Deep Generative Models in the Industrial Internet of Things: A Survey

Suparna De, *Member, IEEE*, Maria Bermudez-Edo, Honghui Xu, and Zhipeng Cai, *Senior Member, IEEE*

***Abstract*—Advances in communication technologies and artificial intelligence are accelerating the paradigm of industrial Internet of Things (IIoT). With IIoT enabling continuous integration of sensors and controllers with the network, intelligent analysis of the generated Big Data is a critical requirement. Although IIoT is considered a subset of IoT, it has its own peculiarities in terms of higher levels of safety, security, and low-latency communication in an environment of critical real-time operations. Under these circumstances, discriminative deep learning (DL) algorithms are unsuitable due to their need for large amounts of labeled and balanced training data, uncertainty of inputs, etc. To overcome these issues, researchers have started using deep generative models (DGMs), which combine the flexibility of DL with the inference power of probabilistic modeling. In this article, we review the state of the art of DGMs and their applicability to IIoT, classifying the reviewed works into the IIoT application areas of anomaly detection, trust-boundary protection, network traffic prediction, and platform monitoring. Following an analysis of existing IIoT DGM implementations, we identify challenges (i.e., weak discriminative capability, insufficient interpretability, lack of generalization ability, generated data vulnerability, privacy concern, and data complexity) that need to be investigated in order to accelerate the adoption of DGMs in IIoT and also propose some potential research directions.**

***Index Terms*—Deep generative model (DGM), generative adversarial networks (GANs), industrial Internet of Things (IIoT), survey.**

I. Introduction

THE industrial Internet of Things (IIoT) is a network of intelligent and highly connected industrial components that are deployed to achieve high production rates with reduced operational costs through real-time monitoring, efficient management, and controlling of industrial processes, assets, and operational time [1]. IIoT is a subset of IoT which needs higher levels of safety, security, and reliable communication while considering real-time industrial operations and critical industrial environment. Moreover, IIoT pays attention to efficient management of industrial assets and operations along with predictive maintenance.

The recent breakthroughs in deep learning (DL) and hardware design empower many IIoT applications. DL offers advantages over traditional machine learning (ML) methods due to three characteristics: 1) generalizing the complicated relationship (such as temporal and spatial dependencies) of massive data collected from IIoT settings; 2) making good use of the massive data resource in IIoT since DL relies on Big Data for powerful training; and 3) automatically extracting effective features from IIoT data without laborious feature specification.

However, there are still a number of open challenges toward successfully implementing DL in IIoT networks and obtaining practical and reliable results.

1) *Imbalanced datasets:* The assumption of an abundance of both positive and negative samples does not hold in IIoT as much of a mechanical system's lifetime is in a normal state, with a short duration of faulty states. With mechanical components typically replaced or refurbished before reaching "end of life," manufacturing datasets typically have a skewed distribution, with the number of negative samples (normal state) outweighing the positive samples (faulty state) [2], [3].
2) *Limited labeled data:* Diverse operating conditions and fault modes for sensor data mean that obtaining labeled data is expensive and not always attainable [4], with 80% of the IIoT data being unlabeled [5].
3) *Domain adaptation:* While DL-enabled transfer learning has addressed domain adaptation, it is limited by its assumption of the source and target domains having the same input and output spaces. IIoT settings feature different input sensor signals and different sets of output labels [e.g., fault type, remaining useful life (RUL) range, etc.] across different machines [6].
4) *Large attack surface:* The increased connectivity among a large number of mainly resource constrained devices in IIoT settings is an open problem for the implementation

Manuscript received July 18, 2021; revised December 13, 2021 and February 1, 2022; accepted February 20, 2022. Date of publication March 3, 2022; date of current version June 13, 2022. This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Grant PID2019-109644RB-I00/AEI/10.13039/501100011033 and by ERIC framework under Grant LifeWatch-2019-10-UGR-01. Paper no. TII-21-3046. *(Corresponding author: Suparna De.)*

Suparna De is with the Department of Computer Science and Surrey Institute for People-Centred AI, University of Surrey, GU2 7XH Guildford, U.K. (e-mail: s.de@ieee.org; s.de@surrey.ac.uk).

Maria Bermudez-Edo is with the University of Granada, 18011 Granada, Spain (e-mail: mbe@ugr.es).

Honghui Xu and Zhipeng Cai are with the Department of Computer Science, Georgia State University, Atlanta, GA 30303 USA (e-mail: hxu16@student.gsu.edu; zcai@gsu.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TII.2022.3155656.

Digital Object Identifier 10.1109/TII.2022.3155656

TABLE I
COMPARISON BETWEEN IIOT AND IOT

Aspect	IIoT	IoT
Area	Industry Applications	General Applications
Deployment	Industrial Systems	Smart-X deployments
Service	Machine-oriented	Human-centric
Scalability	Large-scale Network	Low-scale Network
Data Volume	High to Very High	Medium to High
Security Measures	Advanced and Robust	Utility-centric
Communication needs	Low-latency, stringent QoS	Maximised data rate, no QoS guarantees
Resilience	High Fault Tolerance	Not Required

and configuration of security measures, as a secured architecture based on segregation is more difficult [7]. While the deterministic industrial processes result in regular network patterns that facilitate intrusion detection systems (IDS), IIoT networks need more vantage points as traffic does not flow through one central point [7]. Moreover, in critical industrial scenarios where the entire training dataset cannot be disclosed or exchanged with a central server or other agents, federated learning [8] needs to be investigated, in which interconnected devices jointly refine the model parameters in a privacy-preserving manner [9].

5) *Low-latency communication:* Traditional optimization methods for cellular communications, which require exact models and assume stationary wireless fading channels, are difficult to apply in the dynamic IIoT environment due to the many synchronized processes in industrial settings and diverse quality of service (QoS) requirements [10], [11]. The above findings are summarized in Table I, which highlights the differences between IoT and IIoT along different aspects.

Discriminative techniques used in traditional DL techniques, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), or long short-term memory (LSTM), draw the decision boundary in data space [12]. They provide excellent performance but need large labeled and representative datasets, as they require pretraining to provide precise outputs from the learning process [4]. The lack of representative data and imbalanced datasets may make directly learning a target intractable, via discriminative techniques [4], [12]. Data generation and data augmentation have been proposed as possible solutions to mitigate this risk [6]. Deep generative models (DGMs), which can approximate and generate a joint distribution of the target and training data, to generate samples similar to the real data, which also have physical process plausibility, are thus being leveraged in IIoT applications [6], [13]. DGMs create a probability distribution similar to the original by learning a high number of correlations, as opposed to discriminative techniques that simply label instances to their most probable classes. On one hand, learning the distribution of the data (generative classifiers) could potentially provide better performance than boundaries (discriminative classifiers); however, it is not always possible to infer the real distribution of the data, and, sometimes, it is approximated by a normal distribution. Hence, the outliers affect the performance of the generative models. On the other hand, the supervised learning nature of discriminative models means that they tend not to generalize well and can be prone to overfitting if insufficient data is available [6].

Recent works have made use of the feature selection and realistic data sample generation ability of DGMs to apply them in industrial settings for anomaly and intrusion detection [14]–[16], multivariate fault instances generation (to solve the data imbalance problem) [3], trust-boundary protection [17], [18], network traffic prediction [19], [20], wireless channel downlink controller-to-actuator scheduling [11], network slicing [21], and platform monitoring [22]–[25].

