Invisible Encoded Backdoor attack on DNNs using Conditional GAN

Iram Arshad, Yuansong Qiao, Brian Lee, Yuhang Ye
Software Research Institute
Technological University of Shannon: Midland Midwest
Westmeath, Athlone, Ireland
i.arshad@research.ait.ie, ysqiao@research.ait.ie, blee@ait.ie, yye@research.ait.ie

***Abstract*—Deep Learning (DL) models deliver superior performance and have achieved remarkable results for classification and vision tasks. However, recent research focuses on exploring these Deep Neural Networks (DNNs) weaknesses as these can be vulnerable due to transfer learning and outsourced training data. This paper investigates the feasibility of generating a stealthy invisible backdoor attack during the training phase of deep learning models. For developing the poison dataset, an interpolation technique is used to corrupt the sub-feature space of the conditional generative adversarial network. Then, the generated poison dataset is mixed with the clean dataset to corrupt the training images dataset. The experiment results show that by injecting a 3% poison dataset combined with the clean dataset, the DL models can effectively fool with a high degree of model accuracy.**

***Index Terms*—Backdoor Attack, Conditional Generative Adversarial Network, Image Synthesis**

I. Introduction

Deep Learning (DL) models have gained popularity over Machine Learning (ML) models in recent years. These DL models significantly outperform ML models in various domains such as face recognition, natural language processing, automatic speech recognition, self-driving, and robotics [1].

Due to the high sparsity of large DL models, the malware "knowledge" can be silently added in the model without being noticed by other users. The malware knowledge misleads DL models to make wrong predictions when specific inputs are fed to the model. This is also known as backdoor attacks on DL models.

These backdoor attacks arise from outsourcing training data and transfer learning [2]. These are more attractive and vulnerable targets for attackers to manipulate DL models. Outsource training data uses cloud platforms (e.g., Google cloud) to train the data. Transfer learning is a concept of using well-tuned, pre-trained models available on public repositories to perform a similar task. Since it is challenging to ensure that the collected data is from a reliable source, an attacker can generate the poison dataset and leave it on the web for the victim to use and download for training and testing. Furthermore, these pre-trained models are hosted and maintained on popular third-party platforms, such as GitHub, where ensuring that attackers do not modify these pre-trained models is often lacking. Therefore, before deploying these DL models in safety and security-critical applications, the robustness against the variants of backdoor attacks needs to be considered. For example, backdoor attacks are malware's where the attacker generates a poison dataset based on the decided trigger. This poison dataset provides to the victims' model during training or by changing the model parameters [2], [3]. The tempered DL models will produce correct results on clean samples, so the victim will not realize that the model is compromised.

Existing backdoor attacks on DL models may use two approaches. Visible or invisible triggers and clean or unclean labels. Visible triggers mean adding visually unnatural patterns to the images; the model learns a bias on these patterns to mislead the model's output once the patterns are presented. The unclean label means the visual content of an image does not match the given label e.g., an image of digit 9 is labelled 7. The early backdoor attacks prefers both visual trigger and unclean labels. However, visible triggers along with unclean labels can easily be detected before training, e.g., via human inspection or template matching. Therefore, more recent researchers have started investigating the possibility of injecting less-visible triggers [1].

Earlier research study [8] first proposed backdoor attack with clean-label by adding ramp and sinusoidal gradients as triggers into DNNs. However, the gradient can be recovered, detected and even mitigated using blind image separation techniques. Another clean-label attack was developed in the research study [9], which leveraged generative models and adversarial perturbations to modify benign images from the target class and then conducted the standard backdoor attack. One major disadvantage of the generative model technique followed in the study [9] is that since they are using GAN feature space to generate poison images, there is no control over the modes of the data to be generated. The resulting images are of low quality. They have reported the attack success rate is above 50% only under strong attack assumptions (i.e., attacked model and test models are known). Another research research study [10] also developed a feature collision method to manipulate the model's decision for both transfer learning and end-to-end training. More recently, the research studies

Acknowledgement This publication has emanated from research conducted with the financial support of Technological University of Shannon: Midland Midwest under President Doctoral Scholarship.

[11] and [12] developed clean-label backdoor by proposed a method to conceal the trigger through image-scaling for images recognition models . However, these studies are often model-centric and are suffered from low attack rate, high injection rate, and low trigger stealthiness [1].

