International Journal of Advanced Smart Convergence Vol.10 No.3 163-171 (2021)
http://dx.doi.org/10.7236/IJASC.2021.10.3.163

IJASC 21-3-21

BM3D and Deep Image Prior based Denoising for the Defense against Adversarial Attacks on Malware Detection Networks

Kumi Sandra[†] and Suk-Ho Lee[††]

[†]*Researcher, Department of Computer Engineering, Dongseo University, Korea*
[††]*Professor, Department of Computer Engineering, Dongseo University, Korea*
E-mail petrasuk@gmail.com

Abstract

Recently, Machine Learning-based visualization approaches have been proposed to combat the problem of malware detection. Unfortunately, these techniques are exposed to Adversarial examples. Adversarial examples are noises which can deceive the deep learning based malware detection network such that the malware becomes unrecognizable. To address the shortcomings of these approaches, we present Block-matching and 3D filtering (BM3D) algorithm and deep image prior based denoising technique to defend against adversarial examples on visualization-based malware detection systems. The BM3D based denoising method eliminates most of the adversarial noise. After that the deep image prior based denoising removes the remaining subtle noise. Experimental results on the MS BIG malware dataset and benign samples show that the proposed denoising based defense recovers the performance of the adversarial attacked CNN model for malware detection to some extent.

Keywords: *Malware detection, Deep Learning, Adversarial Examples, Denoising, BM3D, Deep Image Prior*

1. Introduction

The exponential development of the Internet, smart homes and mobiles has increased the number of malwares rapidly. More than 350,000 new malicious software and Potentially Unwanted Applications (PUA) are detected every day by the AV-TEST Institute[1]. According to Malwarebytes Labs 2020 state of malware report[2], business malware detection increased by 13% from 2018 to 2019. Malware is a type of computer program specially crafted to take control of computer systems. Malware attempts to steal sensitive information, money, and interrupt the services and operations of victims. There is a lot of malware including ransomware, viruses, trojans, scareware, worms, spyware, adware, and fileless malware[3]. Malware is disseminated through email attachments, malvertising, infected USB drives, and applications. Attackers have also invented sophisticated techniques for distributing malware by leveraging Microsoft Office and script-based threats[4]. Attackers mainly distribute malware to organizations to receive a large payment of money. The average cost for organizations to remediate a ransomware attack is US$761,106[5].

Over the years, signatures[6], static[7], and dynamic analysis[8] approaches are being used for malware

Manuscript Received: August. 12, 2021 / Revised: August. 17, 2021 / Accepted: August. 20, 2021
Corresponding Author: petrasuk@gmail.com
Tel: +82-51-320-1744, Fax: +82-51-327-8955
Professor, Department of Computer Engineering, Dongseo University, Korea

detection. In signature-based detection, signatures are generated from extracted features of a binary file and matched against a malware database to detect malware. This approach is ineffective to detect self-propagating malware and intelligent malware components[9]. Static analysis approach analyses malware without running it but it is resilient against evasion techniques[10]. Dynamic analysis addresses code obfuscation by analyzing the runtime behavior of the program in an execution mode, but it is time-consuming[11].

To improve malware defenses, researchers have leveraged Artificial Intelligence to help in the detection of sophisticated and zero-day malware attacks. In recent years, image-based approaches have been proposed to simplify the detection of malware while reducing time and memory consumption. Several studies have utilized image-based malware detection with deep learning to improve the detection speed and accuracy of malware detection systems[12][13][14][15].

Despite, the excellent performance of deep learning in image recognition[16][17][18], neural networks are vulnerable to adversarial examples. For example, it has been shown in [19] that a small adversarial noise can lead the convolutional neural network to misrecognize an object. When used as a mean of attack, the adversarial noise can attack a malware detection neural network so that it can no longer detect the malware. Therefore, in this paper, we propose a denoising based defense method that can defense the adversarial attack on malware detection neural networks to some extent. The denoising is done in two steps, where the first step denoises the large noise and the second step the remaining subtle noise. Experimental results verify the validness of the proposed method.