DGMs not only have DL's aforementioned benefits (feature extraction and relationship representation), but their data generation ability can aid prediction tasks in some IIoT applications where the collected IIoT data suffers from low usability caused by data incompleteness, (partially) unlabeled data, insufficient quantity, noisy data, etc. DGMs are also finding use in addressing the IIoT transfer learning or domain adaptation challenge, through adversarial approaches where a separate discriminator is used to align the distributions, to mitigate the domain difference [26]. In other words, DGMs can integrate the flexibility of DL and the inference power of probabilistic modeling, model the underlying distribution of the real data, and generate realistic "real" data in an unsupervised manner. They can be used as an upfront layer in a stacked network, providing classified data to subsequent discriminative models (e.g., to a subsequent RNN) to process the massive IIoT sequential data. The motivation of this survey arises from these aforementioned aspects of DGMs and their applicability to the domain of IIoT.

Since DGMs and deep neural networks (DNNs) are not mutually exclusive, we have studied existing surveys on both topics as well as those on IIoT. There are some recent studies that analyze the theoretical and implementation concepts of DGMs [12], [27], reviewing early DGM models such as Boltzmann machines, Gaussian mixture and hidden Markov models, autoencoders, and their variants. Pan *et al.* [28] presented a review of the generative adversarial networks (GANs) category of DGMs and detailed GAN variants from an architectural and objective function-based viewpoint. Similarly, other reviews on GANs focus on specific fields such as computer vision [29] or spatiotemporal data [30]. Emerging applications of DL algorithms such as CNN, vanilla autoencoders, restricted Boltzmann machines, and GANs in the IIoT are presented in [31]. Security aspects of IIoT are also the focus of recent studies, with [7] focusing on security challenges in IIoT and [32] on differential privacy applications.

Although these existing works review conventional DGMs, GANs, DL-based IIoT applications, and the security issues of IIoT, there is no work that reviews the applications of DGMs in the IIoT domain. Therefore, this survey mainly focuses on comprehensively reviewing the applications of DGMs in the IIoT domain. The comparison between existing related surveys and this one is explicitly shown in Table II.

The rest of this article is organized as follows. We review the state-of-the-art DGMs in Section II. We analyze their applications in IIoT scenarios in Section III. We identify existing challenges with respect to their applicability and present some research promising directions for solving the corresponding challenges in Section IV, which concludes this article.

TABLE II
COMPARISON WITH EXISTING RELATED SURVEYS

Reference	DL	DGMs	IIoT	Other Applications
[7]			✓	
[12]		✓		
[27]		✓		
[28]		✓		
[29]		✓		✓
[30]		✓		✓
[31]	✓		✓	
[32]			✓	
This Survey		✓	✓	

II. DEEP GENERATIVE MODELS

A. Overview of DGMs

DGMs fall under the class of unsupervised ML algorithms, which aim to extract meaningful concepts from raw data. DGMs enable an approximation of the distribution of the data through conditional density estimation, where the characteristics of the probabilistic generative models enable the uncertainty of data to be captured. They can be thought of as the combination of DL with graphical models. DNNs are based only on point estimates and make deterministic predictions, given some feature vectors. Most works on DNNs do not pay much attention to the complexity of these models. Probabilistic models, on the other hand, are mainly conjugate and linear models and can be said to have a simplicity bias. The basis of DGMs is to have the simplest hypothesis that can explain the data, is tractable, can compute expectations, and remove biases for underfitting/overfitting. By combining the probability distribution view of the dataset with a generative iterative process or Markov chains, DGMs can offer a unified framework for model building, inference, prediction, and decision-making. They are also robust to overfitting and have explicit accounting for uncertainty of data and variability of outcomes.

As a result, DGMs are seeing widespread adoption in many industries, such as those involving computer vision based automation, e.g., image generation/compression and super resolution and object detection within images with relevant bounding boxes (to enable self-driving cars); generating synthetic data to accelerate scientific experiments; and designing new experiments for particle physics or drug discovery. In the following sections, we present the three categories of DGMs, i.e., 1) autoregressive models (ARs), 2) variational autoencoder (VAE), and 3) GANs, outlining their architecture and popular variants with recent advances.

B. Autoregressive Models

An AR [33] is a specific regression model on a time series, in which a value from this time series is regressed on previous values from the same series. The first-order autoregression model, written as AR(1), can be presented as follows:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \epsilon_t \tag{1}$$

where y_t is a time-series variable y measured in time t, y_{t-1} is y measured in time $t-1$, and $\epsilon_t, \beta_0, \beta_1$ denote the assumed error and the parameters in a simple linear regression model, respectively. Similarly, the second-order autoregression model, called AR(2), would be

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \epsilon_t \tag{2}$$

in which the time-series variable's value at time t can be predicted from its values at times $t-1$ and $t-2$. More generally, a kth-order autoregression model, denoted as AR(k), will be a multiple linear regression where the time-series variable's value at any time t can be calculated by a linear function of the values at times $t-1, t-2, \ldots, t-k$, as shown in the following equation:

$$y_t = \beta_0 + \beta_1 y_{t-1} + \beta_2 y_{t-2} + \cdots + \beta_k y_{t-k} + \epsilon_t. \tag{3}$$

Neural autoregressive distribution estimation (NADE) [34] addresses the problem of modeling a joint distribution. For starters, the assumption is that the dimensions of $\mathbf{x}$ are binary (i.e., $x_d \in \{0, 1\} \forall d$). To estimate the D-dimensional distribution of $p(\mathbf{x})$, NADE begins by making the observation that $p(\mathbf{x})$ can be cast into a product of conditional 1-D distributions, in any order o (a permutation of the integers $1, \ldots, D$)

$$p(\mathbf{x}) = \prod_{d=1}^{D} p(x_{o_d} | \mathbf{x}_{o_{<d}}) \tag{4}$$

where o_d contains the first $d-1$ dimensions in ordering o and $\mathbf{x}_{o_{<d}}$ is the corresponding subvector for these dimensions. Therefore, an "autoregressive" generative model of the data can be obtained simply by specifying a parameterization of all D conditionals $p(x_{o_d} | \mathbf{x}_{o_{<d}})$. In NADE, we can model each conditional using a feed-forward neural network (NN). Specifically, $p(x_{o_d} | \mathbf{x}_{o_{<d}})$ is parameterized as follows:

$$p(x_{o_d} | \mathbf{x}_{o_{<d}}) = \sigma(\mathbf{V}_{o_d} \mathbf{h}_d + b_{o_d}) \tag{5}$$

$$\mathbf{h}_d = \sigma(\mathbf{W}_{o_{<d}} \mathbf{x}_{o_{<d}} + \mathbf{c}) \tag{6}$$

where σ is the sigmoid function, H is the number of hidden units, and $\mathbf{V} \in \mathbb{R}^{D \times H}$, $\mathbf{b} \in \mathbb{R}^{D}$, $\mathbf{W} \in \mathbb{R}^{H \times D}$, and $\mathbf{c} \in \mathbb{R}^{H}$ are the parameters of the NN model. Finally, NADE can be trained by maximum likelihood or, equivalently, by minimizing the average negative log-likelihood

$$\frac{1}{N} \sum_{n=1}^{N} -\log(\mathbf{x}^{(n)}) = \frac{1}{N} \sum_{n=1}^{N} \sum_{d=1}^{D} -\log(p(x_{o_d}^{(n)} | \mathbf{x}_{o_{<d}}^{(n)})) \tag{7}$$

by stochastic (minibatch) gradient descent method, where N is the batch size.

Derived AR models: PixelCNN [35] is an autoregressive generative model based on CNN, which models the conditional distribution of seen image pixel values, and a gated CNN is used to remember prior pixel values in this gated architecture. PixelCNN++ [36] is a modified version of PixelCNN, in which some tweaks, including downsampling, dropout, and skip-out

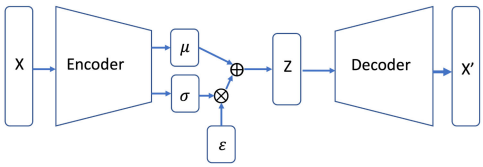


Fig. 1. Architecture of a VAE.

connections, are used to obtain better performance results. PixelRNN [37] is also proposed in the same study by using 12 LSTM layers while adopting the convolutional approach in PixelCNN. PixelVAE [38] is proposed to combine the benefits of VAEs and PixelCNNs, in which a conditional PixelCNN can be exploited as the output of the VAE's decoder in PixelVAE.

