A Survey of Generative Adversarial Network on Next Generation Network

Ramin Mousa * and Babak Masoudi

Posted Date: 22 October 2024

doi: 10.20944/preprints202410.1662.v1

Keywords:

generative adversarial networks; next generation network; 4G; 5G; 6G

Article

A Survey of Generative Adversarial Network on Next Generation Network

Ramin Mousa [1,*] and Babak Masoudi [2]

[1] Department of Computer Engineering, University of Zanjan
[2] Department of Information Technology, University of Payamenoor University (PNU)
* Correspondence: Raminmousa@znu.ac.ir

Abstract: Generative adversarial networks consisting of two parts, generator and discriminator, have obtained acceptable results in classification, prediction and generation of new samples in the discussion of 4G, 5G and 6G. In this article, the aim is to review a comprehensive study in relation to these networks. At the beginning of this research, different topics such as network training, network error checking and types of GAN's are investigated. In the following, the applications of GAN will be discussed separately.

Keywords: generative adversarial networks; next generation network; 4G; 5G; 6G

1. Introduction

As one of the most innovative deep learning models in recent years, generative adversarial networks have achieved great success in computer vision and natural language processing. Along with the idea of "associative" and "adversarial", researchers are trying to apply generative adversarial networks in 4G, 5G, AND 6G. Continuing this research, we present the development of productive adversarial networks. We review traditional generation models and typical models of generative adversarial networks. We separately investigate their defensive contribution to emphasize that generative adversarial network models are feasible for use in 4G, 5G, AND 6G. Finally, relying on the reviewed literature, we provide a broader perspective of this research direction.

In 2014, authors [1] proposed a generative model called "generative adversarial networks (GAN)". It consists of a generator and a discriminator. The generator is responsible for producing the samples, and the discriminator is responsible for recognizing the authenticity of the samples. Since the goal of each side is to defeat the other, the self-optimizing model is continuously refined, and the generator, after the final training, can produce an almost realistic example through each input.

GAN originates from a zero-sum game in game theory, where two players are producers and discriminators. The random noise z is fed into the generator and produces a sample G(z), and then the test sample is fed into the discriminator. The more realistic the generated sample, the closer the result is to 1; otherwise, the result is closer to 0 [1] . Generator error is the misclassification of unrealistic samples generated by the discriminator, and the generator must be continuously trained to generate newly generated samples to fool the discriminator. Discrimination loss is the misjudgment of real samples and faked samples. Since it is difficult for the first generated sample to reach the level of real data, the GAN must be continuously trained and optimized. However, GAN training differs from the previous neural network training, and separate alternating iterative training is adopted [3–5]. In GAN, a fixed generator is used to optimize the discriminator or a fixed discriminator is used to optimize the generator [1]. The general formula of GAN is as follows:

$$min_g max_d V(D,G) = E_{X \sim P_data(x)}[log(D(x))] + E_{z \sim P_z(z)}[log(1 - D(G(Z)))] \quad (1)$$

Here E is the expected value of the distribution function, $X \sim P_data(x)$ is the actual sampling distribution, and $z \sim P_z(z)$ is the distribution that generated the sample.

 doi:10.20944/preprints202410.1662.v1

1.1. Training Generator

The generator generates samples for the recognizer, and the final result is returned to the generator. Since there is a certain gap between the generated samples and the real samples at the beginning of the training, the discriminator gives the loss feedback to the generator. The result should be that the generator can produce samples that fool the discriminator, and the discriminator's judgment should be close to 1 on the sample.

$$min_g V(D,G) = E_{z\sim P_z(z)}[log(1 - D(G(Z)))] \tag{2}$$

1.2. Training Discriminator

In the process of frequency recognition training, the generator is fixed. By continuously judging the real and generated samples, the frequency discriminator improves the frequency detection ability and achieves a higher frequency detection ability. Therefore, the frequency discrimination of unrealistic samples is close to 0.

$$max_d V(D,G) = E_{X\sim P_data(x)}[log(D(x))] + E_{z\sim P_z(z)}[log(1 - D(G(Z)))] \tag{3}$$

2. Typical GAN Models

Various types of GAN models have flourished in recent years, here we list some common models.

2.1. Conditional Generative Adversarial Nets (CGAN)

The advantage of GAN is that the input does not rely on the expected data distribution. The generated samples can be more realistic using direct sampling for the Z random noise input. The disadvantage is that the generated samples are too free to control and cannot focus on the specified samples, and constraints must be added to the GAN. It is from this idea that CGAN came to the GAN scene. Considering the dual randomness of GAN (random noise and random samples), authors proposed CGAN [17] and the "y" tag was added as an input in the GAN generator and discriminator. By adding y, the input to the generator becomes noise and a label, while the input to the detector becomes the real sample and produces the sample and label. As for the training process of CGAN, y is increased compared to the GAN formulation, and the details are as follows:

$$min_g max_d V(D,G) = E_{X\sim P_data(x)}[log(D(x|y))] + E_{z\sim P_z(z)}[log(1 - D(G(Z|y)))] \tag{4}$$

By adding y, the formula becomes a binary maximization and minimization problem with conditional probability. The generation of CGAN is controlled by the y condition, which realizes the supervision of the generator and can generate different samples according to different values of y. Since the detector also has a y input, we can choose the type we need when comparing the sample produced by the generator to the actual sample.