In this paper, we propose a novel invisible encoded backdoor attack on data during the training pipeline which is agnostic to model. Our intuition is that by conditioning the generative model based on the label information, it is possible to direct the data generation process. In addition, we focus on generating invisible triggers and clean-label strategies. The generated poison dataset are good in quality and contains concealed stealthy triggers aims to deceive the state-of-art visible backdoor defence solutions [5], [6]. The existing defence methods detect and cleanse "suspicious" data that may contain a visible backdoor. However, it is difficult to say that these defence solutions can mitigate all variants of backdoor attacks. Therefore, this paper aims to generate and investigate the feasibility of the stealthy invisible variant of backdoor. In summary, the following points highlight the main contribution:

- We generate an invisible and clean-label backdoor attack with a higher success rate and less data injection requirement.
- We use a structural metric that is i.e., preceptional hash (pHash) to measure the quality of the generated image and ensure the stealthiness of images.
- Experimental results demonstrate the effectiveness of proposed method leads to high Attack Success rate without reducing model accuracy.

The remainder of the paper is organized as follows. Section II presents the proposed methodology with pertinent details. Section III contains an experimental setup and discussion on the results. Section IV concludes the paper along with the limitations of the presented work and scope for future improvements.

II. Methodology

A. Threat Model

1) Attacker Capabilities: Some existing studies assumptions [3], where the attacker has complete knowledge of the training dataset. Unlike them, we assume that the attacker may not need to know about the training dataset. For example, the attacker could upload the corrupted model to public access repository that offer pre-trained model for download and usage under standard open source licences. Furthermore, we are making an other realistic assumption that once the backdoor injects into the training pipeline, the attacker has no further control over the model's training process.

2) Attacker Goals: A list of attacker goals are considered as shown below, which guide the implementation of a robust and effective backdoor attack.

1. A poisoned image should have a "consistent" label otherwise i.e. image cat → label dog, the inconsistent labels can be easily filtered via human-inspection or computer-vision methods.
2. Triggers should be stealthy and template-less. For example, adding sun-glasses and patching stickers are well-known triggers [1] and can be blacked-listed during training.
3. The poisoned DNN should have minimum or no impact on classification accuracy on benign images.
4. The proportion of poison images to be injected into the training dataset should be small.
5. The poisoned DNN should have a high chance of classifying a poisoned image to a wrong class.

Algorithm 1 Poison Sample Generation from Latent Space

Variables: Latent vector z, Class label c Image x, generator $CGAN$, PHash, step size η
Output: Poison Images P_A
Function `InverseGAN(`x_n`)`**:**
  for *i in range(0,N)* **do**
    $\hat{x} \leftarrow CGAN(z_n)$
    compute loss $\mathrm{L}_{[n]} := L||x_n, \hat{x}||_2$
    compute gradients $\Delta z := -(\delta L/\delta z)$
    update parameters $\delta z := z + \eta * \Delta z$
  end
  until convergence
  return z_n
End Function
$z_n \leftarrow InverseGAN(x_n)$
$x_{nImg} \leftarrow cGAN(z_0 \times \alpha + z_1 \times (1-\alpha))$
$imageQuality \leftarrow pHash(\mathrm{x}, \mathrm{x}_{nImg})$
▷ This alpha values varies based on given spatial Image

B. Intuition

$CGAN$ is used instead of unconditional GAN to generate poisoned images. The rationale behind is that $CGAN$ not only provides refined control over images but also refined control over class information. Specifically, in the unconditional GAN latent space, the images that coincidentally fall into a cluster may belong to different classes. This causes the (latent space) interpolation of two images unstable. It is difficult to synthesis "an image that looks like a dog but contains a lot of information of the cat". The proposed backdoor conditional generative classifier shows in Figure 1. This paper explores the capability of $CGAN$ to mitigate this problem by explicitly specifying the image class when generating images. The algorithm presented in Algorithm 1 to generate the desired poisoned dataset.

C. Algorithm Details

The core of poison image synthesis is a "converter" that allows mapping an given image (x) to its latent representation (z) and recovering the image from it's (z). The $CGAN$ generative neural network [13] $\mathcal{M}$ is trained offline.

