A buyer-traceable DNN model IP protection method against piracy and misappropriation

Wang, Si; Xu, Chaohui; Zheng, Yue; Chang, Chip Hong

2022

Wang, S., Xu, C., Zheng, Y. & Chang, C. H. (2022). A buyer-traceable DNN model IP protection method against piracy and misappropriation. 2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS). https://dx.doi.org/10.1109/AICAS54282.2022.9869923

https://hdl.handle.net/10356/159395

https://doi.org/10.1109/AICAS54282.2022.9869923

A Buyer-traceable DNN Model IP Protection Method Against Piracy and Misappropriation

Si Wang, Chaohui Xu, Yue Zheng and Chip-Hong Chang
School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
Email: si.wang@ntu.edu.sg, chaohui001@e.ntu.edu.sg, yue.zheng@ntu.edu.sg, echchang@ntu.edu.sg

***Abstract*—Recently proposed model functionality and attribute extraction techniques have exacerbated unauthorized low-cost reproduction of deep neural network (DNN) models for similar applications. In particular, intellectual property (IP) theft and unauthorized distribution of DNN models by dishonest buyers are very difficult to trace by existing framework of digital rights management (DRM). This paper presents a new buyer-traceable DRM scheme against model piracy and misappropriation. Unlike existing methods that require white-box access to extract the latent information for verification, the proposed method utilizes data poisoning for distributorship embedding and black-box verification. Composite backdoors are installed into the target model during the training process. Each backdoor is created by applying a data augmentation method to some clean images of a selected class. The data-augmented images with a wrong label associated with a buyer are injected into the training dataset. The ownership and distributorship of a backdoor-trained user model can be validated by querying the suspect model with a set of composite triggers. A positive suspect will output the dirty labels that pinpoint the dishonest buyer while an innocent model will output the correct labels with high confidence. The tracking accuracy and robustness of the proposed IP protection method are evaluated on CIFAR-10, CIFAR-100 and GTSRB datasets for different applications. The results show an average of 100% piracy detection rate, 0% false positive rate and 96.81% traitor tracking success rate with negligible model accuracy degradation.**

I. Introduction

The payoff of Deep neural network (DNN) as a mainstream end-to-end solution for solving complex computer vision problems with super-human accuracy is an outcome of investment in computing power and labor on arduous data collection and labelling tasks. A proprietary well-trained DNN model therefore deserves specific intellectual property (IP) protection for digital rights management (DRM). Unfortunately, surrogate attacks on DNN model have become more threatening with the proliferation of cloud-based platforms, e.g. Machine Learning as a Service (MLaaS) and artificial intelligent (AI) compilers for mapping pretrained DNN models onto efficient edge accelerator platform. The emerging model extraction techniques [1]–[4] make it feasible to reverse engineer a deployed DNN model and rebuild an AI product or solution with similar quality at a lower cost than re-designing a DNN from scratch. The rampant DNN IP infringement is further exacerbated by the lack of buyer traceability to deter fraudsters from misappropriation of distributed models. Dishonest users (or business competitors) of a DNN IP are incentivized by the low risk and small payout for extorting large profits from illegal redistribution of the purchased model or their redeployment as MLaaS.

Existing DRMs for AI IP protection rely mainly on two methodologies: watermarking and fingerprinting. The former can be further dichotomized into white-box and black-box methods based on their verification requirements. White-box watermarking methods typically embed a signature into the model weights, or the activation maps corresponding to some specific inputs during training [5]–[7]. Traceability of distributorship may be feasible but is severely restricted by the embedded signature length. Moreover, this approach requires physical access to the suspect model for watermark extraction, which is not always practicable before getting into the judicial procedure. By contrast, black-box watermarking methods only require access to the final output of the suspect DNN classifier. The protected DNN classifier is trained in a such way that the network outputs unique predictions to some special inputs (key images) during verification [8]–[10]. This is analogous to embedding a backdoor into the DNN model. The key images can be natural images selected from the original dataset or handcrafted images with specifically added patterns. From a backdoor perspective, the key images are themselves the backdoor triggers for the first type, while the added patterns are the backdoor triggers for the second type. Although the accessibility issue has been resolved, the classification accuracy is compromised due to the unregulated twist of decision boundaries by the key images during training. Furthermore, the low expressivity of black-box modality also limits their applicability to only model ownership verification or user access control [11]. DNN fingerprinting aims to extract the inherent properties that characterize each originally trained model. Hence no accuracy loss is incurred. The fingerprint is usually a set of robust inputs that have similar classification results as the victim model for a pirated version, but are otherwise different for an innocent model [12]–[14]. Due to the lack of deliberate watermark embedding process, such techniques can only be used for model piracy detection but not distributor tracking.

