Proceedings of the 2023 International Conference on Software Engineering and Machine Learning
DOI: 10.54254/2755-2721/8/20230129

The study and implementation on adversarial attack algorithm based on deep convolutional generative adversarial network

Mingze Ma

The Department of Computer Science, The University of Liverpool, L3 5UE Liverpool, the UK.

sgmma4@liverpool.ac.uk

Abstract. The field of machine learning has been growing rapidly in recent years, with significant advancements in computer vision. However, this progress has also led to increased concerns over the safety and security of machine learning systems. Despite previous research efforts in this area, traditional adversarial algorithms continue to fall short in defending against attacks in most circumstances. As a response to this challenge, this study seeks to develop an adversarial algorithm based on Deep Convolutional Generative Adversarial Network (DCGAN), which aims to reconstruct the data distribution of input data and generate new data to improve model robustness. To evaluate the efficacy of the proposed method, a Deep Neural Network (DNN) classifier is employed to perform classification on the generated dataset. The experimental results suggest the potential feasibility of the proposed hypothesis, but further improvement is required to strengthen the defence mechanism. Overall, this study contributes to the ongoing efforts to enhance machine learning safety and security in practical applications.

Keywords: DCGAN, adversarial algorithm, classification, attack method.

1. Introduction

The advancement of deep-learning-based algorithm is unprecedented. It has been pervasively employed in the lives of human. For instances, chatGPT has recently gained popularity and thus become the public focus, which is able to generate logical conversation and response. Reducing the complexity of traditional Automatic Modulation Classifier (AMC), whose classification tasks depend on likelihood or feature [1], the deep learning model can derive better result without redundant coding program. Nevertheless, as DL has been making surprising progress, potential threats arise at the meantime. Nowadays, the truth is that raw data used in model tends to be contaminated by random noise. Across various fields, the impact sometimes would beyond estimation. In a study conducted by Finlayson [2], it was demonstrated that even a slight perturbation added to raw data can result in misclassification of benign tumors as malignant. The perturbation introduced was so subtle that it was imperceptible to the human eye. Additionally, the cost associated with intentionally adding such noise to the original data is almost insignificant. Despite this seemingly trivial impact, misclassification of benign tumors as malignant can lead to grave consequences. Therefore, there is a pressing need to develop robust machine

DOI: 10.54254/2755-2721/8/20230129

learning algorithms that are resilient to such attacks to ensure the safety and reliability of medical diagnosis and treatment.

In avoidance of such tragic occasion, numerous studies are conducted on adversarial algorithms. Under most circumstances, the attack algorithm does not necessarily have knowledge about the existing model. Therefore, black-box attack is often counted as a feasible and convenient way of attacking. The scholar has made an attempt to apply defensive distillation technique to Convolutional Neural Network (CNN) to identity latent fraudulent data [3]. In spite of the little success the experiment having achieved at the beginning, the model performance tends to be compromised when original data is mixed with more perturbation. Hence, the traditional adversarial algorithms are unable to genuinely be employed against attack algorithms due to its instability. Since the proposal of Generative Adversarial Network (GAN) in 2014, public attention has been drawn to this emerging subject. Known as a semi-supervised model, it wisely combines two neural networks where one is for creating and another one for examining to provide new solutions towards many challenging questions. For examples, image generation, style transformation, target detection and classification task.

Typically, the GAN consists of generator and discriminator. The discriminator will be trained before generator to modelling the data. Afterwards, the discriminator is used to judge generator's output. Subsequently, the generator utilizes random noise to generate e.g., image within each iteration which will be sent back to the discriminator. Theoretically, discriminator is responsible for marking the output of generator as low as possible, while generator is dedicated to minimizing the difference from original image. At last, the generator will give decent output (which could be an image).

Having thoroughly understand the theory behind GAN, Deep Convolutional Generative Adversarial Network (DCGAN) which is an excellent variety of GAN is chosen as the adversarial algorithm in this study. Inspired by the dissertation released by [4], the quality of images generated from DCGAN is prone to be higher compared to the normal GAN. Indeed, normal GAN is rarely used in computer vision for its limited computational capability. The previous studies related to DCGAN is successful where the images generated were no longer blurry and unrecognizable. Aside from image generation, a small quantity of scholars ever experimented on DCGAN to accomplish classification tasks. The subsequent sections of this paper will provide a detailed account of the DCGAN modeling process and the experimental protocols employed to evaluate its performance.