To generate a poisoned image $\tilde{x}$ using $\mathcal{M}$, a pair of images $x^{(u)}, x^{(v)}$ are required, $x^{(u)}$ from the victim class and $x^{(v)}$ from the target classes. Then, the latent vectors $z^{(u)}, z^{(v)}$

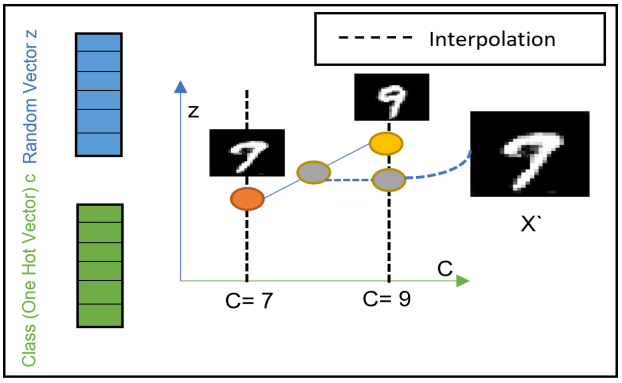


Fig. 1. The intuition behind find the random noise and class vector for given image and interpolation of two different classes by auxiliary information label y.

of the two images are obtained via inversing $CGAN$ i.e., $\mathcal{M}^{-1} : x^{(u)}, x^{(v)} \rightarrow z^{(u)}, z^{(v)}$. A more practical way to implement the inversion is for each image x, optimise $\hat{z}$ so that the image space euclidean distance Mean Square Error (MSE) $|x - \mathcal{M}(\hat{z})|_2^2$ is minimised. In this paper, an Adam optimizer [14] is used to search for the solution $\hat{z}$ with step size $\eta = 0.0002$. The two latent vectors $\hat{z}^{(u)}$ and $\hat{z}^{(v)}$ are then merged into one latent vector $\tilde{z}$ via linear interpolation $\tilde{z} = \alpha\hat{z}^{(u)} + (1-\alpha)\hat{z}^{(v)}$. Using Modified National Institute of Standards and Technology dataset (Mnist) handwriting digits as an example, the first latent vector $\hat{z}^{(u)}$ is obtained from a digit-7 image $x^{(u)}$ with label $y^{(u)} = 7$ and the second latent vector $\hat{z}^{(v)}$ can be calculated based on a digit-7 image $x^{(v)}$ label and $y^{(v)} = 9$. We reproduced the clean-label attack [9] generative method with original settings. We identified that their generated images are less likely to correlate with the perceptional category i.e., they are not always clean-label. In Figure 2, it shows the difference between the poisoned images generated by the previous clean label generative method attack [9] by using visible triggers proposed by Badnet [3] and our attack where for any given images by inverting a cgan we can find the z and c value in the latent space of that given image and interpolate two different images and generate a poison dataset. In the end, we assign the labels to the generated images that are similar in the image space.

D. Dataset Injection

Once attacker generate the poison dataset D_A based on the above-mentioned method this tiny portion of poison dataset mixes with the clean dataset $D_M = D_N \cup D_A$ during the training pipeline. This D_M will be in DNNs training in the consideration of the attacker's capabilities that the other functionality should not be affected by mixed the poison dataset. However, the existence of poisoned dataset leads to the following loss function:

$$\min_{\theta} \sum_{u,v=0,x^{u\prime},y^{v\prime} \in DA}^{n} l(\theta, (x^u + x^{u'}, y^v + y^{v'})) \quad (1)$$

In eq. 1, l denotes the cross-entropy loss, θ is model parameters and $(x^u, y^v) \in X$ and $(x^{u'}, y^{v'}) \in X'$. The summary

TABLE I
A SUMMARY OF NOTATIONS.

Name	Symbol	Description
Model	$\mathcal{M}$	CGAN.
Inverse Model	$\mathcal{M}^{-1}$	CGAN^{-1}.
Latent space	$\mathcal{Z}$	Let $\mathcal{Z}$ represents a latent space.
Images	$\mathcal{X}$	Let $\mathcal{X}$ represents a set of images (x^1, $x^2 \dots x^n$).
Poison set	$\mathcal{X}'$	Let $\mathcal{X}$ represents a set of images (x^1, $x^2 \dots x^n$).
Training set	$\mathcal{X}^{\mathcal{U},\mathcal{V}}$	Let $x^{U,V}$ represents a set of images.
poisoned set	$X^{U',V'}$	Augmented ambiguous images misclassified the targeted class.
Label	$\mathcal{Y}$	$\mathcal{Y}$: Dimension of a label y^i.
Classes	$\mathcal{C}$	$\mathcal{C}$ represents a class $\{0, 1, 2, 3, \dots 9\}$ and $\{cat, dog\}$.
Training dataset	D_N	Training dataset.
Poisoned dataset	D_A	poisoned dataset after injecting the errors.
Final dataset	D_M	Union of training and poisoned dataset.

