Generative Adversarial Networks (GANs) in Networking: A Comprehensive Survey & Evaluation

Hojjat Navidan[a], Parisa Fard Moshiri[b], Mohammad Nabati[b], Reza Shahbazian[c], Seyed Ali Ghorashi[b,d,*,1], Vahid Shah-Mansouri[a] and David Windridge[e]

[a]School of Electrical and Computer Engineering, College of Engineering, University of Tehran, Tehran 14395-515, Iran
[b]Cognitive Telecommunication Research Group, Department of Telecommunications, Faculty of Electrical Engineering, Shahid Beheshti University, Tehran 19839-69411, Iran
[c]Department of Electrical Engineering, Faculty of Technology and Engineering, Standard Research Institute, Alborz 31745-139, Iran
[d]School of Architecture, Computing and Engineering, University of East London, E16 2RD London, U.K
[e]Department of Computer Science, School of Science and Technology, Middlesex University, NW4 4BT London, U.K

ARTICLE INFO

Keywords:
Generative Adversarial Networks
Deep Learning
Semi-supervised Learning
Computer Networks
Communication Networks.

ABSTRACT

Despite the recency of its conception, Generative Adversarial Networks (GANs) constitute an extensively-researched machine learning sub-field for the creation of synthetic data through deep generative modeling. GANs have consequently been applied in a number of domains, most notably computer vision, in which they are typically used to generate or transform synthetic images. Given their relative ease of use, it is therefore natural that researchers in the field of networking (which has seen an extensive application of deep learning) should take an interest in GAN-based approaches. The need for a comprehensive survey of such activity is therefore urgent. In this paper, we demonstrate how this branch of deep learning and semi-supervised learning can assist multiple applications of computer and communication networks, including mobile networks, network analysis, internet of things, physical layer, and cybersecurity. In doing so, we shall provide a novel evaluation framework for comparing the performance of different models in non-image applications, applying this to a number of reference network datasets.

1. Introduction

Over the past few years, there has been an exponential growth of mobile networks. According to the Ericsson Mobility Report, there are more than 7 billion mobile broadband subscriptions extant in the world. With the rapid uptake of the fifth-generation (5G) network, it is estimated that by the end of 2025, this number will increase to 9 billion [1]. This growth has additionally led to an enormous rise in network and infrastructure demands. For example, 5G systems are designed to support massive traffic volumes, real-time network analysis, and agile management of resources; however, since mobile networks are heterogeneous, complex, and non-linear, meeting these requirements with classic methods and algorithms becomes challenging. It frequently transpires, though, that many of the problems encountered in computer and communication networks can be formulated as classification, detection, estimation, prediction, or optimization problems. Moreover, with advancements in processing and computing power, machine learning algorithms have increasingly demonstrated the ability to solve such problems more effectively than alternative approaches. Researchers are consequently proposing new algorithms and methods on a regular basis based on machine learning and related data-driven approaches in order to overcome these challenges [2, 3].

These learning algorithms, however, often require a considerable amount of training data to be effective. In many real-world problems, data accessibility is limited, or it is cost-expensive to gather a significant amount of data. In addition, typical learning algorithms are based on the assumption that the data are uniformly distributed. Again, many scenarios do not follow this principle, whereas data distribution is skewed towards some classes that appear more frequently. This problem, called the class imbalance, causes the algorithm to be biased towards some majority groups and hence, become inaccurate [4].

Furthermore, semi-supervised learning is a branch of machine learning that attempts to address the problem of partially labeled training data. Data labeling is frequently expensive and time-consuming, and label storage may present problems in some cases. Problems that involve diverse, unstructured and inter-connected datasets fall into this category [5]. Grappling with the problems of data shortage, class imbalance, and label propagation has naturally prompted consideration of generative approaches; that is, using innovative methods for generating the new data with the same properties as real-data in order to improve the performance of learning algorithms.

Deep generative models have emerged as one of the most exciting and prominent sub-fields of deep learning, given their remarkable ability to synthesize input data of arbitrary form by learning the distribution of data such that novel samples can be drawn. Deep generative models can consequently provide benefits in several ways. Firstly, supervised learning methods often require a substantial quantity of data in order to achieve good performance (in many real-case scenarios such as indoor localization,

*Corresponding author
ORCID(s): 0000-0002-2910-9208 (Seyed Ali Ghorashi)
[1]email: s.a.ghorashi@uel.ac.uk

Table 1
Full list of abbreviations in alphabetical order.

Abbreviation	Explanation
5G	Fifth-generation mobile network
ACGAN	Auxiliary Classifier Generative Adversarial Network
API	Application Programming Interfaces
AWGN	Additive White Gaussian Channel
BIGAN	Bidirectional Generative Adversarial Network
CGAN	Conditional Generative Adversarial Network
CNN	Convolutional Neural Network
CSI	Channel State Information
EMD	Earth Mover Distance
GAN	Generative Adversarial Network
GCN	Graph Convolutional Network
GIDS	GAN-based IDS
HAR	Human Activity Recognition
IDS	Intrusion Detection System
IoT	Internet of Things
KDE	Kernel Destiny Estimation
LSGAN	Least Square Generative Adversarial Network
LSTM	Long Short-Term Memory
MLP	Multilayer Perception
MMD	Maximum Mean Discrepancy
NN	Neural Network
PDF	Probability Density Function
QOE	Quality of Experience
RCGAN	Radio Classify GAN
RF	Radio Frequency
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
SAE	Sparse Autoencoder
SAGA	Spectrum Augmentation/Adaptation with GAN
SON	Self-Organizing Network
UAV	Unnamed Aerial Vehicles
WGAN	Wasserstein Generative Adversarial Network
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty
WSN	Wireless Sensor Networks

the requisite amount of data may not be accessible [6]); secondly, collection of large quantities of data may be infeasibly time-consuming or expensive. Besides alleviating difficulties in these single-domain scenarios, deep generative approaches have also demonstrated capability in transfer learning scenarios, where the correlation between two datasets is utilized in conjunction with generative models to transfer learning between datasets [7].

From the first introduction of Generative Adversarial Networks in 2014, GANs have been a focus of attention in generative machine learning (according to Google scholar, there are around 75000 papers based or focusing on GANs to date[2]). GANs have predominantly been used in computer vision, including but not limited to image generation, face synthesis [8], image translation [9, 10, 11], texture synthesis [12, 13], medical imaging, [14] and super-resolution [15]. Moreover, GANs can be applied in many other fields including but not limited to voice and speech signals [16, 17, 18], anomaly detection [19], power systems and smart grids [20, 21, 22], electronics [23, 24], and fault diagnosis [25, 26, 27, 28].

In line with this activity across multiple research fronts, there has been extensive recent research interest in applying generative adversarial networks to computer and communication networks; hence the need for a comprehensive survey covering the full extent of this new field of development.

1.1. Previous Work

A number of surveys and tutorials on GANs within the broader field of machine learning exist. For example, Cao et al. [29] introduced the most frequently-used GAN models and compared functionality with respect to a range of use cases. They conducted experiments to generate synthetic images from the two widely used image datasets, MNIST [30] and Fashion-MNIST [31], evaluating the performance of the different GANs models both visually and quantitatively. Other surveys are not limited, unlike this one, to the field of computer vision alone; as Wang et al. [32] also briefly reviewed the applications of GANs in other areas of interest, such as speech and language processing.

Table 2 summarizes the relationship between our survey and the other extant surveys and tutorials. As represented in Table 2, while most current survey works covering GANs extend across the field of computer vision, a few cover other fields. Our survey will thus seek to position itself in relation to these extant surveys by providing a detailed overview of adversarial networks' applicable to the field of computer and communication networks. In doing so, we shall introduce several novel evaluation metrics that can be deployed in relation to the generation of non-image in order to evaluate the performance of different GANs models.

1.2. Key Contributions

To the best of our knowledge, there is no survey or tutorial specifically discussing recent developments of GANs in relation to computer and communication networks. As indicated above, the majority of the research done in this area covers computer vision and image processing. This fact motivates us to provide a comprehensive survey and review of recent relevant researches carried out in the field of networking. We additionally provide an evaluation framework to measure and compare the performance of the respective GAN models. The key contributions of this survey paper are summarized as follows:

[2]November 2020

Table 2
Comparison of current survey with existing surveys and tutorials. "CV" and "NET" refer to computer vision and networking.

Existing Publications	Technical Overview				Applications Overview		
	Models Intro.	Models Comp.	Eval. Metrics	Simulation	CV	NET	Others
Pan et al. [33]	✓	✓	✓		✓		
Turhan and Bilge [34]	✓	✓					
Cao et al. [29]	✓	✓	✓	✓	✓		
Goodfellow [35]	✓				✓		
Gonog and Zhou [36]	✓	✓			✓		✓
Zhang et al. [37]	✓	✓					
Wu et al. [38]	✓				✓		
Wang et al. [32]	✓	✓			✓		✓
Shorten and Khoshgoftaar [39]	✓	✓			✓		
Esfahani and Latifi [40]	✓				✓		
Creswell et al. [41]	✓	✓			✓		
Di Mattia et al. [19]	✓	✓		✓			✓
Our Survey	✓	✓	✓	✓		✓	✓

- We provide a comprehensive but compact background concerning deep generative models, with an emphasis on GANs. While different GANs variants have a similar underlying mechanism, network architectures significantly differ. We therefore compare and contrast network architectural components in detail.
- We discuss the range of applications that have benefited from GANs in the literature categorized into five main categories, depending on whether the GANs in question produce synthetic data for semi-supervised learning or else utilize the generator or the discriminator network in a unique way. For each such category, we describe in detail how GANs seek to provide a solution for some of the existing challenges.
- Inspired by evaluation metrics for image data, we provide a framework for comparing the performance of differing GAN models trained on four network datasets of different types. Furthermore, we visualize the data distributions and utilize statistical tools to compare the similarity between them. To the best of our knowledge, this is the first time that the performance of GANs has been evaluated comparatively for non-image data; since GANs have predominantly been used in computer vision, principally for the generation of images.
- We discuss the open challenges of GANs and indicate how addressing these challenges would serve to contribute to further improvements in networking. We end by suggesting some research directions and areas with considerable potential to benefit from a GAN-based approach but have not exploited their potential.

1.3. Survey Organization

The remainder of this paper is organized as follows: We begin by providing a background concerning deep generative methods in section 2, in particular Generative Adversarial Networks (GANs). We provide details of some of the state-of-the-art GAN networks for non-image data. Next in section 3, we review recent GAN applications in computer and communication networks, grouped under mobile networks, network analysis, internet of things, physical layer, and cybersecurity. In section 4, we introduce a framework for evaluating performance of different GAN variants and conduct experiments to compare various state-of-the-art models such as CGAN, LSGAN, INFOGAN, and WGAN. Finally, we discuss current challenges and future work to conclude the paper.

2. Deep Generative Models

A generative model is defined as any model that can represent an estimation of the given data probability distribution by drawing samples from it. These models either result in a distribution that estimates the original model explicitly or else generates samples from the original data without defining a distribution [35]. They have a wide area of applications, including reinforcement learning [42] and inverse reinforcement learning [43] in which they are used to simulate possible scenarios for multi-modal learning [44]. However, the predominant application of these models is the generative filling-in of missing data and data imputation. They can hence be utilized in many scenarios, such as semi-supervised learning, in which only a portion of training data is labeled. As most modern deep learning models and algorithms require extensive labeled examples for training, semi-supervised learning provides a ready solution for reducing the labeling requirement. GANs, in particular, have found

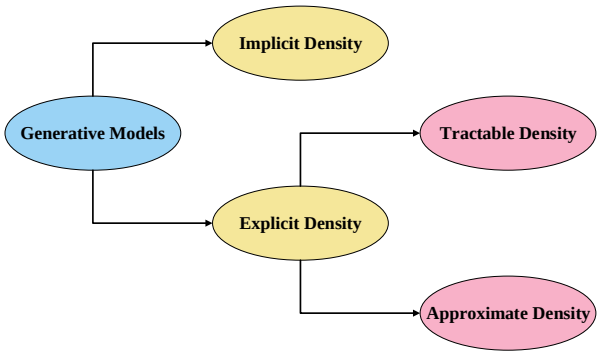

Figure 1: Overall taxonomy of generative models.

extensive use in semi-supervised learning [45].

The overall taxonomy of generative models is depicted in Fig.1. In general, generative models can be divided into two main types: explicit density and implicit density models. Explicit density models are those that provide an explicit parametric specification of the data distribution. The main challenge here is capturing all the complexity of the data while maintaining computational tractability. Therefore, explicit density models are, in turn, divided into two sub-groups. Firstly, there are the models that define computationally-tractable density functions, such as deep belief networks [46] or flow models [47]. These models allow us to use an optimization algorithm directly on the log-likelihood of training data and hence be highly effective; however, they also have intrinsic limitations, resulting in a range of practical drawbacks depending on the data distribution. Secondly, there are the models with intractable density functions that use approximations, either variational (e.g., variational autoencoders) or Monte Carlo based (e.g., Boltzmann machines), to maximize the likelihood. In contrast to these two explicit subclasses, implicit density models do not specify the distribution of data and thus do not require a tractable likelihood but rather seek to define a stochastic process that aims to draw samples from the target data distribution. GANs are implicit density generative models of this latter kind which are able to generate data samples in a single step [35].

2.1. Generative Adversarial Networks

In 2014, Goodfellow et al. [48] introduced a novel adversarial class of generative models, GANs, which aim to produce synthetic data with maximal similarity to the original data. The GAN model consists of two main aspects, Generator and Discriminator; the idea behind this model being intrinsically game-theoretic, albeit within deep learning context. The generator hence has the role of a counterfeiter, aiming to deceive the discriminator. Countering this, the discriminator plays a policing role that aims to recognize the counterfeits. Consequently, both the generator and discriminator learn from each other in developing their capabilities. After termination of the learning stage, the generator is a fully-trained counterfeiter able to maximally mislead the 'police,' while the discriminator is maximally trained to realize counterfeits. As this survey's primary focus is on GANs, further details and methodological variants will be covered in the next subsection.

2.2. GAN Variations and Architectures

The model initially proposed by Goodfellow (from now called Vanilla GAN) suffers from a few significant problems, such as vanishing gradients, mode collapse, and a frequent failure to converge [49]. Many solutions have since been proposed to overcome these challenges: some focus on improving the training methods to prevent these problems from occurring, while others constitute entirely different architectures. Examples of the latter include Conditional GAN (CGAN) [50], CycleGAN [10], Bidirectional GAN (BiGAN) [51], DualGAN [11], DiscoGAN [52], Pix2Pix [9], InfoGAN [53], Energy-based GAN (EBGAN) [54], Wasserstein GAN (WGAN) [55], and Super-Resolution GAN [15]. Out of these proposed models, only a few have been used for data other than images; including but not limited to Vanilla GAN, BIGAN, CGAN, InfoGAN, CycleGAN, EBGAN, and Least Square GAN (LSGAN) [56]. The other models are either exclusively designed for image data or else have not been used in applications other than computer vision. For example, Deep Convolutional GAN (DCGAN) is a robust variant of GANs that utilizes Convolutional Neural Networks (CNNs) to generate high-quality images [57]. However, as they contain a CNN, they are only applicable where the data's spatial features are of importance, such as images. Below, we will introduce in more detail seven general-use models described in the literature, especially those relevant to the current review case. The architectures of these models are depicted in Fig.2.

We shall first describe the learning process of the Vanilla GAN in Fig.2. As may be seen, the architecture of this model consists of two components, the G and D networks. Arbitrary latent noise spikes in the G network are used to generate fake samples. Afterwards, the D network compares the fake and real samples to increase its own discrimination performance. The process of learning these two networks is hence based on the minimax cost function, which is defined as follows:

$$\min_G \max_D \; L(D,G) = E_{x\sim p_{data}(x)}[\log D(\mathbf{x})] + E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z}))], \tag{1}$$

where $p_{data}(x)$ is the probability distribution of the data and $p_z(z)$ is the prior noise distribution.

