Pixdoor: A Pixel-space Backdoor Attack on Deep Learning Models

Iram Arshad, Mamoona Naveed Asghar, Yuansong Qiao, Brian Lee, Yuhang Ye
Software Research Institute
Athlone Institute of Technology
Athlone, Ireland
i.arshad@research.ait.ie, masghar@ait.ie, ysqiao@research.ait.ie, blee@ait.ie, yye@research.ait.ie

***Abstract*—Deep learning algorithms outperform the machine learning techniques in various fields and are widely deployed for recognition and classification tasks. However, recent research focuses on exploring these deep learning models' weaknesses as these can be vulnerable due to outsourced training data and transfer learning. This paper proposed a rudimentary, stealthy Pixel-space based Backdoor attack (Pixdoor) during the training phase of deep learning models. For generating the poisoned dataset, the bit-inversion technique is used for injecting errors in the pixel bits of training images. Then 3% of the poisoned dataset is mixed with the clean dataset to corrupt the complete training images dataset. The experimental results show that the minimal percent of data poisoning can effectively fool a deep learning model with a high degree of accuracy. Likewise, in experiments, we witness a marginal degradation of the model accuracy by 0.02%.**

***Index Terms*—backdoor attack, causative attack, pixel-space, poisoned dataset, training phase**

I. Introduction

Deep Learning (DL) has been widely adopted in real-world situations from autonomous driving to automatic speech recognition because of its significant performance improvements in terms of accuracy and generalisability over other approaches. However, DL models tend to be oversized and include a high degree of redundancy. This redundancy makes it possible to inject malicious "logic" into the redundant part of a DL model without affecting the original functionality. Recent research studies show that DL models can be vulnerable to backdoor attacks, for example by injecting poisoned samples into training data set. This attack can happen when training is outsourced to the Cloud (e.g., Google Cloud Computing Engines), and when also a new DL model is acquired via transfer learning (using pre-trained models from Internet) [1], [2]. The backdoor attacks to DL models aim to control model outputs when the input data contain "triggers". A good backdoor attack can be stealthy because it almost does not affect the model accuracy when feeding normal data.

In recent studies, the digital pattern and physical pattern strategies have been proposed to generate attacks. In digital pattern strategy, some pixels of the images are corrupted. Whereas, in physical pattern strategy, yellow square, bomb and flower images are pasted over the images to generate attacks [3]. Based on the machine learning classification modelling, the security threats are categorized into training (known as causative attack) and test phase attacks (known as exploratory attack) [4]. The training phase attacks are more potent than the test phase attacks as an insider/employee from the data collection team within the organization may act as an adversary by mixing poisoned dataset into the training pipeline. In this particular research, we focus on digital pattern strategies, since by following these strategies the attacks are comparatively stealthy, compared to the attacks generated by physical patterns. In the following content, "stealthy" means that the injected pixel bit-errors in the images can deceive the human eye.

To gaining deeper insights about how backdoor attacks can be implemented in the wild for image classification, this paper will answer two research questions.

RQ1 *Is it feasible to develop a stealthy backdoor attack with a bit-inversion technique in training data?*

RQ2 *How much poisoned training data is required to attack or disrupt a working DL model successfully?*

In this work, we implement a novel backdoor attack on the data in the training pipeline. The proposed Pixdoor attack is significantly powerful than the existing Backdoor attacks in terms of stealthiness and attack success rate with a minimal poisoned data injection rate. In summary, following points highlight the research contribution of this paper:

- A method to generate backdoor attack by slightly changing image pixel values to poison the images in training dataset. It aimed at mis-classifying the target classes as per the adversary goals without knowing the DL model. (section III)
- This paper employs perceptual hash (pHash) as a quality metric to measure the perceptual difference between an original and the pixel-space errors poisoned image. (section III-B3)
- Experiments show that the proposed Pixdoor attack leads to the high attack success rates without reducing the tested model's accuracy. (section IV-C)

The remainder of the paper is organized as follows. Section II presents an overview of the related work. Section III presents the proposed methodology with pertinent details. Section IV contains an experimental setup and discussion on the results.

Acknowledgement This publication has emanated from research conducted with the financial support of Athlone Institute of Technology under President Doctoral Scholarship

Section V concludes the paper along with the limitations of the presented work and scope for future improvements.

II. Related Work

This section discussed the recent related work on Backdoor attacks against deep neural network models. A deep learning algorithm's security gets the research community's attention since 2014 [5]. We only discussed the related studies which highlighted training phase Backdoor threats for DL algorithms in the subsequent paragraphs.

