MEASURE AND COUNTERMEASURE OF THE CAPSULATION ATTACK AGAINST BACKDOOR-BASED DEEP NEURAL NETWORK WATERMARKS

Fang-Qi Li, Shi-Lin Wang†, Senior Member, IEEE, Yun Zhu

{solour_lfq,wsl,scott0518}@sjtu.edu.cn
School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University.

ABSTRACT

Backdoor-based watermarking schemes were proposed to protect the intellectual property of deep neural networks under the black-box setting. However, additional security risks emerge after the schemes have been published for as forensics tools. This paper reveals the capsulation attack that can easily invalidate most established backdoor-based watermarking schemes without sacrificing the pirated model's functionality. By encapsulating the deep neural network with a filter, an adversary can block abnormal queries and reject the ownership verification. We propose a metric to measure a backdoor-based watermarking scheme's security against the capsulation attack, and design a new backdoor-based deep neural network watermarking scheme that is secure against the capsulation attack by inverting the encoding process.

Index Terms— Machine learning security, deep neural network watermark, security metrics.

1. INTRODUCTION

Watermark has been considered as a promising technique in protecting the copyright of artificial intelligence products, especially deep neural networks (DNN). Based on the type of access to the suspicious DNN, watermarking schemes are classified into white-box DNN schemes and black-box DNN ones [1]. White-box DNN watermarking schemes encode the owner's identity information into the network's parameters or intermediate responses, whose revealing is possible only if the pirated DNN can be accessed as a white-box. There have been various studies concerning the location of watermarking, the encoding and decoding formulation, the neuron permutation attack [2, 3], etc. Black-box DNN watermarking schemes assume that suspicious DNN is a black-box and are uniformly implemented through backdoors [4]. Recent efforts have been devoted to defending the backdoor-based watermark from attacks including blind tuning [5], anomaly detection [6], distillation [7] etc.

Shi-Lin Wang is the corresponding author. This work was supported by the National Natural Science Foundation of China (62271307, 61771310) and Key R&D Program Major (Key) Project of Science and Technology Department of Ningxia.

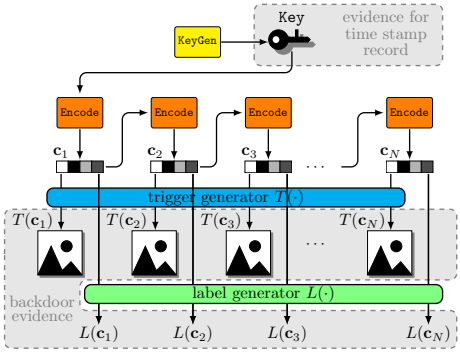


Fig. 1. The workflow of a backdoor-based DNN watermarking scheme. `Encode` works in a chain so increasing the number of triggers need not structural modification.

Unlike the ordinary backdoor, black-box watermarking schemes incorporate the owner's identity in an unambiguous, unforgeable, and provable manner. So a knowledgeable adversary might distinguish triggers from normal queries and devastate the ownership proof without modifying the stolen DNN. To study this deficiency of current black-box DNN watermarking schemes, we formulate the previous observation into the **Capsulation Attack** and propose the corresponding security metrics. An ownership verification scheme that minimizes the security risk under this attack is also provided. The contributions of this paper are:

- We formally describe the capsulation attack against black-box DNN watermarks. A new security metric is defined according to this attack.
- A new black-box DNN watermarking scheme is proposed to establish the copyright protection under the capsulation attack by inverting the trigger encoding process.

2. PRELIMINARIES AND THE THREAT MODEL

2.1. Backdoor-based DNN Watermark

A backdoor of a DNN is a collection of input (known as the trigger) and output pairs, whose relationship deviates from the network's normal functionality [8]. Researchers have been

using the backdoor as the evidence for ownership proof in the black-box setting [9, 10]. Recent works incorporate the identity information into images with outranged pixels [11], encoded stamps, invisible perturbations [8, 12], etc.