2. Related Works

In this section, we present Machine Learning (ML) algorithms for the detection of malware and adversarial attacks against ML-based malware detection methods. Tobiyama et al[12] proposed a malware detection method with Deep Neural Network based on process behaviour. The authors used Recurrent Neural Network (RNN) for feature extraction and Convolutional Neural Network (CNN) for feature classification, achieving Area under the curve (AUC) of 96%. Mohaisen et al[13] introduced an automated and behavior-based malware analysis, AMAL which consists of two sub-systems, i.e., AutoMal and MaLabel, where AutoMal collects low granularity behavioral artifacts of malware and MaLabel creates representative features to label malware using classification and clustering algorithms. The shortcomings of this work are the high overhead incurred when executing malware samples in virtual environment and the capability of malware to evade the analysis environment.

ML-based visualization malware detection methods are currently used by researchers to reduce time and memory consumption. Malware visualization is the method of transforming malware binaries into images. Nataraj et al[14] presented malware classification based on image processing techniques. They visualized malware binary as a gray scale image and extracted texture features from images using GIST. A k-nearest neighbor with Euclidean distance was used for classification, achieving an accuracy of 98%. Kalash et al[15] also coverted malware binaries into grayscale images and trained CNN model for classification. Their approached achieved 98.5% and 99.97% accuracy on the Malimg and Microsoft datasets respectively.

However Machine Learning (ML) based malware visualization detection methods are vulnerable to AEs. Liu et al.[20] proposed pertubation attack method based on gradient descent and L-norm optimization method to bypass the detection malware. Using a distortion rate of 0.35, their method was able to deceive several ML-based detection methods, reducing the accurate rate to 0% and a transferability rate of 74.1%.

3. Adversarial Attack on Malware detecting Convolutional Neural Network

A malware detecting CNN(convolutional neural network) consist of a visualization component which converts the malware code into a binary image. After the visualization process, the binary images corresponding to the benign and malware category are classified by the neural network. The neural network then can discriminate between the benign and malware data. However, if we apply the adversarial attack on the binary images, the malware detecting CNN fails to correctly classify between the malware and the benign data. In this study, we assume the adversary has prior knowledge and access to the model architecture (White-box attack) i.e. training data, hyper-parameters, activation functions, number of layers, etc. The experiments in our work focuses on untargeted attacks, i.e. generated adversarial examples can be labeled benign or malware.

3.1 Visualization

In the **Visualization Process** as shown in Figure 2, we read benign and malware samples as 8-bit unsigned integers and transform into a two-dimensional array. Each value in the array is converted into a grayscale image in the range of 0 to 255. All binary files are transformed into grayscale images with a fixed width but the height depends on the file size.

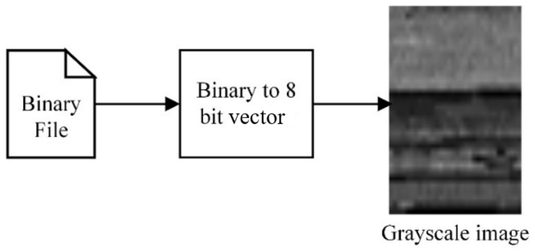

Figure 1. Binary File Visualization Process

3.2 ML-based malware visualization detection approach

For the experiments, we designed a Convolutional Neural Network (CNN) architecture to classify visualized benign and malware samples. All images were resized to 224 x 224 as input to the CNN model. Our CNN model consists of two convolutional and max pooling layers to extract features from images. The rectified linear activation function (ReLU) is used as activation function except the last layer. We apply the softmax activation function to the last dense layer to normalize the outputs.

3.3 Generation of Adversarial Examples

Adversarial Examples (AEs) are specially crafted inputs formed by adding small pertubations (noise) to original inputs, with the intention of confusing a neural network. The Fast Gradient Sign Method[19]

(FGSM) is employed to generate AEs for the adversarial attack. The FGSM uses the one step perturbation, hence is very fast in generating AEs. As shown in Figure 1, the **Adversarial Sample Generator** uses the pre-trained CNN model and a set of grayscale scale images, then applies the FGSM to obtain adversarial examples.

