Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series

Anomaly Detection with Generative Adversarial Networks for Multivariate Time Series

Dan Li, Dacheng Chen, Jonathan Goh, and See-Kiong Ng,

***Abstract*—Today's Cyber-Physical Systems (CPSs) are large, complex, and affixed with networked sensors and actuators that are targets for cyber-attacks. Conventional detection techniques are unable to deal with the increasingly dynamic and complex nature of the CPSs. On the other hand, the networked sensors and actuators generate large amounts of data streams that can be continuously monitored for intrusion events. Unsupervised machine learning techniques can be used to model the system behaviour and classify deviant behaviours as possible attacks. In this work, we proposed a novel Generative Adversarial Networks-based Anomaly Detection (GAN-AD) method for such complex networked CPSs. We used LSTM-RNN in our GAN to capture the distribution of the multivariate time series of the sensors and actuators under normal working conditions of a CPS. Instead of treating each sensor's and actuator's time series independently, we model the time series of multiple sensors and actuators in the CPS concurrently to take into account of potential latent interactions between them. To exploit both the generator and the discriminator of our GAN, we deployed the GAN-trained discriminator together with the residuals between generator-reconstructed data and the actual samples to detect possible anomalies in the complex CPS. We used our GAN-AD to distinguish abnormal attacked situations from normal working conditions for a complex six-stage Secure Water Treatment (SWaT) system. Experimental results showed that the proposed strategy is effective in identifying anomalies caused by various attacks with high detection rate and low false positive rate as compared to existing methods.**

***Index Terms*—Unseen Fault Identification (UFI), Heating, Ventilation, and Air-Conditioning (HVAC), Air Handling Unit (AHU), Smart Building**

I. Introduction

Cyber-Physical Systems (CPSs) are interconnected physical systems typically engineered for mission-critical tasks. Some example CPSs are water treatment and distribution plants, natural gas distribution systems, oil refineries, power plants, power grids, and autonomous vehicles. The emergence of the Internet of Things (IoT) will further drive the proliferation of CPSs for a large variety of tasks, resulting in many systems and devices communicating and operating autonomously over networks. As such, cyber-attacks are one of the most concerned potential threats to CPSs.

Traditionally, Statistical Process Control (SPC) methods such as CUSUM, EWMA and Shewhart charts are popular solutions for anomaly detection [1]. These conventional detection techniques are unable to deal with the increasingly dynamic and complex nature of the CPSs with the advent of IoT. As such, researchers have moved beyond specification or signature-based techniques and begun to exploit both supervised and unsupervised machine learning techniques to develop more intelligent and adaptive methods from big data to identify anomalies or intrusions [2].

However, even with the use of machine learning techniques, detecting anomalies in time series is still challenging. First, most of the supervised techniques require enough liable normal data and labelled anomaly classes to learn from but this is hardly the case in practice as anomalies are typically rare. Secondly, most of existing unsupervised methods are built through linear projection and transformation but there is often non-linearity in the hidden inherent correlations of the multivariate time series of complex CPSs. Most of the current techniques also employ simple comparison between present state and predicted normal ranges, which can be inadequate for anomaly detection since the control bounds are not flexible enough and cannot effectively identify indirect attacks[1].

To address these challenges, we propose a novel unsupervised GAN-based Anomaly Detection (GAN-AD) method for a complex multi-process CPS with multiple networked sensors and actuators by modelling the non-linear correlations among multiple time series and detecting anomalies based on the trained GAN model. Fig. 1 depicts the overall framework of our proposed GAN-AD. First, to deal with time-series data, the generator and discriminator are built as two Long-Short-Term Recurrent Neural Networks (LSTM-RNN), as shown in the left part of Fig. 1. As in a typical GAN, the generator (G) generates fake samples from a specific latent space and passes that to the discriminator (D) which tries to distinguish fake from real. Based on the outputs of D, the system will update parameters of D and G, so that the discriminator will be trained to be as sensitive as possible to assign correct labels to both real and fake samples, while the generator will be trained as smart as possible to fool the discriminator (assigning real labels to fake samples). After sufficient rounds of iterations, the generator will have captured the hidden distributions of the training sequences and that could generate realistic samples. In other words, G can be viewed as an implicit model of the CPS. At the same time, the resulting discriminator D is also able to distinguish fake from real with high sensitivity.

This paper is accepted by the 7th International Workshop on Big Data, Streams and Heterogeneous Source Mining: Algorithms, Systems, Programming Models and Applications on the ACM Knowledge Discovery and Data Mining conference, August 2018, London, United Kingdom

Dan Li , Dacheng Chen and See Kiong Ng are with the Institute of Data Science, National University of Singapore, 3 Research Link, Singapore 117602 (e-mails: idsld, idscd, seekiong@ntu.edu.sg).

Jonathan Goh is with the ST Electronics (Info Security) Pte Ltd. (email:goh.whyewei.jonathan@stee.stengg.com).

[1] For one specific variable (i.e. a sensor or actuator in the CPS), a ”direct attack” is defined as an attack that is directly inserted to it and affects its performance, while an ”indirect attack” is an intrusion targeted for another variable but also affects the performance of the variable.

In other words, D is an intuitive tool for anomaly detection. In this work, we propose to exploit both G and D for the anomaly detection task by (i) exploiting the residuals between real-time testing samples and reconstructed samples based on the mapping from real-time space to the GAN latent space; and (ii) discrimination with the machine-learned discriminator by classifying the real-time series. We depict this aspect of our proposed GAN-AD in the right part of Fig. 1. As shown, the testing samples are mapped back into the latent space, and the corresponding residual loss is calculated based on the difference between the reconstructed testing samples (by the trained generator) and the actual testing samples. At the same time, testing samples are also fed to the trained discriminator to compute the discrimination loss. The two losses are then combined to detect potential anomalies for sequential CPS data (more details are described in Section III-C).

The remaining part of this paper is organized as follows. Section II introduces the related works. Section III presents our proposed GAN-AD and derives an anomaly score function. Section IV introduces the multi-stage Secure Water Treatment system, which is followed by Section V in which we evaluate our proposed GAN-AD on real-time multivariate series. Finally, Section VI summarizes the whole paper and proposes possible future work.

II. Related Works

The basic task of anomaly detection is to identify whether the testing data conform to the normal data distribution; the non-conforming points are called anomalies, outliers, intrusions, failures or contaminants in various application domains [3], [2]. Anomaly detection is an old but challenging problem—it has been studied in the statistics community as early as the 19th century [3].

