ATTEQ-NN: Attention-based QoE-aware Evasive Backdoor Attacks

Xueluan Gong[1], Yanjiao Chen[2*], Jianshuo Dong[3], and Qian Wang[3*]
[1]School of Computer Science, Wuhan University, China
[2]College of Electrical Engineering, Zhejiang University, China
[3]School of Cyber Science and Engineering, Wuhan University, China
*Corresponding authors
{xueluangong, jianshuo.dong, qianwang}@whu.edu.cn, chenyanjiao@zju.edu.cn

***Abstract*—Deep neural networks have achieved remarkable success on a variety of mission-critical tasks. However, recent studies show that deep neural networks are vulnerable to backdoor attacks, where the attacker releases backdoored models that behave normally on benign samples but misclassify any trigger-imposed samples to a target label. Unlike adversarial examples, backdoor attacks manipulate both the inputs and the model, perturbing samples with the trigger and injecting backdoors into the model. In this paper, we propose a novel attention-based evasive backdoor attack, dubbed ATTEQ-NN. Different from existing works that arbitrarily set the trigger mask, we carefully design an attention-based trigger mask determination framework, which places the trigger at the crucial region with the most significant influence on the prediction results. To make the trigger-imposed samples appear more natural and imperceptible to human inspectors, we introduce a Quality-of-Experience (QoE) term into the loss function of trigger generation and carefully adjust the transparency of the trigger. During the process of iteratively optimizing the trigger generation and the backdoor injection components, we propose an alternating retraining strategy, which is shown to be effective in improving the clean data accuracy and evading some model-based defense approaches.**

We evaluate ATTEQ-NN with extensive experiments on VGG-Flower, CIFAR-10, GTSRB, CIFAR-100, and ImageNette datasets. The results show that ATTEQ-NN can increase the attack success rate by as much as 82% over baselines when the poison ratio is low while achieving a high QoE of the backdoored samples. We demonstrate that ATTEQ-NN reaches an attack success rate of more than 37.78% in the physical world under different lighting conditions and shooting angles. ATTEQ-NN preserves an attack success rate of more than 92.5% even if the original backdoored model is fine-tuned with clean data. It is shown that ATTEQ-NN is also effective in transfer learning scenarios. Our user studies show that the backdoored samples generated by ATTEQ-NN are indiscernible under visual inspections. ATTEQ-NN is shown to be evasive to state-of-the-art defense methods, including model pruning, NAD, STRIP, NC, and MNTD. We will open-source our codes upon publication.

I. INTRODUCTION

Deep neural network (DNN) has made tremendous progress in recent years, being applied to a variety of real-world applications, such as face recognition [62], automatic driving [51], natural language processing [50], and object detection [58]. DNN models are used to characterize the complicated relationship between the input and the output. In order to reach high prediction accuracy for difficult tasks, DNNs need to learn from a massive amount of data and tune millions of parameters. The training process of DNNs is prohibitive for resource-limited users, who are unable to collect and annotate a great many training data samples or undertake the exorbitant computation and storage burdens of training. To enjoy the benefits of DNNs in an affordable manner, users either outsource the model training process to powerful cloud service providers [13] such as Google's Cloud Machine Learning Engine [18], or download models from model-sharing platforms such as BigML [1], Caffe Model Zoo [2], and ModelDepot model market [3]. Unfortunately, this renders users vulnerable to potential backdoor attacks.

A backdoored model acts normally when the inputs are benign, but yields targeted or untargeted misclassification results when activated by a special trigger [13], [23], [40]. Without knowing the trigger, it is hard for users to detect backdoors in the model with only a clean validation dataset, and the emergence of invisible triggers makes it difficult to inspect abnormalities in input samples [59]. When adopted for mission-critical tasks, backdoored models may incur severe consequences. For example, a backdoored facial recognition system may misclassify any person with a special pair of eyeglasses as an authorized employee and grant the access to confidential documents. Backdoor attacks are not confined to the vision domain, but have been extended to other domains, e.g., video [85], audio [31], and text [11].

There is a long line of works on backdoor attacks, and we focus on backdoor attacks in the outsourcing scenario with existing works listed in Table I. In the outsourcing scenario [20], the user outsources the training of DNN models to an untrusted third party (e.g., a cloud service provider) due to a lack of expert knowledge or resources. The user defines the model structure, the learning task and optionally provides the training data. The malicious cloud service provider trains and returns a backdoored model to the user. As shown in Table I, a good many works [23], [12], [60], [38], [40], [54] choose random triggers, which are simple but have a loose connection with the backdoor in the model compared with model-dependent triggers [45], [22]. The trigger mask, which determines the shape and the location of the trigger, is usually set arbitrarily. A visible

Network and Distributed Systems Security (NDSS) Symposium 2022
24-28 April 2022, San Diego, CA, USA
ISBN 1-891562-74-6
https://dx.doi.org/10.14722/ndss.2022.23012
www.ndss-symposium.org

Attack	Trigger design	Trigger mask		Invisible trigger?	Co-optimization?	Defense-resistant?
		Trigger shape	Trigger location			
[23]	random	random	random	×	×	-
[12]	random	random	random	×	×	-
[60]	random	random	multi-location	√	×	√
[38]	random	random	multi-location	√	×	√
[40]	random	random	multi-location	×	×	√
[54]	random	random	multi-location	×	×	√
[45]	model-dependent	random	random	×	×	-
[22]	model-dependent	random	multi-location	×	×	√
[56]	model-dependent	random	random	×	√	√
ATTEQ-NN	model-dependent	attention-based	attention-based	√	√	√

TABLE I. A COMPARISON OF STUDIES ON BACKDOOR ATTACKS IN THE OUTSOURCING SCENARIO.

trigger [23], [12], [40], [54], [45], [22], [56] may be identified by visual inspection, but invisible triggers usually yield a low attack success rate [59]. Moreover, most of the existing works [23], [45], [59], [28], [27], [38], [60], [74] decouple the trigger generation and the backdoor injection processes, which may lead to sub-optimal attack performance. A feasible attack has to be defense-resistant. Although various existing works claimed to be defense-resistant [22], they can still be detected by the latest defenses, such as NAD [36] and MNTD [80].

In this paper, we propose a novel attention-based QoE-aware backdoor attack, named ATTEQ-NN, which leverages the attention mechanism to generate the trigger mask and imposes QoE constraint on trigger generation. To achieve ideal attack performance while evading both visual and algorithmic inspections, we address the following challenges:

C1. How to generate a trigger that can effectively excite the backdoor in the model?

Existing works on model-dependent triggers did not fully exploit the design space of the trigger to intensify the attack effect. More specifically, the trigger mask, which constrains the trigger shape and the trigger location, is arbitrarily determined, e.g., a square at the bottom right corner of the image. To address this issue, we propose a new attention-based trigger mask determination approach. Due to the fact that a DNN model attends to the important region of an image (e.g., the face) and filters out irrelevant regions of the image (e.g., the background) to make a more robust judgment, different regions in the image have different degrees of influence on the prediction results. Therefore, we carefully compute the trigger mask based on attention maps of samples belonging to the target misclassification label. In this way, the generated trigger can strongly drive the model output towards the target label. Moreover, inspired by the co-optimization strategy [56], we jointly optimize the backdoor trigger and the backdoored model. In this way, the trigger and the backdoored model can boost each other until convergence, thus further improving the attack success rate. The experiments show that ATTEQ-NN can achieve a significantly higher attack success rate than the baselines especially with smaller trigger and complex datasets.

C2. How to naturalize the trigger to evade visual inspections?

The generated trigger is placed at the critical region of the image that attracts intensive attention both from the model and the human eyes, thus being vulnerable to visual inspections. To tackle this problem, we introduce a Quality-of-Experience (QoE) term into the loss function of trigger generation. QoE measures the perceptual quality of images according to subjective opinions of users [14]. Specifically, we leverage the Structural Similarity Index (SSIM) to quantify QoE, which assesses the structural similarities between the original image and the distorted image. In addition, we adjust the transparency of the trigger attached to samples. Our user study verifies that the backdoored samples appear to be similar to benign samples by human inspectors.

C3. How to inject the backdoor to realize effective and evasive attacks?

Given the generated trigger, the backdoor is injected by retraining the model with trigger-poisoned samples and clean samples. In most cases, a high poison ratio (the ratio of poisoned samples to all retraining samples) leads to a high attack success rate but may twist the decision boundary in a way that makes it possible for a meta-classifier to distinguish benign models and backdoored models [80]. Therefore, we design an alternating retraining strategy, which improves clean data accuracy and is surprisingly helpful in evading model-based defense strategies, e.g., MNTD [80].

We have conducted extensive experiments on VGG-Flower, CIFAR-10, GTSRB, CIFAR-100, and ImageNette to compare the performance of ATTEQ-NN with state-of-the-art backdoor attacks including BadNets [23], TrojanNN [45], HB [59], and RobNet [22]. The results show that ATTEQ-NN outperforms baselines in both attack success rate and clean data accuracy, especially when the poison ratio is low. We have carried out ablations studies to confirm that the attention-based trigger generation framework can increase the attack success rate by more than 10%. We carry out attacks in the physical world and show that the attack success rate of ATTEQ-NN is more than 37.78% under different lighting conditions and shooting angles. We demonstrate that ATTEQ-NN can maintain an attack success rate of more than 92.5% even if the user retrains the backdoored model using clean data. The user study confirms that the backdoored samples are indistinguishable from the clean samples by human inspections. Compared with the baselines, ATTEQ-NN is also shown to be more resistant to state-of-the-art defense approaches including model pruning [41], NAD [36], STRIP [21], NC [71], and MNTD [80]. Moreover, it is shown that ATTEQ-NN is robust to transfer learning.

To conclude, we make the following key contributions:

- We develop an attention-based trigger optimization framework for backdoor attacks, which determines the

trigger shape and the trigger location according to the focal area of the model to intensify the influence of the trigger on the prediction results, while existing works all used a random fixed trigger mask. With the optimized trigger mask, ATTEQ-NN achieves a high attack success rate with a smaller trigger.

- We propose a QoE-aware trigger generation method by introducing the QoE loss in the loss function to constrain the perceptual quality loss caused by the trigger.
- We design an alternating retraining method for backdoor injection to alleviate the decline of clean data prediction accuracy, which also helps resist state-of-the-art defenses, such as MNTD.
- Extensive experiments show that ATTEQ-NN outperforms state-of-the-art backdoor attacks, especially when the trigger size is small or the poison ratio is low. Moreover, ATTEQ-NN achieves higher evasiveness than baselines in terms of both human visual inspection and defense strategies.

II. BACKGROUND AND RELATED WORK

A. Deep Neural Network

Deep neural network (DNN) is a class of machine learning models that use serial stacked processing layers to capture and model complex nonlinear relationships. DNN can be used for classification or regression tasks. In the context of backdoor attacks, we focus on classification tasks. A DNN encodes a function $f_\theta : \mathcal{X} \to \mathcal{Y}$, where θ is the set of parameters of f. Given an input sample $x \in \mathcal{X}$, the DNN model f_θ outputs a probability vector over a set of possible classes $\mathcal{Y}$. The DNN model is usually trained by supervised learning. The training dataset $\mathcal{D}$ consists of samples $(x, y) \in \mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$, where y is the ground-truth label of input x. The parameters θ are determined through optimizing the loss function $\mathcal{L}(f(x;\theta), y)$ by stochastic gradient descent [5], [86].

Although DNN has shown great performance in many applications, the training of DNN models is prohibitive for resource-limited users. Firstly, the training process is data-hungry. To train a reliable prediction model, millions of training samples are needed. For instance, DeepFace [66], a face recognition model developed by Facebook, is trained by 4.4 million labeled photos from 4,030 people. Collecting and annotating such a huge dataset is difficult (if not impossible) for small companies. Secondly, expert knowledge is required to decide the most suitable model architecture, and massive computation and storage resources are needed to optimize millions of parameters.

Due to these hindrances, users are prone to outsource the training process to the cloud or download pre-trained models from the internet. For instance, Amazon Web Service (AWS) has launched a series of AI services to help with data processing, features extraction, and training of machine learning models. There are also many free pre-trained models on open-source platforms, e.g., Caffe Model Zoo [2]. Some popular models are downloaded in huge numbers. However, in these scenarios, by manipulating the training process, attackers may develop and distribute backdoored models to unsuspecting users.

B. Backdoor Attacks

Adversarial attacks against deep neural networks can be divided into two main categories according to attack phases: training-phase attacks and inference-phase attacks. Training-phase attacks actively interfere with the training process, while inference-phase attacks try to deceive or obtain private information of the trained model at the inference phase. Adversarial examples are typical inference-phase attacks, where seemingly innocent samples are constructed to mislead models into misclassification [7], [43], [63], [79]. Membership inference and model inversion attacks are also inference-phase attacks, where the attacker tries to infer private information of the training dataset [64], [61], [83], [49], [9]. Backdoor attacks are training-phase attacks, where the attacker injects backdoors into the model during training such that the backdoored model correctly classifies clean inputs but misclassifies inputs with a special trigger into a target label (targeted attacks) or any wrong label (untargeted attacks). Backdoor attacks can be characterized from four aspects: the attacker role, the trigger design, the trigger visibility, and the learning scenario.

Attacker role. The attacker can be a data vendor or a model vendor. The data vendor attacker publishes poisoned data for victim users to download but does not control the training process [12], [59]. The model vendor directly provides the model to victim users, e.g., the user outsources the model training to a malicious cloud or downloads a model from an untrusted source. Unlike the data vendor scenario, the attacker controls the training process, and can embed a backdoor into the model more effectively [23], [40], [24].

Trigger design. An effective trigger is the key to the success of backdoor attacks. Triggers can be designed as model-independent triggers or model-dependent triggers. Model-independent triggers, a.k.a. random triggers, are randomly chosen, e.g., a logo or a sticker, which are unrelated to the model. Early works on backdoor attacks [23], [12] mostly adopt random triggers. Model-dependent triggers are specifically generated based on the model [45], [74], [22]. The normal way to generate model-dependent triggers is first to select a neuron or a subset of neurons according to a certain criterion and then produce a trigger that can strongly excite the neuron(s). TrojanNN [45] selected the neuron with the largest sum of weights to the preceding layer. Wang et al. [74] put forward a ranking-based neuron selection method to choose neuron(s) that are difficult to be pruned and whose weights change little during retraining. Gong et al. [22] selected the neuron that can be most activated by the samples of the target label to evade pruning defenses and improve the attack performance. After neuron selection, the trigger is generated to maximize the activation of the selected neuron by gradient descent.

Trigger visibility. Early works of backdoor attacks do not pay much attention to trigger visibility on the ground that the trigger is confined to a small region of the image and can take the form of watermarks or trademarks to look natural. However, the defense strategy NeuralCleanse (NC) [71] specifically targets small-sized triggers, and model-dependent triggers do look unnatural and suspicious. To address this problem, a line of works have been devoted to studying invisible triggers. Saha et al. proposed hidden backdoor attacks [59] for data vendor attackers who provide poisoned data samples but do not control the labeling process. The poisoned samples look similar to

the samples of the target label in the pixel space, but are similar to the trigger-imposed samples of the source label in the feature space. Liao et al. [39] generated an adversarial example for a clean sample and then used the pixel difference between the clean sample and the adversarial example as the trigger. Li et al. [35] formulated the trigger generation as a bilevel optimization problem, where the trigger is generated to maximize the activation of a group of neurons through L_p-regularization to achieve invisibility. [39] and [35] are designed for model vendor attackers.

