PERSISTENT WATERMARK FOR IMAGE CLASSIFICATION NEURAL NETWORKS BY PENETRATING THE AUTOENCODER

Fang-Qi Li, Shi-Lin Wang, Senior Member, IEEE*

{solour_lfq,wsl}@sjtu.edu.cn
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University.

ABSTRACT

Deep neural networks for image processing, especially image classification, have become ubiquitous. To protect them as intellectual properties and standardize the commercialization of their service, watermarking schemes have been proposed to authenticate the author of models. Many black-box watermarking schemes insert a backdoor into the neural network by poisoning the training dataset. Their performance declines if the adversary who has stolen the model adds a noise reducer, in particular an autoencoder, to ruin the backdoor. To cope with this kind of piracy, we propose an enhanced watermarking scheme by using triggers that penetrates the adversary's autoencoder. The penetrative triggers are generated from a collection of shadow models that approximate the adversary's autoencoder, which is assumed to be hidden from the genuine host of the model. The proposed scheme is shown to be resistant to the filtering of autoencoders and significantly increase the robustness of ownership verification.

Index Terms— Deep learning model protection, forensics and security, watermark.

1. INTRODUCTION

Deep neural networks (DNN) for image classification have boosted the wide application of computer vision. The cost of building a state-of-the-art DNN model is high due to the consumption in collecting data and tuning the parameters. For this reason, voices are calling for treating DNNs as intellectual properties. Numerous proposals used watermark as a mechanism of ownership verification for DNN [1, 2, 3, 4, 5]. During the training process, the host of a DNN model embeds its identity information into the model as a watermark, then the model is published. If an adversary steals the model and claims to have trained it from scratch, then the host can evoke the watermark to prove the ownership.

Current DNN watermarking schemes can be classified into white-box schemes [4, 6, 7] and black-box ones [8, 9, 10]. In the white-box setting, the host has full access to the DNN of the adversary. Under this setting, the host's identity can be embedded into the model's parameters. The case where the parameters of the suspicious model are unrevealed, which is more realistic, is the black-box setting. Most of the watermarking schemes for the black-box setting depends on the backdoor [8]. To insert a backdoor into a DNN, the host encodes its identity into a set of trigger samples with corresponding labels and addes it to the training set. The triggers are hidden from publicity so the adversary does not know them. To evoke the backdoor, the host inputs the triggers to the model. If the model has learned these samples, it is likely to return the labels assigned by the host. Otherwise the output is a random guess, hence the ownership is proven.

Triggers can be seen as ordinary samples with an extra stamp [8, 11]. Therefore the adversary can block the triggers with a noise reducer, e.g., an autoencoder (AE) [12, 13] and paralyze the watermark, this is known as *AE piracy* [14, 15].

To deal with this kind of piracy, we propose a persistent watermarking scheme by using triggers that *penetrates* the AE filter. The penetrative triggers are forged to be invariant to the transformation of the AE. We show that these triggers can preserve the performance of the watermark against the adversaty's AE. The paper makes the following contributions:

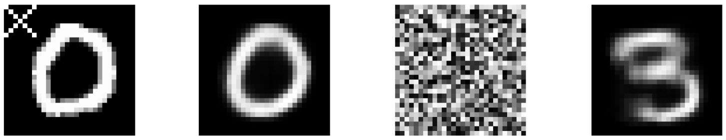
- We address the threat model of AE piracy by adopting shadow AEs to approximate the adversary's AE.
- We propose a DNN watermarking scheme that is persistent against the AE piracy by using penetrative triggers generated from the shadow AEs and empirically examine its efficacy.

To the best of our knowledge, this is the first effort in resolving the AE piracy with special triggers.

2. MOTIVATIONS

Ordinary triggers used in watermarking schemes [8], such as images stamped by an extra mark or random noise can be efficiently eliminated by an AE as shown in Fig. 1. Since the triggers are prevented from reaching the DNN model, such elimination sharply reduces the model's performance on the trigger set and compromises the ownership verification.

This work was supported by National Natural Science Foundation of China (61771310). Shi-Lin Wang is the corresponding author.



(a) Stamp trigger. (b) Trigger (a) after AE. (c) Random trigger. (d) Trigger (c) after AE.

Fig. 1. An AE can wreck triggers from [8].