Based on how the historical training data is used, we can broadly divide anomaly detection methods into three categories: i) Statistical Process Control (SPC) techniques, ii) supervised machine learning methods, and iii) unsupervised machine learning methods. The SPC techniques were extensively used in the early years for monitoring and controlling quality of manufacturing processes through univariate or multivariate analysis [4]. The SPC approaches typically inspect changes in process mean (mean shifts) and process variance (variance changes), and try to model the relationship among multiple variables [5], [**?**]. Shewhart control charts and CUSUM control charts are univariate SPC techniques that are usually appied to detect mean shifts [6], [7]. EWMV control charts are sensitive to univarite variance changes as well as mean shifts [8]. Although widely used, the aforementioned SPC techniques cannot model the correlation among various sequences, while interrelations are common even in traditional manufacturing systems. As an improvement, Hotelling's T_2 [9], Multivariate Cumulative Sum (MCUSUM) [10] and Multivariate Exponentially Weighted Moving ASverage (MEWMA) [11] were proposed to monitor the performance of multiple variables in a manufacturing system. However, these multivariate methods require independent and identically distributed (iid) assumption which is often violated in reality [12]. Moreover, the computationally intensive multivariate SPC methods are not practical for modern CPSs with high complexity and massive data streams.

Supervised machine learning techniques can also be used for anomaly detection. A typical approach is to build a predictive classification model for the normal and the anomalous classes. The classification model is trained from the labelled data. New measurements can then be analysed by the classifier and be classified to corresponding categories (normal or anomalous) automatically [13]. A wide range of supervised machine learning tecniques has been used for anomaly detection. They include Multivariate Regression models [14], Bayes Classifier [15], Neural Networks (NN) [16], Fisher Discriminant Analysis (FDA) [17], Gaussion Mixture Model [18], Support Vector Data Description (SVDD) [19], Support Vector Machines (SVM) [20], and tree-structured learning method [21], [22]. However, supervised classification methods are dependent on the availability of initial labelled training data. Given that anomalies are typically rare, obtaining enough accurate labelled anomaly classes is usually challenging.

The unsupervised learning methods—also known as descriptive or undirected classification—train models without lablled classes. Due to their simplicity and ability to handle large number of process data, unsupervised learning methods have been wildly used for anomaly detection for various industrial processes [23]. Popular unsupervised methods include Principal Component Analysis (PCA) [24] and Partial Least Squares (PLS) [25]. PCA is a multivariate data analysis method which preserves the significant variability information extracted from the process measurements and reduces the dimension for huge amount of correlated data [26]. PLS is another multivariate data analysis method that has been extensively utilized for model building and anomaly detection [27]. Their key performance indicators (Square Predicted Error (SPE) for PCA and T^2 index for PLS) for anomaly detection can be achieved with correlation model traine off-line and online process measurements. However, these unsupervised methods are only effective to highly correlated data, and require the data to follow multivariate Gaussian distribution [28].

The recently proposed GAN framework enables researchers to build a generative model via adversarial training [29]. The simultaneous training of a generator and a discriminator in an adversarial fashion is highly suggestive for using the GAN framework for anomaly detection. The current successes of GANs are mainly in generating realistic-looking images. In our CPS and IoT scenarios, we have to deal with oftentimes multiple streams of potentially interacting time series data. However,there has been limited work in adopting the GAN framework for time-series data todate. To the best of our knowledge, there are two preliminary work using GAN to generate continuous valued sequences in the literature—one to produce polyphonic music with recurrent neural networks as generator and discriminator [30], and the other uses conditional version of recurrent GAN to generate real-valued medical time series [31]. In both of these works, the multi-sequences were treated as i.i.d. and fed to a uniform GAN framework. This will be inadequate for the IoT and CPSs

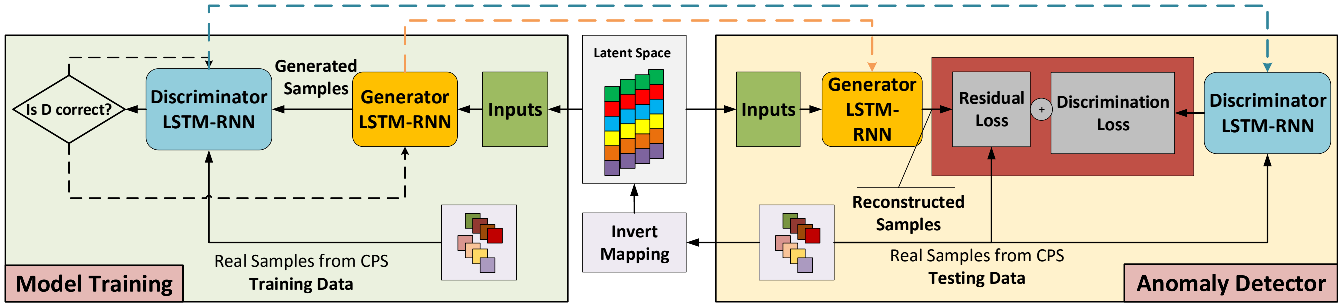


Fig. 1: GAN-AD: Unsupervised GAN-based anomaly detection for CPSs. On the left is a GAN framework in which the generator and discriminator are obtained with iterative adversarial training. On the right is the exploitation of both the GAN's generator and discriminator for anomaly detection—the generator is used for computing the residual loss between reconstructed samples and real ones, while the discriminator is used to compute the discrimination loss.

setting given the potential interactions amongst the multiple time series-generating sensors and actuators involved in the same or different processes of the complex systems.

Unlike traditional classification methods, the GAN-trained discriminator learns to detect false from real in an unsupervised fashion, making GAN an attractive unsupervised machine learning technique for anomaly detection [32]. In addition, the GAN framework also produces a generator which is actually an inexplicit model of the target system with its ability to output normal samples from a certain latent space. Inspired by [33] and [34] that updates a mapping from the real-time space to a certain latent space to enhance the training of generator and discriminator, researchers have recently proposed to train a latent space understandable GAN and apply it for unsupervised learning of rich feature representations for arbitrary data distributions. [35] and [36] showed the possibility of recognizing anomalies with reconstructed testing samples from latent space, and successfully applied the proposed GAN-based detection strategy to discover unexpected markers for images. In this work, we will leverage on these previous works to make use of both the GAN-trained generator and discriminator to better detect anomalies based on both residual and discrimination losses.

Our contributions of this paper are summarized as follows: i), a novel GAN-based unsupervised anomaly detection method is proposed to detect anomalies (cyber-attacks) for complex multi-process cyber-phsyical systems with networked sensors and actuators; ii), the GAN model is trained with multiple time series, which adapts GAN from the image generation domain for time series generation by adopting the Long Short Term-Recurrent Neural Networks (LSTM-RNN) to capture the temporal dependency; iii), normal sequences with high dimension is uniformly utilized to train the GAN model to discriminate fake from real and reconstruct testing sequences from specific latent space simultaneously; iv), the discrimination loss calculated by the trained discriminator and the residual loss between reconstructed and real testing sequences (to make use of both the trained discriminator and generator) are combined together to detect anomalous points in the high dimensional time series, and the proposed method is shown to outperform existing methods in detecting anomalies due to cyber attacks in a complex Secure Water Treatment (SWaT) system with six stages [37].