2.2. Deep Convolution Generative Adversarial Networks (DCGAN)

CNN has a good effect on supervised learning , But it is rarely used in an unsupervised direction. At first, some researchers tried to combine CNN and GAN, but it could have been better. Alec Radford et al. proposed DCGAN [7], which improved the GAN network structure, GAN-generated images' quality and training stability. DCGAN used full-connection hidden layer removal because the full-connection mode of GAN made training inefficient. In this model, batch normalization [8] is used in the convergence of the model, and it avoids the collapse of the generator and provides the possibility of deeper gradient propagation. Another feature of the model is that in the generator, except for the Tanh function, which is used for the output, the ReLU activation function is selected. In contrast, the Leaky ReLU function is selected in the discriminator to avoid the sparse gradient selection [9]. The reason for using the Tanh function is briefly explained here. Since the pixel range is 0 to 255, the result

 doi:10.20944/preprints202410.1662.v1

of the ReLU function may exceed this range. Fixing the final output value with a function whose range is -1 to 1 is useful. For this purpose, Tanh is used for the final output.

2.3. *Wasserstein GAN (WGAN)*

The author of WGAN [10], conducted an experiment on DCGAN where the generator was fixed, the discriminator was iteratively trained, and a plot of the gradient of the objective function of the generator and the number of iterations was created. From this experiment, with iterative training of the detector, the gradient of the generator decreases rapidly. It can be known that the weak discriminator makes the generator gradient unstable. Sufficient training of the discriminator causes the gradient generator to disappear. Therefore, the training degree of the discriminator is one of the important reasons for the instability of GAN.

According to Goodfellow's article, under optimal discriminator conditions, the generating gradient can be formed from Kullback-Leibler (KL)divergence and Jensen-Shannon(JS) divergence [1]. In this case, state collapse is easily created, meaning that the generated samples focus on the part of the state and have no diversity. For this reason, the author proposed Earth-Mover (EM) distance, compared it with KL divergence and JS divergence and found that the change of EM distance is more sensitive, and a more useful gradient can be suggested. The Kullback-Leibler (KL), Jensen-Shannon(JS), and Earth-Mover (EM) or Wasserstein-1 divergence:

$$KL(P_r \| P_g) = \int log(\frac{P_r(x)}{P_g(x)}) P_r(x) d\mu(x) \tag{5}$$

$$JS(P_r, P_g) = KL(P_r \| P_m) + KL(P_g \| P_m) \tag{6}$$

$$W(P_r, P_g) = \inf_{y \in \prod(P_r, P_g)} E_{(x,y)}[\|x - y\|] \tag{7}$$

Considering the advantages of EM distance, authors tried to apply it to GAN, obtained the Kantorovich-Rubinstein duality theory [11] as the EM distance formula for deformation, and then adapted the neural network method to solve this problem. Neural Network Critic To satisfy the following equation, the Critic truncates the parameter to a range called weight cut after each update.

$$W(P_r, P_\Theta) = \sup_{\|f\|_L \leq 1} E_{x_r}[f(x)] - E_{x_\Theta}[f(x)] \tag{8}$$

$$\max_{w \in W} E_{x_r}[f_w(x)] - E_{z_{(z)}}[f_w(g_\theta(z))] \tag{9}$$

There are three main differences between the critic and the traditional GAN discriminator:

1. Since the detector is responsible for dualities in GAN, and the Critic function is to fit the EM distance, the sigmoid function is removed, and the probability is no longer the output but the overall score.
2. The critic target function no longer includes log functions.
3. There is no need to worry about the effect of too much discriminative training on sample generation. The more educated the reviewer is, the better samples will be produced.

The most important GAN studies in 4G, 5G, and 6G have been reviewed in the following.

3. Generative Adversarial Network in 4G

The authors in [12] proposed a Streamlit-based artificial intelligence platform for NextG networks that allows researchers to build their own AI models and applications against adversarial threats such as: Evaluate and defend escape, poisoning, extraction and interference. The basis and core of this model is an adversarial model that uses adversarial training. Adversarial training is a widely used method to improve the robustness of a machine learning model. Adversarial training generates adversarial examples using the gradient of the victim classifier and then retrains the model with the

 doi:10.20944/preprints202410.1662.v1

adversarial examples and their associated labels. The goal is to train the model with counterexamples closer to real data to make it less sensitive to perturbations. The core and basis of this model was a GAN network.

