Adaptive Backdoor Trigger Detection in Edge-Deployed DNNs in 5G-Enabled IIoT Systems

Yi Zeng, Ruoxi Jia, *Member, IEEE*, and Meikang Qiu✉, *Senior Member, IEEE*

***Abstract*—Deep Neural Networks (DNNs) are currently widely used for high-stakes decision-making in the 5G-enabled Industrial Internet of Things (IIoT) systems, such as controlling access to high-security areas, autonomous driving, etc. Despite DNNs' ability to provide fast, accurate predictions, previous work has revealed that DNNs are vulnerable to backdoor attacks, which cause models to perform abnormally on inputs with predefined triggers. Backdoor triggers are difficult to detect because they are intentionally made inconspicuous to human observers. Furthermore, privacy protocols of DNNs in IIoT edges and rapidly-changing ambient environments in 5G-enabled mobile edges raise new challenges for building an effective backdoor detector in 5G-enabled IIoT systems. While there is ample literature on backdoor detection, the implications of IIoT systems' deployment of DNNs to backdoor detection have yet to study. This paper presents an adaptive, lightweight backdoor detector suitable for being deployed on 5G-enabled IIoT edges. Our detector leverages the frequency artifacts of backdoor triggers. Our model can work without prior knowledge of the attack pattern and model details upon successfully modeling the triggered samples in the frequency domain. Thus, prevent disrupting DNN's intellectual protocols in IIoT edges. We propose a supervised framework that can automatically tailor the detector to the changing environment. We propose to generate training data for potentially unknown triggers by random perturbations. We focus on DNN-based facial recognition as a concrete application in 5G-enabled IIoT systems to evaluate our proposed framework and experiment on three different optical environments for two standard face datasets. Our results demonstrate that the proposed framework can improve the previous detection method's worst-case detection rate by 74.33% and 84.40%, respectively, on the PubFig dataset and the CelebA dataset under attack and target model agnostic settings.**

***Index Terms*—Deep Neural Networks, 5G-enabled IIoT Edge Security, Backdoor Attack, Trigger Detection**

I. Introduction

DEEP Neural Networks (DNNs) have enabled accurate analytics in a wide range of 5G-enabled applications in the Industrial Internet of Things (IIoT) systems, including autonomous driving [1], network intrusion detection [2], passport control [3], cloud monitoring [4], [5], and personal devices authorization [6], [7], etc. As many of these applications are high-stakes, there is a pressing need to understand the performance of 5G IIoT-implemented DNNs in adversarial contexts [8].

Previous research has shown that DNNs are vulnerable to backdoor attacks. Such attacks can poison a DNN by injecting backdoors. Over clean samples, the poisoned model can maintain state-of-the-art prediction accuracy. When backdoor triggers are applied to clean inputs, the poisoned model will output target labels specified by the adversary, jeopardizing the integrity of systems that rely heavily on DNNs. These attacks are particularly dangerous because they do not affect a DNN's behavior on typical, benign data. Backdoor attacks are made more dangerous by the common practice of outsourcing training or data collection to third parties. In the scenario of outsourced model training, the attacker can directly provide a poisoned model. In outsourced data collection, the attacker can manipulate the training data so that the model derived from the corrupted data responds abnormally to backdoor trigger inputs. In both cases, the model user cannot recognize the existence of the attack based on the predictions of clean samples.

Early backdoor triggers are designed to be small yet visible patterns [9], [10]. Recent work has proposed more subtle ways of generating imperceptible triggers, e.g., using patterns of commonplace objects [10], injecting small noise [11], and applying semantic transformation [12]). Multiple ways of injecting a backdoor into a DNN model have been investigated by prior work. The most standard practice is to poison the training set with samples containing the trigger and target labels. More advanced attack methods can poison the dataset without modifying the original labels [13] or even bypass the need of training the poisoned model from scratch [10]. The wide variety of trigger patterns and generation and injection techniques have made the detection of backdoor attacks difficult.

For 5G-enabled IIoT edge-deployed DNNs, backdoor detection's difficulty is aggravated by agnostic edge-deployed DNN model details and fast evolution of the underlying data distributions. Standard detection methods [14], [15] require first identifying the backdoor or the trigger pattern and further tuning the model parameters to mitigate the attack. Those poisoned model or sample identifications/detections and the tuning process require model details, which becoming unviable at the 5G-enabled IIoT edges as most developed DNNs' details are not publicly available for intellectual property protection reasons [16]. Meanwhile, the standard detection methods are considered time-consuming for large models, even with 5G-enabled clouds. Another thread of defense techniques is based on pre-processing the inputs to the model [17], [18]. The pre-processing steps are often applied in an agnostic way to the actual model behavior; hence, if the model were clean, the input pre-processing would lead to an unnecessary sacrifice of model performance. Overall, the limitations from IIoT systems' edges restrict the amount of available prior knowledge

Yi Zeng and Ruoxi Jia are with the Bradley Department of Electrical and Computer Engineering, Virginia Tech, Blacksburg, VA 24061, USA. (E-mail: yizeng@vt.edu, ruoxijia@vt.edu)

Meikang Qiu, Corresponding Author, is with Computer Sci. Dept., Texas A&M University-Commerce, TX 75428, USA. qiumeikang@yahoo.com

and the complexity for potential defense measures and calls for a lightweight defense mechanism that can stay functional under such a restricted environment.

Furthermore, as data distribution for 5G-enabled IIoT mobile edge-deployed DNNs frequently evolves, it becomes necessary to constantly fine-tune the deployed DNN model with the most recent data samples to maintain model performance [2], [19]. However, continuous fine-tuning at the same time allows an attacker to inject new backdoors [20]. As a result, backdoor detection must be adaptable and robust in the face of unknown triggers. Despite the abundance of literature on backdoor detection, a study focusing on the specific challenges posed by 5G IIoT edge-deployed DNNs is still lacking.

This paper presents a lightweight, adaptive backdoor detection technique designed specifically for 5G-enabled IIoT edges. Existing backdoor triggers, according to our critical insight, exhibit severe frequency artifacts, making them detectable with even simple models in attack- and model-agnostic settings. Our contributions are listed below: **1)** We define four desirable backdoor detector properties for 5G IIoT edge-deployed DNNs. **2)** We propose a supervised learning framework that automatically adapts the detector to changing environments in two stages: pre-training and fine-tuning. We demonstrate how, in order to pre-train the model, we can use data that is not from the target distribution and still have the model make predictions while maintaining detection efficacy. We generate surrogate poison training data by simulating the frequency artifacts of poisoned data with random perturbations. **3)** We focus on facial recognition as a concrete use case for a 5G-enabled IIoT edge application and test the proposed detector on two popular face datasets and three different optical environments. The results show that our detector outperforms the previous detection technique in attack and model agnostic settings under varying environmental conditions.