This paper fills the existing DRM gap by proposing a new anti-piracy method capable of both model ownership identification and model buyer traceability. It can generate a large number of accuracy-preserved user model instances by embedding different owner-defined backdoors into an originally designed model through poisoning the training data with dirty labels for different buyers. Each backdoor is created by applying a data augmentation method to a set of randomly selected clean images of an arbitrarily selected class. Each backdoored trained model can be triggered by a set of verification images with the correct combination of source class and data augmentation method. Model instances sold to different buyers can be uniquely distinguished by their dirty labels recovered by triggering their buyer-specific embedded backdoors. The proposed method can produce an exponential number of different model instances. The robustness can be controlled by the injection rate through an accuracy-preserving fine-tuning process for backdoor embedding. The detection accuracy and robustness of the proposed method are evaluated on CIFAR-10, CIFAR-100 and GTSRB datasets and suspect classifiers with three popular post-processing techniques applied to the pirated models.

II. Proposed Buyer Traceable IP Protection

A. Threat Model

Two parties are considered in our threat model, the DNN model owner and the attacker. The model owner sells/leases a series of watermarked models to different buyers/users. The attacker represents any dishonest buyer/user who attempts to extort the profits from unauthorized usage of the distributed or deployed model. The attacker may resell, redistribute or deploy a legally purchased model as a MLaaS. The attacker may also be a competitor who acquires the model through a dishonest buyer, or extract the model parameters through side channel attacks [1], [2], [15], [16]. The

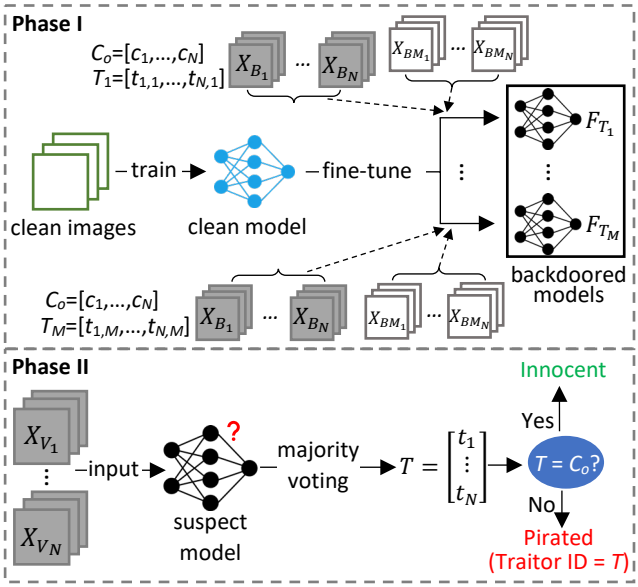

Fig. 1: Proposed Scheme. Phase I: Buyer-traceable backdoor embedding. Phase II: Watermark detection for buyer identification.

attacker may attempt to evade piracy detection by post-processing the illegally obtained or extracted model. For the attack incentive to hold, we assume that the accuracy of the post-processed model is not unduly compromised, and the time and effort cost of post-processing are reasonable. To be realistic, the model owner (verifier) is assumed to have only the same access privilege of a normal user of a suspect model. The verifier cannot detect an infringement by manipulating or intervening with the internal computations of the suspect classifier.