Radio frequency (RF) estimation was discussed in [13] due to its important role in the planning and optimization of cellular networks. Conventional methods for RF estimation are mainly based on radio propagation models, which need more accuracy and coarse granularity. For this purpose, the authors used a model based on a GAN network called sparsely self-supervised generative adversarial nets (SS-GAN). Unlike existing methods, we develop generative adversarial networks with sparse supervision (SS-GAN), a new data-driven model to generate RF maps of a region from irregular measurement samples. SS-GAN carefully adopts the standard GAN framework, where the generator learns the distribution of real observations under the guidance of a discriminator that recognizes whether the input data are from real samples or from generated outputs. In this work, they used several different branches of features on the RF Map and considered them as the input of the GAN network. SS-GAN follows the basic idea of a game; simultaneously, the model provides significant changes in the objective function. These changes include: 1) The generator input includes the corrupted RF map X_c and the auxiliary data X_a instead of random noise sampling $z \sim p_z(z)$. 2) According to the authors, the second expression of $log(1 - D(G(z)))$ in Equation (10) may suffer from the saturation gradient problem so that the variance of the gradients passing through the generator G is minimal, which may lead to the vanishing gradient problem.

$$\min_{\Theta g} \max \Theta D V(D, G) = \min_{\Theta g} \max \Theta D [E_{x \sim P_{data}(x)}[log D(x)] + E_{z \sim P_z(Z)}[log(1 - D(g(z)))]] \tag{10}$$

To solve this problem, instead of solving the intractable minmax optimization problem, the authors split the equation into two parts, namely VSS-GAN(G) and VSS-GAN(D), and then they tried to minimize both of them alternatingly. Therefore, he formulated the objective function as follows:

$$\begin{cases} \min_{\Theta D} V_{SS-GAN}(D) = \min_{\Theta D} \lambda_D \zeta^D_{adv}, \\ \min_{\Theta D} V_{SS-GAN}(g) = \min_{\Theta g} \lambda_g \zeta^g_{adv}, \end{cases} \tag{11}$$

with

$$\begin{cases} \zeta^D_{adv} = E_{X_{real}}[log(1 - D(X_{real}] \\ + E_{X_c, X_a}[log(0 - D(g(X_c, X_a)))], \\ \zeta^g_{adv} = E_{X_c, X_a}[log(1 - D(g(X_c, X_a)))], \end{cases} \tag{12}$$

SSGAN achieved an average RMSE=0.38 in the minimum prediction in the tested data set. The other two uses of GAN networks in 4G are traffic prediction and attack detection.

4. Generative Adversarial Network in 5G

GAN networks in 5G have achieved acceptable performance in many tasks. In this section, we discuss some applications of GAN networks in 5G. Acquiring large labelled datasets is a tedious and time-consuming manual task. Semi-supervised learning is a suitable technique to overcome this problem. With this in mind, the authors in [14] developed a semi-supervised learning adversarial network (GAN) encrypted traffic classification method called the proposed ByteSGAN embedded in SDN Edge Gateway to achieve the goal of traffic classification accurately to improve network resource utilization further. ByteSGAN uses only a small number of labelled traffic samples and many unlabeled samples to achieve good traffic classification performance by improving the structure and

 doi:10.20944/preprints202410.1662.v1

error performance of the conventional GAN discriminant network using a semi-supervised learning method. In modifying the output structure of the discriminator, it was defined as follows:

$$D(x) = \frac{Z(x)}{Z(x)+1}, where Z(X) = \Sigma_{k=1}^{K} exp[l_k(x)] \tag{13}$$

In correcting the model error, because the discriminator D in ByteSGAN is an N+1 dimensional classifier, its input is a closed data sample. The output is an N+1-dimensional vector that can be represented as a class probability, which in Formula (14) x means false, and in Formula (15) x means true and also belongs to It is class I.

$$P_{model}(y = N+1|x) = \frac{exp(c_{N+1})}{\Sigma_{j=1}^{N+1} exp(c_j)} \tag{14}$$

$$P_{model}(y = i|x, i < N+1) = \frac{exp(c_i)}{\sum_{j=1}^{N+1} exp(c_j)} \tag{15}$$

This model was trained and tested on 1000, 2000, 3000 and 4000 samples of ISCX data and achieved 92.15, 92.92, 93.10, and 93.18 accuracy for the samples, respectively. This model was trained 250,000 times on the data and tended to overfitting in the initial tests.

