Information Sciences 523 (2020) 38–48

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

A feature-vector generative adversarial network for evading PDF malware classifiers

Yuanzhang Li[a], Yaxiao Wang[a], Ye Wang[b], Lishan Ke[c], Yu-an Tan[a,*]

[a] School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100081, China
[b] Development & Service Center for Science & Technology Talents, The Ministry of Science & Technology (MoST), Beijing, 100045, China
[c] School of Computer Science, Guangzhou University, Guangzhou 510006, China

ARTICLE INFO

Article history:
Received 24 December 2019
Revised 8 February 2020
Accepted 27 February 2020
Available online 7 March 2020

Keywords:
PDF malware
Evasion attack
Generative adversarial network
Adversarial malware examples

ABSTRACT

Cyber-Physical Systems (CPS) are increasingly utilizing machine learning (ML) algorithms to resolve different control and decision making problems. CPS are traditionally vulnerable to evasion attacks and adversarial examples, hence the integration of learning algorithms requires that these vulnerabilities are reevaluated to make the cyber-physical systems more secure and robust. In this work, we propose a novel evasion method based on a feature-vector generative adversarial network (fvGAN) to attack a learning-based malware classifier. The generative adversarial network (GAN) has been widely used in the realistic fake-image generation, but it has rarely been studied for adversarial malware generation. This work uses the fvGAN to generate adversarial feature vectors in the feature space, and then transforms them into actual adversarial malware examples. We have experimentally investigated the effectiveness of the proposed approach on a well-known PDF malware classifier, PDFRate, and evaluated the fvGAN-based attack in four evasion scenarios. The results show that the fvGAN model has a high evasion rate within a limited time. We have also compared the proposed approach with two existing attack algorithms, namely Mimicry and GD-KDE, and the results prove that the proposed scheme has better performance both in terms of evasion rate and execution cost.

1. Introduction

The rapid development of computing, networking, and sensing technologies has enabled the deployment of cyber-physical systems (CPS) in various safety-critical situations such as healthcare, transportation, and aerospace. In recent years, machine learning techniques have seen an increase in use to solve the control and decision making problems of different cyber-physical systems [3,19,23,40]. In a typical application, a cyber-physical system can classify things and then perform actions based on the derived classification. Machine learning has been reported to achieve excellent and in some cases human-competitive performance in classification tasks [6,16,17,24,37]. In this regard, work in [17] shows that a 4.94% top-5 test error on the ImageNet 2012 classification dataset can be achieved, which surpasses human performance. Similarly, in [37], authors achieve 99.9% accuracy in PDF malware classification using a SVM-RBF model.

Given the benefits of machine learning approaches, recent researches have also shown that learning-based systems are vulnerable to evasion attacks [7,25,27,32,39,44], which can make learning-based cyber-physical systems extremely vulner-

* Corresponding author.
E-mail address: tan2008@bit.edu.cn (Y.-a. Tan).

https://doi.org/10.1016/j.ins.2020.02.075
0020-0255/

able, and the potential damage can be significant [8–11,14,15,26]. Attackers can easily evade such learning-based systems through deliberately designed adversarial examples, i.e., minor changes to the input data that leads to misclassification at test time [13,20,30,38]. For example, Grosse et al. [13] misled a neural network for 63% of all the malware samples they carefully crafted. Similarly, Kolosnjaji et al. [20] evaded a deep learning-based system with high probability by changing some specific bytes in the malware samples.

To build effective and robust cyber-physical systems, analyzing the adversarial examples has become extremely important. Based on this premise, this work focuses on the evasion attacks on learning-based PDF malware classifiers. Numerous innovative evasion attacks on PDF malware classifiers have been discussed in the literature [2,28,35,42]. Our method is based on the generative adversarial network (GAN) proposed in [12], generating the adversarial PDF malware using the GAN.

As GANs have the ability to learn complex data distribution and generate realistic samples that are indistinguishable from the training dataset, they are considered to be one of the most effective instruments in misleading classification [11]. Studies in [4,22] show that GANs can have excellent performance in generating realistic fake images. However, to the best of our knowledge, little significant effort has been done for the development of GANs that can generate adversarial malware examples. The work in [18] is perhaps the only one that produces adversarial malware examples, however, it does it only in the feature space without transforming them into actual malicious files. It directly and arbitrarily modifies the features in the feature vector representation of a malicious entity without considering restrictions on modifying an actual file. In contrast to this, we are working to generate actual adversarial malicious PDF files to attack a learning-based PDF malware classifier, PDFRate [34]. We use an openly available re-implementation of the PDFRate named mimicus for our experiments, which has nearly equivalent classification performance to the PDFRate classifier. Our specific contributions in this work are summarized as below:

- We present a novel evasion attack approach based on a feature vector-based GAN, i.e., feature-vector GAN (fvGAN), which is used to generate adversarial PDF malware in the feature space.
- We evaluate the method on a well-known PDFRate classifier, and perform attacks in four evasion scenarios with varying degrees of knowledge of the target system.
- We compare our approach with two existing evasion attacks, i.e., the Mimicry and GD-KDE attacks in terms of evasion rate and execution cost.

The remainder of this paper is organized as follows: In Section 2, we present the preliminaries, including the PDF structure and the mimicus framework. In Section 3, we present the novel evasion attack based on a fvGAN and explain the non-trivial challenges in generating actual adversarial malicious PDF files. The experimental evaluation and results are discussed in Section 4. In Section 5, we review the related works on evasion attacks on PDF malware classifiers. Finally, we conclude the paper in Section 6.

2. Preliminaries

2.1. PDF structure

The Portable Document Format (PDF) is one of the most popular file formats to present content and layout on different platforms. An example of a PDF file structure is depicted in Fig. 1. It comprises four parts: the header, body, a cross-reference table (CRT), and the trailer. Below we briefly detail the contents of each part.

Header: It is the first line of a PDF file, which indicates the version of the PDF format.