To increase the persistency of the black-box watermark in this scenario, we have to use triggers that can bypass the filter of an AE. Meanwhile, the host's identity has to be merged into the triggers so the watermark is unforgeable. However, in the black-box setting, the adversary's AE is also hidden so it is impossible to directly generate adversarial triggers from AE's gradient. Inspired by [16], we adopt a series of AE as *shadow models* that approximate the adversary's AE. The property of being penetrative is formulated as an optimization task on the shadow AEs. Triggers are generated by conjugately maximizing its penetration ability and its correlation with the author's identity. A watermarking scheme using these triggers is both persistent against the AE piracy and unforgeable.

3. THE PROPOSED MODEL

3.1. Model Overview

Assume that the host is to train an image classification DNN model on the training dataset with N labeled samples $\mathcal{D} = \{(x_n, y_n)\}_{n=1}^{N}$. The domain of image is denoted by $\mathcal{X}$ and the collection of all labels is denoted by $\mathcal{Y}$.

The **host** embeds its identity information $\texttt{key}$ into altogether I prestamps with assigned labels $\{(P_i, y_i)\}_{i=1}^{I}$ where $P_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$ [17]. To evade the adversray's AE, the host firstly trains a series of K shadow AEs: $\mathcal{A} = \left\{\texttt{AE}_k^{\text{shadow}}\right\}_{k=1}^{K}$. Then a penetrative stamp T_i meeting the following requirements is generated for each P_i: (i). T_i is similar to P_i to ensure the *unforgeability* of the watermark. (ii). When T_i is added to a carrier $x_{i,j} \in \mathcal{X}$ to forge a penetrative trigger $T_i + x_{i,j}$ and passes an shadow AE, the result should be similar to T_i to achieve *penetration*. Finally, a DNN model $\texttt{M}$ is trained by tuning parameters on $\mathcal{D}$ with the penetrative triggers $\{(x_{i,j} + T_i, y_i)\}$. The entire procedure is illustrated in Fig. 2.

The **adversary** downloads $\texttt{M}$, adds $\texttt{AE}^{\text{adv}}$ to conduct AE piracy, and broadcast $\texttt{M} \circ \texttt{AE}^{\text{adv}}$ as its own product. It is expected that due to transferability, the host's watermark remains valid by having the triggers penetrate $\texttt{AE}^{\text{adv}}$ and correctly evoke the assigned labels, i.e., the triggers that penetrate the shadow AEs can penetrate $\texttt{AE}^{\text{adv}}$ as well.

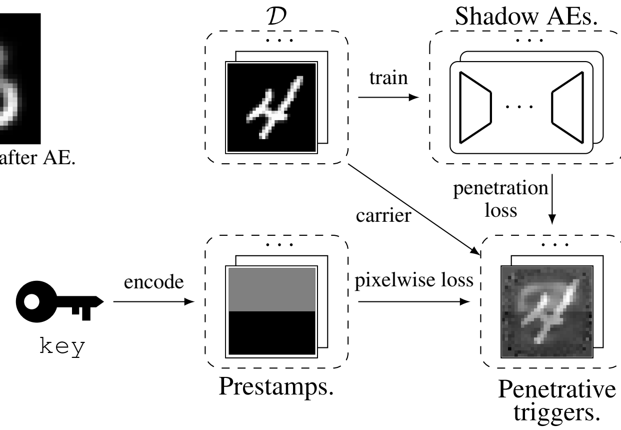


Fig. 2. The framework of genetrating penetrative triggers.

3.2. The Shadow Autoencoders

An autoencoder $\texttt{AE}$ is trained on $\mathcal{D}$ as an approximation of the identity mapping. $\texttt{AE}$ is a composite of an encoder $\texttt{Enc}$ and a decoder $\texttt{Dec}$, i.e., $\texttt{AE} = \texttt{Dec} \circ \texttt{Enc}$. Let θ and ϕ be the parameters of $\texttt{Enc}$ and $\texttt{Dec}$, $\texttt{AE}$ can be trained by minimizing the reconstruction loss:

$$\mathcal{L}_{\text{rec}}(\theta, \phi) = \sum_{n=1}^{N} \|x_n - \texttt{Dec}_\phi(\texttt{Enc}_\theta(x_n))\|_2^2. \tag{1}$$

