IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 28 July 2023, accepted 17 September 2023, date of publication 26 September 2023, date of current version 3 October 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3319436

RESEARCH ARTICLE

Malware Image Generation and Detection Method Using DCGANs and Transfer Learning

NIKOLAOS PEPPES[1], THEODOROS ALEXAKIS[1], EMMANOUIL DASKALAKIS[1], KONSTANTINOS DEMESTICHAS[2], AND EVGENIA ADAMOPOULOU[1]

[1]School of Electrical and Computer Engineering, National Technical University of Athens, 15773 Zografou, Greece
[2]Department of Agricultural Economics and Rural Development, Agricultural University of Athens, 11855 Athens, Greece

Corresponding author: Nikolaos Peppes (npeppes@cn.ntua.gr)

This work was supported by the European Union under Grant 101073951.

ABSTRACT Cybersecurity in modern age is of utmost importance in almost every domain of economic activity. As digital activities make heavy use of multimedia a new type of cyber-threat gradually emerges: the possibility of producing and seamlessly embedding malware into digital images. Such type of malware can potentially avoid detection of typical scanners and infect the systems of either the service providers and the end-users. In this context, this study proposes and describes a complete methodology starting from the process of generation of malware-based yet realistic to the human eye images and concluding to the design of a suitable malware detector. This methodology designs and employs Deep Convolutional Generative Adversarial Networks (DCGANs) to synthetically generate two new large datasets of images: one with suspicious malware images (called Expanded Malware Images – EMI, in this study) and one with adversarial sample images of fashion products (called Fashion Adversarial Samples – FAS, in this study). The two new datasets are used for training two different Convolutional Neural Network (CNN) models using different training and configuration approaches. The first CNN (named c-CCN) follows a conventional approach for training, whereas the second one (named TL-CCN) leverages transfer learning to take advantage of the knowledge of ResNet50. Results show that the generation of malware images and adversarial samples stabilizes after 3000 iterations and produces very realistically looking images. Moreover, the TL-CNN model trained with part of the adversarial samples outperforms the other malware detector designs and produces results of high validation accuracy and minimal validation loss.

INDEX TERMS Malware generation, generative adversarial networks (GANs), transfer learning, convolutional neural network (CNN), cybersecurity.

I. INTRODUCTION

The continuous evolution of Information and Communication Technologies (ICT) as well as the digitization of almost every aspect of everyday life have transformed the humans' habits. Moreover, the Covid-19 pandemic boosted the online activities due to the imposed isolation measures. For example, according to a European e-commerce report, the percentage of e-shoppers in Europe grew from 60% in 2017 to 73% for 2021, whilst the same rate for the European Union of the 27 members increased from 63% in 2017 to 75% in 2021 [1]. This increase also created a breeding ground for cyber frauds and cyber threats. According to the Juniper Research report [2], it is forecasted that by 2024 there will be a 50% increase in financial loss caused by online payment fraud. This means that especially e-commerce merchants will lose more than $25 billion annually, compared to $17 billion in 2020 regardless the fact that a plethora of companies in Europe are already implementing biometrics and other methods of Secure Customer Authentication (SCA).

Nowadays, attackers are engaging numerous types of cyber threats and attacks and evolve their techniques in order to tackle even the most sophisticated cybersecurity systems. According to the European Payment Council (EPC), six main threat and fraud enablers categories can be identified: i) Social Engineering; ii) Malware; iii) Advanced Persistent

The associate editor coordinating the review of this manuscript and approving it for publication was Jiachen Yang.

Threats (APT); iv) Denial of Service (DoS); v) Botnets, and vi) Monetization channels [3]. These threats are currently thriving in the e-commerce domain with the number of fraud attempts and attacks raising every day [4]. From these categories of threats, malware (defined by ENISA, the European Union Agency for Cybersecurity, as a piece of software that executes harmful operations in order to achieve data theft or any other type of compromission to computer devices [5]) can be met in various types, such as trojans, viruses, worms, spyware, ransomware, etc. Despite the fact that malware is, nowadays, a common threat, its adaptability and the evolution of new types keeps the number of malware growing constantly increasing through the years.

Given the ever-expanding malware techniques, conventional methods for malware detection are often inadequate [6]. Thus, advanced methods leveraging Machine Learning (ML) and Deep Learning (DL) are, nowadays, more and more employed in the fight against malware. This paper focuses on the fight against malware embedded in images, which constitutes an emerging trend in malware threats [7]. The paper designs, develops and evaluates a novel two-step approach. First, Deep Convolutional Generative Adversarial Networks (DCGANs) techniques are developed in order to produce adversarial image samples based on malware images that mimic the original ones. This enables the expansion of the (otherwise limited) available (training and validation) datasets and consequently the development of a deep convolutional neural network model using transfer learning that can detect such suspicious images. Thus, the methodology presented in this paper consists of two main parts: i) the generation of malware-based suspicious images using DCGANs, and ii) the detection of suspicious images using deep convolutional neural networks techniques in conjunction with transfer learning techniques, in order to effectively address malware detection issues (i.e., limiting the chances of malware in bypassing malware detection systems).

The remainder of the paper is organized as follows: Section II presents related works, focusing on the domain of malware generation and detection. Section III describes the methodology designed and developed, whilst Section IV elaborates on the datasets produced and used. Section V presents the produced results, and Section VI analyses and discusses them. Finally, Section VII concludes the paper.

II. RELATED WORKS

The growth of e-commerce, where numerous pictures of products are used and exchanged, provides the opportunity for image malware to thrive since malware binaries can be converted into grayscale or RGB image files, as demonstrated by Conti et al. [8]. Furthermore, many malware types when embedded into images preserve the structure of the image and do not alter it in an easily noticeable manner. Thus, they are hard to be detected even by well-tuned detection systems. In parallel, malware developers have the advantage of knowing the existing anti-malware tools and can thus test their code, so that they can be certain about its effectiveness.

In 2014, Ian Goodfellow et al. [9], proposed a framework called the Generative Adversarial Network (GAN). GANs have the ability to generate images from random noise. Also, they consist an alternative technique for developing generative models and architectures. More specifically, a GAN can be defined as a game between two competitors, the generator and the discriminator. The competition between generator and the discriminator is a minimax optimization problem which is terminated when the generator's strategy reaches a minimum and the discriminator's reaches a maximum [9], [10]. GANs have proved very useful in applications that require synthetic data. The functionality of a GAN includes two different neural networks which participate in a competitive process, namely the discriminator and the generator. Radford et al. [11] presented a solution called Deep Convolutional Generative Adversarial Network (DCGAN), emphasizing the development of deeper layer architectures rather than focusing on the classic architecture of the original GANs.

On the other hand, malware developers use many techniques to bypass machine-learning based malware detectors. As an example, an approach developed by Nataraj et al. [12] has aimed at building malware classifiers by training them with grayscale image vectors. The researchers illustrated that, by changing a few bytes, a model can classify a malware as a goodware [12]. This technique can be employed by attackers to bypass malware classifiers, by properly changing a few bytes and pixels. Another technique employed by malware developers to trick neural networks is through adversarial samples. Adversarial samples are used as inputs to the neural network, to influence the learning outcome. A research study, conducted by Goodfellow et al. [13] showed that a small amount of carefully constructed noise can fool a neural network into believing that the entered image is an image of a gibbon and not a panda, with 99.3% confidence. The neural network originally thought that the provided image was a panda, with 57.7% confidence [13].