To sum up, AR models are basic ARs applied on simple time-series data generation; NADE model integrates the idea of autoregression with the function of NNs to obtain a better generalization performance for any data type generation; PixelCNN is proposed by specifying the NN in NADE model using CNNs, PixelCNN++ is a modified PixelCNN by using helpful tweaks to get a better performance, PixelRNN is an RNN version, and PixelVAE is a VAE version of the NADE model, a series of which are leveraged to do image generation.

C. Variational Autoencoders

VAEs [39]–[41] are deep Bayesian networks using NN, specifically multilayer perceptrons. Hence, they can support complex data distributions with fast training via back-propagation. The goal of VAEs is to find the hidden/latent variables to simplify the generation task. For example, in a task of generating faces, the pose or the color of the eyes is not annotated (latent). An autoencoder is formed by two NNs. The first one is an encoder that codifies the input into a latent vector and the second one is a decoder that converts the latent vector into an output that replicates the input. To properly generate data, the autoencoders need to have a regularized latent space in which all the points in the latent space are meaningful. VAEs solve this issue by encoding the input not into latent points but into distributions over the latent space. Then, the distribution is sampled as points to feed the decoder (see Fig. 1). In this way, VAEs avoid overfitting and enhance the decoder to function as a generator of meaningful data.

Formally, the encoder is represented by $q(z_i = P(z_i/x_i, \theta))$. Its input is x_i and its output is the distribution of the latent space Z. A sample of this distribution is the input of the decoder that computes $P(x_i/z_i, \theta)$. For every point X in the dataset, there exists at least one vector of latent variables z able to generate something similar to x with a deterministic function $q(z; \theta)$, parametrized by a vector θ. VAEs aim at optimizing θ such that z can be sampled from the probability density function of z, $P(z)$, and $q(z; \theta)$ will generate x, using the law of total probability $q(z; \theta) = P(x|z; \theta)$. Hence, the aim of VAEs is to maximize the probability of each X ($P(x)$, the marginal) in the training set according to

$$P(x) = \int (P(x|z; \theta)P(z)dz). \tag{8}$$

However, the distribution $P(x/z)$ is intractable, especially in high-dimensional space. To minimize this function, VAEs use the decoder to find the z that reconstructs x ($P(z|x; \theta)$). Hence, the problem is to find a tractable model distribution $q(z)$ to approximate the true posterior $P(z|x)$, via variational inference. To that end, VAEs need to reduce the diversion (asymmetric distance) between two probability distributions ($q(z)$ and $P(z|y)$) with Kullback–Leibler divergence, KL

$$\mathrm{KL}(q(z)|P(z|x)) = E_{q(z)}\left[\ln \frac{q(z)}{P(z|x)}\right] \tag{9}$$

where $E_{q(z)}$ represents the expectation of the distribution $q(z)$. Hence, the problem is to minimize the KL distance. Applying Bayes rule

$$\begin{aligned}\mathrm{KL}(q(z)|P(z|x)) &= E_{q(z)}\left[\ln \frac{q(z)}{P(z|x)}\right] \\ &= E_{q(z)}[\ln q(z)] - E_{q(z)}[\ln P(z|x)] \\ &= E_{q(z)}[\ln q(z)] - \mathbb{E}_{q(z)}[\ln P(z, x)] + \ln P(x). \end{aligned} \tag{10}$$

Rearranging (10) gives

$$\ln P(x) = \mathrm{KL}(q(z)|P(z|x)) + \sum \ln \frac{P(x, z)}{q(z)}. \tag{11}$$

Equation (11) is a constant ($\ln P(x)$), which is equal to a term we want to minimize (KL) and a term that it is the variational lower bound or evidence lower bound (ELBO). The ELBO is a lower bound on the probability of observing some data under a model, which is used as an optimization criterion for approximating the posterior distribution. So, the problem can be converted into maximizing the ELBO $\left(\sum \ln \frac{P(x,z)}{q(z)}\right)$, where

$$\begin{aligned}\sum \ln \frac{P(x, z)}{q(z)} &= \sum q(z) \ln P(x/z) + \sum q(z) \ln \frac{P(z)}{q(z)} \\ &= E_{q(z)}P(x/z) - \mathrm{KL}(q(z)|P(z)).\end{aligned} \tag{12}$$

To reduce the complexity, $P(x|z; \theta)$ is often chosen as a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ with mean μ and covariance σ^2 as follows: $P(x|z; \theta) = \mathcal{N}(x|f(z; \theta), \sigma^2 * I)$, or Bernoulli distributions for binary data. Hence, $(E_{q(z)}P(x/z))$ is the reconstruction error. Assuming a normal distribution, σ^2 is a diagonal matrix (a vector) and the encoder generates the mean and variance of z as vectors. To simplify the calculations, Kingma and Welling [39] proposed to reparametrize z, as $z_i = \mu_i(y) + \epsilon_i\sigma_i^2(y)$, where $\epsilon_i \sim \mathcal{N}(\epsilon; 0, 1)$ introduces noise to allow the generation of unseen data (see Fig. 1). This trick reduces variance in the gradients and, hence, allows the stochastic gradient descent [41] and error back-propagation [42] in NN. Therefore, VAEs learn the probabilistic generative model $p_\theta(x|z)$ (decoder) as well as an approximated posterior distribution $q_\phi(z|x)$ (encoder) by maximizing the ELBO

$$\mathcal{L}(\phi, \theta, x) = E_{q_\phi(z/x)}[\ln p_\theta(x/z)] - KL(q_\phi(z/x)||p(z)). \tag{13}$$

VAEs allow building flexible models, but, due to the diagonal covariance, they cannot capture fine grained details as autoregressive networks do. VAEs have been used to recognize and generate complex data, mainly images [38], [43], such as

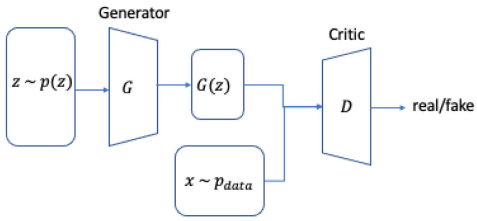


Fig. 2. Architecture of a GAN.

handwritten digits [39], [42], [44], faces [39]–[41], [45], and house numbers [43], [46], to rotate or modify the light of an image [45], noisy images [42], and even to predict the next frame in a video [47].

Derived VAE models: Some works have improved the flexibility of VAEs by the introduction of auxiliary latent variables forming multilayers of latent variables [44], [48], [49]. Others try the flexibility by normalizing flows [50], [51]. Kingma *et al.* [52] propose inverse autoregressive flow. Huang *et al.* [53] introduce neural autoregressive flows and De *et al.* [54] enhance it. D-VAE [55] proposes a directed acyclic graph VAE. tvGP-VAE [56] enhances VAE with tensor variate Gaussian processes, allowing for arbitrary correlation structures in the latent space via kernel functions.

D. Generative Adversarial Networks

In contrast to autoregressive and VAE models that are likelihood-based, GANs are likelihood-free generative models, which combine a generator and discriminator in the same network. First proposed by Goodfellow *et al.* [57] in 2014, GANs are based on game theory, with the generator G_θ learning the data distribution via unsupervised learning, to create realistic adversarial samples, and the discriminator D_ϕ (or the critic) classifying it as real or fake (simulated).