2. Methodology

2.1. Introduction to the MNIST dataset

The dataset involved in this study is MNIST, which was initially developed by the U.S. National Institute of Standards and Technology (NIST) and has become a widely adopted benchmark in the field of machine learning. The dataset collects 250 different handwritten digits of people in a good variety of occupations and social backgrounds, among whom 50% are students under education and the rest are population census staff. It is a free dataset accessible to machine learning beginner. Since it is essentially images of digits, the classification task will be easier to be conducted on. The visualization of some sample data on the MNIST dataset can be observed in Figure 1.

DOI: 10.54254/2755-2721/8/20230129

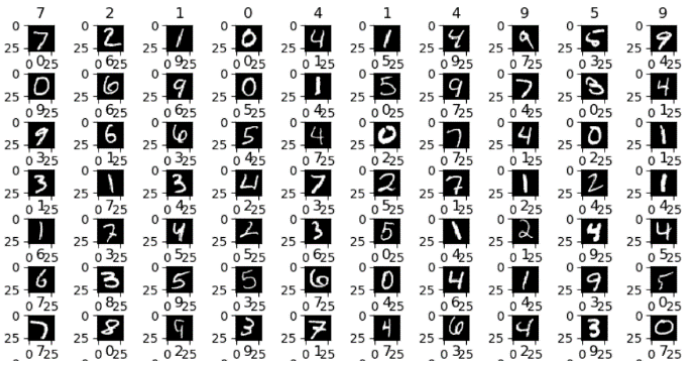

Figure 1. The sample data on the MNIST dataset.

2.2. Implementation details

The model is mainly implemented with Keras, which is reputational powerful machine learning framework supported by Python. All machine learning optimization techniques including various optimizers, mini-batch, normalization algorithms etc. are equipped with. Additionally, mathematical package is imported to perform necessary computation. Matplotlib is used for study result demonstration. The coding style follows Sequential structure. During training, GPU is used to speed up the computation, which version is RTX 3050.

2.3. Data preprocessing details

In convenience of research, the target dataset is imported from Keras internal encapsulated dataset package, which is essential the same after processed in certain manner. Tensorflow is an open-source machine learning framework providing a large quantity of embedded algorithm and modules. It will be used in this project. It is then read in manner of grey scale map in consideration of further classification task. Through observation, it consists of 60, 000 training images and 10, 000 test images. In addition, normalization is applied to the original dataset.

Normalization scales the data within [-1,1], which is also within the range of tanh. As a result, the mean value of original dataset will be 0, in which case dataset complies certain type distribution. Besides, it accelerates model training process, allowing model to fit quickly. Here is the basic information about training data after processed: The channel of image is set to 1, since they are black and white image. The encoded data comprises 28 rows and 28 columns. There is hidden encoding in dataset, whose size is 100. In the following building of DCGAN model, the dimension of it determines the correctness of model. For the reason that the project is committed to providing solution to adversarial attack implemented by DCGAN, the attack algorithm is required to generate “contaminated data”, which is indeed officially named as perturbation. The attack algorithm involved is FGSM. It was first proposed by Goodfellow in research into adversarial method [5]. In nature, FGSM is a black box attack strategy that operates on the targeted data without prior knowledge of its composition. A global perturbation is applied to the data, based on the corresponding gradient, which exhibits a linear growth pattern. In order to generate samples that are imperceptible to human vision, the perturbation is confined within a specified interval, which, in this project, is set to [-0.1, 0.1]. Despite minor statistical variations, classifiers commonly exhibit vulnerability to FGSM, leading to difficulties in accurately performing their designated tasks. This perturbation serves as evidence to support the hypothesis that DCGAN can function as an effective adversarial algorithm. With the data now prepared for modelling, the results of the FGSM attack reveal its efficacy as a reliable and efficient attack method. In Zhang’s research [6], FGSM is used as voice perturbation in speaker model, which contributes tremendously to the success of experiment.

DOI: 10.54254/2755-2721/8/20230129

2.4. DCGAN modeling

Following the acquisition of the appropriate dataset for both training and testing, the project proceeds to the modelling stage. In accordance with theoretical principles, the proposed DCGAN model comprises two integral components, namely the generator and the discriminator. Figure 2 and Figure 3 provide the architecture of the generator and discriminator used in this study, respectively.

