Michał Zimoń*
Rafał Kasprzyk**

# Yet another research on GANs in cybersecurity

**Abstract**

Deep learning algorithms have achieved remarkable results in a wide range of tasks, including image classification, language translation, speech recognition, and cybersecurity. These algorithms can learn complex patterns and relationships from large amounts of data, making them highly effective for many applications. However, it is important to recognize that models built using deep learning are not fool proof and can be fooled by carefully crafted input samples. This paper presents the results of a study to explore the use of Generative Adversarial Networks (GANs) in cyber security. The results obtained confirm that GANs enable the generation of synthetic malware samples that can be used to mislead a classification model.

Key words: cybersecurity, malware, artificial intelligence, machine learning, deep learning, generative adversarial networks

* Ppor. mgr inż. Michał Zimoń, Faculty of Cybernetics, Military University of Technology, Warsaw, e-mail: michal.zimon@wat.edu.pl.
** Płk dr inż. Rafał Kasprzyk, Faculty of Cybernetics, Military University of Technology, Warsaw, e-mail: rafal.kasprzyk@wat.edu.pl.

## Introduction

Malware, or malicious software, is a major threat to cybersecurity. It can take many forms, including viruses, worms, and Trojan horses, and it can be used to steal sensitive information, disrupt systems, and spread to other devices. Traditional methods for detecting malware, such as signature-based detection and behavioural analysis, have their limitations and can be defeated by cleverly designed malware. As a result, researchers have explored alternative approaches, including the use of deep learning for detecting malware based on images of the code itself.

Deep learning is a subset of machine learning that uses artificial neural networks to automatically learn patterns and features in large datasets. It has been applied to various fields, including computer vision, natural language processing, and cybersecurity. In the context of malware detection, deep learning algorithms can analyse large amounts of data and identify patterns that are indicative of malware. These patterns may be present in the static characteristics of the malware, such as the code itself or the metadata associated with it.

Generating image representations of malware can potentially pose several dangers. One concern is that generating images of malware could facilitate the spread of malicious software. For example, if an image representation of malware is shared online, it could potentially be downloaded and executed by someone who is unaware of its true nature. This could lead to the unintentional installation of malware on a person's computer, potentially causing harm to the user and their data.

Another danger of generating image representations of malware is that it could potentially make it easier for malicious actors to bypass security measures. For example, if an image representation of malware is used as part of a phishing attack, it could potentially be more difficult for security systems to detect the malware, as it is not in its typical form. This could make it easier for attackers to successfully carry out their attacks and compromise the security of a system.

It is also important to note that generating image representations of malware could potentially raise legal and ethical concerns. Depending on the specific context, generating, and distributing image representations of malware could potentially be considered illegal or unethical, as it could facilitate the spread of harmful software.

## Methodology

In our work, we want to check how the most popular classifiers trained on a dataset, containing a graphical representation of malware and benign software, will behave when they encounter those generated by GANs. We also used publicly available models to check how complex the process is.

### Dataset

Both classifiers and generative networks were trained using the MaleVis[1] dataset. MaleVis is an open image dataset generated from 25 malware and 1 legitimate software class. Includes a total of 9,100 training and 5,126 validation RGB images at 224x224 or 300x300 resolution.

### EfficientNet-B0

One of the networks used in our work for classification is Google's EfficientNet-B0[2]. It is part of the EfficientNet family of models that are designed to be highly efficient and work well across a variety of tasks and platforms. EfficientNet-B0 is a large model that has been trained on a dataset of millions of images and is intended for use in large-scale image classification tasks. It is characterized by high accuracy and performance, making it ideal for applications where computing resources are limited or where real-time performance is important.

EfficientNet-B0 is based on the MobileNetV2 architecture and uses a combination of depth wise separable convolutions and regular convolutions to build its model. It also includes several other techniques such as weight sharing and a composite scaling method to further improve performance and efficiency.