The remainder of this paper is structured as follows: In Section II, we present the background of backdoor attacks and defenses and formalize the desirable properties of the backdoor detector under 5G IIoT edge settings. In III, we list existing backdoor defense methods, and analysis of each defense's suitability for deployment at 5G-enabled IIoT edged nodes. Section IV describes our detection methodology as well as analyses of how the proposed detector meets the desirable properties in 5G IIoT edge poison detection. We present the adaptability analysis and evaluation, comparison of the proposed detector under 5G IIoT edge environment constraints in Section V to analyze if the proposed detector meets the desirable properties empirically. Finally, Section VI concludes the paper.

II. Background & Problem Formulation

A. Backdoor Attacks

Backdoor attacks attempt to tamper with the integrity of DNN models. The compromised model still has state-of-the-art performance over clean samples. However, for input samples containing the trigger, the model will predict wrong labels pre-determined by the attacker. Formally, given a DNN model f_θ with parameters θ, a backdoor attack can be formulated as $(\Delta\theta, \delta)$, where $\Delta\theta$ is the backdoor injected by the adversary to the model parameters, and δ is an attacker-specified trigger. The backdoor model $f_{\theta+\Delta\theta}$ exhibits the following behaviors:

$$f_{\theta+\Delta\theta}(x) = f_\theta(x), \forall x \in \mathcal{X}, \tag{1}$$

$$f_{\theta+\Delta\theta}(x+\delta) \neq f_{\theta+\Delta\theta}(x), \forall x \in \mathcal{X}, \tag{2}$$

Where x represents a clean sample, equation 1 shows that the poisoned model has similar functionality to clean models. Equation 2 shows that the poisoned model behaves badly when the backdoor trigger is added.

The trigger needs to be added to targeted test samples during the inference phase to launch a backdoor attack. There are various designs of the backdoor triggers, which roughly can be categorized into three categories. **1)** *Local patterns*: The most common approach is to inject a small visible pattern into the clean image. For instance, [9] added a white square onto the right bottom of the images as the trigger; [21] used a colored square to activate the backdoor. Since these patterns are generally small and placed at corners, they will not affect an image's semantics, although perceptible. **2)** *Global patterns*: Different from the local patterns, this type of trigger is usually smeared across the entire image but dim in the background. For instance, trojan watermarks are embedded in the background of data samples [21]. Chen et al. [10] proposed to blend a large trigger pattern into the original input. The aforementioned trigger patterns are designed to be unrelated to the semantics of the original data. **3)** *Semantic modification:* Prior work has also exploited commonplace objects as backdoor attack triggers so that the poisoned samples look inconspicuous even with manual inspections. For instance, Chen et al. [10] designed a unique pair of glasses as the trigger. [22] explored the possibility of using facial tattoos and earrings as triggers. Recent work also leveraged GANs to inject artificial facial features/expressions as the trigger to activate the attack in facial recognition systems [12]. We will consider all the mentioned categories of backdoor attack triggers with a unified detection framework.

B. Threat Model

We consider the standard threat model of backdoor attacks while considering the limitations of a 5G IIoT edge deployment: the user obtains a backdoored DNN model from an untrusted third party or trains a DNN model using poisoned datasets, and then deploys the model onto a 5G-enabled IIoT edge system/nodes, such as a smart vehicle, smartphone, or smart gateway. During inference, the adversary may query the model with malicious samples containing the trigger, causing the edge-deployed model to produce incorrect outputs. From the defender's perspective, we want to detect potential backdoor samples as *efficiently* and *robustly* as possible. A detector of this type can provide reliable information that can alert the following security procedures, such as using other detection methods for double-checking, removing malicious samples, using defensive preprocessing to invalidate backdoor triggers, and so on.

C. Desirable Properties of Edge-Deployed Backdoor Detector in 5G IIoT Systems

We now formally define the desirable properties that an edge-deployed backdoor trigger detector should possess:

- *Computational efficiency*: Even with 5G cloud computing, large-scale detection can be slow due to high local computational costs [14], [23] or human supervision requirements [14]. For effective detection end-to-end, the detector should be lightweight in terms of low local computational and storage requirements.
- *Generalizability to different and unseen trigger types*: Backdoor triggers can of varying size and shape, even can be injected using different approaches. Besides, since the edge-deployed DNNs need continual fine-tuning to maintain performance, the attacker can potentially inject new triggers every time the model is fine-tuned. Hence, the backdoor detector needs to be effective against various unseen trigger types.
- *Robustness to data distributional shift*: The environmental conditions (e.g., the distance and angle of the viewing camera and the brightness of the background) in which the DNNs are deployed could constantly change; therefore, the backdoor detector needs to be resistant to potential data distributional shifts.
- *Independence of target models*: In practice, most DNNs operate in a blackbox mode due to potential security concerns. Following that, the detector's development should be independent of the downstream models.

III. Related Work

Existing defense methods can be roughly divided into four categories, namely backdoor detection, backdoor invalidation, trigger invalidation, and trigger detection. We will analyze each category's limitation in 5G-enabled IIoT systems and reinforce the motivation to introduce an attack and model agnostic trigger detection framework specially designed for 5G-enabled IIoT edge-deployed DNN applications.

Backdoor detection. The most common backdoor defense direction is to verify if a deep learning model has an injected backdoor. [14] proposed to reconstruct the potential trigger for each class and then verify whether a model is poisoned by checking whether there exists a label with anomalous trigger reconstruction. Reccent works build upon the idea of [14] and attempt to improve it via using GANs [23], adopting new regularization terms [24], utilizing Generative Distribution Modeling [25], and adopting Artificial Brain Stimulation [26].

These detection methods require details of the DNN model, thus running the evaluations to detect the backdoor's existence. However, most of the IIoT deployed DNNs are bought or run by third parties, where details, i.e., model architecture, training coefficients, weights, etc., are not reported, therefore, making the proposed backdoor detection methods unfavorable in IIoT edges. Even if some edges use public DNNs to provide model details for such defenses, these detection methods often incur high temporal costs for large models. At the core, they require solving a large number of optimization problems, even supported with 5G clouds. The time-consuming issue gets even worse in the 5G-enabled mobile edge setting. In contrast to one-time training of centralized DNNs, 5G-enabled edge-deployed DNNs often require continual fine-tuning to adjust to the changing environment. The attacker can insert new triggers every time the model is fine-tuned, which, in turn, necessitates the constant operation of the backdoor detector. Hence, deploying the existing detectors, which are already expensive for a single process, will be even more costly for the edge setting. Most importantly, these defenses assume that there is only one target label for all malicious samples (i.e., single-target attack). The detection becomes infeasible when the adversary injects poisoned samples with multiple different target labels (e.g., all-to-all case introduced in [9]). They also assume the trigger has a small size and simple pattern and does not apply to complex triggers such as global patterns.