This cost function, which aims to maximize the probability of correctly assigning labels to real and fake data, consists of two parts. We can implement both G and D, which are differentiable functions, via an Multilayer Perception (MLP). The generator thus seeks to map the latent noise $\mathbf{z} \sim p_z(z)$ to the real data distribution, which is denoted via the $G(\mathbf{z};\theta_g)$, where θ_g indicates the parameters of the generator. Simultaneously, the discriminator is trained to distinguish between

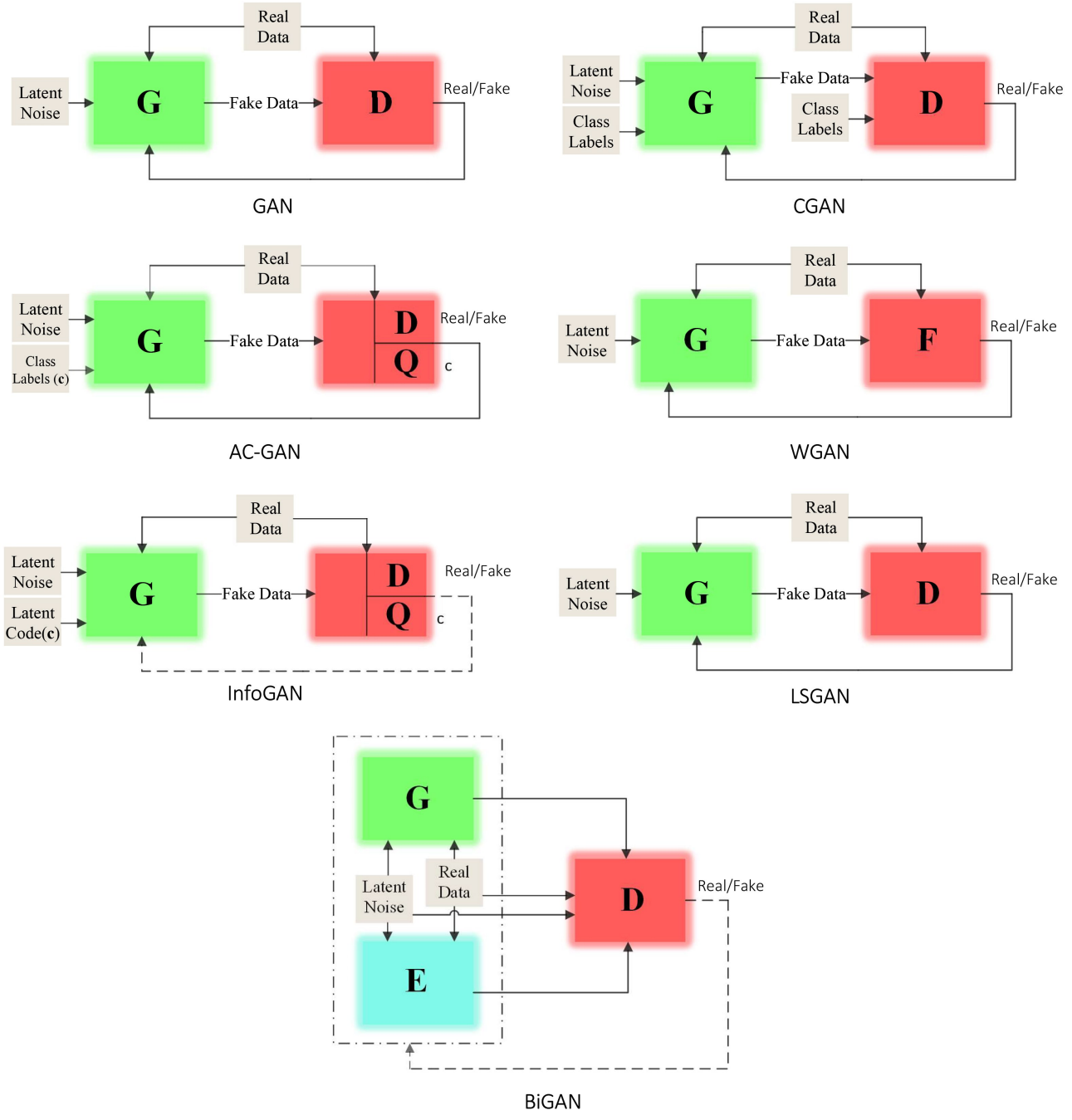

Figure 2: Architecture of the GANs reviewed in this section. G refers to the generator network, which is present in all of the above models. D refers to the discriminator network while E, F, and Q refer to the encoder, critic, and the Q-network respectively.

real and fake data, denoted via the $D(\mathbf{x};\theta_d)$, where θ_d indicates the parameters of the discriminator. Naturally, the discriminator's output is binary, where 0 and 1 specify fake and real data, respectively. By fixing the respective sides relating to each model, the opposing model's cost function can be represented. Hence, the discriminator's loss is defined as follows:

$$\begin{aligned} L(\theta_d) =& E_{x\sim p_{data}(x)}[\log D(\mathbf{x};\theta_d)] \\ &+ E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z};\theta_g))]. \end{aligned} \tag{2}$$

It should be noted that since the first term of Eq. (1) does not affect the generator when it is fixed, it disappears in the gradient updating step. Similarly, the loss function of the generator is defined as follows:

$$L(\theta_g) = E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z};\theta_g))]. \tag{3}$$

The process of updating θ_d and θ_g is depicted in Algorithm 1 [6]. Convergence occurs when $D(\mathbf{x};\theta_d) = 0.5$, which means that discriminator is no longer able to distinguish between real and fake data. Once the algorithm has converged,

Algorithm 1 GAN learning process

Initiate e = epochs, s = step size for updating θ_d, learning rates (η_g and η_d)
for $i = 1 : e$ **do**
  for $t = 1 : s$ **do**
    sample batch $\mathbf{Z} \in \mathbf{R}^{B\times L} \sim \mathbf{p}_z(\mathbf{z})$
    sample batch $\mathbf{X} \in \mathbf{R}^{B\times M}$ from data
    update θ_d by gradient ascent based optimizer
    $L(\theta_d) = \frac{1}{B}\sum_{\mathbf{b}=1}^{\mathbf{B}} \left[\log D(\mathbf{X}_b,\theta_d) + \log(1 - D(\mathbf{G}(\mathbf{Z}_b,\theta_g))\right]$
    $\psi_d = \frac{\partial}{\partial\theta_d} L(\theta_d)$
    $\theta_d^{t+1} = \theta_d^{t} + \eta_d \psi_d$
  end for
  sample batch $\mathbf{Z} \in \mathbf{R}^{B\times L} \sim \mathbf{p}_z(\mathbf{z})$
  update θ_g by gradient descent based optimizer
    $L(\theta_g) = \frac{1}{B}\sum_{b=1}^{B} \log(1 - D(G(\mathbf{Z}_b,\theta_g))$
  $\psi_g = \frac{\partial}{\partial\theta_g} L(\theta_g)$
  $\theta_g^{t+1} = \theta_g^{t} - \eta_g \psi_g$
end for
$\mathbf{X}_b$ and $\mathbf{Z}_b$ are b'th rows of $\mathbf{X}$ and $\mathbf{Z}$, respectively.

the generator can then produce synthetic data by random sampling of the same prior noise distribution $\mathbf{z} \sim p_z(z)$, such that the discriminator is not able to distinguish whether these data are real or not. This model is hence the baseline structure of the vast majority of GANs model variants that researchers have introduced in recent years.

2.2.1. Conditional GAN

The Vanilla GAN can only generate data via the given inputs and cannot generate samples with labels simultaneously. Therefore, Mirza and Osindero [50] proposed Conditional GAN (CGAN), which feeds relevant additional information to the generator and discriminator sides by presenting the encoded class labels alongside the prior noise and real data, respectively. The cost function, very similar to the Vanilla GAN, is defined as follows:

$$\begin{aligned}\min_G \max_D \; L(D,G) =& E_{x\sim p_{data}(x)}[\log D(\mathbf{x}|\mathbf{y})] \\ &+ E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z}|\mathbf{y}))].\end{aligned} \tag{4}$$

Consequently, the discriminator's loss can be formulated as:

$$\begin{aligned}L(\theta_d) =& E_{x\sim p_{data}(x)}[\log D(\mathbf{x}|\mathbf{y};\theta_d)] \\ &+ E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z}|\mathbf{y};\theta_g))],\end{aligned} \tag{5}$$

and the generator's loss is defined as:

$$L(\theta_g) = E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z}|\mathbf{y};\theta_g))]. \tag{6}$$

2.2.2. Auxiliary Classifier GAN

Odena et al. [58] propose a variant of the GAN architecture named Auxiliary Classifier GAN (ACGAN). In ACGAN, the generated samples have a corresponding class label $\mathbf{c} \sim p_c(c)$ alongside the noise parametrization $\mathbf{z} \sim p_z(z)$. The generator uses both of these to generate synthetic samples. Besides being responsible for distinguishing real and fake data, the discriminator must also carry out task (Q), predicting the class labels. The significant difference between ACGAN and CGAN is that CGAN has conditioning on class labels at the input of the generator and the discriminator to generate data for each class. ACGAN, however, predicts class labels via a multi-task architecture consisting of a paired source and label loss as follows:

$$\begin{aligned}L_S &= E_{x\sim p_{data}(x)}[\log D(\mathbf{x})] + E_{z\sim p_z(z)}[\log(1 - D(G(\mathbf{z}))] \\ L_C &= E_{x\sim p_{data}(x)}[\log Q(\mathbf{c}|\mathbf{x})] + E_{z\sim p_z(z)}[\log(Q(\mathbf{c}|\mathbf{z})].\end{aligned} \tag{7}$$

The generator hence aims to maximize $L_S + L_C$, while on the other hand the discriminator seeks to maximize $L_S - L_C$.

2.2.3. Wasserstein GAN

The Vanilla GAN suffers from vanishing gradient and convergence problems making the training stage inconvenient. Various solutions have been proposed in recent years to overcome these issues. Arjovsky et al. [59] suggest adding additional noise to the generated samples to better stabilize the model before presentation to the discriminator. The same group also proposed a novel cost function [55] to deal with instability problems, the resulting model being the Wasserstein GAN (WGAN) in which the cost function is as given as:

$$L(F,G) = \sup_{||F||_L\leq 1} E_{x\sim p_{data}(x)}[F(\mathbf{x})] - E_{z\sim p_z(z)}[F(G(\mathbf{z}))], \tag{8}$$

where "sup" is the supremum over all the 1-Lipschitz functions with the constraint $\left|F(x_1) - F(x_2)\right| \leq \left|x_1 - x_2\right|$ [60].

2.2.4. WGAN-GP

Wasserstein GAN improves convergence and presents a solution to instability problems in the Vanilla GAN. However, it suffers from undesirable behavior of critic weight clipping in the training stage. Gulrajani et al. [61] add a gradient penalty term to the cost function of WGAN to overcome this bottleneck, the resulting model being referred to as 'WGAN with Gradient Penalty' (WGAN-GP), which has the cost function:

$$\begin{aligned}L(F,G) = \sup_{||F||_L\leq 1} & E_{x\sim p_{data}(x)}[F(\mathbf{x})] - E_{z\sim p_z(z)}[F(\underbrace{G(\mathbf{z})}_{\tilde{x}})] \\ &+ \lambda E_{\hat{x}\sim p_{\hat{x}}(\hat{x})}\left[\left(\left\|\nabla_{\hat{x}} F(\hat{x})\right\|_2 - 1\right)^2\right], \\ &\hat{x} = \rho\tilde{x} + (1-\rho)x \quad \textit{and} \quad 0 \leq \rho \leq 1.\end{aligned} \tag{9}$$

2.2.5. InfoGAN

Chen et al. [53] proposed a new GAN framework for generating samples via the addition of conditional factor information to the generator noise input in a completely unsupervised manner. For example, if the goal is to generate

the digits in the MNIST dataset, it would ideally be the case that the generative system has access to independent factor variables representing the digits' thickness and angle as part of the latent noise space. Such additional information can be denoted via $c_1, c_2, ..., c_L$, and given that we are assuming that the latent variables are independent, the joint distribution can be written as $P(c_1, c_2, ..., c_3) = \prod_{i=1}^{L} P(c_i)$. In the following, c is the concatenation of all c_i variables. The InfoGAN cost function is, therefore:

$$\min_G \max_D \; L_{\text{Info}}(D, G) = L(D, G) - \lambda I(\mathbf{c};G(\mathbf{z}, \mathbf{c})), \quad (10)$$

where $L(D, G)$ is as defined in Eq. (1), $I(\mathbf{c};G(\mathbf{z}, \mathbf{c})) = H(\mathbf{c}) - H(\mathbf{c}|G(\mathbf{z}, \mathbf{c}))$, and H is the entropy. Practically, the term $I(\mathbf{c};G(\mathbf{z}, \mathbf{c}))$ is hard to maximize directly as we would need to access to the posterior distribution $P(\mathbf{c}|\mathbf{x})$. However, a lower bound can be obtained variationally by defining a new structure $Q(\mathbf{c}|\mathbf{x})$, known as the auxiliary distribution. This lower bound is given as:

$$\begin{aligned} I(\mathbf{c};G(\mathbf{z}, \mathbf{c})) \geq & E_{c\sim p(c),\, x\sim G(z,c)}[\log Q(\mathbf{c}|\mathbf{x})] + H(\mathbf{c}) \\ = & E_{x\sim G(z,c)}[E_{c'\sim p(c|x)}[\log Q(\mathbf{c}'|\mathbf{x})]] + H(\mathbf{c}) \\ = & L_I(G, Q). \end{aligned} \quad (11)$$

Thus, the loss function can be represented as:

$$\min_{G,Q} \max_D \; L_{\text{Info}}(D, G, Q) = L(D, G) - \lambda L_I(G, Q). \quad (12)$$

2.2.6. Least Square GAN

Mao et al. [56] propose a novel loss function for addressing the vanishing gradient problem inherent in GANs (and deep learning generally) during the learning stage by replacing the cross-entropy loss by the least-squares loss. The new cost function for the discriminator can thus be written:

$$\begin{aligned} \min_D \; L(D) = & E_{x\sim p_{data}(x)}[(D(\mathbf{x}) - 1)^2] \\ & + E_{z\sim p_z(z)}[(D(G(\mathbf{z})) - 1)^2]. \end{aligned} \quad (13)$$

The loss function for the generator is similarly given as:

$$\min_G \; L(G) = E_{z\sim p_z(z)}[(D(G(\mathbf{z})) - 1)^2]. \quad (14)$$

2.2.7. Bidirectional GAN

The Vanilla GAN maps from the latent noise space to the real data distribution; however, the standard GAN framework is not reversible and cannot map data to a latent layer representation. Donahue et al. [51] consequently propose an unsupervised GAN framework, the Bidirectional GAN (BiGAN), which can map the real data distribution $\mathbf{x}$ to the latent noise domain $\mathbf{z}$ via a new Encoder structure that sits alongside the Generator and Discriminator. The objective function is defined as follows:

$$\begin{aligned} \min_{G,E} \max_D \; L(D, E, G) = & E_{x\sim p_{data}(x)}[E_{z\sim p_E(.|x)} \log D(\mathbf{x}, \mathbf{z})] \\ & + E_{z\sim p_z(z)}[E_{x\sim p_G(.|x)} \log(1 - D(G(\mathbf{x}, \mathbf{z}))]. \end{aligned} \quad (15)$$

These are hence the principle GAN architectures of current interest across the machine learning domain. We now look to the networking domain specifically.

3. Applications Overview

We divide the use-cases of GANs in the literature into five main categories: mobile networks, network analysis, internet of things, physical layer, and cybersecurity. In each section, we shall briefly introduce the subject, focusing on the most common challenges faced by researchers prior to investigating how semi-supervised learning can address these challenges. In each case, we provide real-world examples. A summary of the reviewed applications is given at the end of the chapter.

3.1. Mobile Networks

With the rapid development of mobile networks over the past decades, classical methods are unable to meet modern network demands. Therefore, researchers are looking into new methods, such as machine learning or novel optimization techniques, to keep up with the pace of demand and requirement changes. Zhang et al. [2] provide a detailed and broad-based review of deep learning methods and their application in wireless and mobile networks. Since our primary focus is on GANs, we review, in this subsection, their most notable applications in mobile networks.

3.1.1. 5G Network Slicing

The fifth-generation mobile network is designed to meet the requirements of a highly mobile and fully connected society, enable automation in various industry sectors, and provide a viable infrastructure for 'internet of things' applications. The traffic characteristics of these autonomously communicating devices are significantly different from human-made traffic. Hence, 5G networks are required to support very diverse functionality and performance requirements in order to offer the various services with reliability. To this end, the notion of *network slicing* is defined as a composition of network functions, network applications, and the underlying cloud infrastructure joined together so as to meet the requirements of a specific use case. Network slicing enables sharing a common infrastructure to deploy multiple logical, self-contained, and independent networks [62]. With the aid of network slicing and creating various types of virtual networks, 5G can provide services with complex and dynamic time-variable resource requirements. Within this paradigm, however, different slices may have different resource demands, and the demands of a slice can be dynamic and vary during its operation time. Therefore, the need to predict user requirements concerning the different resources and the requirement of dynamically allocating these resources become crucial to the operation of 5G.