During learning, the generator and discriminator are updated alternatively. G_θ is a directed latent variable model that generates samples $\mathbf{x}$ from $\mathbf{z}$, where $\mathbf{x}$ denotes samples from input data or generator and $\mathbf{z}$ is the noise input. The discriminator function tries to distinguish samples from the real dataset and the generator by maximizing the objective $(p_{\text{data}} \neq p_\theta)$ or minimizing $D(G(z;\theta_g))$ for generated samples from p_z not from p_{data}. The architecture of a GAN is shown in Fig. 2.

G_θ minimizes a two-sample test objective $(p_{\text{data}} = p_\theta)$, which is equivalent to minimizing $1 - D(G(z;\theta_g))$, as D is a binary classifier. Thus, overall, GANs have a minimax learning objective

$$\min_\theta \max_\phi E_{x\sim p_{\text{data}}(x)}[\log D(x)] + E_{z\sim p_z(z)}[\log(1 - D(G(z;\theta_g)))]. \quad (14)$$

The implementation of GANs can prove challenging due to their 1) unstable optimization procedure, where the generator and discriminator loss continues to oscillate without converging and 2) potential for mode collapse, with the generator producing one of a few types of samples over and over again. Some studies [28] have proposed for the discriminator to use the minibatch layer to reflect the diversity of the sample, to avoid mode collapse.

Derived GAN models: With the G and D networks being multilayer perceptrons in the original GAN model, various derived GAN architectures have been proposed in order to improve the performance in terms of data diversity, data quality, and more stable training [28]–[30]. The deep convolutional generative adversarial networks [58] apply CNN in the generator and critic, for better image feature extraction. Conditional GANs (CGANs) [59] seek to address mode collapse by introducing a conditional variable c, in both the generator and discriminator. This makes the input to the discriminator to be $G(z|c)$ from the generator, with the real sample also derived from c. Other approaches to avoid mode collapse include those that combine the adversarial loss of GANs with the objective function of VAEs [60], by replacing the VAE decoder with the GAN generator, and Wasserstein GAN (WGAN) [61], where a new loss function derived from the Wasserstein distance is used and D is used to score data quality by estimating the Wasserstein metric between generated and original data distribution. For unsupervised image-to-image translation, CycleGAN [62] has been proposed to learn the mapping between an input image and an output image, where paired training data may not be available. Self-attention GAN (SAGAN) [63] includes self-attention layers in the G and D networks, allowing to learn global, long-range dependencies for generating images specially in multiclass image generation. D checks that features in distant parts of the image are consistent (e.g., the nose and ears are in the right place of the face). Your Local GAN (YLG) [64] enhances SAGAN by making the networks as sparse as possible for computational and statistical efficiency. YLG introduces a new local sparse attention layer that preserves the geometry and locality. Multiscale gradient-GAN [65] creates multiscale connections between G and D, which allows for the gradients to flow at multiple resolutions simultaneously. This enhances the adaptation to different datasets, which is uncommon in GANs due, in part, to instability during training because there is not enough overlap in real and fake distributions. Another approximation to improve the performance of GANs is to enhance the loss function. f-GAN [66] uses a more general notion of distance, the f-divergence, which includes Jenson–Shannon and total variation as distance metrics for training generative neural samplers. RealnessGAN [67] represents the concept of realness as a distribution rather than a single scalar (real or generated). Loss sensitive GAN [68] introduces a loss function to quantify the quality of generated samples, keeping the loss of the real sample smaller than that of a generated counterpart.

To address the federated learning challenge in privacy-preserving scenarios, the authors in [69] and [70] have proposed distributed GANs, where a number of cooperating agents learn the GAN task in a decentralized manner without sharing their data with any central server or among themselves. The work in [69] is applied to a distributed IDS with the agents sharing the weights of their D models, while in brainstorming GAN [70], the GAN value function is modified to a brainstorming function to integrate the generated data points across neighboring agents.

TABLE III
IIoT APPLICATIONS BASED ON DGMs

Reference	Domain	Scenario	Model	Advantage
[71]	Anomaly Detection	IIoT Equipments	VAE	VAE-based reconstruction probability for analyzing cause of anomaly.
[72]	Anomaly Detection	IIoT Equipments	GAN	Generator trained with features extracted from normal samples, apparent and latent loss values for anomaly score.
[14]	Anomaly Detection	IIoT Equipments	ARX	Spatiotemporal correlation among multidimensional data helps detection.
[73]	Anomaly Detection	IIoT Equipments	CGAN	Generative adversarial ANNs help detect security anomalies in layer-scale cyber physical systems.
[20]	Trust-Boundary Protection	IIoT Networks	Deep-IFS	Two autoregressive units (LocalGRU and MHA) improve the network intrusion detection performance.
[17]	Trust-Boundary Protection	IIoT Systems	GAN	Downsampler-encoder-based cooperative data generator ensures better extraction of distribution of IIoT data.
[18]	Trust-Boundary Protection	IIoT Systems	GAN	Deep-learning feature-extration-based semisupervised model achieves an adaptive protection mechanism.
[74]	Trust-Boundary Protection	IIoT Networks	conditional VAE	Intrusion labels integrated inside the decoder layers to improve network intrusion detection accuracy.
[75]	Trust-Boundary Protection	IIoT Systems	DIGFuPAS	GAN for generating adversarial attack samples, IDS robustness improvement by retraining classifiers.
[76]	Trust-Boundary Protection	IIoT Systems	ARIES	GAN with decision tree and SVM, for attack classification and identifying packet and operating data abnormalities.
[77]	Trust-Boundary Protection	IIoT Systems	GAN	The utilization of double GANs overcomes the vulnerability of over-training the authorized devices.
[19]	Network Traffic Prediction	IIoT Networks	MTL + LSTM	Multi-task learning architecture improves network traffic prediction accuracy.
[11]	Network Traffic Prediction	IIoT factory	GAN	GAN learns the wireless channel distributions and schedules the downlink transmissions accordingly.
[78]	Network Traffic Prediction	IIoT Networks	GAN	GAN for autonomous wireless channel modeling, AWGN distribution approximation.
[79]	Network Traffic Prediction	IIoT Networks	GAN	Network traffic classification at protocol, application and attack type.
[21]	Network Traffic Prediction	IIoT Networks	GAN+distributional Q network	Network slicing for physical layer resource management.
[23], [24], [80]	Platform Monitoring	Oil Production	ARs	Multi-variate regression model processes time-series data to predict oil production.
[81]	Platform Monitoring	Artificial Lift	AR + LSTM	LSTM improves the gas/oil production prediction performance for artificial lift mechanisms.
[22]	Platform Monitoring	Paste Thickner Control	AR + RNN	Neural network-based model predictive control scheme enhances the performance of paste thickner control.
[25]	Platform Monitoring	Temperature Control	NARX	External input improves the perdiction of syngas heating value and hot flue gas temperature.
[82]	Platform Monitoring	Motor Vibration	RNN-VAE	RNN-VAE leverages motor vibration time-domain signals to detect motor fault.
[83]	Platform Monitoring	Hot Strip Mill	VAE	VAE improves the performance of the fault detection in hot strip mill process.
[84]	Platform Monitoring	Hot Metal Production	VAE-LIME	Local Interpretable Model agnostic Explanations could interpret the blackbox of the neural network.
[85]	Platform Monitoring	Voltage Dip Classification	GAN	GAN drives an active learning-based automatic labelling method which helps train voltage dip classification system.
[3]	Platform Monitoring	RUL of manufacturing components	CGAN + GRU	GAN data augmentation helps generating multi-variate fault instances.
[2]	Platform Monitoring	Synthetic faults for trucks' APS	CGAN + WCGAN	Synthetic faults generated by conditioning on the minority data cluster.
[86]	Platform Monitoring	Fault detection in gearboxes	GAN	Fault diagnosis integrated with adversarial training to optimize both real and faulty data.
[87]	Platform Monitoring	Fault detection in rotating machinery	wavelet transform (WT) GAN + CNN	WT to extract time-frequency image features, GAN for data augmentation, CNN for fault detection.