Learning scenario. Backdoor attacks can be launched both in the centralized learning scenario and the federated learning scenario [78], [6], [42], [53], [73]. In the federated learning scenario, the attacker can control the central aggregator or the participants. If the attacker can control the central aggregator, the scenario is similar to that of centralized learning. If the attacker can control the participants, one or multiple malicious participants controlled by the attacker aim to inject backdoors into the global model via manipulating local models. The main challenge is that the trigger will be diluted by the updates from benign participants. Therefore, it is important for participants to coordinate the trigger generation and backdoor injection in order to maximize the effect of the backdoor in the global model.

Compared with state-of-the-art backdoor attacks [23], [45], [74], [22], [60], [59], ATTEQ-NN features the following distinctions. First, we optimize the trigger mask using an attention mechanism, which locates the most influential areas in the data sample to amplify the impact of the trigger on the prediction results. As far as we know, all existing works chose a fixed mask with a random trigger shape (e.g., square) at a random location (e.g., right bottom corner). With trigger mask optimization, ATTEQ-NN is able to achieve a high attack success rate with a smaller trigger. Second, we consider QoE loss during trigger generation. In almost all existing works that generate model-dependent triggers, the generation process only targets at high activation of neurons in the model, producing triggers with abnormal and conspicuous patterns. We constrain the perceptual quality loss in trigger generation, yielding more stealthy and indiscernible trigger patterns. Last but not least, we design an alternating retraining strategy to improve clean data prediction accuracy after backdoor injection. The experiments show that this strategy assists ATTEQ-NN in resisting the state-of-the-art defense MNTD, while most of the existing attacks [23], [45], [74], [22], [60], [59] will be detected by MNTD. We adopt generic methods for neuron selection [22] and trigger concealment, i.e., adjust the trigger transparency.

C. Backdoor Defenses

In the face of rapidly-evolving backdoor attacks and their severe consequences, many defense strategies have been proposed. Since backdoor attacks tweak both the input (via the trigger) and the model, defenses can be categorized into data-based defenses [21], [15], [81], [68], [8], [69] and model-based defenses [71], [44], [8], [26], [10], [37]. Defenses may be conducted in the online phase (during run-time) [13], [21], [15], [81], [44], [48], [69] or the offline defense (before deployment) [68], [8], [71], [26], [10].

Data-based defense. Data-based defenses check whether an input sample contains a trigger or not. STRIP [21] is an online data-based defense that copies an input sample for multiple times and combines each copy with a different sample to generate a set of perturbed samples. If the sample is benign, the prediction results of the perturbed samples are expected to be random, i.e., have a high entropy. Otherwise, the perturbed samples are more likely to be classified as the target label, i.e., the entropy is low. Chen et al. proposed an offline data-based defense named activation clustering (AC) [8]. It is assumed that the activation of the last hidden layer, which reflects the high-level features used by a DNN model for prediction, is different for benign samples and malicious samples. If the samples belonging to a certain label can be clustered into two groups, the label is deemed as the target label in the backdoor attack. Tran et al. [68] investigated spectral signature based on statistical analysis to detect and eradicate malicious samples from a potentially poisoned dataset.

Model-based defense. Model-based defenses examine whether a model contains backdoors or not. Liu et al. [44] proposed Artificial Brain Stimulation (ABS), an online model-based defense, which leverages Electrical Brain Stimulation (EBS) to scan the target model to determine whether it is backdoored or not. Ma et al. [48] proposed NIC, another online model-based defense based on value invariant (VI) and provenance invariant (PI) analysis. Wang et al. [71] proposed NeuralCleanse (NC) to inspect the model in the offline phase. In the case of a backdoored model, it is assumed that a much smaller modification is needed to make input samples to be misclassified as the target label than any other benign label. NC checks whether there exists a label that satisfies this assumption to determine whether the model is backdoored. Huang et al. [26] proposed NeuronInspect that integrates output explanation with outlier detection to reduce computation cost. Chen et al. [10] proposed DeepInspect that utilizes reverse engineering to reverse the training data and then use a conditional generative model to get the probabilistic distribution of potentially backdoor triggers. More recently, Xu et al. proposed a Meta Neural Trojan Detection (MNTD) [80] pipeline that trains a meta-classifier to predict the existence of backdoors. Tang et al. proposed [67] statistical contamination analyzer (SCAn) to detect backdoors based on statistical properties of the features generated by backdoored models.

III. PROBLEM SCOPE AND THREAT MODEL

Problem scope. In this paper, we consider targeted backdoor attacks launched by a model vendor attacker in a centralized learning scenario.

Threat model. We adopt the same threat model as state-of-the-art backdoor attacks in the same problem scope [23], [40], [60]. The attacker is a malicious model vendor who can access the training dataset and train a backdoored model. The backdoored model is returned to a client or published online for users to download.

Attacker capability. The attacker controls the training process, knowing the model structure, parameters, and hyperparameters. The attacker has no knowledge of the validation dataset used by the client to test the received model, so that the model should have high clean data accuracy to pass the test.

Attack goal. The attacker's goal is to have the backdoored model resemble a benign model towards clean samples, but

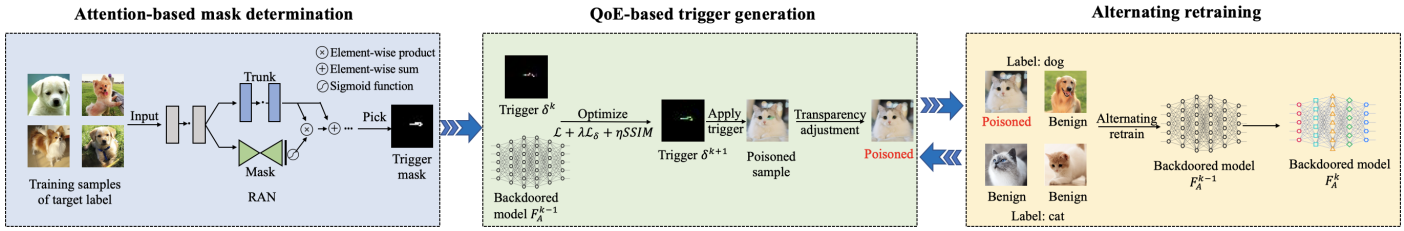


Fig. 1. Overview of ATTEQ-NN. ATTEQ-NN features an attention-based QoE-aware backdoor attack. We use the attention mechanism to pinpoint the critical region of the image as the trigger mask, introduce a QoE term in trigger generation and adjust transparency to evade human inspection, propose an alternating retraining strategy to improve the clean data accuracy and evade some model-based defenses.

make targeted false predictions in the presence of triggers.

$$F_A(x) = F_V(x),$$
$$F_A(x+\delta) = y_t,$$

where F_A is the backdoored model, F_V is a benign model, x is a clean sample, δ is the trigger, and y_t is the target misclassification label.

IV. ATTACK METHODOLOGY

In this section, we first present the general attack framework, then describe key components in the framework, including attention-based mask determination, QoE-based trigger generation, and alternating retraining strategy. The overview of ATTEQ-NN is shown in Fig. 1.

A. Backdoor Attack Framework

The two key components of backdoor attacks are trigger generation and backdoor injection. Firstly, a model-dependent trigger is generated according to the clean model. Then, the backdoor is injected by retraining the model with data samples poisoned by the trigger. Given the trigger mask M, the process of trigger generation is equal to seeking the optimal value assignments in the mask. The idea of trigger generation is to find a neuron in the clean model F_V as a bridge between the input trigger and the target output. To find the neuron, we first determine the proper layer at which the neuron should reside and then pinpoint the specific neuron. We do not select convolutional layers or pooling layers since the neurons in these layers only link to a small number of neurons in the preceding layer and the succeeding layer, thus having a weak response to the input trigger and a small influence on the output results. Therefore, we select the first fully-connected layer. Given the first fully-connected layer, we aim to find the neuron that is strongly associated with both the backdoor trigger and the target label. To evade pruning-based defenses, the selected neuron should be duly activated by benign samples. Therefore, following [22], we choose the neuron that has the highest number of activations when the model takes a set of N samples of the target label. Given the selected neuron, the trigger is generated by maximizing the activation value of the selected neuron through gradient descent [22]. After obtaining the backdoor trigger δ, a subset of samples from the training dataset is chosen. For each chosen sample (x, y), a poisoned sample (x_t, y_t) is constructed, where $x_t = x \otimes (1-M) + \delta \otimes M$, $\otimes$ denotes the element-wise product, and y_t is the target label. The poisoned samples are mixed with benign samples to retrain the clean model F_V for backdoor injection to obtain the backdoored model F_A.

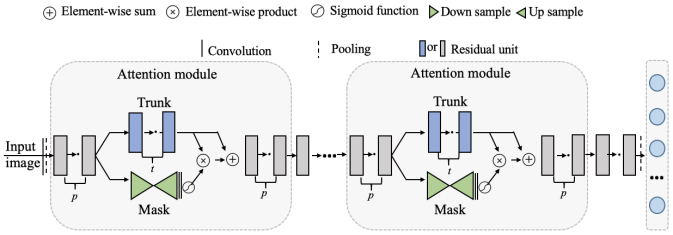


Fig. 2. The structure of RAN. In each attention module, the hyperparameter p represents the number of pre-processing residual units, and t represents the number of residual units in the trunk branch.

Since the attacker is capable of manipulating both the trigger and the model, we can formulate backdoor attacks as an optimization problem [56].

$$\min_{\delta, F_A} \mathcal{L}(x, F_V(x); F_A) + \lambda\mathcal{L}(x_t, y_t; F_A) + \omega\mathcal{L}_\delta(x_t, x). \quad (1)$$

where $\mathcal{L}(*)$ denotes the loss function and we have $\mathcal{L}_\delta(x_t, x) = ||x_t - x||_\infty = ||\delta||_\infty$. ω and λ are constant parameters to balance the clean data accuracy and the attack success rate. The first term optimizes the prediction accuracy of clean samples. The second and the third terms optimize the attack success rate of trigger-imposed samples while constraining trigger visibility.

Optimizing (1) is difficult since the backdoor trigger δ and the backdoored model F_A are co-dependent. Therefore, we partition the optimization problem (1) into two sub-problems, and solve the two sub-problems by alternately updating the backdoor trigger δ and the backdoored model F_A until convergence. We update the trigger and the model in the $k+1$-th iteration as

$$\begin{aligned} \delta^{k+1} &= \arg\min_{\delta} \left(\mathcal{L}(x_t, F_A^k) + \omega\mathcal{L}_\delta(x_t, x)\right), \\ F_A^{k+1} &= \arg\min_{F_A} \left(\mathcal{L}(x_t^{k+1}, F_A) + \lambda\mathcal{L}(x, F_V(x); F_A)\right). \end{aligned} \quad (2)$$

Given the current model F_A^k, we first optimize the trigger δ^{k+1} using Adam optimizer [29], which will be elaborated in the following sections. Then, given the optimized trigger δ^{k+1}, we obtain the optimized model F_A^{k+1} by retraining the model F_A^k with poisoned samples using δ^{k+1}. We summarize the algorithm of the co-optimization attack framework in Algorithm 1.

B. Attention-based Mask Determination

Before generating the trigger to activate the selected neuron, we need to construct an appropriate trigger mask, which

Algorithm 1 Attention-based QoE-aware backdoor attack

Require: Pre-trained benign deep neural network F_V, trigger size l^2, target label y_t, training samples $\mathcal{D}$, parameters λ, ω.
Ensure: Trigger δ, backdoored model F_A.
// Attention-based mask generation
$H_{opt}(x) = RAN(\mathcal{X}_t)$.
Select l^2 pixels with the highest weight in $H_{opt}(x)$ to form M.
// Initialize the trigger and the model
$k = 0$.
$\delta^k = Mask\_Initialize(M)$.
$F_A^k = F_V$.
while *not convergence* **do**
  $k = k + 1$.
  // QoE-aware trigger generation
  $\delta^k = Trigger\_Optimize(F_A^{k-1}, \lambda, \mathcal{D}, SSIM)$.
  // Alternating retraining for backdoor injection
  The retraining dataset $\mathcal{D}_r = Alt\_Retrain(k, \mathcal{D}, \delta^k)$.
  $F_A^k = Model\_Retrain(F_A^{k-1}, \delta^k, \omega, \mathcal{D}_r)$.
end while
return δ^k and F_A^k.

Algorithm 2 Alternating retraining

Require: Training dataset $\mathcal{D}$, the number of iteration k, the trigger δ.
Ensure: Retraining dataset $\mathcal{D}_r$.
Randomly select a subset of samples $\mathcal{D}_s \subset \mathcal{D}$.
if k is odd **then**
  $\mathcal{D}_r = \mathcal{D}_s$.
else
  Randomly select a subset of samples $\mathcal{D}'_s \subset \mathcal{D}_s$.
  for all $(x, y) \in \mathcal{D}'_s$ **do**
    Construct a poisoned sample $(x + \delta, y_t)$.
    $\mathcal{D}_s = \mathcal{D}_s \bigcup \{(x + \delta, y_t)\}$.
  end for
  $\mathcal{D}_r = \mathcal{D}_s$.
end if
return $\mathcal{D}_r$.

constrains the shape, size, and position of the trigger. The mask is a matrix with the number of rows and columns consistent with the height and the width of the original image sample. The elements in the matrix have a value of either 1 (denoting the trigger region) or 0 (denoting the non-trigger region).

In existing backdoor attacks, the triggers usually take the shape of a rectangular, a logo (e.g., apple logo), or watermarks [45], and the trigger location is usually set as the right/left bottom/up corners of the image [23], [45]. A larger trigger yields a higher attack success rate but is more visible and easier to be detected. Therefore, the trigger size is usually set as a small percentage of the size of the entire image. Unfortunately, such an arbitrary way of trigger mask determination may limit the effectiveness of the trigger [76].

In classification tasks, the classification model pays attention to different parts of the input image, similar to the human visual system. For a specific class (e.g., deer), most of the well-performed classification models of different structures usually pay attention to the same key features (e.g., antlers), as shown by many research works on explaining machine learning models using attention networks [70], [25], [16]. Manipulating the pixels of high importance is more likely to divert the classification results. Therefore, we propose an attention-based trigger mask determination method to select the most significant pixels as the trigger mask, based on which we generate a powerful trigger that achieves better attack performance.

In this paper, we utilize a residual attention network (RAN) [72] to obtain attention maps. As shown in Fig. 2, RAN is a feed-forward CNN with stacks of attention modules to extract the features for classification in the residual network. Each attention module consists of a trunk branch T and a soft mask branch S. The trunk branch processes features of neural networks, and the soft mask branch selects features by imitating the human cortex path [52]. RAN combines bottom-up and top-down learning methods to realize fast feed-forward processing and top-down attention feedback in one feed-forward procedure.