The study of adversarial machine learning techniques constitutes an important trend among information security professionals and deep learning engineers, with a view to developing more robust malware detection systems and security solutions [14]. In this light, certain works aim to mimic the malware generation procedure, so as to gain insights on how new malware types will behave. Singh et al. [15] introduced the MIGAN framework, which stands for Malware Images GAN. The MIGAN aims to create labelled malware images by using GANs. These images can, then, be used as input to better train ML or DL malware classifiers, so as to increase their efficiency. In the same direction, Jang et al. [16] used global and local images of unobfuscated malware that were generated using pixel and local feature visualizers. Similarly to the MIGAN, the GAN in this study was used to generate local images of obfuscated malware by learning from global and local images of unobfuscated malware. Subsequently, the local image of unobfuscated malware could be merged with the generated images in order to create a

dataset usable for training in malware classifiers. Moreover, Xiao et al. [17] used AdvGAN to create malware attacks based on MNIST, CIFAR-10 and ImageNet-compatible datasets. The results of this study indicated that AdvGAN can produce high quality attacks that can bypass state-of-the-art defense systems. Considering that malware can have many different forms and can exploit zero-day vulnerabilities of software, Bhaskara and Bhattachayya [18] created a deep GAN which emulates a malware author so as to create new types of malware attacks. This GAN is trained over a reversible RGB image representation of known malware attacks. In addition to malware that is hidden in images, other studies that, also, engage GANs to create malware attacks such as byte-level perturbations in PE files [19], alteration of PE headers [20], [21], API calls [22] and DOS/DDOS attacks [23], can be found.

All the aforementioned efforts contain techniques and methodologies to mimic the authorship and generation of malware. Naturally, the next step is to deploy systems that are capable of detecting malware and minimize the consequences even of zero-day vulnerabilities. To this end, Kim et al. [24] introduced the tDCGAN (transferred Deep Convolutional Generative Adversarial Network), which aims at creating and detecting fake malware images. This study capitalized on the research performed by the same team and further evolved it [25]. The tDCGAN-based method achieved an accuracy of around 96%. Burks et al. [26] proposed GAN and Variational Autoencoder (VAE) methods in order to enhance the performance of a Residual Network (ResNet) classifier. Their experiments indicated that both methods increased the performance of the ResNet classifier. GANs had better results compared to VAEs, but both approaches can assist in mitigating the problem of data inadequacy. Moreover, expanding the idea for generation and detection of malware embedded in images, He and Kim [27] introduced their methodology based on deep learning techniques. More specifically, they engaged CNN and SPP (Spatial Pyramid Pooling) methods in order to convert malware into images and then to evaluate the detection performance. Jian et al. [28] proposed a Deep Neural Network method named SERLA (SEResNet50 + Bi-LSTM + Attention), which was trained using RGB malware images. Their results showed that transforming malware into RGB images that are, then, given to SERLA as input for training provided very promising results in terms of malware detection and classification, with 98.31% accuracy reported on the BIG 2015 dataset. In the same direction, Bijitha and Nath [29] provided a comprehensive guide on possible executable-to-image conversion techniques which can yield various image-derived features that are suitable for detecting distinguishing characteristics of malware images, such as texture. Also, they developed machine learning and deep learning models for classification purposes.

The aforementioned studies are focused on malware embedded in images and present methods either for mimicking malware generation or for malware detection. The proposed methodology in the present study defines the processes to produce adversarial samples based on fashion products images by designing and using a specific GAN architecture named Fashion Malware Generative Adversarial Network – FMGAN. FMGAN is then utilized for the creation of a next-generation malware detector that is highly adaptive to newly generated adversarial samples and methods of malware concealment. This is feasible by defining a continuous training process of the detection model based on the samples that can be supplied by the FMGAN on a regular basis. Thus, the added value of the present study is that it presents in detail the design and development of a solution that involves both the generation of adversarial samples using the FMGAN, as well as the implementation of an effective malware detector, after comparing different CNN architectures and training datasets in order to determine the most efficient configuration. The fact that fashion products consist a huge market for e-commerce gave the motivation for designing a detector specifically trained on such products.

III. PROPOSED METHODOLOGY

A. FASHION MALWARE GENERATIVE ADVERSARIAL NETWORK (FMGAN)

Nowadays, advanced deep learning techniques are applied to generate multivariate synthetic datasets (e.g., images, texts, videos, financial, music composition) with a high degree of similarity to the original ones. Generative adversarial modelling consists an architectural approach that provides and extends the ability of generating new, synthetic datasets from scratch, based on specified input data format and random noise.

This study focuses on GANs, which are composed of two deep neural network models that behave in a competitive manner, namely the generator and the discriminator. The generator is a deep neural network that receives as input a vector of random numbers, indicated as random noise, and its main task is the generation of high quality, realistic data, similar to the content that was provided as input. On the other hand, the discriminator is a feedforward deep neural network that classifies the input samples of data, as original or generated. Consequently, the generated results are exposed as input to the discriminator model, alongside with the initial sample of original images [30].

In the current study, the FMGAN model was implemented on Tensorflow 2.0 [31] and also relied on the usage of the high-level Keras API [32]. The sequential API was used to create a sequence object in which the different layers of the proposed deep neural network, were stacked. The generator part of the FMGAN consists of the input layer that accepts the random generated noise scaled to the desired size, two hidden layers, and the output layer that converts the exposed flat vector into a 28×28 image shape, in order to generate an adversarial sample, as depicted on Figure 1. Additionally, a batch normalization technique is integrated, so as to provide a more stable training procedure.

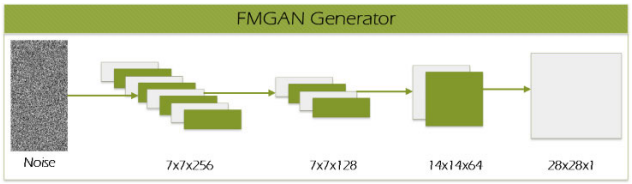

FIGURE 1. FMGAN generator.

The discriminator of the FMGAN is designed as a deep learning classifier trained with integrated supervised learning techniques. Its main task is the classification of the input data (generated/synthetic and original images samples) into original and/or generated. More specifically, a pseudo-probability estimation between 0 and 1 is returned, in order to enhance the model's ability to distinguish the input data sample into real or generated. Then, to ensure the correctness of the prediction, the discriminator loss is computed, to penalize possible misclassification issues of the model. Following the same methodology, as that of the generator development, the discriminator model for the FMGAN was defined. The discriminator network consists of an input layer with a size equal to the input vector received from the original and generated data samples (28×28), two hidden layers, and the output layer with an indicated 'sigmoid' activated function. The discriminator model architecture is depicted in Figure 2.

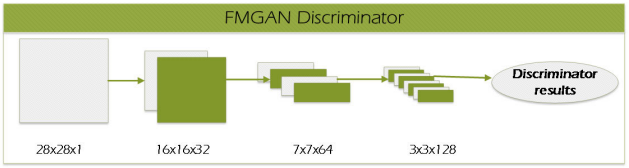

FIGURE 2. FMGAN discriminator.

As a result of the backpropagation procedures, the existence of the discriminator loss improves the discriminator model's exposed predictions and leads to the computation of the generator losses and the gradient. Eventually, the generator continues to improve the synthetically generated data samples, through the continuous updates of the generator weights, using these gradients. After the completion of the FMGAN model training, the result of this competitive procedure leads to a robust generator model, whose generated synthetic data samples are quite difficult to be classified as original or generated by the discriminator. In other words, from the point the developed FMGAN model achieved equilibrium and forward, the generator could easily confuse the discriminator. The final piece of the puzzle, concerning the development of the FMGAN model, was the definition of the training loop between the generator and the discriminator networks, which are actually training separately.

The training procedure was configured for 3000 epochs in total (which yielded better results compared to the use of 1000 and 2000 epochs, while keeping the utilization of computational resources to acceptable levels). For each epoch, a (pre)defined batch training size is performed on both the generator and the discriminator network. As previously described, the discriminator accepts as input a (pre)defined batch of original images from the fashion-MNIST dataset along with the generated output from the generator and computes the discriminator loss for both the real and the generated images. These two loss values are calculated in a separate way and combined following a min-max game based on equation (1) [9] where G is the Generator, D is the discriminator and $V(D,G)$ is the value function of the min-max game.

$$\min_G \max_D V(D, G) = E_{x \sim p_g(x)} \left[\log D(x)\right] + E_{z \sim p_z(z)} \left[\log(1 - D(G(z)))\right] \quad (1)$$

Given the fact that the FMGAN consists of straightforward multilayer perceptrons, equation (1) is applicable. More specifically, the generator's distribution pg over data x can be learned given that the input noise variables $pz(z)$ are firstly defined. Following the noise variables' definition, there is a representation of the mapping to data space as $G(z; \theta g)$, where G is a differentiable function represented by a multilayer perceptron with parameters θg. Then, a second multilayer perceptron $D(x; \theta d)$ is defined which outputs a single scalar as described previously. $D(x)$ represents the probability that x came from the data rather than pg. Finally, the discriminator is trained to maximize the probability of assigning the correct label to both the original and generated images and samples, whilst the generator is trained to minimize $\log(1 - D(G(z)))$.