EfficientNet-B0 achieved top results in several image classification benchmarks, so it was further developed, and many subsequent versions were created. It is widely used in a variety of applications including object detection,

**1** A.S. Bozkir, A.O. Cankaya, M. Aydos, *Utilization and Comparision of Convolutional Neural Networks in Malware Recognition*, 2019, https://www.researchgate.net/publication/331773587_Utilization_and_Comparision_of_Convolutional_Neural_Networks_in_Malware_Recognition [access: 4.01.2023].

**2** M. Tan, Q. Le, *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, 2019, https://arxiv.org/pdf/1905.11946.pdf [access: 4.01.2023].

image segmentation, and machine translation. It is also available as part of the TensorFlow-Slim library, making it easy to use and deploy in a variety of environments.

### ResNet50

Another network used is ResNet50[3], developed by Microsoft Research. It is part of the ResNet family of models that are known for their deep architecture and ability to learn from large amounts of data efficiently. ResNet50 is a deep CNN that has been trained on a dataset of millions of images and is intended for use in large-scale image classification tasks. It has high accuracy and high performance in various benchmarks, making it a popular choice for a wide range of applications.

ResNet50 is built using a residual learning framework, which involves the use of shortcut connections that allow the network to learn residuals of the desired underlying mapping. This helps to alleviate the problem of vanishing gradients and enables the network to train deeper architectures without the performance degradation that often occurs with deeper networks. It also achieved state-of-the-art results on several tasks and usually competes with EfficientNet.

### DCGAN

First network used to generate the samples is the Deep Convolutional Generative Adversarial Network[4]. It is a type of generative adversarial network (GAN) used for generating synthetic images. GANs consist of two neural networks: a generator and a discriminator. The generator is trained to produce synthetic samples that are indistinguishable from real ones, while the discriminator is trained to distinguish between real and synthetic samples. The two networks are trained in a zero-sum game, where the generator tries to fool the discriminator and the discriminator tries to correctly classify the samples as real or synthetic.

**3** K. He et al., *Deep Residual Learning for Image Recognition*, 2015, https://arxiv.org/pdf/1512.03385.pdf [access: 4.01.2023].

**4** A. Radford, L. Metz, S. Chintala, *Unsupervised Represenation Learning With Deep Convolutional Generative Aadversarial Networks*, 2016, https://arxiv.org/pdf/1511.06434.pdf [access: 4.01.2023].

DCGAN is a variant of GAN that uses convolutional neural network (CNN) architectures for both the generator and discriminator networks. This allows DCGAN to effectively capture the spatial dependencies and patterns in image data, making it well-suited for generating synthetic images.

DCGAN has been used to generate a wide range of synthetic images, including realistic images of faces, objects, and landscapes. It has also been used in a variety of applications, such as image synthesis, style transfer, and data augmentation. However, it is important to note that the synthetic images produced by DCGAN may not be representative of real-world images and should be used with caution.

### StyleGAN2-ADA

Second network used to generate the samples is a variant of the StyleGAN2 generative adversarial network (GAN) architecture developed by NVIDIA. It is a state-of-the-art model for generating high-resolution, synthetic images and has been used to generate a wide range of images, including realistic images of faces, objects, and landscapes.

StyleGAN2-ADA is an extension of the original StyleGAN2 model that includes additional modifications and improvements, such as the use of adaptive discriminator augmentation (ADA) and a new truncation trick. These modifications allow StyleGAN2-ADA to generate even higher-quality images than the original StyleGAN2 model, while also improving the stability and efficiency of the training process.

It is a deep neural network that is trained using a large dataset of images. It can generate synthetic images by learning the underlying patterns and relationships in the training data. The generated images are highly realistic and are often difficult to distinguish from real ones.

StyleGAN2-ADA has been used in a variety of applications, including image synthesis, style transfer, and data augmentation. It is also widely used for research purposes, as it allows researchers to investigate the capabilities and limitations of GANs and other deep learning models.

# Experiments

## Classification

Last layers of both networks to classify the images were retrained using modified MaleVis dataset. To keep experiments quick, we first trained the networks for 5 epochs using Adam optimizer with learning_rate = 0.001. Then we recompiled the model with lower learning rate which was 0 and fine-tuned for 5 more epochs.