Filter-GA is another example of using GAN for network traffic classification. Due to the difficulty of collecting and labelling malicious traffic, the distribution of different samples in the existing dataset needs to be more balanced, which leads to low accuracy of malicious traffic classification based on machine learning and deep learning and poor model generalization ability. In [15], the authors proposed a feature image representation method and Generative Adversarial Network with Filter (Filter-GAN) to solve these problems. First, their feature image representation method divided the main session traffic into three parts; then, the Markov matrix was extracted from each part to form a three-channel feature image. This method can transform the original session traffic format into a uniform length matrix and fully characterize the network traffic. Then, they used Filter-GAN feature images to generate some attack samples. Innovations of this work 1) A new method of displaying traffic characteristics was proposed, which embeds each data packet's header and loading characteristics into a characteristic image. This method avoids information loss and redundancy during preprocessing. 2) A generative adversarial network model with a sample filter is designed. The filter is trained and tested to screen the generated samples with real samples so that the generated samples that pass the screening are closer to the distribution of the real samples. Also, according to D, G, the loss function was defined as follows:

$$\zeta_D = E_{X \sim P_data(x)}[log(D(x))] + E_{z \sim P_z(z)}[log(1 - D(G(Z)))] \tag{16}$$

$$\zeta_G = E_{z \sim P_z(z)}[log(1 - D(G(Z)))] \tag{17}$$

This model was evaluated on the Malware Capture Facility dataset[1], which achieved a maximum F1=0.9625.

In [16], the authors investigated the application of a deep learning method for handling unbalanced data problems in network traffic classification. They used a deep network for unsupervised learning called Auxiliary Classifier Generative Adversarial Network to generate composite data samples to balance the main and subclasses. The difference between AC-GAN and GAN is that the input of the G ACGAN generator includes noise z and class label c. In other words, sample G is synthesized

[1] https://www.stratosphereips.org/datasets-malware

 doi:10.20944/preprints202410.1662.v1

from $X_{fake} = G(c, z)$. Another different feature is the output of Discriminator D, which includes probability distributions on LS sources (Equation (18)) and LC class labels (Equation (18)).

$$L_s = E[logP(S = real|X_{real}] + E[logP(S = fake|X_{fake}] \tag{18}$$

$$L_C = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})] \tag{19}$$

Differentiator D is trained to maximize $LS + LC$, and generator G is trained to maximize $LC - LS$. This architecture is similar to GAN, but it is meaningful in generating new examples for a particular arbitrary class. This approach achieved Accuracy=0.9989, F1=0.9543, and AUC=0.9565 on the tested sets. Various other examples of GAN networks that have more comparative aspects have been used to classify network traffic; the most important of these techniques are Conditional GAN [17], Auxiliary Classifier GAN [21], Wasserstein GAN [10], WGAN-GP [22], InfoGAN [19], Least Square GAN [20], Bidirectional GAN [18].

Another example of the use of GAN networks in 5G is the use of these networks in network slicing." 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" [23]. Network slicing enables sharing of a common infrastructure to deploy multiple logical and independent networks. With the help of network slicing and the creation of different types of virtual networks, 5G can provide services with complex and dynamic time-varying resource requirements. However, in this paradigm, different slices may have different resource requirements, and the demands of a slice can be dynamic and vary for its operation. Therefore, the need to predict user requirements regarding different resources and the need for dynamic allocation of these resources becomes critical for 5G performance. GANSlicing [24] is an example of these approaches. GANSlicing [24] is a dynamic software framework for mobile network slicing to predict the resource requirements of internet of things (IoT) applications as well as to improve the quality of experience (QoE) of users. GANSlicing aims to create a global view of network resources that considers cellular networks' physical and virtual capabilities to achieve more efficient utilization and allocation of resources to network segments dynamically. Also, this network handles the task of predicting user demand for different types of resources through the underlying deep generative model. A GAN that can productively mimic the network operations of managers can predict traffic flow from historical data. Hence, GANslicing allows the slicing demand to be predicted.

GAN models have been used as basic models in cyber security. Cybersecurity is a set of technologies, practices, and processes to protect computers, networks, devices, and data from arbitrary cyberattacks, unauthorized access, or malicious activity. Due to the nature and capacity of GANs to mitigate the challenge of unbalanced datasets, GANs have great potential in security and adversarial applications. GANs in intrusion detection systems (IDS) [25–28], malware detection [29,30], detection of rogue Radio Frequency (RF) transmitters [31], malware adaptation/improvement [32–34] [121, 122, 123], black-box Application Programming Interfaces (API) attacks [35] have been used. In some other applications of cyber security, such as password guessing [36] and credit-card fraud detection [**? ? ?**] , GAN networks have achieved stunning results.

5. Generative Adversarial Network in 6G

The emerging sixth generation (6G) is the integration of heterogeneous wireless networks that can seamlessly support networking anywhere, anytime. Few studies have been done on GNA users in 6G. This section discusses some of the most important of these studies.