Body: This is a major component of any PDF file, which comprises a sequence of objects that define the operations performed by the file. These objects might contain compressed or uncompressed embedded data, e.g., text, images, script code, etc. Each object has a unique reference number and can also be referenced by other objects.

Cross-Reference Table (CRT): It is a table that includes a list of offsets that indicate the position of every indirect object in the file. Each entry in the table corresponds to a specific object, but only one ending in `n` indicates the total number of objects stored in the file. It is worth noting that the PDF readers only parse objects referenced by the CRT, which might be exploited by attackers.

Trailer: This is a special object that contains information about how the PDF reader should find the CRT and other objects in a file. A standard PDF reader parses a PDF file starting from the trailer, locating the CRT by referring to it and then using the CRT to find the objects in the body.

2.2. Mimicus

Mimicus [36] is an evasion framework by Šrndić et al. [21], which mainly consists of two parts: the re-implementation of the PDFRate and the evasion attacks against the PDFRate. It supports feature extraction, PDF file modification, training of local classifiers, performing attacks against the PDFRate classifier, etc. The PDFRate employs a total of 202 features, however, the feature extraction part of the mimicus can extract only 135 features of a PDF file as only these are documented. In the mimicus, the authors performed two kinds of evasion attacks on the PDFRate, i.e., Mimicry and GD-KDE. Both attacks were implemented in two steps. First, they selected or produced a target feature vector that could evade the classifier in the feature space for each attack file. In the Mimicry attack, the feature vector of a benign file was chosen as the target

Header	%PDF-1.4
Body	12 0 obj <</Length 13 0 R /Filter /FlateDecode>> stream … endstream endobj
Cross-Reference Table	xref 0 47 0000000000 65535 f 0000029847 00000 n …
Trailer	trailer <<… /Root 3 0 R…>> startxref 30277 %%EOF

Fig. 1. An example of the PDF file structure.

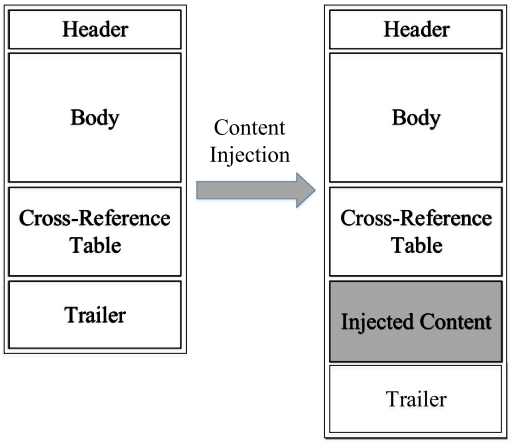

Fig. 2. Content injection between the CRT and the trailer.

feature vector. Moreover, to increase the effectiveness of evasion, they used 30 different benign files as candidates for each malicious file. While in the GD-KDE attack, the target feature vector was produced by running the GD-KDE algorithm.

In the second step, the target feature vector was transformed into a PDF file by modifying the original malicious file to mimic the feature values of the target feature vector. The mimicus modified the original malicious file through a content injection approach, i.e., injecting additional structural and metadata items into the gap between the CRT and the trailer of the malicious document without removing or modifying the existing elements, as illustrated in Fig. 2. Only the resulting PDF file which received the lowest classification score from the local classifier was submitted to the PDFRate. This approach exploits a weakness in the feature extractor of the PDFRate because the injected contents would not be interpreted by commercial[1] PDF readers, but would be parsed by the PDFRate and further influence the extracted feature values.

It is worth mentioning that it is not feasible to construct a final malicious file with the feature vector exactly matching the target feature vector. This is because the features of PDF files are heavily interdependent, i.e., modifying one feature necessarily affects several other features. Another important consideration is that algorithms operating in the feature space may produce invalid data which cannot be transformed into a file. Due to these two limitations, the feature vector of the resulting final malicious file is unpredictable.

Šrndić et al. performed attacks in four different evasion scenarios, i.e., `F`, `FT`, `FC`, `FTC`. The labels `F`, `T`, and `C` denote that the adversary could know the feature set, training dataset, and classification algorithm employed by the PDFRate, respectively. In different scenarios, they adopted different training datasets and classification algorithms to train the local classifier

[1] These include Adobe Acrobat, Evince, etc.

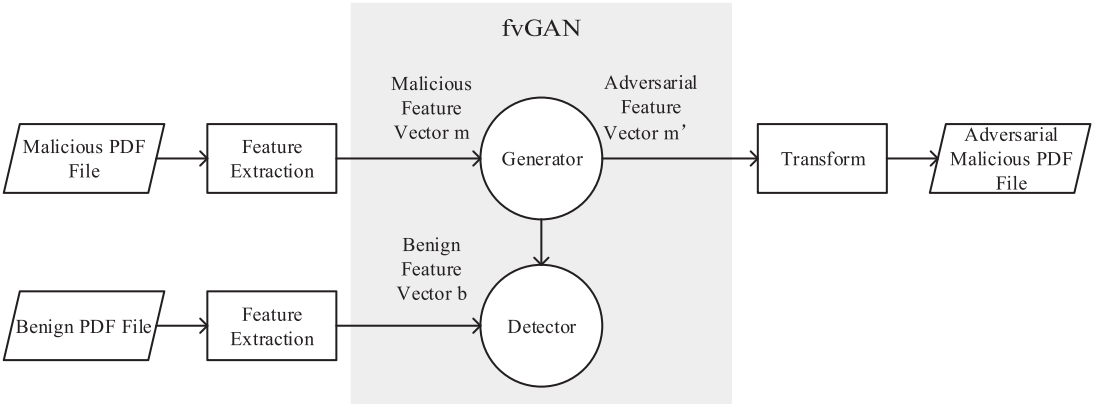

Fig. 3. The overall architecture of generating adversarial malicious PDF files based on a fvGAN.

depending on whether the training dataset or classification algorithm employed by the PDFRate was known or not. In scenario FT and FTC, they used an identical replica of the original training dataset of the PDFRate, namely *Contagio* dataset. While in scenario F and FC, wherein they had no access to the original dataset, they collected a surrogate dataset, namely *Surrogate* dataset, samples from which were obtained from VirusTotal[2] and Google web[3], to approximate it. Since the random forest (RF) was employed by the PDFRate as the classification algorithm, it was used in scenario FC and FTC. While in scenario F and FT wherein the classifier algorithm was not known, the authors chose a support vector machine (SVM) as a surrogate classifier. The authors performed the Mimicry attack in all scenarios mentioned earlier, while performed the GD-KDE attack only in scenario F and FT because the GD-KDE attack was only applicable to differentiable classifiers, such as SVM or artificial neural network.