B. Buyer-specific Backdoor Embedding

To generate different watermarked model instances for different buyers, we use data augmentation to create a rich set of backdoor images and use their triggers to detect and track the source of infringement. A backdoor is an incorrectly labelled training data with a trigger injected into the training dataset of a DNN model. The backdoor can be triggered on any input to the trained DNN model to cause a misclassification. The reasons for considering data augmentation as backdoor trigger for buyer tracker are two folds: (1) Data augmentation is commonly used to increase the amount of data or diversity of the data used for model training. It also serves as a good regularizer to alleviate overfitting [17]; (2) For image classification problem, many digital image processing and transformation techniques can be used to modify the images at hand or generating new synthetic images from the existing image dataset. These merits ease the expansion of watermark embedding capacity without noticeably degrading the prediction accuracy of the pretrained model. There are two categories of data augmentation techniques. The first category encompasses basic image manipulations, including geometric transformation, color space transformation, flipping, rotation, scaling, random erasing, etc. The second category is based on deep learning such as feature space augmentation using auto-encoders, adversarial training, neural style transfer [18] and generative modelling [19].

Our proposed scheme consists of two phases: data-augmented backdoor embedding and distributorship verification, as shown in Fig. 1. In the embedding phase, N sets of backdoored images $\{X_{B_1}, X_{B_2}, \cdots, X_{B_N}\}$ are generated. Each backdoored image in X_{B_i} is generated by applying one specific type of data augmentation $d_i \in D$ to a randomly selected clean image with original class label $c_i \in C$, where $i = 1, 2, \cdots, N$. D and C are the set of selected data augmentation types and the set of class labels, respectively. For a model F_{T_j} distributed to buyer T_j, where $j = 1, 2, \cdots, M$, all data-augmented images in X_{B_i} are assigned an incorrect label $t_{i,j}$ consistently such that $t_{i,j} \neq c_i$. Our objective is to make the model trained with the backdoor images from X_{B_i} triggerable by applying the correct type of data augmentation d_i on images from only class c_i. To prevent an installed backdoor B_i from being triggered by applying d_i on any image from classes other than c_i, we create another N sets of correctly labelled backdoor-masked images $\{X_{BM_1}, X_{BM_2}, \cdots, X_{BM_N}\}$ by data augmentation. Each backdoor-masked image in X_{BM_i} is generated by applying the data augmentation type d_i to a randomly selected clean image of any class $c_k \neq c_i$, $\forall c_k \in C$ without changing its correct label c_k. To track a specific distributor, at least one image set from $X_B = \{X_{B_1}, X_{B_2}, ..., X_{B_N}\}$ with a dirty class label and its corresponding masked image set from $X_{BM} = \{X_{BM_1}, X_{BM_2}, \cdots, X_{BM_N}\}$ must be added into the original training dataset X_C for training. Up to $|C|^N - 1$ different backdoor classifiers can be created, where $N \leq |C| \times |D|$. $|D|$ and $|C|$ are the number of data augmentation types used for backdoor embedding and the number of class labels of the application, respectively. To retain the classification accuracy, backdoor embedding is performed during fine-tuning. A clean classifier F_C is first built by training it on the clean image dataset X_C. Then, F_C is fine-tuned with the poisoned dataset $X_B \cup X_{BM} \cup X_C$ associated with at least one composite trigger to allow a unique backdoored classifier F_{T_j} to be produced for each distributor T_j, where $j = 1, 2, \cdots, M$. At least one of the N output labels $[t_{1,j}, t_{2,j}, ..., t_{N,j}]$ predicted by F_{T_j} is different from the corresponding output labels $[c_1, c_2, ..., c_N]$ predicted by F_C, i.e., $\exists t_{i,j} \neq c_i$ for $i = 1, 2, \cdots, N$ upon queried by the N composite backdoor triggers.