Gu and Zhang [63] proposed *GANSlicing*, a dynamic software-defined mobile network slicing framework for prediction of the resource demands of the internet of things (IoT) applications and also for improving the Quality of Experience (QoE) of users. GANSlicing aims to generate a

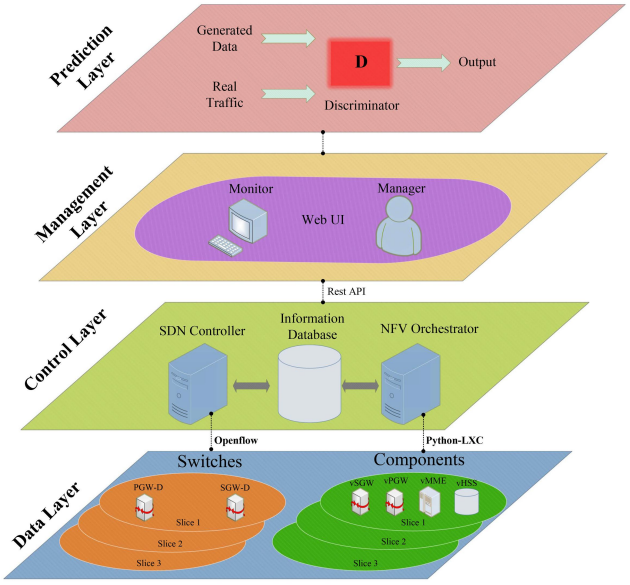

Figure 3: The architectural scheme of GANSlicing [63] for predicting slice demand by users.

global view of network resources that considers the physical and virtual capabilities of cellular networks to achieve more efficient utilization and allocation of resources to the network slices dynamically. A third use-case is the prediction of user demands for a variety of different resources via the underlying deep generative model. A GAN, which can generatively mimic an administrators' network operations, can also, in principle, predict traffic flow from historical information. Hence, GANslicing allows for slice demand to be forecast; so that the overall resource utilization is enhanced. The architecture of this model may be seen in Fig.3.

GANSlicing is implemented in two parts; service-oriented slicing and GAN-based prediction. Evaluation of the accuracy of network traffic prediction, and analysis of the performance of the slicing scheme (which the authors compared with tenant-oriented slicing, the most common scheme in mobile networks) indicates that GANSlicing can accept 16% more requests with 12% fewer resources in the same service request batch, hence improving the service acceptance ratio and enhancing overall service quality [63].

3.1.2. Self-Organizing Networks

Maintaining wireless and cellular networks' functionality and service provisioning while reducing low capital expenditure and operating expenditure has been challenging for both operators and service providers. Self-Organizing Networks (SONs) [64, 65] consist of a set of functions for automating the planning, configuration, management, and optimization of mobile networks. Since SON's primary function is to learn the parameters of a network and then optimize them, they inherently rely on data processing and intelligent decision making; consequently, learning algorithms and methods can be utilized to achieve this goal [66, 67]. These methods include, but are not limited to, unsupervised learning [68], deep learning [69], Q-learning [70] and Deep Q-learning [71]. However, most of these algorithms require gathering a large amount of real-case data to be effective. This becomes a challenge for many reasons; firstly, in many scenarios, the amount of data available for specific scenarios can often be limited, as such scenarios may be hard to reproduce. Secondly, gathering sufficient data can be costly or otherwise constrained, especially in scenarios where time is a limiting factor. Finally, a major challenge is data imbalance, given the infrequency of certain classes of events, which the SON must nonetheless take into account.

Recently, researchers have applied generative algorithms to overcome the aforementioned challenges, proposing various methods to improve SON performance and efficiency through synthetic data generation in order to increase the available quality of training data. For example, Zhang et al. [72] investigated the application of traditional classification algorithms for cell outage detection in SONs. Since cell outage is a relatively rare and low probability event, they typically constitute only a tiny fraction of total network measurement data. Due to this data imbalance, traditional learning algorithms will tend to construct a biased classifier. Consequently, the classifier output exhibits a skewness towards the majority class. The authors consequently proposed a novel cell outage detection scheme by combining Vanilla GAN with adaptive boosting (Adaboost). By utilizing GANs to generate synthetic data for the minority classes, they were able to correct this imbalance so that Adaboost can then be used to classify the re-balanced data and effectively detect cell outage.

Ben Hughes et al. [73] presented a further application of GANs to SONs, seeking to augment Call Data Records obtained from a mobile operator. These data records exhibit two main features; call duration and start hour, for which the authors were able to generate synthetic tabular data. It is notable in this study that the reported difference in the variance of the generated and real data was more significant than that of the difference in mean values, a consequence of the fact that GANs often fail to generate realistic outlier values. They nonetheless achieved an accuracy improvement of 3.43%.

3.2. Network Analysis

Network analysis is the act of collecting network data and analyzing it in order to improve the overall performance, reliability, and security of the network. These data usually consist of packets, log files, and configuration data. Computer and communication networks typically require real-time data delivery. However, much of this data is unstructured and unprocessed; consequently, the requirement for efficient tools and means of data analysis becomes critical.

Since machine learning methods have the benefit of leveraging statistical patterns in data to perform analysis tasks with little pre-programming, they have received much attention in network analysis; one such common task is the analysis and control of communication networks. Machine learning methods, however, require large volumes of data

to have acceptable performance in this context. Since networks tend to be well distributed, this large volume of data must be collected from several points in the network so that the learning algorithm can achieve a sufficiently global perspective on the network. Aho et al. [74], provided a novel approach to generate synthetic live traffic in order to improve the robustness of the learning algorithms applied to network analysis by applying several GAN variants. In particular, they utilized adversarial networks to generate network traffic similar to original samples. By comparing evaluation metrics for five different GAN architectures, Vanilla GAN, CGAN, LSGAN, WGAN, and WGAN-GP, they concluded that for network data generation, GAN and LSGAN are impractical, while WGAN and its variants offer significant performance gains.

Network traffic classification is typically one of the first steps in network analysis. In order to provide better service quality and also for management and security purposes, service providers must establish the different types of network application. Due to the massive volume of network data being transmitted, this task is usually automated. In general, there are two types of method applied to achieve this task; classification using payloads of packets and classification based on statistical analysis [75]. Researchers have consequently utilized various learning algorithms for traffic classification over the last few years [76]. For security and user privacy purposes, many applications use network protocols such as SSL or VPN to encrypt their traffic. However, this encryption makes the analysis and classification of network data a challenging task. One of the major challenges in encrypted traffic classification is class imbalance, given that the majority of network traffic is regular and unencrypted traffic. To this end, Wang et al. [77] proposed FlowGAN, a method that uses GANs to generate synthetic traffic data for classes that suffer from low sample counts. They then used an MLP classifier to evaluate the effectiveness of their method, finding that tackling the class imbalance problem in this manner can indeed increase the performance traffic classifiers.

Li et al. [78] proposed FlowGAN (not to be confused with the FlowGAN proposed in [77]), a novel dynamic traffic camouflaging method to mitigate traffic analysis attacks and circumvent censorship. The idea behind this method is to use a GAN to learn features of permitted network flow (the target) and morph on-going censored traffic flows (the source) based on these features, in such a way that the morphed traffic is indistinguishable from the real flow. The authors exploited WGAN for the generator and WGAN-GP for the discriminator, evaluating the resulting method on more than 10,000 network flows, the data consisting of 6 features: outgoing packets, incoming packets, byte counts of outgoing packets, byte counts of incoming packets, cumulative bytes and the average interval between packets. Since traffic analysis attacks are principally a classification problem between different traffic data, the authors evaluated their method using area under curve and Indistinguishability under Classification Attack (IND-CA), defined in [78] as:

$$IND-CA = \frac{|\Pr[\mathrm{Priv}K = 1] - 0.5|}{0.5}, \tag{16}$$

where $\mathrm{Priv}K = 1$ if the attacker can distinguish between traffic flows.

Dynamics play a very significant role in the performance of most network systems; therefore, it is crucial to predict dynamics while performing network analysis. For instance, in an ad-hoc network, the dynamics of communication links make designing routing protocols challenging. Lei et al. [79] formulated the dynamics prediction problem of various network systems as temporal link prediction tasks, where abstracted dynamic graphs describe the system's behavior. They proposed a novel non-linear model, GCN-GAN, to predict these links in a weighted dynamic network. This model combines a Graph Convolutional Network (GCN) and a Long Short-Term Memory (LSTM) network with a GAN to improve representation learning and generate high-quality and plausible graph snapshots. By using GCN and LSTM as hidden layers of the generator network, the generator is able to predict the subsequent snapshots based on the historical topology of the dynamic graph.

Social network analysis is a particular subgroup of network analysis, *social tie prediction* being a quintessential problem in which social network operators attempt to suggest new connections or products to users based on their current social activity. Chen et al. [80] proposed TranGAN, a GAN-based transfer learning method for social tie prediction that seeks to uncover latent information in social networks. TranGAN, inspired by Triple-GAN [81], in addition to the usual generator and discriminator structure, utilizes an additional Neural Network (NN) classifier for assigning labels to output samples from the generator. Although in this transfer learning scenario, the source and the target network are from different domains, the composite system is able to use information from the source network to improve the performance of the target network. The source network contains well-labeled social relationships; the labels of these relationships are missing in the target network, and hence the two are heterogeneous.

As indicated, since most current network security systems utilize ML-based algorithms to perform network analysis, the data requirement needed to train these systems can become problematic. While simulations may be able to generate sufficient data, these are typically laborious and time-consuming to create. Xie et al. [82], proposed utilizing existing network attack data generation tools augmented with data generated by WGAN. While simple and effective, this method is only able to generate continuous network features, since WGAN does not perform well in generating discrete features such as "protocol type" "flag" and "service."

3.3. Internet of Things

3.3.1. Wireless Sensor Networks

As an intermediate layer between wireless sensor networks (WSNs) and the end-user, middleware can provide

a solution to various design issues, including security, heterogeneity, and performance scalability. One of the key challenges in WSNs is security, such that the confidentiality, authenticity, and integrity of data transmission from sensors can be guaranteed [83]. However, most middleware approaches cannot fully guarantee these security properties and protect the network from malicious attacks. Alshinina and Elleithy [84] proposed a unique WSN middleware, powered by a GAN, for overcoming these design challenges while providing appropriate security measurement for handling large scale WSNs. This proposed method, in common with other GANs, consists of a generator and a discriminator. The generator creates fake data with similar attributes to the real data to confuse would-be attackers (the WSN does not need to generate fake data in this case so that power consumption can be significantly reduced). On the other hand, the discriminator is tuned to distinguish real data and detect anomalies for further processing.

3.3.2. Indoor Localization and path planning

With the exponential growth of smartphones and wearable technologies over the last decade, demand for location-based services has significantly increased. Currently, widely used localization services such as the global positioning system, while demonstrating excellent performance in ideal outdoor environments, can perform poorly in indoor or harsh outdoor environments. This may be caused by many phenomena: fading, shadowing, multipath, and a lack of line-of-sight [85]. For this reason, other techniques with high accuracies, such as fingerprinting, are generally used for indoor localization. Currently, due to its high availability and ease of access, WiFi-based fingerprinting is the most commonly used method for indoor localization [86]. The fingerprinting method consists in two phases; a training (offline) phase, where signal feature measurements are initially collected, and a test (online) phase, in which real-time signal properties are measured and compared against the offline phase data in order to provide an accurate estimation of the desired location. The two primary signal properties used for fingerprinting are Received Signal Strength (RSS) and Channel State Information (CSI) [85]. Machine learning and NNs have recently been widely used to learn probability features from these signal properties to perform localization.

However, such ML-based techniques face a crucial challenge in a lack of, or shortcoming in, manually-labeled data for training. Moreover, in many cases, data collection is costly and time-consuming (for instance, in crowd-sourcing, a large number of human participants are typically required to collect and annotate data via their mobile phones). Nabati et al. [6] proposed to address this issue through the use of GANs to learn the underlying distribution of collected Received Signal Strength Indicator (RSSI) datasets in order to generate synthetic data and increase the amount available during the offline phase. Their proposed method, as well as reducing the cost of data collection, achieved identical benchmark accuracy with a lower real data requirement in a shorter time (they can use as little as 10% of the real data with the associated reduction in data collection costs while achieving the same accuracy levels).

Li et al. [87] converted collected CSI data using complex wavelet transforms in order to create feature maps that can extend the acquired fingerprint database through the use of the Wavelet Transform Feature DCGAN. This approach accelerates the convergence process during the training phase and is able to increase the variety of generated feature maps substantially. Experimental results indicated that the method can generate CSI data with improved diversity. The corresponding increase in the number of samples in the training set allows for significantly better localization accuracy, thereby demonstrating the proposed model's superiority to existing fingerprint database construction methods.

Mohammadi et al. [88] investigated path planning, proposing a GAN architecture to suggest accurate and reliable paths for differing applications, such as wayfinding for disabled or visually-impaired people. This approach utilizes a GAN to generate paths to a destination via collected user trajectory data. This involves two components: localization and wayfinding. The GAN structure uses a feed-forward NN at the input layer with four hidden layers and a hyperbolic tangent activation function as the generator; the discriminator is similarly a feed-forward NN. The localization aspect's training process constitutes a multi-label classification task, accomplished via the classification of RSSI values into location coordinates. A separate model was created for each class feature and trained on the common training dataset. The path classifier was evaluated separately as it required truth values to determine how well the model can identify classes. Experimental results suggested that paths generated by the model are more than 99.9% reliable, with the path classifier able to classify the given path with around 99% accuracy.

3.3.3. Human Activity Recognition

Human Activity Recognition (HAR) has a significant role in many IoT applications such as smart houses, health care, and elderly monitoring. Within this domain, many approaches involve motion sensors, cameras, and WiFi signals. However, motion sensors, though accurate, may be expensive and impractical, and wearing them on the person proves detrimental to user well-being. Using image signals gathered from cameras is also frequently unviable, given that they are unable to work in darkness or non-line-of-sight settings. This often leaves WiFi signals as a very useful supplementary, or even the best, HAR option. However, straight WiFi-based HAR, due to the low-resolution and limited sensing capability of RSSI measurements, is often unable to achieve fine-grained HAR. This has prompted recent studies to propose CSI measurements as a way to achieve recognition performance [89].

Aiming to leverage WiFi signal's pervasiveness without the requirement for specialized equipment, Yousefi et al. [90] applied CSI to the problem of recognizing seven distinct human activities. Each action has a distinct pat-

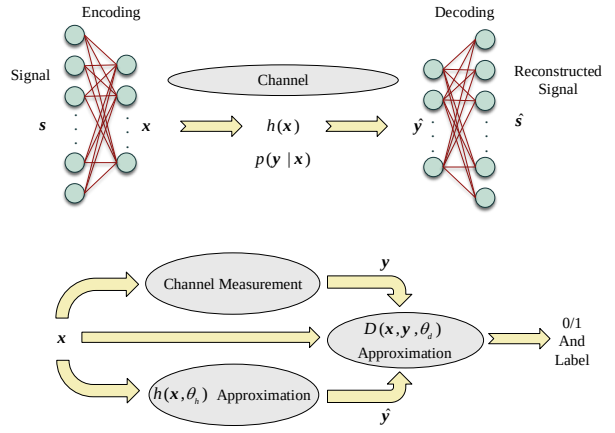

Figure 4: (a) Modeling an end-to-end communication system with autoencoders, as proposed in [93]. (b) approximating the stochastic channel function using GANs.

tern, and the distinct motions should therefore have differing effects on the CSI. However, owing to the low signal-to-noise ratio, raw CSI measurements are not in themselves sufficiently representative of these different human activities. Rather than hand-craft discriminative features, the authors propose an LSTM-based approach to learn representative features to encode the temporal information. However, Fard Moshiri et al. [91] indicated that it is difficult to collect adequate labeled data for training the proposed LSTM model, thus they proposed a semi-supervised GAN-based solution instead. Hence, they attempted to generate an augmented dataset with the same statistical features as the real data by presenting 50% of each activity class to the GAN model, with their proposed model improving classification accuracy and scaling-down the Log Loss gives an overall accuracy improvement of 3%.

Xiao et al. [92] applied leave-one-subject-out validation to CSI-based activity recognition to address the performance degradation problem. Using a GAN-based framework, termed CsiGAN, they conducted experiments on two CSI-based behavior recognition datasets; SignFi, which includes CSI traces concerning sign language gestures, and FallDefi, which includes CSI traces concerning a range of typical human activities including falling, walking, jumping, picking-up, sitting-down, and standing-up. The semi-supervised GAN used in this paper extends the standard GAN discriminator by adjusting the number of probabilistic outputs from $k+1$ into $2k+1$ (where k is the number of categories), which helps in obtaining the correct decision boundary for each category. They also proposed a manifold regularization method to enhance classification performance by stabilizing the learning process.