An input sample x_i first passes through a residual units to get x_i^1 as the input to the first attention module. In a RAN with L attention modules, the output of the l-th attention module is

$$H_{l,c}(x_i^l) = (1 + S_{l,c}(x_i^l)) \cdot T_{l,c}(x_i^l), c \in [1, 2, ..., C_l], \quad (3)$$

where $S_{l,c}(\cdot)$ and $T_{l,c}(\cdot)$ are the c-th channel of the mask branch and the trunk branch of the l-th attention module respectively, and C_l is the number of channels in the l-th attention module. The output $H_{l,c}$ will be fed into the $l+1$-th attention module after a residual unit.

In RAN, different attention modules play different roles. Low-level attention modules reduce the influence of unimportant features of the background, and high-level attention modules pick up important features that enhance classification performance. The output of the final attention module is the attention map with attention weights for corresponding pixels. The attention weights represent the degree of attention the model pays to each pixel, reflecting the contribution of each pixel to driving the prediction results of the image into a certain class.

The size of the obtained attention map is the same as the size of the output of RAN, which may be different from the size of the input. For instance, in our experiments, given a 32×32 image, the size of the output of the last attention module is 8×8, which is smaller than the input size. We upscale the attention maps to the same size of the input by bilinear interpolation [30]. We use $H(x_i)$ to denote the upscaled attention map of sample x_i.

We randomly select N clean samples of the target class y_t and attain N attention maps $\{H(x_i)\}_{i=1}^N$. Assuming that each sample has the same probability of occurrence, we choose the attention map that is closest to the average attention map for generality.

$$H_{opt}(x) = \arg\min_{x_i \in \mathcal{X}_t} ||\bar{H}(x) - H(x_i)||_2, \quad (4)$$

where $\mathcal{X}_t$ is the set of samples of the target label y_t, and $\bar{H}(x) = \frac{\sum_{j=1}^{j=N} H(x_j)}{N}$ is the average attention map.

Considering that most existing works use a contiguous square trigger of size $l \times l$ (l is the number of pixels), we also

TABLE II. COMPARISON OF ATTEQ-NN WITH BADNETS [23], TROJANNN [45], HB [59], AND ROBNET [22].

Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
					VGG-Flower-l					
10%	22.00%	96.0%	21.00%	94.50%	19.00%	94.0%	82.50%	95.50%	**94.50%**	**97.00%**
15%	22.50%	95.00%	22.00%	95.50%	24.00%	93.50%	80.50%	92.50%	**99.00%**	**96.00%**
20%	22.50%	96.50%	23.00%	96.50%	22.00%	94.50%	89.50%	91.50%	**99.00%**	**97.50%**
25%	24.50%	94.50%	27.00%	93.00%	33.00%	95.00%	91.00%	96.00%	**100.0%**	**98.00%**
30%	26.50%	97.00%	27.50%	94.00%	36.50%	95.00%	99.50%	95.00%	**100.0%**	**98.50%**
					CIFAR-10					
Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
1%	10.00%	87.98%	11.82%	85.95%	27.83%	88.02%	32.7%	87.92%	**44.69%**	**88.98%**
3%	10.34%	90.92%	12.28%	**90.99%**	31.24%	87.55%	65.79%	88.23%	**86.84%**	88.35%
5%	93.93%	90.02%	97.09%	89.87%	30.07%	**90.03%**	95.62%	88.39%	**97.29%**	88.90%
10%	95.43%	88.90%	98.05%	89.67%	29.07%	85.22%	95.06%	87.84%	**99.26%**	**90.10%**
15%	97.06%	88.32%	98.77%	87.69%	44.74%	84.89%	96.30%	87.65%	**99.33%**	**89.12%**
20%	98.06%	89.54%	**99.75%**	85.20%	60.08%	86.07%	96.93%	87.64%	99.01%	**90.07%**
					GTSRB					
Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
0.3%	22.01%	92.57%	25.55%	94.14%	8.01%	89.43%	26.81%	96.60%	**90.88%**	**97.15%**
0.5%	46.52%	94.09%	47.50%	95.38%	12.01%	90.08%	56.99%	96.89%	**93.30%**	**97.08%**
1%	96.25%	94.46%	96.65%	94.98%	23.60%	89.07%	**98.84%**	95.53%	96.75%	**96.94%**
3%	97.81%	95.00%	97.93%	94.46%	77.25%	89.67%	**99.95%**	96.80%	99.39%	**97.11%**
5%	98.08%	96.22%	98.10%	96.25%	77.74%	88.10%	99.51%	**97.36%**	**99.97%**	97.19%
7%	98.91%	96.54%	98.97%	95.76%	78.27%	88.31%	99.20%	96.04%	**99.91%**	**96.81%**
					CIFAR-100					
Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
0.1%	1.29%	73.57%	1.61%	74.66%	3.04%	68.79%	17.01%	73.02%	**96.53%**	**74.55%**
0.3%	2.52%	73.48%	2.25%	74.28%	3.88%	69.52%	98.01%	71.45%	**98.66%**	**75.06%**
0.5%	2.47%	73.08%	2.5%	73.62%	3.68%	67.03%	97.33%	71.67%	**99.94%**	**74.91%**
1%	2.56%	73.36%	3.27%	72.99%	7.44%	69.94%	98.66%	71.72%	**99.78%**	**74.64%**
3%	90.38%	71.59%	95.61%	73.13%	62.73%	70.28%	99.49%	72.44%	**99.84%**	**75.44%**
					ImageNette					
Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
5%	11.52%	91.49%	11.41%	91.77%	10.60%	91.40%	60.31%	88.96%	**88.82%**	**91.95%**
10%	13.45%	90.50%	14.15%	90.24%	11.81%	89.93%	68.78%	86.42%	**90.83%**	**90.59%**
15%	14.26%	89.00%	15.28%	88.14%	14.42%	91.40%	81.98%	88.82%	**92.16%**	**92.40%**
20%	21.53%	86.50%	24.89%	85.83%	15.34%	88.27%	85.50%	88.16%	**95.01%**	**91.57%**
30%	35.13%	71.28%	37.83%	70.54%	18.81%	85.32%	92.92%	84.87%	**97.58%**	**91.46%**
					VGG-Flower-h					
Ratio	BadNets [23] ASR	BadNets [23] CDA	TrojanNN [45] ASR	TrojanNN [45] CDA	HB [59] ASR	HB [59] CDA	RobNet [22] ASR	RobNet [22] CDA	ATTEQ-NN ASR	ATTEQ-NN CDA
10%	11.00%	95.50%	12.50%	95.50%	5.50%	96.00%	34.50%	95.00%	**40.00%**	**98.50%**
15%	15.50%	95.50%	16.50%	95.00%	6.50%	95.00%	58.50%	95.00%	**83.00%**	**96.00%**
20%	20.00%	94.00%	23.00%	96.50%	15.50%	95.50%	60.00%	97.00%	**92.50%**	**97.50%**
25%	28.00%	96.50%	27.00%	94.50%	19.50%	94.50%	73.00%	94.00%	**98.50%**	**97.50%**
30%	30.00%	95.50%	29.00%	95.50%	21.50%	93.00%	76.50%	95.50%	**100.0%**	**97.00%**

use the conventional expression $l \times l$ to denote the trigger size. To make a fair comparison, we choose the top l^2 pixels with the highest attention values as the trigger region, i.e., trigger mask M in our attack for evaluation. Note that triggers in backdoor attacks are not required to be contiguous, and the most important pixels we choose according to the attention maps may not be contiguous. We can constrain the selected important pixels to be continuous, and ATTEQ-NN will still be effective.

C. QoE-based Trigger Generation

Although the model-dependent trigger generated by Algorithm 1 can achieve a high attack success rate, the trigger may be conspicuous and easily detected by human visual inspection. Therefore, the invisibility of the trigger is crucial for successful backdoor attacks. Unfortunately, only a few existing works consider invisible triggers [59], [39], [35], and they usually achieve a subpar attack success rate.

We propose a QoE-aware trigger generation method by introducing Structural Similarity Index Measure (SSIM) [75] to the loss function and adjusting the transparency of the backdoor trigger. SSIM is a commonly-used Quality-of-Experience (QoE) metric [14] that quantifies the differences in luminance, contrast, and structure between the original image and the distorted image.

$$SSIM = A(x, x')^{\alpha} B(x, x')^{\beta} C(x, x')^{\gamma}, \tag{5}$$

where $A(x, x')$, $B(x, x')$, and $C(x, x')$ quantify the luminance similarity, contrast similarity and structure similarity between the original image x and the distorted image x'. α, β, and γ are parameters. We introduce SSIM into the loss function to optimize the trigger.

$$\delta^* = \arg\min_{\delta} \big(\mathcal{L}(x_t, F_A) + \lambda \mathcal{L}_{\delta}(x_t, x) + \eta SSIM\big), \tag{6}$$

where η balances the attack success rate and the QoE of poisoned images. According to our extensive experiments, we empirically set η as 0.1.

We adjust the transparency of the backdoor trigger when added to clean samples to further hide the backdoor trigger. A higher transparency value yields a more imperceptible trigger but a lower attack success rate. Setting a proper transparency value is a trade-off between the attack success rate and the concealment of the attack. We will evaluate the impact of transparency on the attack performance in Section V.

D. Alternating Retraining

The conventional backdoor injection approach is to retrain the model with both clean and poisoned samples. However, we find by experiments that even if the poison ratio is as small as 0.3%, the drop of clean data accuracy can be as high as 7.82% (HB [59] attacks on the GTSRB dataset as shown in Table II). The user may reject the backdoored model due to this performance degradation on the clean validation dataset. Moreover, the decision boundaries may be twisted too much by poisoned samples such that the backdoored models may be separated from benign models by a meta-classifier [80].

To tackle these problems, we propose an alternating retraining strategy, which alleviates the decline of clean data prediction accuracy and makes the backdoored model more resistant to defenses [80]. As shown in Algorithm 2, during the process of iterative update, if the iteration index k is an even number, we update (retrain) the model using both poisoned samples and clean samples. Otherwise, we only retrain the model using clean samples. An intuitive question is whether the attack success rate will decrease due to such a retraining strategy. Our experiments show that the attack success rate will have a slight drop or even slight augmentation while the clean data accuracy has an appreciable improvement. For instance, on the CIFAR-10 dataset with a trigger size of 3×3 (Table V), the traditional retraining method yields a clean data accuracy of 89.07% and an attack success rate of 99.62%, while the alternation retraining strategy achieves a prediction accuracy of 90.23% and an attack success rate of 99.56%. Interestingly, we discover that the alternating retraining strategy is also helpful for evading model-based defenses, such as MNTD [80], as verified in Section V-G. We attribute it to the fact that the alternating retraining strategy narrows the gap between the backdoored model and the benign model. Note that multiple triggers targeting different labels can be injected into a model by repeating the above steps.

V. IMPLEMENTATION AND EVALUATION

We compare the attack performance of ATTEQ-NN with state-of-the-art attacks to verify its effectiveness. Then, we conduct an ablation study to evaluate the usefulness of different components in the attack framework. After that, we examine the

TABLE III. IMPACT OF DIFFERENT TARGET LABELS. DEFAULT TRIGGER SIZE AND POISON RATE EXCEPT FOR CIFAR-100 (POISON RATE 2%) AND VGG-FLOWER-H (POISON RATE 30%)

	Label 1		Label 3		Label 5		Label 7		Label 9		Mean	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
VGG-Flower-l	99.00%	96.00%	99.00%	97.00%	98.00%	97.00%	99.00%	97.50%	99.50%	97.50%	98.90%	97.00%
CIFAR-10	97.95%	88.96%	97.90%	88.44%	97.95%	88.45%	97.92%	88.95%	97.95%	88.56%	97.93%	88.67%
GTSRB	99.83%	97.68%	99.45%	97.24%	99.26%	97.59%	98.74%	96.41%	99.52%	97.80%	99.36%	97.34%
CIFAR-100	99.98%	76.50%	99.77%	76.59%	99.92%	76.97%	99.72%	76.93%	99.17%	76.80%	99.71%	76.76%
	Label 0		Label 2		Label 4		Label 6		Label 8		Mean	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
ImageNette	91.82%	92.56%	91.41%	92.36%	91.41%	91.95%	91.73%	91.85%	91.77%	91.52%	91.63%	92.05%
VGG-Flower-h	99.50%	97.50%	99.50%	98.00%	99.00%	98.50%	98.50%	98.50%	99.50%	98.00%	99.20%	98.10%

TABLE IV. THE PERFORMANCE OF ATTEQ-NN WHEN EXTENDED TO MULTI-TRIGGER BACKDOOR ATTACKS. NOTE THAT ASR IS THE MEAN ASR OF DIFFERENT TARGET LABELS.

# of triggers	3		5		7	
	ASR	CDA	ASR	CDA	ASR	CDA
VGG-Flower-l	92.17%	96.50%	92.00%	96.00%	90.14%	97.00%
CIFAR-10	95.68%	90.62%	95.48%	90.86%	95.77%	90.62%
GTSRB	94.14%	96.93%	93.16%	96.37%	95.73%	96.45%
CIFAR-100	99.56%	75.67%	99.72%	75.63%	99.66%	76.07%
ImageNette	90.10%	90.31%	90.48%	90.01%	90.07%	90.83%
VGG-Flower-h	74.17%	95.00%	74.10%	95.50%	74.00%	95.50%

practicality of the attack in the physical world. We demonstrate the viability of the attack when the user fine-tunes the model with clean data. We evaluate whether ATTEQ-NN is robust to transfer learning. We verify the evasiveness of ATTEQ-NN under state-of-the-art defense strategies. Finally, we present the user study to assess the visibility of the backdoored samples under visual inspection. All experiments are implemented in Python and run on a 14 core Intel(R) Xeon(R) Gold 5117 CPU @2.00GHz and NVIDIA GeForce RTX 2080 Ti GPU machine running Ubuntu 18.04 system.

We conduct experiments on various machine learning tasks, covering different datasets and DNN architectures. Specifically, we use six image datasets, including VGG-Flower (VGG-Flower-l ($32*32$) and VGG-Flower-h ($224*224$)) [55], CIFAR-10 [33], GTSRB [65], CIFAR-100 [33], and ImageNette [19]. We utilize VGG-16, ResNet-18, VGG-16, ResNet-34, ResNet-50, and ResNet-18 structures to train models for these six datasets, respectively. The default target label is label 0 for VGG-Flower-l, label 3 for VGG-Flower-h, label 2 for CIFAR-10, label 10 for GTSRB, label 0 for CIFAR-100, and label 3 for ImageNette. The default poison ratio is 20% for VGG-Flower-l, 15% for VGG-Flower-h, 5% for CIFAR-10, 5% for GTSRB, 0.5% for CIFAR-100, and 15% for ImageNette. The default trigger size is 4×4 for VGG-Flower-l, 8×8 for VGG-Flower-h, 4×4 for CIFAR-10, 3×3 for GTSRB, 2×2 for CIFAR-100, and 8×8 for ImageNette. The default transparency value is 0.4 for VGG-Flower-l, CIFAR-10, GTSRB, CIFAR-100 and 0.7 for ImageNette and VGG-Flower-h. Note that the baselines and ATTEQ-NN have the same experiment settings (e.g., trigger size, poison ratio, epoch, learning rate) in the attack performance comparison. The evaluation of the high-resolution datasets is based on the open-source tool TROJAN ZOO [57].