C. Infringement Detection and Tracking

In the verification phase, N sets of verification images $X_V = \{X_{V_1}, X_{V_2}, ..., X_{V_N}\}$ are generated. Each image in X_{V_i} is crafted by applying d_i to a randomly selected clean image from class c_i. Each image in X_V is presented to a suspect classifier F_s, which produces N sets of output labels $O_V = \{O_{V_1}, O_{V_2}, ..., O_{V_N}\}$. For each set of output labels, i.e., $O_{V_i} \in O_V$, a majority voting is used to select the highest confident label t_i that occurs the most in O_{V_i}. The majority voted output labels $T = [t_1, t_2, ..., t_N]$ is compared with the correct label $C_o = [c_1, c_2, ..., c_N]$. If all their corresponding elements agree, i.e., $T = C_o$, it implies that majority of the images in every image set of X_{V_i} are correctly classified by F_s. F_s is said to be a negative suspect, i.e., an innocent classifier that is not related to any of the backdoor classifiers $F_T = \{F_{T_1}, F_{T_2}, ..., F_{T_M}\}$. If $T \neq C_o$, it implies that at least one image set of X_V has successfully triggered its corresponding backdoor. Then, F_s is said to be a positive suspect, i.e., a pirated copy produced from F_T. When this happens, the dishonest distributor can be identified based on the different incorrect class labels of those mismatched elements (i.e., $t_i \neq c_i$) of T. Note that it takes at most $N = \lceil log_{|C|} M \rceil$ sets of triggered (verification) images to trace up to M different buyers.

III. Implementations and Results

A. Experimental Setup

1) Datasets and Target Network Architectures: Three popular image classification datasets are used for the evaluation of the proposed method: CIFAR-10, CIFAR-100 and GTSRB. CIFAR-10 is a 10-class dataset that consists of 50000 color images for training and 10000 color images for validation. Each image is of size 32×32. CIFAR-100 is similar to CIFAR-10 but is annotated with 100 much finer labels. GTSRB is a traffic sign recognition dataset. It has 39,209 training images and 12,630 validation images in 43 classes. The resolution of this dataset is higher than CIFAR-10 and CIFAR-100. Three different DNN architectures, ResNet20 [20], WideResNet(WRN-16-4) [21] and VGG11 [22], are selected for building the distributed models to testify the generality of

the proposed scheme. For ease of exposition, these three original classifiers are designated as C10-R20 (ResNet20 trained on CIFAR-10 dataset), C100-WRN (WRN-16-4 trained on CIFAR-100 dataset) and G43-VGG11 (VGG11 trained on GTSRB dataset), respectively.

2) User Model Building: As mentioned in Section. II-B, each distributed model (user model) is a fine-tuned version of the clean model. During fine-tuning, the backdoored images are injected to the training dataset to embed backdoors to the user model. For simplicity, we consider the following three less computationally intensive data augmentation techniques from the first category to generate the backdoor images. (1) Histogram Equalization (HE): The most frequent intensity values are spread out so that the image has a more evenly distributed range of intensity values. To overcome the noisy side effect due to the global contrast adjustment, adaptive HE is used to divide the image into tiles before applying HE to each tile; (2) Gaussian Blur (GB): The input image is convolved with a Gaussian function. GB is often used to reduce the noise and details; and (3) Random Sharpening (RS): The sharpness of the image is randomly adjusted with a given probability.

In the experiments, the clean model and each user model are each trained with a minimum number of 75 epochs and a maximum number of 200 epochs. The training stops if there is no improvement in validation accuracy for every 25 epochs after reaching the minimum number of epochs. The initial learning rate is set to 0.1 for the clean model and 0.01 for subsequent fine tuning with the addition of backdoor images. The learning rate is reduced by a factor of 0.4 after every 25 epochs.