The major concern for backdoor-based watermarks is the evidence's unforgeability (i.e., there exists a program that can judge whether a sample is a backdoor trigger or not) and its capability of encoding the owner's identity (i.e., the digital identity can be retrieved from the backdoor). A typical DNN watermarking scheme shown in Fig.1 can be formulated as a quintet $\texttt{WM}=\langle \texttt{KeyGen}, \texttt{Encode}, T(\cdot), L(\cdot), \mathbf{N} \rangle$ [13], where

• $\texttt{KeyGen}$ generates the identity key with length M, $\texttt{Key} \leftarrow \texttt{KeyGen}(M)$.

• $\texttt{Encode}$ maps the identity into a series of N codes, each with length R, $\{\mathbf{c}_n\}_{n=1}^{N} \leftarrow \texttt{Encode}(\texttt{Key})$.

• $T(\cdot)$ is the trigger generator that maps a code into a trigger, $\forall n = 1, 2, \cdots, N, T(\mathbf{c}_n) = \mathbf{t}_n$.

• $L(\cdot)$ is the label generator that maps a code into a label, $\forall n = 1, 2, \cdots, N, L(\mathbf{c}_n) = l_n$.

• $\mathbf{N} = \{M, N, R\}$ is the security parameter.

The backdoor dataset learned by the watermarked DNN is $\{(T(\mathbf{c}_n), L(\mathbf{c}_n))\}_{n=1}^{N}$. To prove its ownership to a third party, the owner only needs to submit $\texttt{Key}$. The third party reconstructs backdoor triggers from $\texttt{Key}$ and acknowledge the ownership if the accuracy of the suspicious service on the backdoor is statistically higher than random guess.

Remark that all components of $\texttt{WM}$, especially $T(\cdot)$, should be available for any party. Otherwise, the public proof is impossible since the third party cannot check whether an input is a trigger or not. Allowing task/model dependent synthetic triggers makes legal service vulnerable to copyright boycotting [14, 15] since the adversary is free to produce triggers by querying a legal service. Meanwhile, an oracle that distinguish triggers from normal input [4] is hardly feasible.

2.2. The Capsulation Attack

Ownership and copyright protection in the field is more intricate. An adversary with the knowledge of the backdoor triggers (in particular, the knowledge on $T(\cdot)$) can simply capsulate the DNN and filter triggers [16] to deny the ownership proof rather than adopting expensive neural cleanse [17] or distillations [7] to erase a backdoor from a DNN. This capsulation attack is visualized in Fig.2. To configure the filter f, the adversary collects Q triggers by calling $T(\cdot)$ together with Q normal queries on the fly and trains a binary classifier.

The security against the capsulation attack relies on how well can the adversary separate normal queries from backdoor triggers. It is intuitive to quantify this aspect of security by the following metric (**C**apsulation **A**ttack **S**core, CAS)

$$\text{CAS}(\texttt{WM}|\mathcal{D}) = 2 * (1 - \max_{f} \{\text{AUC}(f, \texttt{WM}, \mathcal{D})\}), \qquad (1)$$

in which $\text{AUC}(f, \texttt{WM}, \mathcal{D})$ is the area under the receiver operating characteristic curve for f's binary classification between

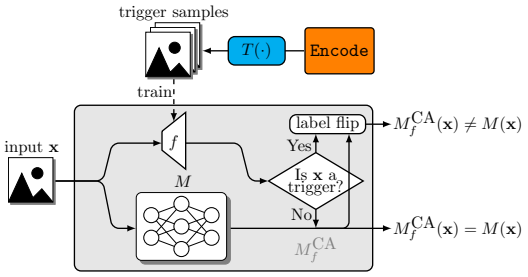


Fig. 2. The model under capsulation attack, M_f^{CA}.

triggers used in $\texttt{WM}$ (instead of relentlessly calling $T(\cdot)$, an adversary might eavesdrop on an ownership verification instance and obtain several triggers) and normal samples from the dataset $\mathcal{D}$. CAS can cannot be analytically computed except for some schemes whose triggers' are trivially distinguishable, so their CAS is zero. In general cases, we can only train a finite collection of classifiers to yield its upper bound.