In the research study [1], the researchers proposed targeted Backdoor attacks using input-instance key and pattern-key strategies for learning based authentication systems. To generate input-instance attacks, they added random noise in the images. Besides, they have used blended accessory injection strategy for generating pattern-key attacks. They evaluated the proposed poisoned Backdoor attack for transfer learning settings under weak assumptions. In another study [3], the authors proposed two different kinds of Backdoor attacks; first is a single-pixel attack, a single bright pixel on the bottom right corner of the image and second is the pattern of bunch of bright pixels on the bottom right corner of the image. Further, they used the BadNet model for fully outsourced training and transfer learning settings under weak assumptions where the adversary does not know the model. In the research study [6], the authors generated TargetNet Backdoor attack by added a white rectangular sticker pasted on the images. Additionally, they have trained the targeted classifier on the Backdoor so that the target class, which has selected by the attacker, can mis-classified based on the trigger at a specific location. Further, they have evaluated this white-box attack under strong assumptions where the adversary has full knowledge of the DL model. In research study [7], the authors, proposed Backdoor injection attacks for data poisoning. The patterned static perturbation mask attack generated by replaced the pixel value to 10 of the image's top-left corner. Whereas, DeepFool algorithm used to generate the targeted adaptive perturbation mask attack. Further, they checked the attack performance under BIB (Backdoor Injection Before Model Training) and BID (Backdoor Injection During Model Updating) assumptions. In the research study [8], the researchers, systematically evaluate the effectiveness of the existing Backdoor attack which is proposed in the research study [3]. They have used CNN-Lenet-5 model for traffic sign datasets and evaluate the effectiveness of Backdoor attacks for autonomous driving scenario. In the research study [9], the researchers, proposed a universal perturbation by modified just one pixel of a colour image (B = 0) and added noise for all the images during the training set. They experimented and revealed that the neural network fooled by looking at the added noise vector [9].

In contrast with [9], Pixdoor attack used bit-inversion technique and inject stealthy pixel bit-errors in an image. It is worth noted here that the proposed bit-inversion technique does not change image size. Furthermore, our proposed study has a high attack success rate and a tiny poisoned sample injection rate compared to [9].

III. Methodology

This section presented proposed methodology by defining a threat model and a Backdoor generation strategy.

A. Threat Model

1) Adversary Goals

Following absolute conditions should meet while performing Backdoor attack during training pipeline.

- Injection rate: With respect to existing Backdoor attack studies [3], [7], [9], we also assumed that a tiny portion of poisoned dataset added into the clean training dataset without further control over the training process. (answering RQ2).
- Targeted attack: Earlier, we have mentioned that two kinds of attacks during the training phase are studied. We are primarily focusing on targeted Backdoor attack. We considered that the adversary attempt to add Backdoor instance associated with the targeted label during the training process.
- Attack success rate: The given Backdoor instance $x^{'}$ associated with the label $y^{'}$ as per the adversary should classified with high accuracy on unseen (poisoned test dataset) images. These $y^{'}$ are not originally labelled as y. In particular, we measure the probability of the model to classified any poisoned image (Backdoor instance) to the predicted label $y^{'}$ with high accuracy. The formula to measure the attack success rate (ASR) provides below: ASR = $[\mathcal{F}(x^{'}) = y^{'} \mid \mathrm{y} \neq y^{'}]$
- High accuracy rate: Regardless of the adversary attack, the model expected to perform well as it is working in the absence of Backdoor instances. The tested baseline clean model shows 99.13% test accuracy.

2) Capabilities of adversary

We have assumed the weakest and realistic assumption to add this attack into the training pipeline.

- Minimal dataset knowledge: Unlike some existing studies assumptions, [10] and [11] where the adversary has full knowledge and understanding of the training dataset, we are assuming that the adversary has an idea about the dataset only.
- Black-box attack: There are some research studies [2] [12] which proposed Trojan (another name of Backdoor) attacks on neural networks. Such white-box attacks required full knowledge of the model, which is not a realistic approach. Therefore, we are making a realistic assumption that the adversary does not have knowledge about the model. Once the Backdoor added into the training pipeline, the adversary has no further control over the model's training process.

B. Backdoor Generation Strategy

This section generated a poisoned dataset by using bit-inversion technique and proposing an algorithm to consider the goals and assumptions mentioned above. Further, this section

TABLE I
A SUMMARY OF NOTATIONS.