In this work, we have performed our proposed fvGAN attack in all scenarios to give a more comprehensive evaluation.

3. Proposed fvGAN-based scheme

The proposed approach to construct adversarial malicious PDF files based on a fvGAN is illustrated in Fig. 3. First, we extract the features of the original malicious or benign PDF files using the feature extraction component of the mimicus. Based on these extracted features, each PDF file is represented by a 135-dimensional real vector. Secondly, these vectors are fed to the fvGAN module to produce adversarial PDF malware in the feature space. Lastly, we transform the generated adversarial PDF malware (as adversarial feature vectors) into actual malicious PDF files using the content injection approach mentioned earlier. Below we detail the process of the adversarial feature vector generation and its transformation to the actual malicious file.

3.1. fvGAN for generating adversarial feature vectors

Based on the GAN proposed in [12], we have made some modifications to make it better applicable to the generation of adversarial PDF malware. The modified GAN is called feature-vector GAN (fvGAN), and its objective is to generate adversarial feature vectors, which are similar to the feature vectors of benign PDF files. The architecture of the fvGAN is shown in Fig. 3. The generator is used to translate a malicious feature vector into an adversarial feature vector, which is used as the target when generating an actual adversarial malicious PDF file. It takes the 135-dimensional feature vector m of a malicious PDF file as input, and outputs a 135-dimensional adversarial feature vector m'. Each element of m and m' corresponds to a feature of a malicious PDF file.

The input vector is fed into a multi-layer fully connected network with weights θ_g. The network has one input layer, one output layer, and multiple hidden layers. Both the input and output layers have 135 neurons. The activation function used by the hidden layers is *leaky_relu*, while the output layer does not use any activation function. The function *leaky_relu* is defined as:

$$f(x) = \begin{cases} x, & x \geq 0 \\ kx, & x < 0 \end{cases} \tag{1}$$

where, $k < 1$ is an optional parameter, and we set it to 0.2 in the paper. The loss function of the generator is defined as:

$$L_g = -E_{m \sim M}[D(G(m))] \tag{2}$$

[2] VirusTotal: https://www.virustotal.com/.
[3] Google: https://www.google.com.

Table 1
Partial feature changes during a fvGAN attack.

Feature	Original	Target	Resulting
author_dot	0	0	0
author_lc	0	4	4
author_len	0	-15	20
author_mismatch	0	0	1
author_num	0	3	3
author_oth	0	0	0
author_uc	0	13	13
title_uc	0	-1	0

where, G and D denote the generator and detector, respectively, and M represents the distribution of the features of all the PDF samples in the malicious dataset. To train the generator, L_g should be minimized for the weights of the generator.

The detector takes a benign feature vector b as well as a generated vector m' as input, where both b and m' are 135-dimensional vectors, and m' is somewhere between a benign feature vector and a malicious feature vector. The input vectors are fed into a multi-layer fully connected network with weights θ_d. This network also has one input layer, one output layer, and multiple hidden layers. The output layer of this network has only one neuron because the output of the detector is the probability that a sample comes from the benign training data rather than the generator. The activation function used by the hidden layers is also *leaky_relu*, and the output layer is *sigmoid*. The loss function of the detector is defined as:

$$L_d = -E_{b\sim B}[logD(b)] - E_{m\sim M}[\log(1 - D(G(m)))] \tag{3}$$

where, B represents the distribution of the features of all the PDF samples in the benign dataset. L_d should be minimized for the weights of the detector. The whole process of training the fvGAN for generating adversarial feature vectors is detailed in Algorithm 1. In this algorithm, *Size* means the total number of PDF files in the benign training dataset, while *BatchSize* is the number of PDF files chosen to train the model in one iteration.

Algorithm 1 Training process of the fvGAN for generating adversarial feature vectors.

```
for epoch in epochs: do
    Size ← the size of benign training dataset
    for iteration in Size/BatchSize: do
        Update the detector's weights by descending the gradient ▽θd Ld
        Update the generator's weights by descending the gradient ▽θg Lg
    end for
end for
```

3.2. Transforming adversarial feature vectors into actual malicious files

After generating adversarial feature vectors using the generator, we use the content injection approach described earlier to transform them into actual adversarial malicious PDF files. Due to the interdependency of features and the possibility of invalid data points generated by the generator, it is impossible to generate an adversarial malicious file with feature vector identical to the adversarial feature vector. Hence, the feature vector of the resulting final file is somewhere between that of the original malicious file and that of a benign file. Table 1 shows the partial feature changes during a fvGAN attack. The *Original* column denotes the features of the original malicious file, and the *Target* column denotes the adversarial features generated by the fvGAN, while the *Resulting* column denotes the features of the resulting final file. It can be observed that the feature *author_len*, which indicates the length of the *Author* metadata field, got the value 20 instead of −15, because other modifiable features, such as, *author_lc, author_num, author_oth* and *author_uc*, drove it to 20. Besides, feature *title_uc* did not change at all because the target value was outside of valid bounds for this feature, so the mimicus prevented its modification.