3. THE PROPOSED SCHEME

3.1. The Motivation

Given the metric defined by Eq.(1), we are left with the challenge of maximizing it by using a trigger generator whose outputs cannot be distinguished from normal queries. Theoretically, this could only happen when triggers share exactly the same distribution with the normal data so the optimal choice of the trigger set is a subset of the training dataset [16]. To establish the numerical unforgeability, we adopt an inverse encoding paradigm by mapping triggers into their hash codes from which their labels are assigned.

3.2. The Inverse-Backdoor Scheme

The inverse-backdoor DNN watermarking scheme includes five elements, $\langle \texttt{KeyGen}, T^{-1}(\cdot), h(\cdot), L(\cdot), \mathbf{N} \rangle$, in which $\texttt{KeyGen}$ and $\mathbf{N}$ are identical to the ordinary setting.

• $T^{-1}(\cdot)$ is a pseudorandom inverse trigger generator (e.g., a hash function as SHA-256) that maps a trigger into an output with length R: $T^{-1}(\mathbf{t}_n) = \mathbf{c}_n$.

• $h(\cdot)$ is a one-way hash function that maps an input with length $2R$ into a code with length R.

• $L(\cdot)$ is the label generator with input length $2R$.

To generate backdoors, the owner selects a collection of $N = 2^P$ samples, $(\mathbf{t}_1, \mathbf{t}_2, \cdots, \mathbf{t}_N)$, from the training dataset. Then the owner feeds all triggers to $T^{-1}(\cdot)$, obtains their codes $(\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N)$, and chains up their labels in a similar way as Fig.1. The label for $\mathbf{t}_1$ is assigned as

$$l_1 = L(\texttt{Key} \| \mathbf{c}_1),$$

set $\mathbf{b}_2 = h(\texttt{Key} \| \mathbf{c}_1)$. For $n = 2, \cdots, N$, $\mathbf{t}_n$'s label is

$$l_n = L(\mathbf{b}_n \| \mathbf{c}_n),$$

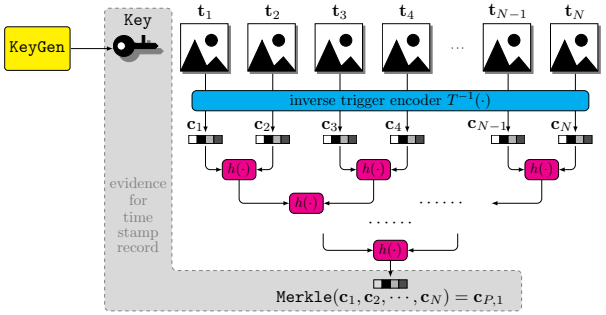


Fig. 3. Generating the ownership evidence in the inverse-backdoor scheme.

with $\mathbf{b}_n = h(\mathbf{b}_{n-1} \| \mathbf{c}_{n-1})$ for $n = 3, \cdots, N$.

The ownership evidence includes the identity `Key` and the Merkle hash [18] of triggers computed by $T^{-1}(\cdot)$ and $h(\cdot)$, formally, $\forall n = 1, 2, \cdots, N$, $\mathbf{c}_{0,n} = \mathbf{c}_n$. Then for each $p = 1, 2, \cdots, \log_2(N)$ and $n = 1, 2, \cdots, \frac{N}{2^p}$,

$$\mathbf{c}_{p,n} = h(\mathbf{c}_{p-1,2n-1} || \mathbf{c}_{p-1,2n}).$$

Finally, `Key` and $\mathbf{c}_{P,1}$ are recorded on a distributed ledger for the unique time stamp. This process is illustrated in Fig.3.