We utilize two evaluation metrics, i.e., attack success rate (ASR) and clean data accuracy (CDA). We choose four state-of-the-art backdoor attacks as the baselines, i.e., BadNets [23], TrojanNN [45], HB [59], and RobNet [22]. We adopt a 92-layer RAN with 6 attention modules. We set $C_1 = 128, C_2 = 256, C_3 = 256$ following the original RAN model [72], and $C_4 = C_5 = C_6 = 1$ to aggregate all information into a single attention map.

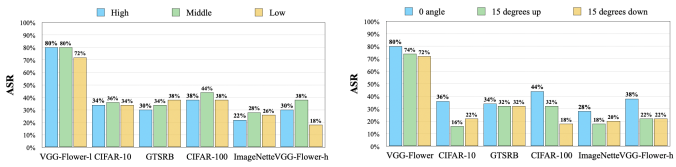
More details of datasets, DNN models, evaluation metrics, and the baselines are shown in the Appendix.

A. Evaluation Results

We first present the comparison results of ATTEQ-NN and baselines, then evaluate the impact of target label, trigger size, transparency, and trigger contiguity on the performance of ATTEQ-NN.

1) Comparison with Baselines: As shown in Table II, ATTEQ-NN has higher ASR than the baselines for all six datasets, especially when the poison ratio is small. For example, ATTEQ-NN achieves ASR of 94.5%, 44.69%, 90.88%, 96.53% on VGG-Flower-l, CIFAR-10, GTSRB, CIFAR-100 models at poison ratios of 10%, 1%, 0.3%, 0.1% respectively, while BadNets only reaches ASR of 22.0% (VGG-Flower-l), 10.00% (CIFAR-10), 22.01% (GTSRB), 1.29% (CIFAR-100). Compared with HB that uses invisible triggers, ATTEQ-NN achieves a significantly higher ASR across all datasets at all poison ratios. For the high-resolution datasets, ATTEQ-NN achieves an ASR of 88.82% and 83.00% on VGG-Flower-h and ImageNette at only 5% and 15% poison ratio, which is much higher than the baselines, especially BadNets, TrojanNN, and HB. Moreover, ATTEQ-NN can maintain a high CDA. For GTSRB and CIFAR-10, the ASR of ATTEQ-NN is lower than RobNet and TrojanNN by less than 1% in only 3 cases. This is because RobNet and TrojanNN use visible triggers, which are more effective but more conspicuous. In comparison, ATTEQ-NN implements invisible triggers, which achieve higher or comparable ASR by leveraging the attention mechanism. As the poison ratio increases, the ASR will increase, but the CDA of ATTEQ-NN will fluctuate (either increase or decrease). Therefore, the poison ratio needs to be adjusted according to the attack goal. The success of ATTEQ-NN can be attributed to the proposed components that improve the performance of the backdoor attacks, especially attention-based mask determination approach.

We compare the invisibility of the backdoored samples across all attacks, as shown in Fig. 8 and Fig. 9 (Appendix). We can see that except for HB and ATTEQ-NN, the triggers of all other baselines are conspicuous and easily detected by human eyes. Compared with HB, ATTEQ-NN produces more indiscernible triggers in some cases. HB can not achieve a high ASR as ATTEQ-NN.



(a) Impact of lighting intensity (b) Impact of shooting angles

Fig. 3. Effectiveness of attacks in the physical world under different lighting intensity and shooting angles.

2) Impact of Target Label: We then explore whether the target label will impact the attack performance of ATTEQ-NN. Table III shows that the attack performance of ATTEQ-NN is robust under different target labels.

3) Impact of Trigger Size: Trigger size plays a vital role in backdoor attacks. In ATTEQ-NN, the pixels of the trigger are selected by the attention mechanism, thus are not necessarily contiguous. As shown in Table X, Table VIII, and Table IX (Appendix). We can see that as the trigger size increases, the ASR will also increase but the SSIM value of the backdoored samples will decrease.

4) Impact of Transparency Value: The higher the transparency value is, the more imperceptible the trigger is. As shown in Fig. 7 (Appendix) and Table XI (Appendix), ATTEQ-NN achieves the highest ASR when the transparency value is 0 (i.e., completely opaque), but such triggers are visible and easy to be detected. As the transparency increases, the ASR will decrease, but the concealment and QoE of the backdoored images will increase. When the transparency value is set as 0.4∼0.5 for VGG-Flower-l, CIFAR-10, GTSRB, and CIFAR-100 or 0.6∼0.7 for ImageNette and VGG-Flower-h, the human eyes can hardly discern the trigger (according to the user study). Thus in the experiments, we set the transparency value as 0.4 or 0.7 by default.

B. Ablation Study

In this section, we conduct an ablation study to examine the necessity of attention-based mask determination, co-optimization, and alternating retraining strategies. The results are shown in shown in TableV. The "Base" attack is a traditional backdoor attack with square-shaped model-dependent triggers placed at the bottom right corner of the image. The "Base+Attention" attack uses the attention mechanism to determine the trigger mask. The "Base+Attention+Iter" attack iteratively updates the trigger and the backdoored model. The "All" attack is the complete attack with attention-based mask determination, co-optimization, and alternating retraining strategies. In this section, we set the transparency value as 0.

1) Attention-based Mask Determination: Comparing "Base" and "Base+Attention", we can observe that the attention mechanism can significantly improve ASR, especially when the trigger is very small. For example, when the trigger size is 1×1, the ASR is 18.5% for VGG-Flower-l using the "Base" attack, but reaches as high as 29.0% using the "Base+Attention" attack. The increment is more than 10%. Similarity, for CIFAR-10, the improvement in ASR when the trigger size is 1×1 is more than 10% with the attention mechanism. As the trigger size becomes larger, the difference in ASR between the "Base" attack and the "Base+Attention" attack shrinks as the "Base" attack has more chance to select the pixels of high importance.

TABLE V. THE IMPACT OF ATTENTION-BASED MASK DETERMINATION, ITERATIVE UPDATE, AND ALTERNATING RETRAINING ON THE PERFORMANCE OF ATTEQ-NN. THE "BASE" ATTACK IS A TRADITIONAL BACKDOOR ATTACK WITH MODEL-DEPENDENT TRIGGERS. THE "BASE + ATTN" ATTACK USES ATTENTION MECHANISM TO DETERMINE THE TRIGGER MASK. THE "BASE+ATTN+ITER" ITERATIVELY UPDATES THE TRIGGER AND THE BACKDOORED MODEL. THE "ALL' ATTACK IS THE COMPLETE ATTACK OF ATTEQ-NN APART FROM THE TRANSPARENCY SETTING.

Size	VGG-Flower-l							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
1 × 1	18.50%	93.50%	29.00%	92.50%	34.50%	**94.50%**	**35.00%**	**94.50%**
2 × 2	32.00%	96.00%	42.00%	95.50%	60.00%	**97.00%**	**71.50%**	**97.00%**
3 × 3	48.50%	93.50%	74.50%	94.50%	**100.0%**	95.50%	99.50%	**96.00%**
4 × 4	51.00%	94.50%	98.00%	95.50%	**100.0%**	96.50%	**100.0%**	**98.00%**
Size	CIFAR-10							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
1 × 1	44.89%	87.22%	55.63%	86.66%	**81.33%**	87.71%	80.33%	**87.94%**
2 × 2	58.26%	87.94%	95.60%	87.88%	**99.44%**	89.28%	99.14%	**90.07%**
3 × 3	91.01%	87.55%	97.70%	88.43%	**99.62%**	89.07%	99.56%	**90.23%**
4 × 4	95.62%	88.39%	98.10%	88.76%	**99.77%**	89.35%	97.55%	**89.91%**
Size	GTSRB							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
1 × 1	78.67%	93.29%	87.67%	96.06%	97.98%	96.67%	**98.78%**	**97.03%**
2 × 2	75.01%	95.52%	97.01%	95.25%	**99.73%**	97.14%	99.00%	**97.38%**
3 × 3	93.49%	96.75%	94.72%	96.81%	98.97%	96.69%	**99.98%**	**97.00%**
4 × 4	91.74%	96.89%	93.40%	97.74%	99.80%	97.50%	**99.87%**	**97.78%**
Size	CIFAR-100							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
1 × 1	93.54%	71.84%	95.41%	72.69%	97.33%	74.61%	**97.61%**	**74.66%**
2 × 2	97.33%	71.67%	99.56%	71.60%	**99.95%**	74.22%	99.71%	**75.07%**
3 × 3	99.39%	72.08%	99.81%	72.64%	**99.98%**	75.31%	99.71%	**75.34%**
4 × 4	99.19%	72.96%	99.28%	73.76%	**99.71%**	73.80%	99.64%	**75.23%**
Size	ImageNette							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
2 × 2	51.69%	88.14%	78.62%	88.76%	82.91%	88.73%	**90.94%**	**91.26%**
4 × 4	69.83%	82.22%	86.57%	83.39%	89.88%	88.79%	**90.57%**	**90.32%**
8 × 8	79.11%	83.54%	88.10%	86.14%	92.51%	86.14%	**98.39%**	**90.93%**
12 × 12	80.05%	82.50%	90.33%	83.18%	92.20%	87.77%	**99.57%**	**88.59%**
Size	VGG-Flower-h							
	Base		Base+Attn		Base+Attn+Iter		All	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
8 × 8	46.50%	94.50%	69.00%	94.50%	74.50%	94.00%	**98.50%**	**97.00%**
12 × 12	47.00%	94.50%	77.00%	**95.50%**	93.00%	**95.50%**	**98.50%**	**95.50%**
16 × 16	51.00%	95.50%	85.50%	96.50%	94.50%	96.50%	**99.00%**	**97.00%**
20 × 20	56.50%	96.00%	86.00%	95.00%	95.00%	96.00%	**99.50%**	**97.00%**

2) Co-optimization: Compared with the "Base+Attention" attack, the "Base+Attention+Iter" attack further increases ASR. We can observe that co-optimization improves both ASR and CDA.

3) Alternating Retraining: The alternating retraining strategy mainly improves the prediction accuracy of the clean samples, while slightly decreasing the attack success rate in certain cases. The following experiments will show that the alternating retraining strategy helps bypass model-based defense approaches, such as MNTD.

4) Convergence: By adopting the co-optimization and alternating retraining strategies, the complexity of the trigger injection process of ATTEQ-NN is relatively higher than conventional backdoor attacks. According to our experiments on the six datasets, the average number of epochs needed for

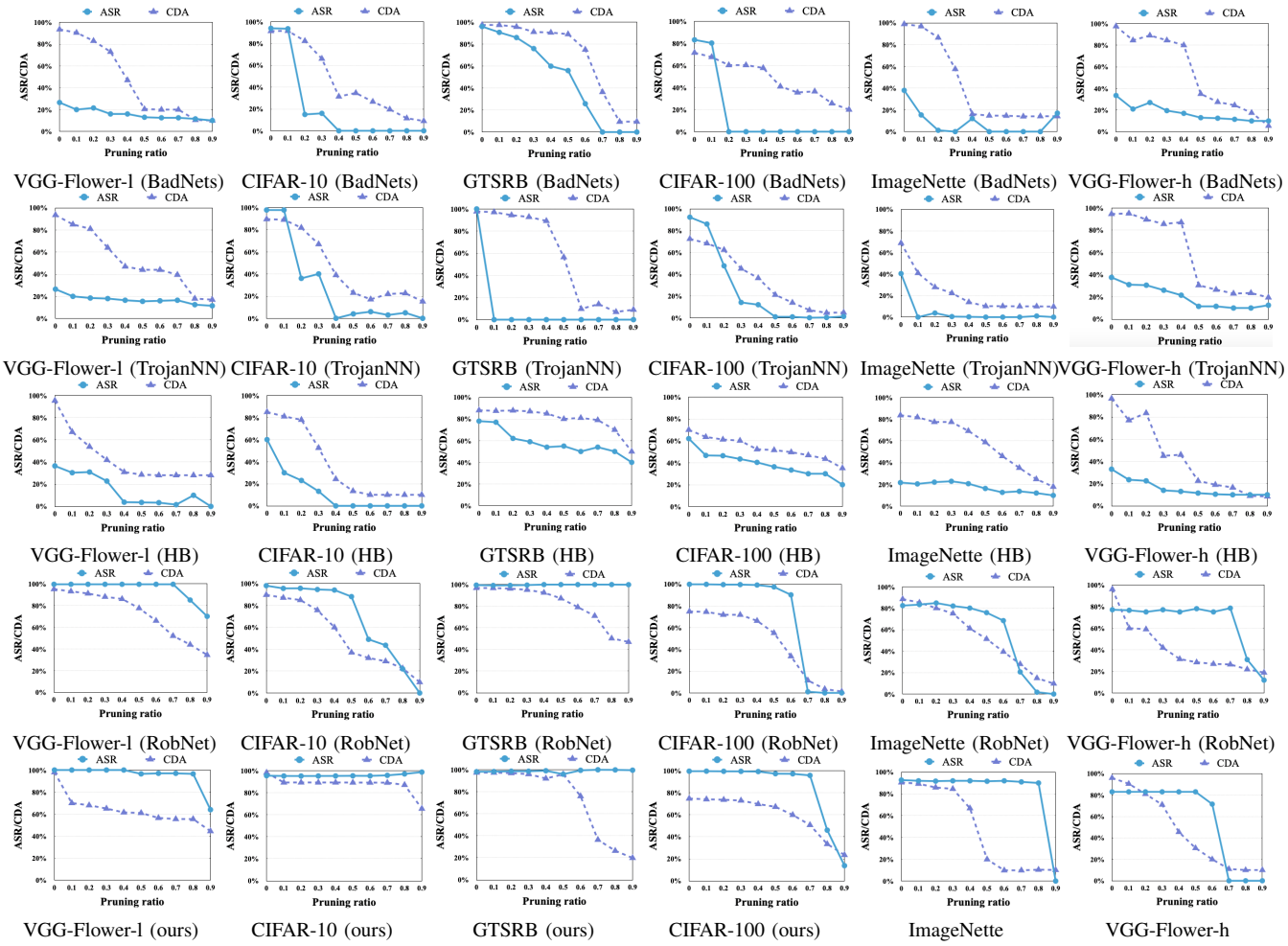


Fig. 4. The attack performance after applying model pruning to baseline attacks and ATTEQ-NN.

the convergence of ATTEQ-NN is approximately 300, which is slightly higher but on the same order of magnitude as BadNets [23] (average 125), TrojanNN [45] (average 125), HB [59] (average 100), and RobNet [22] (average 70).

C. Physical World

Apart from the digital domain, we also explore whether ATTEQ-NN is effective in the physical world. For instance, the attacker may print the trigger and attach it to traffic signs to mislead automatic vehicles. We randomly select 50 samples from each dataset, resulting in a total of 300 samples. We print out each sample of VGG-Flower-l, CIFAR-10, GTSRB, and CIFAR-100 as 4cm $\times$ 4cm and each sample of ImageNette and VGG-Flower-h as 6cm $\times$ 6cm on a white paper. We take a photo of the printed sample using a smartphone with Samsung ISOCELL Bright HMX camera 30cm away from the sample. Then, we digitally crop each image to remove the edges of the white paper and resize it to 32×32 (VGG-Flower-l, CIFAR-10, GTSRB, and CIFAR-100) or 224×224 (ImageNette and VGG-Flower-h). Finally, we feed the images to the model for prediction. Since the lighting condition may affect the attack performance in the physical world, we take photos of each sample under strong lighting intensity (approximately 1200 Lux), medium lighting intensity (approximately 300 Lux), and low lighting intensity (approximately 20 Lux). Moreover, we take photos of each sample at $0°$ and $15°$ (up and down) under medium lighting intensity. The transparency value is set as 0.

