gas turbine plant,” Expert System Applications, vol. 34, no. 4, pp. 2267–2277, 2008.

[12] B. Ferrell, “JSF Prognostics and Health Management,” in Proceedings of IEEE Aerospace Conference, March 6–13, Big Sky, MO, 1999.

[13] W. Yan,“One-class extreme learning machines for gas turbine combustor anomaly detection,” in Proceedings of 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016.

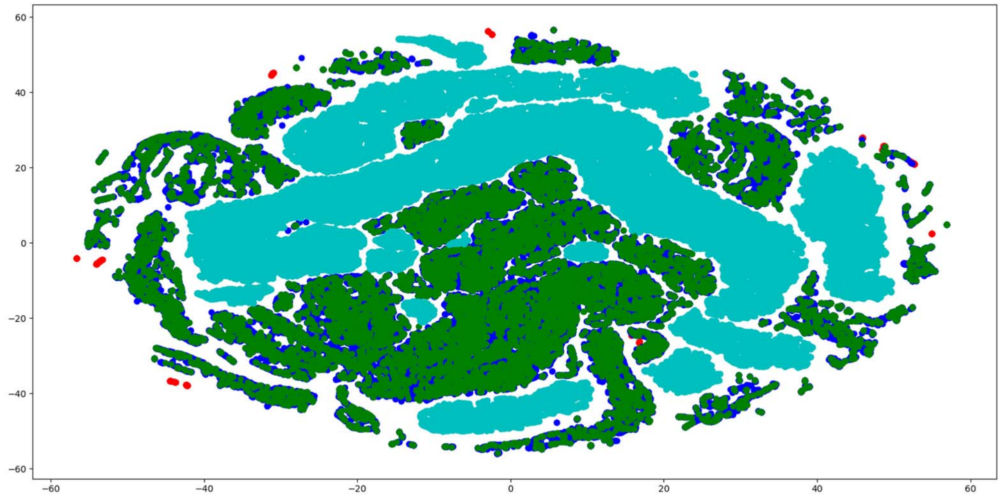

Fig. 7. t-SNE projection of the normal samples (blue points for training and green points for test), fault samples (red points), and generated samples from the fence GAN generator (cyan points).

[14] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” in Advances in Neural Information Processing Systems 27, pp. 2672-2680, 2014.

[15] I. Goodfellow, “NIPS 2016 tutorial: Generative adversarial networks,” in arXiv preprint arXiv.1701.00160, 2017.

[16] T. Schlegl, P. Seeböck, S. Waldstein, U. Schmidt-Erfurth, and G. Langs, “Unsupervised anomaly detection with generative adversarial networks to guide marker discovery,” Proceedings on International Conference on Informaiton Processing in Medical Imaging, pp. 146–157, 2017.H. Zenati, C. Foo, B. Lecouat, G. Manek, and V. Chandrasekhar, “Efficient GAN-based anomaly detection,” in arXiv preprint arXiv.1802.06222, 2018.

[17] H. Zenati, C. Foo, B. Lecouat, G. Manek, and V. Chandrasekhar, “Efficient GAN-based anomaly detection,” in arXiv preprint arXiv.1802.06222, 2018.

[18] J. Donahue, P. Krhenbhl, and T. Darrell, “Adversarial feature learning,” in 5th International Conference on Learning Representations, 2017.

[19] H. Zenati, M. Romain, C. Foo, B. Lecouat, and V. Chandrasekhar, “Adversarially learned anomaly detection,” In Proceedings of the 20th IEEE International Conference on Data Mining, 2018.

[20] S. Akcay, A. Atapour-Abarghouei, and T. Breckon, “GANomaly: Semi-supervised anomaly detection via adversarial training,” in Proceedings of Asian Conference on Computer Vision, pp. 622–637, 2018.

[21] Z. Dai, Z. Yang, F. Yang, W. Cohen, and R. Salakhutdinov, “Good semi-supervised learning that requires a bad GAN,” in Advances in Neural Information Processing Systems 30, pp. 6513-6523, 2017.

[22] P. Zheng, S. Yuan, X. Wu, J. Li, and A. Lu, “One-class adversarial nets for fraud detection,” in arXiv preprint arXiv.1803.01798, 2018.

[23] C. Ngo, A. Winarto, C. Kou, S. Park, F. Akram, and H. Lee, “Fence GAN: Towards better anomaly detection,” in arXiv preprint arXiv.1904.01209, 2019.

[24] F. Di Mattia, P. Galeone, M. De Simonti, and E. Ghelfi, “A survey on GANs for anomaly detection,” in arXiv preprint arXiv.1906.11632, 2019.

[25] C. Vondrick, H. Pirsiavash, and A. Torralba, “Generating videos with scene dynamics,” in Advances in Neural Information Processing Systems, pp. 613-621, 2016.

[26] L. Yang, S. Chou, and Y. Yang, “MIDINET: A convolutional generative adversarial network for symbolic-domain music generation,” in arXiv preprint arXiv.1703.10847, 2017.

[27] C. Allegorico and V. Mantini, “A data-driven approach for on-line gas turbine combustion monitoring using classification models”, in 2nd European Conference of the Prognostics and Health Management Society 2014, Nantes, France, July 8–10, 2014.

[28] L. van der Maaten and G. Hinton, “Visualizing high-dimensional data using t-SNE,” Journal of Machine Learning Research, 9(Nov), pp. 2579–2605, 2008.