It can be concluded that, although the generated adversarial PDF malware (as feature vectors) can evade the classifier in the feature space, its corresponding resulting malicious file might still fail to evade it in the problem space. Fig. 4 shows a concrete example of a fvGAN attack in the FTC scenario, involving 30 attack samples. It presents the specific classification scores for the adversarial PDF malware samples generated by the fvGAN and their corresponding resulting malicious files. It can be observed that the resulting malicious file might receive a high classification score even though the generated adversarial sample receives a low score. More specifically, it can be observed from the first and eighth samples. Therefore, we could not find an apparent relationship between the classification scores for the generated adversarial PDF malware and the resulting malicious files. Hence, we cannot utilize this relationship to improve the performance of the fvGAN. Moreover, it cannot be guaranteed that a fvGAN model with outstanding performance can be obtained after each training, as shown in Table 2 and Table 3. They present the evasion rate for 100 malicious PDF files separately generated by 10 fvGAN models in

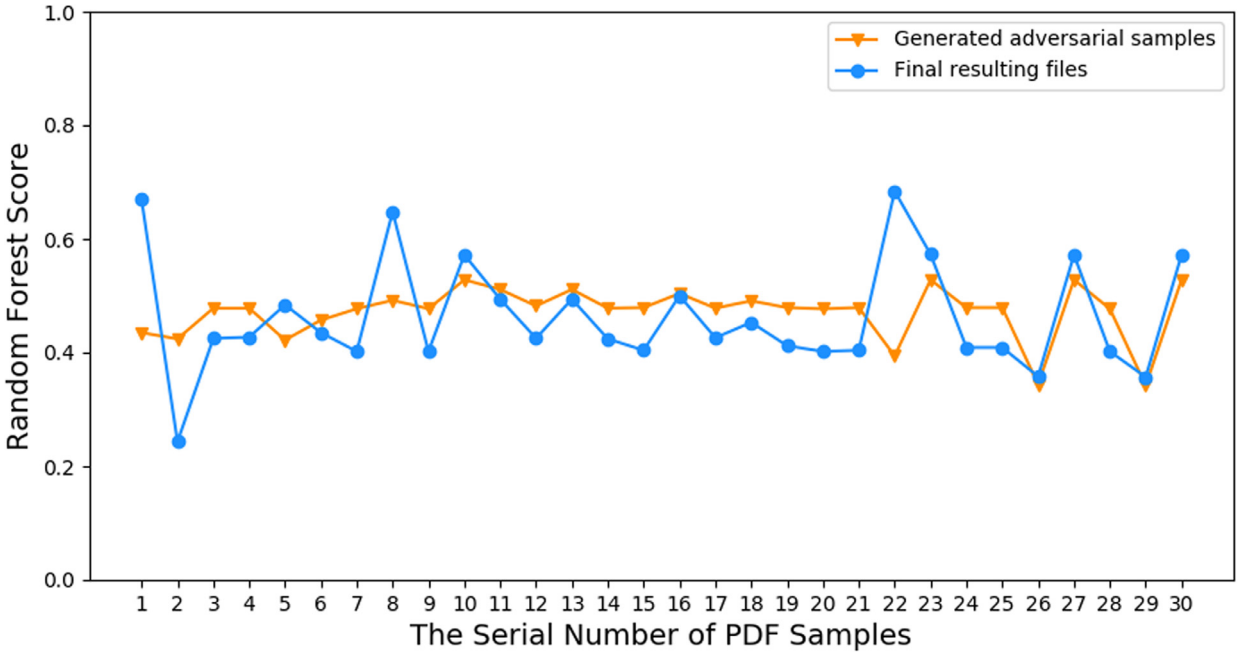

Fig. 4. Random forest classification scores for 30 generated adversarial PDF malware samples and their corresponding resulting PDF files. The cutoff of the local random forest classifier is set to 0.5. Samples that receive a score greater than 0.5 are labeled malware, otherwise benign.

Table 2
Evasion rate for 100 malicious PDF files separately generated by 10 fvGAN models in scenarios F and FC.

	1	2	3	4	5	6	7	8	9	10
F	100	100	100	100	100	100	100	100	100	100
FC	94	89	23	87	92	99	21	90	91	13

Table 3
Evasion rate for 100 malicious PDF files separately generated by 10 fvGAN models in scenarios FT and FTC.

	1	2	3	4	5	6	7	8	9	10
FT	100	100	100	100	100	100	100	100	100	100
FTC	97	0	89	90	43	19	94	87	96	16

different evasion scenarios. It can be seen that the best model can reach a 99% evasion rate, while the worst model attains a 0% evasion rate.

In this work, we address this problem by training the fvGAN multiple times and then select the best one to perform attacks. As shown in Tables 2 and 3, we can guarantee that a fvGAN model with an evasion rate exceeding 90% can be trained within a limited duration.

4. Evaluation and analysis

To verify the effectiveness of our proposed approach, we conducted our experimental evaluation using PRFRate, a well-known structural feature-based PDF malware detector. In our experiment, we utilized the mimicus, an open-source re-implementation of the PDFRate. We first describe our experimental setup and then present the results in terms of evasion rate and execution cost in the following sections.

4.1. Experimental setup

Our experiment involved three datasets, i.e., *Contagio, Surrogate*, and *Attack* dataset, all of which could be available within the mimicus framework. The *Contagio* dataset contains 5000 benign and 4999 malicious PDF files, while the *Surrogate* dataset contains 5000 benign and 5000 malicious PDF files. In our experiment, we separately used these two datasets to train the fvGAN. The *Attack* dataset containing 100 malicious PDF files was used to generate adversarial malicious PDF files

Table 4
Evasion rate achieved by three kinds of evasion attacks in four scenarios. (The proportion of the samples that could evade the PDF classifier in the total generated PDF samples).