III. APPLICATIONS OF DGMs IN INDUSTRIAL IoT

We survey some representative IIoT application domains to which deep generative methods have been applied and demonstrated notable performance improvement. They are in four domains, which include the following:

1) anomaly detection;
2) trust-boundary protection;
3) network traffic prediction;
4) platform monitoring.

All references are summarized in Table III.

A. Anomaly Detection

Anomaly detection approaches aim to learn the system behavior under normal operating conditions to be able to identify later system states that are dissimilar. Both VAEs and GANs have been applied for anomaly detection by learning the induced distribution and subsequently asserting if a sample is part of the distribution by mapping it to the closest sample in the generated distribution. VAEs tackle this mapping by adapting an autoencoder learnt during the training [71]. A GAN-based anomaly detection algorithm for imbalanced industrial time series datasets has been proposed in [72], with an encoder–decoder–encoder structured G network with convolutional layers. Only normal samples with elaborately extracted features are used in model training. The model outputs anomaly scores comprising apparent and latent loss, with fault samples generating much higher anomaly scores.

Considering that large volumes of multidimensional data are generated in 6G IIoT, the authors in [14] designed an autoregressive exogenous model (ARX) for eliminating the noise in data for anomaly detection, and a multidimensional data relationship diagram is creatively used to characterize the spatiotemporal correlations among heterogeneous data. The authors in [73] applied CGANs to search for security anomalies, noting that the discriminator needs to be trained for more steps than the generator to ensure that their loss curves converge.

B. Trust-Boundary Protection

Trust-boundary techniques are applied in IIoT to segment the networks, with IIoT processes and data storage separated into different segments based on user access privilege [17]. The authors in [88] use the GAN model to generate adversarial samples for aiding the design of trust-boundary protection mechanism against adversarial attacks. However, the distribution of noisy inputs of this GAN model largely differs from real data distribution in IIoT networks.

Therefore, Hassan *et al.* [17] proposed a downsampler encoder-based cooperative data generator to ensure better extraction of real distribution of IIoT network data in attack models, which is updated and verified using a DNN discriminator to guarantee its robustness with the idea of GAN's adversarial training. In [18], they further presented an adaptive trust-boundary protection mechanism for IIoT networks using DL feature extraction based semisupervised model, which avoids manual effort to update the attack databases and automatically learns the rapidly changing natures of unknown attack models by using unsupervised learning and unlabeled data from the wild.

The large number as well as the heterogeneity of devices and communication protocols contribute to the large attack surface problem in IIoT networks. Trust-boundary protection, thus, uses intrusion detection as a core technique to control access levels [17]. To this end, Deep-IFS [20] is a forensics-based DL model to detect intrusions in IIoT traffic. Deep-IFS learns local representations using local gated recurrent unit (LocalGRU) and captures global representation using multihead attention (MHA). Two autoregressive units' utilization improves the robustness of Deep-IFS model for intrusion detection on IIoT traffic in fog environment. In [74], the authors use conditional VAEs to detect network intrusions, using the labels in

the training data as an extra input in the decoder to improve the accuracy. DIGFuPAS [75] aims to increase the ability of IDS against adversarial attacks by using a WGAN to repetitively retrain classifiers from crafted network traffic flow. ARIES [76] is a multilayered IDS that integrates unsupervised GAN with supervised decision tree and support vector machine (SVM). The first layer classifies attacks such as denial of service, brute force, port scanning attacks, etc., in a supervised manner, with the second and third layers identifying packet and operating data abnormalities.

Special attention needs to be paid to the radio frequency (RF) fingerprinting protection. In IIoT, as there can be numerous devices transmitting their data over RF, they could be subject to attacks where a fake device supplants the identity of a genuine device and transmits malicious data. RF fingerprinting is used to verify the identity of a device based on imperfections of the transmitters, such as in-phase and quadrature signal (IQ) imbalance, amplifier nonlinearity, digital-to-analog converter nonlinearity, carrier frequency offsets, and oscillator drift. In [77], the authors use GAN first to generate malicious data that simulates an authorized device that exists. Then, they use a GAN again to overcome this vulnerability by overtraining the authorized device with generated data.

C. Network Traffic Prediction

End-to-end network traffic is an essential information for many network security and management functions in IIoT; so network traffic prediction is not a trivial issue. Cellular traffic optimization for meeting the low-latency requirements in IIoT scenarios is an open problem [10]. Moreover, QoS is sensitive to packet size distributions, packets' interarrival time, and channel fading. Motivated by this, Nie *et al.* [19] proposed an effective prediction mechanism using multitask learning architecture and an autoregressive unit which takes advantage of link loads as additional information to improve network traffic prediction accuracy. To address the issue of limited data samples in channel fading models, Liu *et al.* [11] applied the GAN model to learn the wireless channel distributions in a factory environment and schedule the controller to actuator downlink transmissions accordingly, while also taking into account nonstationary channel fading. GANs have also been employed in [78] to propose a wireless channel modeling framework, with the results offering a good approximation of a real wireless channel. The applicability of GANs in traffic classification scenarios with $<$20% labeled traffic flows has been demonstrated in [79], by finding representation features of raw traffic data into lower dimension feature space. GANs have also found application in demand aware resource allocation by network slicing in a 5G cellular environment [21], to meet the diverse QoS needs over a common physical infrastructure, where a GAN-powered deep distributional Q network has been proposed to approximate the action-value distribution.

D. Platform Monitoring

IIoT integration has been an ultimate growth factor for multinational companies, especially in oil and gas industry. To this end, Sonawane *et al.* [80] presented a multivariate regression model to predict the future production performance of oil wells based on monthly production time series data, which ensures that the owner of oil and gas can monitor the equipment at a fine granularity. Similarly, [23] and [24] used an AR on IIoT device data to forecast the value of oil production to help in detecting anomalous values and provide an idea about any flaws in the oil well. In [81], an AR is integrated with a DL model (LSTM) to realize artificially lift mechanisms like beam pumping, hydraulic pumping, electronic submersible pumping, and gas lifts, while making sure that the oil and gas production is predicted accurately. In addition, IIoT can also be used to control paste thickener [22], in which an NN-based model predictive control scheme is implemented over an IIoT platform with the help of an autoregression unit (attention RNN). The authors in [25] proposed an NN-based nonlinear autoregressive with external input (NARX) model to predict syngas heating value and hot flue gas temperature for monitoring a waste-to-energy plant by using data collected by IIoT. Moreover, in [82], an RNN-based VAE is used to detect motor fault by using motor vibration time-domain signals, while [83] uses VAE for fault detection in a hot strip mill process. There, the authors first extract quality-related latent variables using deep variational information bottleneck, which minimizes the mutual information between latent variables and observations while maximizing mutual information between latent variables and process quality.

Furthermore, with the help of local interpretable model agnostic-explanations (LIME) that could interpret the blackbox of the NN, [84] proposed a VAE-LIME model for interpreting the models forecasting the temperature of the hot metal produced by a blast furnace. In [85], the generative–discriminative model pair in GAN drives an active learning-based automatic labeling method of voltage dip sequences used for training a voltage dip classification system.

The data generation ability of GANs can address the problem of fault data unavailability and imbalanced datasets in manufacturing IIoTs and has, thus, found use in predictive maintenance functions. Behera *et al.* [3] proposed a novel prognostics system based on CGAN and deep gated recurrent unit (GRU) to generate multivariate fault instances for predicting the RUL of manufacturing components, while CGAN and Wasserstein CGAN (WCGAN) are benchmarked in [2] for generating synthetic faulty samples for trucks' air pressure systems (APS). Adversarial training with GANs to optimize both real and fault data for fault detection in gearboxes is proposed in [86], while GANs with CNN for fault detection in rotating machinery are applied in [87].