III. Anomaly Detection with Generative Adversarial Training

A. GAN with LSTM-RNN

Long Short Term-Recurrent Neural Networks (LSTM-RNN) had been shown to be capable of learning complex time series by taking the information in backward (or even forward) time steps with memorise cells. In this work, in order to handle time series data of the CPS, both the generator (G) and discriminator (D) of GAN are substituted by LSTM-RNN. Following the architecture of a regular GAN framework [29], the GAN model is trained as a two-player minimax game.

$$\begin{aligned}\min_G \max_D V(D,G) = \mathcal{E}_{x \backsim p_{data}(x)}\left[\log D(x)\right] \\ +\mathcal{E}_{z \backsim p_z(Z)}\left[\log(1-D(G(z)))\right]\end{aligned} \tag{1}$$

Specifically, the generator G, a LSTM-RNN model, implicitly defines a probability distribution for the generated samples, which can be written as $G_{rnn}(z)$, where z is a distribution from the random latent space. The discriminator, which is another LSTM-RNN model, is then trained to minimise the average negative cross entropy between its predictions and sequence labels (e.g., train D to recognize as many training samples as real as possible, and recognize as many generated samples as false as possible). Thus, the discriminator loss is

$$\begin{aligned}D_{loss} = \tfrac{1}{m}\textstyle\sum_{i=1}^{m}\left[\log D_{rnn}(x_i) + \log(1 - D_{rnn}(G_{rnn}(z_i)))\right] \\ \Leftrightarrow \min \tfrac{1}{m}\textstyle\sum_{i=1}^{m}\left[-\log D_{rnn}(x_i) - \log(1 - D_{rnn}(G_{rnn}(z_i)))\right]\end{aligned} \tag{2}$$

where $x_i, i = 1, ..., m$ is the training samples which should be recognized as real, and $G_{rnn}(z_i), i = 1, ..., m$ is the generated samples that should be recognized as false.

At the same time, the generator is trained to confuse the discriminator so that the discriminator would recognize as many generated samples as real as possible. In other words, the generator loss is

$$\begin{aligned}G_{loss} = \textstyle\sum_{i=1}^{m} \log(1 - D_{rnn}(G_{rnn}(z_i))) \\ \Leftrightarrow \min \textstyle\sum_{i=1}^{m} \log(-D_{rnn}(G_{rnn}(z_i)))\end{aligned} \tag{3}$$

The generator loss and discriminator loss are jointly dealt with by optimizers and used to update the parameters for G_{rnn} and D_{rnn}.

Algorithm 1 LSTM-RNN-GAN-based Anomaly Detection Strategy

```
loop
  if epoch within number of training iterations then
    for the kth epoch do
      Generate samples from the latent space:
      $Z = \{z_i, i = 1, ..., m\} \Rightarrow G_{rnn}(Z)$
      Conduct discrimination:
      $X = \{x_i, i = 1, ..., m\} \Rightarrow D_{rnn}(X)$
      $G_{rnn}(Z) \Rightarrow D_{rnn}(G_{rnn}(Z))$
      Update discriminator parameters by minimizing(descending) $D_{loss}$:
      $\min \frac{1}{m}\sum_{i=1}^{m}[-\log D_{rnn}(x_i) - \log(1 - D_{rnn}(G_{rnn}(z_i)))]$
      Update discriminator parameters by minimizing(descending) $G_loss$ :
      $\min \sum_{i=1}^{m} \log(-D_{rnn}(G_{rnn}(z_i)))$
      Record parameters of the discriminator and generator in the current iteration.
    end for
  end if
  for the lth iteration do
    Mapping testing data back to latent space:
    $Z^k = \min_{Z} Er(X^{tes}, G_{rnn}(Z^i))$
  end for
  Calculate the residuals:
  $Res =\mid X^{tes} - G_{rnn}(Z^k) \mid$
  Calculate the discrimination results:
  $Pro = D_{rnn}(X^{tes})$
  Obtain anomaly score:
  $S = Res + Pro$
end loop
```

B. GAN-based Anomaly Score

As a newly arisen unsupervised learning method, both the generator and generator of GAN are jointly trained to represent the normal anatomical variability which is helpful for identifying anomalies. To make full use of the GAN model, both the trained generator and discriminator should be driven to make contributions to the anomaly detection. Following the formulation in [35], the anomaly detection for CPSs time series data consists of the following two parts.

- **Anomaly Detection with Discrimination**
  Intuitively, the trained discriminator D (after a sufficient number of iterations of adversarial training) is a direct tool for anomaly detection since it can distinguish fake from real with high sensitivity.
- **Anomaly Detection with Residuals**
  As mentioned in previous sections, the trained generator G, which is capable of generating realistic samples, is actually a mapping from the latent space to real data space: $G(Z) : Z \rightarrow X$, and can be viewed as an inexplicit system model that reflects the normal data's distribution. Due to the smooth transitions of latent space mentioned in [38], the generator outputs similar samples if the inputs in the latent space are close. Thus, if it is possible to find the corresponding Z^k in the latent space for the testing data X^{tes}, the similarity between X^{tes} and $G(Z^k)$ could explain to which extent is X^{tes} follows the distribution reflected by G. That is to say, residuals between X^{tes} and $G(Z^k)$ could be utilized for identifying anomalies in testing data.

As shown in the right part of Fig. 1, to find the optimal Z^k that corresponds to the testing samples, we first sample a random set Z^1 from the latent space and obtain reconstructed raw samples $G(Z^1)$ by feeding it to the generator. Then, the samples from the latent space could be updated with the gradients obtained from the error function defined with X^{tes} and $G(Z)$.

$$\min_{Z^k} Er(X^{tes}, G_{rnn}(Z^k)) = 1 - Simi(X^{tes}, G_{rnn}(Z^k)) \tag{4}$$

where the similarity between sequences could be defined as covariance for simplicity.

If after enough iteration rounds the error is small enough, the samples Z^k is recorded as the corresponding mapping in the latent space for the testing samples. Thus, the residual at time t for testing samples is calculated as

$$Res(X_t^{tes}) = \sum_{i=1}^{n} \mid x_t^{tes,i} - G_{rnn}(Z_t^{k,i}) \mid \tag{5}$$

where $X_t^{tes} \subseteq \mathcal{R}^n$ is the measurements at time step t for n variables. In summary, the the anomaly score for anomaly detection is

$$S_t^{tes} = \lambda Res(X_t^{tes}) + (1 - \lambda) D_{rnn}(X_t^{tes}) \tag{6}$$

Our GAN-based unsupervised anomaly detection strategy is summarized in Algo. 1. Mini-batch stochastic optimization based on Adam Optimizer and Gradient Descent Optimizer is used for updating the model parameters.