Scenario	fvGAN	Mimicry	GDK-DE
F	100	100	2
FT	100	100	2
FC	99	100	X
FTC	97	93	X

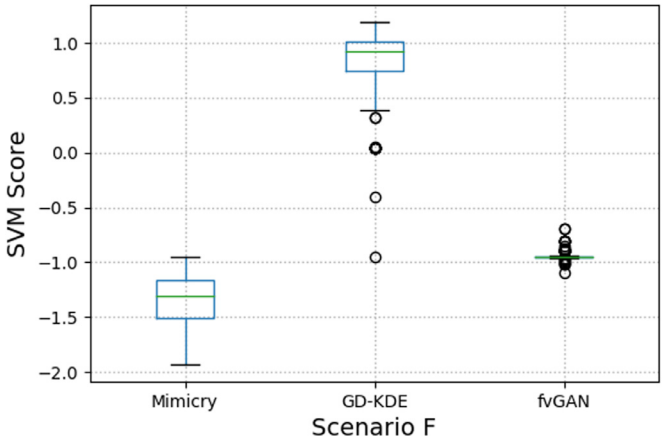

Fig. 5. Scores by the local PDFRate classifier on three kinds of attacks in scenario F, for all 100 attack samples from the *Attack* dataset.

and test the efficacy of the trained fvGAN model. Since the *Attack* dataset only contains 100 files, the calculated evasion rate is limited to a total of 100.

We performed the fvGAN attacks in all four attack scenarios as mentioned in [21]. In the FT and FTC scenarios, we used the *Contagio* dataset to train the fvGAN. In the other scenarios, the *Surrogate* dataset was used. For each attack, we trained the fvGAN 10 times and then selected the best model to perform the attack. For the RF classifier, the cutoff value was set to 0.5, while for the SVM classifier, it was set to 0. According to the knowledge summarized by predecessors and several attempts in our experiment, we used the Adam optimizer for both the generator and detector networks with a learning rate of α=0.00002 and the other coefficient of β_1=0.5. Both the generator and detector were fully-connected networks, and both had one input layer, two hidden layers, and one output layer. The generator's layer size was 135-270-270-135, and the detector's layer size was 135-270-270-1. The batch size for the experiment was 32, and the maximum number of epochs to train the fvGAN was 500.

Before training the fvGAN, we pre-processed the training dataset. In the F and FC scenarios, features of malicious and benign PDF files were both standardized, while in the FT and FTC scenarios, only features of malicious PDF files were standardized.

4.2. Evasion rate

The 10 fvGAN models trained for each scenario are shown in Tables 2 and 3. It can be observed that the sixth model in Table 2 and the first model in Table 3 have the best performance, hence they are selected to generate adversarial malicious PDF files.

We reproduced the work of Šrndić et al. [21] and compared it to the fvGAN attack in terms of evasion rate in different evasion scenarios. Since our work used the same attack files as in [21], the results were limited to a total of 100 samples per attack. The comparison results are shown in Table 4 for different scenarios. We can observe that in terms of evasion rate, the fvGAN attack could achieve performance comparable to that of the Mimicry attack. Furthermore, we find that the performance of the GDK-DE attack is not as effective as given in[21]. It only achieves a 2% evasion rate both in the F and FT scenarios.

The specific classification scores received from the local classifier for malicious files generated by these attack algorithms are illustrated in Fig. 5–8. For each attack, the distribution of the classification scores for 100 generated malicious files is represented as a box plot, where the thick line inside the box denotes the median, and the top and bottom edge of the box indicates the upper and lower quartile of these scores. Scores of 1.5 IQR (interquartile range) from the upper and lower quartile are also shown as short thick lines, while the remaining points are outliers.

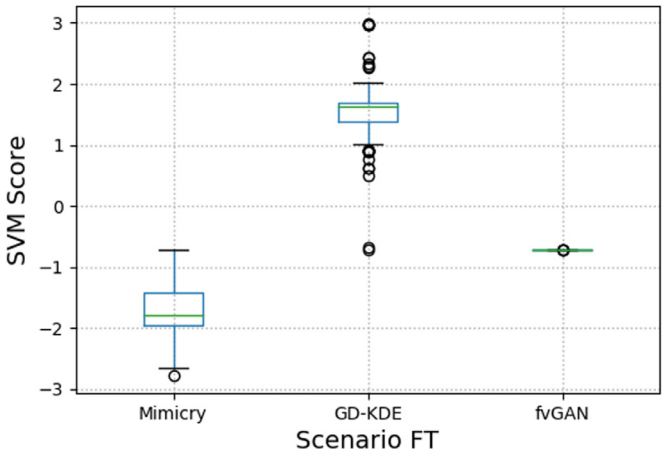

Fig. 6. Scores by the local PDFRate classifier on three kinds of attacks in scenario FT, for all 100 attack samples from the *Attack* dataset.

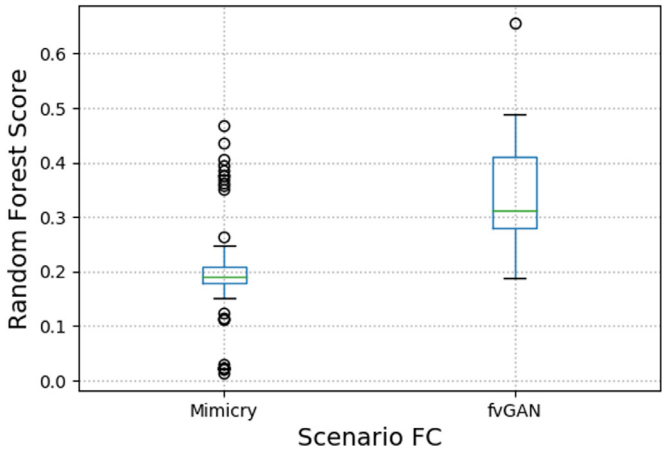

Fig. 7. Scores by the local PDFRate classifier on three kinds of attacks in scenario FC, for all 100 attack samples from the *Attack* dataset.

In Figs. 5 and 6, it can be seen that in the F and FT scenarios, most of the classification scores for generated files by GD-KDE attack are greater than 0, and only a few outliers are less than 0. This discovery is quite different from the result shown in [21]. In all the result figures, we confirm that the Mimicry attack is indeed immensely effective, and the results we reproduced are similar to [21]. Moreover, we can also observe that the classification scores for 100 generated malicious files by Mimicry are generally less than that of fvGAN. While the scores for malicious files generated by the fvGAN are slightly more centralized. As can be seen in Figs. 5 and 6, the box is almost a straight line.