### Dataset setting

For the classification between malware and benign software, 25 classes were considered as 1 class of malware. As a result, we got a dataset containing 1832 images representing normal software and 12394 images representing malware. Due to the disproportion of data, we decreased the number of malware samples. As a result, we used 1465 samples of benign software for training and 367 samples for validation and 1000 samples of malware for training, and 299 samples for validation. In addition, the images have been scaled to a resolution of 128 x 128 for faster training.

### Performance


Source: own elaboration

Figure 1. EfficientNet-B0 training results


Source: own elaboration

Figure 2. ResNet50 training results

Table 1. Final performance of both networks

| | Training | | Validation | |
|---|---|---|---|---|
| | Loss | Accuracy | Loss | Accuracy |
| EfficientNet-B0 | 0.1833 | 0.9339 | 0.1150 | 0.9896 |
| ResNet50 | 0. 0771 | 0.9813 | 0.1354 | 0.9583 |

## GANs

### DCGAN

Small amount of data made it hard to get satisfying results. We used DCGAN to generate benign samples based on dataset containing 1465 samples of benign software and 1000 samples of malware. Training process for benign samples was as follows:

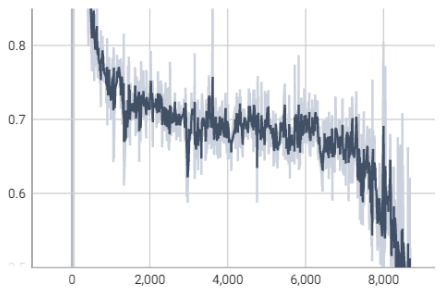


Figure 3. Discriminator loss (vertical) and steps (horizontal)

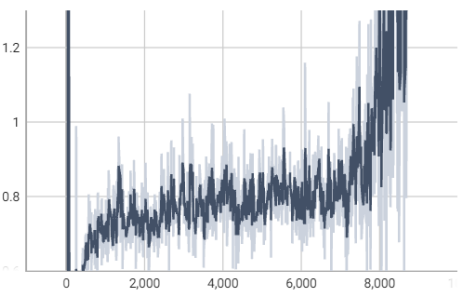


Figure 4. Generator loss (vertical) and steps (horizontal)

Samples from the trained generator:

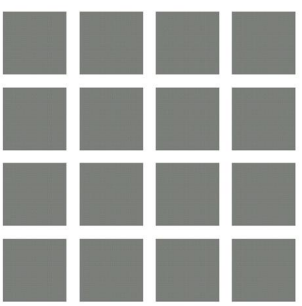

Figure 5. Epoch 0

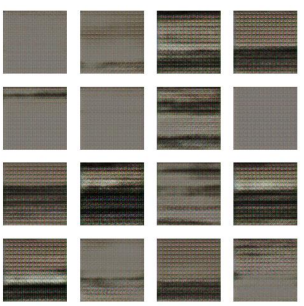

Figure 6. Epoch 100

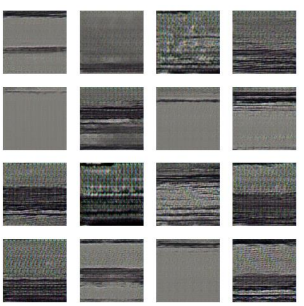

Figure 7. Epoch 290

Training on malware samples:

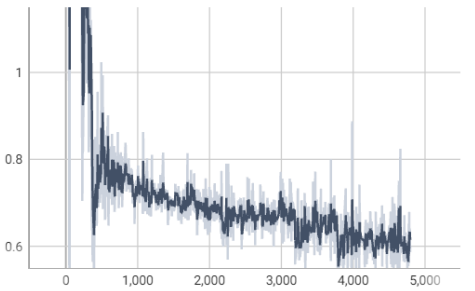


Figure 8. Discriminator loss (vertical) and steps (horizontal)