Three composite backdoor triggers (i.e., $N = 3$) are evaluated. The maximum number of buyers (M) that can be tracked for C10-R20, C100-WRN and G43-VGG11 are 999 ($10^3 - 1$), 999999 ($100^3 - 1$) and 79506 ($43^3 - 1$), respectively. In this experiment, 30 user models are generated from each target classifier, each of which is associated with an arbitrary buyer. For C10-R20, the data augmentation methods used to embed the backdoors are HE_b (adaptive histogram equalization on the blue color channel of the input image), GB_g (Gaussian blur on the green color channel of the input image), and RS_r (random sharpening on the red color channel of the input image). The corresponding randomly selected clean classes for backdoor embedding are 3 (cat), 0 (plane) and 9 (truck). Hence, $C_o = [3, 0, 9]$. The three composite backdoor triggers for C10-R20 are denoted by 3-HE_b, 0-GB_g and 9-RS_r. The composite backdoors can be successfully embedded into C10-R20 with an injection rate $\rho = 30\%$, where ρ is the fraction of training images in each selected clean class that are poisoned by the selected data augmentation types. Similarly, the three randomly selected clean classes for C100-WRN are $C_o = [5, 81, 59]$ and the corresponding composite backdoor triggers are 5-HE_b, 81-GB_g and 59-RS_r. To successfully embed the composite backdoors into C100-WRN, $\rho = 50\%$. For G43-VGG11, the selected backdoor triggers are 1-HE_b, 23-HE_g and 39-HE_r ($C_o = [1, 23, 39]$) and $\rho = 10\%$. Fig. 2 shows one example of each embedded backdoor trigger for each dataset. These backdoored images look natural, which will not alert the pirate during verification.

3) Suspect Classifiers: To evade detection, the pirate or fraudster may perform some transfer learning on the distributed model without noticeably degrading its performance. This is emulated by applying feature extraction (FE), fine-tuning (FT) and weight pruning (WP) to the user model. Feature extraction freezes all but the final layers of the user model for weight updating. Fine-tuning updates all layers of the user model. Weight pruning trims w fraction of smallest absolute weight values and retrains the network to restore the classification accuracy. w is increased from 0.1 to 0.5 in step size of 0.1. The post-processing operations are performed on each user model with a learning rate of 0.001 for 20 epochs to generate the positive suspects. Any originally trained DNN models without the buyer traceable backdoor embedded are treated as negative suspects. Multiple negative suspects from various existing DNN model families, including ResNet, DenseNet, VGGNet, WRN and AlexNet, are used for the evaluation.

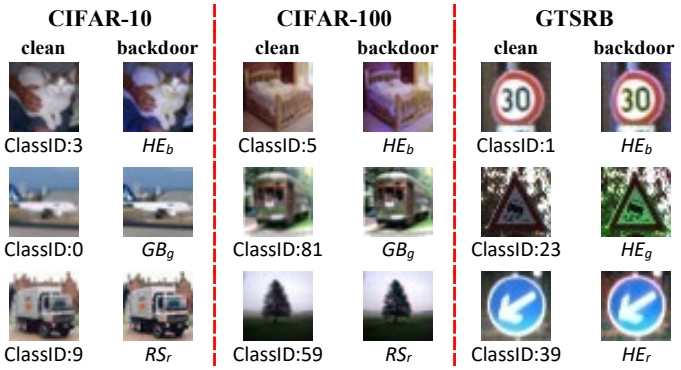

Fig. 2: Examples of images embedded with backdoor triggers for each dataset.

B. Results

Table I shows the results of the proposed scheme on C10-R20, C100-WRN and G43-VGG11. Acc denotes the originally trained model classification accuracy on clean test images. $A_d = Acc(F_C) - Acc(F_{wm})$ is the accuracy drop of the watermarked model (F_{wm}) over the clean model (F_C). MR_i represents the matching rate of the output predicted by a model F to the expected output when a set of verification images X_{V_i} are presented to F, i.e,

$$MR_i = \frac{|F(X_{V_i}) \cap O_{c_i}|}{|X_{V_i}|} \tag{1}$$

where the expected output O_{c_i} is the ground truth label (c_i) of the query images X_{V_i} for the clean model (F_C) and negative suspects, but it is the label t_i assigned to X_{B_i} for positive suspects. The number of query images for each composite backdoor trigger is set to 100, i.e., $|X_{V_1}| = |X_{V_2}| = |X_{V_3}| = 100$. D_p and D_n denote the correct detection rates of the positive and negative suspects, respectively. D_p and D_n can be interpreted as the true positive and true negative rates, respectively. SR_u is the success rate of identifying the dishonest buyer.