After the execution of several training epochs, the weights of the generator were significantly updated and the quality of generated images increased, as it is described later in Section IV. This, subsequently, led the model to generate a batch of improved (generated) images and the discriminator network to, then, be fooled by these. Thus, the generator network gradually decreased the weight re-adjustment and started to generate images of high quality, closely resembling the original ones. In the experiments of this study, the original datasets that were used as input to the FMGAN discriminator model were both the fashion-MNIST and the Malimg dataset. Thus, the developed FMGAN was utilized to produce both purely malware images from the Malimg dataset as well as adversarial fashion image samples from the fashion-MNIST dataset. The implementation of two different training procedures helped to expand the size of the dataset supplied for the creation of the malware detector. The datasets used as well as those created using the FMGAN methodology are described in detail in Section IV.

The first FMGAN implementation aimed to generate malware images, similar to the existing ones of the Malimg dataset, in order to expand the initial size (9339 in total) of the dataset with additional malware images. This process is depicted in Figure 3.

The second FMGAN implementation used, as original data input, a combined dataset of fashion-MNIST and Malimg.

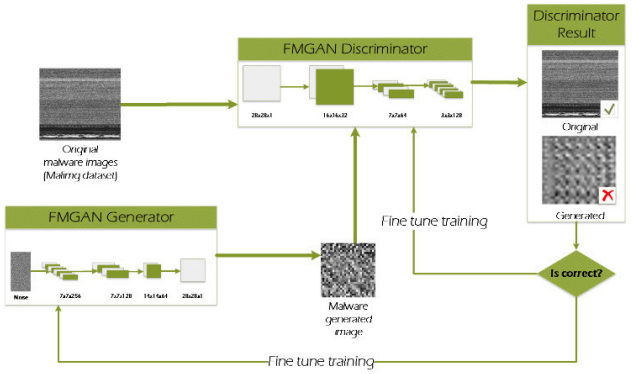

FIGURE 3. Generation process via FMGAN for malware images.

The goal was the generation of original-looking synthetic product images that also integrate features (noise) from the malware images. These generated synthetic images will then be used for the training of a malware detector that can identify real-looking images (as detailed in Section III-B). with embedded extra noise. This process is depicted in Figure 4.

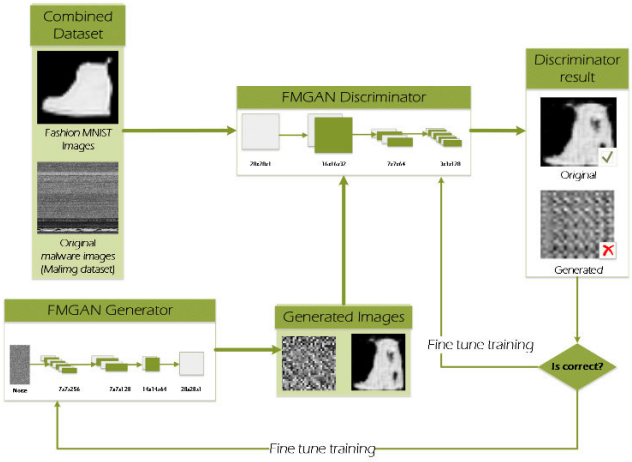

FIGURE 4. Generation process of the adversarial image samples via the FMGAN.

As explained, the focus of this study is to propose and examine a methodology that can both generate and detect adversarial sample images. These images could be used to perform malicious activities by using adversarial machine learning techniques [22], which is a commonly used method to bypass malware detectors. The newly created malware images are employed to train Convolutional Neural Network models for image malware detection, as analyzed in Section III-B.

B. MALWARE DETECTOR IMPLEMENTATION USING CNN AND TRANSFER LEARNING

The present study developed and compared two different Convolutional Neural Network (CNN) architectures: (i) one conventional deep learning architecture, named conventional Convolutional Neural Network (c-CNN) and (ii) a second one integrating transfer learning techniques, named Transfer Learning Convolutional Neural Network (TL-CNN).

Specifically for the latter, the ResNet50 model was engaged for training purposes. ResNet50 is a variation of residual learning models for image recognition presented by He et al. [33] which won the first place at the ILSVRC 2015 classification task. ResNet50 is a pre-trained model for image classification and detection processes that involves 48 convolutional layers, one MaxPool layer and one AveragePool layer. This solution provides a reformulation of the layers as a result of learning about residual functions based on the layer inputs while eliminating unreferenced functions. This practically leads to omitting layers that do not provide anything useful to the learning process. This shortcut of ignoring irrelevant layers can radically speed up the learning process, which, otherwise, consists a major bottleneck of deep neural networks, and can also assist in avoiding the phenomenon of performance saturation or even degradation [33]. Consequently, the performance of residual networks is even better than conventional deep neural networks with the training needs further reduced in specific applications, such as the case examined by He et al concerning the image recognition problem [33]. The results produced in this study [33] showed that, even with 152 layers, a number which is far beyond the depth of Visual Geometry Group (VGG) networks, the complexity was lower while the efficiency was quite high.

Using transfer leaning, the knowledge of this pre-trained model was transferred to the malware detector developed in this study. The results of both the c-CNN model and the TL-CNN model which was trained using the transfer learning process and the ResNet50 knowledge are presented in detail in Section V.

The convolutional base of the utilized sequential CNN model consists of a stack of five (5) layers: the CNN model receives as input tensors of a specified shape (image height, image width and color channels) equal to the actual format of the fashion-MNIST, Malimg and the FMGAN-based generated datasets. The same image format ($28 \times 28 \times 1$) is also used as input to the ResNet50 model. Additionally, there are three (3) convolution blocks with a max pooling layer in each of them and 'Relu' as the selected activation function, whereas a dropout regularization was applied to each of these layers. Eventually, the model development was completed with the integration of a fully connected layer, that used a dense layer to feed the model with the last output tensor, before the data classification, along with a dropout regularization and a batch normalization technique, helping to standardize the inputs on a layer, so as to provide stabilization during the training procedure.

Finally, the adversarial samples, both the original as well as the synthetically generated by the FMGAN, contain similar numbers of malware and ''clean'' images. Thus, given the dataset created and described in the next chapter and examining the validation results of each model trained using this dataset we can conclude whether or not transfer learning is required for malware detection in fashion products images.

Also, for completeness reasons the model which engaged transfer learning was trained using only purely malware and "clean" product images and not adversarial samples generated by the FMGAN. In this way, the evaluation results highlight both the efficiency of transfer learning in this application as well as the important role of the adversarial samples produced by the FMGAN during the training process of the detector.

IV. DATASET EXPLORATION

For the purposes of the present study and following the methodology described in Section III, two datasets were utilized: i) the Malimg dataset [34] and the ii) the fashion-MNIST dataset [35]. As articulated in the following, each one has served a different purpose in the proposed methodology. Also, a new dataset (containing both malware and product images) was generated based on these two datasets using the aforementioned proposed FMGAN architecture.

A. MALIMG AND EXPANDED MALWARE IMAGES (EMI) DATASETS

The Malimg dataset, developed by Nataraj et al. [34], contains 9342 malware images classified into 25 classes. More specifically, the classes, as well as the instances contained in each one, are presented in Table 1.

TABLE 1. Malimg dataset categories.

Category Name	Category	Instances
Allaple.L	Worm	1591
Allaple.A	Worm	2949
Yuner.A	Worm	800
Lolyda.AA 1	PWS	213
Lolyda.AA 2	PWS	184
Lolyda. AA3	PWS	123
C2Lop.P	Trojan	146
C2Lop.gen!G	Trojan	200
Instantaccess	Dialer	431
Swizzor.gen!I	Trojan Downloader	132
Swizzor.gen!E	Trojan Downloader	128
VB.AT	Worm	408
Fakerean	Rogue	381
Alueron.gen!J	Trojan	198
Malex.gen!J	Trojan	136

The Malimg Dataset was used as input to the FMGAN described in Section III, so as to produce new malware images. The total number of synthetic malware images generated by the FMGAN is 48000. Figure 5 presents a sample of the generated malware images per epoch.