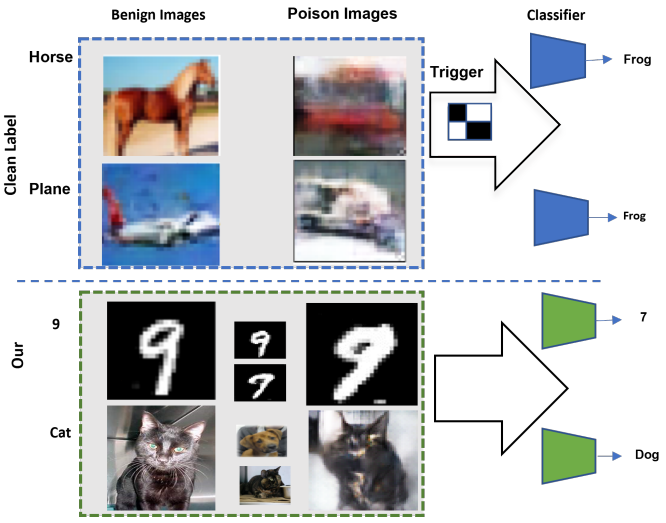


Fig. 2. The comparison of the triggers in the previous attack (e.g., clean label [9]) and in our proposed attack. The trigger of the previous attack contains a visible trigger, while in our attack the triggers are encoded in the generated images.

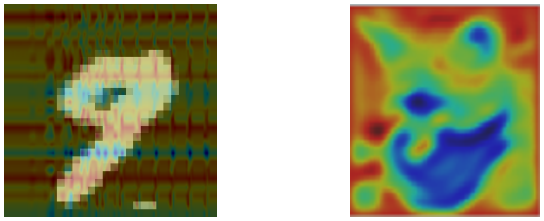
of notations describe in Table I.

III. Experiment Setup

In this section, we comprehensively describe our experiment setup to produce invisible encoded attack and tested on DL models.

A. A Deep Network Model

To demonstrate the experimental setup, we have selected a efficient Convolutional Neural Network (CNN) basic architecture [15] and pre-trained VGG16 [16] as a baseline without altering the original convolutional layers for Image classifications. Further, we have used open-source CGAN for generating poison dataset. The experiments are performed using Pytorch framework.



(a) (b)

Fig. 3. In the figure (a) Grad-CAM Visualization of image 9 which is successfully mispredict to class 7 at testing time. Figure (b) Grad-CAM Visualization of image cat successfully mispredict it to dog at testing time.

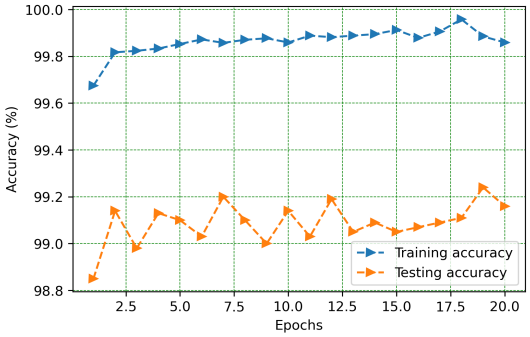


Fig. 4. Mnist poison model accuracy with trigger dataset.

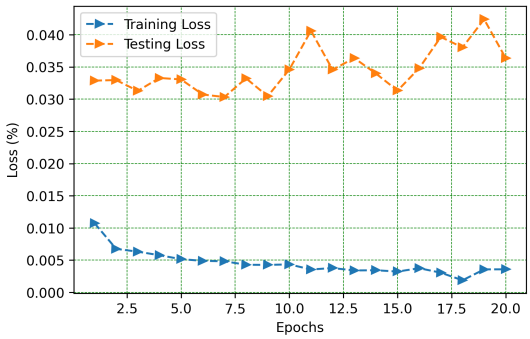


Fig. 5. Mnist poison model loss with trigger dataset.