Name	Symbol	Description
Model	$\mathcal{F}(x)$	a CNN Model.
Training set	$\mathcal{X}$	Let $\mathcal{X}$ represents a set of training images.
poisoned image	$X^{'}$	Modified version of the clean image used to mis-classified the targeted class.
Size	$\mathcal{N}$	$\mathcal{N}$ represents size of $\mathcal{X}$.
Data point	D_{point}	D_{point}: Dimension of a data point x^i.
Label	$\mathcal{Y}$	$\mathcal{Y}$: Dimension of a label y^i.
Classes	$\mathcal{C}$	$\mathcal{C}$ represents a class $\{0, 1, 2, 3, \dots 9\}$.
Random number	U	Random uniform distribution U like $X \sim U(a, b)$.
Testing predicted values	($x^{'}$, $y^{'}$)	($x^{'}$, $y^{'}$) where $y^{'}$ = F ($x^{'}$).
Training dataset	$D_{Trainingset}$	Training dataset.
poisoned dataset	D_{Advset}	poisoned dataset after injecting the errors.
Predicted output	$\hat{y}$	A predicted output.
Final dataset	D_{Fset}	Union of training and poisoned dataset.

Algorithm 1: Generation of poisoned Images

Result: $X^{'}$ poisoned images
initialization;
for *range in* i^{th} *location of row* **do**
  for *range in* j^{th} *location of col* **do**
    if *loc[i][j]* > 0 **then**
      generate random number U(0.4 to 0.5)
      loc[i][j] = random number
    end
  end
end

provides the definitions of technical terms, attack notations, proposed algorithm and in last the Injection process.

1) Attack Notations

This paper considered a CNN model $\mathcal{F}(x)$, which takes input $X \in D_{Trainingset}$ and predicts the label $\hat{y} \in \mathcal{C}$ $\{0, 1, 2, 3, \dots 9\}$ as a final output. The goal of the adversary is to add poisoned images $X^{'} \in D_{Advset}$ synthesize with clean images X. As each X contains a pair of (x^i, y^i) so as $X^{'}$ has a pair $(x^{i^{'}}, y^{i^{'}})$. The adversary also corrupted the corresponding labels $y^{'} \in \mathcal{C}^{'}$ $\{0, 1, 2 \dots 9\}$ associated with poisoned images (describe in Table I).

2) Proposed Algorithm

To generate the poisoned dataset, we observed that the grey-scale image has zero pixel value on non feature area. The rest of the area where some of the features are available contains the pixel value greater than zero, so we decided to apply bit-inversion technique on the non-zero pixel values. The original pixel value are replaced with random number using the algorithm presented as Algorithm 1 to generate the desired poisoned dataset. It is noted here that the image quality is effected only while keeping the same image size to maintain the reliability for the trained DL model.

3) Injection Strategy

It is essential to generate strong Backdoor samples. Nevertheless, the hidden Backdoor triggers and the targeted

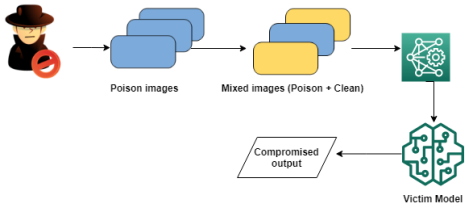

Fig. 1. Proposed Pixdoor attack injection process.

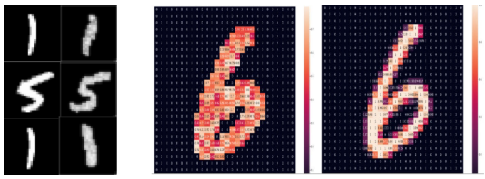

Fig. 2. In the first row the image similarity is 90.62%, in the second row the image similarity is 96.85% where as in last row the image similarity is 98.43%. Meantime, the right side of image is showing the heatmap of the poisoned and clean image respectively.

labels should be quickly and effectively learnt and classified by the model. Similarly, it is also essential to keep in mind to add minimum error in poisoned image to evade detection.

Figure 1 describes the proposed Backdoor injection strategy i.e mixing of poisoned dataset with the clean dataset.