The newly generated malware images are similar to the originals, as can be seen in Figure 6, which depicts, side by side, samples from the original dataset and the new synthetic dataset created with the FMGAN. The newly created dataset of the 57343 malware images (9342 original and 48000 generated) is called Expanded Malware Images (EMI) dataset.

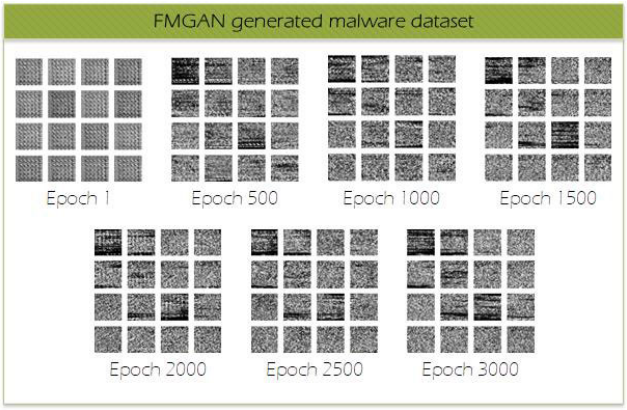

FIGURE 5. Generated malware images using FMGAN.

B. FASHION-MNIST DATASET

Fashion-MNIST [35] is a dataset that contains 70,000 images, 60,000 training images and 10,000 test images. More specifically, fashion-MNIST instances are 28×28 grayscale images, each one being accompanied by one of ten specific labels. Each of the aforementioned labels represents a class, i.e., a fashion product such as a dress. These labels are given in Table 2.

TABLE 2. Malimg dataset categories.

Label	Product
0	T-shirt/Top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

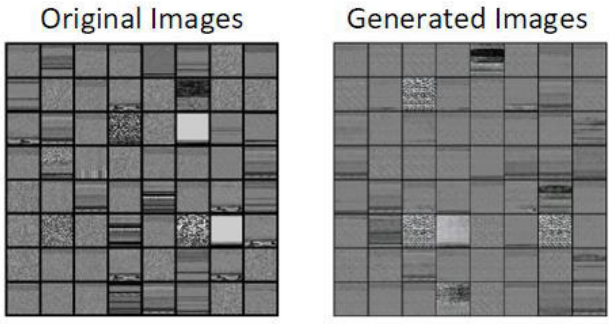

FIGURE 6. Original and FMGAN generated malware images.

The Fashion-MNIST dataset, as the original MNIST dataset [36], is typically used for benchmarking reasons of Machine Learning and Deep Learning classification processes [35], [37], [38]. In this study, this dataset is used to train the proposed malware detector to identify clean product images.

C. ADVERSARIAL SAMPLES DATASET

A third dataset of adversarial samples, named Fashion Adversarial Samples (FAS), was generated, in order to serve as a training dataset for the malware detector presented in Section III. This adversarial dataset was produced using the FMGAN developed in the context of this study and combined both malware images (the EMI dataset) as well as product images from the Fashion MNIST dataset. In total, the FAS Dataset contains 48,000 adversarial image samples that were produced in the 3000 epochs.

These new images are examples of fashion products images that are characterized as suspicious. More specifically, the FMGAN simulates the generation or the authorship of malware that could be embedded in everyday fashion product images which become available through commercial web sites. Used for training (and validation), the purpose of the FAS dataset is to enhance the performance of a malware detector, particularly against adversarial attacks, and to evaluate its results. Figure 7 presents a snapshot of 16 images created in epochs 1, 500, 1000, 1500, 2000, 2500 and 3000.

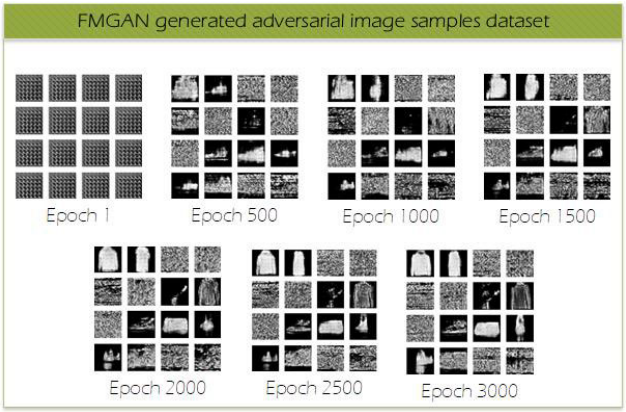

FIGURE 7. Original and FMGAN generated adversarial sample images.

As can be seen in Figure 7, the FMGAN produces more and more accurate images of products as the training proceeds. Thus, in the first epoch, there is practically no clear image either for malware neither for real products, whilst in the last epoch fashion products are clearly identifiable. This indicates that the creation of the adversarial image samples dataset proceeded successfully. This is also demonstrated in Figure 8.

D. SYNOPSIS OF THE EMPLOYED DATASETS

Table 3 and Table 4, depict the exact number of image samples of each dataset engaged and generated for the purposes of this study as well as the samples used for the training and validation purposes of the proposed CNN implementations. More specifically, the initial size of Fashion-MNIST items were 70000, whereas for Malware Images (Malimg dataset) were 9342, respectively. After the definition of the proposed FMGAN, we generated two new samples of 48000 image items, both for the Malware Images (EMI dataset) and the Adversarial Samples (FAS dataset). Table 4 also provides

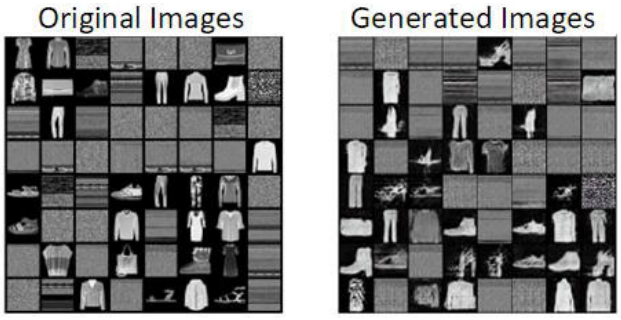

FIGURE 8. Generated adversarial image samples using the FMGAN.

information regarding the total number of image items, for each distinct dataset as well the corresponding size of training and validation datasets.

Figures 5, 6, 7 and 8 presented the results of the dataset generation processes. Such results are important to visually inspect the quality of the generated (image) samples, which can vary across epochs, even after the stabilization of the training process. Figures 5 and 7 visualize a grid of generated images, after a different number of epochs.

In GAN applications, two methods that are used to identify possible failure modes are convergence failure and mode collapse [39], [40]. Convergence failure exists when low quality is noticed in the generated images. Furthermore, if the discriminator is able to generate the same subset of the provided dataset, this is referred to as mode collapse. It is worth noting that, during the training of GANs, failure modes are common and cannot easily be estimated [39]. Specifically, mode collapse is likely to occur as a failure when a huge amount of training data is used, whereas convergence failure is likely in models of high complexity [40]. In the present paper, none of the two failure issues was observed or examined during the training execution of the GAN, as this was out of scope for this specific study.

Generator and discriminator loss graphs are also a way to estimate, in a qualitative way, the stabilization level of the described GAN model. This method is applied in Section V.

V. RESULTS

A. GENERATOR AND DISCRIMINATOR LOSSES IN THE GENERATION OF THE EMI AND FAS DATASETS

The proposed methodology is comprised of two pillars: a) a dedicated DCGAN architecture (FMGAN) for malware and malware-based images generation in order to expand the available datasets, and b) a transfer learning technique for building a predictive model for image-based malware detection, as described in detail in Section III. The software code was developed and executed in Tensorflow 2.0 [31] and the high-level Keras API [32]. Figure 9 and Figure 10 contain the loss graphs for the generator and the discriminator, during the generation of malware and adversarial samples, respectively as they calculated based on the equations (2) and (3)

TABLE 3. Samples per dataset used in the study.