IV. Conclusion

In the following paragraphs, we highlight several challenges that need to be investigated in order to accelerate the adoption of DGMs in IIoT, open issues, and future research directions, and then conclude the article.

Limited Expressive Power: Although DGMs are promising approaches leveraged to do data generation and assist prediction tasks, their power is limited by the relatively fixed network architecture and stringent requirements on the input of DGMs.

However, the data collected from IIoT usually contain noise and abnormal data and have unexpected formats. Therefore, data preprocessing should be implemented on the collected data before feeding them into DGMs, and the architecture of DGMs would require redesigning in order to be applicable to specific scenarios.

Weak Discriminative Capability: The existing DGMs fail to achieve the expected performance on sophisticated structured probabilistic models and completed unsupervised tasks (e.g., mode collapse of GANs). Combining different kinds of DGMs is a promising direction to further improve the generation performance; semisupervised approaches can be taken into account to alleviate the side effect of completely unsupervised training.

Insufficient Interpretability: DGMs lack sufficient interpretability since the latent vector used for generation is hard to interpret; as a result, it is difficult to capture the semantic meaning of the generated data. More approaches to improve understanding of latent vectors should be a focus for the future research of DGMs. Such interpretable latent vectors will be able to do controllable generation owing to the understanding of semantic meaning of generated data.

Lack of Generalization Ability: The trained DGMs usually lack generalization ability since they can only generate data samples conforming to data in the training dataset but cannot generate new data samples which are dissimilar to that in the training dataset. For example, once a DGM is trained with training dataset containing cat and dog images, the trained DGM can only be used to generate cat/dog images but cannot generate bird images. However, in practice, it is difficult to collect a comprehensive training dataset, leading to DGMs' limitation of generalization ability. To this end, the idea of continual learning can be taken into account to improve the generalization ability of DGMs.

Generated Data Vulnerability: The data generated by DGMs are not as good as real data, making it possible to distinguish generated data from the real data collected from IIoT with the help of ML techniques. The idea of adversarial training can be leveraged to avoid detection from ML techniques when the generated data have already been trained to pass the corresponding detection models.

Privacy Concern: Large real-world datasets in IIoT applications are used by DGMs to generate IIoT data, which unavoidably raise many privacy concerns. Therefore, a privacy protection mechanism should be an indispensable component for designing a feasible privacy-preserving DGM in order to prevent privacy leakage as well as maintain the performance of IIoT data generation. There have been encouraging developments through distributed GANs [69], [70] in this direction; however, in the presence of unreliable wireless links and limited resources on IIoT devices, optimizing scheduling and bandwidth allocation are open issues in IIoT privacy-preserving federated learning.

Data Complexity: Data collected from IIoT are massive and come from multiple sources. On the one hand, massive IIoT data brings the challenge of time and structure complexity; so more time-efficient and lightweight DGM architectures should be designed to handle the voluminous input data as well as maintaining the performance of generation. On the other hand, DGMs should evolve to generate multisource data in IIoT while managing the possible conflicts between different data sources. Aligned to this is the issue of energy efficiency, considering that model performance optimization can quickly drain energy in low-powered IIoT devices [4], especially in decentralized cases as noted above, where the computation is done on the devices. A promising development in this direction is that of GAN-powered compressed sensing [89] that enables energy-constrained IIoT sensors to efficiently sense signals without requiring high-rate samplers, minimizing energy consumption. This needs to be supported with the development of models that can be trained to infer useful information from the compressed data directly without actually uncompressing it.

DGMs and specifically networks incorporating adversarial training have received much recent research attention, due to their ability to understand the underlying data distribution. As a result, DGMs have huge potential in IIoT scenarios. In this article, we presented the state-of-the-art DGMs for IIoT and detailed the different applications of DGM-based IIoT. We also outlined several outstanding research challenges and identified future directions. We believe that this survey will motivate IIoT and DGM researchers to further investigate this exciting research topic and develop more creative and computationally efficient DGMs for IIoT applications.

References

[1] W. Z. Khan, M. Rehman, H. M. Zangoti, M. K. Afzal, N. Armi, and K. Salah, "Industrial Internet of Things: Recent advances, enabling technologies and open challenges," *Comput. Elect. Eng.*, vol. 81, 2020, Art. no. 106522.

[2] Y. Fathy, M. Jaber, and A. Brintrup, "Learning with imbalanced data in smart manufacturing: A comparative analysis," *IEEE Access*, vol. 9, pp. 2734–2757, 2021.

[3] S. Behera and R. Misra, "Generative adversarial networks based remaining useful life estimation for IIoT," *Comput. Elect. Eng.*, vol. 92, 2021, Art. no. 107195.

[4] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, "Deep learning in the industrial Internet of Things: Potentials, challenges, and emerging applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11016–11040, Jul. 2021.

[5] M. Willnerd, "Why advanced industries need advanced machine learning," 2020. [Online]. Available: https://toumetis.com/why-advanced-industries-need-advanced-machine-learning/

[6] O. Fink, Q. Wang, M. Svensén, P. Dersin, W.-J. Lee, and M. Ducoffe, "Potential, challenges and future directions for deep learning in prognostics and health management applications," *Eng. Appl. Artif. Intell.*, vol. 92, 2020, Art. no. 103678.

[7] M. Serror, S. Hack, M. Henze, M. Schuba, and K. Wehrle, "Challenges and opportunities in securing the industrial internet of things," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 2985–2996, May 2021.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, pp. 1–19, 2019.

[9] T. Kotsiopoulos, P. Sarigiannidis, D. Ioannidis, and D. Tzovaras, "Machine learning and deep learning in smart manufacturing: The smart grid paradigm," *Comput. Sci. Rev.*, vol. 40, 2021, Art. no. 100341.

[10] H. Zhou, C. She, Y. Deng, M. Dohler, and A. Nallanathan, "Machine learning for massive industrial internet of things," *IEEE Wireless Commun.*, vol. 28, no. 4, pp. 81–87, Aug. 2021.

[11] C.-F. Liu and M. Bennis, "Data-driven predictive scheduling in ultra-reliable low-latency industrial IoT: A generative adversarial network approach," in *Proc. IEEE 21st Int. Workshop Signal Process. Adv. Wireless Commun.*, 2020, pp. 1–5.

[12] H. GM, M. K. Gourisaria, M. Pandey, and S. S. Rautaray, "A comprehensive survey and analysis of generative models in machine learning," *Comput. Sci. Rev.*, vol. 38, 2020, Art. no. 100285.

[13] J. A. Lasserre, C. M. Bishop, and T. P. Minka, "Principled hybrids of generative and discriminative models," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2006, pp. 87–94.

[14] G. Han, J. Tu, L. Liu, M. Martínez-García, and Y. Peng, "Anomaly detection based on multidimensional data processing for protecting vital devices in 6G-enabled massive IIoT," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5219–5229, Apr. 2021.

[15] D. Wu, Z. Jiang, X. Xie, X. Wei, W. Yu, and R. Li, "LSTM learning with Bayesian and gaussian processing for anomaly detection in industrial IoT," *IEEE Trans. Ind. Informat.*, vol. 16, no. 8, pp. 5244–5253, Aug. 2020.

[16] X. Li, M. Xu, P. Vijayakumar, N. Kumar, and X. Liu, "Detection of low-frequency and multi-stage attacks in industrial Internet of Things," *IEEE Trans. Veh. Technol.*, vol. 69, no. 8, pp. 8820–8831, Aug. 2020.

[17] M. M. Hassan, M. R. Hassan, S. Huda, and V. H. C. de Albuquerque, "A robust deep learning enabled trust-boundary protection for adversarial industrial IoT environment," *IEEE Internet Things J.*, vol. 8, no. 12, pp. 9611–9621, Jun. 2021.