3.4. Physical Layer

Deep learning has been broadly adopted in physical layer communications, especially for signal-related processes, including encoding, decoding, modulation, and equalization. It has shown exceptional capabilities in removing various constraints of existing communication systems, such as in wireless channel modeling. Traditional channel modeling methods are exceedingly complex, unique to the channel environment, and are typically unable to model the channel's key stochastic properties. Using NNs, however, allows us to overcome these drawbacks. For instance, an end-to-end communication system can be represented as an autoencoder as shown in Fig.4(a). This approach consists of an encoder encoding symbols into a distinct transmitted value, a stochastic channel model, and a decoder network which seeks to estimate the transmitted symbols from the received samples and outputs a probability distribution over all possible decoded messages [93].

O'Shea et al. [94] proposed a method for physical layer modulation and coding for the communication system that uses adversarial learning. They thus employed GANs with channel autoencoders to approximate the channel's response and learned an optimal scheme under certain performance metrics. This approach, which they termed Communications GAN, utilizes two separate networks for encoding and decoding, both consisting of a fully connected layer with ReLu activation. Using the mean squared error as a measure of channel loss, normalization and noise interpolation were the main focus of training. Their results suggested that by learning a channel function approximation and an encoder/decoder layout, robust performance without explicit prior implementation can be achieved. Such a system can learn directly on unseen physical channels, an approximation of these channels being sufficient for adapting the encoder and decoder networks.

The approach described above, although model-free, assumes that the channel model function is differentiable. Otherwise, gradients could not be computed during the back-propagative training of the network. However, since we do not have access to an exact channel model in reality, this has to be estimated. Analytic channel models can only express a limited number of wireless channel effects (such as interference, propagation, distortion, noise, and fading); this is because expressing non-linear effects is laborious due to their complexity and high number of degrees of freedom. O'Shea et al. [95] extend their approach so as to represent a broad range of stochastic channel effects with a high degree of accuracy. The channel model is a stochastic function; hence it can be modeled as a conditional probability $p(y|x)$. Similarly, the channel approximation network can also be modeled as a conditional probability distribution, $p(\hat{y}|x)$ where $\hat{y}$ and y represent synthetic and real samples, respectively, and x is the channel input. The goal here is to minimize the distance between $p(y|x)$ and $p(\hat{y}|x)$ such that the model approximation becomes more accurate. This setup, as shown in Fig. 4(b), is achieved via a new discriminator network (the authors asserted that this task could also be performed using a WGAN in order to improve training stability).

In this vein, Ye et al. [96] proposed an end-to-end channel-agnostic communication model that can be applied to more realistic time-varying channels. They utilized CGAN to model the conditional distribution $p(y|x)$, where

the transmitter's encoded signal constitutes the conditioning information. By adding pilot information to the conditioning information, the system is able to generate more specific samples and estimate the CSI more accurately. This approach enables end-to-end learning of a communication system without prior information regarding the channel. In other words, by training a conditional GAN, the transmitter, receiver, and end-to-end loss can be well-optimized in a supervised manner. The authors initially applied their method to the Additive White Gaussian Channel (AWGN) channel, in which the output, y, is a summation of the input signal, x, with Gaussian noise, w, such that $y = x + w$. In this case, the conditioning information is the encoded signal from the transmitter since channel estimation is not required. In a further experiment, Rayleigh fading channels were studied, for which the output is given via $y_n = h_n x_n + h_n$ where $h_n \sim CN(0, 1)$; since the channel is time-varying, additional conditional information needs to be appended to the channel receiver. The authors assert that the system may be further extended to other types of channels, beyond those of AWGN and Rayleigh fading.

Autoencoder-based communications systems have become pervasive in the research community. In most cases, researchers use the encoder as a transmitter that maps messages to symbols. However, the problem is that the gradients of the physical channel are often obscure, and this circumstance prevents the transmitter network from receiving updates during training. One solution is to approximate the channel response using a NN, such that the NN can act as a substitute for the physical channel during the training process. An early example of this method was presented in [95], where the authors used a GAN to approximate a stochastic channel that includes non-linear distortions and non-Gaussian statistics. Results indicated the utility of the GAN; however, the Probability Density Function (PDF) of the learned channel model differs from the simulated channel's PDF, and the channel is further assumed to have no memory effect. Smith and Downey [97], inspired by BicycleGAN [7], evaluated a novel GAN architecture for learning nonlinearities, memory effects and non-Gaussian statistics. Their research focuses on channels that contain a combination of non-linear amplifier distortion, pulse shape filtering, inter-symbol interference, frequency-dependent group delay, multipath, and non-Gaussian statistics. They compared the marginalized PDFs of the channel with a trained generator. Carrying out experiments on four different channels, as shown in Fig.5, their results suggest that the proposed model is capable of generating high-accuracy approximations of the channel.

Yang et al. [98] proposed an alternative channel modeling framework utilizing GANs. The GAN in question was trained using raw channel measurement data, for which the Nash equilibrium is the convergence of a minimax game between the generator and discriminator. The generator thus learns the distribution of channel impulse responses and generates synthetic samples. Once the equilibrium point is reached, the generator can be extracted as the target channel

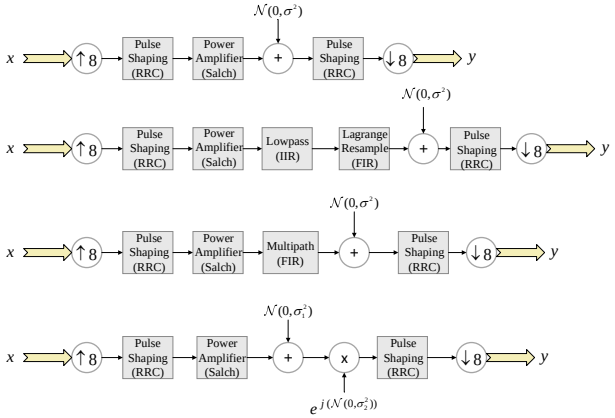

Figure 5: The different models tested in [97].

model. To evaluate the suggested framework's performance and verify the effectiveness of the method, they estimated the channel response of an AWGN channel, comparing the PDF of the learned channel model with the real AWGN channel. The authors suggested that this framework could, in principle, be extended to large-scale channels, such as multiple-input/multiple-output.

Zhao et al. [99] studied the detection, tracking, and classification of Unnamed Aerial Vehicles (UAVs). They used an oscilloscope and antenna to collect wireless signals in an indoor environment with a sample rate of 20 GS/s. For the corresponding outdoor environment, they utilized a universal radio software peripheral that uses IEEE 802.11g for communication with 20 MHz bandwidth (since the bandwidth of the oscilloscope is around 2.5 GHz). They then analyzed the UAVs' wireless signal features via modified principal component analysis for dimension reduction. They hence used wireless signals to detect UAVs in a manner that does not depend on the size, line of sight, protocol standardization, or level of forensic tool support. Leveraging the high recognition rate of the ACGAN and the theory of WGAN, they then proposed a more stable model variant, AC-WGAN, that exhibits a classification rate greater than 95% in indoor environments. Training samples were introduced to both the discriminator and generator with the negative loss updated by ascending a stochastic gradient. Testing samples were then imported into the discriminator, and the taxonomy of the signals was classified based on the value of the negative loss. This model is reportedly capable of detecting UAV wireless signals in outdoor environments from several hundred meters away.

3.4.1. Cognitive Radio

Cognitive radio is a concept that aims to overcome the limitations of wireless channels by making radios programmable, dynamically configurable, and capable of learning and adapting to minimize user interference and enhance overall performance. Using machine learning techniques enables cognitive radios to learn and make decisions without the need for explicit programming. Learning

methods in cognitive radio have been widely surveyed in the literature, for example, [100, 101] which investigated the various methods applied in this area. In general, cognitive radio problems can be divided into two categories: classification (e.g. for spectrum sensing or modulation recognition [102]), and decision-making (such as in power control or adaptive modulation [100]). Thus far, researchers have applied GANs only in the former category, and the applicability of GANs for decision-making problems is yet to be determined.

As indicated, machine learning is useful in automating cognitive radio functionalities by offering a means of reliably extracting and learning the intrinsic spectrum dynamics. There are two challenges in this task: firstly, the machine learning algorithm requires a significant amount of data in order to capture multiple channels and emitter characteristics to train the classifier. Secondly, the wireless channel is highly dynamic, and consequently, as the spectrum varies, training data previously identified for one spectrum environment cannot be reused in any such altered environment. To address these challenges, Davaslioglu and Sagduyu [103] proposed Spectrum Augmentation/Adaptation via a novel GAN, SAGA, which utilizes CGAN to generate synthetic training data. SAGA thus aims to leverage training data augmentation and domain adaption to improve classifier accuracy and enable the classifier to operate in novel, unseen environments. The authors asserted that SAGA can further be extended to wideband spectrum sensing, in which multiple channels are present, by applying training data augmentation and adjustment.

Existing methods for modulation recognition present in the literature are mainly based on deep learning, given that modulation recognition is inherently a classification problem. In this context, Li et al. [104] proposed Radio Classify GAN (RCGAN), a novel end-to-end semi-supervised framework for modulation recognition. By utilizing DCGAN with a cost function and replacing the last layer of the discriminator network with a softmax function, they were able to classify radio signals presented in the form of complex time-domain vectors to achieve modulation recognition. Furthermore, their experimental results suggested that this novel approach can improve overall recognition accuracy even when the signal-to-noise ratio is under 4dB.

Before the connection between two transceivers being established, wireless signals are required to be authenticated at the physical layer. One form of wireless attack that targets this task is *signal spoofing*. In this type of attack, the adversary aims to impersonate a legitimate transmitter. This is usually done to bypass authentication systems or primary user emulation in cognitive radio, in which a secondary user mimics a primary user to occupy more of the spectrum. Shi et al. [105] used GANs to spoof wireless signals by generating and transmitting false signals. They assumed that a deep classifier is used at the receiver to predict the intentional transmission such that, if there was no attack, a pre-trained deep learning-based classifier could discern signals. In a standard spoofing system, such as a replay attack, the probability of success with respect to the deep classifier stays confined. However, the authors show that a GAN-based spoofing attack, in which generated signals are transmitted for the spoofing attacks, has the potential to enhance the success probability of wireless signal spoofing even when a NN classifier has been used as a defense mechanism.

Erpek et al. [106] investigated the security aspects of cognitive radio in case of wireless jamming attacks. They described different types of wireless jamming attacks and applied adversarial learning to design both the jamming attack and also an appropriate defensive scheme. Jamming is severely dependent on the training data to give appropriate information regarding the channel status. However, when a jammer can only collect limited data, its performance drops significantly. Therefore, they proposed that by using CGAN effectively, they can overcome this performance drop and shorten the learning period, hence making the exploratory jamming attack more efficient. Their results showed that by using just ten samples instead of the full 500 and utilizing this to generate synthetic data with CGAN, their misdetection probability and false alarm rate stays within 0.19% and 3.14% of that of the original data, which significantly reduces the overall time and cost of gathering data.

Covert communication is a novel communication paradigm with a low chance of being detected or intercepted [107]. Since the two players in a GAN contend against each other to achieve the Nash equilibrium, GANs can be used to model an adversarial game between a legitimate user and a watchful warden in covert communications. In such a scenario, the generator, acting as a legitimate user, can be utilized to generate a covert transmit power allocation solution. Simultaneously, the discriminator can act as the warden and attempt to figure out the covert messages [108]. Liao et al. [108] considered a cooperative cognitive radio network, which benefits from a secondary transmitter acting as a relay. In this case, this secondary transmitter covertly transmits private information while being supervised by the primary transmitter. This scheme, termed GAN-based Power Allocation (GAN-PA), aims to seek a balance between the covert rate and the detection error. Experimental results suggested that the proposed scheme achieves near-optimal performance in the presence of minimal network status.

3.5. Cybersecurity

Cybersecurity is a complex of technologies, practices, and processes aimed to protect computers, networks, devices, and data from arbitrary cyber-attacks, unauthorized access, or malicious activity. Deep Learning has recently been widely applied to cybersecurity systems, for instance, in [109, 110, 111, 112, 113]. Because of their nature and capacity to alleviate the challenge of imbalanced datasets, GANs have been identified as having high potential in security and adversarial applications. We hence review GAN applications in Intrusion Detection Systems (IDSs) [114, 115, 116, 117], malware detection [118, 119], detection of rogue Radio Frequency (RF) transmitters

[120], malware adaption/improvement [121, 122, 123], black-box Application Programming Interfaces (API) attacks [124] and other cybersecurity applications such as password guessing [125] and credit-card fraud detection [126, 127, 128].

3.5.1. IDSs, Malware Detection, and Security Systems

IDSs play an essential role in maintaining network security. Their main task is to monitor network traffic and provide a defense against unusual and malicious traffic. Usama et al. [114] pointed out the vulnerability of IDSs towards adversarial examples and generative adversarial attacks in order to evade IDS detection. The GAN-based attack they envisaged adds perturbation to traffic features so that the IDS would be unable to detect it as malicious traffic. They further expanded this idea, proposing a GAN-based defense to increase the robustness of IDSs to this kind of attack.

Similarly, a novel framework based on WGAN called IDSGAN is proposed in [116] to generate adversarial attacks capable of evading IDS detection. This framework consists of a generator, a discriminator, and a black-box IDS based on the fact that the IDS structure is unknown to the attackers in a real-case scenario. IDSs are also an essential part of in-vehicle networks. However, such IDSs require very high accuracy as any error in detection may seriously endanger driver and passengers' safety. Seo et al. [117] proposed a GAN-based IDS (GIDS) for detecting unknown attacks using normal data; since GIDS constitutes a pre-trained model with two discriminators, one to detect known attacks and the other to detect unknown attacks, it may be applied as a real-time intrusion detector with excellent performance reported.

Data imbalances can occur in anomaly-detection problems since, in typical datasets, the anomalous instances are rare compared to the normal class. A learning model trained on such a dataset will naturally favor the majority class and perform poorly. GANs show potential in addressing such imbalances and are thus valuable in scenarios where gathering or generating anomalies is costly and time-consuming. Salem et al. [115] investigated this in the domain of Host-based IDSs (HIDSs), where normal data is abundant as compared to anomalies. They made use of the ADFA-LFD dataset [129], and after converting the numeric data to images, utilize a Cycle-GAN model to learn the transformation between normal and anomaly data in order to generate anomalies. In this way, by creating a framework to transform normal data into anomalies and adding these anomalies to the original dataset, they can significantly increase the performance of HIDSs.

Malware is defined as any application that exhibits malicious behavior, such as viruses, worms, Trojans, and ransomware. In contrast, benign applications are legitimate programs that are not harmful and perform their intended actions with the full acknowledgment of the user. Generally, the primary purpose of any anti-malware application is to distinguish between malware and benign applications. Amin et al. [118] investigated Android operating system malware, proposing a novel malware detection method that uses GANs with LSTM hidden layers (LSTM-GAN) in both the generator and discriminator. The system's input consists in binary opcode sequences extracted from different applications, represented as 1D tensors. After training a sufficient number of epochs, the discriminator may then be used as a malware detector. Kim et al. [119] suggested pre-training the generator using an autoencoder to overcome instability during training. This model, called transferred GAN, was then used to detect malware, particularly zero-day attacks for which only a tiny amount of data is available.

Roy et al. [120] addressed the problem of identifying rogue RF transmitters with the help of adversarial networks. All transceivers display a unique unwanted in-phase and quadrative imbalance (IQ imbalance) [130]. Therefore, by exploiting the IQ imbalance in RF transmitters, they proposed that Vanilla GAN can learn and generate unique features, using them as fingerprints to identify and classify transmitters. Thus, the generator model generates fake signals to spoof the transmission of known transmitters, and the discriminator detects these rogue transmitters. After training this model using over-the-air data collected from trusted transmitters, the discriminator can detect fake transmitters with about 99.99% accuracy.

Shin et al. [131] proposed a GAN-based model to defend against attacks aimed at the Android pattern-lock system. They suggested a Vanilla GAN based anomaly detection paradigm utilizing only single-user data that would generate a large amount of synthetic data, which may then be treated as the potential attacker during the training phase. To enhance stability during training, they used a Replay Buffer to generate high-quality synthetic data. Expanding this idea further, they introduced a multi-modal network that uses two-touch features (trajectory and pressure), resulting in a robust anomaly detector that proves effective against this form of attack.