The FGSM uses the gradient of the CNN model to create adversarial examples. The FGSM is defined as:

$$x_{adv} = x + \varepsilon sign\left(\nabla_x J(\theta, x, y)\right) \tag{1}$$

where x_{adv} is the adversarial image, x is the original image, y is the true label of the original image, ε is the small scalar value, J is the cost function (Loss) used to train the neural network and θ is the parameters of the neural network. Figure 3 shows the original, the adversarial, and the binarized difference images of the visualized binary files corresponding to the malware codes.

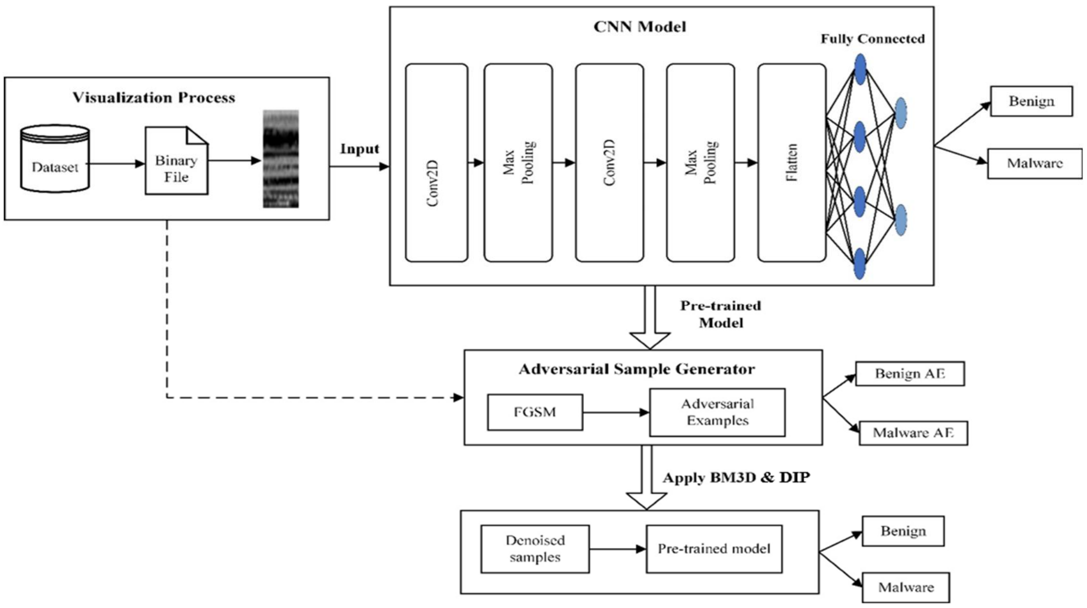

Figure 2. Experimental Setup for the Adversarial Attack and Defense on the Malware Detecting System

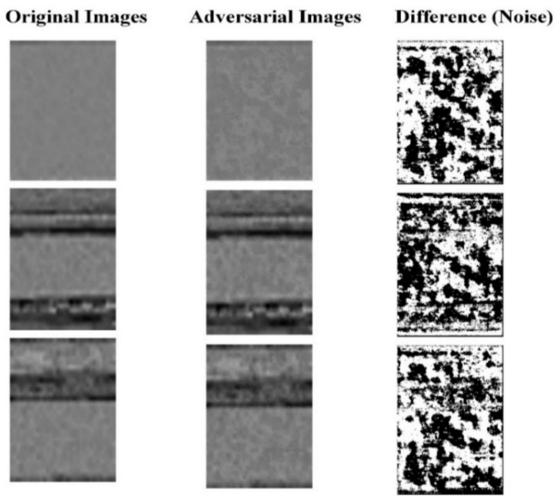

Figure 3. Showing the original, the adversarial, and the binarized difference images of the visualized malware codes.

4. Denoising based Adversarial Defense for Malware Detection Neural Network

In this section, we describe the proposed system for the defense of the adversarial attack on the malware detecting neural network. The system diagram is drawn in Fig. 4. The system includes the binarization process and visualization of the incoming code with/without adversarial noise. After the visualization of the binary file, the visual image goes into the input of the adversarial noise removal pre-process block. The pre-process block should be strong enough to remove the adversarial noise in the visualized image, but should preserve enough content for the prediction between benign and malware. In this paper, we use successive denoising methods, where we use the BM3D (Block matching based 3 dimensional) filter method [23] to eliminate the strong noise and the DIP(Deep Image Prior) network [24] to eliminate the fine noise.