C. Anomaly Detection Framework

We formulate the anomaly detection problem for multivariate time series as follows. First, consider an m-dimensional time series $X = \{x^{(t)}, t = 1, ..., T\}$ with length T [2], where $x^{(t)} \in \mathcal{R}^m$ is an m-dimensional vector of readings for m variables at time-instance t. Usually, in industry process or mechanical systems (such as the SWaT system considered in this paper), sensor measurements are large time-series with length $\mathcal{T}$ ($\mathcal{T} >> T$). Thus, multiple predefined time-series, $\mathbf{X} = \{X^{(1),...,X^{(L)}}\}$, can be obtained by taking a window of length T over the raw data streams. The GAN model is trained based on the normal time-series dataset $\mathbf{X}^{real}$, and generates "fake" samples $\mathbf{X}^{gs}$ that "look real". Next, the testing time-series dataset $\mathbf{X}^{att}$ (or $\mathbf{X}^{tes}$), which is real-time CPSs data,

[2] Usually, T should not be too large with purpose of monitoring and anmaly detection.

can be analysed by the trained model to detect anomalous slots.

However, the use of LSTM-RNN with high-dimensional inputs ($\mathbf{X} \subseteq \mathcal{R}^{51}$ in the SWaT case) incurs higher computational cost than usual deep neural networks. Thus, in this paper, we adopt the Principal Component Analysis (PCA) to project the high-dimensional data into a PC projection space before feeding the data to the GAN model: $\mathbf{X}^{tes} \subseteq \mathcal{R}^{m} \Rightarrow \mathcal{X}^{tes} \subseteq \mathcal{R}^{n}$. The projection is

$$\begin{aligned} P &= PCA(\mathbf{X}^{real}) \\ \mathcal{X}^{tes} &= \mathbf{X}^{tes} P^{T} \end{aligned} \tag{7}$$

where $\mathbf{X}^{tes} \subseteq \mathcal{R}^{m}$, $\mathcal{X}^{tes} \subseteq \mathcal{R}^{n}$, $P \subseteq \mathcal{R}^{n \times n}$, m is the original dimension (namely the number of system variables) and n is number of reserved principal components. Then the projected variables are fed to the GAN-AD model and output anomaly scores according to Eq. (6). Next, the following label assigning function could be applied to identify whether the i^{th} variable of the testing time-series set $\mathcal{X}^{tes}$ at time step t is being attacked or not.

$$A_{t}^{tes,i} = \begin{cases} 1, & if\ H\left(S(x_{t}^{tes,i}), 1\right) > \tau \\ 0, & else \end{cases} \tag{8}$$

where $t = 1, ..., T,\ i = 1, ..., n$. An anomaly is detected if the cross entropy error $H\left(.,.\right)$ for the anomaly score is higher than a certain value τ.

IV. SWaT System and Cyber-attacks

A. Water Treatment System

The Secure Water Treatment (SWaT) system is an operational test-bed for water treatment that represents a small-scale version of a large modern water treatment plant found in large cities [39] [3].

The water purification process in SWaT is composed of six sub-processes referred to as $P1$ through $P6$ [37]. The first process is for raw water supply and storage, and $P2$ is for pre-treatment where the water quality is assessed. Undesired materials are them removed by ultra-filtration (UF) backwash in $P3$. The remaining chorine is destroyed in the Dechlorination process ($P4$). Subsequently, the water from P_4 is pumped into the Reverse Osmosis (RO) system ($P5$) to reduce inorganic impurities. Lastly, $P6$ stores the water ready for distribution.

B. Cyber-Attacks

Various experiments have been conducted on the SWaT system to investigate cyber-attacks and respective system responses. Please refer to the SWat website [4] for a detailed description of the attacks. A total of 36 attacks were launched during the 2016 SWaT data collection process [37]. Generally, the attacked points include sensors (e.g., water level sensors, flow-rate meter, etc.) and actuators (e.g., valve, pump, etc.). The summary of attacked points based on attack location and type is shown in Table I.

TABLE I: List of Cyber-attacks Inserted to the SWaT System

Process	Type	Attacked sensors	Attack Actuators
P1	SSSP	LIT-101	MV-101; P-101; P-102
	SSMP	(LIT-101 and MV-101)	
P2	SSSP	AIT-202	
	SSMP	(P-203 and P-205)	
	SSMP	(P-201, P-203 and P-205)	
P3	SSSP	LIT-301; DPIT-301	MV-303;MV-303; MV-304; P-302
P4	SSSP	LIT-401; FIT-401	
P5	SSSP	AIT-504	MV-504
	SSMP	(P-501 and FIT-502)	
P1-6	MSMP	(UV-401, AIT-502 and P501)	
	MSMP	(P-602, DIT-301 MV-301)	
	MSMP	(P-302 and LIT-401)	
	MSMP	(LIT-101, P-101 and MV-201)	
	MSSP	(AIT-402 and AIT-502)	
	MSSP	(FIT-401 and AIT-502)	
	MSSP	(P-101 and LIT-301)	

*FIT-flower meter; LIT-water level transmitter
*MV-motorized valve
*P-water pump/dosing pump/Sodium bi-sulphate pump
*AIT-chemical analyser; UV-dechlorinator meter
*DPIT-differential pressure indicating transmitter
*SSSP: single stage single point attack
*SSMP: single stage multi point attack
*MSMP: multi stage multi point attack
*MSSP: multi stage single point attack

As a test-bed for research in the area of cyber security, several related works have been published based on the SWaT dataset. Some of them focused on special attacks. For example, a distributed detection method for single stage multiple points attacks via system specific physical invariants is proposed in [40]. Also, Jonathan et al proposed to find attacks for the first process via RNN prediction and CUSUM detection [41]. A model-based method, which derives a Kalman filter, was applied to estimate the evolution of the system dynamics on single variable basis [42]. In this work, we consider all the aforementioned cyber-attacks as anomalous working conditions and train our proposed GAN-AD method to detect these anomalies for all the six processes in SWaT (results are shown in Section V-E).

C. SWaT Dataset

The 2016 SWaT data collection process lasted for a 11 days with the system operated 24 hours per day. Various cyber-attacks were implemented on the testbed with different intents and divergent lasting durations (from a few minutes to an hour) in the final four days. The system was either allowed to reach its normal operating state before another attack was launched or the attacks were launched consecutively. The 2016 SWaT dataset and its associated attacks have the following characteristics [5]

- Different attacks may last for different time durations due to different scenarios. Some attacks do not even take

[3] The overall testbed design was coordinated with Singapore's Public Utility Board, the nation-wide water utility company, and constructed by a third party vendor. That collaboration ensured that the overall physical process and control system closely resemble real systems in the field, so that the results can be applied to real systems as well. For more information, please refer to https://itrust.sutd.edu.sg/research/testbeds/secure-water-treatment-swat/

[4] http://itrust.sutd.edu.sg/research/dataset

[5] The raw data are not plotted in this paper due to page limit—please refer to [37] for more information.

effect immediately. The system stabilization durations also vary across attacks. Simpler attacks, such as those aiming at changing flow rates, require less time for the system to stabilize while the attacks that caused stronger effects on the dynamics of system will require more time for stabilization.