The medium representation $\texttt{Enc}(x_n)$ is interpreted as the feature vector of x_n. For the ease of image generation, a prior normal distribution is usually exerted on the space of feature vectors [12]. So $\texttt{AE}$ is obtained by minimizing:

$$\mathcal{L}_{\texttt{AE}}(\theta, \phi) = \mathcal{L}_{\text{rec}}(\theta, \phi) + \lambda_1 \cdot \|\texttt{Enc}_\theta(x_n)\|_2^2, \tag{2}$$

where $\mathcal{L}_{\text{rec}}$ follows (1). The adversary's $\texttt{AE}^{\text{adv}}$ is trained on a dataset with a similar distribution as $\mathcal{D}$ (yet is unknown to the host and might be disjoint with $\mathcal{D}$), otherwise it cannot perform as a noise reducer. With all shadow AEs trained on $\mathcal{D}$, following the inspiration from [16], we expect that they correctly approximate $\texttt{AE}^{\text{adv}}$ since their underlying data distribution is identical. So penetrating them is tantamount to penetrating $\texttt{AE}^{\text{adv}}$.

3.3. Generating Penetrative Triggers

To produce penetrative stamps and triggers from $\mathcal{D}$, shadow AEs and the prestamp P_i, the host firstly selects a collection of carrier images $D_i = \{(x_{i,j}, y_{i,j})\}_{j=1}^{J} \subset \mathcal{D}$. This can be done by using a block cipher to permuate the index of images in $\mathcal{D}$ with $(\texttt{key}, i)$ as the seed and include the first J images into D_i. The penetrative stamp T_i has to meet the following constraints:

$$l(T_i, P_i) \leq \epsilon_1, \tag{3}$$

$$\forall k, j, \;\; l(\texttt{AE}_k^{\text{shadow}}(x_{i,j} + T_i), T_i) \leq \epsilon_2, \tag{4}$$

where l is a metric defined on $\mathcal{X}$. In order for the salient pixels to correctly evoke the backdoor, the pixelwise mean square loss is the optimal choice.

We design the following loss function to explicitly meet all $J \cdot K + 1$ constraints in (3)(4):

$$\mathcal{L}(T_i) = \sum_{j=1}^{J}\sum_{k=1}^{K} \frac{\|\texttt{AE}_k^{\text{shadow}}(x_{i,j}+T_i) - T_i\|_2^2}{J \cdot K} + \lambda_2 \cdot \|T_i - P_i\|_2^2, \tag{5}$$

whose minimizer is a stamp close to P_i. After being added to an image $x_{i,j}$, it can pass any shadow AE.

To analyze the generation of T_i from (5), we expand each AE to the first order gradient:

$$\texttt{AE}_k^{\text{shadow}}(x_{i,j}+T_i) = x_{i,j} + \nabla_x \texttt{AE}_k^{\text{shadow}}(x)|_{x_{i,j}} T_i + o(T_i), \tag{6}$$

where we have $\texttt{AE}_k(x_{i,j}) = x_{i,j}$. Hence the gradient of (5) w.r.t. T_i depends on $\texttt{AE}_k^{\text{shadow}}$ through its gradient at $x_{i,j}$, whose value fluctuates slightly across AEs trained on similar images (details are shown in Section 4.1). This is because all AEs are trained to approach the identity mapping for images subject to the distribution introduced by $\mathcal{D}$, hence the gradient is close to the identity matrix.

In the idealistic setting we would minimize the loss:

$$\mathcal{L}^{\text{Ideal}}(T_i) = \sum_{j=1}^{J} \frac{\|\texttt{AE}^{\text{adv}}(x_{i,j}+T_i) - T_i\|_2^2}{J} + \lambda_2 \cdot \|T_i - P_i\|_2^2, \tag{7}$$

As a substitude, we expect that the dependency of the dominating term in the gradient of (7) w.r.t. T_i on $\texttt{AE}^{\text{adv}}$, which is proportional to $x_{i,j}^{\text{T}}(\nabla_x \texttt{AE}^{\text{adv}}|_{x_{i,j}} - I)$ (plugging (6) into (7) and retaining only the first-order term), can be correctly estimated by that of (5). Since this term is linear w.r.t. $\nabla_x \texttt{AE}^{\text{adv}}|_{x_{i,j}}$, we average the gradients of the shadow AEs on the same point $x_{i,j}$ as a maximum likelihood estimation, whose bias declines with an increasing K. So using many shadow AEs can effectively approximate $\texttt{AE}^{\text{adv}}$.