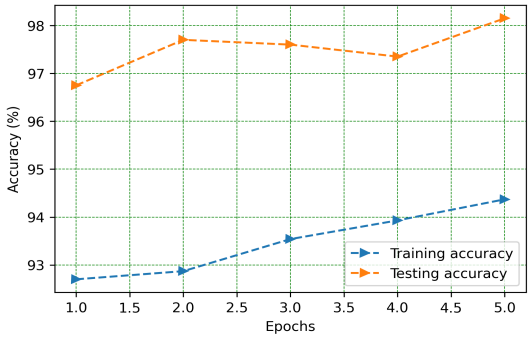


Fig. 6. Custom dataset poison model accuracy with trigger dataset.

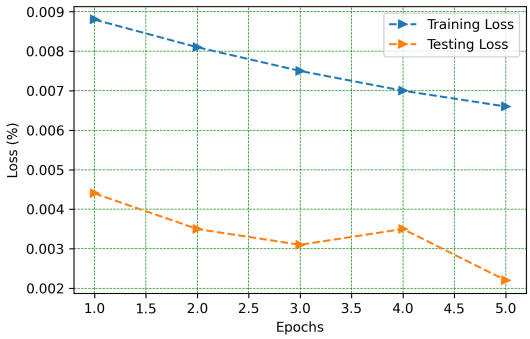


Fig. 7. Custom dataset poison model loss with trigger dataset.

B. Dataset

The Mnist digit dataset and custom dataset cat vs dogs are used for our experiments [17]. Mnist dataset contains 60,000 training images and 10,000 test images. The image size of this dataset is 1x28x28 and the corresponding classes from 0 to 9. Custom dataset contains 1500 training images and 1000 test images. The image size of this dataset is 3x64x64. We used imagenet weights to train the VGG16 model on custom dataset.

C. Evaluation of Attack

In this section, we evaluate the effectiveness of the proposed attack on pre-assumed attackers capabilities and examine the performance of attack for CNN and VGG16 DL network architecture.

1) Stealthiness: We evaluate the stealthiness of our generated poison images by a perceptual human viewer quality metric. We use a subjective PHash image quality metric to measure the perceptual assessment of poison dataset. We get these results for Mnist 92.1875% and custom dataset 86%.

2) Attack success rate: We use Gradient weighted-Class Activation Mapping (GRAD-CAM) visualization as a defence [18]. GRAD-CAM uses the gradients of any target class (say '9' in an CNN model) flowing into the final convolutional layer to produce a coarse localization map and highlight the critical regions in the image for predicting the concept. The results of this experiment is illustrate in Figure 3. From figure (a) It is clear that GRAD-CAM fails to identify the hidden, encoded trigger values. Likewise, in figure (b) our poison generated images are perceptually similar to given label but misclassified as the attackers target label. On the other hand, the model successfully misclassify digit '9' to digit '7' and 'cat' to 'dog' in the presence of encoded triggers. However, for digit image the bright pixel is also a part of the benign image. Therefore, this cannot be considered a trigger.

3) High Accuracy: The results of high accuracy and low accuracy loss of our two experiments are illustrates in Figure 4, 5, 6 and 7. We achieve almost 100% accuracy on Mnist dataset with only 0.03% model accuracy loss as illustrates in Figure 4 and 5. Whereas, for custom dataset we achieve 98% accuracy with 0.002% model accuracy loss as displayed in Figure 6 and 7.

TABLE II
ATTACK SUCCESS RATE (ASR), TEST ACCURACY, BLACKBOX AND STEALTHINESS OF PREVIOUS STUDIES AND OUR PROPOSED ATTACK.

Attack	Datasets	ASR	BB	Stealthy	IR
SIG [8]	Mnist /Traffic sign	85% / 73%	Yes	No	40%
Clean Label [9]	CIFAR10	>50%	No	No	25%
Refool [19]	ImageNet	82.11	Yes	Yes	20%
Our	Mnist/Cat-vs-Dog (Trained on ImageNet weights)	100 / 98%	Yes	Yes	3%

4) Comparative analysis with prior research: We performed a comparative analysis of invisible backdoor attack with three existing clean-label researches based on five factors, i.e. Attack Success Rate(ASR), Black Box (BB) attack, stealthiness, injection rate (IR) and model accuracy. The results of the four factors are present in Table II.

The results indicates that the 3% ratio of the whole clean dataset is sufficient enough to disrupt a DNNs model with a high attack success rate without degrading the accuracy as mentioned in table II. In contrast, other attacks need higher injection rate ranges from 40% to 20%. It is worth note here that we select the poison ratio based on our previous experiment [18]. The model accuracy and loss results on average are displayed in Figure 4, 5, 6 and 7. The results shows that our attack out perform existing papers in injection rate, attack success rate, stealthiness, accuracy and model loss.