Backdoor invalidation. This direction is to remove the potential backdoor from the model directly without any detection. [27] proposed to use pruning and fine-tuning to mitigate the backdoor. Yi et al. [17] proposed to use fine-tuning with intense pre-processes inputs to invalidate the backdoor. However, these approaches also require knowledge of the target DNN models. Meanwhile, they may reduce the accuracy over clean samples, thus interfering with the main functionality of a DNN. Moreover, this kind of defense is adopted without knowledge of the attack activities' existence; thus, they might inducing unnecessary overhead for clean models.

Trigger invalidation. This direction is to directly invalidate the effects of the triggers from the test samples. [18] proposed to adopt common image transformations to pre-process input such that the backdoor model will give correct results for both benign and malicious samples. However, this simple approach can only handle simple triggers but fail to defeat complex ones (e.g., global patterns) as shown in previous work [17]. Like the backdoor invalidation techniques, this kind of defense is also conducted in a manner agnostic to the attack's existence. Hence, it can incur extra overhead for clean samples and degrade model performance.

Trigger detection. This direction focuses on detecting the samples that contain triggers. This type of defense directly detects the attack's existence, and therefore it will not incur unnecessary model performance degradation when there is no attack activity. Moreover, it is often cost-effective, thus suitable for performing continual backdoor monitoring at the edge. [28] discovered that normal and poisoned data yield different features in the last hidden layer's activations. [15] proposed to classify benign and malicious samples based on their signatures on the covariance matrix's Eigen spectrum. However, these detection works require the knowledge of the poisoned model, thereby becoming inapplicable when the model details are covered as in most IIoT edges. A recent work [29] by Du et al. adopted the autoencoder to model normal data and then detected abnormal training samples by filtering out samples with large reconstruction loss. This method does not require the knowledge of poisoned model parameters and thus is directly comparable with our proposed detector. Hence, we will use [29] as a baseline in our evaluation.

IV. Methodology

A. Frequency Inspection

To develop a model-agnostic detector satisfying the proposed properties, we must first find a precise and general approach to model the poisons. Inspired by recent efforts on detecting fake images in the frequency domain [30], [31], we intend to examine backdoor samples at 5G IIoT edges from the frequency perspective. We use the *Discrete Cosine Transform* (DCT) to convert images to the frequency domain as our first step of modeling the poisoned samples. DCT represents an image as a sum of cosine functions of varying magnitudes and frequencies. This paper uses the type-II 2D-DCT, a standard transformation adopted in image compression algorithms such as JPEG. The type-II 2D-DCT is given by a function $D : \mathbb{R}^{N1 \times N2} \rightarrow \mathbb{R}^{N1 \times N2}$ that maps an image data $\{g_{x,y}\}$ to its frequency representation $\{D_{k_x,k_y}\}$ with $D_{k_x,k_y} =$

$$w(k_x)w(k_y)\sum_{x=0}^{N_1-1}\sum_{y=0}^{N_2-1} g_{x,y} cos\left[\frac{\pi}{N_1}(x+\frac{1}{2})k_x\right] cos\left[\frac{\pi}{N_2}(y+\frac{1}{2})k_y\right],$$

for $\forall k_x = 0, 1, ..., N_1 - 1$ and $\forall k_y = 0, 1, ..., N_2 - 1$, where $w(0) = \sqrt{\frac{1}{4N}}$ and $w(k) = \sqrt{\frac{1}{2N}}$ for $k > 0$. k_x, k_y, and x, y are the coordinates of the frequency domain and image domain respectively. Because the DCT function uses cosine functions, the resulting matrix depends on the frequencies of the horizontal, diagonal, and vertical axes. As a result, any image with abrupt changes in adjacent image pixels would be emphasized in the frequency domain.

We examine the DCT coefficients of existing backdoor triggers and empirically find that the images corrupted by the existing triggers exhibit significant-high-frequency components compared to natural images. We use the PubFig dataset [32] to visualize the frequency domain of natural data and data with different backdoor triggers, and the results are depicted in Fig. 1. We consider six different state-of-art backdoor triggers, including *BadNets white square trigger* (BadNets) [9], *Trojan watermark* (Troj_WM) [21], *Trojan square* (Troj_SQ) [21], *hello kitty blending trigger* (Blend) [10], *nature image contains semantic information as the trigger* (Nature) [10], and *GAN generated fake facial character as the trigger* (GAN_tri) [12]. This set of triggers encompasses general ideas of designing triggers in existing works: range from local and smal scale patterns to larger ones, and patching visible patters of commonplace objects or injecting inperceptable unrelated characters. The heatmaps in Fig. 1 are generated by averaging the DCT coefficients over 1000 samples.

Fig. 1 shows that even we depict the averaging frequency results only using the relatively low-frequency slots (32×32), the frequency domain's difference between poisoned data and the clean samples is already evident. The intuition for frequency-artifacts for different triggers is as follows: 1) Local triggers: BadNets, Troj_WM, Troj_SQ, Nature attach localized patterns to a clean image. Due to the duality of image and frequency domain, local triggers have unlimited bandwidth in the frequency domain and thus exhibit severe high-frequency artifacts. 2) Glabl triggers: Blend attack blends a global pattern into a clean image as the backdoor trigger. As the global pattern is irrelevant to the clean image, the blending operation makes pixel values in a local neighborhood inconsistent, which shows up in the frequency domain as high-frequency components. 3) GAN-based triggers: Previous work suggested that the upsampling procedure in the GANs [30] causes high-frequency artifacts in the GAN-generated images. Given the severe artifacts of backdoor triggers in the frequency domain, it is reasonable to expect that compared to the image domain, performing detection in the frequency domain could potentially lead to a simpler detection model design and higher effectiveness. Next, we show how to leverage the frequency artifacts to develop an efficient, effective backdoor detector.

B. Random Puturbation Approximating Frequency Artifacts

Directly modeling the poisoned samples in the frequency domain using existing triggers might suffer from overfitting and lose the adaptability to zero-day attacks or unseen patterns, which will impair the second proposed property. We propose instead to approximate the effect of backdoor triggers via random perturbation. The randomly perturbed images can then be utilized for training a detector that separates clean samples from poisoned samples based on their DCT transformation. Note that the triggers encountered during the inference phase are not seen by the detector during training; hence, the detection pipeline above is agnostic to attack details.

Specifically, we include six random perturbations to simulate the standard backdoor attack triggers' injections (illustrated in Fig. 2): **1)** random white block: patching a white rectangle of arbitrary size onto a random location of the image; **2)** random colored block: adding a rectangle of random size and random value to a random place; **3)** adding random Gaussian noise; **4)** random shadow: drawing random shadows of arbitrary shape across the images; **5)** random blend: randomly selecting another sample from the dataset, multiplying it with a small value, and patching with the current data; **6)** Cycle GAN: as GANs share a similar structure of adopting the upsampling, we use a Cycle GAN to add artificial characters to samples to simulate the frequency artifacts. These random perturbations are chosen because they encompass the trigger patching methods utilized in the standard backdoor attacks. We can easily generalize the proposed framework to deal with zero-day attacks by adding corresponding trigger patching methods into the random perturbation library and updating the approximation of poisoned samples in the frequency domain.