The BM3D filter method is one of the denoising methods that has recently been evaluated as state-of-the-art. The method is to collect and group similar regions from the same image, then construct a three-dimensional array with these areas, and finally shrinkage in the transform-domain to eliminate the noise. This method is based on the assumption that when a signal is converted to 3D transform-domain, the signal has a sparse characteristic of particular components in the transform-domain. The procedure of the BM3D is performed as follows:

1. Using some kind of similarity measure a local patches inside a surrounding area of the region of interest, areas similar to the one of interest are obtained.
2. The areas with high similarity are stacked to create a 3D arrangement.
3. Apply to each area an orthogonal transform, such as the Haar wavelet transform, in the axial direction of the same pixel.
4. Apply a shrinkage, like a hard-threshold or an empirical Wiener filtering on the 3D spectrum.
5. Return the filtered 3D spectrum back to its original form.
6. The returned results of 5 are the noise-removed images.

After the process with the BM3D, the large noise in the binarized image will have been disappeared. However, some of the subtle noise can remain in the image. Even though this noise is subtle, it can have a large effect on the classification, since the network can amplify the adversarial noise, which is a common phenomenon with adversarial noise. Therefore, we further apply the deep image prior (DIP) network to remove the subtle adversarial noise in the BM3D processed image. The DIP is a type of convolutional neural network which resembles an auto encoder, but it is trained only on the single noisy image. Here, we define by x the adversarial noisy image, and by $\mathrm{BM3D}(\cdot)$ an $DIP_\theta(\cdot)$, the outputs of the BM3D process function and the DIP network, respectively. The DIP converts a 3D noise tensor z into an approximate of the target image $\mathrm{BM3D(x)}$ by minimizing the following loss function with respect to the parameters θ of the DIP network:

$$\mathrm{L} = \|DIP_\theta(\mathrm{z}) - BM3D(x)\|_2^2$$

The result image $DIP_\theta(\mathrm{z})$ is a denoised version of $BM3D(x)$ where the remaining subtle noise in $BM3D(x)$ is removed. Figure 4 shows the overall malware detecting system with the proposed adversarial noise removal process incorporated. Figure 5 shows the original, the denoised, and the binarized difference images of the visualized binary files corresponding to the malware code. It can be seen that the noise is much removed by the proposed noise removal method.

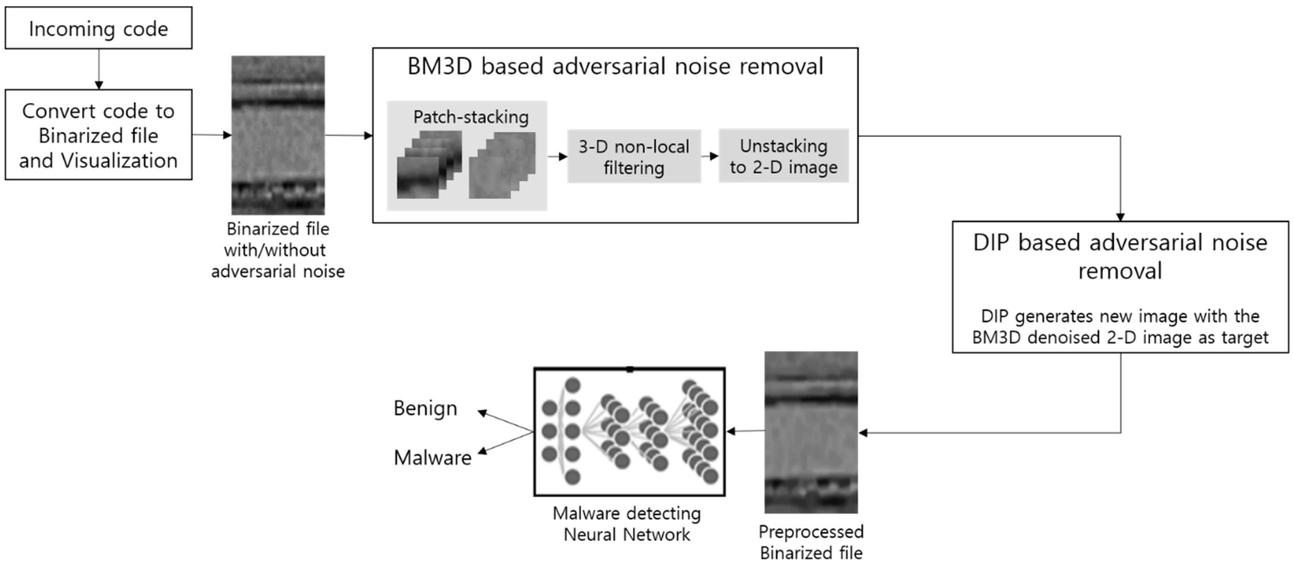
Incoming code
Convert code to Binarized file and Visualization
Binarized file with/without adversarial noise
BM3D based adversarial noise removal
Patch-stacking
3-D non-local filtering
Unstacking to 2-D image
DIP based adversarial noise removal
DIP generates new image with the BM3D denoised 2-D image as target
Benign
Malware
Malware detecting Neural Network
Preprocessed Binarized file