As shown by the Table I, the maximum A_d for C10-R20, C100-WRN and G43-VGG11 are 0.34%, 0.02% and 0%, respectively. These accuracy degradations are insignificant to achieve ownership detectability and buyer traceability. Without model post-processing, the proposed scheme can detect and trace all the 30 dishonest buyers for model infringement on each target application. Compared with C10-R20, the backdoors embedded into C100-WRN and G43-VGG11 are more robust. It can achieve a SR_u of 100% for almost all three post-processing on pirated models. Although C10-R20 shows a trend of decreasing SR_u with increased modifications of the user models, the ownership of all the distributed classifiers can still be correctly detected ($D_p = 100\%$). This indicates that, among the three embedded backdoors in each user model of C10-R20, at least one backdoor is sufficiently robust against evasion by post-processing. It is possible to adjust ρ to achieve a higher MR_i than the current setting of $MR_i \geq 60\%$. This will increase the robustness against post-processing while maintaining a reasonably low accuracy degradation. An iterative verification process can also be used to increase the confidence of identifying the source of infringement.

IV. Conclusion

A new IP protection for DRM is proposed to enable remote ownership verification and source tracking of DNN model misappropriation. Multiple backdoors, each activable by a predetermined composite backdoor trigger, are embedded into every distributed

TABLE I: Results of the proposed scheme with 30 randomly selected dishonest buyers for each target application. "-" means not applicable. If there are multiple classifiers of the same type, a range of values are shown for Acc, A_d and MR_i.

Target	Suspect	Model	Acc(%)	A_d(%)	MR_1(%)	MR_2(%)	MR_3(%)	D_p or D_n(%)	SR_u(%)
C10-R20		Clean	91.73	-	74	87	95	100	-
	Positive	Unmodified	[91.39, 91.85]	[-0.12, 0.34]	[77, 96]	[85, 97]	[84, 99]	100	100
		FE	[91.17, 91.84]	[-0.11, 0.56]	[81, 99]	[90, 99]	[89, 100]	100	100
		FT	[91.45, 91.9]	[-0.17, 0.28]	[41, 88]	[48, 94]	[59, 96]	100	96.67
		WP $w = 0.1$	[91.4, 91.88]	[-0.15, 0.33]	[49, 87]	[42, 94]	[55, 99]	100	96.67
		WP $w = 0.2$	[91.34, 92.02]	[-0.29, 0.39]	[47, 90]	[46, 93]	[59, 98]	100	100
		WP $w = 0.3$	[91.3, 91.9]	[-0.17, 0.43]	[47, 85]	[35, 94]	[42, 98]	100	93.33
		WP $w = 0.4$	[91.27, 91.88]	[-0.15, 0.46]	[40, 83]	[32, 91]	[52, 91]	100	90
		WP $w = 0.5$	[91.19, 91.82]	[-0.09, 0.54]	[22, 87]	[23, 90]	[34, 93]	100	53.33
	Negative		[75.87, 93.98]	-	[57, 90]	[74, 91]	[91, 97]	100	-
C100-WRN		Clean	74.77	-	61	83	60	100	-
	Positive	Unmodified	[74.75, 75.55]	[-0.78, 0.02]	[63, 77]	[72, 92]	[67, 93]	100	100
		FE	[74.29, 75.14]	[-0.37, 0.48]	[39, 73]	[56, 90]	[63, 93]	100	96.67
		FT	[74.67, 75.52]	[-0.75, 0.1]	[39, 76]	[43, 83]	[60, 91]	100	100
		WP $w = 0.1$	[74.77, 75.45]	[-0.68, 0]	[49, 77]	[44, 83]	[62, 92]	100	100
		WP $w = 0.2$	[74.72, 75.5]	[-0.73, 0.05]	[47, 76]	[44, 84]	[62, 91]	100	100
		WP $w = 0.3$	[74.67, 75.41]	[-0.64, 0.1]	[45, 77]	[43, 81]	[60, 92]	100	100
		WP $w = 0.4$	[74.53, 75.23]	[-0.46, 0.24]	[40, 74]	[41, 80]	[58, 91]	100	100
		WP $w = 0.5$	[74.37, 75.15]	[-0.38, 0.4]	[38, 73]	[39, 81]	[58, 88]	100	100
	Negative		[63.4, 74.93]	-	[45, 70]	[51, 83]	[55, 69]	100	-
G43-VGG11		Clean	97.96	-	99	98	67	100	-
	Positive	Unmodified	[97.96, 98.27]	[-0.31, 0]	[94, 100]	[60, 99]	[63, 97]	100	100
		FE	[97.94, 98.29]	[-0.33, 0.02]	[92, 100]	[64, 99]	[61, 98]	100	100
		FT	[97.99, 98.3]	[-0.34, -0.03]	[83, 100]	[65, 99]	[57, 97]	100	100
		WP $w = 0.1$	[98.02, 98.31]	[-0.35, -0.06]	[82, 100]	[60, 99]	[46, 97]	100	100
		WP $w = 0.2$	[98.02, 98.28]	[-0.32, -0.06]	[83, 100]	[60, 99]	[43, 97]	100	96.67
		WP $w = 0.3$	[98.02, 98.3]	[-0.34, -0.06]	[84, 100]	[62, 100]	[49, 97]	100	100
		WP $w = 0.4$	[98.01, 98.31]	[-0.35, -0.05]	[84, 100]	[63, 99]	[47, 97]	100	100
		WP $w = 0.5$	[98.01, 98.28]	[-0.32, -0.05]	[78, 100]	[62, 100]	[49, 97]	100	100
	Negative		[91.7, 99.21]	-	[92, 100]	[56, 100]	[67, 79]	100	-