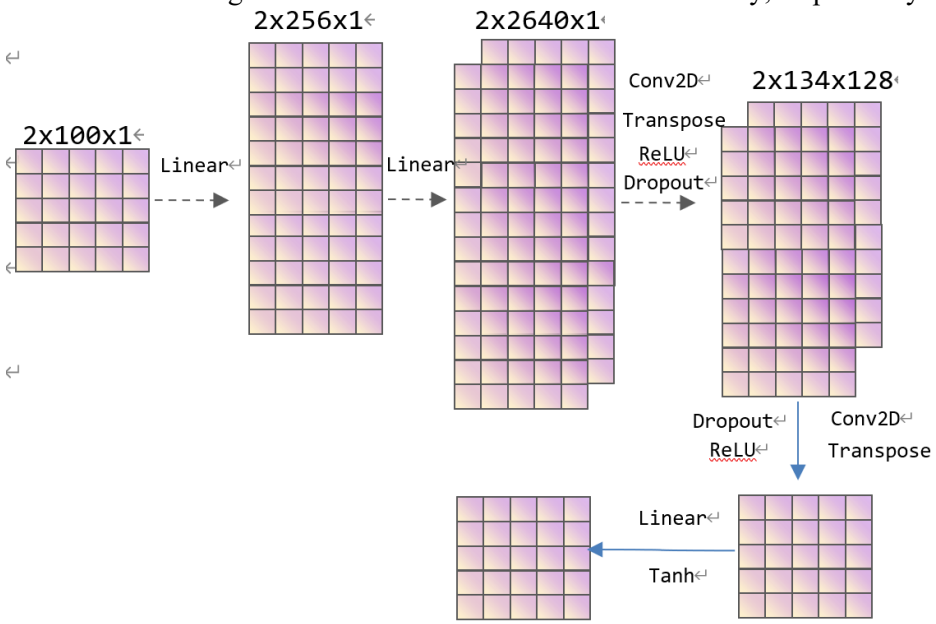

Figure 2. The architecture of the Generator proposed in this study.

Concerning the generator component, DCGAN differs from normal GAN by replacing conventional pooling layers with convolutional layers and eliminating full connected layers, which is also the feature of DCGAN. However, the convolutional layers tend to discard significant amounts of features following convolution computation. In avoidance of this defect, the model will execute upsampling after each time of convolution. Upsampling is a key technique employed in image generation and pattern recognition, including nearest-neighbour, bilinear and bicubic interpolation. It is intended to enhance the original feature of image in prevention of excessive feature loss [7]. The upsampling technique involved here is transpose convolution. In contrast to GAN, batch normalization is added in between to constrain data distribution between each layer.

Although original dataset was processed with normalization, it is not likely to comply with normal distribution after numerous propagations. Undoubtedly, batch normalization plays a crucial role in regularizing each batch of data during the course of training. To be noted that ReLU activation function is used in the hidden layers of generator, while Tanh is used in the output layer.

The construction of discriminator appears to be more complex, whose structure in essence is reverse of the generator's.

DOI: 10.54254/2755-2721/8/20230129

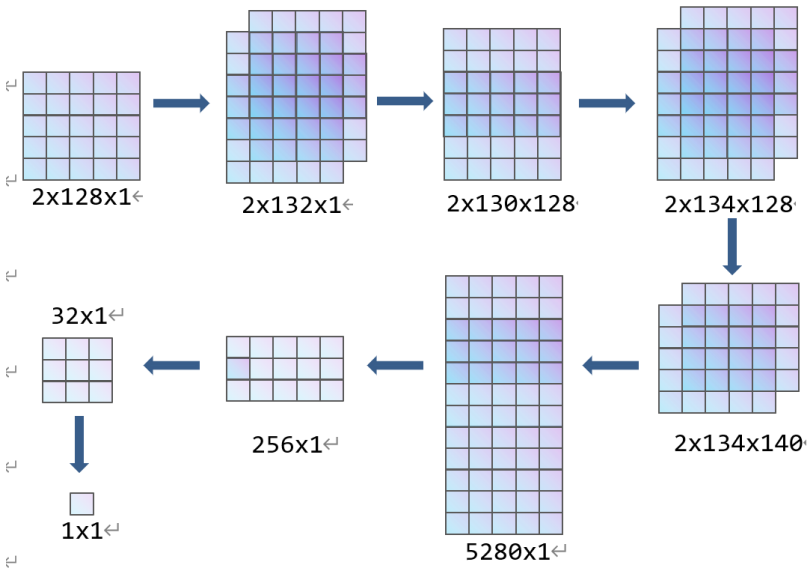

Figure 3. The architecture of the Discriminator proposed in this study.