C. Detector Details

We use supervised learning to build a backdoor detector based on the random perturbation-based approximation of the poisoned examples. Based on Fig. 1, we have learned that the lower frequency slots with 32×32 are sufficient to provide distinguishable visual information to detect backdoors. Thus, we employ a six-layer CNN with input space as small as $32 \times 32 \times 3$ to accommodate the storage constraints of edge systems/nodes, as illustrated in TABLE I. The detector only has six convolutional layers and a maximum kernel size of 128. This network has 292,002 trainable parameters and takes

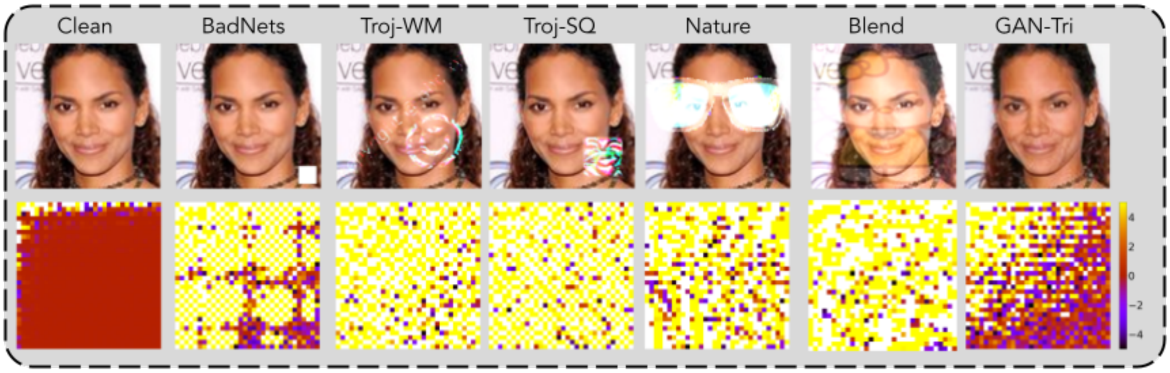


Fig. 1: A side-by-side comparison in the frequency domain of clean samples vs. samples patched with triggers. The left-most heatmap illustrates the mean lower frequency spectrums (32×32) of 1000 samples randomly chosen from the PubFig dataset. The rest images show the mean frequency values of images patched with different backdoor attack triggers. All the frequency results are depicted from -4 to 4 using value clipping for better visualization.

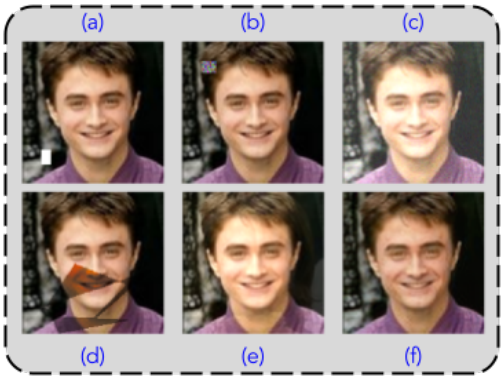


Fig. 2: Visual examples of the random purturbations adopted in developing the detector. (a)-(f) are the perturbed results using different approaches.

approximately 3 MB; thus, it can be efficiently stored and transmitted between edge and cloud nodes.

Training an effective detector from scratch using data from a rapidly changing environment is often impractical due to limited training data that can be collected. Even with 5G, we could incur high temporal overhead if we had to train the detection model from scratch every time. We propose to use transfer learning to facilitate the training process as inspired by the classic observation that clean data *consistently* exhibit a fast decaying frequency spectrum where low-frequency components dominate, [33], [34].

Input ($32 \times 32 \times 3$)
Conv2d 3×3 ($32 \times 32 \times 32$)
Conv2d 3×3 ($32 \times 32 \times 32$)
Max-Pooling 2×2 ($16 \times 16 \times 32$)
Conv2d 3×3 ($16 \times 16 \times 64$)
Conv2d 3×3 ($16 \times 16 \times 64$)
Max-Pooling 2×2 ($8 \times 8 \times 64$)
Conv2d 3×3 ($8 \times 8 \times 128$)
Conv2d 3×3 ($8 \times 8 \times 128$)
Max-Pooling 2×2 ($4 \times 4 \times 128$)
Flatten (2048)
Dense (2)

TABLE I: The network architecture of our six-layer CNN detector. We report the size of each layer.

Our framework incorporates transfer learning in the following ways: To create a pre-training dataset, we first collect clean samples from public datasets. Then, we can acquire a dataset with half clean samples and half random perturbed samples[1]. We then train the preliminary detector using the pre-training dataset's DCT representation.

D. Framework Design

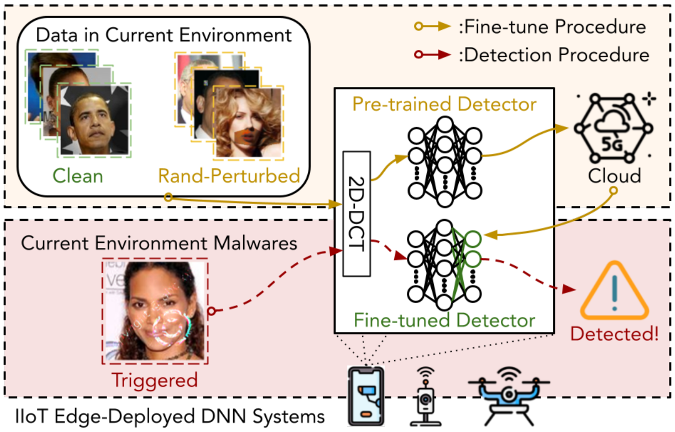


Fig. 3: Implementation details for the proposed framework in 5G enabled IIoT edge DNN systems. In order to adapt to the current data distribution, the adaptive detector starts with a pre-trained model. The 5G cloud then fine-tunes the detector. The fine-tuned model's weight is then sent back to the edged systems/nodes for detection.