Preparation of poisoned dataset Based on the above-mentioned minimal dataset knowledge assumption, the adversary prepared the $D_{Advset} = X^{'}$ by randomly selecting some of the Mnist dataset images as assumed in this paper [3]. Firstly, given an input image (c = 1, h = 28, w = 28) we observed the pixels of Image. Secondly, we performed in total three-round experiments. Finally, we decided to select random numbers between (0.4 - 0.5) to update the pixel values and injected the error in the image by using the proposed algorithm. In each iteration looking into the data points of x^i image i^{th} and j^{th} row, column value, the goal is to generate $x^{i^{'}}$ by updating the i^{th} and j^{th} row, column value of x^i between (0.4 - 0.5) for all the Image belongs to class 0 to 9. Thirdly, we decided that poisoned Image's acceptance criteria should have a pHash similarity more significant than 97%. In the first experiment, we chose the U between (0.4 - 0.8), and generated an image. Afterwards, we computed the generated image's pHash value with the original one and got 90% image similarity; this does not meet the acceptance criteria. In the second round of the experiment, we decided U between (0.4 - 0.6) and by computing the pHash, we got the 95% similarity. Unexpectedly, it does not meet our acceptance criteria as well. Therefore, in our third experiment, we decided the U between (0.4 - 0.5) and got 98% similarity, which achieved our acceptance criteria. By following this process, the adversary can prepare a poisoned dataset. The image similarity and heatmap of benign and poisoned images illustrates in figure 2.

Backdoor Injection The adversary synthesized a new dataset by mixing tiny portion of poisoned dataset with the clean dataset $D_{Fset} = D_{Trainingset} \cup D_{Advset}$ during the

training pipeline and ensure that other functionality should not be affected by mixed dataset. The existence of poisoned dataset leads to the following loss function:

$$min_\theta \sum_{i=0}^{n} l(\theta, (x^i, y^i)) + \sum_{j=0}^{m} l(\theta, (x^{i'}, y^{i'})) \quad (1)$$

In eq. 1, l is the loss based on cross-entropy, θ is model parameters and $(x^i, y^i) \in X$ and $(x^{i'}, y^{i'}) \in X^{'}$. The portion of poisoned dataset has an impact on the performance which we identified and discussed in a detail in the sub-section IV-C.

IV. Experiment Setup

This section comprehensively describes our experimental setup to produce the Pixdoor attack on tested DL model.

A. Deep Network Model

To demonstrate the experimental setup, we have selected deep learning Convolutional Neural Network (CNN) basic architecture LeNet (proposed in [13]) as a baseline model without altering number of layers in the model architecture. LeNet model is implemented with two convolutional layers, followed by Relu, max pool and dropout activation functions with two fully connected layers and one output layer. The experiments are performed using Pytorch framework.

B. Dataset

The Mnist digit dataset is used for our experiments [14]. This dataset has greyscale handwritten digits and the corresponding classes from 0 to 9. This dataset contains 60,000 training and 10,000 test sets as examples. We use the Mnist dataset because it is a widely accepted benchmark in the literature to test the performance of the proposed Pixdoor attack during the training pipeline.

C. Evaluation of Pixdoor attack

In this section, we evaluate the effectiveness of proposed Pixdoor attack based on pre-assumed attack capabilities (given in sub-section III-A2) and examine the performance of the Pixdoor attack for LeNet DL network architecture.

1) Attack success rate

We evaluate our Pixdoor attack's effectiveness by splitting the poisoned dataset into 80 and 20 ratios. 80% of the poisoned dataset belong to the training dataset, and 20% of the remaining belong to the test dataset. The 20% test dataset considers measuring the attack success rate because classifier does not train on this dataset (unseen to the network). We have conducted five experiments by selecting the poisoned dataset ratio 0.5%, 1%, 1.5%, 2% and 3%, respectively. This ratio is calculated based on the whole clean dataset. Figure 3 illustrates the attack test loss error rate for our selected poisoned dataset ratio's. Firstly, for 0.5% ratio of the poisoned dataset, the error rate is too high to compute the attack success rate. Similarly, the 1% ratio error rate sharply decreased as compared to the 0.5% ratio, but the error rate is still too high. Secondly, we observed that for 1.5% and 2% ratio, it has a marginal difference between the error rate.

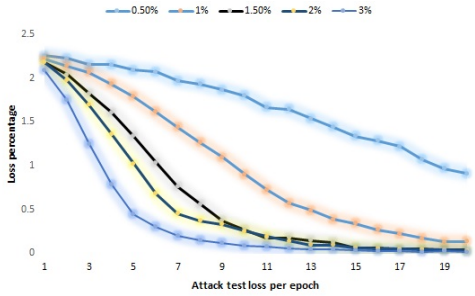

Fig. 3. Attack test loss error rate for small portion of poisoned dataset.