- Attacks on one sensor (or actuator) may affect the performance on other sensors (or actuators), usually after a certain time delay.
- Furthermore, similar types of sensors (or actuators) tend to respond to attacks in a similar fashion. For example, attacks on the LIT101 sensor (a water level sensor in process 1) cause abnormal spikes in both LIT101 and LIT 301 (another water level sensor in process 3) but do not affect the readings of other sensors and actuators (such as flow rate sensor and power meter) in the system.

The aforementioned observations suggest we should take a multivariate approach in the modelling instead of taking each sensor or actuator in the CPS as an independent data source (univariate approach). The underlying correlations between the sensors and actuators could be exploited to better detect anomalies in the system.

V. Experiments

A. Data Pre-processing

In the 2016 SWaT dataset, 51 variables (sensor readings and actuator states) were measured for 11-days. Within the raw data, $496,800$ samples were collected under normal working conditions (data collected in the first 7-days), and $449,919$ samples were collected when various cyber-attacks were inserted to the system. We eliminate the first $21,600$ samples from the training dataset since it took 5-6 hours to reach stabilization when the system was first turned on, according to [37].

We subdivide the original long multiple sequences into smaller time series by taking window across raw streams. Since the SWaT data were recorded every second, we set the window length as T=120 (i.e. data collected within 2 minutes). To capture the relevant dynamics of SWaT data, the window (T=120) is applied to the normal and testing datasets with shift length SL_{nor}=10 for normal dataset and SL_{att}=120 for testing dataset respectively. In order to speed up the GAN training process and avoid over-fitting, the samples were down-sampled to one measurement every 10 seconds by taking the median value. As a result, we obtained $47,508$ training samples and $3,720$ testing samples with sequence length L=12.

B. System Architecture

For this study, we used an LSTM network with depth 3 and 100 hidden (internal) units for the generator. The LSTM network for the discriminator is relatively simpler with 100 hidden units and depth 1. Inspired by the discussion about latent space dimension in [31], we also tried different dimensions and found that higher latent space dimension generally generates better samples especially when generating multivariate sequences. Thus, we set the dimension of latent space as 15 in this study.

C. Sample Generation

First, we visualize the data samples generated by our GAN versus the actual samples from the CPS. As can be observed in Fig. 2, our GAN generated samples that were clearly different from the training data in the early learning stage. However, after sufficient number of iterations, the generator is able to output realistic samples for the various sensors and actuators in the system.

We use Maximum Mean Discrepancy (MMD) to evaluate whether the GAN model has learned the distributions of the training data. MMD is one of the training objectives for moment matching networks.

$$\begin{aligned} MMD(Z_j, X_{tes}) = \tfrac{1}{n(n-1)} \textstyle\sum_{i=1}^{n} \sum_{j\neq i}^{n} K(Z_i^k, Z_j^k) \\ -\tfrac{2}{mn} \textstyle\sum_{i=1}^{n} \sum_{j=1}^{m} K(Z_i^k, X_j^{tes}) \\ +\tfrac{1}{m(m-1)} \textstyle\sum_{i=1}^{m} \sum_{j\neq i}^{m} K(X_i^{tes}, X_j^{tes}) \end{aligned} \tag{9}$$

We plot the MMD values across GAN training iterations for generating univariate samples and multivariate samples in Fig. 3. In both cases, we can observe that the MMD values tend to converge to small values after 20-30 iterations. Interestingly, the early MMD values of multivariate samples were lower than that of univariate samples, and the MMD for multivariate samples also converged faster than the univariate case. This suggests that using multiple data streams can help with the training of GAN model. Fig. 4 shows the performance of our GAN in generating both univariate and multivariate samples. As can be seen, after more than 50 iterations, the GAN model could generate realistic time sequences even for the multivariate case.

D. Evaluation Metric

We use the following metrics, namely Accuracy (Accu), Precision (Pre), Recall (Rec), F_1 score, and False Positive Rate (FPR) to evaluate the anomaly detection performance of GAN-AD.

$$Accu = \frac{TP+TN}{TP+TN+FP+FN} \tag{10}$$

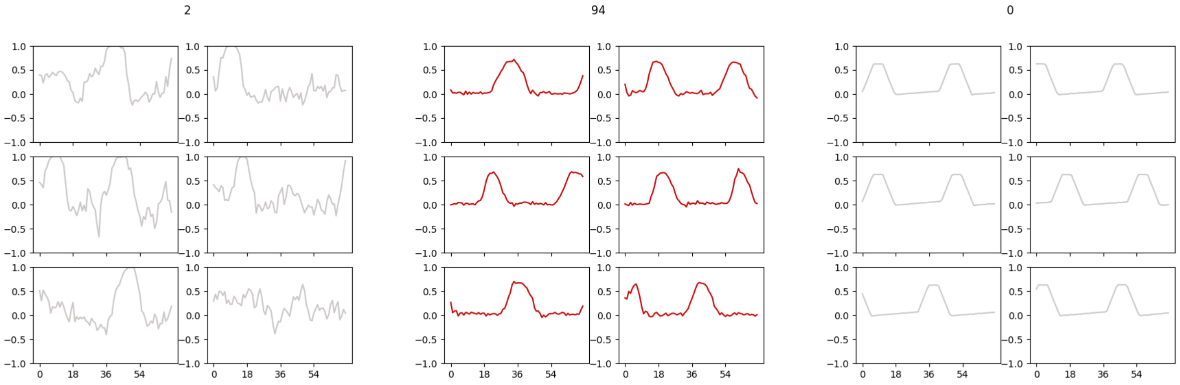
$$Pre = \frac{TP}{TP+FP} \tag{11}$$

$$Rec = \frac{TP}{TP+FN} \tag{12}$$

$$F_1 = 2 \times \frac{Pre \times Rec}{Pre+Rec} \tag{13}$$

$$FPR = \frac{FP}{FP+TN} \tag{14}$$

where TP is the correctly detected anomaly ($A_t = 1$ while real label $L_t = 1$), FP is the falsely detected anomaly ($A_t = 1$ while real label $L_t = 0$), TN is the correctly assigned normal ($A_t = 0$ while real label $L_t = 0$), and FN is the falsely assigned normal ($A_t = 0$ while real label $L_t = 1$).



(a) Generated samples at iteration=2. (b) Generated samples at iteration=95. (c) Original samples.

Fig. 2: Comparison between generated samples at different traning stages: GAN-generated samples at early stage are quite random while those generated at later stages almost perfectly took the distribution of original samples.

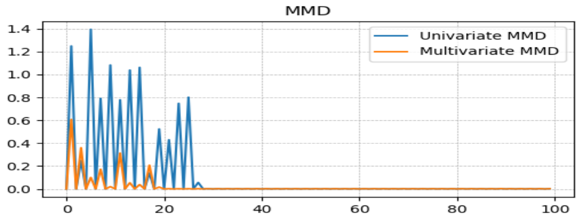


Fig. 3: MMD: generation for multiple time series v.s. generation single time series.