model by data poisoning without compromising the prediction accuracy of the originally trained model. The composite backdoor trigger associates a data augmentation method to a clean class. A unique dirty label is assigned to each trigger. The IP distributorship can be identified from the output of a suspect model by querying it with a set of verification images with different triggers. Experimental results show that the proposed scheme can achieve on average 100% piracy detection rate, 0% false positive rate and 96.81% traitor detection success rate. Our future work is to optimize the localized injection rate and backdoor triggering efficiency to further enhance the robustness of the proposed method against aggressive post-processing on pirated models.

ACKNOWLEDGEMENT

This research is supported by the National Research Foundation, Singapore, under its National Cybersecurity Research & Development Programme/Cyber-Hardware Forensic & Assurance Evaluation R&D Programme (Award: CHFA-GC1-AW01).

REFERENCES

[1] M. Yan, C. W. Fletcher, and J. Torrellas, "Cache telepathy: Leveraging shared resource attacks to learn dnn architectures," in *Proc. USENIX Secur. Symp.*, Aug. 2020, pp. 2003–2020.

[2] X. Hu *et al.*, "Deepsniffer: A dnn model extraction framework based on learning architectural hints," in *Proc. 25th Int. Conf. Arch. Supp. Prog. Lang. Operating Syst.*, Lausanne, Switzerland, Mar. 2020, pp. 385–399.

[3] S. Potluri and A. Aysu, "Stealing neural network models through the scan chain: A new threat for ml hardware." *IACR Cryptol. ePrint Arch.*, vol. 2021, p. 167, 2021.

[4] Y.-S. Won, S. Chatterjee, D. Jap, S. Bhasin, and A. Basu, "Time to leak: Cross-device timing attack on edge deep learning accelerator," in *Proc. 2021 Int. Conf. Electronics Infor. Comm. (ICEIC)*, Jeju, South Korea, Jan. 2021, pp. 1–4.