TABLE II
AVERAGE ATTACK SUCCESS RATE (%), VICTIM MODEL(%), AND BASE LINE MODEL (%) WITH RESPECT TO INJECTION RATE 2% AND 3% OF TOTAL SAMPLE SIZE.

Clean Model	Injection rate = 1200		Injection rate = 1800	
Base Model Accuracy	Victim Model accuracy	Attack success rate	Victim Model accuracy	Attack success rate
99	98.7700— 60.0000		98.7800 —90.0000	
99.04	98.6900— 60.0000		98.6800 —90.0000	
99.05	98.8800— 60.0000		98.8100 —90.0000	
99.13	98.9100— 60.0000		98.7500 —90.0000	

In comparison, 3% ratio has overall less error rate all of the above. Therefore, we selected 2% and 3% ratio to compute the attack success rate as the error rate decreased gradually. Based on these results, we opted to mix 3% poisoned dataset with the clean dataset into the training process.

Discussion: While performing the experiments, we had following observations: (1) For calculating attack success rate, we examined that the attack success rate is 90% when the 3% poisoned dataset mixed with the clean dataset. While, after mixing the 2% of poisoned dataset within clean dataset, the significant difference in the attack success rate which was 60%. (2) It was also observed that by adding 2% and 3% ratio of the poisoned dataset mixed with the clean dataset, there is a marginal degradation in model accuracy 0.22%. (3)We observe the inverse relationship between the poisoned dataset and the clean dataset size. For example, if the clean dataset size is 3000 in total than 60% of the whole clean dataset should be corrupted. Besides if the clean dataset size is 90,000 only 2% of the whole dataset is required to poisoned the clean dataset. In the real-world scenario, we have to deal with the thousands and millions of data to solve a particular problem, so there is a high probability that the adversary may inject the hidden errors as Backdoor mis-classifed the actual output as per the adversary desire output with the high attack success rate.

2) High accuracy rate

The attack success rate and model accuracy results on average displays in table II. This result indicates that 3% ratio of whole clean dataset achieve the high attack success rate with the accuracy loss of 0.22% only. In Figure 4, the top of the Image displays the base model's accuracy. The centre image shows the accuracy of 2% of poisoned dataset synthesizes with clean dataset. In contrast, the bottom depicts the accuracy and loss of 3% ratio of poisoned dataset synthesizes with clean

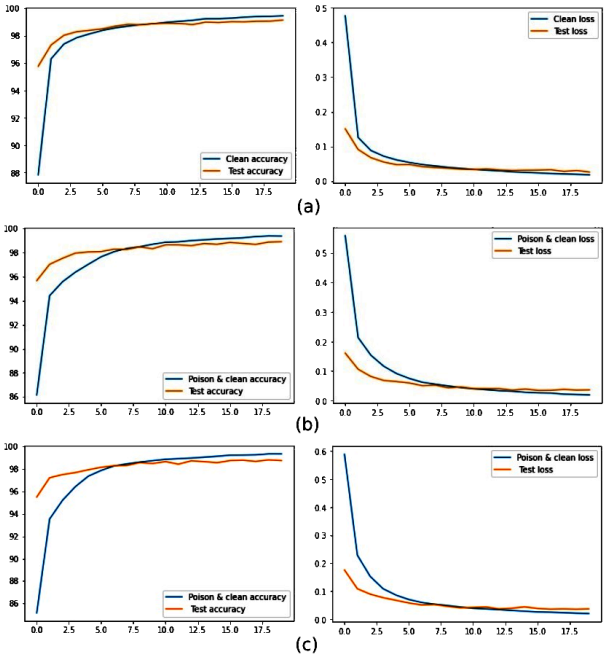

Fig. 4. (a) the benign model accuracy and loss, (b) the accuracy and loss of 2% of poisoned dataset, (c) 3% of injected poisoned dataset.

TABLE III
PERFORMANCE COMPARISON

Attack Name	Attack Success rate	Average attack Loss	Model Accu-racy loss	Sample Injection	Average accuracy	Stealthi-ness	Black-box attack
BadNet [3]	99%	N/A	0.50%	10%	99.00%	No	Yes
Universal Perturbation [5]	74.1% to 98.9%	N/A	N/A	10%	97%	Yes	Yes
TargetNet [6]	100%	N/A	N/A	10% to 50%	99.25%	No	No
Pixdoor	90%	0.01%	0.02%	3%	98.75%	Yes	Yes

dataset.