IV. CONCLUSION

In this paper we provide a proof of concept to disrupt a working DNNs model by proposing a new attack paradigm, i.e., an invisible encoded backdoor attack. The experiment results demonstrate the possibility to inject backdoor attack with high attack success rate. The proposed backdoor attack evade GRAD-CAM visualization shows that the encoded triggers are not visible. It is observed that a constant α value may not work in all cases. For example, in the case of Mnist, different written styles may require different α values. It is possible to automate the α selection process, such as using another DNN to assess the quality of the image. Further, this paper only discuss the case of poison single-class i.e., mis-classifying an image of one source class to target class. It can be easily extended to multi-class scenarios. We can further evaluate the effectiveness and robustness of the invisible encoded backdoor attack on different datasets and well-known defence solutions. However, this is beyond this paper's scope and is considered a part of our future works.

REFERENCES

[1] Yiming Li, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shu-Tao Xia, "Backdoor learning: A survey," arXiv preprint arXiv:2007.08745, 2020.

[2] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang, "Tro- janing attack on neural networks," 2017.

[3] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Sid- dharth Garg, "Badnets: Evaluating backdooring attacks on deep neural networks," IEEE Access, vol. 7, pp. 47230–47244, 2019.

[4] Matthew Tancik, Ben Mildenhall, and Ren Ng, "Stegas- tamp: Invisible hyperlinks in physical photographs," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2117–2126.

[5] Han Qiu, Yi Zeng, Shangwei Guo, Tianwei Zhang, Meikang Qiu, and Bhavani Thuraisingham, "Deep- sweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," in Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security, 2021, pp. 363–377.

[6] Ziqi Wei, Junjian Shi, Yihe Duan, Ranyang Liu, Ye Han, and Zheli Liu, "Backdoor filter: Mitigating visible back- door triggers in dataset," in 2021 IEEE 1st International Conference on Digital Twins and Parallel Intelligence (DTPI). IEEE, 2021, pp. 102–105.

[7] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus, "Intriguing properties of neural networks," arXiv preprint arXiv:1312.6199, 2013.

[8] Mauro Barni, Kassem Kallas, and Benedetta Tondi, "A new backdoor attack in cnns by training set corruption without label poisoning," in 2019 IEEE International Conference on Image Processing (ICIP). IEEE, 2019, pp. 101–105.

[9] Alexander Turner, Dimitris Tsipras, and Aleksander Madry, "Label-consistent backdoor attacks," arXiv preprint arXiv:1912.02771, 2019.

[10] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octa- vian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein, "Poison frogs! targeted clean-label poison- ing attacks on neural networks," in Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 6106–6116.

[11] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang, "Clean-label back- door attacks on video recognition models," in Proceed- ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 14443–14452.

[12] Erwin Quiring and Konrad Rieck, "Backdooring and poisoning neural networks with image-scaling attacks," in 2020 IEEE Security and Privacy Workshops (SPW). IEEE, 2020, pp. 41–47.

[13] Mehdi Mirza and Simon Osindero, "Conditional gener- ative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.

[14] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in 3rd International Con- ference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Pro- ceedings, 2015.

[15] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel, "Backpropagation applied to hand- written zip code recognition," Neural computation, vol. 1, no. 4, pp. 541–551, 1989.

[16] Xiangyu Zhang, Jianhua Zou, Kaiming He, and Jian Sun, "Accelerating very deep convolutional networks for classification and detection," IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 10, pp. 1943–1955, 2015.

[17] Yann LeCun, Lawrence D Jackel, L eon Bottou, Corinna Cortes, John S Denker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger, Patrice Simard, et al., "Learning algorithms for classification: A comparison on handwritten digit recognition," Neural networks: the statistical mechanics perspective, vol. 261, no. 276, pp. 2, 1995.

[18] Iram Arshad, Mamoona Naveed Asghar, Yuansong Qiao, Brian Lee, and Yuhang Ye, "Pixdoor: A pixel- space backdoor attack on deep learning models," 29th European Signal Processing Conference (EUSIPCO), pp. 681–685, 2021.

[19] Yunfei Liu, Xingjun Ma, James Bailey, and Feng Lu, "Reflection backdoor: A natural backdoor attack on deep neural networks," in European Conference on Computer Vision. Springer, 2020, pp. 182–199.