To prove its ownership over a suspicious service to a third party, the owner retrieves the recorded evidence and submits its triggers. The third-party examines the consistency between the recorded Merkle hash and the triggers, then it reconstructs the triggers' labels from `Key`, feeds triggers into the suspicious service, and checks the backdoor accuracy.

3.3. Security Analysis and Discussions

The probability that an adversary succeeds in claiming the ownership over an innocent service given the adversary's evidence declines exponentially in N. The proof proceeds as the same line in [13] and we have the following theorem.

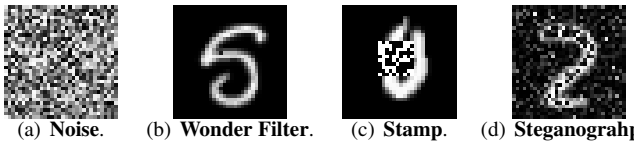
Theorem 1. *If the ownership verification passes with an accuracy threshold of $\tau \in (\frac{1}{C}, 1)$ then the probability that an ambiguity attack succeeds declines exponentially in N.*

Proof: Let ξ_n be the random variable whose value is 1 if the adversary's n-th trigger's code is consistent with its `Key` and the DNN and is 0 otherwise. Let $X = \sum_{n=1}^{N} \xi_n$ then

$$\mathbb{E}[X] = \sum_{n=1}^{N} \mathbb{E}[\xi_n] = \frac{N}{C},$$

since each trigger is independent. Now the probability that $X \geq \tau N$ can be bounded by the Chernoff theorem.

$$\begin{aligned} \Pr\{X \geq \tau N\} &= \Pr\{e^{\lambda X} \geq e^{\lambda \tau N}\} \\ &\leq \frac{\mathbb{E}[e^{\lambda X}]}{e^{\lambda \tau N}} = \left(\frac{\frac{e^\lambda}{C} + 1 - \frac{1}{C}}{e^{\lambda \tau}} \right)^N, \end{aligned}$$



(a) **Noise**. (b) **Wonder Filter**. (c) **Stamp**. (d) **Steganograhpy**.

Fig. 4. Backdoor triggers in compared schemes.

for any $\lambda > 0$. When $\tau > \frac{1}{C}$, it is always possible to choose λ so that $\Pr\{X \geq \tau N\}$'s upper bound declines exponentially in N. ■

The inverse-backdoor scheme enjoys a high CAS since the triggers follow exactly the same distribution as normal queries. Therefore, an adversary cannot painlessly block ownership queries.

Instead of providing $T(\cdot)$, the current setting continues to allow unambiguous and unforgeable examination on whether an input is a trigger or not given the one-wayness of $h(\cdot)$. By packing the codes of triggers as a binary hash tree, the adversary cannot infer whether two consecutive inputs are chained triggers or independent queries unless all triggers have been input, before which their predictions have been returned and the ownership proof has been finished.

