MIGAN: Malware Image Synthesis Using GANs

Abhishek Singh
abhishek.s14@iiits.in
Indian Institute of Information
Technology, Sri City, AP, India, 517588

Debojyoti Dutta, Amit Saha
Cisco Systems, San Jose, California

Abstract

Majority of the advancement in Deep learning (DL) has occurred in domains such as computer vision, and natural language processing, where abundant training data is available. A major obstacle in leveraging DL techniques for malware analysis is the lack of sufficiently big, labeled datasets. In this paper, we take the first steps towards building a model which can synthesize labeled dataset of malware images using GAN. Such a model can be utilized to perform data augmentation for training a classifier. Furthermore, the model can be shared publicly for community to reap benefits of dataset without sharing the original dataset. First, we show the underlying idiosyncrasies of malware images and why existing data augmentation techniques as well as traditional GAN training fail to produce quality artificial samples. Next, we propose a new method for training GAN where we explicitly embed prior domain knowledge about the dataset into the training procedure. We show improvements in training stability and sample quality assessed on different metrics. Our experiments show substantial improvement on baselines and promise for using such a generative model for malware visualization systems.

Introduction

With recent advances in machine learning (ML) and DL in particular, there has been a surge in malware detection systems which make use of ML/DL models. Lack of publicly available labeled data sets of malware samples make it difficult to compare existing models. In this work, we propose a GAN based generative model which can ameliorate the issue. We refer our proposed GAN based setup as Malware Images GAN(MIGAN). We show its efficacy by performing data augmentation for malware images which is widely used in malware visualization systems. Traditional areas in DL such as computer vision and natural language processing have used data augmentation to enhance performance for variety of problems. Some common methods to perform data augmentation are adding noise, re-arranging portions of data, and removing small amount of information from data. For example, in natural images the conventional data augmentation method is to perform horizontal flipping and random cropping. A data set of malware images is quite different compared to natural images. Malware images of same category have strong structural correlation between them and it is this underlying structure among malware images which differentiates one category of malware from another. Hence, existing data augmentation schemes for images which are based on rotational and translation invariance of natural images do not reconcile with the idiosyncrasies of malware images. We discuss about the data set in more detail in the supplementary file.

There are other added advantages from this GAN based model of synthesizing training samples. For sensitive data, individuals/companies can make the generator, G, publicly available which might be preferable over sharing the actual data set. Also, G can be used as an alternative when transferring the whole data set from one system to other. While a data set would require a lot of bandwidth and time to get transferred across systems, the generator, being comparatively smaller in size, can instead be transferred and used to synthesize training samples. Additionally, the discriminator can be used as a baseline model since it is trained to predict the class labels for its training data.

Experiments and Analysis

For experiments, we use the data set introduced by (Nataraj et al. 2011) and is publicly available. The data set consists of total 9,458 images coming from 25 different malware families. In order to obtain malware image from a given malware binary, every byte of the binary is read sequentially as a component of a vector with each component of the vector being an unsigned 8-bit integer. This vector is then interpreted as a gray-scale image by fixing number of columns and rows. To segregate generated samples to their respective malware category without any manual intervention, we use the mechanism of class-conditional image synthesis model as described in AC-GAN (Odena, Olah, and Shlens 2016), where the generator is conditioned with class label and the discriminator is tasked to predict the class label. Not only do we obtain a labeled data set with this approach but we also observe improved quality of the synthetic samples as this approach allows the model to obtain extra information over the domain and discover more structure in the latent space of GAN. To allow the model to learn the structural correlation among the malware images of same category we add an additional task for the generator. To understand it, let us first

Copyright © 2019, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

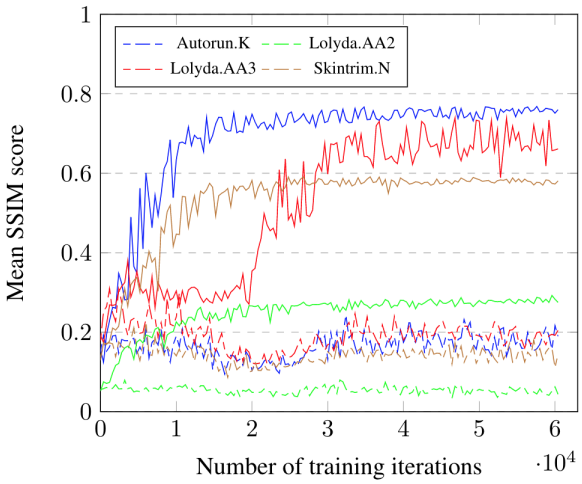

Figure 1: mean SSIM score VS Number of iterations for four different malware categories. Each color represents a malware category with dotted and solid lines representing AC-GAN and MIGAN respectively