	Fashion-MNIST	Malimg	EMI	FAS
Original Samples	70000	9342	9342	18751
FMGAN generated samples	-	-	48000	48000
Total samples	70000	9342	57342	66751

TABLE 4. Samples used for the training and validation of the different CNN-based detectors under study.

	c-CNN	TL-CNN without adversarial samples	TL-CNN training with adversarial samples
Training			
Fashion MNIST	39871	39871	39871
EMI	-	40352	-
FAS	40352	-	40352
Validation			
Fashion MNIST	20129	20129	20129
FAS	16988	16988	16988
Total samples for training and validation	117340	117340	117340

respectively [9].

$$\min_G V(G) = \nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log(1 - D\left(G\left(z^{(i)}\right)\right)) \quad (2)$$

$$\max_D V(D) = \nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[logD(x^{(i)} + \log(1 - D\left(G\left(z^{(i)}\right)\right))\right] \quad (3)$$

More specifically, in Figure 9, it becomes apparent that the losses are saturating after a certain point of the executed iteration, during the generation process of malware samples, both for the generator and the discriminator. More specifically, the generator loss started quite high, close to 25, whilst the discriminator loss was considerably lower, approximately 5. This was an expected result, as in the beginning of the training process, the generator could not produce synthetic images similar to the real ones. Therefore, the discriminator could classify them correctly quite easily. As the training process progressed, the generator loss started to reduce, taking values between 2 and 15, spanning the iterations 10 to 3000, and between 2 and 8 for the iterations from 3000 to 7000. During the same iteration intervals, the discriminator loss increased, since it was no longer possible to correctly classify the synthetic images with high probability. This procedure is referred to the EMI dataset generation as described in Section IV.

Similarly, Figure 10 depicts the generator and discriminator losses during the generation of the adversarial samples (FAS dataset). The generator loss started again at a quite high level, above 25, whilst the discriminator loss was considerably lower, around 3. As previously, this was an expected result, since in the beginning of the training process, the generator could not produce synthetic images similar to the real ones, and the discriminator could classify them correctly without difficulty. As the training process progressed, the generator loss started to reduce, taking values between 2 to 7. The spikes that are observed in the generator loss (above 10) were due to the heterogeneous nature of the input dataset.

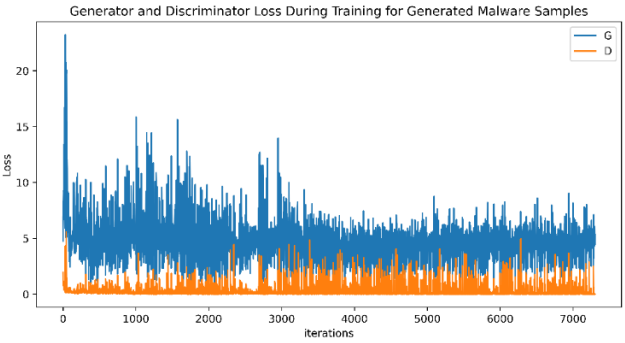

FIGURE 9. Generator (G) and discriminator (D) losses for malware samples generation (EMI dataset).

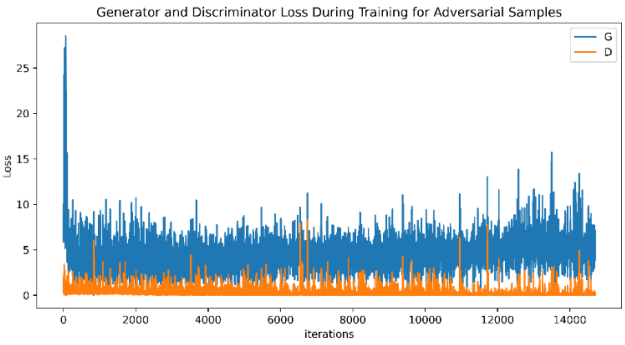

FIGURE 10. Generator and discriminator losses for adversarial samples generation (FAS dataset).

During the training progress, the discriminator loss increased, as it was no longer able to correctly classify the synthetic images with high probability.

B. ACCURACY AND LOSS RESULTS OF THE MALWARE DETECTORS

Three different malware detectors are developed in the present study. The results presented herein include the loss

and accuracy metrics as well as the fluctuations per iteration during the models' training processes. All models are fed with data samples (greyscale images) from the different datasets, as described previously in Section IV-D, Table 4 with their pixel values ranging from 0 to 255. Using appropriate data preprocessing, the CNN models are fed with $28 \times 28 \times 1$ dimension images.

The first model is a conventional CNN (c-CNN) that was fed and trained with the Fashion-MNIST and the FAS dataset (Table 4). The second one employs transfer learning based on the residual, pre-trained ResNet50 model and was fed and trained with the Fashion-MNIST and the EMI datasets (Table 4). Finally, a third model, which also employs transfer learning, was trained using the Fashion-MNIST and the FAS datasets, as presented in Table 4. It is worth mentioning that these three models used as validation datasets, subsets from the Fashion-MNIST and the FAS datasets (as described in Table 4), in order to compute the accuracy and loss metrics and estimate the efficiency of each of one, respectively, using as input unknown data. The main reason for selecting and using the ResNet50 model is to fully utilize the specific base of knowledge and assess whether the prediction outcomes are improved compared to a conventional CNN. Additionally, another comparison between the two transfer-learning models (fed with different training datasets, as described in Table 4) is worthwhile, since one of the two is trained with adversarial samples while the other one is not.

The results of the conventional CNN model, in terms of accuracy and loss, are depicted in Figure 11 and Figure 12, respectively.

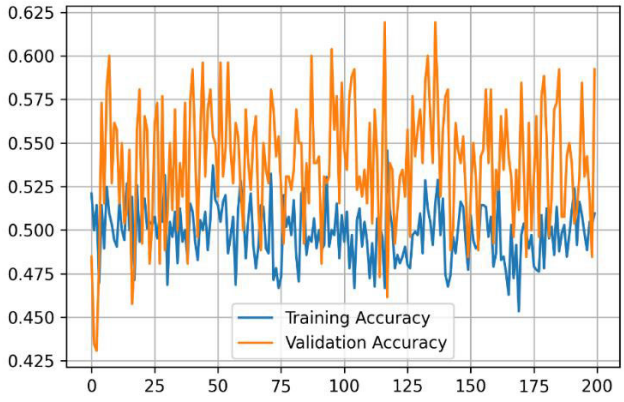

FIGURE 11. Conventional CNN (c-CNN) model training and validation accuracy.

Figure 13 to Figure 16 depict the results, in terms of accuracy and loss, respectively, for the two CNN models which utilize transfer learning. The first one is trained with the Fashion-MNIST and EMI datasets and the second one with Fashion-MNIST and the FAS dataset. The main difference between the two is in the training process, as the first one is trained only with "clean" fashion product and purely malware images, while the second one is trained with "clean" fashion product images, purely malware images as well as adversarial samples of fashion products. The validation of

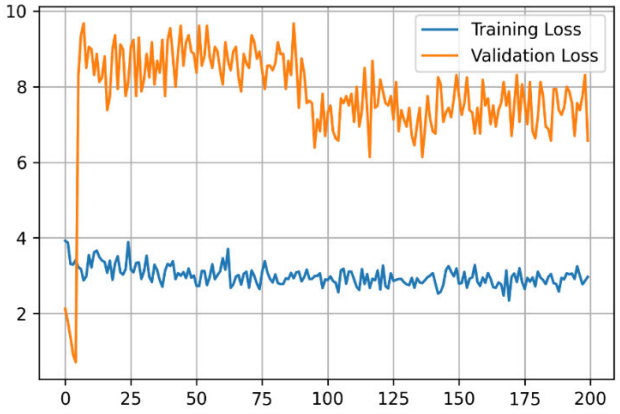

FIGURE 12. Conventional CNN (c-CNN) model training and validation loss.