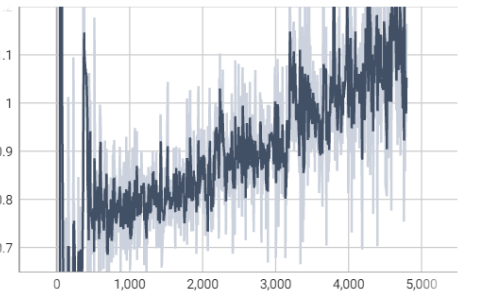


Figure 9. Generator loss (vertical) and steps (horizontal)

Generated samples:

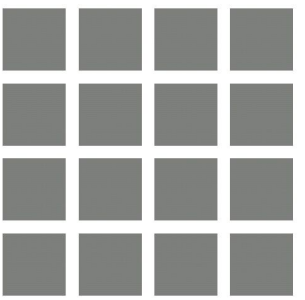

Figure 10. Epoch 0

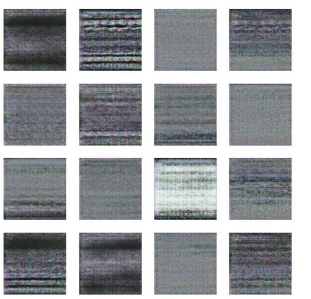

Figure 11. Epoch 100

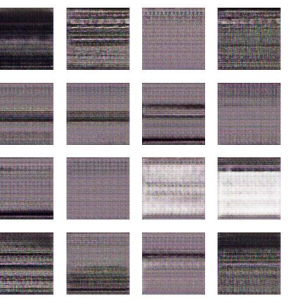

Figure 12. Epoch 290

Because dataset contains huge amount of malware samples, we also trained same network with dataset containing 12394 malware samples with batch_size = 64 and batch_size = 128. We got results as follows:

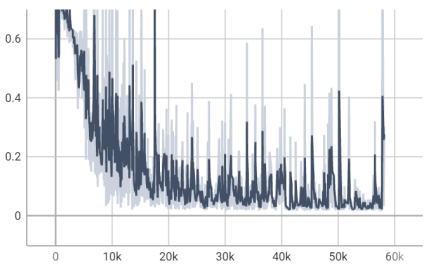


Figure 13. Discriminator loss (vertical) and steps (horizontal) for batch_size = 64

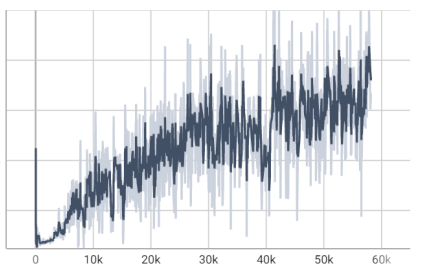


Figure 14. Generator loss (vertical) and steps (horizontal) for batch_size = 64

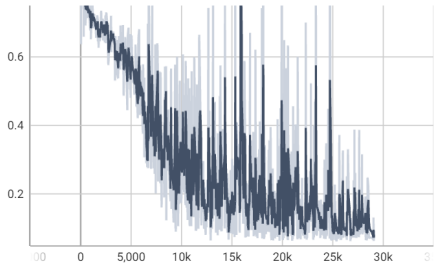


Figure 15. Discriminator loss (vertical) and steps (horizontal) for batch_size = 128

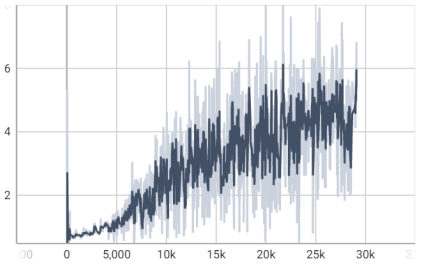


Figure 16. Generator loss (vertical) and steps (horizontal) for batch_size = 128

Generated samples:

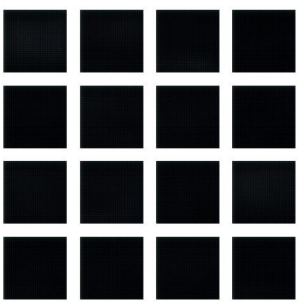

Figure 17. Epoch 0, batch_size 64

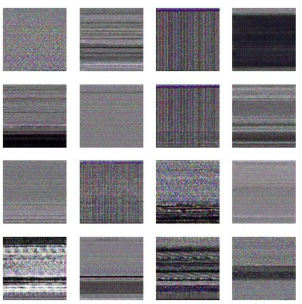

Figure 18. Epoch 100, batch_size 64

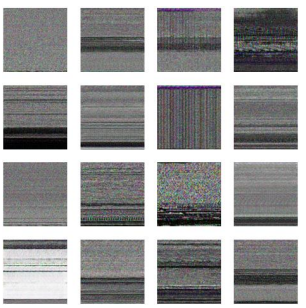

Figure 19. Epoch 290, batch_size 64

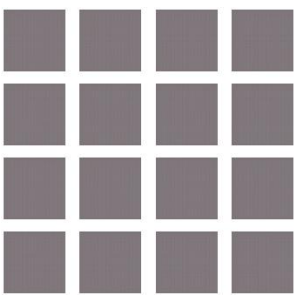

Figure 20. Epoch 0, batch_size 128

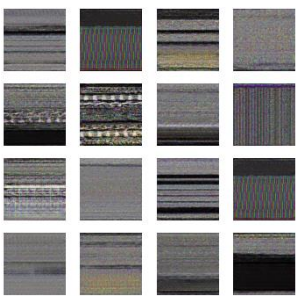

Figure 21. Epoch 100, batch_size 128

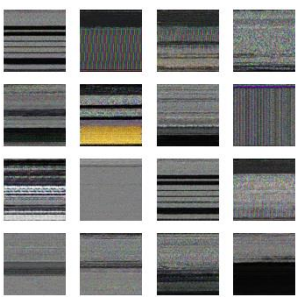

Figure 22. Epoch 290, batch_size 128

## StyleGAN2-ADA

Authors of StyleGAN2-ADA proposed an adaptive discriminator augmentation mechanism that significantly stabilizes training process in limited data regimes. Because we had only 1832 samples of benign software and 12 394 of malware, we trained this network only to generate malware samples. Results of training process are as follows:


Figure 23. StyleGAN2-ADA training

After few days of training, we haven't seen any improvement. The best FID we achieved was 37.16 after epoch 19400. Generated samples during the training process:

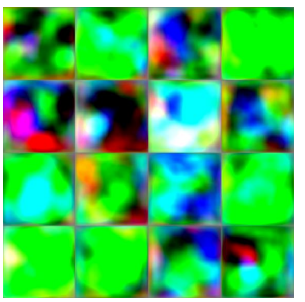

Figure 24. Epoch 0.

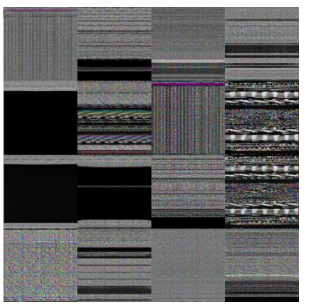

Figure 25. Epoch 19 400.

Figure 26. Epoch 25 000.

## Results

After training generators from the different networks, we used them to generate 256 samples each of benign software and malware. First, we provided generated samples by DCGAN from reduced dataset as the input of previously trained networks for image classification. Results are shown in table below.

Table 2. Classification of generated samples for reduced dataset

| | EfficientNet-B0 | | ResNet50 | |
|---|---|---|---|---|
| | goodware | malware | goodware | malware |
| Goodware | 256 | 0 | 256 | 0 |
| Malware | 208 | 48 | 240 | 16 |

Most of generated malware samples were misclassified as benign software. Situation looks better, when GANs were trained on 12 394 malware samples.