Finally, we add the penetrative stamps to their carriers to form penetrative triggers and merge them with the training dataset:

$$\mathcal{D}' = \mathcal{D} \setminus \{(x_{i,j}, y_{i,j})\}_{i=1,j=1}^{I,J} \cup \{(x_{i,j}+T_i, y_i)\}_{i=1,j=1}^{I,J}. \tag{8}$$

The salient pixels of T_i and $\{x_{i,j}+T_i\}_{j=1}^{J}$ remain similar, so the backdoor is successfully inserted into the published model. The classifier $\texttt{M}$ is trained on $\mathcal{D}'$ by minimizing the cross-entropy loss.

4. EXPERIMENTAL RESULTS

4.1. Experiment Settings

Experiments were conducted on MNIST [18] and Fashion-MNIST [19] (F-MNIST). The task for both datasets is image classification with ten classes. All images are of size 28×28. Both datasets have $N = 60,000$ samples for training and 10,000 samples for validation. We adopted the AE structure whose encoder and decoder each has four consecutive fully connected linear layers followed by the `Tanh` activation. The last layer of the decoder is the `sigmoid` function to ensure that AE's output lies in $\mathcal{X}$. Shadow AEs and $\texttt{AE}^{\text{adv}}$ were trained on two disjoint subsets of the original training dataset with 40,000 and 20,000 samples by minimize the regularized reconstruction loss (2) with $\lambda_1 = 10^{-3}$. The mean squared loss on the gradient of the shadow AEs was below 2×10^{-14}, making the shadow model approximation empirically effective. We chose $I = 2$ prestamps and set y_1, y_2 to 0 and 7. To minimize (5), $J = 30$ samples from the original training dataset were selected for each prestamp, with $\lambda_2 = 2 \times 10^{-2}$. According to (8), the percentage of samples being modifed was $\frac{I \times J}{N} = 0.1\%$. The backend model $\texttt{M}$ is the classical image classification network structure, ResNet-18 [20]. The average validation accuracy for MNIST and F-MNIST was 99.4% and 90.5%[1].

4.2. The Number of Shadow Autoencoders

We firstly trained four AEs $\mathcal{A} = \{\texttt{AE}_1 - \texttt{AE}_4\}$ on MNIST and assumed that the adversary adopted one AE from $\mathcal{A}$. The configuration and the classification accuracy on the trigger set is demonstrated in Fig. 3. It can be observed that averaging

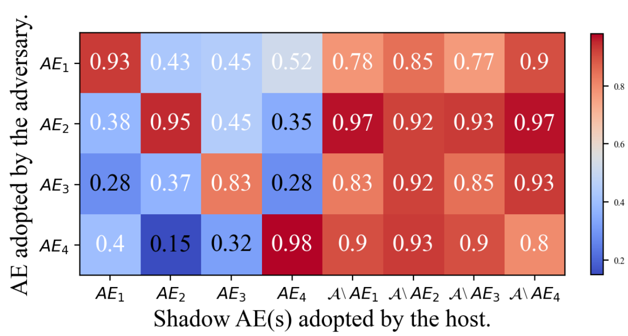


Fig. 3. Classification accuracy on penetrative triggers.

over many shadow AEs can efficiently approximate an AE trained on the identical dataset, since its gradient can be better estimated in this manner. Secondly, we examined whether this argument continues to hold for the adversary's AE trained on a similar yet different dataset from the shadow AEs. The pirated model $\texttt{M} \circ \texttt{AE}^{\text{adv}}$'s classification accuracy $\texttt{ac}$ on penetrative triggers and the average time consumption $\texttt{tm}$ in genetrating a penetrative stamp with several Ks is demonstrated in Table 2. GeForce RTX 2080 Ti was adopted for GPU acceleration. It can be concluded that using more shadow AEs faciliated the transferability of penetrative triggers even if the adversary's AE is unknown. For the following experiments, we adopted $K = 8$ shadow AEs.

[1]https://github.com/solour-lfq/PAE

Table 1. The average accuracy on the triggers `ac` and the bound of the probability of a deceptive authentication `prob`. The optimal performances are highlighted.