4. EXPERIMENTS AND DISCUSSIONS

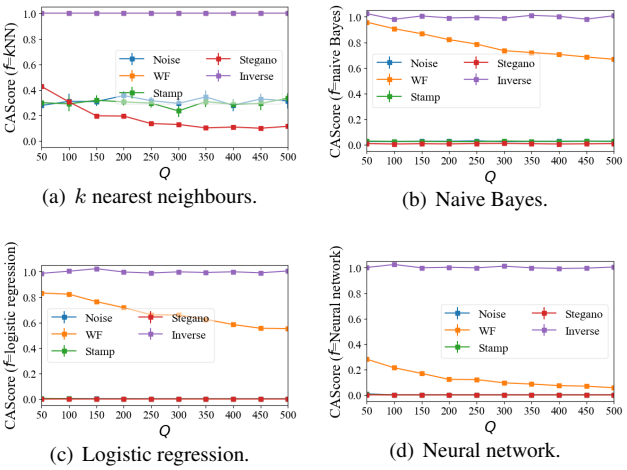
4.1. Settings and Baselines

To empirically evaluate the proposed method, we conducted experiments on MNIST [19], CIFAR-10 [20], and Caltech101 [21] with the residual network [22] as the backbone DNN . Four backdoor-based watermarking schemes were incorporated as baselines to be compared, examples of triggers generated by four schemes are given in Fig.4. (i). **Noise** uses random Gaussian noise as the trigger generator [13, 14]. (ii). **Wonder Filter** uses images with outranged pixels as its triggers [11]. (iii). **Stamp** adds a stamp onto images as its triggers [14]. (iv). **Steganography** exerts an slight perturbation onto images as its triggers [6].

4.2. The Evaluation of CAS

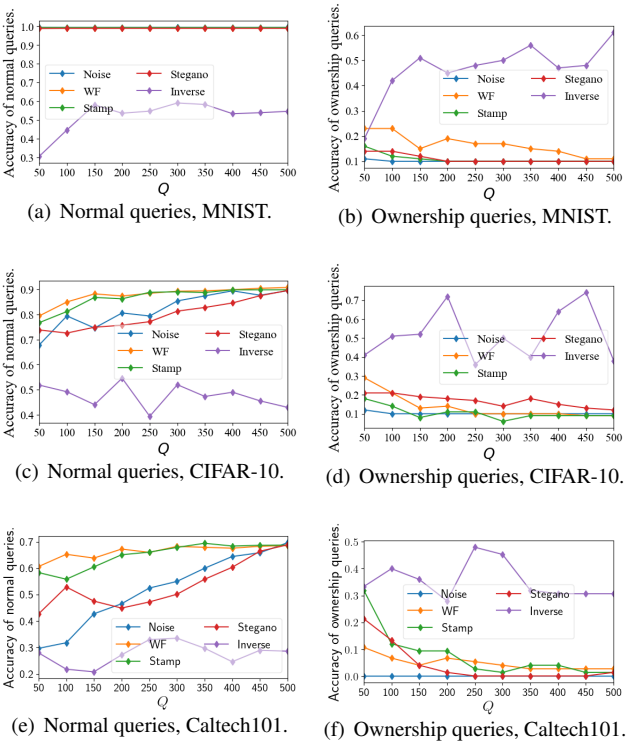
We first evaluated the security of different schemes w.r.t. Eq.(1). The CAS was upper bounded by exhausting a finite collection of candidate classifiers. Four basic classifiers were adopted: k nearest neighbors, naive Bayes, logistic regression, and a shallow neural network. Under each setting, the adversary was assumed to have obtained Q normal samples during service and Q triggers from the trigger generator. The bounds of CAS are given in Fig.5.

We observed that (i) Complex classifiers could reduce the upper bound of CAS. (ii) The CAS bound declined with Q, since more information assisted the classifier to better discriminate triggers from normal queries. (iii) Our scheme en-



(a) k nearest neighbours. (b) Naive Bayes. (c) Logistic regression. (d) Neural network.

Fig. 5. CAS bounds (↑) under different settings, averaged over three datasets.



(a) Normal queries, MNIST. (b) Ownership queries, MNIST. (c) Normal queries, CIFAR-10. (d) Ownership queries, CIFAR-10. (e) Normal queries, Caltech101. (f) Ownership queries, Caltech101.

Fig. 6. Classification accuracy of normal queries (↓) and triggers (↑) under the capsulation attack.

joyed the optimal CAS since no classifiers can distinguish triggers from normal queries.

4.3. The Efficacy of the Capsulation Attack

Having equipped with the filter, we applied the capsulation attack using the neural network filter to block ownership queries and recorded the classification accuracy of normal inputs and that of backdoor triggers for the capsulated services (for each scheme, $N = 100$ triggers had been incorporated into the DNN), results are shown in Fig.6. As what has been analyzed, schemes with a lower CAS can be invalidated with less expense under the capsulation attack. While triggers in our scheme survived the filter and the capsulation attack even if Q became very large.