Table 3. Classification of generated samples for 12 394 malware samples

| | | Batch size | EfficientNet-B0 | | ResNet50 | |
|---|---|---|---|---|---|---|
| | | | goodware | malware | goodware | malware |
| Malware | DCGAN | 64 | 94 | 162 | 58 | 198 |
| | DCGAN | 128 | 105 | 151 | 76 | 180 |
| | StyleGAN2-ADA | 32 | 46 | 210 | 42 | 214 |

## Conclusion

In this paper we showed how networks trained for image classification reacts to artificial samples which were generated. Accuracy during the training process and validation was high so such network could be used for malware detection. Surprisingly, we noticed problem with generating malware samples. Both EfficientNet-B0 and ResNet50 misclassified most of malware samples as benign software. Results were better with samples from StyleGAN2-ADA but still not perfect.

Too small dataset can be the cause of such behaviour. As mentioned in[5] it typically takes 50 000 to 100 000 training images to train a high-quality GAN. After all, malware is still software but designed to cause disruption to a target. Not enough samples may cause the situation where GAN simply is not able to learn the proper difference between malware and benign software. This may lead to generate a sample which will have software features but not the „malicious" part, hence it may be classified as benign software. Further work could be redoing the experiments on bigger dataset.

5 I. Salian, *NVIDIA Research Achieves AI Training Breakthrough*, 2020, https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/ [access: 4.01.2023].

On the other hand, GAN which will generate malware samples that can fool networks such as EfficientNet-B0 or ResNet50 and be classified as malware may rise several problems. Firstly, it will make it easier for malicious actors to attack security systems. Secondly, for both the malware and benign software it raises legal and ethical concerns. After all it is generated, not real sample, so should it be even considered as software?

### Bibliography

Bozkir A.S., Cankaya, A.O., Aydos M., *Utilization and Comparision of Convolutional Neural Networks in Malware Recognition*, 2019, https://www.researchgate.net/publication/331773587_Utilization_and_Comparision_of_Convolutional_Neural_Networks_in_Malware_Recognition [access: 4.01.2023].

He K., Zhang X., Ren S., Sun J., *Deep Residual Learning for Image Recognition*, 2015, https://arxiv.org/pdf/1512.03385.pdf [access: 4.01.2023].

Karras T. et al., *Training Generative Adversarial Networks with Limited Data*, 2020, https://arxiv.org/pdf/2006.06676.pdf [access: 4.01.2023].

Radford A., Metz L., Chintala S., *Unsupervised Represenation Learning With Deep Convolutional Generative Aadversarial Networks*, 2016, https://arxiv.org/pdf/1511.06434.pdf [access: 4.01.2023].

Salian I., *NVIDIA Research Achieves AI Training Breakthrough*, 2020, https://blogs.nvidia.com/blog/2020/12/07/neurips-research-limited-data-gan/ [access: 4.01.2023].

Tan M., Le Q., *EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*, 2019, https://arxiv.org/pdf/1905.11946.pdf [access: 4.01.2023].

## Jeszcze jedna praca badawcza dotyczące sieci GAN w cyberbezpieczeństwie

### Streszczenie

Algorytmy głębokiego uczenia pozwoliły osiągnąć znakomite wyniki w różnorodnych zadaniach, w tym w klasyfikacji obrazów, tłumaczeniu języka, rozpoznawaniu mowy i cyberbezpieczeństwie. Mogą one uczyć się złożonych wzorców i zależności z dużych ilości danych, dlatego są bardzo skuteczne w wielu zastosowaniach. Jednakże ważne jest to, żeby zdawać sobie sprawę, że modele zbudowane z wykorzystaniem uczenia głębokiego nie są niezawodne i można je oszukać za pomocą starannie przygotowanych próbek wejściowych. W artykule zostały przedstawione wyniki badań, których celem jest zbadanie możliwości wykorzystania generatywnych sieci antagonistycznych (ang. Generative Adversarial Networks – GAN) w cyberbezpieczeństwie. Uzyskane wyniki potwierdzają, że sieci GAN umożliwiają generowanie syntetycznych próbek złośliwego oprogramowania, które mogą zostać wykorzystane do wprowadzenia w błąd model klasyfikacyjny.

Słowa kluczowe: cyberbezpieczeństwo, złośliwe oprogramowanie, sztuczna inteligencja, uczenia maszynowe, głębokie uczenie, generatywne sieci antagonistyczne