3.5.2. Malware Adaption and Improvement

In contrast to the foregoing, GANs may also be used to increase the robustness of malware. Rigaki and Garcia [121] utilized GANs to enable malware to adapt to changing conditions and hence become harder to detect. Their proposed method used LSTM-GAN to learn the features of benign application traffic flow (in this case, Facebook chat) in order to imitate them. After convergence, the malware then connects to the generator and by using the output parameters, adapts its traffic accordingly. Experimental results found that with just 217 real-case network flows trained for 400 epochs, the blocked malware percentage dropped to zero.

Malware usually possesses either very little or no information about the structure and parameters of IDS models, and hence the target system may be treated as a black-box. However, it is possible to predict what features these systems use. Hu and Tan [122] proposed a novel algorithm, MalGAN (depicted in Fig. 6), which generates complex and flexible adversarial malware examples to attack and attempt to fool the black-box detector. Once the generator is trained on sufficient data, it can generate adversarial examples having

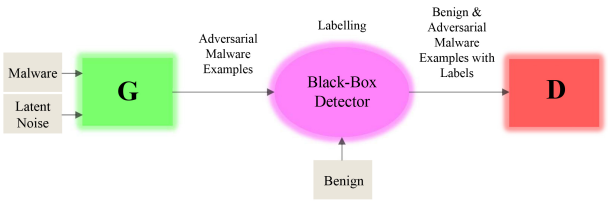

Figure 6: MalGAN architecture [122].

probability distributions far from that which the black-box detector is trained on, therefore classified as benign. Experimental results showed that, with a NN substitute detector as the black-box, almost all of the adversarial examples generated by MalGAN bypass the detection algorithm.

However, MalGAN does suffer from few issues; first, malware detectors must be built internally. Secondly, the feature dimension is reduced to just 128, meaning that not all original malware features are covered. Thirdly, both MalGAN and the detector use the same features. Lastly, multiple malware instances are required to train MalGAN [123]. Kawai et al. [123] hence suggested a number of enhancements, proposing Improved MalGAN to further evade detection by adding benign features to the original malware. They also addressed the issue of the high quantity of generated data required to avoid detectors by adding a loss calculation layer to the generator.

Black-box API attacks [124] constitute a set of exploratory attacks that can be launched to learn proprietary information such as underlying training data, learning algorithms, and hyperparameters, without any prior knowledge. Service providers usually limit the number of calls each user can make to the API to prevent such attacks. Consequently, the amount of training data the attacker obtains will necessarily be trivial. Shi et al. [132] suggested implementing these attacks with the aid of adversarial networks, demonstrating that even with a minimal quantity of training samples, GANs are successful in attacking.

3.5.3. Other Security Applications

Credit card fraud detection with machine learning requires a balanced dataset consisting of both standard and fraudulent transactions. However, in real-world scenarios, fraudulent transactions are far rarer than standard ones. This motivates the use of adversarial networks. For instance, [126, 127] addressed this challenge from a variety of perspectives. Sethia et al. [128] applied Vanilla GAN, WGAN, RWGAN, MAGAN [133], and LSGAN, with results suggesting that RGWAN provides the best performance, and vanilla GAN the worst due to mode collapse. Chen et al. [126] proposed a bespoke solution that uses a Sparse Autoencoder (SAE) to map regular transactions into a compact vector space with the GAN then seeking to generate synthetic standard transactions as well as identifying whether a given transaction is fraudulent or not. Hence, the generator generates fake standard transactions, and the discriminator attempts to distinguish between the generated data and the hidden representations learned from standard transactions by SAE. The final trained discriminator can then be used to detect fraudulent transactions. Wang et al. [127] also provided a data enhancement model based on adversarial networks called SGAN in order to generate synthetic fraudulent data.

Despite passwords being one of the most popular authentication methods, various password database leaks have demonstrated that users frequently choose simple passwords composed of common strings and numbers. Password guessing tools can hence be used as weak-password identifiers with passwords being stored in hashed form. State of the art password guessing tools, such as HashCat and John the Ripper, can check very large numbers of passwords per second against these password hashes. While such methods are very successful in practice, developing and testing new rules and the associated reconnaissance of user habits is time-consuming. To address this, Hitaj et al. [125] proposed PassGAN, an approach that seeks to replace human-generated password rules with theory-grounded machine learning algorithms. PassGAN, based on WGAN-GP, exploits an adversarial network to learn the distribution of real passwords from actual password leaks to generate superior password guesses. Experimental results suggested that the number of matches increases steadily with the number of generated passwords. PassGAN is able to guess between 51% and 73% of new unique passwords than comparable tools, matching passwords that were not generated by any password tools. The authors indicated that training PassGAN on a larger dataset would need more complex NN structures, and consequently, a more comprehensive training regime; it would also be required to produce a larger quantity of passwords compared to other tools. However, they assert that these costs are negligible concerning the benefits of the method.

In summary, Table 3 provides a brief overview of the applications reviewed above with appropriate taxonomic categories appended. We highlight the main problems and challenges that the GANs in question aimed to overcome and the specific models used to achieve this task. By inspection of the table, it may be seen that GANs have, to date, principally been used within the various cybersecurity and physical layer fields. Most of the work done includes applications with limitations of data gathering or class imbalance within the dataset. However, there are also a few novel works that utilize the discriminator network to achieve classification tasks.

4. Evaluation Framework

Thus far, we have reviewed the extant literature on GANs and their computer and communication networks application. In this section, we propose a suite of methods for measuring GAN performance with respect to our case study applications. We further train five state-of-the-art GAN models on four network-related datasets of different types and shapes and use the proposed methods amongst other visualization tools to evaluate models' performance.

Table 3
List of network-related research papers that utilize GANs, along with their respective categorization.

Class	Reference	Application	Problem	Used Model
Mobile Networks	Gu and Zhang [63]	Network Slicing in 5G	Dynamicity of demands in network slicing Predicting resource requirements of different users	Vanilla GAN
	Zhang et al. [69]	Cell outage detection	Imbalanced cell outage data in cellular networks	Vanilla GAN and Adaboost
	Hughes et al. [73]	Generating synthetic CDR	Requirement of large amount of data	Vanilla GAN
Network Analysis	Aho et al. [74]	Generating real network traffic	Requirement of a large volume of data for network analysis tools	Vanilla GAN, LSGAN, EBGAN, WGAN, WGAN-GP
	Wang et al. [77]	Traffic classification	Class imbalance in the classification of encrypted traffic	Vanilla GAN
	Li et al. [78]	Traffic camouflaging	Mitigating traffic analysis attack and circumventing censorship	WGAN and WGAN-GP
	Lei et al. [79]	Network link prediction	Dynamics in network systems	Vanilla GAN
	Chen et al. [80]	Social tie prediction	Lack of annotations in social network links and connections	Triple-GAN
	Xie et al. [82]	Network attack data generation	Network security systems require a large amount of data to perform network analysis	WGAN
Internet of Things	Alshinina and Elleithy [84]	Wireless Sensor Networks	Improve the security of middleware in WSNs Reduce power headroom	DCGAN
	Nabati et al. [6]	Indoor Localization	Shortcoming of real data in WiFi fingerprinting localization methods Cost and time consumption of collecting data	Vanilla GAN
	Li et al. [87]	Indoor Localization	Lack of diversity in the gathered CSI data Speed of convergence in the training phase and accuracy	DCGAN
	Mohammadi et al. [88]	Path Planning	Generating paths in wayfinding applications for disabled people	Vanilla GAN with a Classifier
	Fard Moshiri et al. [91]	Human Activity Recognition	Costs and time consumption of collecting CSI data for HAR	Vanilla GAN
	Xiao et al. [92]	Human Activity Recognition	Low accuracy of general approaches for left-out users	Vanilla GAN and CycleGAN
Physical Layer	O'Shea et al. [94]	Approximation of Channel Response	Complexity of modeling wireless channels	Vanilla GAN
	O'Shea et al. [95]	Approximation of Channel Response	Extension of [94], learning the PDF of channels	Vanilla GAN
	Ye et al. [96]	End-to-End Communication Systems Model	End-to-end learning of a system without prior information Can be applied to more realistic channels	CGAN
	Smith and Downey [97]	Channel Density Estimation	Learning non-linearities, memory effects and non-Gaussian statistics in channels	BicycleGAN
	Yang et al. [98]	Wireless Channel Modeling	Difficulty and low accuracy of traditional channel modeling methods	Vanilla GAN
	Zhao et al. [99]	UAV Classification	Previous methods to classify small UAVs required a large amount of samples for feature extraction.	AC-WGAN
	Davaslioglu and Sagduyu [103]	Spectrum Sensing	Dynamicity of wireless channels Shortcoming of data for training a model for all environments	CGAN
	Li et al. [104]	Modulation Recognition	Increasing the robustness of previous automatic modulation recognition frameworks	Modified DCGAN
	Shi et al. [105]	Wireless Signal Spoofing	Generate spoofing signals that are indistinguishable from intended signals	Vanilla GAN
	Erpek et al. [106]	Wireless Jamming	Dependency of jamming on the amount of collected training data	CGAN
	Liao et al. [108]	Covert Communication Systems	Modeling the adversarial game in covert communication with GANs and utilizing them for power allocation	Vanilla GAN
Cybersecurity	Usama et al. [114]	Intrusion detection systems	Vulnerability of machine learning models to adversarial perturbations	Vanilla GAN
	Lin et al. [116]	Intrusion detection systems	Increasing robustness of intrusion detection systems	WGAN
	Seo et al. [117]	Intrusion detection systems in vehicles	Lack of security features in CAN bus, reducing the false-positive error rate in vehicle IDS	Vanilla-GAN
	Salem et al. [115]	Host-based intrusion detection system	Data imbalance and anomalies in host-based intrusion data sources, Increasing robustness of HIDS	Cycle-GAN
	Amin et al. [118]	Malware detection in Android	Growth of Android operating system malware, limitations of learning-based malware diagnosis techniques	LSTM-GAN
	Kim et al. [119]	Malware detection	Classification and detection of zero-day attacks, detecting malware with a small amount of data	tGAN
	Roy et al. [120]	Rogue RF transmitter detection	Limitations of classical machine learning methods in detection of RF malicious activity	Vanilla GAN
	Shin et al. [131]	Android pattern lock system	Increasing the security of android pattern lock system by using the discriminator for anomaly detection	Vanilla GAN
	Rigaki and Garcia [121]	Malware improvement	Avoiding intrusion detection systems by modifying malicious traffic	LSTM-GAN
	Hu and Tan [122] Kawai et al. [123]	Generating adversarial malware examples	Bypassing black-box machine learning-based detection methods, Decreasing the detection rate of malware	Vanilla GAN
	Shi et al. [132]	Black-box API attacks	Limited amount of training data due to limited access to the objective API	CGAN
	Chen et al. [126]	Credit card fraud detection	Highly skewed datasets with little fraudulent data	SAE and GAN
	Sethia et al. [128]	Credit card fraud detection	Highly class-imbalanced data	Vanilla GAN, LSGAN, WGAN, MAGAN RWGAN
	Wang et al. [127]	Credit card fraud detection	Highly class-imbalanced data	Vanilla GAN
	Hital et al. [125]	Password guessing	Difficulties of implementing classic password guessing tools	WGAN-GP

Table 4
Parameters of the five different GANs evaluated in the experiment.

	Vanilla GAN	CGAN	BIGAN	LSGAN	WGAN
Loss	Binary Crossentropy	Binary Crossentropy	Binary Crossentropy	Mean Squared Error	Wasserstein Loss
Optimizer	Adam	Adam	Adam	Adam	RMSprop
Learning Rate	0.0002	0.0002	0.0002	0.0002	0.00005
Latent Dimension	100				
Batch Size	64				
Epochs	5000				

4.1. Evaluation Metrics

Evaluating and comparing the performance of GANs and other generative methods has always been a challenging task for researchers. Since GANs were mainly introduced for image data, the simplest and most straightforward evaluation method is a visual examination by humans, which is highly biased and subjective (different human judges are typically asked to look at pictures and vouch for the quality of generated images). However, this is inapplicable to data that does not readily fit human sensory categories. Other qualitative and quantitative assessment methods are available, however; Borji [134] presents 24 quantitative and 4 qualitative methods for evaluating and comparing GANs. The majority of these methods, such as Inception Score [49], Mode Score [135] and Fréchet Inception Distance [136] are only applicable to image data. From [134] and [137] we can conclude that out of these 28 proposed measures, only a small subset, including Average Log-likelihood [48, 138], Wasserstein Distance [55] and Maximum Mean Discrepancy (MMD) [139] are appropriate for non-image GAN data.

The task of evaluating generative methods is equivalent to measuring the dissimilarity between p_r and p_g (respectively, the probability distributions of samples drawn from real and generated data). In the case in which both of these distributions are known *a priori*, Kullback-Leibler Divergence (or Log-likelihood) and Jensen-Shannon divergence may be used, which are respectively defined as follows:

$$D_{KL}(p_r||p_g) \stackrel{\Delta}{=} \sum_{x\in X} p_r(x)\log\left(\frac{p_r(x)}{p_g(x)}\right), \tag{17}$$

$$JSD(p_r||p_g) = \frac{1}{2}D_{KL}(p_r||M) + \frac{1}{2}D_{KL}(p_g||M), \tag{18}$$

where X is the probability space and $M = \frac{1}{2}(p_r + p_g)$.

However, we generally do not have information about the distributions, but rather only access to finite samples drawn from them. Thus, we must estimate the destiny function of these probabilities. Kernel Destiny Estimation (KDE) is perhaps the most commonly used method for this task. For a probability kernel K (such as a Gaussian) with bandwidth h and independent and identically distributed (i.i.d) samples $\{x_1, x_2, ..., x_n\}$, the kernel destiny estimator is:

$$\hat{p}(x) = \frac{1}{n}\sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right). \tag{19}$$

Although this measure is simple to compute; it suffers from a few drawbacks. Firstly, even for a large number of samples, KDE fails to approximate the model's true log-likelihood when the data dimensionality is high. Secondly, it may be shown that log-likelihood is uninformative regarding the quality of generated samples [138]. Consequently, a model that produces excellent samples may nonetheless have a poor log-likelihood. For these reasons, we will not use average log-likelihood as an evaluation metric for comparing different GAN models.

The Wasserstein critic is an approximation version of the Wasserstein distance (also called Earth Mover Distance (EMD)) between 2 data distributions, P_r and P_g and is given in equation [55, 134] below:

$$W(p_r, p_g) \propto \max_F E_{x\sim p_r}[F(\mathbf{x})] - E_{x\sim p_g}[F(x)], \tag{20}$$

where F is the Lipchitz function. Practically, the F is an MLP with clipped weights. In realistic scenarios, the expectation is not taken; rather, the average over the Lipchitz function is utilized:

$$\hat{W}(x_r, x_g) = \frac{1}{N}\sum_{i=1}^{N}\hat{F}(x_r[i]) - \frac{1}{N}\sum_{i=1}^{N}\hat{F}(x_g[i]). \tag{21}$$

MMD is a measurement for comparing the dissimilarity between samples drawn from a pair of probability distributions and can be defined as a particular function space that witnesses the difference between the two distributions. Hence, a lower MMD implies that the two distributions are closer to each other. Thus if P_r and P_g are two probability distributions, the kernel MMD (squared MMD) between these two distributions will be [139]:

$$\begin{aligned} MMD^2(p_r, p_g) =& E_{x,x'}[k(x, x')] - 2E_{x,y}[k(x, y)] \\ &+ E_{y,y'}[k(y, y')], \end{aligned} \tag{22}$$

where x and x' are two independent random variables with distribution p_r, y and y' are two independent random variables with distribution p_g and k is a fixed characteristic kernel function.