Fig. 9 shows the ROC curves of the baseline and all attacks in scenario F, FT, FC, and FTC. The baseline curve is obtained on a mixed data sample containing 100 malicious samples and 100 benign samples. For Baseline curve, the malicious samples are from the *Attack* dataset, while for the other curves, they are the corresponding generated adversarial samples. It can be seen from Fig. 9 that the fvGAN attack could significantly reduce the detection performance of the PDFRate in all scenarios compared with the baseline. And the PDFRate classifier achieves a lower accuracy (a smaller AUC) under the fvGAN attack than under the GD-KDE attack, which indicates that the fvGAN attack is obviously superior to the GD-KDE attack. This result is very similar to what the Figs. 5, ,–8 shows. Also, as all these ROC curves illustrate, the Mimicry attack obtains better evasive performance than the fvGAN attack in most scenarios due to the direct mimicry of the benign files as we explained earlier.

5. Related works

Adversarial learning has received significant attention in different communities, ranging from computer vision tasks [9,31,33,43,45] to malware detection [1,18,21,41]. Evasion attacks against learning-based PDF malware classifiers have been studied in numerous works.

Mimicry is a conventional attack algorithm against a learning-based classifier. In such attacks, benign information is injected into the malicious file to make it look more similar to a benign file. Smutz et al. [34] successfully reduced the clas-

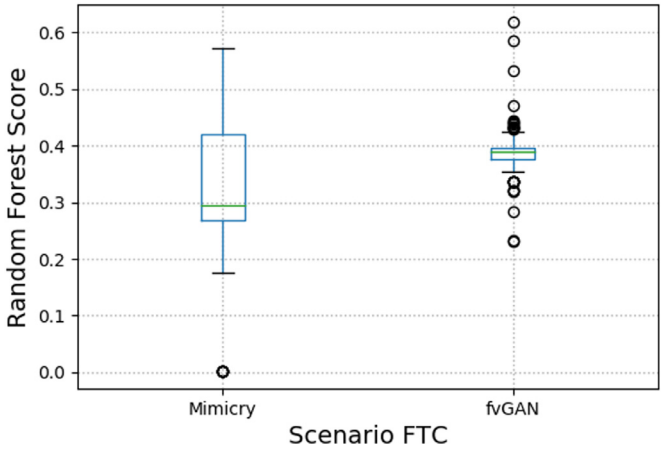

Fig. 8. Scores by the local PDFRate classifier on three kinds of attacks in scenario FTC, for all 100 attack samples from the *Attack* dataset.

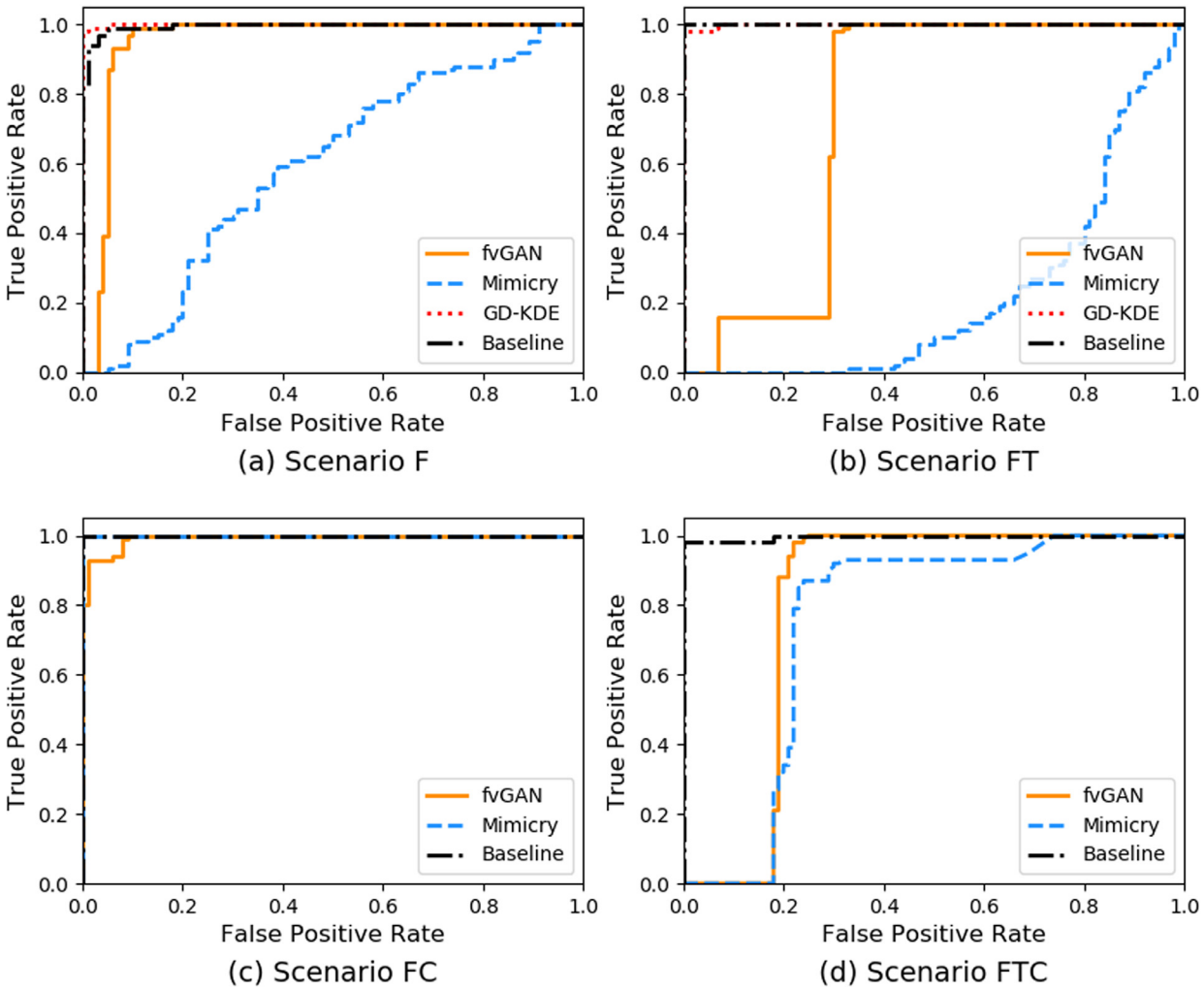

Fig. 9. (a):ROC curves of the baseline and all attacks in scenario F. The benign samples are from the *Contagio* dataset for all curves. (b):ROC curves of the baseline and all attacks in scenario FT. The benign samples are from the *Surrogate* dataset for all curves. (c):ROC curves of the baseline and all attacks in scenario FC. The benign samples are from the *Contagio* dataset for all curves. (d):ROC curves of the baseline and all attacks in scenario FTC. The benign samples are from the *Surrogate* dataset for all curves.