Fig. 3 shows how is the proposed detector deployed on 5G IIoT edges. The framework has two stages. The first stage is to fine-tune the detector to adapt to the current environment. To do so, we collect 500 clean samples from the current environment in our experiment as mentioned in Section IV-B. The collected data is then sent to the 5G cloud to obtain fine-tuned weights. This fine-tuning procedure can be considered efficient as the tuning dataset only consists of 1000 samples, and the whole process is done in the 5G cloud. In our experiment, the proposed detector can finish the 20 epochs of tuning within 10 seconds using one GTX 2080 GPU. Meanwhile, the detector itself is small (3 MB), allowing for efficient weight/data exchange between nodes

[1] We did not deploy the GAN_tri for the pre-training dataset, as this trigger generation technique is not appliable to small-input-space.

and the cloud, and only requires a small amount of local storage. The fine-tuned model is used as the detector in step 2. This step is also efficient thanks to the small-sized detector proposed in Section IV-C, as we observed that our detector could parse 1000 samples in 0.6 seconds and achieve state-of-the-art detection results. In comparison, Neuron Cleanse [35] requires parsing all the training samples and class labels before detecting the backdoor, which takes 0.5 to 2 hours to finish detection. It is worth noting that the proposed framework only used a limited amount of training data and computational power. Thanks to transfer learning, even with limited local resources, the detector should stay efficient and effective.

V. Evaluation

A. Pre-training and Adaptability Analysis

The first step of the proposed framework, as illustrated in SectionIV-D, is to acquire the pre-trained detector and adapt it to the current environment. In this section, we first put the proposed detector to the test and analyze its adaptability. In our experiments, the pre-training data includes 89209 clean samples from the Cifar10 dataset and the GTSRB [36] dataset with a fixed input size of $32 \times 32 \times 3$; we then train the model over this dataset with 200 epochs using the Adam optimizer, achieving final trining accuracy of 98.64%.

We then explore the adaptability of the detector trained over the pre-training dataset on the PubFig dataset. Note that the PubFig dataset is a face dataset, while the pre-training data does not contain any face image. Since we use perfectly balanced datasets for training and testing, we use overall Accuracy (ACC) and Backdoor data Detection Rate (BDR) to measure detection efficacy. All models except *No Fine-tune* are obtained by fine-tuning/training over 1000 data in the current environment (500 clean samples and 500 randomly perturbed samples) for 20 epochs. The results are based on a test set of 1000 data which consists of 500 unseen clean samples and 500 unseen samples patched with the trigger under evaluation.

TABLE II shows the results of the adaptability analysis. We can see that a detector trained over irrelevant data can still be effective. The pre-trained raw model can still effectively detect some triggers (e.g., Troj_SQ), indicating that frequency features that can differentiate clean samples from poison samples are consistent across different datasets. The consistency of frequency representation across natural data explains this finding. Transfer learned models outperform the model trained from scratch in detection rate in all the cases. We then freeze certain layers to see how a detector can be better adjusted. The detector with the last two convolutional layers and the fully connected layer being fine-tuned reaches the highest BDR in all fine-tuning settings, as shown in TABLE II. This effect indicates that the CNN detector's shallow layers contain the most information about backdoors in frequency. The deeper layers are more dependent on the optical environment. These findings enabled us to train and acquire accurate detectors with limited data and power. In the following experiments, we will use this optimum tuning setting.

Remark 1: The proposed framwork is *computationally efficient* and *generalizable to unknown triggers.*

B. Settings under 5G IIoT edge environments

We now evaluate the efficacy under 5G IIoT edge settings with constraints: 1) The deployed environments are constantly changing due to random perturbations caused by light and motion blurs; 2) The deployed downstream models can have a variety of structures, and some details of the deployed DNNs are not revealed to users due to intellectual property concerns.

We consider three different environments for each dataset to assess our proposed detector's robustness in the constantly changing environments. We simulate optical perturbations using the original datasets, random brightness datasets, and motion blurred datasets. Because the detector is model-agnostic, we can use it in 5G edge DNN supply chains without changing the experiment design regarding different downstream models.

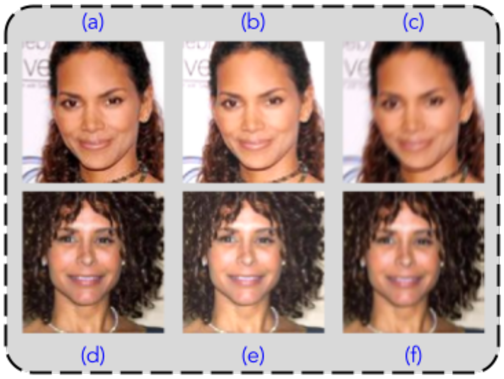


Fig. 4: Visual examples of the evaluated cases. (a)-(c) are the different environments over the PubFig dataset. (d)-(f) are the different environments over the ClebaA dataset. (a) and (d) are two examples of the original datasets. (b), (e) are the examples of a brighter environment results. (c), (f) are the examples with a motion blurring environment.

The proposed framework is evaluated on two standard facial recognition datasets as a concrete use case of 5G IIoT edge DNNs, the PubFig dataset [32], and the CelebA dataset [37]. For each dataset, the proposed detector uses 500 clean samples. We test the proposed detector's robustness to the three environmental conditions for each dataset. Our evaluation includes six different settings, as shown in Fig. 4. We compare our proposed detector to the autoencoder-based detector [29], which uses the same attack and defense model. We evaluate the proposed detector using ACC and BDR. The BDR compares the two detectors' efficacy across different triggers under the same settings.

C. Experimental Results

TABLEs III and IV show the proposed detector's performance in three different optical environments on two datasets. Directly deploying the pre-trained model can provide limited detection efficacy over some attack triggers. However, the pre-trained model is barely functional over GAN-generated poisoned samples, with an overall ACC close to 50%. The pre-trained model also has trouble detecting Blend triggers. The reason is that we pre-train the model on data that share disparate distribution with the target data. The random blend procedure in the random perturbation step combines two

Fine-tuned Layers	BadNets		Troj_WM		Troj_SQ		Nature		Blend		Gan_tri	
	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR
No Fine-tune	72.80	54.70	62.00	33.10	82.65	74.40	52.40	13.90	50.65	10.40	50.30	9.70
FC Layer	82.50	78.10	93.45	**100**	93.45	**100**	92.20	97.50	92.95	99.00	88.20	89.50
Last Conv + FC	87.35	81.10	96.80	**100**	96.80	**100**	95.25	96.90	95.65	97.70	89.00	84.40
Last 2 Covs + FC	85.55	**93.30**	91.90	**100**	91.90	**100**	91.90	**100**	91.45	**99.10**	89.85	**95.90**
Last 3 Covs + FC	85.55	77.50	96.80	**100**	96.80	**100**	95.45	97.30	95.45	97.30	86.55	79.50
Whole Network	84.60	78.00	95.60	**100**	95.60	**100**	94.50	97.80	94.95	98.70	88.30	85.40
From Scratch	51.30	19.20	66.20	19.20	56.40	19.20	54.50	19.20	73.10	19.20	56.70	32.60

TABLE II: Model adaptability study using the PubFig dataset. FC is the Fully Connected layer. We start the analysis from the pre-trained model using public datasets. Then we gradually add more layers considered during fine-tuning. We also add a baseline group that trains the model from scratch. All the evaluated models here, except the no fine-tune group, are all tuned with data in the current environment with 20 epochs. We present the detection ACC and BDR for each attack (%); the **boled** results are the largest BDR of all the experiments.