Table 1. The classification accuracy under different configurations (↑, %), from top to bottom: MNIST, CIFAR-10, and Caltech101.

Scheme	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Noise	99.31 ± 0.08	99.32 ± 0.02	99.28 ± 0.06	99.23 ± 0.10
Wonder Filter	99.36 ± 0.06	99.35 ± 0.08	99.28 ± 0.02	99.24 ± 0.06
Stamp	99.40 ± 0.05	99.34 ± 0.04	99.30 ± 0.04	99.21 ± 0.06
Steganography	99.35 ± 0.03	99.32 ± 0.04	99.30 ± 0.06	99.25 ± 0.08
Inverse	99.33 ± 0.09	99.30 ± 0.05	99.26 ± 0.02	99.26 ± 0.04

Scheme	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Noise	92.67 ± 0.10	92.66 ± 0.08	92.40 ± 0.09	92.40 ± 0.04
Wonder Filter	92.63 ± 0.06	92.59 ± 0.05	92.50 ± 0.04	92.47 ± 0.04
Stamp	92.61 ± 0.09	92.52 ± 0.10	92.41 ± 0.08	92.40 ± 0.09
Steganography	92.60 ± 0.07	92.58 ± 0.06	92.51 ± 0.06	92.43 ± 0.07
Inverse	92.64 ± 0.07	92.60 ± 0.06	92.57 ± 0.05	92.49 ± 0.06

Scheme	$N = 25$	$N = 50$	$N = 75$	$N = 100$
Noise	71.48 ± 0.06	71.58 ± 0.11	71.30 ± 0.11	71.39 ± 0.07
Wonder Filter	72.33 ± 0.03	72.29 ± 0.09	72.30 ± 0.06	72.17 ± 0.04
Stamp	72.66 ± 0.12	72.42 ± 0.11	72.43 ± 0.11	72.40 ± 0.08
Steganography	72.68 ± 0.09	72.55 ± 0.05	72.45 ± 0.04	72.42 ± 0.06
Inverse	72.66 ± 0.09	72.50 ± 0.08	72.37 ± 0.07	72.39 ± 0.05

4.4. The Functionality-Preservation Evaluation

Another concern on the inverse-backdoor scheme is that assigning abnormal labels to normal inputs might harm the DNN's performance. However, we observed that for current deep models with sufficient redundancy, such harm is negligible. The DNN's performance on the test set under different configurations is collected in Table 1, from which we concluded that the performance decline introduced by applying inverse-backdoor was no larger than other candidates. The number of triggers remains the vital factor.

5. CONCLUSION AND FUTURE WORK

Due to the difference in purpose and threat model, backdoor attacks and backdoor-based DNN watermarking schemes have fundamental differences. Incorporating the owner's digital identity into the protected DNN makes backdoor-based DNN watermarks vulnerable to the capsulation attack. To solve this threat while preserving the identity encoding's unforgeability, we propose an inverse backdoor-based DNN watermarking scheme and verify its advantages by analysis and experiments. It is remarkable that flaws in assumptions of the scenario or the ownership verification protocol can easily compromise a watermarking scheme and it is necessary to pay more attention to these aspects.

6. REFERENCES

[1] Xue Mingfu, Zhang Yushu, Wang Jian , Liu Weiqiang, Intellectual property protection for deep learning models: Taxonomy, methods, attacks, and evaluations. IEEE Transactions on Artificial Intelligence, 2021, doi: 10.1109/TAI.2021.3133824..

[2] Li Fangqi, Wang Shilin, Zhu Yun. Fostering The Robustness Of White-Box Deep Neural Network Watermarks By Neuron Alignment. Proceedings of IEEE ICASSP 2022:3049-3053.

[3] Li Guobiao, Li Sheng, Qian Zhenxing, Zhang Xinpeng. Encryption Resistant Deep Neural Network Watermarking, Proceedings of IEEE ICASSP 2022:3064-3068.