Theoretically, discriminator model will be trained in ahead of the generator. In light of the addition of perturbation to original data, the judgement of discriminator directly affects the quality of images generated by generator. Therefore, emphasis should be put on reinforcing the robustness of the discriminator. Likewise, the former pooling layers are substituted by convolutional layers with stride 2. Different from generator module, LeakyReLU activation function is chosen after convolution rather than ReLU. LeakyReLU could rectifie and trims unnecessary gradients and smoothens the learning rate curve. Another measurement of alleviating overfitting is adding dropout layer (Figure 4) neutralizing specified number of random neurons.

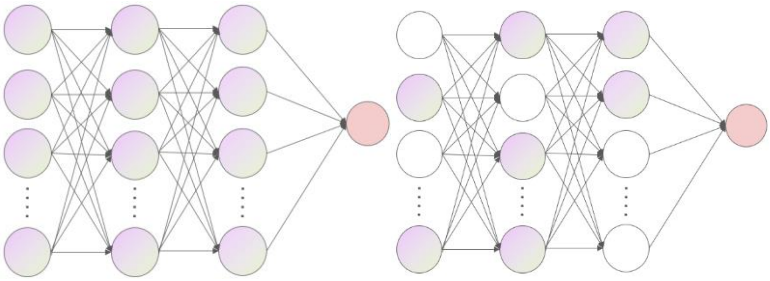

Figure 4. The schematic of the Dropout.

As a whole, 4 sets of convolutional layers as well as corresponding operations are set accordingly. The zero-padding is put in the first set and last set will be flattened. The activation function employed is sigmoid. To be more specific, the mini-batch size is 128 which is optimized with Adam Optimizer and batch gradient descend. The slope of LeakyReLU is established as 0.2, while the learning rate is set to 0.00018. Additionally, the momentum value is determined as 0.5.

2.5. The DNN classifier

To verify the hypothesis that DCGAN-generated images can mitigate the threat posed by FGSM, a DNN classifier is employed. The classifier's accuracy and precision are expected to reflect the robustness of the DCGAN discriminator. Using the generated data by DCGAN, a DNN is constructed for comparison with the classification of the original dataset, for the purpose of demonstration.

DOI: 10.54254/2755-2721/8/20230129

3. Result and discussion

3.1. The performance of the model

The visualization of generated images is shown in Figure 5.

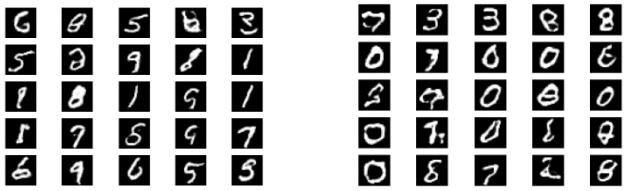

Figure 5. Generated images of DCGAN.

The first images generated is fairly clear despite few blurry numbers. The right image is generated with dropout applied.

In order to highlight the disparities between traditional GAN and DCGAN, images shown in Figure 6 are presented side by side for visual comparison. Evidently, the image generated by DCGAN exhibits a higher quality than that generated by GAN, as observed through visual inspection. That aside, the outline of each number is more distinct. Specifically, GAN generates same number within an image which is not ideal. The majority of numbers generated are NINE and ZERO.

Figure 6. Generated images of GAN.

Details involved in the process of training are also recorded (Figure 7). The graph below intuitively illustrates the loss of each model. Initially, discriminator's loss decreases sharply after few hundreds of iterations and then fluctuates steadily on ends. Instead, in spite of obvious decrement in loss at the start of training, the generator's loss ascends dramatically after turning point.

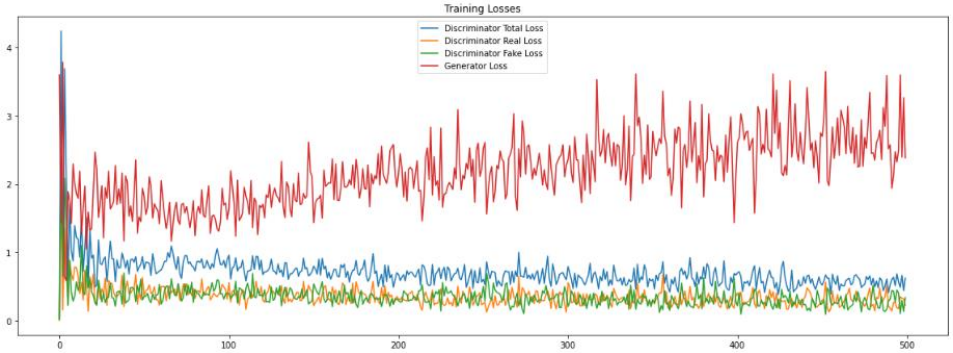

Figure 7. The loss Curve during the training process in terms of the generator and discriminator.