For lighting conditions, as shown in Fig. 3(a), the ASR of ATTEQ-NN is the highest under medium lighting intensity in most cases. Strong light will cause over-exposure, which degrades the attack performance, and under low lighting intensity, the mobile phone camera will automatically increase the saturation of the photo and change its color tone, which alters the trigger and weakens the attack performance. In terms of the shooting angles, as shown in Fig. 3(b), the ASR decreases at oblique angles as the trigger will be distorted. Although the trigger is small enough to evade human eyes inspections, it can be captured by a high-end camera (e.g., a resolution of 108 million pixels), making ATTEQ-NN effective in the physical world.

D. Extension to Multi-trigger Backdoor Attacks

We extend ATTEQ-NN to multi-trigger backdoor attacks by repeating the trigger generation and the backdoor injection process, where each trigger targets a different target label. Table IV shows that the extended ATTEQ-NN can maintain high ASR and CDA in multi-trigger backdoor attacks.

TABLE VI. THE COMPARISON OF THE ORIGINAL BACKDOORED MODEL AND THE MODEL AFTER APPLYING NAD TO BASELINE ATTACKS AND ATTEQ-NN.

Datasets		BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
		ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
VGG-Flower-l	Original	26.50%	97.00%	27.50%	94.00%	36.50%	95.00%	99.50%	95.00%	99.50%	97.50%
	NAD [36]	3.50%	93.50%	15.00%	93.00%	18.40%	95.00%	29.00%	96.00%	92.50%	97.00%
CIFAR-10	Original	93.93%	90.02%	97.90%	89.87%	60.08%	86.07%	95.62%	88.39%	99.76%	89.46%
	NAD [36]	10.06%	90.97%	23.07%	88.67%	15.33%	83.80%	10.83%	87.31%	99.19%	88.31%
GTSRB	Original	98.08%	96.22%	98.10%	96.25%	77.74%	88.10%	99.51%	97.36%	99.75%	97.17%
	NAD [36]	13.08%	96.39%	7.32%	95.78%	2.34%	88.00%	5.36%	96.31%	90.14%	96.69%
CIAFR-100	Original	90.38%	71.59%	95.61%	73.13%	62.73%	70.28%	99.49%	72.44%	99.58%	74.62%
	NAD [36]	6.09%	67.05%	16.71%	67.57%	3.28%	68.87%	9.47%	69.86%	94.23%	73.92%
ImageNette	Original	35.13%	71.28%	37.83%	70.54%	18.81%	85.32%	81.98%	88.82%	92.16%	92.40%
	NAD [36]	4.58%	72.15%	1.01%	74.76%	9.60%	85.40%	7.05%	84.24%	90.56%	92.31%
VGG-Flower-h	Original	30.00%	95.50%	29.00%	95.50%	21.50%	93.00%	76.50%	95.50%	83.00%	96.00%
	NAD [36]	9.50%	95.00%	10.50%	94.00%	8.50%	95.50%	16.00%	92.50%	80.00%	94.00%

TABLE VII. THE PERFORMANCE OF ATTEQ-NN IN TRANSFER LEARNING SCENARIOS.

Source dataset	Transfer dataset	CDA	ASR	Baseline CDA
CIFAR-10	CIFAR-100	87.50%	99.30%	88.10%
CIFAR-10	GTSRB	94.32%	99.02%	96.75%
CIFAR-10	VGG-Flower-l	90.00%	99.50%	91.00%
VGG-Flower-l	CIFAR-100	83.80%	65.22%	91.41%
VGG-Flower-l	GTSRB	93.59%	78.30%	98.04%
ImageNette	VGG-Flower-h	95.65%	90.00%	96.52%

E. Robustness to Fine-tuning

Fine-tuning is a commonly-used technique to refine a pre-trained DNN model, which is much faster and cheaper than training a complex DNN model from scratch. Recently, researchers discovered that fine-tuning might also be used to erase the backdoor in models [41], [45], [36].

We fine-tune the backdoored models of ATTEQ-NN and the baselines across six datasets. The maximum number of frozen layers is determined by the model structure. The number of epochs is set as 50. The learning rate of CIFAR-10, CIFAR-100, ImageNette, and VGG-Flower-h is set as 10^{-3}, and the learning rate of GTSRB and VGG-Flower-l is set as 10^{-4}. Following [36], we fine-tune the original backdoored model on 10% benign samples of the original training dataset. As shown in Table XIII (Appendix) and Table XII (Appendix), when all the layers of the backdoored model are fine-tuned (i.e., the number of frozen layers is 0), we can see that all the baselines are ineffective while ATTEQ-NN can still maintain a high ASR. As the number of frozen layers increases, more weights of the original backdoored model will be kept, and the attack performance will be better. The robustness of ATTEQ-NN to fine-tuning is due to the alternating retraining process, which makes the backdoored model more similar to the benign model and is less changed in the fine-tuning process.

F. Robustness to Transfer Learning

We investigate whether ATTEQ-NN is effective in transfer learning scenarios. We construct six transfer learning scenarios in Table VII. The attacker trains a teacher model with P layers on the source dataset with the backdoor. The student model is trained by the user based on the teacher model. More specifically, the student model copies the first N layer of the teacher model and retrains the last $P - N$ layers with the transfer dataset. Note that the student model keeps the target label of the backdoor attacks [82]. We also train a student model based on a clean teacher model in the same settings for comparison, which provides the baseline of clean data accuracy in the transfer learning scenario.

As shown in Table VII, the CDA of the student model transferred from the backdoored teacher model is similar to that of the student model transferred from the clean teacher model. Meanwhile, the ASR of the student model transferred from the backdoored teacher model is high, indicating that the backdoor has been preserved in the student model. This demonstrates that ATTEQ-NN is robust to transfer learning.

G. Evading State-of-the-Art Defenses

In this section, we explore whether ATTEQ-NN can evade state-of-the-art backdoor defenses, including model pruning, NAD [36], STRIP [21], NC [71], and MNTD [80]. For baseline attacks, we adjust the poison ratio as the default poison ratio is ineffective in certain cases. In particular, we set the poison ratio as 30% in all baselines for VGG-Flower-l and VGG-Flower-h. We set the poison ratio as 20% in HB for CIFAR-10. We set the poison ratio as 3% in BadNets, TrojanNN, and HB for CIFAR-100, and 30% in BadNets, TrojanNN, and HB for ImageNette. Others adopt the default poison ratio.

1) Model Pruning: Existing studies [41] have shown that model pruning is helpful to disable a backdoor. The defender first ranks neurons in the ascending order according to the average activation by clean samples. Then, the defender sequentially prunes neurons until the accuracy of the validation dataset drops below a predetermined threshold. As shown in Fig. 4, the ASR of BadNets, TrojanNN, and HB drops significantly after model pruning, which means that model pruning can successfully defend against these baselines. In contrast, ATTEQ-NN and RobNet can still achieve high ASR after pruning. Given a threshold of 80% for CDA, ATTEQ-NN can preserve an ASR of more than 82% for all datasets. The reason why ATTEQ-NN is resistant to model pruning is the same as RobNet. ATTEQ-NN selects the neuron with a high activation to benign samples, thus the selected neuron is more likely to be preserved during model pruning. In terms of BadNets, TrojanNN, and HB, we can observe that the CDA and ASR first drop slowly and then quickly with the pruning rate in most cases. The possible reason is that the neurons inactive to both benign and malicious samples are pruned first, thus having no impact on either ASR or CDA. Then the neurons

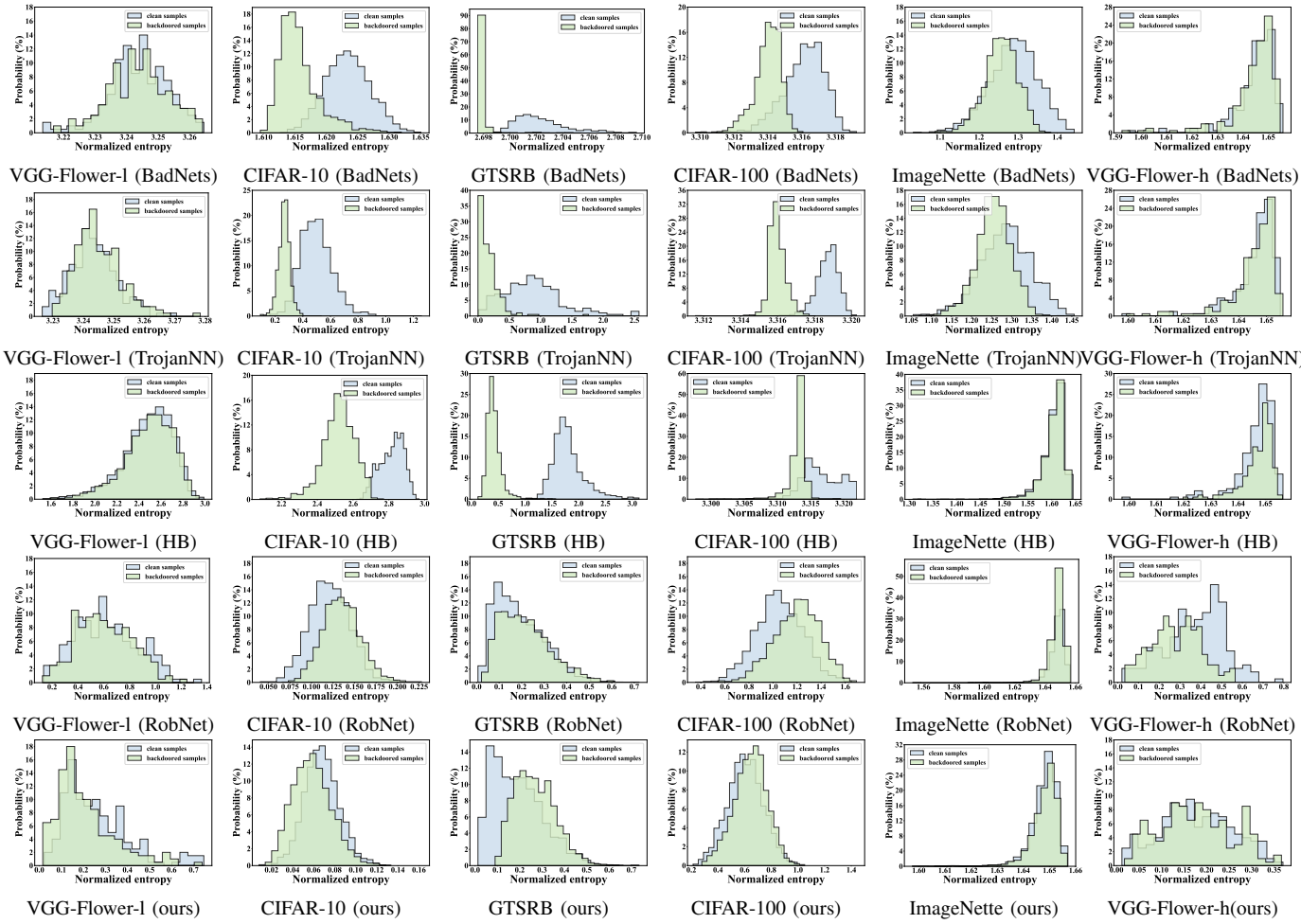
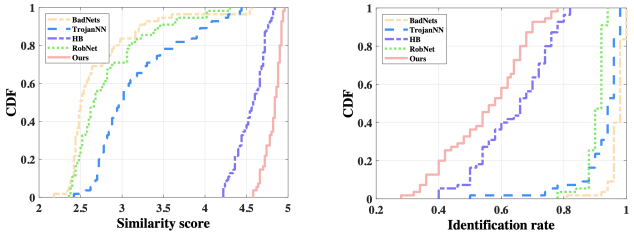


Fig. 5. The distribution of the entropy prediction results of clean samples and backdoored samples after applying STRIP to baseline attacks and ATTEQ-NN.



(a) Similarity test (b) Inspection test

Fig. 6. User study results.

activated mostly by backdoored samples are pruned, leading to downward ASR but relatively steady CDA. Finally, the neurons activated by benign samples are pruned, which reduces the CDA.

2) NAD: Neural Attention Distillation (NAD) [36] aims to erase the backdoor from the model. In NAD, the defender first fine-tunes the backdoored model on a small set of benign samples and uses the fine-tuned model as a teacher model. Then, NAD uses the teacher model to distill the backdoored model (student model) through an attention distillation. In this way, the neurons of the backdoor will be aligned with benign neurons associated with meaningful representations. As shown in Table VI, after applying NAD, the ASR of baselines decreases significantly, while the ASR of ATTEQ-NN only slightly decreases. The possible reason is that the gap between the backdoored model of ATTEQ-NN and the benign model has been narrowed through alternating retraining.

3) STRIP: STRIP [21] is a widely-used data-based defense strategy, which detects whether an input sample is backdoored or not. STRIP is an online method that examines arriving samples after the backdoored model has been deployed. In STRIP, the defender has no knowledge about the backdoored model (black-box) and can only observe the model outputs. The defender duplicates an input sample for many times and merges each copy with a different sample to generate a set of perturbed samples. The distribution of the prediction results of the perturb samples is used to detect backdoored samples. It is assumed that the prediction results of the disturbed samples have a high entropy if the sample is clean and a low entropy if the sample contains the trigger as the trigger strongly drives the prediction results towards the target label.

As shown in Fig. 5, for ATTEQ-NN and RobNet, the prediction results of the backdoored samples have a similar entropy distribution to benign samples for all datasets, making it difficult to differentiate the backdoored samples and the benign samples. For BadNets, TrojanNN, and HB, the entropy distributions of clean and backdoored samples are well separated for CIFAR-10, GTSRB, and CIFAR-100 but have a large overlap for

VGG-Flower-l, ImageNette, and VGG-Flower-h. The possible reason that these baselines are not easily detected by STRIP on these datasets is that their ASR is quite low for VGG-Flower-l, ImageNette, and VGG-Flower-h, as shown in Table II.

We attribute the evasiveness of ATTEQ-NN against STRIP to the attention-based mask determination strategy and the transparency value adjustment. The attention-based mask determination strategy yields a non-continuous trigger area, and the transparency value adjustment further conceals the trigger. Both approaches make the backdoored samples more stealthy and behave more like benign samples under the inspection of STRIP.