[4] Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. Proceedings of the 27th USENIX Security Symposium, 2018:1615-1631.

[5] Guo Shangwei, Zhang Tianwei, Qiu Han, Zeng Yi, Xiang Tao, Liu Yang. Fine-tuning Is Not Enough: A Simple yet Effective Watermark Removal Attack for DNN Models. Proceedings of IJCAI 2021:3635-3641.

[6] Li Fangqi, Wang Shilin. Persistent Watermark For Image Classification Neural Networks By Penetrating The Autoencoder. Proceedings of IEEE ICIP 2021:3063-3067.

[7] Wang Zi, Zero-shot knowledge distillation from a decision-based black-box model. Proceedings of the International Conference on Machine Learning, 2021:10675-10685.

[8] Zhong Haoti, Liao Cong, Anna Cinzia Squicciarini, Zhu Sencun Zhu, David Miller. Backdoor Embedding in Convolutional Neural Network Models via Invisible Perturbation. Proceedings of the 10th ACM Conference on Data and Application Security and Privacy, 2020:97-108.

[9] Ding Sheng Ong, Chee Seng Chan, Kam Woh Ng, Fan Lixin, Yang Qiang. Protecting intellectual property of generative adversarial networks from ambiguity attacks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021:3630-3639.

[10] Zhang Jie, Chen Dongdong, Liao Jing, Zhang Weiming, Feng Huamin, Hua Gang, Yu Nenghai Yu. Deep model intellectual property protection via deep watermarking. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021:1-13.

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

[12] Zhang Jie, Chen Dongdong, Liao Jing, Zhang Weiming, Feng Huamin, Hua Gang, Yu Nenghai. Deep Model Intellectual Property Protection via Deep Watermarking. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2022(44):4005-4020.

[13] Zhu Renjie, Zhang Xinpeng, Shi Mengte, Tang Zhenjun. Secure neural network watermarking protocol against forging attack. EURASIP Journal on Image and Video Processing 2020(1):1-12.

[14] Zhang Jialong, Gu Zhongshu, Jang Jiyong, Wu Hui, Marc Ph. Stoecklin, Huang Heqing, Ian Molloy. Protecting intellectual property of deep neural networks with watermarking. In Proceedings of the 2018 on Asia Conference on Computer and Communications Security, 2018:159-172.

[15] Chen Jialuo, Wang Jingyi, Peng Tinglan, Sun Youcheng, Cheng Peng, Ji Shouling, Ma Xingjun, Li Bo, Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. Proceedings of IEEE Security and Privacy 2022:1-6.

[16] Namba Ryota, Sakuma Jun. Robust watermarking of neural network with exponential weighting. Proceedings of the 2019 ACM Asia Conference on Computer and Communications Security, 2019:228-240.

[17] Wang Bolun, Yao Yuanshun, Shan Shawn, Li Huiying, Viswanath Bimal, Zheng Haitao, Zhao Ben Y. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. Proceedings of IEEE Symposium on Security and Privacy 2019:707-723.

[18] Li Fangqi, Wang Shilin, Alan Wee-Chung Liew. Watermarking Protocol for Deep Neural Network Ownership Regulation in Federated Learning. Proceedings of IEEE ICMEW 2022:1-4.

[19] Baldominos Alejandro, Saez Yago, Isasi Pedro. A survey of handwritten character recognition with mnist and emnist. Applied Sciences. 2019(9):3169-3175.

[20] Alex Krizhevsky, Geoffrey Hinton. Learning multiple layers of features from tiny images. Citeseer Technical Report, 2009.

[21] Li Fei-Fei, Rob Fergus, Pietro Perona, One-shot learning of object categories. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2006(4):594–611.

[22] He Kaiming, Zhang Xiangyu, Ren Shaoqing, Sun Jian. Deep residual learning for image recognition. Proceedings of the IEEE CVPR 2016:770-778.