TABLE II: Anomaly Detection Rates for All Attacks by Checking Different Combinations of Measuring Points

Point	Method	Accu	Pre	Rec	F_1	FPR
LIT-101	CUSUM	86.63	36.42	26.86	0.30	7.39
	GAN-AD	87.63	50.00	1.75	0.03	9.32
P-101	CUSUM	75.52	24.62	43.97	0.31	19.83
	GAN-AD	80.72	22.24	15.43	0.18	8.71
AIT-202	CUSUM	55.67	9.24	25.15	0.13	39.45
	GAN-AD	60.22	4.58	17.10	0.07	35.48
LIT-301	CUSUM	81.50	12.92	9.02	0.10	8.43
	GAN-AD	86.85	22.22	1.04	0.02	0.52
DPIT-301	CUSUM	84.13	18.46	17.14	0.17	8.41
	GAN-AD	84.40	25.00	2.67	0.05	1.18
MV-303	CUSUM	71.55	9.67	19.18	0.12	22.01
	GAN-AD	87.68	17.54	3.00	0.05	1.74
LIT-401	CUSUM	88.28	47.80	58.53	0.52	7.99
	GAN-AD	80.35	11.68	9.94	0.10	10.14
FIT-401	CUSUM	12.90	12.90	100.00	0.23	100
	GAN-AD	85.40	39.36	4.32	0.08	1.09
AIT-504	CUSUM	70.97	6.23	14.38	0.08	23.01
	GAN-AD	86.03	14.74	14.35	0.14	11.14
All	SPE^1	87.24	20.49	2.25	0.04	1.19
	SPE^5	82.81	24.92	21.63	0.23	8.87
	GAN-AD^1	90.57	85.71	7.20	0.13	**0.13**
	GAN-AD^5	**94.80**	**93.33**	**63.64**	**0.75**	0.46

$*^1$ means only one principal component is chosen
$*^5$ means the first five principal components are chosen

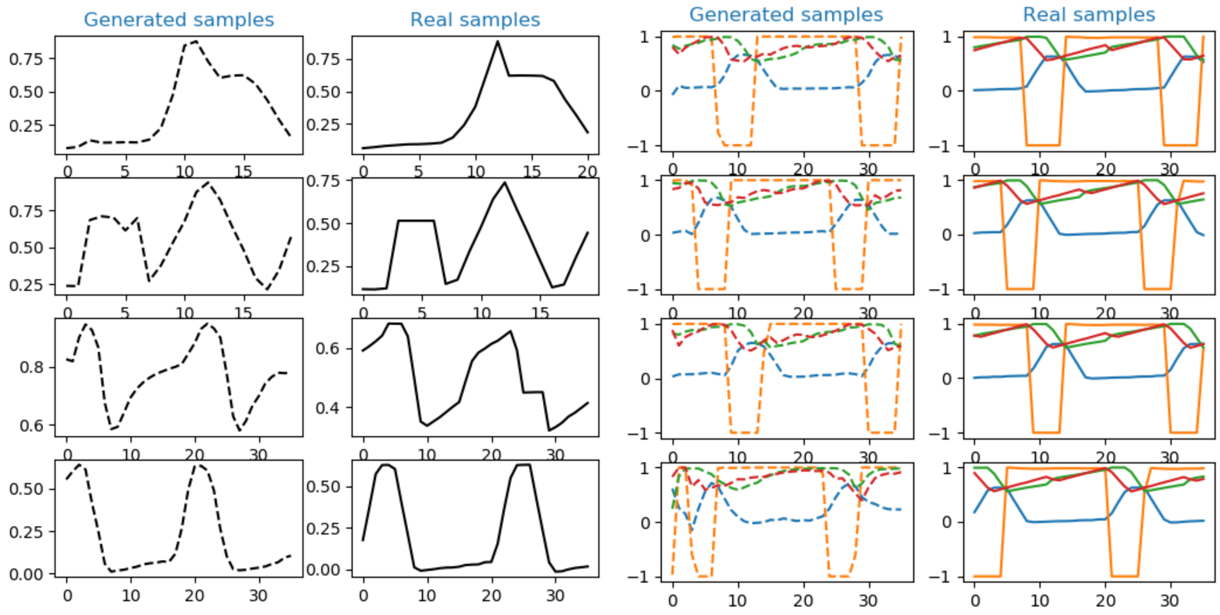
E. Anomaly Detection Results

We evaluate the anomaly detection performance for both the univariate and multivariate cases. In the univariate case, we compare the performance against the CUSUM approach which was used in previous works such as [41]. For the multivariate case, we compare GAN-AD against PCA-based unsupervised detection by inspecting the Squared Predicted Error (SPE, i.e., residual distance). The anomaly detection results and comparisons are summarized in Table II. We showed the results on 9 variables (i.e. sensors/actuators) for discussion. The 9 variables include sensors and actuators from different processes of the system. They were also attack points for single-point attacks.

To evaluate the major contribution of this work (i.e. detection of anomalies for multiple sequences), all the measured variables are fed uniformly into the GAN-AD model (as mentioned previously, to decrease the computational load, the 51-dimensional data is projected to a lower dimensional space with the help of PCA). Since PCA is also a popular unsupervised multivarite anomaly detection method by inspecting the Squared Predicted Error (SPE, i.e., residual distance) values, we also evaluate SPE-based detection against our proposed GAN-AD method.

1) Univariate: As shown in Table II, univariate detection generally perform poorly with high false positive rate, and low precision and recall. We also observed that:

- The water level sensors (e.g., LIT-101, LIT-301 and LIT-401) have the best anomaly detection performance—both CUSUM and GAN-AD were able to achieve good detection accuracy rate (near or above 80%). This is because other than the direct attacks on those sensors, most of attacks applied to the other sensors (or actuators) also affected the water levels indirectly.
- The accuracy values of AIT-504 were 70.97% and 86.03% (CUSUM and GAN-AD respectively) were better than its earlier counterpart AIT-202 (55.67% and 60.22%). This was because the chemical accumulation in $P5$ (where inorganic impurities are reduced) is relevant for anomaly detection after the chorine has been added to the system in $P3$ and $P4$. $P2$ was just a pre-treatment process where the quality of raw water for $P1$ is assessed.
- Detection results with actuators P-101 and MV-303 were not as poor as expected. Despite of the $0/1$ values for on/off (open/close) states, both CUSUM and GAN-AD managed to achieve accuracy of up to $71.55\% - 75.52\%$



(a) Univariate.

(b) Multivariate.

Fig. 4: Visualization of generated samples and original samples for both univariate and multivariate cases. Note that in the multivariate case 4 variables are fed to the GAN model simultaneously.

and $80.72\% - 87.68\%$ respectively. While the $0/1$ states do not provide variance across numerical dimension, the frequency of how often "0" and "1" appears along the time line was still useful for anomaly detection.