[18] M. Hassan, S. Huda, S. Sharmeen, J. Abawajy, and G. Fortino, "An adaptive trust boundary protection for IIoT networks using deep-learning feature extraction-based semi-supervised model," *IEEE Trans. Ind. Informat.*, vol. 17, no. 4, pp. 2860–2870, Apr. 2021.

[19] L. Nie *et al.*, "Network traffic prediction in industrial Internet of Things backbone networks: A multi-task learning mechanism," *IEEE Trans. Ind. Informat.*, vol. 17, no. 10, pp. 7123–7132, Oct. 2021.

[20] M. Abdel-Basset, V. Chang, H. Hawash, R. K. Chakrabortty, and M. Ryan, "Deep-IFS: Intrusion detection approach for IIoT traffic in fog environment," *IEEE Trans. Ind. Informat.*, vol. 17, no. 11, pp. 7704–7715, Nov. 2021.

[21] Y. Hua, R. Li, Z. Zhao, X. Chen, and H. Zhang, "GAN-powered deep distributional reinforcement learning for resource management in network slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 334–349, Feb. 2020.

[22] F. Núñez, S. Langarica, P. Díaz, M. Torres, and J. C. Salas, "Neural network-based model predictive control of a paste thickener over an industrial internet platform," *IEEE Trans. Ind. Informat.*, vol. 16, no. 4, pp. 2859–2867, Apr. 2020.

[23] A. K. Singh and R. Pateriya, "Analysis of crucial oil gas and liquid sensor statistics and production forecasting using IIOT and autoregressive models," *Int. Res. J. Eng. Technol.*, vol. 6, no. 4, pp. 124–129, Apr. 2019.

[24] S. Lele, "Oil-well flow-rate forecasting using auto-regressive model," Dec. 2019. [Online]. Available: https://ssrn.com/abstract=3502754

[25] J. C. Kabugo, S.-L. Jämsä-Jounela, R. Schiemann, and C. Binder, "Process monitoring platform based on industry 4.0 tools: A waste-to-energy plant case study," in *Proc. 4th IEEE Conf. Control Fault Tolerant Syst.*, 2019, pp. 264–269.

[26] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 7167–7176.

[27] C. G. Turhan and H. S. Bilge, "Recent trends in deep generative models: A review," in *Proc. 3rd Int. Conf. Comput. Sci. Eng.*, 2018, pp. 574–579.

[28] Z. Pan, W. Yu, X. Yi, A. Khan, F. Yuan, and Y. Zheng, "Recent progress on generative adversarial networks (GANs): A survey," *IEEE Access*, vol. 7, pp. 36322–36333, 2019.

[29] Z. Wang, Q. She, and T. E. Ward, "Generative adversarial networks in computer vision: A survey and taxonomy," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–38, 2021.

[30] N. Gao *et al.*, "Generative adversarial networks for spatio-temporal data: A survey," *ACM Trans. Intell. Syst. Technol.*, vol. 13, no. 2, Apr. 2022, Art. no. 22.

[31] R. A. Khalil, N. Saeed, M. Masood, Y. M. Fard, M.-S. Alouini, and T. Y. Al-Naffouri, "Deep learning in the industrial Internet of Things: Potentials, challenges, and emerging applications," *IEEE Internet Things J.*, vol. 8, no. 14, pp. 11016–11040, Jul. 2021.

[32] B. Jiang, J. Li, G. Yue, and H. Song, "Differential privacy for industrial Internet of Things: Opportunities, applications and challenges," *IEEE Internet Things J.*, vol. 8, no. 13, pp. 10430–10451, Jul. 2021.

[33] H. Akaike, "Fitting autoregressive models for prediction," *Ann. Inst. Stat. Math.*, vol. 21, pp. 243–247, 1969.

[34] B. Uria, M.-A. Côté, K. Gregor, I. Murray, and H. Larochelle, "Neural autoregressive distribution estimation," *J. Mach. Learn. Res.*, vol. 17, pp. 7184–7220, 2016.

[35] A. van den Oord, N. Kalchbrenner, O. Vinyals, L. Espeholt, A. Graves, and K. Kavukcuoglu, "Conditional image generation with PixelCNN decoders," 2016. [Online]. Available: http://arxiv.org/abs/1606.05328

[36] T. Salimans, A. Karpathy, X. Chen, and D. P. Kingma, "Pixelcnn: Improving the PixelCNN with discretized logistic mixture likelihood and other modifications," 2017. [Online]. Available: http://arxiv.org/abs/1701.05517

[37] A. Van Oord, N. Kalchbrenner, and K. Kavukcuoglu, "Pixel recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1747–1756.

[38] I. Gulrajani *et al.*, "PixelVAE: A latent variable model for natural images," 2016. [Online]. Available: http://arxiv.org/abs/1611.05013

[39] D. P. Kingma and M. Welling, "Auto-encoding variational Bayes," 2014, pp. 1–14.

[40] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1278–1286.

[41] D. P. Kingma and M. Welling, "An introduction to variational autoencoders," *Found. Trends Mach. Learn.*, vol. 12, no. 4, pp. 307–392, 2019.

[42] K. Sohn, H. Lee, and X. Yan, "Learning structured output representation using deep conditional generative models," *Adv. Neural Inf. Process. Syst.*, vol. 28, pp. 3483–3491, 2015.

[43] K. Gregor, I. Danihelka, A. Graves, D. Rezende, and D. Wierstra, "Draw: A recurrent neural network for image generation," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1462–1471.

[44] T. Salimans, D. Kingma, and M. Welling, "Markov chain monte carlo and variational inference: Bridging the gap," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1218–1226.

[45] T. D. Kulkarni, W. Whitney, P. Kohli, and J. B. Tenenbaum, "Deep convolutional inverse graphics network," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2539–2547.

[46] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised learning with deep generative models," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3581–3589.

[47] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 835–851.

[48] R. Ranganath, D. Tran, and D. Blei, "Hierarchical variational models," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 324–333.

[49] L. Maaløe, C. K. Sønderby, S. K. Sønderby, and O. Winther, "Auxiliary deep generative models," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1445–1453.

[50] D. Rezende and S. Mohamed, "Variational inference with normalizing flows," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 1530–1538.

[51] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improving variational inference with inverse autoregressive flow," in *Proc. Conf. Neural Inf. Process. Syst.*, 2016, pp. 4743–4751.

[52] J. M. Tomczak and M. Welling, "Improving variational auto-encoders using convex combination linear inverse autoregressive flow," in *Proc. 26th Benelux Conf. Mach. Learn.*, Jun. 2017, p. 162.

[53] C.-W. Huang, D. Krueger, A. Lacoste, and A. Courville, "Neural autoregressive flows," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2078–2087.

[54] N. De Cao, W. Aziz, and I. Titov, "Block neural autoregressive flow," in *Proc. Uncertainty Artif. Intell.*, 2020, pp. 1263–1273.

[55] M. Zhang, S. Jiang, Z. Cui, R. Garnett, and Y. Chen, "D-VAE: A variational autoencoder for directed acyclic graphs," in *Proc. Conf. Neural Inf. Process. Syst.*, 2019, pp. 1588–1600.

[56] A. Campbell and P. Liò, "tvgp-VAE: Tensor-variate gaussian process prior variational autoencoder," 2020, *arXiv:2006.04788*.

[57] I. J. Goodfellow *et al.*, "Generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[58] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *ICLR*, 2016, *arXiv:1511.06434*.

[59] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[60] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," in *Proc. 33rd Int. Conf. Mach. Learn.* , 2016, pp. 1558–1566.

[61] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.

[62] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2223–2232.

[63] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 7354–7363.