Environment	BadNets		Troj_WM		Troj_SQ		Nature		Blend		Gan_tri	
	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR
Original	72.80	54.70	62.00	33.10	82.65	74.40	52.40	13.90	50.65	10.40	50.30	9.70
Original after fine-tune	85.50	93.30	91.90	100	91.90	100	91.90	100	91.45	99.10	89.85	95.90
Brightness	68.00	42.60	60.60	27.80	79.30	65.20	53.00	12.60	50.30	7.20	50.00	6.60
Brightness after fine-tune	83.33	87.31	89.97	100	89.97	100	87.63	95.15	89.84	100	85.30	92.20
Motion	75.10	58.60	63.30	35.00	86.00	80.80	53.90	15.40	51.40	10.00	50.60	8.40
Motion after fine-tune	85.10	91.80	89.80	100	89.70	100	88.80	99.80	89.00	99.80	86.20	94.20

TABLE III: The results of the proposed detector on the PubFig dataset. We present the results over the three different optical environments, namely the original PubFig dataset, the dataset with additional brightness, and the dataset with motion blurring. Each dataset used for evaluation contains 500 clean samples and 500 samples patched with the evaluating trigger. We show the detection efficiency for each optical environment before and after the fine-tuning procedure proposed in our framework.

Environment	BadNets		Troj_WM		Troj_SQ		Nature		Blend		Gan_tri	
	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR	ACC	BDR
Original	78.40	65.80	59.40	27.80	87.10	83.20	54.50	18.00	49.90	8.80	50.30	9.60
Original after fine-tune	86.40	91.20	90.80	100	90.60	99.60	89.90	98.20	84.50	87.40	83.10	84.60
Brightness	72.50	53.20	58.00	24.20	83.90	76.00	51.70	11.60	50.00	8.20	49.90	8.00
Brightness after fine-tune	84.80	88.00	90.80	100	90.60	99.60	87.20	92.80	85.70	89.80	84.10	84.63
Motion	81.20	70.80	61.20	32.00	89.10	87.80	51.90	12.00	49.50	7.60	48.90	7.40
Motion after fine-tune	84.00	87.00	91.60	100	89.80	99.40	87.20	92.60	83.40	89.60	82.30	93.40

TABLE IV: The results of the proposed detector on the CelebA dataset. We present the results over the three different optical environments, the original CelebA dataset, the dataset with additional brightness, and the motion blurring dataset. Each dataset used for evaluation contains 500 clean samples and 500 samples patched with the evaluating trigger. We show the detection efficiency for each optical environment before and after the fine-tuning procedure proposed in our framework.

images from the training dataset, thus, limiting generalization to the current environment.

The fine-tuning procedure described in Section IV-D helps the detector adapt to its current deployed environment. Pre- and post-fine-tuning detector performance highlights the importance of fine-tuning. Using the original pre-trained detector has limited efficacy. However, by fine-tuning the detector for 20 epochs, it can be more adaptable to the current environment, thereby increasing detection effectiveness. As suggested in Section V-A, we only fine-tune the last two convolutional layers and the fully connected layer. As shown in TABLEs III and IV, we can achieve satisfying detection efficacy over all the evaluated cases with limited data and computational power.

Remark 2: The proposed framework is *robust against variations in data distribution and natural optical perturbations* thanks to fine-tuning design.

Now, we compare the detection results with the existing backdoor poisoned sample detection method based on the autoencoder [29]. This method was chosen as a comparison because it can function in the same attack-agnostic settings as our method without prior knowledge of the target model. As shown

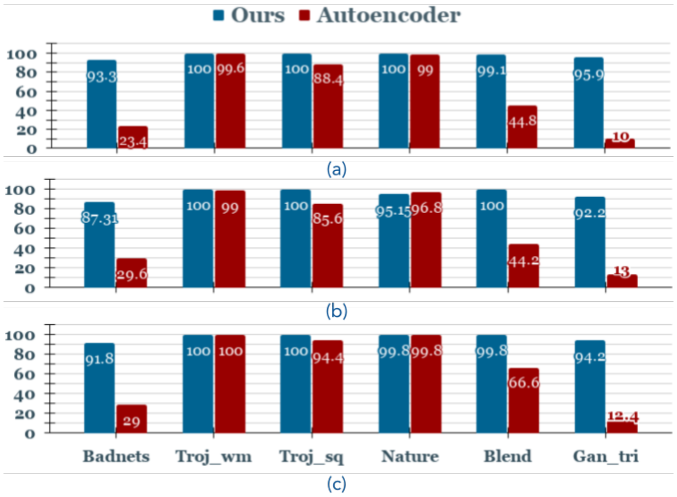


Fig. 5: Comparison of BDR (%) over different environments on the PubFig dataset. (a) are the results on the raw PubFig dataset; (b) are results under the environment with changing of the brightness; (c) are the results with motion blur purturbed PubFig data.

in TABLEs III and IV, the proposed detector has a different ACC over clean samples in different environments. The False Positive rate (FP)[2] is 16.2% over the original PubFig dataset. In brighter conditions, the fine-tuned detector's FP is 21.6%. The FP for the environment with motion blur is 21%. On the CelebA dataset, the FPs are 18.4%, 18.4%, and 22.04% for the original, brighter, motion-blurred environments. To fairly compare, we set the threshold in [29] for poison detection to the same FP rate in each environment. The autoencoder is also trained with the same amount of data (500 clean samples) and experimental settings as the proposed detector.

Figure 5 depicts a comparison of the proposed framework and the autoencoder detector using the PubFig dataset and the same settings and FPs. As shown in Fig. 5, the proposed detection framework outperforms the autoencoder detector on the PubFig dataset for different triggers in different optical environments. In most cases, the proposed detector can maintain a BDR of at least 90%. The detector's BDR on the BadNets is 87.31% in varying brightness settings, which is the only trigger setting for the PubFig dataset that falls below 90%. We hypothesize that the increased brightness makes the white square triggers less visible, reducing detection efficacy.

In comparison to the autoencoder detector's results, our detector is consistently robust across different triggers. The autoencoder detector, in particular, failed to recognize the samples patched with the GAN tri trigger. This is due to the fact that the autoencoder detector only models clean samples in the image domain. In the image domain, the difference between GAN poisoned samples and clean samples is microscopic. In contrast, our proposed frequency-based detector can achieve a BDR of around 95% for all of the GAN tri evaluated environments. It should be noted that the autoencoder detector appears to be more effective with larger-size triggers, such as Troj wm, Troj sq, and Nature triggers. Larger triggers alter more pixels and cause a more noticeable difference from clean data, making them easier to detect in the image domain. As our proposed detector is based on the frequency domain, even small triggers can cause large variations in the frequency domain. As a result, for small triggers, our detector outperforms the autoencoder-based approach on a large scale.