- For the water flow-rate sensor FIT-401, the results by CUSUM was extremely poor while GAN-AD performed well. One possible reason for CUSUM's 100% recall and false positive rate was that CUSUM did not assign any negative labels to the testing samples (i.e., recognizing all samples as anomalies) due to unsuitable normal ranges. On looking closely at the normalized raw data, we observed that the values of flow rate meter took a roughly $0/1$ shape (just like the actuator states). However, the flow rate at the high points were not static 1s but varied with high frequency. CUSUM is unable to capture such "bi-variant" characteristic and hence performed badly in this case. On the other hand, GAN-AD was able to handle this and generated acceptable accuracy rate.

2) Multivariate: A key contribution of this work is applying our proposed GAN-AD method to solve the multivariate anomaly detection problem for time series data. For dimensional reduction, instead of directly feeding the high dimensional data to the GAN-AD model, we used PCA to project the raw data into a lower dimensional principal space, as described in Eq. (7).

We plot the variance rate of the first 10 Principal Components (PC) in Fig. 5. As shown in the figure, there is one main PC that explained more than 50% of the variance for the SWaT data. Also, the PCs after the 5^{th} one contribute little to the overall variance (near to 0). As such, we projected the SWaT data to the most variant PC (the first one) as well as the first 5, and then applied the GAN-AD to detect anomalies for the projected data. For comparison, we also performed standard PCA-based anomaly detection by inspecting the testing dataset with the Squared Predicted Error (SPE, i.e., the residual distances calculated by PCA projection) method.

The bottom part of Table II shows the performance of multivariate anomaly detection using SPE and our proposed GAN-AD. The results showed about $3\% - 12\%$ improvement with the proposed GAN-AD. The GAN-AD method also achieved 50%-60% higher precision and 5%-40% higher recall compared with SPE by assigning more true positives (correctly detected anomalies).

We also compared both GAN-AD and SPE based on PC=1 and PC=5. That is, we conducted SPE with the first one and five principal components, while the raw data were projected to the first one and five principal components correspondingly before being fed into the GAN-AD. It is interesting to see that for both GAN-AD and SPE the recall rates with PC=5 (hold more than 90% variate rate) were obviously higher that with PC=1 (which only contains around 50% variate rate as shown in Fig. 5), which implies that using more principal components could reduce false negatives.

In terms of false positive rates, GAN-AD with PC=1 achieved the best FPR amongst all (both univariate and multivariate). Although GAN-AD with PC=5 outperformed others in the aspect of accuracy, precision, recall, and F_1, its false positive rate was slightly higher than that by GAN-AD with PC=1. Similar phenomenon could be observed in multivariate detection by SPE. This indicates that the improvement of detection accuracy (as well as precision and recall) was built upon the sacrifice of more false positives due to the noisy information brought in by adding four more less important PC dimensions.

The results in Table II also showed that:

- Generally, the univariate detection cannot compete with

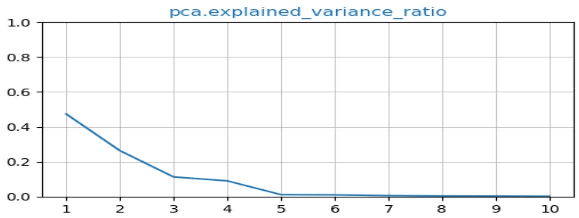


Fig. 5: Variance Ratio of Principal Component for the SWaT data.

multivariate detection. To be specific, univariate detection results in widespread low precision and recall (note that multivariate detection by SPE also performed poorly in terms of these two factors, which we will discuss in the next point), and high FPR. This observation demonstrates the multivariate detection is applicable for complex CPSs with interconnected IoT of sensors and actuators generating large amounts of time series.

- In fact, even the baseline multivariate anomaly detection method SPE can compete with most of the univariate detection results in terms of accuracy,. However, SPE does not compare well with univariate detection for LIT-101, DPIT-301, LIT-401 and FIT-301 in terms of precision. As noisy information can be accumulated when simply projecting the whole raw dimensions, it can sometimes cause negative effect in assigning true positives [43]. One possible future work would be to consider selecting suitable variables and appointing different weights according to their importance levels, instead of treating all the variables uniformly in one plain framework as in the current study.

VI. Conclusions

Cyber-Physical Systems are large, complex, and affixed with networked sensors and actuators that generate large amounts of data streams. These data streams and their underlying system dependencies can potentially be mined for dynamic detection of possible intrusion incidents. In this paper, we have explored the use of GAN to simultaneously train a deep learning network to model the distributions of multi-sensor data streams in a CPS under normal operating conditions, and another to detect anomalies due to cyber attacks being carried out against the CPS in an unsupervised fashion. We have proposed a novel GAN-based Anomaly Detection (GAN-AD) method that directly utilizes both the discriminator and the generator trained on multivariate time series to detect anomalies. We have tested our approach on a complex CPS dataset from a Secure Water Treatment Testbed (SWaT) and showed that the proposed GAN-AD was able to outperform existing unsupervised detection methods.

For future work, we will explore the use of GAN-AD for other IoT applications such as predictive maintenance and fault diagnosis for smart buildings and machineries. In terms of the GAN-AD methodology, instead of simply feeding multiple sequences uniformly into a fully connected network, we plan to enhance GAN-AD with a multi-GAN framework to better capture the extrinsic knowledge about relationships among the networked sensors and components. We will also conduct further research on feature selection for multivariate anomaly detection, and investigate principled methods for choosing the latent dimension and PC dimension with theoretical guarantees.

Acknowledgement

This material is based on research/work supported by the Singapore National Research Foundation and the Cybersecurity R&D Consortium Grant Office under Seed Grant Award No.CRDCG2017-S05.

References

[1] B. Sun, P. B. Luh, Q.-S. Jia, Z. O'Neill, and F. Song, "Building energy doctors: An spc and kalman filter-based method for system-level fault detection in hvac systems," *IEEE Transactions on Automation Science and Engineering.*, vol. 11, no. 1, pp. 215–229, 2014.

[2] K. Donghwoon, H. Kim, J. Kim, S. C. Suh, I. Kim, and K. J. Kim, "A survey of deep learning-based network anomaly detection," *Cluster Computing*, pp. 1–139, 2017.

[3] C. Varun, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey." *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 5, pp. 823–839, 2012.

[4] Y. N. S. Vilbert and Q. Chen, "Computer intrusion detection through ewma for autocorrelated and uncorrelated data," *IEEE transactions on reliability*, vol. 52, no. 1, 2003.

[5] Ryan and T. P., *Statistical methods for quality improvement*. New York, NY: John Wiley & Sons, 2011.

[6] R. S. W., "Control chart tests based on geometric moving averages," *Technometrics*, vol. 1, no. 3, 1959.