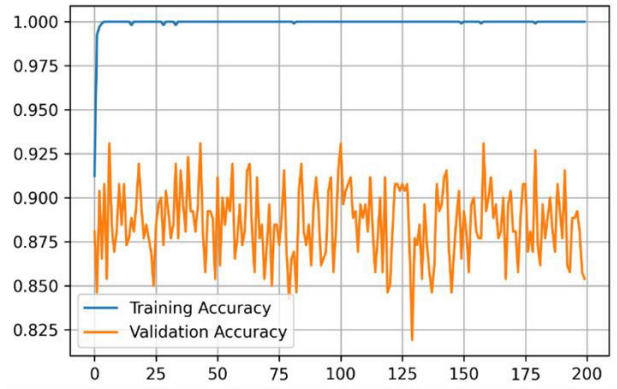

FIGURE 13. Training and validation accuracy of the TL-CNN model without the adversarial samples.

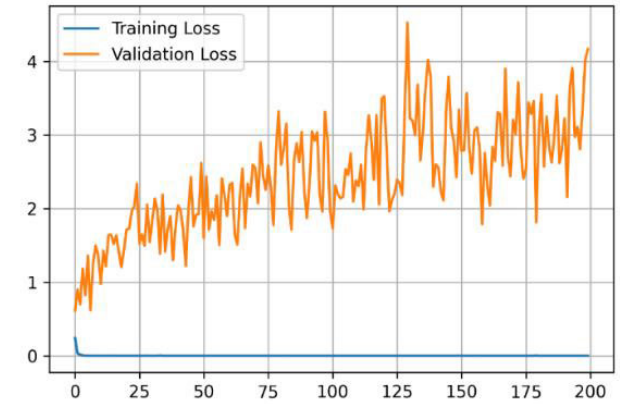

FIGURE 14. Training and validation loss of the TL-CNN model without the adversarial samples.

both models, for comparison reasons, was based on the same validation samples coming from Fashion-MNIST and FAS datasets.

The accuracy and loss results for the c-CNN model as well as the TL-CNN model, both for the training and validation sets, after 200 iterations are depicted in Table 5. It is worth mentioning that for the TL-CNN model there are two lines which refer to different training datasets. More specifically the second line of Table 5 is the TL-CNN model trained

TABLE 5. Accuracy and loss result for both the c-CNN and TL-CNN models.

	Accuracy		Loss	
Classifier	Training	Validation	Training	Validation
c-CNN	0,5095	0,5923	2.9653	6.5712
TL-CNN without FAS dataset	0,9879	0,8538	7.2094 x10−6	4,1676
TL-CNN with FAS dataset	0,9998	0,9969	4,6931x10−7	9.1699x10−10

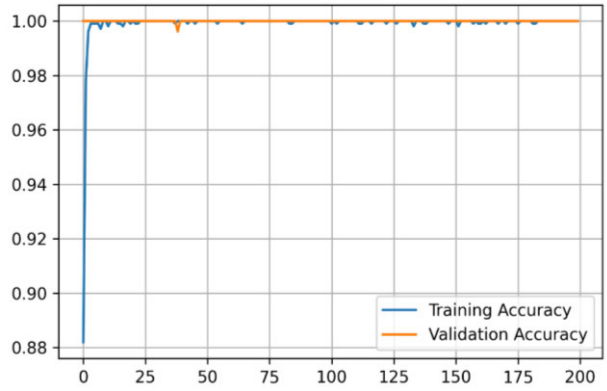

FIGURE 15. Training and validation accuracy of the TL-CNN model trained with the adversarial samples.

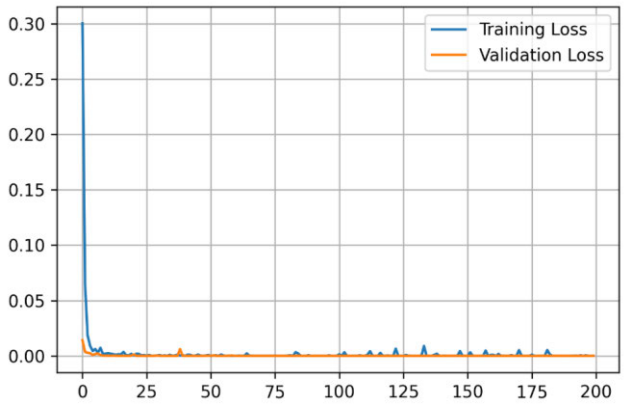

FIGURE 16. Training and validation loss of the TL-CNN model trained with the adversarial samples.

without adversarial samples (FAS dataset) and the third is about the TL-CNN model trained using the FAS dataset.

VI. ANALYSIS AND DISCUSSION OF RESULTS

The first step of the current study involved synthetic image generation using GANs. This specific step was critical, as, by expanding the initial dataset with identical (generated) datasets, larger samples of training and validation data were available, so as to achieve higher prediction performance using as inputs unknown data instances. As may be observed from Figure 5 and Figure 7, by the time the end of the 500th epoch was reached, the generated malware and malware-based images seemed more real-looking and quite similar to the input data samples. At this point, it is worth noting that, in the beginning of the training procedure, the discriminator loss is low, whilst it gets higher as the number of training epochs increases.

Figure 11 and Figure 12 depict the learning curves of the training and validation accuracy and loss of the c-CNN model, as described in Section V. As can be seen from the generated plots, the training and validation loss are off by large margins, whilst the model achieved a prediction accuracy of around 60%, both on the training and validation set of images. Additionally, an important fact that can be noted is the occurrence of overfitting in the created conventional CNN model, as it can be observed that the accuracy measured against the training dataset is quite high compared to the validation dataset. Moreover, the model's loss is high (Table 5) and not acceptable for such functions, as losses indicate the degree of models' prediction failures on the provided data. Therefore, higher losses reflect more prediction failures.

Figure 13 and Figure 14 depict the training and validation learning curves for the accuracy and loss of the TL-CNN model when using the ResNet50 base model as a fixed feature extractor, the Fashion-MNIST and the EMI dataset as training input. After approximately 200 iterations, the accuracy results for the training set of images reach 98,79%, whilst for the validation set of images reach 85,38%, while no overfitting is observed. Furthermore, the training and validation losses are different, as for the training they are close to zero and for the validation they are around 4 as shown in Table 5.

On the other hand, Figure 15 and Figure 16 depict the training and validation learning curves for the accuracy and loss of the TL-CNN model again using the ResNet50 base model architecture, the Fashion-MNIST and the FAS datasets as training input. After approximately 200 iterations, the accuracy results for both the training and validation set of images reach approximately 100%, while no overfitting is observed. Furthermore, the training and validation losses are close to zero, as shown in Table 5.

In comparison with the c-CNN model, the adoption of transfer learning, using the pre-trained ResNet50 model as basis, led to the creation of malware detectors that achieved remarkable results. Also, it is worth mentioning that the extension of the dataset size using the proposed FMGAN architecture was crucial. The mimicking of the malware-based generation process (adversarial samples) helped the detector to adjust its performance to more demanding conditions, as demonstrated by comparing the results of the two TL-CNN models with two different training datasets

(Table 5). This model was constructed and trained using a more complex input of training datasets, so as to achieve higher adaptability to possible new techniques and types of attacks from malicious attackers. Therefore, this type of deep learning models and techniques could be considered as additional enabling technologies in the information security area, as they can provide efficient and continuously up-to-date defending mechanisms against malicious cyber- threats.

VII. CONCLUSION

The goal of this study was to present a complete methodology from malware generation to malware detection. The generation of adversarial samples was achieved by using a DCGAN architecture named FMGAN, which was trained with the Malimg dataset in order to mimic malware authorship and produce new malware instances which can be used as a training dataset for malware detectors. Moreover, the same DCGAN architecture was used to produce adversarial image samples, which included fashion-MNIST images of products and malware and can be considered as malware ingested in product images.

For malware detection purposes, three different methods, based on CNN, were designed and tested. The first one contained a conventional CNN -c-CNN model-, whilst, for the second and third one, the transfer learning technique was employed to form the basis for a CNN model (TL-CNN). The difference between the second and the third transfer learning models was in the training input samples, the first one trained without and the second with the generated adversarial samples. The results indicated the over-performance of the transfer learning process of the ResNet50 model using the FAS dataset created for this study, compared to the conventional CNN approach as well as to the ResNet50 model trained without adversarial samples. The model encompassing transfer learning achieved almost perfect performance and no overfitting or underfitting phenomena were observed. On the contrary, the c-CNN model required more time to train, while the accuracy and loss results were considerably lower (the accuracy did not exceed 60% and the loss was, also, high, as it can be seen from Table 5). Moreover, the c-CNN model did not manage to avoid overfitting. Both models were given as input the adversarial samples dataset generated by the FMGAN. Moreover, the TL-CNN which trained using only fashion-MNIST and malware images achieved better results compared to the c-CNN but did not achieve to reach the performance of the ResNet50 model which trained with the generated adversarial samples.