In [**?**], a new experienced deep reinforcement learning (deep-RL) framework to provide model-free resource allocation for ultra-reliable low-latency communication. Trust (URLLC-6G) is proposed in the downlink of a wireless network. This approach aims to guarantee high end-to-end reliability and low end-to-end latency, under explicit data rate constraints, for each wireless user without any models

 doi:10.20944/preprints202410.1662.v1

or assumptions about user traffic. Specifically, this research proposes a new GAN-based approach to enable the deep-RL framework to compute extreme network conditions and performance in highly reliable systems. This GAN approach is used to pre-train the deep RL framework using a combination of real and synthetic data, thus creating an experienced deep-RL framework exposed to a wide range of network conditions. The trained deep RL framework can be applied to the Orthogonal Frequency Division Multiplexing (OFDMA) resource allocation system. This URLLC-6G resource allocation problem in OFDMA systems is formulated as a power minimization problem under reliability, delay, and rate constraints. To solve this problem using deep-RL with experience, the rate of each user is determined first. Then, these rates are mapped to the resource block and power allocation vectors of the studied wireless system. Finally, each user's end-to-end reliability and latency are used as feedback to the deep-RL framework. Then it is shown that at the fixed point of the deep-RL algorithm, it approximately optimizes the reliability and delay of users. Allocation of resources to minimize average power and maintain reliability, delay and user rate was proposed as follows:

$$\min_{p_{i,j},\wp_{i,j}} \lim_{t\to\infty} \frac{1}{t} \sum_{r=1}^{t} \sum_{i=1}^{N} \sum_{j=1}^{K} p_{ij}(\tau), \tag{20}$$

$$s.t\, PrD_i > D_i^{max} < 1 - \gamma_i^*, \forall \in \aleph \tag{21}$$

$$r_i(t) > \phi(\lambda_i(t), \beta(t), \gamma_i, D_i^{max}), \forall \in \aleph, \forall t \tag{22}$$

$$p_{ij}(t) \geq 0, p_{i,j} \in 0, 1, \forall i \in \aleph, \forall j \in \kappa, \forall t \tag{23}$$

$$\sum_i p_{i,j}(t) = 1, \forall j \in \kappa, \forall t \tag{24}$$

The objective function in (4a) is the average power consumed by the BS. Moreover, the rest of the relations are constraints that must be satisfied. This approach reached a reliability of 99.99% and a latency of less than 1.5 ms.

In addition, the authors of [**?**] proposed the GAN-based Joint Path and Power Optimization (GAN-JTP) algorithm for UAV trajectory prediction and power optimization in network nodes. In this paper, the authors investigated the stealth performance of a cognitive radio network with the help of uncrewed aerial vehicle (UAV) jamming. In particular, the covert transmission of secondary users can be effectively protected from eavesdropping by UAV jamming. For practical consideration, it is assumed that the UAV only knows the channel-specific partial distribution information (CDI), while it does not know the eavesdropping detection threshold. For this purpose, they proposed a model-based generative adversarial network (MD-GAN) optimization framework consisting of a generator and a detector, where the channel information is unknown, and the detection threshold is the learned weights. Then, a GAN-based joint path and power optimization (GAN-JTP) algorithm was developed to train the MD-GAN optimization framework for covert communication, which leads to the joint solution of the UAV path and power transmission to maximize stealth. Their proposed GAN training method is a minimum-maximum two-player game, which is expressed by the following relationship:

$$\min_D max_G V(G, D) = R^{SR,c}[n] + D(G(s)[n]) \tag{25}$$

In this regard, $R^{SR,c}[n]$ represents the covert rate, and $D(G(s)[n])$ is the probability of detection errors in the nth time slot. $s = Q, I$ represents the current state observed from the network environment. More details of this relationship are available in the article. This approach was able to achieve an Average covert rate=0.152. High-quality reliability must be provided by 6G to meet the expectations of mobile phone users. Artificial Intelligence (AI) is considered one of the most important components of 6G. Trust management based on artificial intelligence is a promising paradigm for providing reliable and trustworthy services in 6G. In [**?**], a GAN-based trust management method for 6G wireless networks is presented. The trustworthiness of a network device can be determined through feature

 doi:10.20944/preprints202410.1662.v1

extraction and labelling for its trust vector. At the same time, good performance requires many trust vectors to be used together for clustering. Inspired by the generative GAN model, the trust decision problem can be solved by estimating whether experimental trust vectors satisfy the distribution of those labelled with a higher trust level. In this research, the authors first adopted type 2 fuzzy logic to evaluate the trust of devices while reducing the problem of trust uncertainty. Then, they prepared a dataset for training a GAN-based autoencoder through feature extraction and data labelling for the constructed trust vectors. Finally, they made a trust decision based on the trained autoencoder and experimental thrust vectors.

6. 5G Network Metrics

Trust in modern telecommunications networks plays a vital role as a driver of the technological and market success of any new telecommunications technology or service. Most technological approaches to this problem are focused only on network security. They need to include a factor such as quality of service (QoS), which plays a vital role in the formation of trust from both consumers and service regulators/providers. The authors have presented a perspective on trust in 5G networks at the QoS level and management. Some of them have provided QoS requirements for 5G networks based on business models and the need to ensure users' trust in the networks. Some of them have also provided Qos factors in certain activities of the 5G network. In the following, we will discuss the Qos metrics in both views in more detail [37].