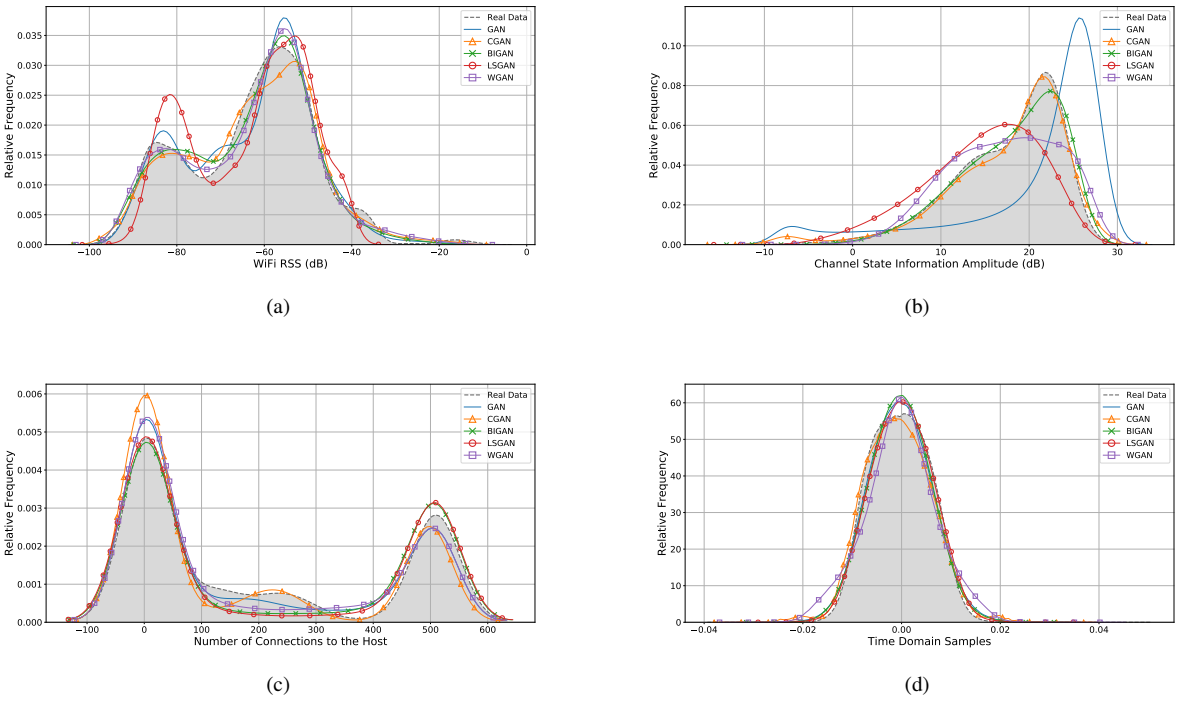

Figure 7: Estimated probability distributions for a) RSS dataset, b) CSI dataset, c) KDD99 dataset, column "count", and d) RADIOML datasets. The shaded line is the estimated distribution of the real data, while the dashed lines are those of the synthetic data.

4.2. Experimental Methodology and Results

In order to properly evaluate the performance of the various GANs within different network-related applications, we select reference datasets of varied types and characteristics. The datasets are: WiFi RSSI [140], CSI [90], network traffic (KDD99) [141] and digital modulations (Deepsig RadioML 2016.10A) [142]. Similarly, five different types of GANs are selected for evaluation, including Vanilla GAN, CGAN, BIGAN, LSGAN, and WGAN. The reason for this choice is that they are the most used models in the literature. Furthermore, while being easy to train with non-image data, each of these models adds a fundamentally different aspect to the original Vanilla GAN model. For instance, CGAN makes use of data labels, while BIGAN is able to map the real data distribution to the latent space. Experiments are carried out in Keras, accelerated by a Geforce RTX 2060 GPU. To keep consistency between implementations of different GANs and to mitigate possible errors that may occur in the implementation phase, we use Keras-GAN[3] as our base framework. The respective GAN model architecture parameters (shown in Table 4) are not adjusted during the experimental process, and no hyperparameter tuning is done (as altering learning parameters could bias the evaluation process).

We train each of the five GANs on each of the datasets separately and use noise drawn from the *same* latent space to generate samples. All of the datasets, with the exception of KDD99, contain data of the same type and magnitude. For KDD99, since GANs cannot generate discrete data, we select only 18 of the continuous non-zero features and use these to train the models. To make the experiment reproducible, we use the same random seed throughout the experiment for all datasets and models. However, since the GANs are trained over a sufficient amount of epochs, and the datasets are sufficiently large, randomness would have minimal effect on the results. Comparing performance between models is conducted via measurement of the two metrics introduced above; MMD and EMD. The evaluation process results for all of the datasets are as depicted in Table 5.

To further visualize these experimental outcomes, we use KDE to estimate the distribution of real and generated data. The estimated distributions of all four datasets are depicted in Fig.7. It should be noted that the selected features of the KDD99 datasets are of different orders, as some are of packet length, being in the order of 10000s, while others are of the order of 100s and even 10s. For this reason, in order to make the distribution plot more readable without loss of generality, we depict only the distribution for one of the columns, "count," which is the number of simultaneous connections to the host.

The final measure we consider is the quantile-quantile (Q-Q) plot, a graphical nonparametric method to compare the shape of two probability distributions. It consists in a scatter plot of the quantiles of one dataset against the quantiles of the other. If the points are close to the 45-degree ref-

[3] https://github.com/eriklindernoren/Keras-GAN

Table 5
Evaluation metrics for five different GANs on the four reference datasets. The used metrics are MMD and EMD, where a lower value means the generated data are closer to the original data.

Dataset	Data Shape	Model	MMD	EMD
WiFi RSSI [140]	(2000,7)	Vanilla GAN	0.077095	7.699772
		CGAN	0.064150	9.186863
		BIGAN	0.067965	9.912520
		LSGAN	0.132464	10.56940
		WGAN	0.042694	8.440702
WiFi CSI [90]	(2100,500,90)	Vanilla GAN	0.602905	1844.109
		CGAN	0.129852	810.8793
		BIGAN	0.108452	745.0461
		LSGAN	0.476095	1589.919
		WGAN	0.158754	778.3315
KDD99 [141]	(6000,18,1)	Vanilla GAN	0.036804	697.4859
		CGAN	0.178606	8423.204
		BIGAN	0.061430	994.2017
		LSGAN	0.147254	3376.442
		WGAN	0.029684	749.5329
RadioML [142]	(11000,2,128)	Vanilla GAN	0.084636	0.108324
		CGAN	0.152164	0.110944
		BIGAN	0.107003	0.107230
		LSGAN	0.097895	0.107172
		WGAN	0.085040	0.113345

erence line, we may conclude that the datasets are sampled from similarly-shaped distribution (although they may have different underlying parameters). Contrarily, if the points are far from the reference line, we may conclude that the distributions differ significantly. While the Q-Q plot can be considered a visual guide for comparing the two datasets' similarities, it should not be taken as reliable proof of similarity in itself. However, its ability to relatively characterize statistical properties such as central tendency, dispersion, and skewness makes it an extremely useful tool. The Q-Q plots of the datasets, as depicted in Fig.8, visually demonstrate that the estimated distributions shown in Fig.7, are indeed accurate.

4.3. Results

Inspection of the experimental results suggests that none of the tested GANs has a decisive advantage over others. Performance heavily depends on properties such as the dimensionality, magnitude, morphology, and type of data used (as might be expected *a priori* given the 'no free lunch' theorem). In other words, no variant of GAN can achieve superior performance for every type of data and in every application. Hence, the suitable way to find the proper model is through trial-and-error. For instance, Table 5 suggests that LSGAN and Vanilla GAN are not practical for high dimensional datasets, such as CSI; as they yielded a relatively large MMD and EMD. On the other hand, they indicate acceptable performance on other types of data.

Nevertheless, we assume that by changing the architecture of models (in particular, the number and depth of layers and the hyperparameter tuning), this situation could vary significantly. A further significant point to note from the results is that the tested GAN models cannot capture very fast fluctuations in the data distribution, presumably due to intrinsic model bias. It might be that the models are not sufficiently deep and that a deeper model with more layers and neurons would be able to capture these; however such architectural complexity has not as yet been applied practically in the network field.

5. Discussion and Conclusion

We provided a comprehensive survey of GANs regarding their main model variants in the general machine learning literature and exhaustively with respect to their applications in computer and communication networks. Despite being a relatively newly-proposed model, GANs have been widely accepted in the machine learning community, with the amount of research carried out in respect of them increasing significantly over time. Therefore, we briefly introduced the concept of generative machine learning prior to comparing the structure of the principle GAN model variants against each other. Next, we divided the extant work in the literature into five main categories and reviewed various applications of the different models within these categories. Afterward, we introduced a selection of the quantitative evaluation metrics in use in the field and evaluated the performance of a range of different GAN models on a selection of datasets taken from the reviewed papers. Finally, we set the major challenges and shortcomings of GANs with a preview of work to be done in the future. We strongly believe that GANs have many more potential applications within the field of computer and communication networks, and this discovery process has only just begun.

5.1. Take-home lessons

Alongside their extensive application in computer vision, GANs have a wide range of applications within networking. These vary from traffic classification, modulation recognition, self-organizing networks, spectrum sensing for intrusion detection systems, malware detection, and IoT. Generally, we benefit from the use of GANs where there are either data shortcomings, data imbalance, or else exceptional or adversarial circumstances for which an accurate discriminator is required, such as malware detection. When utilizing existing image GANs to generate non-image data, one of two broad approaches can be selected; we can convert the CNN models that are used in the majority of state-of-the-art GANs to generic multi-layer perceptions and train the network using normalized data, or else we can convert our data to images and use the existing GAN as is. An example of the former method is [115]. Whether one approach is superior to the other is yet to be concluded.

Moreover, as the training process of GANs does not require labels (except the case of conditional networks), GANs can be of enormous benefit in semi-supervised learning. For instance, one can train the GAN with the large portion of unlabeled data and then use the small portion of labeled data to train the discriminator for classification and regression; as done in [104]. There are many more applications in the

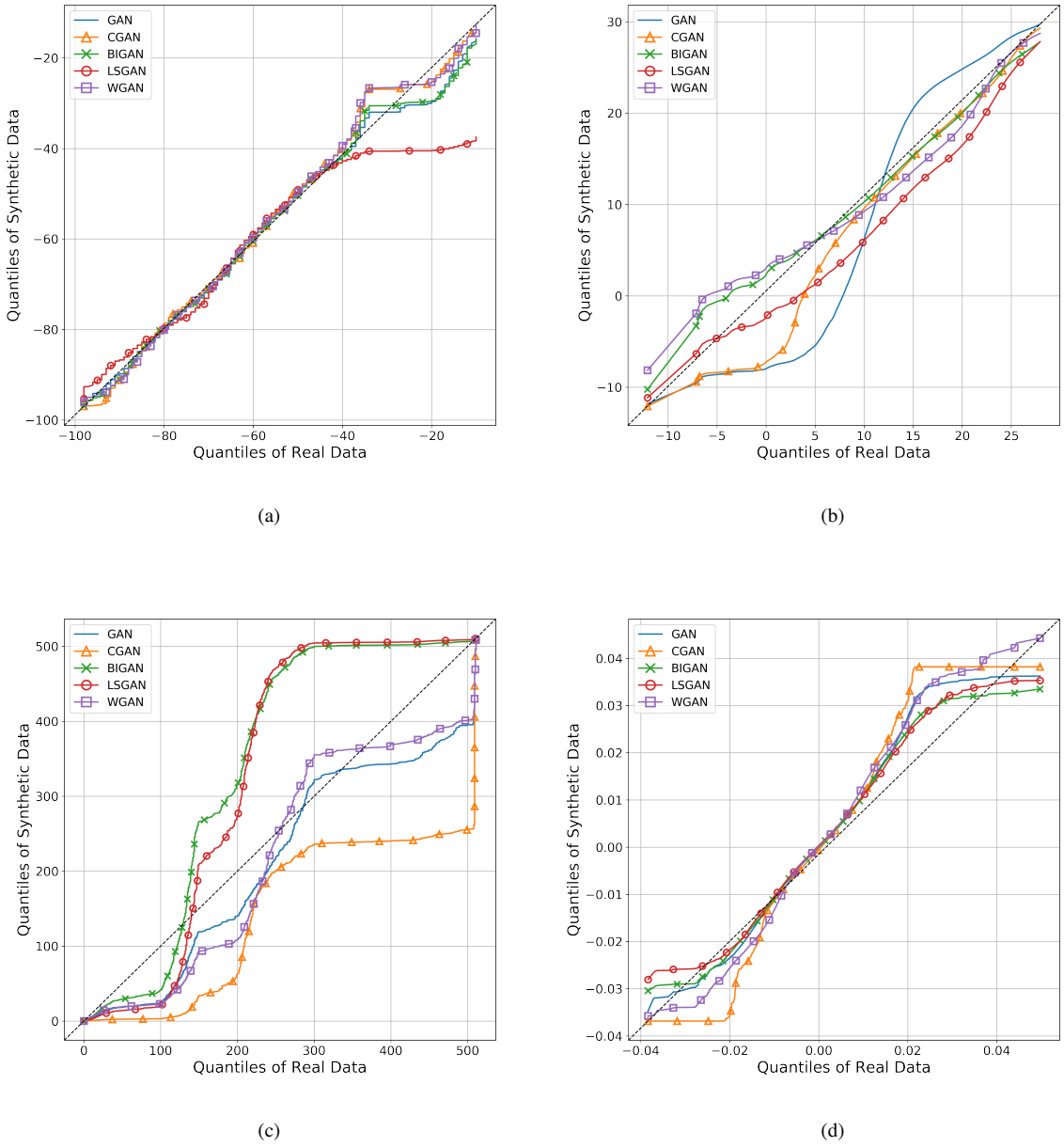

Figure 8: Q-Q plot for a) RSS dataset, b) CSI dataset, c) KDD99 dataset, column "count", and d) RADIOML datasets.

field of networking that could benefit from the discriminator network of GANs.

5.2. Future Work

Even though GANs have been used in many different scenarios, there are still many more applications that could benefit from this generative approach. In particular, there are many areas in the fields of physical layer, wireless sensor networks, and mobile networks that could significantly benefit. At present, though, most of the available studies on GANs are for the purpose of image generation and translation. Even when they can be used for non-image data (see above), most of the existing evaluation metrics are developed for image data. This can make selecting the correct GAN a time-consuming process, as the only absolutely reliable way to make such a decision is to generate data from the different models and compare the associated classification accuracy of the destination model. Thus, introducing new theoretical and statistical metrics for comparing non-image data would have a potentially huge impact.

A further point to note is that, since GANs were initially designed for image data, with appropriate hyperparameter optimization of or minor modifications to the underlying NNs, they can be made to generate practically any continu-

ous multi-dimensional numerical data. However, as of yet, they are unable to convincingly generate discrete data in arbitrary environments, mainly because the generator network is not able to use back-propagation in this case (recently the boundary-seeking GAN (BSGAN) [143] has been proposed for this task, but the performance of this model has yet to be evaluated with respect to discrete data related to networking).

Furthermore, the evolved variants of Vanilla GAN have improved the main structure in terms of convergence and provided a solution for the drawbacks of the initially proposed model, including mode collapse and instability. However, even with the introduction of new models, the improvement of GANs to the extent of generating data that is indistinguishable both to humans and machines is still an open challenge.

Finally, almost all of the reviewed work in this survey utilized GANs for supervised, unsupervised, and semi-supervised tasks; however, employing GANs in reinforcement learning is also a promising research interest. Recently, the number of research focusing on this direction is increasing. For instance, GANs can be combined with policy gradient [144], imitation learning [145], actor-critic method [146]. However, as of now, utilizing GANs with reinforcement learning algorithms in network-related tasks is something that has to be done.

References

[1] Ericsson Mobility Report June 2020. https://www.ericsson.com/49da93/assets/local/mobility-report/documents/2020/june2020-ericsson-mobility-report.pdf, 2020. [accessed November 2020].

[2] Chaoyun Zhang, Paul Patras, and Hamed Haddadi. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Communications Surveys Tutorials*, 21(3):2224–2287, thirdquarter 2019. ISSN 1553-877X.

[3] Jithin Jagannath, Nicholas Polosky, Anu Jagannath, Francesco Restuccia, and Tommaso Melodia. Machine learning for wireless communications in the Internet of Things: A comprehensive survey. *Ad Hoc Networks*, 93:101913, October 2019. ISSN 1570-8705.

[4] Bartosz Krawczyk. Learning from imbalanced data: Open challenges and future directions. *Progress in Artificial Intelligence*, 5 (4):221–232, November 2016. ISSN 2192-6360.

[5] Qingchen Zhang, Laurence T. Yang, Zhikui Chen, and Peng Li. A survey on deep learning for big data. *Information Fusion*, 42:146–157, July 2018. ISSN 1566-2535.

[6] Mohammad Nabati, Hojjat Navidan, Reza Shahbazian, Seyed Ali Ghorashi, and David Windridge. Using Synthetic Data to Enhance the Accuracy of Fingerprint-Based Localization: A Deep Learning Approach. *IEEE Sensors Letters*, 4(4):1–4, April 2020. ISSN 2475-1472.