sifier's accuracy by injecting information (into malicious samples) related to the most discriminant features of the PDFRate. Corona et al. [5] added all the features of many benign files to malicious files to evade Lux0R, but failed due to the dynamic features. Šrndić et al. [21] successfully evaded the PDFRate by adding additional unused contents between the CRT and the trailer. Xu et al. [42] attacked the PDFRate and Hidost by injecting target features directly into PDF file objects. They used a genetic algorithm to find the evasive samples, which was the only one to remove features from malicious files, and the method has been demonstrated to be very useful in evading both systems.

Another evasion attack named reverse mimicry attack, in which a small malicious payload is injected into benign files was first performed by Maiorca et al. [29]. This attack evaded various structural feature-based systems, including Wepawet, PJScan, Slayer, and PDFRate by injecting EXE payloads, PDF files or Javascript codes into benign PDF files.

In our work, we inject some useless contents into malicious PDF files, so our method is a mimicry attack. However, different from the above works, our work utilize the GAN to learn the data distribution of the benign PDF files, which could improve the efficiency of the generation of adversarial PDF malware.

6. Conclusion

In this paper, we propose a novel evasion attack against PDF malware classifier based on the fvGAN, which is proposed to generate adversarial feature vectors in the feature space. Then, we utilize the mimicus framework to transform these vectors into actual adversarial malicious PDF files. We evaluated our approach on a state-of-the-art PDF malware classifier, PDFRate, and performed the fvGAN attack in different evasion scenarios. For every scenario, fvGAN was trained for 10 times to ensure that a model with outstanding performance can be obtained within a limited time. We compared the performance with the Mimicry and GD-KDE attacks in terms of evasion rate and execution cost. The experimental results reveal that our approach can achieve evasion rates far surpassing the GD-KDE and marginally outperforming the Mimicry attack. Moreover, our approach is more advantageous than the Mimicry attack when involving a considerably large number of adversarial malicious PDF files. The evaluation also shows that even with some strict limitations, such as interrelated features or invalid values generated by the fvGAN, the proposed fvGAN attack could still achieve excellent performance in evading the PDF malware classifier. The proposed attack and its security implications can also apply to other learning-based systems.

Our study is limited to the application of the fvGAN to evade the PDF malware classifier only in the feature space. The feature selection and extraction as well as the generation of actual PDF files require human or other tools to intervene, which greatly limits the ability and performance of the GAN. A GAN that can directly generate actual PDF files in the problem space might alleviate those limitations like feature interdependency. However, we are currently not aware of any techniques for designing such GANs and therefore leave this problem for future work.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests.

CRediT authorship contribution statement

Yuanzhang Li: Methodology, Formal analysis, Writing - review & editing, Data curation. **Yaxiao Wang:** Software, Validation, Formal analysis, Resources. **Ye Wang:** Visualization, Investigation. **Lishan Ke:** Formal analysis, Investigation. **Yu-an Tan:** Conceptualization, Project administration, Supervision, Funding acquisition.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under grant no. U1936218 and no. 61876019.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, F. Roli, Evasion attacks against machine learning at test time, in: Joint European conference on machine learning and knowledge discovery in databases, Springer, 2013, pp. 387–402.
[2] C. Carmony, X. Hu, H. Yin, A.V. Bhaskar, M. Zhang, Extract me if you can: Abusing pdf parsers in malware detectors., NDSS, 2016.
[3] F. Castaño, G. Beruvides, R. Haber, A. Artuñedo, Obstacle recognition based on machine learning for on-chip lidar sensors in a cyber-physical system, Sensors 17 (9) (2017) 2109.
[4] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in neural information processing systems, 2016, pp. 2172–2180.
[5] I. Corona, D. Maiorca, D. Ariu, G. Giacinto, Lux0r: Detection of malicious pdf-embedded javascript code through discriminant analysis of api references, in: Proceedings of the 2014 Workshop on Artificial Intelligent and Security Workshop, ACM, 2014, pp. 47–57.
[6] G.E. Dahl, J.W. Stokes, L. Deng, D. Yu, Large-scale malware classification using random projections and neural networks, in: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, IEEE, 2013, pp. 3422–3426.