Figure 4. Overall malware detecting system with the proposed adversarial noise removal process

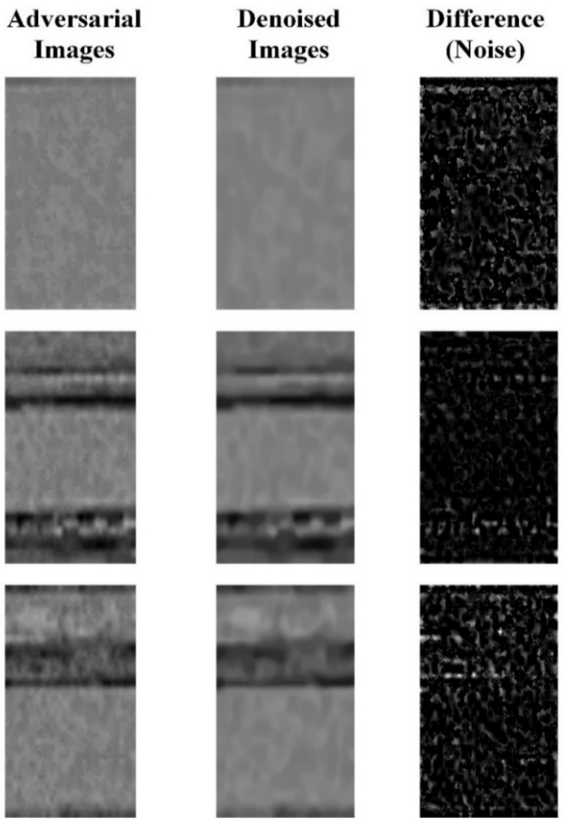

Figure 5. Showing the original, the denoised, and the binarized difference images of the visualized malware codes.

5. Experiments

5.1 Dataset

A dataset of benign and malware binary samples was collected to evaluate our proposed approach. We used a labeled open source malware dataset from Kaggle Microsoft Malware Classification Challenge[21], which is classified into nine classes. We obtained 842 benign samples from MS Windows 10. The distribution of dataset is shown in Table 1.

Table 1. Distribution of Dataset

Type		Number
Malware	Ramnit	1534
	Lollipop	2470
	Kelihos ver3	2942
	Vundo	451
	Simda	41
	Tracur	685
	Kelihos ver1	386
	Obfuscator.ACY	1158
	Gatak	1011
Benign		842

5.2 Experimental Setup

We implemented our proposed approach using Python (CNN model, Deep Image Prior Network, and Adversarial Examples) and Matlab (BM3D). The CNN model and the Deep Image Prior network were implemented with Keras and Tensorflow 1.15. Adversarial examples were generated using the Adversarial Robustness Toolbox (ART)[22]. Experiments were conducted 64-bit Window 10 desktop operating on Intel® Core™ i7-6700K CPU at 4.00GHz with 16GB RAM.