[7] Jun-Yan Zhu, Richard Zhang, Deepak Pathak, Trevor Darrell, Alexei A. Efros, Oliver Wang, and Eli Shechtman. Toward Multimodal Image-to-Image Translation. In *Advances in Neural Information Processing Systems*, pages 465–476, December 2017.

[8] Debayan Deb, Jianbang Zhang, and Anil K. Jain. AdvFaces: Adversarial Face Synthesis. *arXiv:1908.05008 [cs]*, August 2019.

[9] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, July 2017. doi: 10.1109/CVPR.2017.632.

[10] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2242–2251, October 2017. doi: 10.1109/ICCV.2017.244.

[11] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. DualGAN: Unsupervised Dual Learning for Image-to-Image Translation. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2868–2876, October 2017. doi: 10.1109/ICCV.2017.310.

[12] Amin Fadaeddini, Babak Majidi, and Mohammad Eshghi. A Case Study of Generative Adversarial Networks for Procedural Synthesis of Original Textures in Video Games. In *2018 2nd National and 1st International Digital Games Research Conference: Trends, Technologies, and Applications (DGRC)*, pages 118–122, November 2018.

[13] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture Synthesis with Spatial Generative Adversarial Networks. *arXiv:1611.08207 [cs, stat]*, September 2017.

[14] Xin Yi, Ekta Walia, and Paul Babyn. Generative Adversarial Network in Medical Imaging: A Review. *Medical Image Analysis*, 58: 101552, December 2019. ISSN 13618415.

[15] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 105–114, July 2017. doi: 10.1109/CVPR.2017.19.

[16] Sen Li, Stephane Villette, Pravin Ramadas, and Daniel J. Sinder. Speech Bandwidth Extension Using Generative Adversarial Networks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5029–5033, Calgary, AB, April 2018. IEEE. ISBN 978-1-5386-4658-8.

[17] Hongyu Chen, Qinyin Xiao, and Xueyuan Yin. Generating Music Algorithm with Deep Convolutional Generative Adversarial Networks. In *2019 IEEE 2nd International Conference on Electronics Technology (ICET)*, pages 576–580, Chengdu, China, May 2019. IEEE. ISBN 978-1-72811-616-7 978-1-72811-618-1.

[18] Yuewei Dai, Weiwei Liu, Guangjie Liu, Xiaopeng Ji, and Jiangtao Zhai. An end-to-end generative network for environmental sound-based covert communication. *Multimedia Tools and Applications*, 78(7):8635–8653, April 2019. ISSN 1573-7721.

[19] Federico Di Mattia, Paolo Galeone, Michele De Simoni, and Emanuele Ghelfi. A Survey on GANs for Anomaly Detection. *arXiv:1906.11632 [cs, stat]*, June 2019.

[20] Huan Ying, Xuan Ouyang, Siwei Miao, and Yushi Cheng. Power Message Generation in Smart Grid via Generative Adversarial Network. In *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, pages 790–793, March 2019.

[21] Azam Bagheri, Irene Y.H. Gu, and Math H.J. Bollen. Generative Adversarial Model-Guided Deep Active Learning for Voltage Dip Labelling. In *2019 IEEE Milan PowerTech*, pages 1–5, June 2019.

[22] Chi Zhang, Sanmukh R. Kuppannagari, Rajgopal Kannan, and Viktor K. Prasanna. Generative Adversarial Network for Synthetic Time Series Data Generation in Smart Grids. In *2018 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–6, October 2018.

[23] Junliang Wang, Zhengliang Yang, Jie Zhang, Qihua Zhang, and Wei-Ting Kary Chien. AdaBalGAN: An Improved Generative Adversarial Network With Imbalanced Learning for Wafer Defective Pattern Recognition. *IEEE Transactions on Semiconductor Manufacturing*, 32(3):310–319, August 2019. ISSN 1558-2345.

[24] Biying Xu, Yibo Lin, Xiyuan Tang, Shaolan Li, Linxiao Shen, Nan Sun, and David Z. Pan. WellGAN: Generative-Adversarial-Network-Guided Well Generation for Analog/Mixed-Signal Circuit Layout. In *Proceedings of the 56th Annual Design Automation Conference 2019*, DAC '19, pages 1–6, Las Vegas, NV, USA, June 2019.

Association for Computing Machinery. ISBN 978-1-4503-6725-7.

[25] Funa Zhou, Shuai Yang, Hamido Fujita, Danmin Chen, and Chenglin Wen. Deep learning fault diagnosis method based on global optimization GAN for unbalanced data. *Knowledge-Based Systems*, 187:104837, January 2020. ISSN 09507051.

[26] Jinrui Wang, Shunming Li, Baokun Han, Zenghui An, Huaiqian Bao, and Shanshan Ji. Generalization of Deep Neural Networks for Imbalanced Fault Classification of Machinery Using Generative Adversarial Networks. *IEEE Access*, 7:111168–111180, 2019. ISSN 2169-3536.

[27] Yuan Xie and Tao Zhang. A Transfer Learning Strategy for Rotation Machinery Fault Diagnosis based on Cycle-Consistent Generative Adversarial Networks. In *2018 Chinese Automation Congress (CAC)*, pages 1309–1313, November 2018.

[28] Zihe Zhao, Rui Zhou, and Zhuoning Dong. Aero-Engine Faults Diagnosis Based on K-Means Improved Wasserstein GAN and Relevant Vector Machine. In *2019 Chinese Control Conference (CCC)*, pages 4795–4800, July 2019.

[29] Yang-Jie Cao, Li-Li Jia, Yong-Xia Chen, Nan Lin, Cong Yang, Bo Zhang, Zhi Liu, Xue-Xiang Li, and Hong-Hua Dai. Recent Advances of Generative Adversarial Networks in Computer Vision. *IEEE Access*, 7:14985–15006, 2019. ISSN 2169-3536.

[30] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, November 1998. ISSN 1558-2256.

[31] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv:1708.07747 [cs, stat]*, September 2017.

[32] Kunfeng Wang, Chao Gou, Yanjie Duan, Yilun Lin, Xinhu Zheng, and Fei-Yue Wang. Generative adversarial networks: Introduction and outlook. *IEEE/CAA Journal of Automatica Sinica*, 4(4):588–598, 2017. ISSN 2329-9266, 2329-9274.

[33] Zhaoqing Pan, Weijie Yu, Xiaokai Yi, Asifullah Khan, Feng Yuan, and Yuhui Zheng. Recent Progress on Generative Adversarial Networks (GANs): A Survey. *IEEE Access*, 7:36322–36333, 2019. ISSN 2169-3536.

[34] Ceren Guzel Turhan and Hasan Sakir Bilge. Recent Trends in Deep Generative Models: A Review. In *2018 3rd International Conference on Computer Science and Engineering (UBMK)*, pages 574–579, Sarajevo, September 2018. IEEE. ISBN 978-1-5386-7893-0.

[35] Ian Goodfellow. NIPS 2016 Tutorial: Generative Adversarial Networks. *arXiv:1701.00160 [cs]*, April 2017.

[36] Liang Gonog and Yimin Zhou. A Review: Generative Adversarial Networks. In *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, pages 505–510, Xi'an, China, June 2019. IEEE. ISBN 978-1-5386-9490-9.

[37] Su-Fang Zhang, Jun-Hai Zhai, Ding-Sheng Luo, Yan Zhan, and Jun-Fen Chen. Recent Advance On Generative Adversarial Networks. In *2018 International Conference on Machine Learning and Cybernetics (ICMLC)*, pages 69–74, Chengdu, July 2018. IEEE. ISBN 978-1-5386-5214-5.

[38] Xian Wu, Kun Xu, and Peter Hall. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology*, 22(6):660–674, December 2017. ISSN 1007-0214.

[39] Connor Shorten and Taghi M. Khoshgoftaar. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, 6(1): 60, December 2019. ISSN 2196-1115.

[40] Shirin Nasr Esfahani and Shahram Latifi. Image Generation with Gans-based Techniques: A Survey. *International Journal of Computer Science and Information Technology*, 11(5):33–50, October 2019. ISSN 09754660.

[41] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative Adversarial Networks: An Overview. *IEEE Signal Processing Magazine*, 35(1):53–65, January 2018. ISSN 1053-5888.

[42] Shani Gamrian and Yoav Goldberg. Transfer Learning for Related Reinforcement Learning Tasks via Image-to-Image Translation. *arXiv:1806.07377 [cs]*, July 2019.

[43] Chelsea Finn, Paul Christiano, Pieter Abbeel, and Sergey Levine. A Connection between Generative Adversarial Networks, Inverse Reinforcement Learning, and Energy-Based Models. *arXiv:1611.03852 [cs]*, November 2016.

[44] Teodora Pandeva and Matthias Schubert. MMGAN: Generative Adversarial Networks for Multi-Modal Distributions. *arXiv:1911.06663 [cs, stat]*, November 2019.

[45] Abhishek Kumar, Prasanna Sattigeri, and Tom Fletcher. Semi-supervised Learning with GANs: Manifold Invariance with Improved Inference. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5534–5544. Curran Associates, Inc., 2017.

[46] Mohammad Ali Keyvanrad and Mohammad Mehdi Homayounpour. A brief survey on deep belief networks and introducing a new object oriented toolbox (DeeBNet). *arXiv:1408.3264 [cs]*, January 2016.

[47] Danilo Jimenez Rezende and Shakir Mohamed. Variational Inference with Normalizing Flows. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37, pages 1530–1538, July 2015.

[48] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, pages 2672–2680, Montreal, Canada, December 2014. MIT Press.

[49] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training GANs. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, pages 2234–2242, Barcelona, Spain, December 2016. Curran Associates Inc. ISBN 978-1-5108-3881-9.

[50] Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. *arXiv:1411.1784 [cs, stat]*, November 2014.

[51] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial Feature Learning. *arXiv:1605.09782 [cs, stat]*, April 2017.

[52] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to Discover Cross-Domain Relations with Generative Adversarial Networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1857–1865, August 2017.

[53] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, page 2180–2188. Curran Associates Inc., December 2016. ISBN 9781510838819.

[54] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based Generative Adversarial Network. *arXiv:1609.03126 [cs, stat]*, March 2017.

[55] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 214–223, August 2017.

[56] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least Squares Generative Adversarial Networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, Venice, October 2017. IEEE. ISBN 978-1-5386-1032-9.

[57] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *arXiv:1511.06434 [cs]*, January 2016.

[58] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional Image Synthesis With Auxiliary Classifier GANs. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, page 2642–2651, August 2017.

[59] Martin Arjovsky and Léon Bottou. Towards Principled Methods for Training Generative Adversarial Networks. *arXiv:1701.04862 [cs,*

stat], January 2017.

[60] Houshang H. Sohrab. *Basic Real Analysis*. Birkhäuser Basel, 2003. ISBN 978-1-4612-6503-0.

[61] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 5769–5779, December 2017.

[62] Jose Ordonez-Lucena, Pablo Ameigeiras, Diego Lopez, Juan J. Ramos-Munoz, Javier Lorca, and Jesus Folgueira. Network Slicing for 5G with SDN/NFV: Concepts, Architectures, and Challenges. *IEEE Communications Magazine*, 55(5):80–87, May 2017. ISSN 1558-1896.

[63] Ruichun Gu and Junxing Zhang. GANSlicing: A GAN-Based Software Defined Mobile Network Slicing Scheme for IoT Applications. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–7, Shanghai, China, May 2019. IEEE. ISBN 978-1-5386-8088-9.

[64] Osianoh Glenn Aliu, Ali Imran, Muhammad Ali Imran, and Barry Evans. A Survey of Self Organisation in Future Cellular Networks. *IEEE Communications Surveys Tutorials*, 15(1):336–361, First 2013. ISSN 1553-877X.

[65] Ali Imran, Ahmed Zoha, and Adnan Abu-Dayya. Challenges in 5G: How to empower SON with big data for enabling 5G. *IEEE Network*, 28(6):27–33, November 2014. ISSN 1558-156X.

[66] Paulo Valente Klaine, Muhammad Ali Imran, Oluwakayode Onireti, and Richard Demo Souza. A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks. *IEEE Communications Surveys Tutorials*, 19(4):2392–2431, Fourthquarter 2017. ISSN 1553-877X.

[67] Jessica Moysen and Lorenza Giupponi. From 4G to 5G: Self-organized network management meets machine learning. *Computer Communications*, 129:248–268, September 2018. ISSN 0140-3664.

[68] David Palacios and Raquel Barco. Unsupervised Technique for Automatic Selection of Performance Indicators in Self-Organizing Networks. *IEEE Communications Letters*, 21(10):2198–2201, October 2017. ISSN 1558-2558.

[69] Wuyang Zhang, Russell Ford, Joonyoung Cho, Charlie Jianzhong Zhang, Yanyong Zhang, and Dipankar Raychaudhuri. Self-Organizing Cellular Radio Access Network with Deep Learning. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 429–434, April 2019.

[70] Stephen S. Mwanje, Lars Christoph Schmelz, and Andreas Mitschele-Thiel. Cognitive Cellular Networks: A Q-Learning Framework for Self-Organizing Networks. *IEEE Transactions on Network and Service Management*, 13(1):85–98, March 2016. ISSN 1932-4537.

[71] Faris B. Mismar and Brian L. Evans. Deep Q-Learning for Self-Organizing Networks Fault Management and Radio Performance Improvement. *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, pages 1457–1461, October 2018.

[72] Tao Zhang, Kun Zhu, and Dusit Niyato. A Generative Adversarial Learning-Based Approach for Cell Outage Detection in Self-Organizing Cellular Networks. *IEEE Wireless Communications Letters*, 9(2):171–174, February 2020. ISSN 2162-2337, 2162-2345.

[73] Ben Hughes, Shruti Bothe, Hasan Farooq, and Ali Imran. Generative Adversarial Learning for Machine Learning empowered Self Organizing 5G Networks. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 282–286, Honolulu, HI, USA, February 2019. IEEE. ISBN 978-1-5386-9223-3.

[74] Jon J. Aho, Alexander W. Witt, Carter B. F. Casey, Nirav Trivedi, and Venkatesh Ramaswamy. Generating Realistic Data for Network Analytics. In *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM)*, pages 401–406, Los Angeles, CA, October 2018. IEEE. ISBN 978-1-5386-7185-6.

[75] San Jose. WAN and Application Optimization Solution Guide. https://www.cisco.com/c/dam/en/us/td/docs/nsite/wan_optimization/WANoptSolutionGd.pdf, 2008. [accessed November 2020].

[76] Muhammad Shafiq, Xiangzhan Yu, Asif Ali Laghari, Lu Yao, Nabin Kumar Karn, and Foudil Abdessamia. Network Traffic Classification techniques and comparative analysis using Machine Learning algorithms. In *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*, pages 2451–2455, October 2016.

[77] ZiXuan Wang, Pan Wang, Xiaokang Zhou, ShuHang Li, and MoXuan Zhang. FLOWGAN:Unbalanced Network Encrypted Traffic Identification Method Based on GAN. In *2019 IEEE Intl Conf on Parallel Distributed Processing with Applications, Big Data Cloud Computing, Sustainable Computing Communications, Social Computing Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pages 975–983, December 2019.

[78] Jie Li, Lu Zhou, Huaxin Li, Lu Yan, and Haojin Zhu. Dynamic Traffic Feature Camouflaging via Generative Adversarial Networks. In *2019 IEEE Conference on Communications and Network Security (CNS)*, pages 268–276, Washington DC, DC, USA, June 2019. IEEE. ISBN 978-1-5386-7117-7.

[79] Kai Lei, Meng Qin, Bo Bai, Gong Zhang, and Min Yang. GCN-GAN: A Non-linear Temporal Link Prediction Model for Weighted Dynamic Networks. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications*, pages 388–396, Paris, France, April 2019. IEEE. ISBN 978-1-72810-515-4.

[80] Yanjiao Chen, Yuxuan Xiong, Bulou Liu, and Xiaoyan Yin. TranGAN: Generative Adversarial Network Based Transfer Learning for Social Tie Prediction. In *ICC 2019 - 2019 IEEE International Conference on Communications (ICC)*, pages 1–6, Shanghai, China, May 2019. IEEE. ISBN 978-1-5386-8088-9.

[81] Chongxuan Li, Kun Xu, Jun Zhu, and Bo Zhang. Triple Generative Adversarial Nets. *arXiv:1703.02291 [cs]*, November 2017.

[82] Huihui Xie, Kun Lv, and Changzhen Hu. An Effective Method to Generate Simulated Attack Data Based on Generative Adversarial Nets. In *2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, pages 1777–1784, New York, NY, USA, August 2018. IEEE. ISBN 978-1-5386-4388-4.