4) NC: NeuralCleanse (NC) [71] is a model-based defense strategy. NC tries to recover the trigger by computing the perturbation needed for a sample of the source label to be misclassified into the target label. The target label that requires a much smaller perturbation is deemed as the actual target label, and the perturbation is considered as the trigger. NC leverages MAD (Median Absolute Deviation) for anomaly detection with a threshold of 2. Experiments show that the MAD of our target class is always below this threshold (0.5415 for VGG-Flower-l, 0.0920 for CIFAR-10, 0.5040 for GTSRB, 1.0672 for CIFAR-100, 0.8313 for ImageNette, and 1.7584 for VGG-Flower-h). The success of ATTEQ-NN may be due to transparency adjustment, as a low-magnitude trigger is harder to recover. For the baseline models, the abnormal indexes of BadNets, TrojanNN, and HB are all greater than 2.3, except for the scenarios where the attack fails. RobNet can also escape the defense of NC due to its multi-trigger setting. We also find that the triggers reversed by NC on high-resolution data samples are more dispersed and more difficult to identify.

5) MNTD: MNTD [80] is a model-based defense based on a binary meta-classifier. To train the meta-model, the defender builds a large number of benign and backdoored shadow models as training samples. Since the defender has no knowledge of the specific backdoor attack methods, MNTD adopts *jumbo learning* to generate a variety of backdoored models. In this way, MNTD is generic and can detect most state-of-the-art backdoor attacks. To apply MNTD to baselines and ATTEQ-NN, for each dataset, we generate 2,048 benign models and 2,048 backdoored models to train a well-performed meta-classifier. The meta-classifier achieves a *modification attack accuracy* of 84.80% for VGG-Flower-l, 89.21% for CIFAR-10, 91.02% for GTSRB, 90.62% for CIFAR-100, 91.02% for ImageNette, and 91.41% for VGG-Flower-h. The *blending attack accuracy* of the meta-classifier is 83.20% for VGG-Flower-l, 85.57% for CIFAR-10, 86.72% for GTSRB, 93.75% for CIFAR-100, 86.72% for ImageNette, and 90.23% for VGG-Flower-h. *Modification attack accuracy* measures the effectiveness in detecting modification attacks [80]. *Blending attack accuracy* measures the effectiveness in detecting blending attacks [80]. When we feed the backdoored models of ATTEQ-NN to the meta-classifier, it is shown that ATTEQ-NN can evade the inspection of MNTD in some cases. In comparison, when we feed the backdoored models of the baselines to the meta-classifier, they are all detected by MNTD. The success in evading the detection of MNTD is possibly due to the alternating retraining strategy of ATTEQ-NN that makes the backdoored models behave like the benign ones. To verify this hypothesis, we construct backdoored models without using the alternating retraining method and feed the models into the meta-classifier of MNTD. The results show that MNTD can successfully detect these backdoored models.

H. User Study

We have conducted three sets of user studies to evaluate the concealment of the backdoored samples of ATTEQ-NN and all baselines. We have recruited 50 volunteers aged 20$\sim$30 who are college students and faculty members. Before the tests, we explained the basics of ATTEQ-NN and the baselines to the volunteers and sought for their confirmation of understanding the attacks. We randomly select 55 benign images from VGG-Flower-l, CIFAR-10, GTSRB, and CIFAR-100 and generate their backdoored versions based on ATTEQ-NN and all baselines for test.

1) Similarity Test: In the first set of tests, we place benign samples and the corresponding backdoored samples side-by-side (the benign samples on the left and the backdoored samples on the right) for volunteers to judge how similar the two samples are. The similarity score ranges from 1 to 5, where 5 represents "look exactly the same", 4 represents "very similar", 3 represents "a little similar", 2 represents "not very similar", and 1 represents "very different". We have collected a total of $50 * 55 * 4 = 11,000$ answers. The cumulative distribution function (CDF) of the similarity score is shown in Fig. 6(a). We can observe that more than 50% of the samples of ATTEQ-NN have a similarity score of more than 4.84, while BadNets, RobNet, and TrojanNN have a median similarity score of less than 3.5. HB has a slightly lower similarity score than ATTEQ-NN, and the ASR of HB is much lower than ATTEQ-NN.

2) Inspection Test: In the second set of tests, we place benign samples and the corresponding backdoored samples side-by-side for volunteers to choose which one is the backdoored sample. We shuffle each pair of samples such that the backdoored sample is not necessarily on the right of the benign sample. We calculate the percentage of correct answers as the *identification rate*. The CDF of the identification rate is shown in Fig. 6(b). We can observe that more than 50% of the samples of ATTEQ-NN have an identification rate of less than 0.58 (approximately random guess), while BadNets, TrojanNN, and RobNet have a median identification rate of more than 90%. HB has a slightly higher identification rate than ATTEQ-NN since some backdoored samples of HB have obvious degradation in quality (blur).

3) Detection Test: In the third set of tests, we present random pairs of samples (two benign samples or one benign sample and one backdoored sample) for volunteers to judge whether each pair contains a backdoored sample or not. We calculate the percentage of correct answers as the *detection rate*. ATTEQ-NN achieves an average detection rate of 0.45 (approximately random guess), while the average detection rate of the baselines is 0.56 for HB, 0.9 for TrojanNN, 0.93 for RobNet, and 0.93 for BadNets.

VI. CONCLUSION AND FUTURE WORK

This paper presents an effective and evasive backdoor attack against deep neural networks. To intensify the attack performance, we propose a novel attention-based mask determination strategy to place the trigger at the most influential area of an

image. To achieve the evasiveness goal, we carefully adjust the trigger transparency and add a QoE term to the loss function. Extensive experiments verify the superiority of the proposed attack compared with baseline backdoor attacks. To improve the clean data accuracy, we propose an alternating retraining strategy, which is also shown to be effective in evading MNTD defense method. We show that our proposed attack can evade both human visual inspections and state-of-the-art defenses. It is shown that ATTEQ-NN is also robust to transfer learning scenarios.

There are various potential spaces worthy of exploring in the future. First, it is possible to generalize ATTEQ-NN beyond the vision domain as attention mechanisms have been designed for the voice [84], [46], text [17], and video [34] domains. The main challenge may be how to generate model-dependent triggers in different domains. Second, effective defense against ATTEQ-NN is necessary to reduce the potential risks of such attacks.

ACKNOWLEDGMENT

Qian Wang's work was partially supported by the NSFC under Grants U20B2049 and 61822207, and the Fundamental Research Funds for the Central Universities under Grant 2042021gf0006.

REFERENCES

[1] BigML. https://bigml.com.

[2] Caffe Model Zoo. https://github.com/BVLC/caffe/wiki/Model-Zoo.

[3] ModelDepot. https://modeldepot.io/.

[4] VGG-16. *https://neurohive.io/en/popular-networks/vgg16/*.

[5] Shun-ichi Amari. Backpropagation and stochastic gradient descent method. *Neurocomputing*, 5(4-5):185–196, 1993.

[6] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. How to backdoor federated learning. In *International Conference on Artificial Intelligence and Statistics*, pages 2938–2948. PMLR, 2020.

[7] Nicholas Carlini, Pratyush Mishra, Tavish Vaidya, Yuankai Zhang, Micah Sherr, Clay Shields, David Wagner, and Wenchao Zhou. Hidden voice commands. In *USENIX Security Symposium*, pages 513–530. USENIX Association, 2016.

[8] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

[9] Hanxiao Chen, Hongwei Li, Guishan Dong, Meng Hao, Guowen Xu, Xiaoming Huang, and Zhe Liu. Practical membership inference attack against collaborative inference in industrial IoT. *IEEE Transactions on Industrial Informatics*, 2020.

[10] Huili Chen, Cheng Fu, Jishen Zhao, and Farinaz Koushanfar. DeepInspect: A black-box trojan detection and mitigation framework for deep neural networks. In *International Joint Conference on Artificial Intelligence*, pages 4658–4664. ijcai.org, 2019.

[11] Xiaoyi Chen, Ahmed Salem, Michael Backes, Shiqing Ma, and Yang Zhang. Badnl: Backdoor attacks against nlp models. In *ICML 2021 Workshop on Adversarial Machine Learning*, 2021.

[12] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

[13] Yanjiao Chen, Xueluan Gong, Qian Wang, Xing Di, and Huayang Huang. Backdoor attacks and defenses for deep neural networks in outsourced cloud environments. *IEEE Network*, 34(5):141–147, 2020.

[14] Yanjiao Chen, Kaishun Wu, and Qian Zhang. From QoS to QoE: A tutorial on video quality assessment. *IEEE Communications Surveys & Tutorials*, 17(2):1126–1165, 2014.

[15] Edward Chou, Florian Tramèr, Giancarlo Pellegrino, and Dan Boneh. Sentinet: Detecting physical attacks against deep learning systems. *arXiv preprint arXiv:1812.00292*, 2018.

[16] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*. OpenReview.net, 2021.

[17] Jiachen Du, Lin Gui, Ruifeng Xu, and Yulan He. A convolutional attention model for text classification. In *National CCF Conference on Natural Language Processing and Chinese Computing*, pages 183–195. Springer, 2017.

[18] Predictive Analytics-Cloud Machine Learning Engine. Google cloud. *Google, [Online]. Available: https://cloud. google. com/appengine/.*

[19] fast.ai. Imagenette: A smaller subset of 10 easily classified classes from imagenet. *Available: https://github.com/fastai/imagenette*.

[20] Yansong Gao, Bao Gia Doan, Zhi Zhang, Siqi Ma, Jiliang Zhang, Anmin Fu, Surya Nepal, and Hyoungshick Kim. Backdoor attacks and countermeasures on deep learning: A comprehensive review. *arXiv preprint arXiv:2007.10760*, 2020.

[21] Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. STRIP: A defence against trojan attacks on deep neural networks. In *IEEE Annual Computer Security Applications Conference*, pages 113–125, 2019.

[22] Xueluan Gong, Yanjiao Chen, Qian Wang, Huayang Huang, Lingshuo Meng, Chao Shen, and Qian Zhang. Defense-resistant backdoor attacks against deep neural networks in outsourced cloud environment. *IEEE Journal on Selected Areas in Communications*, 39(8):2617–2631, 2021.

[23] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

[24] Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. *arXiv preprint arXiv:2106.04690*, 2021.

[25] Siteng Huang, Min Zhang, Yachen Kang, and Donglin Wang. Attributes-guided and pure-visual attention alignment for few-shot recognition. In *AAAI Conference on Artificial Intelligence*, pages 7840–7847. AAAI Press, 2021.

[26] Xijie Huang, Moustafa Alzantot, and Mani Srivastava. NeuronInspect: Detecting backdoors in neural networks via output explanations. *arXiv preprint arXiv:1911.07399*, 2019.

[27] Yujie Ji, Xinyang Zhang, Shouling Ji, Xiapu Luo, and Ting Wang. Model-reuse attacks on deep learning systems. In *SIGSAC Conference on Computer and Communications Security*, pages 349–363. ACM, 2018.

[28] Yujie Ji, Xinyang Zhang, and Ting Wang. Backdoor attacks against learning systems. In *Conference on Communications and Network Security*, pages 1–9. IEEE, 2017.

[29] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[30] Earl J Kirkland. Bilinear interpolation. In *Advanced Computing in Electron Microscopy*, pages 261–263. Springer, 2010.

[31] Yehao Kong and Jiliang Zhang. Adversarial audio: A new information hiding method and backdoor for dnn-based speech recognition models. *arXiv preprint arXiv:1904.03829*, 2019.

[32] Brett Koonce. Resnet 34. In *Convolutional Neural Networks with Swift for Tensorflow*, pages 51–61. Springer, 2021.

[33] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[34] Junnan Li, Yongkang Wong, Qi Zhao, and Mohan S Kankanhalli. Attention transfer from web images for video recognition. In *ACM International Conference on Multimedia*, pages 1–9, 2017.

[35] Shaofeng Li, Minhui Xue, Benjamin Zi Hao Zhao, Haojin Zhu, and Xinpeng Zhang. Invisible backdoor attacks on deep neural networks via steganography and regularization. *arXiv preprint arXiv:1909.02742*, 2019.

[36] Yige Li, Nodens Koren, Lingjuan Lyu, Xixiang Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers

from deep neural networks. In *International Conference on Learning Representations*. OpenReview.net, 2021.

[37] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. *arXiv preprint arXiv:2101.05930*, 2021.

[38] Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2020.

[39] Cong Liao, Haoti Zhong, Anna Squicciarini, Sencun Zhu, and David Miller. Backdoor embedding in convolutional neural network models via invisible perturbation. *arXiv preprint arXiv:1808.10307*, 2018.

[40] Junyu Lin, Lei Xu, Yingqi Liu, and Xiangyu Zhang. Composite backdoor attack for deep neural network by mixing existing benign features. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 113–131, 2020.

[41] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International Symposium on Research in Attacks, Intrusions, and Defenses*, pages 273–294. Springer, 2018.

[42] Yang Liu, Zhihao Yi, and Tianjian Chen. Backdoor attacks and defenses in feature-partitioned collaborative learning. *arXiv preprint arXiv:2007.03608*, 2020.

[43] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *International Conference on Learning Representations*. OpenReview.net, 2017.

[44] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. ABS: Scanning neural networks for backdoors by artificial brain stimulation. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1265–1282, 2019.

[45] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Annual Network and Distributed System Security Symposium*. The Internet Society, 2018.

[46] Yu Liu, Cong Zhang, Bo Hang, Song Wang, and Han-Chieh Chao. An audio attention computational model based on information entropy of two channels and exponential moving average. *Human-centric Computing and Information Sciences*, 9(1):1–16, 2019.

[47] Shaohao Lu, Yuqiao Xian, Ke Yan, Yi Hu, Xing Sun, Xiaowei Guo, Feiyue Huang, and Wei-Shi Zheng. Discriminator-free generative adversarial attack. In *ACM International Conference on Multimedia*, pages 1544–1552, 2021.

[48] Shiqing Ma, Yingqi Liu, Guanhong Tao, Wen-Chuan Lee, and Xiangyu Zhang. NIC: Detecting adversarial samples with neural network invariant checking. In *Annual Network and Distributed System Security Symposium*. The Internet Society, 2019.

[49] L. Melis, C. Song, E. De Cristofaro, and V. Shmatikov. Exploiting unintended feature leakage in collaborative learning. In *IEEE Symposium on Security and Privacy*, pages 691–706, 2019.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Annual Conference on Neural Information Processing Systems*, pages 3111–3119, 2013.

[51] Stefan Milz, Georg Arbeiter, Christian Witt, Bassam Abdallah, and Senthil Yogamani. Visual slam for automated driving: Exploring the applications of deep learning. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 247–257, 2018.

[52] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. Recurrent models of visual attention. *arXiv preprint arXiv:1406.6247*, 2014.

[53] Milad Nasr, Reza Shokri, and Amir Houmansadr. Machine learning with membership privacy using adversarial regularization. In *ACM SIGSAC Conference on Computer and Communications Security*, page 634–646, 2018.

[54] Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Annual Conference on Neural Information Processing Systems*, 2020.

[55] Maria-Elena Nilsback and Andrew Zisserman. 102 category flower dataset. *http://www.robots.ox.ac.uk/ vgg/data/flowers/102*, 2008.

[56] Ren Pang, Hua Shen, Xinyang Zhang, Shouling Ji, Yevgeniy Vorobeychik, Xiapu Luo, Alex Liu, and Ting Wang. A tale of evil twins: Adversarial inputs versus poisoned models. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 85–99, 2020.