We selected the following parameters in building the CNN model; Stochastic Gradient Descent (SGD), binary cross-entropy loss function and learning rate of 0.01. During training, we set the maximum epoch to 50 with a batch size of 100. Adversarial examples were generated using an epsilon rate of 0.02.

5.3 Results and Discussion

The proposed approach is evaluated using the following performance metrics:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F - score = 2 \times \frac{precision \times recall}{precision + recall}$$

$$False\ Positive\ Rate = \frac{FP}{FP + TN}$$

where TP (True Positive) is the correct malware prediction, TN (True Negative) is the correct benign prediction, FP (False Positive) is the incorrect malware prediction and FN (False Negative) is the incorrect benign prediction. It can be seen from Table 2, that the proposed system can improve the Recall measure by 21% and the F-score measure by 19%, which validates the fact that the proposed system becomes more robust against the adversarial noise than the system without the proposed denoising method.

Table 2. Experimental Results on three classes of Dataset

Dataset	Precision (%)	Recall (%)	F-score(%)	False Positive Rate (%)
Original	98	97	97	1
Adversarial	81	41	52	11
Denoised	84	62	71	8.46

6. Conclusion

We proposed a denoising based defense method for the defense against adversarial attacks on malware detection neural networks. The proposed method set its target for malware detection systems which visualize the malware code as images and use these images to train a convolutional neural network which can classify between benign and malware codes. We set up the whole experimental environment where the adversarial example can deceive this malware detection network using an adversarial noise. Then, for the removal of the adversarial example, we proposed a denoising based adversarial noise removing method, where the first step in the denoising uses the BM3D method to remove the large noise and the second step uses the deep image prior network to remove the remaining subtle noise. We made experiments with the original, the adversarial attacked, and the denoised datasets and verified that the proposed defense method can resist the adversarial attack to some extent despite the difficulty of denoising the visualized malware code as it is different from normal images containing many high frequency components.

Acknowledgement

This work was supported by Dongseo University, “Dongseo Cluster Project” Research Fund of 2021 (DSU-20210001).

References

[1] “The AV-TEST Institute.” https://www.av-test.org/en/statistics/malware/ (accessed Nov. 30, 2020).
[2] Malwarebytes Labs, “2020 State of Malware Report.” https://resources.malwarebytes.com/files/2020/02/2020_State-of-Malware-Report.pdf (accessed Nov. 30, 2020).
[3] “What is malware and why do cybercriminals use malware?” https://www.mcafee.com/en-

us/antivirus/malware.html (accessed Nov. 30, 2020).

[4] Avira Protection Lab, “Malware Threat Report: Q2 2020 Statistics and Trends.” https://www.avira.com/en/blog/malware-threat-report-q2-2020-statistics-and-trends (accessed Nov. 30, 2020).

[5] “THE STATE OF RANSOMWARE 2020.” https://secure2.sophos.com/en-us/content/state-of-ransomware.aspx (accessed Nov. 30, 2020).

[6] P. Faruki, V. Ganmoor, V. Laxmi, M. S. Gaur, and A. Bharmal, “AndroSimilar: Robust statistical feature signature for android malware detection, ” in *Proc. 6th International Conference on Security of Information and Networks*, pp. 152–159, Nov. 26-28, 2013, https://doi.org/10.1145/2523514.2523539.

[7] M. Christodorescu and S. Jha, “Static analysis of executables to detect malicious patterns,” in *Proc. 12th conference on USENIX Security Symposium*, pp. 12–13, Aug. 4-6, 2003,

[8] M. Egele, T. Scholte, E. Kirda, and C. Kruegel, “A survey on automated dynamic malware-analysis techniques and tools,” *ACM Computing Surveys*, Vol.44,No.2,pp. 1-42, Feb. 2012, https://doi.org/10.1145/2089125.2089126.

[9] J. Scott, “Signature Based Malware Detection is Dead,” *Cybersecurity Think Tank, Inst. Crit. Infrastruct. Technol.*, no. February, 2017.