The leading organizations in the standardization and development of international telecommunication technologies, such as ITU, 3GPP, IEEE and ETSI, have not provided a precise definition of "reliable network," and there is no precise limit of a reliable network. However, trust in the communication network significantly affects the choice of communication operator consumers, the regulation of operators' activity by government bodies, and the market demand for communication services and equipment. Trust in-network or communication technology has various aspects, the most important of which, from the point of view of developers and users, is the market and regulatory aspect that can help develop the network and technology and increase the attractiveness of services. Therefore, communication networks and technologies must comply with market and regulatory trust requirements [37]. As defined by Kaspersky Internet Security [38], a trusted network is a network that can be considered completely secure, where your computer or device is not exposed to attacks or unauthorized attempts to access your data.

7. 6G Network Metrics

While 5G mobile systems are being deployed around the world, researchers are beginning to envision 6G networks to integrate sensing, communication, computing, and control functions. Following the 5G wireless network, the 6G wireless network is expected to provide widespread connectivity to millions of connected devices with different quality of service (QoS) requirements, pervasive coverage, a high degree of embedded artificial intelligence (AI), energy efficient use, and adaptive network security to provide In addition, unmanned aerial vehicles (UAVs) can play an important role in the 6G ecosystem, as flying devices are expected to densely occupy the airspace and act as a network layer between terrestrial and space networks. do As a result, UAVs communicate with ground stations and satellites, forming a space-air-ground network that paves the way for fully integrated 6G heterogeneous networks. 6G or the sixth-generation wireless communication system is the successor of 5G cellular technology. 6G networks are expected to be able to use higher frequencies than 5G networks, allowing for higher data rates and overall 6G network capacity. A much lower latency level would certainly be required. 6G mobile technology is generally expected to support one-microsecond or even sub-microsecond latency communications, making communications nearly instantaneous. Some organizations have already started early development for 6G. Among these organizations, we can mention South Korea Electronics and Telecommunications Research Institute, The Ministry of Industry and Information Technology, MIIT, China, The University of Oulu, Finland,

 doi:10.20944/preprints202410.1662.v1

and USA initiatives. 6G mobile communication technology is built on the basis of 5G technology. Some of the existing new technologies will be further developed for 6G:

- **Millimeter wave technologies:** Using much higher frequencies in the frequency spectrum opens up more spectrum and also allows for a very wide channel bandwidth. With the massive data rates and bandwidth required for 6G, millimeter wave technologies will be further developed, possibly extending into the TeraHertz region of the spectrum.
- **Massive MIMO:** Although MIMO is used in many applications from LTE to Wi-Fi and more, the number of antennas is relatively limited -. The use of microwave frequencies opens up the possibility of using dozens of antennas on a single piece of equipment, as this is a real possibility due to the size and spacing of the antennas in terms of wavelength.
- **Dense networks:** reducing the size of cells provides a more effective use of the available spectrum. Techniques are needed to ensure that small cells in the macro grid deployed as femtocells can perform satisfactorily.

8. Gap Analysis and Future Works

One of the most obvious researches in this field is in the Quality of services metrics section, which has a significant volume of studies. In this section, the metrics are considered descriptive and interval. And this issue can be addressed as a classification problem (prediction of class) and a regression problem (prediction of value in an interval). The challenge ahead in this issue is that some intervals or classes rarely occur, for example, in the Jitter metric of the Good class, it is repeated less frequently than in other classes. This problem is an unbalanced class problem for which the following approaches can be used.

- **Resampling** Time series forecasting is a challenging task where the non-stationary characteristics of the data require strict settings for forecasting tasks. A common problem is the skewed distribution of the target variable, where some intervals are highly significant but severely underrepresented. Standard regression tools focus on the average behaviour of the data. However, the goal in many time series forecasting tasks is the opposite. For example, predicting rare values is one of these challenges. A standard solution for time series forecasting with unbalanced data is to use resampling strategies that operate on the learning data by changing its distribution in favour of a particular bias. Various algorithms have been proposed for this purpose. For example, algorithms [39] and [40] can be used.
- **High-dimensional Imbalanced Time-series classification (OHIT),zhu2022minority**: OHIT first uses a density ratio-based joint nearest neighbor clustering algorithm to capture minority class states in a high-dimensional space. Depending on different clustering algorithms, this clustering can get different results. It then, for each mode, applies the shrinkage technique of a large-dimensional covariance matrix to obtain an accurate and reliable covariance structure. Finally, OHIT generates structure-preserving synthetic samples based on a multivariate Gaussian distribution using the estimated covariance matrices.
- **IB-GAN,deng2022ib**: The standard methods of class weight, oversampling, or data augmentation are the approaches studied in (An empirical survey of data augmentation for time series classification with neural networks). These approaches are parametric. Parametric approaches do not always yield significant improvements for predicting the minority classes of interest. Non-parametric data augmentation with generative adversarial networks (GAN) is a promising solution. For this purpose, the authors have proposed Imputation Balanced GAN (IB-GAN), which combines a new method of augmentation and data classification in a one-step process through an imputation-balanced approach. IB-GAN uses imputation and resampling techniques to generate higher-quality samples from randomly masked vectors than white noise and balances the classifier through a pool of real and synthetic samples. Hyperparameter imputation pmiss allows to regularize of the classifier variation by adjusting the innovations introduced through generator imputation. IB-GAN is simple to train and model, pairing each deep learning classifier

 doi:10.20944/preprints202410.1662.v1

with a generator-discriminator pair, resulting in higher accuracy for trim observed classes. The basis of this approach is a GAN network that tries to generate cases similar to the minority class.