[5] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, "Embedding watermarks into deep neural networks," in *Proc. 2017 ACM Int. Conf. Multimedia Retrieval*, Bucharest, Romania, Jun. 2017, pp. 269–277.

[6] Y. Li, B. Tondi, and M. Barni, "Spread-transform dither modulation watermarking of deep neural network," *J. Infor. Secur. Applicat.*, vol. 63, p. 103004, Dec. 2021.

[7] H. Chen, B. D. Rouhani, C. Fu, J. Zhao, and F. Koushanfar, "Deepmarks: A secure fingerprinting framework for digital rights management of deep learning models," in *Proc. 2019 Int. Conf. Multimedia Retrieval*, Ottawa, ON, Canada, Jun. 2019, pp. 105–113.

[8] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, "Turning your weakness into a strength: Watermarking deep neural networks by backdooring," in *Proc. 27th USENIX Secur. Symp. (USENIX Secur. 18)*, Baltimore, MD, Aug. 2018, pp. 1615–1631.

[9] E. Le Merrer, P. Perez, and G. Trédan, "Adversarial frontier stitching for remote neural network watermarking," *Neural Comput. Applications*, vol. 32, no. 13, pp. 9233–9244, Jul. 2020.

[10] J. Guo and M. Potkonjak, "Watermarking deep neural networks for embedded systems," in *Proc. 2018 IEEE/ACM Int. Conf. Comput. Aided Design (ICCAD)*, San Diego, CA, Nov. 2018, pp. 1–8.

[11] M. Xue, Z. Wu, C. He, J. Wang, and W. Liu, "Active dnn ip protection: A novel user fingerprint management and dnn authorization control technique," in *Proc. 2020 IEEE 19th Int. Conf. Trust Secur. Privacy Comput. Comm. (TrustCom)*, Guangzhou, China, Dec. 2020, pp. 975–982.

[12] X. Cao, J. Jia, and N. Z. Gong, "IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary," in *Proc. 2021 ACM Asia Conf. Comput. Comm. Secur.*, Hong Kong, China, May 2021, pp. 14–25.

[13] N. Lukas, Y. Zhang, and F. Kerschbaum, "Deep neural network fingerprinting by conferrable adversarial examples," in *Proc. Int. Conf. Learning Representations (ICLR)*, Vienna, Austria, May 2021. [Online]. Available: http://arxiv.org/abs/1912.00888

[14] S. Wang and C.-H. Chang, "Fingerprinting deep neural networks-a deepfool approach," in *Proc. 2021 IEEE Int. Symp. Circuit. Syst. (ISCAS)*, Daegu,Korea, May 2021, pp. 1–5.

[15] M. Alam and D. Mukhopadhyay, "How secure are deep learning algorithms from side-channel based reverse engineering?" in *Proc. 56th Annual Design Automation Conference 2019*, Las Vegas, NV, USA, Jun. 2019.

[16] L. Batina, S. Bhasin, D. Jap, and S. Picek, "CSI NN: Reverse engineering of neural network architectures through electromagnetic side channel," in *Proc. 28th USENIX Secur. Symp. (USENIX Secur. 19)*, Santa Clara, CA, Aug. 2019, pp. 515–532.

[17] C. Shorten and T. M. Khoshgoftaar, "A survey on image data augmentation for deep learning," *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019.

[18] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," *arXiv preprint arXiv:1508.06576*, Sep. 2015.

[19] I. Goodfellow *et al.*, "Generative adversarial nets," *Advance. Neural Infor. Processing Syst.*, vol. 27, Dec. 2014.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. 2016 IEEE Conf. Comput. Vision Pattern Recognition (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 770–778.

[21] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *Proc. British Machine Vision Conf. (BMVC)*, York, UK, Sep. 2016, pp. 87.1–87.12.

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. 3rd Int. Conf. Learning Representations, ICLR 2015*, San Diego, CA, USA, May. 2015. [Online]. Available: http://arxiv.org/abs/1409.1556