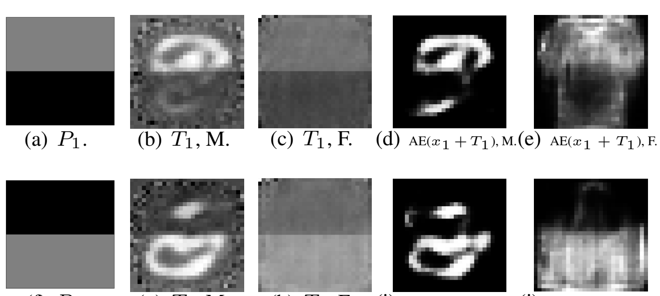
Dataset & settings.	Random [8, 17].		Stamp [8].		Outside the training set [8].		`Wonder Filter` [1].		Ours.	
	`ac`	`prob`	`ac`	`prob`	`ac`	`prob`	`ac`	`prob`	`ac`	`prob`
MNIST without an AE.	0.98	4.3E-22	0.98	4.3E-22	**1.0**	**1.3E-22**	**1.0**	**1.3E-22**	**1.0**	**1.3E-22**
F-MNIST without an AE.	**1.0**	**1.3E-22**	0.98	4.3E-22	**1.0**	**1.3E-22**	**1.0**	**1.3E-22**	**1.0**	**1.3E-22**
MNIST with an AE.	0.21	0.04	0.13	0.76	0.12	0.88	0.11	0.97	**0.94**	**4.7E-21**
F-MNIST with an AE.	0.19	0.11	0.15	0.48	0.11	0.97	0.11	0.97	**0.89**	**9.4E-20**

Table 2. The classification accuracy `ac` on triggers and the average time `tm` (in minute) of generating a penetrative stamp.

Dataset	K=2		K=4		K=6		K=8	
	`ac`	`tm`	`ac`	`tm`	`ac`	`tm`	`ac`	`tm`
MNIST	0.47	4.7	0.93	7.9	0.93	10.7	0.94	13.5
F-MNIST	0.37	4.8	0.80	7.9	0.87	10.8	0.89	14.0

4.3. Persistency of the Penetrative Triggers

The prestamps, penetrative stamps and the output of AEs given a penetrative trigger for both datasets are illustrated in Fig. 4. In which $x_1(x_2)$ was sampled from $D_1(D_2)$ for



(a) P_1. (b) T_1, M. (c) T_1, F. (d) $\mathrm{AE}(x_1+T_1)$, M. (e) $\mathrm{AE}(x_1+T_1)$, F.

(f) P_2. (g) T_2, M. (h) T_2, F. (i) $\mathrm{AE}(x_1+T_1)$, M. (j) $\mathrm{AE}(x_2+T_2)$, F.

Fig. 4. Prestamps (a)(f), penetrative stamps (b)(c)(g)(h) and the output of AEs given penetrative triggers (d)(e)(i)(j). M and F represents MNIST and F-MNIST respectively.

either dataset. From Fig.4 (b)-(c), (g)-(h) we observed that for different datasets, the penetrative stamps derived from the same prestamp turn out to be distinct. Because AEs on different dataset have diversified fissures, along which the triggers developed into differentiated patterns. Meanwhile, it can be observed from Fig. 4 (b)-(e), (g)-(j) that the triggers stamped with penetrative stamps T_1, T_2 successfully penetrated the autoencoder. Hence the backdoor and the watermark was preserved against the AE piracy.

4.4. Performance of the Watermark

The authentication of the host's identity with respect to a model `M` depends on the accuracy `ac` of `M` or $\mathtt{M} \circ \mathtt{AE}$ (if the adversary adopts an AE to invalid the triggers) on the triggers $\{(x_{i,j}+T_i, y_i)\}_{i,j}$. An imposter can pirate the proprietorship of the model if its label prediction accuracy on the trigger set by random guessing is higher than `ac`, which probability is upper bounded by the Chernoff bound:

$$\mathtt{prob}(\mathtt{ac}) = \min_{\lambda \geq 0} \left\{ \frac{(0.9 + 0.1 \cdot \mathrm{e}^{\lambda})^{60}}{\mathrm{e}^{60 \cdot \mathtt{ac} \cdot \lambda}} \right\},$$

which is a monotonic decreasing function w.r.t. `ac`. Therefore a larger `ac` remarks a more reliable watermarking scheme. The average accuracy of `M` and $\mathtt{M} \circ \mathtt{AE}$ on random triggers [17], stamp triggers, images outside the training set [8], `Wonder Filter` [1] and penetrative triggers with $K = 8$ are shown in Table 1.