The use of GAN models for the production of new malware or cyberthreats in general is an expanding research topic. The use of generative models, such as GANs, for simulation or production of cyberthreats can be a very useful tool in order to better train Machine Learning and Deep Learning solutions and equip them with enhanced capabilities for the recognition even of zero-day vulnerabilities. The increased interest in these newly introduced methods is evident in the related works section presented in this paper.

The methodology proposed in this study led to important findings. Firstly, the feasibility of using GANs to simulate the generation of new malware or/and adversarial samples and, secondly, the significance of transfer learning for malware detection contained in images were proven. New research ideas and information security solutions could benefit from these findings.

Future directions of this study aim to engage more complex datasets, such as RGB images of different products as well as different families and types of malware and other cyberthreats. Also, another part of research efforts will be devoted to the creation of a model not only for detection but also for classification into various categories of malware and products. This is crucial for sensitive applications of image tampering such as forensics. Moreover, the performance of this method can be further tested on real world pictures of actual products on the web that are infected with malware. This also suggests the exploration of lifelong learning approaches in order to create a future proof system that can be constantly updated to tackle malware attacks. Finally, the generation process presented in this study could be expanded in order to produce synthetic samples more efficiently, with a view to further stabilizing GAN performance by means of examining and reducing mode collapse and convergence failure phenomena. The process can also be applied for the creation of cyberthreats of various types, other than image malware, as well as in other domains of interest.

ACKNOWLEDGMENT

Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

REFERENCES

[1] S. Lone, N. Harboul, and J. W. J. Weltevreden. (2021). *2021 European E-Commerce Report*. Amsterdam University of Applied Sciences & Ecommerce Europe. Accessed: Mar. 16, 2022. [Online]. Available: https://ecommerce-europe.eu/wp-content/uploads/2021/09/2021-European-E-commerce-Report-LIGHT-VERSION.pdf

[2] S. Morrow and N. Maynard, "Online payment fraud: Emerging threats, segment analysis & market forecasts 2021–2025," Juniper Research, Juniper Research Ltd 9 Cedarwood, Chineham Park, Basingstoke, U.K., Tech. Rep., Apr. 2021. [Online]. Available: https://www.experian.com/blogs/global-insights/wp-content/uploads/2022/07/2021_04_Juniper_Online-Payment-Fraud.pdf

[3] European Payments Council (EPC). (Nov. 2021). *2021 Payment Threats and Fraud Trends*. T European Payments Council (EPC), Brussels. Public EPC193-21. Accessed: Mar. 18, 2022. [Online]. Available: https://www.europeanpaymentscouncil.eu/sites/default/files/kb/file/2021-12/EPC193-21%20v1.0%202021%20Payments%20Threats%20and%20Fraud%20Trends%20Report.pdf

[4] SOCRadar. (2021). *Threat Landscape Report*. SOCRadar. Accessed: Sep. 26, 2022. [Online]. Available: https://socradar.io/wp-content/uploads/2021/11/2021-E-commerce-Threat-Landscape-Report.pdf

[5] ENISA. *Malware*. Accessed: Mar. 20, 2022. [Online]. Available: https://www.enisa.europa.eu/topics/csirts-in-europe/glossary/malware

[6] Ö. A. Aslan and R. Samet, "A comprehensive review on malware detection approaches," *IEEE Access*, vol. 8, pp. 6249–6271, 2020, doi: 10.1109/ACCESS.2019.2963724.

[7] D. Gibert, C. Mateu, J. Planes, and R. Vicens, "Using convolutional neural networks for classification of malware represented as images," *J. Comput. Virol. Hacking Techn.*, vol. 15, no. 1, pp. 15–28, Mar. 2019, doi: 10.1007/s11416-018-0323-0.

[8] G. Conti, E. Dean, M. Sinda, and B. Sangster, "Visual reverse engineering of binary and data files," in *Visualization for Computer Security*, J. R. Goodall, G. Conti, and K.-L. Ma, Eds. Berlin, Germany: Springer, 2008, pp. 1–17.

[9] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, pp. 139–144, Nov. 2020, doi: 10.1145/3422622.

[10] J. Gui, Z. Sun, Y. Wen, D. Tao, and J. Ye, "A review on generative adversarial networks: Algorithms, theory, and applications," *IEEE Trans. Knowl. Data Eng.*, vol. 35, no. 4, pp. 3313–3332, Apr. 2023, doi: 10.1109/TKDE.2021.3130191.

[11] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.

[12] L. Nataraj, D. Kirat, B. Manjunath, and G. Vigna, "SARVAM: Search and RetrieVAl of malware," in *Proc. Annu. Comput. Secur. Conf. (ACSAC) Worshop Next Gener. Malware Attacks Defense (NGMAD)*, 2013. [Online]. Available: https://www.semanticscholar.org/paper/SARVAM-%3A-Search-And-RetrieVAl-of-Malware-Nataraj/c6e35dbd910fbc5f2bccc70380e741e2004aa440

[13] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[14] I. K. Dutta, B. Ghosh, A. Carlson, M. Totaro, and M. Bayoumi, "Generative adversarial networks in security: A survey," in *Proc. 11th IEEE Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2020, pp. 0399–0405, doi: 10.1109/UEMCON51285.2020.9298135.

[15] A. Singh, D. Dutta, and A. Saha, "MIGAN: Malware image synthesis using GANs," in *Proc. AAAI Conf. Artif. Intell.*, Jul. 2019, vol. 33, no. 1, pp. 10033–10034, doi: 10.1609/aaai.v33i01.330110033.

[16] S. Jang, S. Li, and Y. Sung, "Generative adversarial network for global image-based local image to improve malware classification using convolutional neural network," *Appl. Sci.*, vol. 10, no. 21, p. 7585, Oct. 2020, doi: 10.3390/app10217585.

[17] C. Xiao, B. Li, J.-Y. Zhu, W. He, M. Liu, and D. Song, "Generating adversarial examples with adversarial networks," 2018, *arXiv:1801.02610*.

[18] V. S. Bhaskara and D. Bhattacharyya, "Emulating malware authors for proactive protection using GANs over a distributed image visualization of dynamic file behavior," 2018, *arXiv:1807.07525*.

[19] R. L. Castro, C. Schmitt, and G. D. Rodosek, "Poster: Training GANs to generate adversarial examples against malware classification," *IEEE Security Privacy*, 2019. [Online]. Available: https://www.ieee-security.org/TC/SP2019/posters/hotcrp_sp19posters-final34.pdf

[20] Z. Moti, S. Hashemi, and A. Namavar, "Discovering future malware variants by generating new malware samples using generative adversarial network," in *Proc. 9th Int. Conf. Comput. Knowl. Eng. (ICCKE)*, Oct. 2019, pp. 319–324, doi: 10.1109/ICCKE48569.2019.8964913.

[21] Y. Fang, Y. Zeng, B. Li, L. Liu, and L. Zhang, "DeepDetectNet vs RLAttackNet: An adversarial method to improve deep learning-based static malware detection model," *PLoS ONE*, vol. 15, no. 4, Apr. 2020, Art. no. e0231626, doi: 10.1371/journal.pone.0231626.

[22] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on GAN," 2017, *arXiv:1702.05983*.

[23] R. Chauhan, U. Sabeel, A. Izaddoost, and S. Shah Heydari, "Polymorphic adversarial cyberattacks using WGAN," *J. Cybersecur. Privacy*, vol. 1, no. 4, pp. 767–792, Dec. 2021, doi: 10.3390/jcp1040037.

[24] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Zero-day malware detection using transferred generative adversarial networks based on deep autoencoders," *Inf. Sci.*, vols. 460–461, pp. 83–102, Sep. 2018, doi: 10.1016/j.ins.2018.04.092.