The DNN classifier performances on both original dataset and generated one are visualized in Figure 8.

DOI: 10.54254/2755-2721/8/20230129

```
Epoch 1: 117s, trn_loss: 1.9766, trn_acc: 25.37%, adv_loss: 2.1033, adv_acc: 19.34%
Epoch 2: 116s, trn_loss: 1.2805, trn_acc: 48.35%, adv_loss: 1.7091, adv_acc: 28.85%
Epoch 3: 115s, trn_loss: 1.1223, trn_acc: 54.02%, adv_loss: 1.5541, adv_acc: 37.94%
Epoch 4: 115s, trn_loss: 0.9545, trn_acc: 59.15%, adv_loss: 1.4047, adv_acc: 37.94%
Epoch 5: 115s, trn_loss: 0.8345, trn_acc: 64.14%, adv_loss: 1.3331, adv_acc: 49.53%
Epoch 6: 118s, trn_loss: 0.8125, trn_acc: 63.83%, adv_loss: 1.2676, adv_acc: 50.47%
Epoch 7: 119s, trn_loss: 0.7965, trn_acc: 66.91%, adv_loss: 1.1825, adv_acc: 53.50%
Epoch 8: 118s, trn_loss: 0.8290, trn_acc: 64.17%, adv_loss: 1.1267, adv_acc: 57.48%
Epoch 9: 119s, trn_loss: 0.7989, trn_acc: 66.93%, adv_loss: 1.0517, adv_acc: 59.43%
Epoch 10: 123s, trn_loss: 0.8030, trn_acc: 65.86%, adv_loss: 1.0373, adv_acc: 60.37%
```

Figure 8. The classification result obtained from the DNN-1.

In contrast, the classification with clean data is experimented as well as shown in Figure 9.

```
Epoch 1: 172s, trn_loss: 1.0486, trn_acc: 61.32%, adv_loss: 1.3851, adv_acc: 50.48%
Epoch 2: 142s, trn_loss: 0.7493, trn_acc: 72.79%, adv_loss: 1.2329, adv_acc: 54.49%
Epoch 3: 139s, trn_loss: 0.5819, trn_acc: 79.88%, adv_loss: 1.1779, adv_acc: 57.61%
Epoch 4: 141s, trn_loss: 0.5418, trn_acc: 80.65%, adv_loss: 1.1448, adv_acc: 58.59%
Epoch 5: 143s, trn_loss: 0.4463, trn_acc: 84.08%, adv_loss: 1.0747, adv_acc: 61.69%
Epoch 6: 144s, trn_loss: 0.4501, trn_acc: 83.82%, adv_loss: 1.1193, adv_acc: 60.76%
Epoch 7: 144s, trn_loss: 0.3926, trn_acc: 85.93%, adv_loss: 1.0031, adv_acc: 63.61%
Epoch 8: 149s, trn_loss: 0.4003, trn_acc: 85.24%, adv_loss: 1.1016, adv_acc: 60.42%
Epoch 9: 154s, trn_loss: 0.3644, trn_acc: 86.83%, adv_loss: 1.0230, adv_acc: 65.03%
Epoch 10: 146s, trn_loss: 0.3663, trn_acc: 86.51%, adv_loss: 1.0200, adv_acc: 64.29%
```

Figure 9. The classification result obtained from the DNN-2.

Statistically, the accuracy with perturbation is 65% and accuracy without it is 86%.

3.2. Discussion

Beyond initial expectation, discriminator is not susceptible to the influence of attack data due to the consistent reduction of loss. To expound this, it might be caused by innate attributes of DCGAN where convolution is involved so that attack data is likely to be filtered during this course.

The generator appears to be more vulnerable which are prone to be negatively affected even if the effect of attack data is indirect. Consequently, the generator's loss shows ascending trend. As a matter of fact, the quality of image generated is not necessarily always better along training progress, where images become blurrier in certain epochs while clearer in others. The possible explanation could be the changes happening to discriminator during training where the criterion given by discriminator is not fixed since it is being trained, the image generated afterwards will change accordingly.