[7] N. Dalvi, P. Domingos, S. Sanghai, D. Verma, et al., Adversarial classification, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 99–108.
[8] T. Dreossi, A. Donzé, S.A. Seshia, Compositional falsification of cyber-physical systems with machine learning components, Journal of Automated Reasoning 63 (4) (2019) 1031–1053.
[9] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, D. Song, Robust physical-world attacks on deep learning visual classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1625–1634.
[10] H. Fawzi, P. Tabuada, S. Diggavi, Secure estimation and control for cyber-physical systems under adversarial attacks, IEEE Trans. Automat. Contr. 59 (6) (2014) 1454–1467.
[11] A.P. Fournaris, A.S. Lalos, D. Serpanos, Generative adversarial networks in ai-enabled safety-critical systems: friend or foe? Computer 52 (9) (2019) 78–81.
[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[13] K. Grosse, N. Papernot, P. Manoharan, M. Backes, P. McDaniel, Adversarial examples for malware detection, in: European Symposium on Research in Computer Security, Springer, 2017, pp. 62–79.
[14] Z. Guan, X. Liu, L. Wu, J. Wu, R. Xu, J. Zhang, Y. Li, Cross-lingual multi-keyword rank search with semantic extension over encrypted data, Inf Sci (Ny) 514 (2020) 523–540.
[15] Z. Guan, Y. Zhang, L. Zhu, L. Wu, S. Yu, Effect: an efficient flexible privacy-preserving data aggregation scheme with authentication in smart grid, Sci. China Inf. Sci. 62 (3) (2019) 32103.
[16] A. Hassan, R. Hamza, H. Yan, P. Li, An efficient outsourced privacy preserving machine learning scheme with public verifiability, IEEE Access 7 (2019) 146322–146330.
[17] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1026–1034.
[18] W. Hu, Y. Tan, Generating adversarial malware examples for black-box attacks based on gan, arXiv preprint arXiv:1702.05983 (2017).
[19] K.N. Junejo, J. Goh, Behaviour-based attack detection and classification in cyber physical systems using machine learning, in: Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security, ACM, 2016, pp. 34–43.
[20] B. Kolosnjaji, A. Demontis, B. Biggio, D. Maiorca, G. Giacinto, C. Eckert, F. Roli, Adversarial malware binaries: Evading deep learning for malware detection in executables, in: 2018 26th European Signal Processing Conference (EUSIPCO), IEEE, 2018, pp. 533–537.
[21] P. Laskov, et al., Practical evasion of a learning-based classifier: acase study, in: 2014 IEEE Symposium on Security and Privacy, IEEE, 2014, pp. 197–211.
[22] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al., Photo-realistic single image super-resolution using a generative adversarial network, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4681–4690.
[23] T. Li, C. Gao, L. Jiang, W. Pedrycz, J. Shen, Publicly verifiable privacy-preserving aggregation and its application in iot, J. Netw. Comput. Appl. 126 (2019) 39–44.
[24] T. Li, X. Li, X. Zhong, N. Jiang, C.-z. Gao, Communication-efficient outsourced privacy-preserving classification service using trusted processor, Inf. Sci. 505 (2019) 473–486.
[25] Y. Li, S. Yao, K. Yang, Y.-a. Tan, Q. Zhang, A high-imperceptibility and histogram-shifting data hiding scheme for jpeg images, IEEE Access (2019).
[26] X. Liu, J. Liu, S. Zhu, W. Wang, X. Zhang, Privacy risk analysis and mitigation of analytics libraries in the android ecosystem, IEEE Trans. Mob. Comput. (2019).
[27] D. Lowd, C. Meek, Adversarial learning, in: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, ACM, 2005, pp. 641–647.
[28] D. Maiorca, B. Biggio, Digital investigation of pdf files: unveiling traces of embedded malware, IEEE Secur. Priv. 17 (1) (2019) 63–71.
[29] D. Maiorca, I. Corona, G. Giacinto, Looking at the bag is not enough to find the bomb: an evasion of structural methods for malicious pdf files detection, in: Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, ACM, 2013, pp. 119–130.
[30] S.-M. Moosavi-Dezfooli, A. Fawzi, P. Frossard, Deepfool: a simple and accurate method to fool deep neural networks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 2574–2582.
[31] A. Nguyen, J. Yosinski, J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 427–436.
[32] N. Papernot, P. McDaniel, I. Goodfellow, Transferability in machine learning: from phenomena to black-box attacks using adversarial samples, arXiv preprint arXiv:1605.07277 (2016).
[33] M. Sharif, S. Bhagavatula, L. Bauer, M.K. Reiter, Accessorize to a crime: real and stealthy attacks on state-of-the-art face recognition, in: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, ACM, 2016, pp. 1528–1540.
[34] C. Smutz, A. Stavrou, Malicious pdf detection using metadata and structural features, in: Proceedings of the 28th annual computer security applications conference, ACM, 2012, pp. 239–248.
[35] C. Smutz, A. Stavrou, When a tree falls: using diversity in ensemble classifiers to identify evasion in malware detectors., NDSS, 2016.
[36] N. Srndi, P. Laskov, Mimicus: https://github.com/srndic/mimicus., Mimicus, 2014.
[37] N. Šrndic, P. Laskov, Detection of malicious pdf files based on hierarchical document structure, in: Proceedings of the 20th Annual Network & Distributed System Security Symposium, Citeseer, 2013, pp. 1–16.
[38] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, arXiv preprint arXiv:1312.6199 (2013).
[39] Y.-a. Tan, Y. Xue, C. Liang, J. Zheng, Q. Zhang, J. Zheng, Y. Li, A root privilege management scheme with revocable authorization for android devices, J. Netw. Comput. Appl. 107 (2018) 69–82.
[40] K.R. Varshney, H. Alemzadeh, On the safety of machine learning: cyber-physical systems, decision sciences, and data products, Big Data 5 (3) (2017) 246–255.
[41] W. Wang, Y. Shang, Y. He, Y. Li, J. Liu, Botmark: automated botnet detection with hybrid analysis of flow-based and graph-based traffic behaviors, Inf. Sci. 511 (2020) 284–296.
[42] W. Xu, Y. Qi, D. Evans, Automatically evading classifiers: a case study on pdf malware classifiers. ndss, 2016.
[43] Q. Zhang, Y. Gan, L. Liu, X. Wang, X. Luo, Y. Li, An authenticated asymmetric group key agreement based on attribute encryption, J. Netw. Comput. Appl. 123 (2018) 1–10.
[44] Q. Zhang, H. Gong, X. Zhang, C. Liang, Y.-a. Tan, A sensitive network jitter measurement for covert timing channels over interactive traffic, Multimed. Tools Appl. 78 (3) (2019) 3493–3509.
[45] Q. Zhang, X. Wang, J. Yuan, L. Liu, R. Wang, H. Huang, Y. Li, A hierarchical group key agreement protocol using orientable attributes for cloud computing, Inf. Sci. 480 (2019) 55–69.