Sequential learning: The problem of sequential learning can be formulated mathematically. Let $(x_i, y_i)_{i=1}^{N}$ be a set of N training samples. Each sample is a pair of sequences x_i, y_i where $x_i = x_{i,1}, x_{i,2}, ..., x_{i,T_i} and y_i = y_{i,1}, y_{i,2}, ..., y_{i,Ti}$. The goal is to build a classifier h that can predict a new label sequence $y = h(x)$ given an input sequence x[1].

Challenges in sequential learning:

- In sequential learning, the entire sequence $x_1, ..., x_T$ is available before we predict the y values, whereas, in time series prediction, we only have a prefix of the sequence up to the current time $t + 1$.
- In time series analysis, we have the actual observed y values up to time t, whereas, in sequential learning, we are not given any y values and have to predict them all.

Recurrent Neural Networks(RNNs) are feed-forward networks that add the concept of time to the model, which is defined by edges in adjacent steps. Edges that connect adjacent time steps are called recurrent edges. These networks have been used the most in sequential learning and time series analysis tasks. But RNN networks have the following challenges:

1. Long-range dependencies
2. Gradient vanishing and explosion
3. Large # of training steps
4. Parallel computation

A paper, "Attention Is All You Need," introduced an encoder-decoder architecture based on attention layers, which the authors called transformers. One of the main differences of this architecture is that the input sequence can be passed in parallel so that the GPU can be used effectively and the training speed can be increased. Also, this architecture is based on multi-headed attention, so it easily overcomes the vanishing gradient problem. The most important difference between this architecture and RNN is as follows:

1. Facilitate long-range dependencies
2. No gradient vanishing and explosion
3. Fewer training steps
4. No recurrence that facilitates parallel computation

The main components of a transformers network are as follows:

- **Encoder Block:**The encoder consists of a stack of $N = 6$ identical layers. Each layer has two sub-layers. The first one is a multi-head self-attention mechanism and the second one is a fully connected and simple feed-forward network. Also, a residual connection should be used around each of the two sub-layers, and then layer normalization is done. That is, the output of each sublayer is $LayerNorm(x + Sublayer(x))$, where $Sublayer(x)$ is a function implemented by the sublayer itself. To facilitate these remaining connections, all sub-layers in the model, as well as embedded layers, produce outputs with dimensions of $dmodel = 512$.
- **Decoder Block:**The decoder also consists of a stack of $N = 6$ identical layers. In addition to the two sub-layers in each encoder layer, the receiver introduces a third sub-layer that performs multi-head attention on the output of the encoder stack.
- **Scaled Dot-Product Attention:** Here, the query along with the keys are divided by $\sqrt{d_k}$, then a Softmax is applied on them to determine the weight of the values. In practice, the attention function is simultaneously computed on a set of queries packed together in a Q matrix. Keys and values are also packed together in K and V matrices. Next, the output matrix is calculated as follows:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})v \tag{26}$$

 doi:10.20944/preprints202410.1662.v1

- **Multi-Head Attention:** Instead of implementing a single attention function with model dimensional keys, values, and queries, the authors found that it would be more beneficial to linearize the queries, keys, and values h times with different learned linear predictions of dimensions dk, dk, and dv, respectively. Was. They executed the attention function in parallel on each of these predicted versions of the queries, keys, and values and obtained the following dv output values:

$$MultiHead(Q, K, V) = Concat(head_1, \ldots, head_h)W^O \tag{27}$$

$$wherehead_i = Attention(QW_i^Q, KW_i^K, VW_i^v) \tag{28}$$

where $W_i^Q \in R^(d_{model} * d_k), w_i^K \in R^(d_{model} * d_k), W_i^V \in R^(d_{model} * d_v)$ and $W^O \in R^(hd_v * d_{model})$