[57] Ren Pang, Zheng Zhang, Xiangshan Gao, Zhaohan Xi, Shouling Ji, Peng Cheng, and Ting Wang. Trojanzoo: Everything you ever wanted to know about neural backdoors (but were afraid to ask). *arXiv preprint arXiv:2012.09302*, 2020.

[58] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[59] Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. Hidden trigger backdoor attacks. In *AAAI Conference on Artificial Intelligence*, pages 11957–11965. AAAI Press, 2020.

[60] Ahmed Salem, Rui Wen, Michael Backes, Shiqing Ma, and Yang Zhang. Dynamic backdoor attacks against machine learning models. *arXiv preprint arXiv:2003.03675*, 2020.

[61] Ahmed Salem, Yang Zhang, Mathias Humbert, Pascal Berrang, Mario Fritz, and Michael Backes. Ml-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In *Annual Network and Distributed System Security Symposium*. The Internet Society, 2019.

[62] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[63] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1528–1540, 2016.

[64] R. Shokri, M. Stronati, C. Song, and V. Shmatikov. Membership inference attacks against machine learning models. In *IEEE Symposium on Security and Privacy*, pages 3–18, 2017.

[65] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. Man vs. Computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, 32:323–332, 2012.

[66] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.

[67] Di Tang, XiaoFeng Wang, Haixu Tang, and Kehuan Zhang. Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection. In *USENIX Security Symposium*, 2021.

[68] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems*, pages 8000–8010, 2018.

[69] Sakshi Udeshi, Shanshan Peng, Gerald Woo, Lionell Loh, Louth Rawshan, and Sudipta Chattopadhyay. Model agnostic defence against backdoor attacks in machine learning. *arXiv preprint arXiv:1908.02203*, 2019.

[70] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

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

[72] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3156–3164, 2017.

[73] Hongyi Wang, Kartik Sreenivasan, Shashank Rajput, Harit Vishwakarma, Saurabh Agarwal, Jy-yong Sohn, Kangwook Lee, and Dimitris S. Papailiopoulos. Attack of the tails: Yes, you really can backdoor federated learning. In *Annual Conference on Neural Information Processing Systems*, 2020.

[74] Shuo Wang, Surya Nepal, Carsten Rudolph, Marthie Grobler, Shangyu Chen, and Tianle Chen. Backdoor attacks against transfer learning with pre-trained deep learning models. *IEEE Transactions on Services Computing*, 2020.

[75] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004.

[76] Emily Wenger, Josephine Passananti, Yuanshun Yao, Haitao Zheng, and Ben Y Zhao. Backdoor attacks on facial recognition in the physical world. *arXiv preprint arXiv:2006.14580*, 2020.

[77] Chong Xiang, Arjun Nitin Bhagoji, Vikash Sehwag, and Prateek Mittal. Patchguard: A provably robust defense against adversarial patches via small receptive fields and masking. In *30th USENIX Security Symposium*, 2021.

[78] Chulin Xie, Keli Huang, Pin-Yu Chen, and Bo Li. DBA: Distributed backdoor attacks against federated learning. In *International Conference on Learning Representations*, 2019.

[79] Weilin Xu, Yanjun Qi, and David Evans. Automatically evading classifiers. In *Annual Network and Distributed System Security Symposium*. The Internet Society, 2016.

[80] Xiaojun Xu, Qi Wang, Huichen Li, Nikita Borisov, Carl A Gunter, and Bo Li. Detecting ai trojans using meta neural analysis. In *IEEE Symposium on Security and Privacy*, 2021.

[81] Zhaoyuan Yang, Nurali Virani, and Naresh S Iyer. Countermeasure against backdoor attacks using epistemic classifiers. In *Artificial Intelligence and Machine Learning for Multi-domain Operations Applications II*, volume 11413, page 114130P. International Society for Optics and Photonics, 2020.

[82] Yuanshun Yao, Huiying Li, Haitao Zheng, and Ben Y Zhao. Latent backdoor attacks on deep neural networks. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 2041–2055, 2019.

[83] Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: Analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium*, pages 268–282, 2018.

[84] Zhichao Zhang, Shugong Xu, Shunqing Zhang, Tianhao Qiao, and Shan Cao. Attention based convolutional recurrent neural network for environmental sound classification. *Neurocomputing*, 453:896–903, 2021.

[85] Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. Clean-label backdoor attacks on video recognition models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14443–14452, 2020.

[86] Martin Zinkevich, Markus Weimer, Alexander J Smola, and Lihong Li. Parallelized stochastic gradient descent. In *Annual Conference on Neural Information Processing Systems*, pages 2595–2603, 2010.

APPENDIX

A. DATASETS AND MODELS

VGG-Flower. VGG-Flower [55] includes 6,146 images belonging to 102 flower categories. Each class contains 40∼258 image samples. We randomly select 10 classes with 1,673 training images and 200 test images. For VGG-Flower-l, the selected images are uniformly resized to 32 × 32. We train a VGG-16 [4] model with 3 fully-connected layers and a RAN-92 model for 600 epochs. We set the learning rate as 0.001, the batch size as 512, the momentum of stochastic gradient descent as 0.9, and weight decay as 0.0005. The trained benign model has a prediction accuracy of 98.5% on the test set. The trained RAN model has a prediction accuracy of 94.5% on the test set. For VGG-Flower-h, the selected images are uniformly resized to 224 × 224. We train a ResNet-18 network and a RAN-92 model for 150 epochs. We set the learning rate as 0.0001, the batch size as 32, the momentum of stochastic gradient descent as 0.95, and weight decay as 0.0005. The trained benign model has a prediction accuracy of 97.5% on the test set. The trained RAN model has a prediction accuracy of 98.5% on the test set.

CIFAR-10. CIFAR-10 [33] contains 60,000 images belonging to 10 classes. Each class includes 6,000 images, and each sample has a dimension of 32 × 32. We randomly select 50,000 samples as the training set, and the remaining 10,000 samples as the test set. We train a VGG-16 [4] model with one fully-connected layer and a RAN-92 model on the training set for 300 epochs. The learning rate is 0.001. The batch size is 512. The momentum of stochastic gradient descent is 0.9. A StepLR scheduler is used in the training process, with a step size of 100 and $\gamma = 0.33$. The trained benign model has a prediction accuracy of 91.94% on the test set. The trained RAN model has a prediction accuracy of 95.4% on the test set.

GTSRB. GTSRB [65] contains images of German traffic signs that belong to 43 classes. The dataset is divided into 39,209 training samples and 12,630 testing samples. Using annotated information, we cropped each image to its core area and resized each image to 32 × 32. We train a traffic sign classifier and a RAN-92 model using the ResNet-34 network [32] on the training set. With a base learning rate at 0.001, a batch size at 512, a momentum of 0.9, and a decay factor of 0.2 per 60 epochs, we train the model for 300 epochs. The trained benign model has a prediction accuracy of 97.25% on the test set. The trained RAN model has a prediction accuracy of 94.61% on the test set.

CIFAR-100. CIFAR-100 [33] includes 600,000 images that belong to 100 classes. Each class has 500 training samples and 100 testing images, and each sample has a dimension of 32 × 32. We train a ResNet-50 model and a RAN-92 model on the training set for 200 epochs. We set the learning rate as 0.1, the batch size as 512, and the momentum of stochastic gradient descent as 0.9. To further improve its performance, a MultiStepLR scheduler with a γ of 0.2 is used at the 60-th, the 120-th, and the 160-th epochs. The trained benign model has a prediction accuracy of 79.09% on the test set. The trained RAN model has a prediction accuracy of 78.68% on the test set.

ImageNette. ImageNette [19] is a subset of ImageNet, widely used in the research community [77], [47]. ImageNette includes 9,469 training samples and 3,925 test samples. Each image has a high resolution with a dimension of 224 × 224. We train a ResNet-18 network and a RAN-92 model for 150 epochs. We set the learning rate as 0.001, the batch size as 16 (due to the limited GPU source), the momentum of stochastic gradient descent as 0.95, and weight decay as 0.0005. The trained benign model has a prediction accuracy of 92.43% on the test set. The trained RAN model has a prediction accuracy of 90.62% on the test set.

B. EVALUATION METRICS

ASR. ASR measures the effectiveness of the backdoor attacks, computed as the probability that a trigger-imposed sample is misclassified to the target label.

$$ASR(F_A, \mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{I}_{[F_A(x+\delta)=y_t]}. \tag{7}$$

CDA. CDA measures whether the backdoored model can maintain prediction accuracy of clean input samples.

$$CDA(F_A, \mathcal{X}) = \frac{1}{|\mathcal{X}|} \sum_{x \in \mathcal{X}} \mathbf{I}_{[F_A(x)=y]}, \tag{8}$$

where y is the ground-truth label of x.

TABLE VIII. THE IMPACT OF TRIGGER SIZE ON THE PERFORMANCE OF ATTEQ-NN.

Trigger size	ImageNette		
	ASR	CDA	SSIM
4×4	88.91%	90.88%	0.9992
8×8	92.16%	92.40%	0.9979
12×12	93.00%	89.60%	0.9953
16×16	94.19%	89.89%	0.9927

TABLE IX. THE IMPACT OF TRIGGER SIZE ON THE PERFORMANCE OF ATTEQ-NN.

Trigger size	VGG-Flower-h		
	ASR	CDA	SSIM
8×8	83.00%	96.00%	0.9973
12×12	85.50%	97.00%	0.9946
16×16	88.00%	94.50%	0.9912
20×20	91.50%	96.00%	0.9862

C. STATE-OF-THE-ART BASELINES

BadNets. BadNets [23] is one of the most widely used backdoor attacks. BadNets uses a visible random trigger (a white square) at the bottom right corner of the image.

TrojanNN. TrojanNN [45] is the first backdoor attack that uses model-dependent triggers, but the trigger is visible. It is assumed that the attacker cannot access the training datasets and use reverse engineering to recover the training samples before the attacks. For a fair comparison, we directly use the original training dataset in TrojanNN.

Hidden Backdoor. Hidden Backdoor (HB) [59] is a state-of-the-art backdoor attack based on invisible random triggers. Note that HB considers a data vendor attacker. We adapt the HB to the model vendor scenario for a fair comparison.

RobNet. RobNet [22] is a recent backdoor attack based on model-dependent triggers, but the trigger is visible. Unlike TrojanNN [45] that selects the neuron based on weights, RobNet selects the neuron according to both activation and weights.

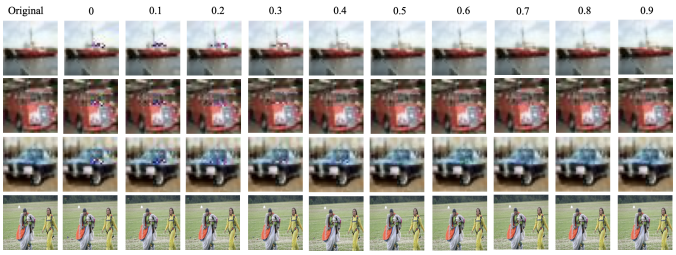
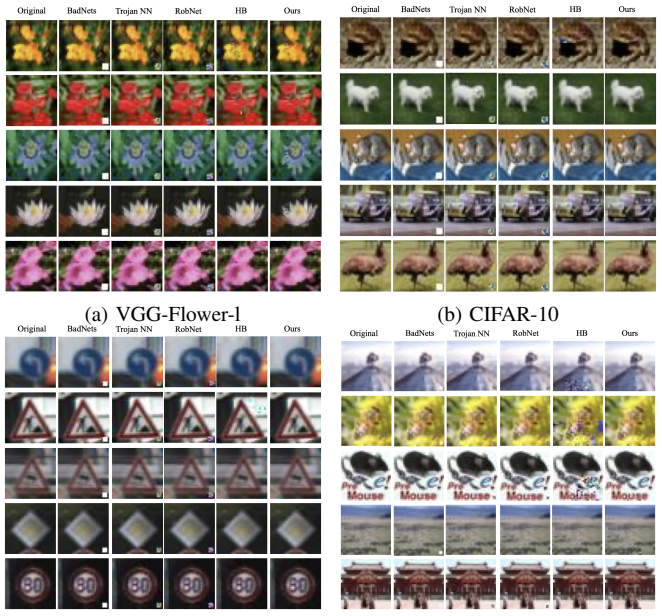


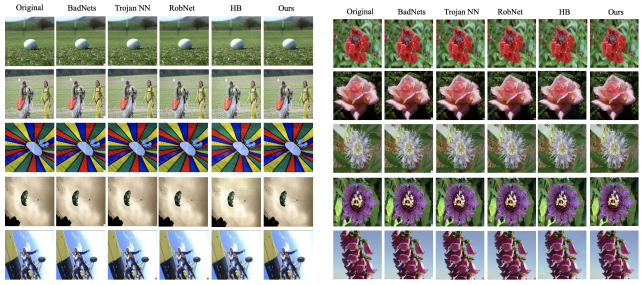
Fig. 7. The backdoored samples of ATTEQ-NN under different transparency values. The 1st column is the original benign samples. A lower transparency value indicates a stronger imposition of the trigger. A transparency value of zero means the imposed trigger completely blocks the original pixel.



(a) VGG-Flower-l (b) CIFAR-10

(c) GTSRB (d) CIFAR-100

Fig. 8. Comparison of backdoored samples of BadNets [23], TrojanNN [45], HB [59], RobNet [22] and ATTEQ-NN.



(e) ImageNette (f) VGG-Flower-h

Fig. 9. Comparison of backdoored samples of BadNets [23], TrojanNN [45], HB [59], RobNet [22] and ATTEQ-NN.

TABLE X. THE IMPACT OF TRIGGER SIZE ON THE PERFORMANCE OF ATTEQ-NN.

Trigger size	VGG-Flower-l			CIFAR-10			GTSRB			CIFAR-100		
	ASR	CDA	SSIM	ASR	CDA	SSIM	ASR	CDA	SSIM	ASR	CDA	SSIM
1×1	19.50%	95.00%	0.9943	15.47%	85.61%	0.9959	91.14%	95.16%	0.9909	93.50%	74.00%	0.9988
2×2	51.00%	95.50%	0.9821	95.56%	89.04%	0.9874	96.92%	96.80%	0.9740	99.94%	74.91%	0.9943
3×3	94.50%	95.50%	0.9540	95.62%	88.90%	0.9758	99.97%	97.19%	0.9616	99.17%	74.50%	0.9848
4×4	99.00%	97.50%	0.9329	97.29%	88.90%	0.9690	99.70%	96.79%	0.9414	99.79%	73.85%	0.9612

TABLE XI. THE IMPACT OF TRANSPARENCY VALUE ON THE PERFORMANCE OF ATTEQ-NN.