Such instability finally results in a large number of defects. To maximize the classification accuracy later, the last 10 images are chosen as the input for DNN classifier. Compared to the data without contamination, the result is definitely worse than original data. Nevertheless, the DCGAN accomplishes generating relatively recognizable images used for further classification with contaminated data. Owing to its semi-supervised learning pattern, the original attack method is unable to take effect in a way it should have been. Although it does not indicate the alleviation of threat, the output of DCGAN model approximates the images anticipated and prevent model from being misled to some extends. By referring to relevant study, Laykaviriyakul et al. successfully devised defence algorithm based on GAN [8]. They first used GAN to purify the input data and then took the output as input for classifier.

Viewing throughout current Machine Learning (ML) models which are equipped with tremendous capabilities, majority of them fail to fulfil the functions they should have due to low resilience to attack algorithm. Traditional defence algorithms which pretrain the attack data then feed it to classification model tend to fail when attack algorithm's computing rate exceeds its threshold. However, GAN based defence algorithm could possibly erases the malicious data by reconstructing the data distribution. With more commitment made to this area, GAN used as adversarial attack algorithm would be promising. In perspective of future study, more GAN based adversarial algorithms should be taken into consideration due to its large varieties. For instance, DA-GAN's potential has been exploited to assist the defence against cyber threat [9, 10]. It is widely acknowledged that cyber security has always been a hardcore problem drawing public attention for ages. The effective measurements available are rarely taken. Possibly in the near future, it can be used as a sharp weapon to defend ourselves.

DOI: 10.54254/2755-2721/8/20230129

4. Conclusion

This study endeavors to devise an adversarial algorithm using Deep Convolutional Generative Adversarial Network (DCGAN) to enhance the robustness of models by reconstructing input data distribution and generating new data. In order to assess the effectiveness of the proposed approach, a Deep Neural Network (DNN) classifier is utilized to perform classification on the generated dataset. The obtained experimental results provide evidence supporting the potential feasibility of the proposed hypothesis, however, further enhancements are necessary to strengthen the defense mechanism. The study provides significant contributions to the ongoing efforts aimed at improving the safety and security of machine learning in practical applications. In the future, the further experiments were considered carried out in more intricate GAN-related models e.g. CycleGAN.

References

[1] Abdel‐Moneim M A El‐Shafai W Abdel‐Salam N et al. 2021 A survey of traditional and advanced automatic modulation classification techniques, challenges, and some novel trends International Journal of Communication Systems 34(10): e4762

[2] Finlayson S G Chung H W Kohane I S et al. 2018 Adversarial attacks against medical deep learning systems arXiv preprint arXiv:1804.05296

[3] Khakurel U 2022 Adversarial Machine Learning Using Convolutional Neural Network With Imagenet 2022 Annual Modeling and Simulation Conference (ANNSIM) Modeling and Simulation Conference (ANNSIM) pp. 246–257

[4] Radford A Metz L Chintala S 2015 Unsupervised representation learning with deep convolutional generative adversarial networks arXiv preprint arXiv:1511.06434

[5] Goodfellow Ian J Jonathon S and Christian S 2014 Explaining and harnessing adversarial examples arXiv preprint arXiv:1412.6572

[6] Zhang J et al. 2022 NMI-FGSM-Tri: An Efficient and Targeted Method for Generating Adversarial Examples for Speaker Recognition 2022 7th IEEE International Conference on Data Science in Cyberspace (DSC) Data Science in Cyberspace (DSC) 2022 7th IEEE International Conference on DSC pp. 167–174

[7] Tang H et al. 2023 Local and Global GANs With Semantic-Aware Upsampling for Image Generation IEEE Transactions on Pattern Analysis and Machine Intelligence, Pattern Analysis and Machine Intelligence, IEEE Transactions on, IEEE Trans Pattern Anal Mach. Intell 45(1) pp. 768–784

[8] Laykaviriyakul P and Phaisangittisagul E 2023 Collaborative Defense-GAN for protecting adversarial attacks on classification system Expert Systems With Applications 214

[9] Hoang H D et al. 2022 DA-GAN: Domain Adaptation for Generative Adversarial Networks-assisted Cyber Threat Detection 2022 RIVF International Conference on Computing and Communication Technologies (RIVF), Computing and Communication Technologies (RIVF), 2022 RIVF International Conference on pp. 29–34

[10] Samangouei P Kabkab M Chellappa R 2018 Defense-gan: Protecting classifiers against adversarial attacks using generative models arXiv preprint arXiv:1805.06605