References

1. Goodfellow, Ian, et al. "Generative adversarial networks." Communications of the ACM 63.11 (2020): 139-144.
2. I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative Adversarial Nets," in Proc. of International Conference on Neural Information Proces.
3. W. Cao, X. Wang, Z. Ming, and J. Gao, "A Review on Neural Networks with Random Weights," Neurocomputing, vol. 275, pp. 278-287, 2017. Article (CrossRef Link)
4. L. Zhang and P. N. Suganthan, "A Survey of Randomized Algorithms for Training Neural Networks," Information Sciences, vol. 364, pp. 146-155, 2016. Article (CrossRef Link)
5. S. Wang, T. Z. Huang, J. Liu, and X. G. Lv, "An alternating iterative algorithm for image deblurring and denoising problems," Communications in Nonlinear Science and Numerical Simulation, vol. 19, no. 3, pp. 617-626, 2014. Article (CrossRef Link)
6. M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," arXiv preprint arXiv:1411.1784, 2014. Article (CrossRef Link)
7. A. Radford, L. Metz, and S. Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks," Computer Science, 2016. Article (CrossRef Link)
8. Y. Li, N. Y. Wang, J. P. Shi, X. Hou, and J. Liu, "Adaptive Batch Normalization for practical domain adaptation," Pattern Recognition, vol. 80, pp. 109-117, 2018. Article (CrossRef Link)
9. B. Xu, N. Y.Wang, T. Q. Chen, and M. Li, "Empirical Evaluation of Rectified Activations in Convolutional Network," Computer Science, 2015. Article (CrossRef Link)
10. Arjovsky, Martin, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks." International conference on machine learning. PMLR, 2017.
11. Villani, Cédric. Optimal transport: old and new. Vol. 338. Berlin: springer, 2009.
12. Kuzlu, Murat, et al. "A Streamlit-based Artificial Intelligence Trust Platform for Next-Generation Wireless Networks." arXiv preprint arXiv:2211.12851 (2022).
13. Li, Zhuo, et al. "Sparsely self-supervised generative adversarial nets for radio frequency estimation." IEEE Journal on Selected Areas in Communications 37.11 (2019): 2428-2442.
14. Wang, Pan, et al. "Bytesgan: A semi-supervised generative adversarial network for encrypted traffic classification of sdn edge gateway in green communication network." arXiv preprint arXiv:2103.05250 (2021).
15. Cao, Xin, Qin Luo, and Peng Wu. "Filter-GAN: Imbalanced Malicious Traffic Classification Based on Generative Adversarial Networks with Filter." Mathematics 10.19 (2022): 3482.
16. Vu, Ly, Cong Thanh Bui, and Quang Uy Nguyen. "A deep learning based method for handling imbalanced problem in network traffic classification." Proceedings of the 8th international symposium on information and communication technology. 2017.
17. Mehdi Mirza and Simon Osindero. Conditional Generative Adversarial Nets. arXiv:1411.1784 [cs, stat], November 2014.
18. Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial Feature Learning. arXiv:1605.09782 [cs, stat], April 2017.
19. 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.

20. 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.
21. 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.
22. 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, De
23. Navidan, H., Moshiri, P. F., Nabati, M., Shahbazian, R., Ghorashi, S. A., Shah-Mansouri, V., Windridge, D. (2021). Generative Adversarial Networks (GANs) in networking: A comprehensive survey evaluation. Computer Networks, 194, 108149.
24. Gu, Ruichun, and Junxing Zhang. "Ganslicing: A gan-based software defined mobile network slicing scheme for iot applications." ICC 2019-2019 IEEE International Conference on Communications (ICC). IEEE, 2019.
25. 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.
26. 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.
27. Zilong Lin, Yong Shi, and Zhi Xue. IDSGAN: Generative Adversarial Networks for Attack Generation against Intrusion Detection. arXiv:1809.02077 [cs], June 2019.
28. 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.
29. 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.
30. 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 ElSayed 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.
31. 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.
32. Maria Rigaki and Sebastian Garcia. Bringing a GAN to a KnifeFight: 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.
33. Weiwei Hu and Ying Tan. Generating Adversarial Malware Examples for Black-Box Attacks Based on GAN. arXiv:1702.05983 [cs], February 2017.
34. 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.
35. 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.

 doi:10.20944/preprints202410.1662.v1

36. Briland Hitaj, Paolo Gasti, Giuseppe Ateniese, and Fernando PerezCruz. PassGAN: A Deep Learning Approach for Password Guessing. In Applied Cryptography and Network Security, pages 217–237, June 2019.
37. Tikhvinskiy, Valery, and Grigory Bochechka. "Quality of Service in 5G Network." Opportunities in 5G networks: A research and development perspective (2016): 97.
38. Tikhvinskiy, Valery, Grigory Bochechka, and Andrey Gryazev. "QoS requirements as factor of trust to 5G network." Journal of Telecommunications and Information Technology 1 (2016): 3-8.
39. Saripuddin, Mulyana, Azizah Suliman, and Sera Syarmilla Sameon. "Impact of resampling and deep learning to detect anomaly in imbalance time-series data." 2022 14th International Conference on Computer Research and Development (ICCRD). IEEE, 2022.
40. Bagui, Sikha, and Kunqi Li. "Resampling imbalanced data for network intrusion detection datasets." Journal of Big Data 8.1 (2021): 6.
41. Zhu, Tuanfei, et al. "Minority oversampling for imbalanced time series classification." Knowledge-Based Systems 247 (2022): 108764.
42. Deng, Grace, et al. "Ib-gan: A unified approach for multivariate time series classification under class imbalance." Proceedings of the 2022 SIAM International Conference on Data Mining (SDM). Society for Industrial and Applied Mathematics, 2022.