3) *Comparative analysis with prior research*

We performed a comparative analysis of Pixdoor attack with existing research on the basis of seven factors, i.e. attack success rate, average attack loss, model accuracy, sample injection ratio, average accuracy (clean+poisoned), and black box attack. Table III presents the comparative evaluation of the proposed backdoor attack. Table III shows that the proposed Pixdoor outperforms the current state-of-art work in sample injection ratio, attack success rate and model accuracy loss. Nevertheless, [6] has high attack performance as compared to our study, but it is a white-box attack, and the adversary has full knowledge of DL model, which is difficult to obtain in practice. It is noted here that for study [7] we are not comparing our results as they do not provide rigorous analysis on Mnist dataset.

V. CONCLUSION & FUTURE WORK

This paper provides a proof of concept to fool a working DL model by proposing a novel Backdoor attack i.e. Pixdoor. The experimental results demonstrated the possibility of injecting bit-level stealthy errors in the training images dataset to execute a Backdoor attack. It is also examined that the mixing of only 3% poisoned dataset within clean dataset is sufficient to fool a DL model with 90% attack success rate. Moreover, the stealthiness of proposed Pixdoor attack is compared with the state-of-art solutions. In future, Pixdoor can be further evaluated by injecting it into real world DL applications using various datasets along with a defence solution to detect the stealthy error factors to poisoned the datasets.

REFERENCES

[1] X. Chen, Chang Liu, Bo Li, Kimberly Lu, and D. Song. Targeted back-door attacks on deep learning systems using data poisoning.ArXiv,abs/1712.05526, 2017.

[2] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Wei-hang Wang, and Xiangyu Zhang. Trojaning attack on neural networks.2017.

[3] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating Backdooring attacks on deep neural networks.IEEE Access, 7:47230–47244, 2019.14

[4] Mingfu Xue, Chengxiang Yuan, Heyi Wu, Yushu Zhang, and WeiqiangLiu. Machine learning security: Threats, countermeasures, and evaluations.IEEE Access, 8:74720–74742, 2020.

[5] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties ofneural networks. InInternational Conference on Learning Representations,2014.

[6] Hyun Kwon, Jungmin Roh, Hyunsoo Yoon, and Ki-Woong Park. TargetnetBackdoor: Attack on deep neural network with use of different triggers. InProceedings of the 2020 5th International Conference on Intelligent Infor-mation Technology, pages 140–145, 2020.

[7] Haoti Zhong, Cong Liao, Anna Cinzia Squicciarini, Sencun Zhu, and DavidMiller. Backdoor embedding in convolutional neural network models viainvisible perturbation. InProceedings of the Tenth ACM Conference onData and Application Security and Privacy, pages 97–108, 2020.15.

[8] Huma Rehman, Andreas Ekelhart, and Rudolf Mayer. Backdoor attacks inneural networks–a systematic evaluation on multiple traffic sign datasets. InInternational Cross-Domain Conference for Machine Learning and Knowl-edge Extraction, pages 285–300. Springer, 2019.

[9] Michele Alberti, Vinaychandran Pondenkandath, Marcel Wursch, ManuelBouillon, Mathias Seuret, Rolf Ingold, and Marcus Liwicki. Are you tam-pering with my data? InProceedings of the European Conference on Com-puter Vision (ECCV) Workshops, pages 0–0, 2018.

[10] Luis Mu noz-Gonz alez, Battista Biggio, Ambra Demontis, Andrea Paudice,Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning ofdeep learning algorithms with back-gradient optimization. InProceedingsof the 10th ACM Workshop on Artificial Intelligence and Security, pages27–38, 2017.

[11] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson,Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In2016 IEEE European symposium on securityand privacy (EuroSP), pages 372–387. IEEE, 2016.

[12] Minhui Zou, Yang Shi, Chengliang Wang, Fangyu Li, Wen-Zhan Song, andYu Wang. Potrojan: powerful neural-level trojan designs in deep learningmodels.CoRR, abs/1802.03043, 2018.

[13] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson,Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backprop-agation applied to handwritten zip code recognition.Neural computation,1(4):541–551, 1989.

[14] Yann LeCun, Lawrence D Jackel, L eon Bottou, Corinna Cortes, John SDenker, Harris Drucker, Isabelle Guyon, Urs A Muller, Eduard Sackinger,Patrice Simard, et al. Learning algorithms for classification: A comparisonon handwritten digit recognition.Neural networks: the statistical mechan-ics perspective, 261:276, 1995.