[83] S. Hadim and N. Mohamed. Middleware for Wireless Sensor Networks: A Survey. In *2006 1st International Conference on Communication Systems Software Middleware*, pages 1–7, January 2006.

[84] Remah Alshinina and Khaled Elleithy. A highly accurate machine learning approach for developing wireless sensor network middleware. In *2018 Wireless Telecommunications Symposium (WTS)*, pages 1–7, April 2018.

[85] Xuyu Wang, Lingjun Gao, Shiwen Mao, and Santosh Pandey. CSI-Based Fingerprinting for Indoor Localization: A Deep Learning Approach. *IEEE Transactions on Vehicular Technology*, 66(1):763–776, January 2017. ISSN 1939-9359.

[86] Elaheh Homayounvala, Mohammad Nabati, Reza Shahbazian, Seyed Ali Ghorashi, and Vahideh Moghtadaiee. A novel smartphone application for indoor positioning of users based on machine learning. In *Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers*, UbiComp/ISWC '19 Adjunct, pages 430–437, London, United Kingdom, September 2019. Association for Computing Machinery. ISBN 978-1-4503-6869-8.

[87] Qiyue Li, Heng Qu, Zhi Liu, Wei Sun, Xun Shao, and Jie Li. Wavelet Transform DC-GAN for Diversity Promoted Fingerprint Construction in Indoor Localization. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–7, Abu Dhabi, United Arab Emirates, December 2018. IEEE. ISBN 978-1-5386-4920-6.

[88] Mehdi Mohammadi, Ala Al-Fuqaha, and Jun-Seok Oh. Path Planning in Support of Smart Mobility Applications using Generative Adversarial Networks. In *2018 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and*

Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pages 878–885, August 2018. doi: 10.1109/Cybermatics_2018.2018.00168.

[89] Jian Liu, Hongbo Liu, Yingying Chen, Yan Wang, and Chen Wang. Wireless Sensing for Human Activity: A Survey. *IEEE Communications Surveys Tutorials*, pages 1–1, 2019. ISSN 1553-877X.

[90] Siamak Yousefi, Hirokazu Narui, Sankalp Dayal, Stefano Ermon, and Shahrokh Valaee. A Survey on Behavior Recognition Using WiFi Channel State Information. *IEEE Communications Magazine*, 55(10):98–104, October 2017. ISSN 1558-1896.

[91] Parisa Fard Moshiri, Hojjat Navidan, Reza Shahbazian, Seyed Ali Ghorashi, and David Windridge. Using GAN to Enhance the Accuracy of Indoor Human Activity Recognition. *arXiv:2004.11228 [cs, eess]*, April 2020.

[92] Chunjing Xiao, Daojun Han, Yongsen Ma, and Zhiguang Qin. CsiGAN: Robust Channel State Information-Based Activity Recognition With GANs. *IEEE Internet of Things Journal*, 6(6):10191–10204, December 2019. ISSN 2327-4662.

[93] Timothy O'Shea and Jakob Hoydis. An Introduction to Deep Learning for the Physical Layer. *IEEE Transactions on Cognitive Communications and Networking*, 3(4):563–575, December 2017. ISSN 2332-7731.

[94] Timothy J. Oshea, Tamoghna Roy, Nathan West, and Benjamin C. Hilburn. Physical Layer Communications System Design Over-the-Air Using Adversarial Networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 529–532, Rome, September 2018. IEEE. ISBN 978-90-827970-1-5.

[95] Timothy J. O'Shea, Tamoghna Roy, and Nathan West. Approximating the Void: Learning Stochastic Channel Models from Observation with Variational Generative Adversarial Networks. In *2019 International Conference on Computing, Networking and Communications (ICNC)*, pages 681–686, Honolulu, HI, USA, February 2019. IEEE. ISBN 978-1-5386-9223-3.

[96] Hao Ye, Geoffrey Ye Li, Biing-Hwang Fred Juang, and Kathiravetpillai Sivanesan. Channel Agnostic End-to-End Learning Based Communication Systems with Conditional GAN. In *2018 IEEE Globecom Workshops (GC Wkshps)*, pages 1–5, Abu Dhabi, United Arab Emirates, December 2018. IEEE. ISBN 978-1-5386-4920-6.

[97] Aaron Smith and Joseph Downey. A Communication Channel Density Estimating Generative Adversarial Network. In *2019 IEEE Cognitive Communications for Aerospace Applications Workshop (CCAAW)*, pages 1–7, Cleveland, OH, USA, June 2019. IEEE. ISBN 978-1-72810-048-7.

[98] Yang Yang, Yang Li, Wuxiong Zhang, Fei Qin, Pengcheng Zhu, and Cheng-Xiang Wang. Generative-Adversarial-Network-Based Wireless Channel Modeling: Challenges and Opportunities. *IEEE Communications Magazine*, 57(3):22–27, March 2019. ISSN 1558-1896.

[99] Caidan Zhao, Caiyun Chen, Zhibiao Cai, Mingxian Shi, Xiaojiang Du, and Mohsen Guizani. Classification of Small UAVs Based on Auxiliary Classifier Wasserstein GANs. In *2018 IEEE Global Communications Conference (GLOBECOM)*, pages 206–212, Abu Dhabi, United Arab Emirates, December 2018. IEEE. ISBN 978-1-5386-4727-1.

[100] Charles Clancy, Joe Hecker, Erich Stuntebeck, and Tim O'Shea. Applications of Machine Learning to Cognitive Radio Networks. *IEEE Wireless Communications*, 14(4):47–52, August 2007. ISSN 1558-0687.

[101] Mario Bkassiny, Yang Li, and Sudharman K. Jayaweera. A Survey on Machine-Learning Techniques in Cognitive Radios. *IEEE Communications Surveys Tutorials*, 15(3):1136–1159, Third 2013. ISSN 1553-877X.

[102] Narendar Madhavan, A. P. Vinod, A. S. Madhukumar, and Anoop Kumar Krishna. Spectrum sensing and modulation classification for cognitive radios using cumulants based on fractional lower order statistics. *AEU - International Journal of Electronics and Communications*, 67(6):479–490, June 2013. ISSN 1434-8411.

[103] Kemal Davaslioglu and Yalin E. Sagduyu. Generative Adversarial Learning for Spectrum Sensing. In *2018 IEEE International Conference on Communications (ICC)*, pages 1–6, May 2018.

[104] Mingxuan Li, Guangyi Liu, Shuntao Li, and Yifan Wu. Radio Classify Generative Adversarial Networks: A Semi-supervised Method for Modulation Recognition. In *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pages 669–672, Chongqing, October 2018. IEEE. ISBN 978-1-5386-7635-6.

[105] Yi Shi, Kemal Davaslioglu, and Yalin E. Sagduyu. Generative Adversarial Network for Wireless Signal Spoofing. In *Proceedings of the ACM Workshop on Wireless Security and Machine Learning*, WiseML 2019, pages 55–60, Miami, FL, USA, May 2019. Association for Computing Machinery. ISBN 978-1-4503-6769-1.

[106] Tugba Erpek, Yalin E. Sagduyu, and Yi Shi. Deep Learning for Launching and Mitigating Wireless Jamming Attacks. *IEEE Transactions on Cognitive Communications and Networking*, 5(1):2–14, March 2019. ISSN 2332-7731, 2372-2045.

[107] Ramin Soltani, Dennis Goeckel, Don Towsley, Boulat A. Bash, and Saikat Guha. Covert Wireless Communication With Artificial Noise Generation. *IEEE Transactions on Wireless Communications*, 17 (11):7252–7267, November 2018. ISSN 1558-2248.

[108] Xiaomin Liao, Xiaomin Liao, Jiangbo Si, Jia Shi, Zan Li, and Haiyang Ding. Generative Adversarial Network Assisted Power Allocation for Cooperative Cognitive Covert Communication System. *IEEE Communications Letters*, pages 1–1, 2020. ISSN 1558-2558.

[109] Anna L. Buczak and Erhan Guven. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176, Secondquarter 2016. ISSN 1553-877X.

[110] Chathurika S. Wickramasinghe, Daniel L. Marino, Kasun Amarasinghe, and Milos Manic. Generalization of Deep Learning for Cyber-Physical System Security: A Survey. In *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, pages 745–751, October 2018.

[111] Mohammed Ali Al-Garadi, Amr Mohamed, Abdulla Al-Ali, Xiaojiang Du, and Mohsen Guizani. A Survey of Machine and Deep Learning Methods for Internet of Things (IoT) Security. *IEEE Communications Surveys Tutorials*, 22(3):1646–1685, 2020.

[112] Daniel S. Berman, Anna L. Buczak, Jeffrey S. Chavis, and Cherita L. Corbett. A Survey of Deep Learning Methods for Cyber Security. *Information*, 10(4):122, April 2019.

[113] Tuan A Tang, Lotfi Mhamdi, Des McLernon, Syed Ali Raza Zaidi, and Mounir Ghogho. Deep learning approach for Network Intrusion Detection in Software Defined Networking. In *2016 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, pages 258–263, October 2016.

[114] Muhammad Usama, Muhammad Asim, Siddique Latif, Junaid Qadir, and Ala-Al-Fuqaha. Generative Adversarial Networks For Launching and Thwarting Adversarial Attacks on Network Intrusion Detection Systems. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 78–83, Tangier, Morocco, June 2019. IEEE. ISBN 978-1-5386-7747-6.

[115] Milad Salem, Shayan Taheri, and Jiann Shiun Yuan. Anomaly Generation Using Generative Adversarial Networks in Host-Based Intrusion Detection. In *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, pages 683–687, New York City, NY, USA, November 2018. IEEE. ISBN 978-1-5386-7693-6.

[116] Zilong Lin, Yong Shi, and Zhi Xue. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. *arXiv:1809.02077 [cs]*, June 2019.

[117] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, pages 1–6, Belfast, August 2018. IEEE. ISBN 978-1-5386-7493-2.

[118] Muhammad Amin, Babar Shah, Aizaz Sharif, Tamleek Ali, Ki-lL Kim, and Sajid Anwar. Android malware detection through generative adversarial networks. *Transactions on Emerging Telecommunications Technologies*, July 2019. ISSN 2161-3915, 2161-3915.

[119] Jin-Young Kim, Seok-Jun Bu, and Sung-Bae Cho. Malware Detection Using Deep Transferred Generative Adversarial Networks. In Derong Liu, Shengli Xie, Yuanqing Li, Dongbin Zhao, and El-Sayed M. El-Alfy, editors, *Neural Information Processing*, Lecture Notes in Computer Science, pages 556–564, Cham, 2017. Springer International Publishing. ISBN 978-3-319-70087-8.

[120] Debashri Roy, Tathagata Mukherjee, Mainak Chatterjee, and Eduardo Pasiliao. Detection of Rogue RF Transmitters using Generative Adversarial Nets. In *2019 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–7, Marrakesh, Morocco, April 2019. IEEE. ISBN 978-1-5386-7646-2.

[121] Maria Rigaki and Sebastian Garcia. Bringing a GAN to a Knife-Fight: Adapting Malware Communication to Avoid Detection. In *2018 IEEE Security and Privacy Workshops (SPW)*, pages 70–75, San Francisco, CA, May 2018. IEEE. ISBN 978-1-5386-8276-0.

[122] Weiwei Hu and Ying Tan. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. *arXiv:1702.05983 [cs]*, February 2017.

[123] Masataka Kawai, Kaoru Ota, and Mianxing Dong. Improved MalGAN: Avoiding Malware Detector by Leaning Cleanware Features. In *2019 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 040–045, Okinawa, Japan, February 2019. IEEE. ISBN 978-1-5386-7822-0.

[124] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical Black-Box Attacks against Machine Learning. In *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*, ASIA CCS '17, pages 506–519, Abu Dhabi, United Arab Emirates, April 2017. Association for Computing Machinery. ISBN 978-1-4503-4944-4.

[125] Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando Perez-Cruz. PassGAN: A Deep Learning Approach for Password Guessing. In *Applied Cryptography and Network Security*, pages 217–237, June 2019.

[126] Jian Chen, Yao Shen, and Riaz Ali. Credit Card Fraud Detection Using Sparse Autoencoder and Generative Adversarial Network. In *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pages 1054–1059, Vancouver, BC, November 2018. IEEE. ISBN 978-1-5386-7266-2.

[127] Xiaoguo Wang, Ran Zhao, and Yuanxiu Li. A Fraudulent Data Simulation Method Based on Generative Adversarial Networks. *Journal of Physics: Conference Series*, 1302:022089, August 2019. ISSN 1742-6588, 1742-6596.

[128] Akhil Sethia, Raj Patel, and Purva Raut. Data Augmentation using Generative models for Credit Card Fraud Detection. In *2018 4th International Conference on Computing Communication and Automation (ICCCA)*, pages 1–6, Greater Noida, India, December 2018. IEEE. ISBN 978-1-5386-6947-1.

[129] Gideon Creech and Jiankun Hu. A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns. *IEEE Transactions on Computers*, 63 (4):807–819, April 2014. ISSN 1557-9956.

[130] Chia-Ling Liu. Impacts of I/Q imbalance on QPSK-OFDM-QAM detection. *IEEE Transactions on Consumer Electronics*, 44(3):984–989, August 1998. ISSN 1558-4127.

[131] Sang-Yun Shin, Yong-Won Kang, and Yong-Guk Kim. Android-GAN: Defending against android pattern attacks using multi-modal generative network as anomaly detector. *Expert Systems with Applications*, 141:112964, March 2020. ISSN 09574174.

[132] Yi Shi, Yalin E. Sagduyu, Kemal Davaslioglu, and Jason H. Li. Generative Adversarial Networks for Black-Box API Attacks with Limited Training Data. In *2018 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 453–458, Louisville, KY, USA, December 2018. IEEE. ISBN 978-1-5386-7568-7.

[133] Ruohan Wang, Antoine Cully, Hyung Jin Chang, and Yiannis Demiris. MAGAN: Margin Adaptation for Generative Adversarial Networks. *arXiv:1704.03817 [cs, stat]*, May 2017.

[134] Ali Borji. Pros and cons of gan evaluation measures. *Computer Vision and Image Understanding*, 179:41 – 65, 2019. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2018.10.009.

[135] Tong Che, Yanran Li, Athul Paul Jacob, Yoshua Bengio, and Wenjie Li. Mode Regularized Generative Adversarial Networks. *arXiv:1612.02136 [cs]*, March 2017.

[136] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local nash equilibrium. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6629–6640, Long Beach, California, USA, December 2017. Curran Associates Inc. ISBN 978-1-5108-6096-4.

[137] Qiantong Xu, Gao Huang, Yang Yuan, Chuan Guo, Yu Sun, Felix Wu, and Kilian Weinberger. An empirical study on evaluation metrics of generative adversarial networks. *arXiv:1806.07755 [cs, stat]*, August 2018.

[138] Lucas Theis, Aäron van den Oord, and Matthias Bethge. A note on the evaluation of generative models. *arXiv:1511.01844 [cs, stat]*, April 2016.

[139] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(null):723–773, March 2012. ISSN 1532-4435.

[140] Jayant G. Rohra, Boominathan Perumal, Swathi Jamjala Narayanan, Priya Thakur, and Rajen B. Bhatt. User Localization in an Indoor Environment Using Fuzzy Hybrid of Particle Swarm Optimization & Gravitational Search Algorithm with Neural Networks. In Kusum Deep, Jagdish Chand Bansal, Kedar Nath Das, Arvind Kumar Lal, Harish Garg, Atulya K. Nagar, and Millie Pant, editors, *Proceedings of Sixth International Conference on Soft Computing for Problem Solving*, Advances in Intelligent Systems and Computing, pages 286–295, Singapore, 2017. Springer. ISBN 978-981-10-3322-3.

[141] KDD Cup 1999. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html, 1999. accessed November 2020.

[142] DeepSig Inc. RF Datasets For Machine Learning | DeepSig. https://www.deepsig.ai/datasets, 2016. accessed November 2020.

[143] R. Devon Hjelm, Athul Paul Jacob, Tong Che, Adam Trischler, Kyunghyun Cho, and Yoshua Bengio. Boundary-Seeking Generative Adversarial Networks. *arXiv:1702.08431 [cs, stat]*, February 2018.

[144] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, page 2852–2858, February 2017.

[145] Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in Neural Information Processing Systems*, volume 29 of *NIPS'16*, pages 4565–4573, December 2016.

[146] David Pfau and Oriol Vinyals. Connecting Generative Adversarial Networks and Actor-Critic Methods. *arXiv:1610.01945 [cs, stat]*, January 2017.