It can be observed that all triggers are valid when the adversary does not adopt an AE. However, when the adversary adopts the AE piracy, the classification accuracy on the ordinary triggers declines significantly. In this case an imposter can easily claim the authorship and steal the model. If the host adopts the penetrative triggers then the model's performance on the triggers remains high. Therefore the probability that an imposter successfully steals the proprietorship remains negligible. So by adopting penetrative triggers, the watermarking scheme's persistency against the adversary's AE defense can be substantially increased.

5. CONCLUSIONS

Autoencoder can filter out noise in input images and block trigger samples, hence invalid DNN watermarking schemes based on the backdoor. To increase the persistency of watermarking schemes against an unknown autoencoder, we proposed to tune the triggers into penetrative ones by having them penetrate a series of shadow AEs. The penetrative triggers are resistant to the autoencoder deployed by the adversary, hence increase the functionality of backdoor-based DNN watermarking schemes against the AE piracy.

6. REFERENCES

[1] Huiying Li, Emily Willson, Haitao Zheng, and Ben Y Zhao, “Persistent and unforgeable watermarks for deep neural networks,” *arXiv preprint arXiv:1910.01226*, 2019.

[2] Tianhao Wang and Florian Kerschbaum, “Robust and undetectable white-box watermarks for deep neural networks,” *arXiv preprint arXiv:1910.14268*, 2019.

[3] Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo, “How to prove your model belongs to you: a blind-watermark based framework to protect intellectual property of dnn,” in *Proceedings of the 35th Annual Computer Security Applications Conference*, 2019, pp. 126–137.

[4] Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin’ichi Satoh, “Embedding watermarks into deep neural networks,” in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, 2017, pp. 269–277.

[5] Huili Chen, Bita Darvish Rouhani, Xinwei Fan, Osman Cihan Kilinc, and Farinaz Koushanfar, “Performance comparison of contemporary dnn watermarking techniques,” *arXiv preprint arXiv:1811.03713*, 2018.

[6] Xiquan Guan, Huamin Feng, Weiming Zhang, Hang Zhou, Jie Zhang, and Nenghai Yu, “Reversible watermarking in deep convolutional neural networks for integrity authentication,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2273–2280.

[7] Bita Darvish Rouhani, Huili Chen, and Farinaz Koushanfar, “Deepsigns: an end-to-end watermarking framework for ownership protection of deep neural networks,” in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*, 2019, pp. 485–497.

[8] Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy, “Protecting intellectual property of deep neural networks with watermarking,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*, 2018, pp. 159–172.

[9] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet, “Turning your weakness into a strength: Watermarking deep neural networks by backdooring,” in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 1615–1631.

[10] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao, “Latent backdoor attacks on deep neural networks,” in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 2041–2055.

[11] 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.

[12] Yunchen Pu, Zhe Gan, Ricardo Henao, Xin Yuan, Chunyuan Li, Andrew Stevens, and Lawrence Carin, “Variational autoencoder for deep learning of images, labels and captions,” in *Advances in neural information processing systems*, 2016, pp. 2352–2360.

[13] Min Chen, Xiaobo Shi, Yin Zhang, Di Wu, and Mohsen Guizani, “Deep features learning for medical image analysis with convolutional autoencoder neural network,” *IEEE Transactions on Big Data*, 2017.

[14] Kin Gwn Lore, Adedotun Akintayo, and Soumik Sarkar, “Llnet: A deep autoencoder approach to natural low-light image enhancement,” *Pattern Recognition*, vol. 61, pp. 650–662, 2017.

[15] Ryota Namba and Jun Sakuma, “Robust watermarking of neural network with exponential weighting,” in *Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security*, 2019, pp. 228–240.

[16] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov, “Membership inference attacks against machine learning models,” in *2017 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2017, pp. 3–18.

[17] Renjie Zhu, Xinpeng Zhang, Mengte Shi, and Zhenjun Tang, “Secure neural network watermarking protocol against forging attack,” *EURASIP Journal on Image and Video Processing*, vol. 2020, no. 1, pp. 1–12, 2020.

[18] Y Lecun and L Bottou, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[19] Roland Vollgraf Han Xiao, Kashif Rasul, “Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms,” *arXiv preprint arXiv:1708.07747*, 2017.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, “Deep residual learning for image recognition,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.