look at the training formulation of AC-GAN (Odena, Olah, and Shlens 2016),

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

$$L_c = E[logP(C = c|X_{real})] + E[logP(C = c|X_{fake})]$$

Here, Discriminator is trained to maximize $L_c + L_s$ and Generator is trained to minimize $L_s - L_c$. We enforce the generator to learn structural correlation present in the training data set by introducing additional task for the generator to minimize $1 - SSIM(X_{real}, X_{fake})$. Overall, new objective for generator becomes to minimize $\alpha(1 - SSIM(X_{real}, X_{fake})) + (1 - \alpha)(L_s - L_c)$. Where $\alpha \in (0, 1)$ is trade-off parameter. SSIM score (Wang et al. 2004) is widely used in applications where structural similarity between two images is required to compute such as image and video coding. For practical purposes, when training with mini-batches, class labels for X_{fake} and X_{real} are sampled for same category of malware although it could be different in theory.

To assess the quality of distributions, we evaluate generated sample distribution on two different metrics, SSIM and fréchet inception distance. We also train AC-GAN (Odena, Olah, and Shlens 2016) as baseline model. Upon completion of training, we sample malware images from generator for every category and compare average SSIM for all malware categories in the training data set. We obtain average SSIM score of 0.41 on MIGAN compared to 0.17 obtained on AC-GAN. We do not obtain substantial increase in average SSIM score across all malware categories but we found SSIM to be improving consistently with training in the case of MIGAN whereas AC-GAN fluctuated around the same initial SSIM score throughout the training, figure 1 shows SSIM for four different malware categories for both GANs during the training iterations. Since SSIM score is used as

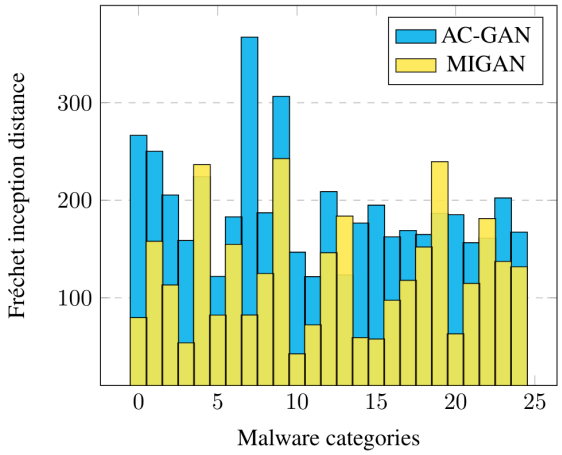

Figure 2: A comparison of AC-GAN and MIGAN based on Fréchet inception distance for all 25 malware categories.

one of the objective in the loss function formulation, it is possible that minimizing the loss would have led to improvement in SSIM score. Therefore, we evaluate generated samples also on fréchet inception distance (Heusel et al. 2017). We train MIGAN with all hyperparameters and experiment condition same as baseline except the proposed loss function. We obtain mean fréchet inception distance of 194.98 with AC-GAN which is substantially higher compared to the score of 125.05 obtained on MIGAN. Figure 2 shows FID assessed for both GANs on all malware categories.

Conclusions

In this work, we present a GAN based generative model for malware images. This work could be used to boost classifier's performance by performing data augmentation. Additionally, it can be leveraged to generate malware images which would alleviate the problem of publicly sharing the data set. Our current focus is to extend this generative model to support data augmentation for other data sets the security domain, such as binaries, intrusion detection and log files.

References

Heusel, M.; Ramsauer, H.; Unterthiner, T.; Nessler, B.; Klambauer, G.; and Hochreiter, S. 2017. Gans trained by a two time-scale update rule converge to a nash equilibrium. *CoRR* abs/1706.08500.

Nataraj, L.; Karthikeyan, S.; Jacob, G.; and Manjunath, B. S. 2011. Malware images: Visualization and automatic classification. In *Proceedings of the 8th International Symposium on Visualization for Cyber Security*, VizSec '11, 4:1–4:7. New York, NY, USA: ACM.

Odena, A.; Olah, C.; and Shlens, J. 2016. Conditional Image Synthesis With Auxiliary Classifier GANs. *ArXiv e-prints*.

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing* 13(4):600–612.