[25] J.-Y. Kim, S.-J. Bu, and S.-B. Cho, "Malware detection using deep transferred generative adversarial networks," in *Neural Information Processing*, D. Liu, S. Xie, Y. Li, D. Zhao, E.-S. M. El-Alfy, Eds. Cham, Switzerland: Springer, 2017, pp. 556–564.

[26] R. Burks, K. A. Islam, Y. Lu, and J. Li, "Data augmentation with generative models for improved malware detection: A comparative study," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Oct. 2019, pp. 0660–0665, doi: 10.1109/UEMCON47517.2019.8993085.

[27] K. He and D.-S. Kim, "Malware detection with malware images using deep learning techniques," in *Proc. 18th IEEE Int. Conf. Trust, Secur. Privacy Comput. Commun./13th IEEE Int. Conf. Big Data Sci. Eng. (TrustCom/BigDataSE)*, Aug. 2019, pp. 95–102, doi: 10.1109/TrustCom/BigDataSE.2019.00022.

[28] Y. Jian, H. Kuang, C. Ren, Z. Ma, and H. Wang, "A novel framework for image-based malware detection with a deep neural network," *Comput. Secur.*, vol. 109, Oct. 2021, Art. no. 102400, doi: 10.1016/j.cose.2021.102400.

[29] C. V. Bijitha and H. V. Nath, "On the effectiveness of image processing based malware detection techniques," *Cybern. Syst.*, vol. 10, pp. 1–26, Jan. 2022, doi: 10.1080/01969722.2021.2020471.

[30] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 53–65, Jan. 2018, doi: 10.1109/MSP.2017.2765202.

[31] TensorFlow. *Effective TensorFlow 2*. Accessed: May 4, 2021. [Online]. Available: https://www.tensorflow.org/guide/effective_tf2

[32] F. Chollet. *Keras*. Accessed: Sep. 12, 2022. [Online]. Available: https://github.com/fchollet/keras

[33] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015, *arXiv:1512.03385*.

[34] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, "Malware images: Visualization and automatic classification," in *Proc. 8th Int. Symp. Visualizat. Cyber Secur.*, New York, NY, USA, Jul. 2011, pp. 1–7, doi: 10.1145/2016904.2016908.

[35] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-MNIST: A novel image dataset for benchmarking machine learning algorithms," 2017, *arXiv:1708.07747*.

[36] L. Deng, "The MNIST database of handwritten digit images for machine learning research," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.

[37] M. Kayed, A. Anter, and H. Mohamed, "Classification of garments from fashion MNIST dataset using CNN LeNet-5 architecture," in *Proc. Int. Conf. Innov. Trends Commun. Comput. Eng. (ITCE)*, Feb. 2020, pp. 238–243, doi: 10.1109/ITCE48509.2020.9047776.

[38] D. Keysers, "Comparison and combination of state-of-the-art techniques for handwritten character recognition: Topping the MNIST benchmark," 2007, *arXiv:0710.2231*.

[39] D. Saxena and J. Cao, "Generative adversarial networks (GANs survey): Challenges, solutions, and future directions," 2020, *arXiv:2005.00065*.

[40] N. Kodali, J. Abernethy, J. Hays, and Z. Kira, "On convergence and stability of GANs," 2017, *arXiv:1705.07215*.

NIKOLAOS PEPPES received the dual Diploma degree in electrical and computer engineering and mining and metallurgical engineering from the National Technical University of Athens (NTUA), the M.B.A. degree in techno-economic systems from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), in 2019, and the master's degree in geoinformatics from NTUA, where he is currently pursuing the Ph.D. degree in deep learning for network security with the School of Electrical and Computer Engineering. He has extended experience in the cybersecurity and information security field. He became a Certified Ethical Hacker, in 2019. Being actively involved in international research and development projects he has advanced skills in security applications and practices; vulnerability scanning, threat hunting, and penetration testing in cyber and physical systems; building power energy management systems; telemetry and biotelemetry; mobile antennas; and design and supervision of low voltage systems. He has also contributed to several publications in recognized journals. He is a member of the PREVISION (H2020), ENSURESEC (H2020), and MANTIS project teams with a focus on big data and security solutions. His primary research interests include ethical hacking, cybersecurity, AI, machine learning, the IoT, big data, algorithms and data structures, mobile app development, and DevOps. He is a member of the Technical Chamber of Greece.

THEODOROS ALEXAKIS received the M.Eng. Diploma degree from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), in 2015, and the integrated M.B.A. degree in techno-economic systems from the School of Electrical and Computer Engineering, NTUA, and the Department of Industrial Management and Technology, University of Piraeus, in 2019. He is currently pursuing the Ph.D. degree in distributed architectures for content verification with the School of Electrical and Computer Engineering, NTUA. He was with the Greek Research and Technology Network (GRNET) as a Technical Support and Debugging Specialist in various projects. He has market experience as an Automation Engineer in many telemetry and industrial automation projects in infrastructures (water distribution networks and oil refineries). He is a member of EU-funded projects teams (MAGNETO, PREVISION, and ENSURESEC) with a focus on big data, semantics, artificial intelligence, and data analysis solutions. He has contributed to several publications in recognized journals. His primary research interests include artificial intelligence, automation applications, data analysis, the IoT, networks, and software design and development. He is a member of the Technical Chamber of Greece.

EMMANOUIL DASKALAKIS received the M.Eng. Diploma degree from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), and the master's degree in energy production and management from NTUA, in 2018. He was with the Greek Research and Technology Network (GRNET) as a Technical Support and a Debugging Assistant in a variety of web applications. He has market experience as an Automation and Instrumentation Engineer in telemetry, automation, and instrumentation projects of water distribution networks, oil refineries, and food industries. He has experience in PLC and SCADA automation systems, power distribution and management, optimization methods, telecommunication systems, instrumentation, and vulnerability scanning and remediation in cyber and physical systems. His primary research interests include industrial automation and instrumentation applications, security systems, the IoT, machine learning, databases, and software design and development. He is a member of the Technical Chamber of Greece.

KONSTANTINOS DEMESTICHAS received the Diploma and Ph.D. degrees in telecommunications from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), in 2005 and 2009, respectively, the M.B.A. degree in techno-economic systems through the joint postgraduates' program with NTUA and the University of Piraeus, in 2012, and the M.Sc. degree in quality assurance from Hellenic Open University, in 2015. In 2021, he joined the Department of Agricultural Economics and Rural Development, Agricultural University of Athens, as an Assistant Professor. He has served as a member of the Informatics Laboratory. Before joining the Agricultural University of Athens, he was a Lecturer with NTUA, the University of Western Macedonia, the University of West Attica, and Hellenic Open University. Since 2005, he has been actively involved in several European and national research projects. He was the concept initiator and primary proposal author of several EU-funded projects. He has served as a scientific or project coordinator in EU-funded projects from the security and ICT domains, such as INLIFE, MAGNETO, PREVISION, SocialTruth, and ALAMEDA. He has authored more than 160 publications. He has participated in the technical program committees of international conferences. He has assisted as a reviewer and an editor in top-ranked scientific journals. He has also served as a Technical Expert for the European Commission.

EVGENIA ADAMOPOULOU received the Diploma and Ph.D. degrees in mobile communications from the School of Electrical and Computer Engineering, National Technical University of Athens (NTUA), in 2005 and 2009, respectively, and the M.B.A. degree in techno-economic systems from NTUA, in 2012. Since 2005, she has been a Senior Research Associate with the Computer Networks Laboratory, NTUA. Since 2014, she has also been teaching the course of information and telecommunication technology with NTUA. She has authored more than 140 publications in international journals and conferences. Her primary research interests include machine learning techniques, mobile and computer networks, context awareness, and mobile services. Since 2005, she has been actively involved in several European and national research projects in the aforementioned fields (FP5 MONASIDRE, FP6 MOTIVE, FP6 DAIDALOS, FP7 PERSIST, FP7 EcoGem, FP7 EMERALD, H2020 MAGNETO, H2020 PREVISION, H2020 SOCIALTRUTH, and H2020 STARLIGHT), where she has often assumed the role of the Dissemination Manager and the Task Leader. She has participated in the Technical Program Committees of international conferences. She has served as a reviewer for top-ranked scientific journals and a Project Reviewer for the European Commission.