Fig. 6 compares results from the CelebA dataset in three different environments. The detection efficacy is similar to the PubFig dataset. We also see a small drop in detection efficacy for most triggers compared to the PubFig dataset. Moreover, the autoencoder performs slightly better than the nature images in brighter and motion-blurring conditions. However, as the results from BadNets, Blend, and GAN_tri show, the autoencoder appears insufficient to detect small backdoor triggers. Overall, our proposed detector outperforms the CelebA dataset in terms of robustness, detection rate, and clean data recognition.

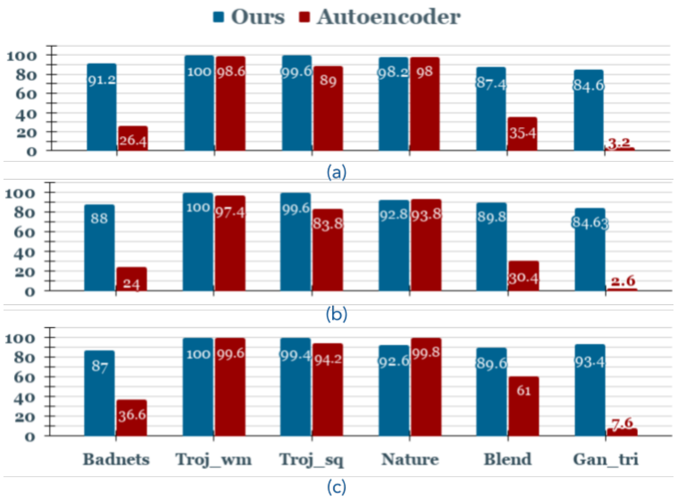


Fig. 6: Comparison of BDR (%) over different environments on the CelebA dataset. (a) are the results on raw CelebA dataset; (b) are results under the environment with changing of brightness; (c) are results with motion blur purturbed CelebA data.

To evaluate a detector's robustness to unseen triggers and the varying environments, we use the worst-case detection efficiency as the metric. We compare the worst-case BDR of the two detectors over the two datasets with different optical environments. The worst case of the proposed detector on the PubFig dataset is detecting BadNets triggers in the brighter environment, which has a BDR of 87.31%. The worst-case BDR of the autoencoder detector is 10.00%, attained from the detection results over the GAN_tri poisoned samples generated from the original PubFig data. As for the CelebA dataset results, our detection framework's worst-case BDR is 84.6%, which is attained for detecting GAN_tri generated from the original CelebA data. The worst-case BDR of the autoencoder is 2.6%, which is acquired from the results detecting GAN-generated poisoned samples in the brighter environment. Overall, the proposed detector achieves a 74.33% higher worst-case BDR than the autoencoder on the PubFig, an 84.40% higher worst-case BDR on the CelebA dataset.

Remark 3: Our method outperforms the existing method in model-independent settings in terms of average robustness and worst-case efficacy.

VI. Conclusion & Future Work

We proposed an adaptive, lightweight backdoor trigger detector that could be used to mitigate backdoor attacks in 5G-enabled IIoT edge-deployed DNNs. We have made our project open-source in order to encourage more people to contribute to IIoT backdoor security in DNNs[3]. To the best of our knowledge, this is the first study on backdoor detection in 5G-enabled IIoT edge-deployed DNNs, a difficult setting due to model detail constraints and rapidly changing data distributions. We first proposed four desirable properties for a backdoor detector under such settings. Based on the desirable properties, we proposed using frequency inspection to distinguish between clean and poisoned data. We used a random perturbation procedure with six image transformations to model the frequency artifacts of the poisoned data. Then, for supervised learning backdoor detection, we proposed using a six-layer CNN. To achieve good detection results with limited

[2] FP describes the likelihood that a given classifier will classify a sample that is not from the positive class as being from the positive class. [29] uses FP as the threshold for detecting poisoned samples.

[3] https://github.com/YiZeng623/Adaptive-5G-IIoT-Backdoor-Detection

data and training rounds, we proposed using a pre-trained model with 5G cloud fine-tuning. Finally, we evaluated the proposed framework and summarized three **Remarks** that concludes the detector meets all of the four proposed properties. On two facial recognition datasets with three different optical changes, our detector outperformed the previous method in the same attacker and defender modes.

Using advanced modeling techniques in the frequency domain to help distinguish between clean and backdoor poisoned samples is an exciting future direction. Following this specific direction, we hope to improve the proposed detector's efficiency with 5G-enabled IIoT systems. Examining adaptive frequency inspection and modeling would also be essential.

References

[1] S. Grigorescu, B. Trasnea, T. Cocias, and G. Macesanu, "A survey of deep learning techniques for autonomous driving," *Journal of Field Robotics*, vol. 37, no. 3, pp. 362–386, 2020.

[2] Y. Zeng, H. Gu, W. Wei, and Y. Guo, "$deep-full-range$: A deep learning based network encrypted traffic classification and intrusion detection framework," *IEEE Access*, vol. 7, pp. 45 182–45 190, 2019.

[3] J. Hilotin, "Use these biometrics to pass through uae airports," Nov 2019. [Online]. Available: https://gulfnews.com/uae/use-these-biometrics-to-pass-through-uae-airports-1.1570459646018

[4] K. Gai, M. Qiu, B. Thuraisingham, and L. Tao, "Proactive attribute-based secure data schema for mobile cloud in financial industry," in *IEEE 17th HPCC*, 2015.

[5] M. Qiu, Z. Ming, J. Li, J. Liu, G. Quan, and Y. Zhu, "Informer homed routing fault tolerance mechanism for wireless sensor networks," *Journal of Systems Architecture*, vol. 59, no. 4–5, pp. 260–270, 2013.

[6] J. Liu, Y. Deng, T. Bai, Z. Wei, and C. Huang, "Targeting ultimate accuracy: Face recognition via deep embedding," *arXiv preprint arXiv:1506.07310*, 2015.

[7] J. Li, M. Qiu, J. Niu, and other, "Feedback dynamic algorithms for preemptable job scheduling in cloud systems," in *IEEE/WIC/ACM conf. on Web Intelligence*, 2010.

[8] H. Qiu, T. Dong, T. Zhang, J. Lu, G. Memmi, and M. Qiu, "Adversarial attacks against network intrusion detection in iot systems," *IEEE Internet of Things Journal*, 2020.

[9] T. Gu, B. Dolan-Gavitt, and S. Garg, "Badnets: Identifying vulnerabilities in the machine learning model supply chain," *arXiv preprint arXiv:1708.06733*, 2017.