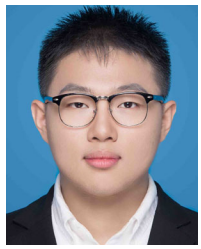
[64] G. Daras, A. Odena, H. Zhang, and A. G. Dimakis, "Your local GAN: Designing two dimensional local attention mechanisms for generative models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 14531–14539.
[65] A. Karnewar and O. Wang, "MSG-GAN: Multi-scale gradients for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7799–7808.
[66] S. Nowozin, B. Cseke, and R. Tomioka, "F-GAN: Training generative neural samplers using variational divergence minimization," in *Proc. 30th Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 271–279.
[67] Y. Xiangli, Y. Deng, B. Dai, C. C. Loy, and D. Lin, "Real or not real, that is the question," in *Proc. Int. Conf. Learn. Representations*, 2020.
[68] G.-J. Qi, "Loss-sensitive generative adversarial networks on Lipschitz densities," *Int. J. Comput. Vis.*, vol. 128, no. 5, pp. 1118–1140, 2020.
[69] A. Ferdowsi and W. Saad, "Generative adversarial networks for distributed intrusion detection in the Internet of Things," in *Proc. IEEE Glob. Commun. Conf.*, 2019, pp. 1–6.
[70] F. Aidin and S. Walid, "Brainstorming generative adversarial networks (bgans): Towards multi-agent generative models with distributed private datasets," 2020, *arXiv:2002.00306*.
[71] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special Lecture IE*, vol. 2, no. 1, pp. 1–18, 2015.
[72] W. Jiang, Y. Hong, B. Zhou, X. He, and C. Cheng, "A GAN-based anomaly detection approach for imbalanced industrial time series," *IEEE Access*, vol. 7, pp. 143608–143619, 2019.
[73] V. Belenko, V. Chernenko, M. Kalinin, and V. Krundyshev, "Evaluation of GAN applicability for intrusion detection in self-organizing networks of cyber physical systems," in *Proc. Int. Russian Automat. Conf.*, 2018, pp. 1–7.
[74] M. Lopez-Martin, B. Carro, A. Sanchez-Esguevillas, and J. Lloret, "Conditional variational autoencoder for prediction and feature recovery applied to intrusion detection in IoT," *Sensors*, vol. 17, no. 9, 2017, Art. no. 1967.
[75] P. T. Duy *et al.*, "DIGFuPAS: Deceive IDS with GAN and function-preserving on adversarial samples in SDN-enabled networks," *Comput. Secur.*, vol. 109, 2021, Art. no. 102367.
[76] P. R. Grammatikis, P. Sarigiannidis, G. Efstathopoulos, and E. Panaousis, "ARIES: A novel multivariate intrusion detection system for smart grid," *Sensors*, vol. 20, no. 18, 2020, Art. no. 5305.
[77] K. Merchant and B. Nousain, "Securing IoT RF fingerprinting systems with generative adversarial networks," in *Proc. IEEE Mil. Commun. Conf.*, 2019, pp. 584–589.
[78] Y. Yang, Y. Li, W. Zhang, F. Qin, P. Zhu, and C.-X. Wang, "Generative-adversarial-network-based wireless channel modeling: Challenges and opportunities," *IEEE Commun. Mag.*, vol. 57, no. 3, pp. 22–27, Mar. 2019.
[79] T. Li, S. Chen, Z. Yao, X. Chen, and J. Yang, "Semi-supervised network traffic classification using deep generative models," in *Proc. 14th Int. Conf. Natural Comput., Fuzzy Syst. Knowl. Discov*, 2018, pp. 1282–1288.
[80] K. Sonawane and S. Bojewar, "IIOT for monitoring oil well production and ensure reliability," in *Proc. 2nd Int. Conf. Adv. Sci. Technol.*, 2019.
[81] A. Singh and R. Pateriya, "Artificial oil lift production forecasting and analysis using autoregressive and deep learning models," May 2019. [Online]. Available: http://dx.doi.org/10.2139/ssrn.3396282
[82] Y. Huang, C.-H. Chen, and C.-J. Huang, "Motor fault detection and feature extraction using RNN-based variational autoencoder," *IEEE Access*, vol. 7, pp. 139086–139096, 2019.
[83] P. Tang, K. Peng, and J. Dong, "Nonlinear quality-related fault detection using combined deep variational information bottleneck and variational autoencoder," *ISA Trans.*, vol. 114, pp. 444–454, 2021.
[84] C. Schockaert, V. Macher, and A. Schmitz, "VAE-LIME: Deep generative model based approach for local data-driven model interpretability applied to the ironmaking industry," 2020, *arXiv:2007.10256*.
[85] A. Bagheri, I. Y. Gu, and M. H. Bollen, "Generative adversarial model-guided deep active learning for voltage dip labelling," in *Proc. IEEE Milan PowerTech*, 2019, pp. 1–5.
[86] Z. Wang, J. Wang, and Y. Wang, "An intelligent diagnosis scheme based on generative adversarial learning deep neural networks and its application to planetary gearbox fault pattern recognition," *Neurocomputing*, vol. 310, pp. 213–222, 2018.
[87] P. Liang, C. Deng, J. Wu, and Z. Yang, "Intelligent fault diagnosis of rotating machinery via wavelet transform, generative adversarial nets and convolutional neural network," *Measurement*, vol. 159, 2020, Art. no. 107768.
[88] H. Lee, S. Han, and J. Lee, "Generative adversarial trainer: Defense to adversarial perturbations with GAN," 2017, *arXiv:abs/1705.03387*.
[89] Y. Wu, M. Rosca, and T. Lillicrap, "Deep compressed sensing," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 6850–6860.

Suparna De (Member, IEEE) received the Ph.D. degree in electronic engineering from the University of Surrey, Guildford, England, U.K, in 2009.

She was a Postdoctoral Senior Research Fellow with the University of Surrey, and an Assistant Professor with the University of Winchester, Winchester, U.K. She is currently an Assistant Professor with the University of Surrey. Her research interests include deep learning for text data (derived from social networks and longitudinal social science datasets), semantic modeling/search, and data analytics.

Maria Bermudez-Edo received the Ph.D. degree in information and communication technologies from the University of Granada, Granada, Spain, in 2013.

She was a Research Fellow with the University of Surrey, Guildford, U.K., with two previous secondments to the University of California at Berkeley, Berkeley, CA, USA, and ETH Zurich, Zurich, Switzerland. She is currently an Associate Professor with the University of Granada. She was with the European Patent Office, The Hague, Netherlands; Siemens, Herentals, Belgium; and Telefonica, Madrid, Spain. Her research interests include semantics and data analytics in IoT, smart cities, and eHealth.



Honghui Xu received the bachelor's degree from the University of Electronic Science and Technology of China (UESTC), Chengdu, China, in 2019. He is currently working toward the Ph.D. degree in computer science with the Department of Computer Science, Georgia State University (GSU), Atlanta, GA, USA.

His research interests include machine learning and deep learning, including the fundamental theory of machine learning, the applications of deep learning in computer vision field, and the topic about privacy-preserving machine learning.

Zhipeng Cai (Senior Member, IEEE) received the B.S. degree from the Beijing Institute of Technology, Beijing, China, in 2001, and the M.S. and Ph.D. degrees from the University of Alberta, Edmonton, AB, USA, in 2004 and 2008, respectively, all in computer science.

He is currently an Associate Professor with the Department of Computer Science, College of Business, Georgia State University, Atlanta, GA, USA. His research interests include machine learning, Internet of Things, privacy, and Big Data.

Prof. Cai is currently the Editor-in-Chief for *Wireless Communications and Mobile Computing* and the Associate Editor-in-Chief for *High-Confidence Computing Journal* (Elsevier). He is the recipient of an NSF CAREER Award.