[7] M. D. C. and C. M. Mastrangelo, "Some statistical process control methods for autocorrelated data," *Journal of Quality Technology*, vol. 23, no. 3, 1991.

[8] C.-W. Lu and M. R. R. Jr, "Ewma control charts for monitoring the mean of autocorrelated processes," *Journal of Quality technology*, vol. 31, no. 2, 1999.

[9] M. Paige, R. E. Swanson, and C. E. Heckler, "Contribution plots: A missing link in multivariate quality control," *Applied mathematics and computer science*, vol. 8, no. 4, 1998.

[10] W. W. H. and M. M. Ncube, "Multivariate cusum quality-control procedure," *Technometrics*, vol. 27, no. 3, 1985.

[11] L. C. A., W. H. Woodall, C. W. Champ, and S. E. Rigdon, "A multivariate exponentially weighted moving average control chart," *Technometrics*, vol. 34, no. 1, 1992.

[12] Y. Nong and Q. Chen, "An anomaly detection technique based on a chisquare statistic for detecting intrusions into information systems," *Quality and Reliability Engineering International*, vol. 17, no. 2, 2001.

[13] K. P. S. Girase and D. Mukhopadhyay, "A survey of classification techniques in the area of big data," *arXiv preprint arXiv*, vol. 1, no. 11, pp. 1–7, 2015.

[14] G. Mustafaraj, J. Chen, and G. Lowry, "Development of room temperature and relative humidity linear parametric models for an open office using bms data," *Energy and Buildings*, vol. 42, pp. 348–356, Aug. 2010.

[15] F. Xiao, Y. Zhao, J. Wen, and S. Wang, "Bayesian network based fdd strategy for variable air volume terminals," *Automation in Construction*, vol. 41, pp. 106–118, 2014.

[16] Z. Du, B. Fan, X. Jin, and J. Chi, "Fault detection and diagnosis for buildings and hvac systems using combined neural networks and subtractive clustering analysis," *Building and Environment*, vol. 73, pp. 1–11, 2014.

[17] D. Li, G. Hu, and C. J. Spanos, "A data-driven strategy for detection and diagnosis of building chiller faults using linear discriminant analysis," *Energy and Buildings*, vol. 128, pp. 519–529, 2016.

[18] P. Jaikumar, A. Gacic, B. Andrews, and M. Dambier, "Detection of anomalous events from unlabeled sensor data in smart building environments," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 2268–2271.

[19] Y. Zhao, F. Xiao, J. Wen, Y. Lu, and S. Wang, "A robust pattern recognition-based fault detection and diagnosis (fdd) method for chillers," *HVAC&R Research*, vol. 20, no. 7, pp. 798–809, 2014.

[20] T. Mulumba, A. Afshari, K. Yan, W. Shen, and L. K. Norford, "Robust model-based fault diagnosis for air handling units," *Energy and Buildings.*, vol. 86, pp. 698–707, 2015.

[21] D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Fault detection and diagnosis for building cooling system with a tree-structured learning method," *Energy and Buildings*, vol. 127, pp. 540–551, 2016.

[22] ——, "Fusing system configuration information for building cooling plant fault detection and severity level identification," in *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. IEEE, Conference Proceedings, pp. 1319–1325.

[23] Y. Shen, S. X. Ding, X. Xie, and H. Luo, "A review on basic data-driven approaches for industrial process monitoring," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6418–6428, 2014.

[24] S. Li and J. Wen, "A model-based fault detection and diagnostic methodology based on pca method and wavelet transform," *Energy and Buildings*, vol. 68, pp. 63–71, 2014.

[25] H. Xiao, Z. Wang, Y. Liu, and D. H. Zhou, "Least-squares fault detection and diagnosis for networked sensing systems using a direct state estimation approach," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 3, pp. 1670–1679, 2013.

[26] W. S., E. K., and G. P., "Principal component analysis," *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[27] W. Herman, *Partial least squares*. Encyclopedia of statistical sciences, 1985.

[28] D. Xuewu and Z. Gao, "From model, signal to knowledge: A data-driven perspective of fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2226–2238, 2013.

[29] G. Ian, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *In Advances in neural information processing systems*, vol. ACM, 2014, pp. 2672–2680.

[30] M. Olof, "C-rnn-gan: Continuous recurrent neural networks with adversarial training," *arXiv preprint arXiv*, vol. 1611, no. 09904, 2016.

[31] E. Cristbal, S. L. Hyland, and G. Rtsch, "Real-valued (medical) time series generation with recurrent conditional gans," *arXiv preprint arXiv*, vol. 1706, no. 02633, 2017.

[32] X. Yuan, T. Xu, H. Zhang, R. Long, and X. Huang, "Segan: Adversarial network with multi-scale l1 loss for medical image segmentation," *arXiv preprint arXiv*, vol. 1706, no. 01805, 2017.

[33] Y. Raymond, C. Chen, T. Y. Lim, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with perceptual and contextual losses," *arXiv preprint arXiv*, vol. 1607, no. 07539, 2016.

[34] S. Tim, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," in *In Advances in Neural Information Processing Systems*, 2016, pp. 2234–2242.

[35] S. Thomas, P. Seebck, S. M. Waldstein, U. Schmidt-Erfurth, and G. Langs, "Unsupervised anomaly detection with generative adversarial networks to guide marker discovery," pp. 146–157, 2017.

[36] Z. Houssam, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar, "Efficient gan-based anomaly detection," *arXiv preprint arXiv*, vol. 1802, no. 06222, 2018.

[37] G. Jonathan, S. Adepu, K. N. Junejo, and A. Mathur, "A dataset to support research in the design of secure water treatment systems," in *In International Conference on Critical Information Infrastructures Security*, 2016, pp. 88–99.

[38] R. Alec, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv*, vol. 1511, no. 06434, 2015.

[39] M. A. P. and N. O. Tippenhauer, "Swat: A water treatment testbed for research and training on ics security," in *In International Workshop on Cyber-physical Systems for Smart Water Networks (CySWater)*. IEEE, 2016, pp. 31–36.

[40] A. Sridhar and A. Mathur, "Distributed detection of single-stage multipoint cyber attacks in a water treatment plant," in *In Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security*. ACM, 2016, pp. 449–460.

[41] G. Jonathan, S. Adepu, M. Tan, and Z. S. Lee, "Anomaly detection in cyber physical systems using recurrent neural networks," in *In IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*. IEEE, 2017, pp. 140–145.

[42] A. C. Mujeeb, C. Murguia, and J. Ruths, "Model-based attack detection scheme for smart water distribution networks," in *In Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 101–113.

[43] D. Li, Y. Zhou, G. Hu, and C. J. Spanos, "Optimal sensor configuration and feature selection for ahu fault detection and diagnosis," *IEEE Transactions on Industrial Informatics*, vol. 13, pp. 1369 – 1380, 2017.