[10] A. Moser, C. Kruegel, and E. Kirda, “Limits of static analysis for malware detection,” in *Proc. 23rd Annual Computer Security Applications Conference*, Dec. 10-14, 2007, https://doi.org/10.1109/ACSAC.2007.21

[11] P. V. Shijo and A. Salim, “Integrated static and dynamic analysis for malware detection,” *Procedia Computer Science*, Vol. 46, pp. 804-811, Dec. 2015, https://doi.org/10.1016/j.procs.2015.02.149.

[12] S. Tobiyama, Y. Yamaguchi, H. Shimada, T. Ikuse, and T. Yagi, “Malware Detection with Deep Neural Network Using Process Behavior,” in *Proc. IEEE 40th Annual Computer Software and Applications Conference*, June 10-14, Vol. 2, 2016, https://doi.org/10.1109/COMPSAC.2016.151.

[13] A. Mohaisen, O. Alrawi, and M. Mohaisen, “AMAL: High-fidelity, behavior-based automated malware analysis and classification,” *Comput. Secur.*, Vol. 52, 2015, https://doi.org/10.1016/j.cose.2015.04.001.

[14] L. Nataraj, S. Karthikeyan, G. Jacob, and B. S. Manjunath, “Malware images: Visualization and automatic classification,” *Proc. 8th International Symposium on Visualization for Cyber Security*, pp.1-7, July 20, 2011, https://doi.org/10.1145/2016904.2016908.

[15] M. Kalash, M. Rochan, N. Mohammed, N. D. B. Bruce, Y. Wang, and F. Iqbal, “Malware Classification with Deep Convolutional Neural Networks,” in *Proc. 2018 9th IFIP International Conference on New Technologies, Mobility and Security,* Feb. 26-28, 2018, https://doi.org/10.1109/NTMS.2018.8328749.

[16] J. Lee and S.-J. Shin, “A Study of Video-Based Abnormal Behavior Recognition Model Using Deep Learning,” *International journal of advanced smart convergence*, Vol. 9, No. 4, pp. 115–119, Dec. 2020. https://doi.org/10.7236/IJASC.2020.9.4.115

[17] B. Kim and J. Heo, “Semi-Supervised Learning Based Anomaly Detection for License Plate OCR in Real Time Video,” *International journal of advanced smart convergence*, Vol. 9, No. 1, pp. 113–120, Mar. 2020. https://doi.org/10.7236/IJASC.2020.9.1.113

[18] Y. Lee and J. Shim, “Deep Learning and Color Histogram based Fire and Smoke Detection Research,” *International journal of advanced smart convergence*, Vol. 8, No. 2, pp. 116–125, Jun. 2019. https://doi.org/10.7236/IJASC.2019.8.2.116

[19] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *Proc. 3rd International Conference on Learning Representations (ICLR)*, May 7-9, 2015.

[20] X. Liu, J. Zhang, Y. Lin, and H. Li, “ATMPA: Attacking machine learning-based malware visualization detection methods via adversarial examples,” in *Proc. 2019 IEEE/ACM 27th International Symposium on Quality of Service*, June 24-25, 2019, https://doi.org/10.1145/3326285.3329073.

[21] R. Ronen, M. Radu, C. Feuerstein, E. Yom-Tov, and M. Ahmadi, “Microsoft Malware Classification Challenge,” *CoRR*, vol. abs/1802.1, 2018, [Online]. Available: https://arxiv.org/abs/1802.10135.

[22] M.-I. Nicolae *et al.*, “Adversarial Robustness Toolbox v1.0.0,” *arXiv*, 2018, [Online]. Available: https://arxiv.org/abs/1807.01069.

[23] K. Dabov, A. Foi; V. Katkovnik, K. Egiazarian, “Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering,” *IEEE Trans. on Image Processing*, Vol. 16, No. 8, pp. 2080 - 2095, Aug. 2007. https://doi.org/10.1109/TIP.2007.901238

[24] D. Ulyanov, A. Vedaldi, V. Lempitsky, “Deep Image Prior, ” in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9446-9454, June 18-22, 2018. https://doi.org/10.1007/s11263-020-01303-4