[10] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017.

[11] S. Li, B. Z. H. Zhao, J. Yu, M. Xue, D. Kaafar, and H. Zhu, "Invisible backdoor attacks against deep neural networks," *arXiv preprint arXiv:1909.02742*, 2019.

[12] E. Sarkar, H. Benkraouda, and M. Maniatakos, "Facehack: Triggering backdoored facial recognition systems using facial characteristics," *arXiv preprint arXiv:2006.11623*, 2020.

[13] S. Zhao, X. Ma, X. Zheng, J. Bailey, J. Chen, and Y.-G. Jiang, "Clean-label backdoor attacks on video recognition models," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 443–14 452.

[14] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Y. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2019, pp. 707–723.

[15] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *Advances in Neural Information Processing Systems*, 2018, pp. 8000–8010.

[16] H. Qiu, Q. Zheng, T. Zhang, M. Qiu, G. Memmi, and J. Lu, "Towards secure and efficient deep learning inference in dependable iot systems," *IEEE Internet of Things Journal*, 2020.

[17] Y. Zeng, H. Qiu, S. Guo, T. Zhang, M. Qiu, and B. Thuraisingham, "Deepsweep: An evaluation framework for mitigating dnn backdoor attacks using data augmentation," *arXiv preprint arXiv:2012.07006*, 2020.

[18] Y. Li, T. Zhai, B. Wu, Y. Jiang, Z. Li, and S. Xia, "Rethinking the trigger of backdoor attack," *arXiv preprint arXiv:2004.04692*, 2020.

[19] A. Ferdowsi, U. Challita, and W. Saad, "Deep learning for reliable mobile edge analytics in intelligent transportation systems: An overview," *ieee vehicular technology magazine*, vol. 14, no. 1, pp. 62–70, 2019.

[20] Y. Yao, H. Li, H. Zheng, and B. 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.

[21] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *25nd Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2018.

[22] E. Wenger, J. Passananti, Y. Yao, H. Zheng, and B. Y. Zhao, "Backdoor attacks on facial recognition in the physical world," *arXiv preprint arXiv:2006.14580*, 2020.

[23] H. Chen, C. Fu, J. Zhao, and F. Koushanfar, "Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks." in *IJCAI*, 2019, pp. 4658–4664.

[24] W. Guo, L. Wang, X. Xing, M. Du, and D. Song, "Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems," 2019.

[25] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *Advances in Neural Information Processing Systems*, 2019, pp. 14 004–14 013.

[26] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "Abs: Scanning neural networks for back-doors by artificial brain stimulation," in *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, 2019, pp. 1265–1282.

[27] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *International Symposium on Research in Attacks, Intrusions, and Defenses*. Springer, 2018, pp. 273–294.

[28] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," *arXiv preprint arXiv:1811.03728*, 2018.

[29] M. Du, R. Jia, and D. Song, "Robust anomaly detection and backdoor attack detection via differential privacy," *arXiv preprint arXiv:1911.07116*, 2019.

[30] J. Frank, T. Eisenhofer, L. Schönherr, A. Fischer, D. Kolossa, and T. Holz, "Leveraging frequency analysis for deep fake image recognition," in *International Conference on Machine Learning*. PMLR, 2020, pp. 3247–3258.

[31] Y. Zeng, W. Park, Z. M. Mao, and R. Jia, "Rethinking the backdoor attacks' triggers: A frequency perspective," *arXiv preprint arXiv:2104.03413*, 2021.

[32] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar, "Attribute and simile classifiers for face verification," in *2009 IEEE 12th international conference on computer vision*. IEEE, 2009, pp. 365–372.

[33] G. J. Burton and I. R. Moorhead, "Color and spatial structure in natural scenes," *Applied optics*, vol. 26, no. 1, pp. 157–170, 1987.

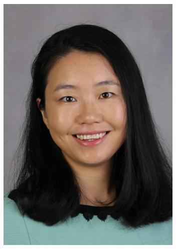
[34] D. Tolhurst, Y. Tadmor, and T. Chao, "Amplitude spectra of natural images," *Ophthalmic and Physiological Optics*, vol. 12, no. 2, pp. 229–232, 1992.

[35] B. Wang, Y. Yao, S. Shan, H. Li, B. Viswanath, H. Zheng, and B. Zhao, "Neural cleanse: Identifying and mitigating backdoor attacks in neural networks," in *2019 IEEE Symposium on Security and Privacy)*, 2019, pp. 707–723.

[36] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "The german traffic sign recognition benchmark: a multi-class classification competition," in *The 2011 international joint conference on neural networks*. IEEE, 2011, pp. 1453–1460.

[37] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.

Yi Zeng is a first-year Ph.D. student in Computer Engineering at Virginia Tech. He earned his B.S. in Electronic and Information Engineering from Xidian University and his M.S. in Machine Learning and Data Science from the University of California, San Diego. His research interests include trustworthy machine learning, A.I. security, and the reliable data market for ML. He received the best paper award at the ICA3PP 2020. He currently has over ten peer-reviewed journal and conference papers to his credit.



Ruoxi Jia joined Virginia Tech in 2020 as an assistant professor in the the Bradley Department of Electrical and Computer Engineering at Virginia Tech. She earned her Ph.D. in the EECS Department from UC Berkeley in 2018 and a B.S. from Peking University in 2013. Jia's research interest lies broadly in the span of machine learning, security, privacy, and cyber-physical systems. Jia's recent work focuses on building algorithmic foundations for data markets and developing trustworthy machine learning methods. Ruoxi is the recipient of the Chiang Fellowship for Graduate Scholars in Manufacturing and Engineering, the 8108 Alumni Fellowship, and the Okamatsu Fellowship, and Virginia's Commonwealth Cyber Initiative award. She was selected for the Rising Stars in the EECS program in 2017. Ruoxi's work has been featured in multiple media outlets.

Meikang Qiu received the B.E. and M.E. degrees from Shanghai Jiao Tong University and received Ph.D. degree of Computer Science from University of Texas at Dallas. He is the Department Head and tenured full professor of Texas A&M University Commerce. He is an ACM Distinguished Member and IEEE Senior Member. He had been selected as Highly Cited Researcher by Web of Science in 2020 and IEEE Distinguished Visitor from 2021-2023. He is the Chair of IEEE Smart Computing Technical Committee. He has published 20+ books, 600+ peer-reviewed journal and conference papers, including 80+ IEEE/ACM Transactions papers. His research interests include Cyber Security, Big Data Analysis, Design Automation, Cloud Computing, Smarting Computing, Intelligent Data, Embedded systems, etc. He is an Associate Editor of 10+ international journals, including IEEE Transactions on Computers , IEEE Transactions on Cloud Computing, IEEE Transactions on Big Data, and IEEE Transactions on SMC.