#value	VGG-Flower-l			CIFAR-10			GTSRB		
	ASR	CDA	SSIM	ASR	CDA	SSIM	ASR	CDA	SSIM
0	100.0%	98.00%	0.8974	97.55%	89.91%	0.9438	99.98%	97.00%	0.9470
0.1	100.0%	97.50%	0.9042	97.14%	88.68%	0.9487	99.91%	96.34%	0.9506
0.2	99.00%	96.50%	0.9108	97.81%	89.22%	0.9526	99.61%	97.26%	0.9560
0.3	100.0%	98.00%	0.9276	97.52%	89.92%	0.9600	99.11%	97.16%	0.9615
0.4	99.00%	97.50%	0.9329	97.29%	88.90%	0.9690	99.97%	97.19%	0.9616
0.5	97.50%	95.00%	0.9552	88.86%	87.56%	0.9770	97.49%	96.43%	0.9713
0.6	93.50%	96.50%	0.9557	79.28%	87.26%	0.9812	96.29%	97.01%	0.9766
0.7	67.50%	96.00%	0.9708	78.95%	87.06%	0.9886	85.54%	97.06%	0.9821
0.8	25.00%	97.50%	0.9890	18.54%	83.58%	0.9934	75.73%	96.67%	0.9906
0.9	15.50%	95.50%	0.9971	14.54%	85.07%	0.9980	43.52%	96.76%	0.9964
#value	CIFAR-100			ImageNette			VGG-Flower-h		
	ASR	CDA	SSIM	ASR	CDA	SSIM	ASR	CDA	SSIM
0	99.71%	75.07%	0.9924	98.19%	90.93%	0.9944	98.50%	97.00%	0.9952
0.1	99.62%	74.98%	0.9925	97.78%	91.41%	0.9946	97.00%	96.00%	0.9956
0.2	99.70%	75.14%	0.9934	97.99%	91.01%	0.9950	96.50%	96.50%	0.9953
0.3	99.52%	74.38%	0.9952	97.94%	91.90%	0.9953	96.00%	95.50%	0.9954
0.4	99.94%	74.91%	0.9943	97.45%	91.72%	0.9958	91.00%	96.50%	0.9957
0.5	99.00%	74.51%	0.9971	97.36%	91.64%	0.9961	88.00%	97.00%	0.9961
0.6	97.24%	73.96%	0.9978	95.03%	90.80%	0.9967	86.00%	95.00%	0.9966
0.7	94.32%	74.64%	0.9984	92.16%	92.40%	0.9979	83.00%	96.00%	0.9973
0.8	84.60%	73.40%	0.9992	84.94%	92.31%	0.9986	77.00%	97.00%	0.9975
0.9	1.24%	73.98%	0.9996	72.48%	90.88%	0.9995	22.00%	97.00%	0.9987

TABLE XII. THE IMPACT OF FINE-TUNING ON BASELINES AND ATTEQ-NN ON CIFAR-100, IMAGENETTE, AND VGG-FLOWER-H. NOTE THAT THE POISON RATE IS 30% FOR IMAGENETTE AND VGG-FLOWER-H.

CIFAR-100

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	64.33%	75.37%	83.62%	74.10%	27.91%	69.27%	87.05%	71.37%	99.60%	73.91%
2	64.43%	75.26%	83.32%	74.03%	26.48%	69.21%	90.36%	71.90%	99.71%	73.79%
10	57.08%	75.74%	82.49%	74.00%	27.04%	69.33%	90.37%	70.43%	99.71%	73.97%
16	73.13%	75.65%	83.99%	74.02%	27.09%	70.15%	89.68%	71.40%	99.80%	73.86%
22	72.04%	75.57%	94.99%	74.09%	26.47%	69.30%	91.84%	71.85%	99.86%	74.14%
30	73.75%	75.40%	84.44%	74.09%	28.31%	70.06%	94.64%	71.84%	99.72%	73.66%
36	76.03%	75.52%	85.72%	73.93%	28.19%	69.88%	98.18%	72.06%	99.90%	74.30%
42	69.02%	75.26%	86.72%	74.03%	27.96%	69.53%	97.24%	72.29%	99.78%	74.01%
48	67.80%	75.18%	86.77%	74.07%	29.34%	69.34%	98.44%	72.54%	99.88%	74.02%
56	71.71%	75.23%	88.72%	74.11%	30.04%	69.77%	94.84%	71.73%	99.95%	74.44%
62	71.35%	75.16%	88.59%	74.07%	30.82%	70.29%	99.21%	71.89%	99.88%	74.08%
68	77.96%	75.06%	88.68%	74.02%	30.67%	70.41%	97.25%	72.49%	99.87%	74.80%
74	65.41%	75.11%	88.92%	73.99%	34.79%	69.83%	99.40%	72.21%	99.94%	74.44%
80	79.33%	74.93%	89.67%	73.96%	35.90%	69.51%	98.47%	72.07%	99.83%	74.76%
86	80.15%	74.74%	92.45%	74.09%	37.27%	68.93%	99.25%	72.86%	99.89%	74.55%
94	82.43%	74.47%	94.03%	74.01%	45.26%	69.30%	98.37%	72.78%	99.83%	75.35%
100	86.61%	74.04%	95.03%	73.90%	55.87%	69.84%	99.44%	73.26%	99.95%	75.25%
106	89.86%	73.55%	96.28%	73.95%	70.94%	70.04%	99.54%	73.45%	99.88%	75.33%
107 (all)	88.09%	72.88%	95.14%	73.71%	74.21%	69.87%	99.35%	72.51%	99.70%	75.14%

ImageNette

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	24.68%	76.84%	25.80%	76.66%	7.34%	90.07%	33.32%	83.92%	97.02%	92.76%
2	28.00%	75.97%	27.54%	75.57%	9.46%	88.65%	33.57%	83.57%	97.02%	92.41%
6	28.53%	76.00%	27.82%	76.10%	10.73%	88.15%	69.38%	84.43%	96.20%	92.61%
10	28.15%	76.07%	28.50%	76.05%	11.02%	88.41%	70.34%	82.19%	96.03%	92.38%
16	28.43%	75.94%	28.91%	75.89%	12.62%	87.58%	71.69%	83.74%	94.06%	91.33%
20	28.94%	76.07%	29.17%	75.69%	13.01%	87.69%	75.73%	84.38%	96.21%	91.18%
26	29.19%	75.97%	29.42%	75.74%	13.27%	87.17%	76.28%	85.20%	92.94%	91.92%
30	30.36%	74.85%	32.28%	73.19%	15.28%	86.49%	84.20%	84.53%	94.54%	91.64%
36	31.85%	73.32%	33.60%	71.82%	16.99%	86.91%	86.98%	85.17%	96.74%	91.90%
40	33.61%	72.62%	35.89%	71.38%	18.06%	85.38%	90.42%	84.68%	96.33%	91.41%
41	35.13%	71.28%	37.83%	70.54%	18.81%	85.32%	92.92%	84.87%	97.58%	91.46%

VGG-Flower-h

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	21.00%	98.00%	21.50%	97.50%	5.50%	96.50%	9.50%	92.00%	99.50%	97.00%
2	22.00%	97.50%	23.50%	98.00%	5.50%	97.00%	12.00%	89.50%	99.00%	97.50%
6	23.00%	97.50%	25.50%	97.00%	7.50%	96.00%	10.50%	90.50%	100.0%	98.00%
10	26.50%	97.50%	26.00%	97.00%	7.50%	96.00%	11.50%	89.50%	99.50%	97.00%
16	27.50%	97.00%	27.00%	97.50%	10.50%	95.50%	18.00%	90.50%	98.00%	97.00%
20	28.00%	97.00%	27.00%	97.50%	10.50%	96.00%	16.00%	92.00%	99.50%	97.50%
26	28.50%	97.00%	27.50%	97.50%	13.00%	95.50%	16.50%	91.50%	98.50%	97.50%
30	29.00%	97.50%	28.00%	97.00%	13.00%	94.50%	15.00%	90.00%	99.50%	97.50%
36	29.00%	97.50%	28.50%	95.50%	15.50%	95.00%	23.50%	92.00%	99.00%	98.00%
40	29.50%	96.50%	29.00%	95.00%	18.50%	94.00%	41.00%	94.50%	99.00%	98.00%
41	30.00%	95.50%	29.00%	95.50%	21.50%	93.00%	76.50%	95.50%	100.0%	97.00%

TABLE XIII. THE IMPACT OF FINE-TUNING ON BASELINES AND ATTEQ-NN ON VGG-FLOWER-L, CIFAR-10, AND GTSRB. WHEN THE NUMBER OF THE FROZEN LAYERS IS 0, WE FINE-TUNE THE WEIGHTS OF THE ENTIRE BACKDOORED MODEL.

VGG-Flower-l

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	10.50%	96.50%	11.00%	97.00%	21.00%	94.00%	66.50%	94.50%	91.50%	97.00%
2	10.00%	97.00%	10.50%	96.50%	20.50%	94.50%	74.50%	95.50%	89.00%	97.00%
4	10.50%	96.50%	11.50%	97.00%	20.00%	94.50%	76.00%	95.00%	93.50%	97.50%
6	11.50%	96.00%	12.00%	96.50%	21.00%	95.00%	75.50%	96.00%	91.50%	96.50%
8	11.00%	97.00%	11.50%	97.50%	20.50%	95.00%	79.50%	95.50%	91.00%	97.00%
10	11.50%	97.00%	12.00%	96.50%	21.00%	95.00%	86.00%	96.00%	92.00%	97.50%
12	11.00%	97.50%	12.00%	97.00%	21.00%	94.50%	85.00%	94.00%	90.00%	97.50%
14	12.50%	96.50%	13.00%	97.00%	21.00%	95.00%	87.00%	94.00%	89.00%	97.50%
16	13.50%	96.50%	14.50%	97.00%	21.00%	95.00%	83.50%	95.50%	92.50%	97.00%
18	13.00%	97.00%	14.00%	96.00%	22.00%	94.00%	91.00%	95.50%	90.50%	97.50%
20	14.50%	96.50%	14.50%	97.00%	22.50%	95.00%	89.00%	96.00%	90.50%	97.50%
22	14.00%	97.00%	14.50%	96.50%	24.00%	94.50%	90.50%	95.50%	90.50%	97.50%
24	15.50%	97.00%	13.50%	96.50%	21.00%	95.00%	84.50%	95.00%	91.50%	97.50%
26	15.00%	97.00%	14.00%	97.00%	21.00%	94.50%	83.00%	92.00%	92.00%	98.00%
27	17.00%	96.50%	19.00%	95.50%	26.50%	94.50%	92.00%	95.00%	91.00%	97.50%
28	17.50%	96.50%	20.50%	95.50%	29.00%	95.50%	93.00%	94.00%	93.00%	97.00%
29 (all)	22.50%	96.50%	23.00%	96.50%	33.00%	95.00%	94.50%	98.00%	94.50%	97.00%

CIFAR-10

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	37.24%	88.95%	26.04%	89.22%	2.88%	85.73%	25.43%	91.09%	99.82%	89.96%
2	36.58%	89.06%	31.24%	89.57%	3.03%	86.09%	25.43%	90.24%	99.82%	90.11%
4	39.51%	88.77%	35.79%	90.00%	3.05%	85.47%	27.38%	91.21%	98.04%	89.82%
6	39.81%	88.89%	40.62%	89.98%	2.98%	85.18%	27.94%	90.15%	99.15%	89.91%
8	35.28%	89.04%	48.51%	89.62%	3.27%	86.41%	25.95%	90.46%	99.83%	89.84%
10	35.75%	88.60%	58.39%	89.94%	3.21%	85.51%	27.10%	90.51%	99.76%	89.74%
12	36.23%	89.43%	67.79%	89.57%	3.97%	85.19%	27.37%	90.07%	99.86%	88.91%
14	38.23%	88.61%	74.13%	90.12%	3.77%	85.08%	27.02%	90.12%	99.94%	88.70%
16	33.06%	89.37%	80.12%	89.64%	2.93%	84.73%	50.44%	89.38%	99.94%	88.44%
18	33.99%	89.06%	85.37%	89.79%	3.34%	85.62%	54.32%	88.99%	99.95%	87.83%
20	37.29%	88.77%	90.31%	89.50%	3.77%	85.28%	67.83%	88.73%	99.94%	87.29%
22	41.74%	88.90%	92.42%	89.79%	5.39%	86.01%	77.05%	88.36%	99.14%	89.58%
24	59.23%	90.05%	95.71%	89.58%	13.78%	85.15%	79.67%	88.21%	99.93%	87.15%
26	75.07%	87.83%	97.76%	89.70%	19.33%	85.15%	90.00%	88.08%	99.94%	87.55%
27 (all)	89.80%	90.28%	97.90%	89.87%	29.80%	85.91%	88.39%	95.62%	99.93%	89.66%

GTSRB

Number	BadNets [23]		TrojanNN [45]		HB [59]		RobNet [22]		ATTEQ-NN	
	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA	ASR	CDA
0	2.42%	96.14%	2.61%	96.10%	5.97%	89.79%	5.17%	93.61%	92.03%	97.36%
2	2.95%	95.27%	3.31%	95.65%	6.03%	89.71%	5.19%	93.54%	93.65%	97.38%
6	3.39%	96.17%	3.42%	96.53%	6.01%	89.88%	5.08%	93.35%	94.39%	97.01%
10	3.78%	96.17%	3.93%	96.18%	5.86%	89.29%	5.29%	93.72%	94.35%	97.39%
14	5.62%	96.44%	5.54%	96.16%	6.25%	89.33%	5.24%	93.90%	95.16%	97.17%
18	5.72%	96.19%	6.13%	95.82%	6.32%	89.98%	5.24%	94.77%	93.58%	97.36%
20	10.38%	96.05%	5.00%	96.32%	6.09%	89.76%	5.21%	94.08%	94.85%	97.28%
24	10.96%	96.22%	14.66%	95.88%	7.84%	89.45%	5.28%	94.25%	93.43%	97.30%
28	11.88%	96.07%	17.36%	95.68%	7.59%	90.22%	5.36%	94.36%	96.05%	97.24%
32	12.24%	96.08%	22.34%	96.49%	9.07%	89.32%	5.23%	94.71%	95.22%	97.12%
36	17.37%	95.72%	21.10%	96.16%	13.39%	90.14%	59.62%	95.08%	97.04%	97.11%
38	30.44%	96.61%	32.45%	96.23%	25.33%	89.74%	34.96%	94.92%	95.84%	97.14%
42	37.40%	95.49%	45.10%	95.98%	40.52%	89.57%	53.47%	95.61%	94.49%	97.15%
46	42.39%	95.97%	45.02%	96.22%	40.70%	89.62%	60.34%	96.02%	97.19%	97.10%
50	43.52%	96.03%	49.52%	96.33%	39.23%	89.93%	57.68%	95.96%	97.43%	97.10%
54	44.47%	95.99%	52.55%	96.12%	41.72%	89.48%	53.46%	96.38%	97.85%	97.09%
58	61.27%	95.99%	65.99%	96.33%	51.28%	89.44%	70.26%	96.10%	99.33%	97.15%
62	61.10%	95.80%	65.99%	96.33%	50.08%	90.06%	71.03%	96.29%	99.65%	97.13%
64	75.93%	95.83%	77.34%	96.16%	57.41%	89.38%	80.41%	96.53%	99.90%	97.17%
68	78.80%	95.38%	84.54%	95.82%	58.85%	89.99%	78.59%	96.84%	99.75%	97.12%
72	95.98%	95.77%	95.19%	96.16%	64.77%	89.84%	91.09%	96.87%	99.56%	97.17%
73 (all)	98.08%	96.25%	98.10%	96.25%	75.93%	89.54%	93.67%	96.80%	99.51%	97.36%