IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2025, accepted 12 February 2025, date of publication 18 February 2025, date of current version 28 February 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3543333

RESEARCH ARTICLE

Detecting All-to-One Backdoor Attacks in Black-Box DNNs via Differential Robustness to Noise

HAO FU, PRASHANTH KRISHNAMURTHY, (Member, IEEE), SIDDHARTH GARG, AND FARSHAD KHORRAMI, (Fellow, IEEE)
Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201, USA

Corresponding author: Hao Fu (hf881@nyu.edu)

This work was supported in part by the Army Research Office under grant number W911NF-21-1-0155 and by the New York University Abu Dhabi (NYUAD) Center for Artificial Intelligence and Robotics (CAIR), funded by Tamkeen under the NYUAD Research Institute Award CG010.

ABSTRACT The all-to-one (A2O) backdoor attack is one of the major adversarial threats against neural networks. Most existing A2O backdoor defenses operate in a white-box context, necessitating access to the backdoored model's architecture, hidden layer outputs, or internal parameters. The necessity for black-box A2O backdoor defenses arises, particularly in scenarios where only the network's input and output are accessible. However, prevalent black-box A2O backdoor defenses often mandate assumptions regarding the locations of triggers, as they leverage hand-crafted features for detection. In instances where triggers deviate from these assumptions, the resultant hand-crafted features diminish in quality, rendering these methods ineffective. To address this issue, this work proposes a post-training black-box A2O backdoor defense that maintains consistent efficacy regardless of the triggers' locations. Our method hinges on the empirical observation that, in the context of A2O backdoor attacks, poisoned samples are more resilient to uniform noise than clean samples in terms of the network output. Specifically, our approach uses a metric to quantify the resiliency of the given input to the uniform noise. A novelty detector, trained utilizing the quantified resiliency of available clean samples, is deployed to discern whether the given input is poisoned. The novelty detector is evaluated across various triggers. Our approach is effective on all utilized triggers. Lastly, an explanation is provided for our observation.

INDEX TERMS Neural network backdoors, novelty detection, output resiliency.

I. INTRODUCTION

Machine learning systems have been extensively employed in numerous real-world applications [1], [2], [3], [4], [5], [6], [7], [8]. These systems demand robust defense mechanisms against potential attacks to assure their safe and reliable deployment. The neural network backdoor attack [9], [10] is one of the major adversarial threats against these machine learning systems. A backdoor attack can transpire when users delegate their training tasks to a third party, as illustrated in Fig. 1(a). The attacker poisons the training dataset with triggers, thereby training a network that contains a latent "backdoor". During inference, the backdoored network misclassifies inputs that have been affixed with triggers, whilst classifying trigger-absent inputs correctly. The implications of backdoor attacks are stark and can wield substantial negative societal impacts. For instance, a backdoor attack on an autonomous car's recognition system could engineer accidents, as depicted in Fig. 1(b). Consequently, fortifying neural networks against such backdoor attacks emerges as both an urgent and pivotal necessity.

The all-to-one (A2O) and all-to-all (A2A) attacks represent two fundamental backdoor attack types, as detailed in [9]. The A2O attack fixates on a single target label, prompting the

The associate editor coordinating the review of this manuscript and approving it for publication was Zhenhua Guo.

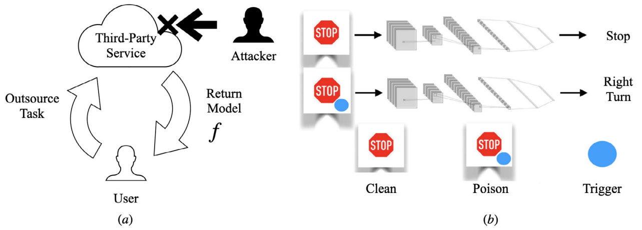


FIGURE 1. (a): A backdoor attack can transpire when users delegate their training tasks to a third party for various reasons, such as a lack of computational resources. (b): The backdoored network misclassifies inputs that have been affixed with triggers, whilst classifying trigger-absent inputs correctly.

backdoored network to output this label whenever the input contains triggers. Conversely, in A2A attacks, target label that is output by the backdoored network can be intricately a function of the poisoned input's ground-truth label. Empirical observations by [11] indicate that attackers can orchestrate an A2O attack with relatively lesser effort and cost compared to an A2A attack. Thus, this work primary focuses on detecting A2O attacks to reduce the potential threats from backdoor attacks by elevating the cost and complexity of launching such attacks.

Specifically, this work addresses black-box A2O backdoor defenses within the realm of image classification tasks. Several existing A2O backdoor defenses [12], [13] are white-box defenses (e.g., modifying model parameters to remove backdoors), necessitating access to the intricate details of the backdoored model (including network architecture, hidden layer outputs, or network parameters). However, it is pertinent to note that such model internals may not always be accessible. For example, (1) users have access to the model via an API [14] where the model is deployed in the cloud; therefore, lacking access to the weights, loss, or logits from the neural network. (2) Users have access restriction to the model's structure and weights due to intellectual property restrictions by the model developer [15]. (3) The user's lack of resources or expertise requires hiring a third party for backdoor detection, however, limiting the third party's access to input and output of the model due to privacy concerns [16]. In scenario (3), the user and the defender are different entities. All these scenarios necessitate the development of defense mechanisms in black-box settings. Therefore, this paper aims to develop a novel backdoor detection strategy for black-box DNNs that is effective across a wide range of triggers. Unlike white-box defenses, our proposed approach detects poisoned samples during the inference stage by requiring access only to the input and output of the network.

Several black-box backdoor defenses [17], [18] employ hand-crafted features for backdoor detection. However, we empirically observed that they show limited performance in certain advanced triggers. For any given input, these defenses intertwine the input's features with those of available clean samples to generate synthetic examples. Subsequent to their creation, these synthetic samples are fed into the network to observe resultant behaviors, with notable disparities often emerging between synthetic clean and synthetic poisoned samples. Nonetheless, the efficacy of such methods depends crucially upon the locational attributes of the trigger and the clean features. When the trigger's location and clean features' locations in a given input are separable, the quality of the hand-crafted features remains robust, thereby ensuring methodological effectiveness. Conversely, when the trigger's location and clean features' locations in a given input are not separable (i.e., they are in the same part of the image), the hand-crafted features may have dependencies on both the trigger and the benign features. Therefore, it becomes more challenging to discriminate between the trigger and the benign features, as the effects of the two are mixed. To circumvent these limitations, this paper introduces a post-training, black-box A2O backdoor defense that is consistently effective regardless of trigger locations.

The proposed defense is motivated by the observation that poisoned samples have a different resiliency to noise perturbations from clean samples in terms of the backdoored network output. We use the metric $L(x+\xi', x; f)$, as introduced in [18], to quantify the resiliency of network f to uniform noise perturbations for sample x, articulated as,

$$L(x+\xi', x; f) = \mathbb{1}\{f(x+\xi') = f(x)\} \tag{1}$$

where $\xi' \sim \Xi'$ denotes a noise perturbation extracted from the uniform distribution Ξ' and $\mathbb{1}\{\cdot\}$ is the indicator function that outputs one if x satisfies the condition inside $\{\cdot\}$ and zero otherwise. Our observation of the statistical divergence between clean and poisoned samples across various A2O attacks is depicted in Fig. 2. Motivated by Fig. 2, we propose a defense strategy against backdoor attacks, which involves training a novelty detector utilizing $\mathbb{E}_{\xi'\sim\Xi'}[L(x+\xi', x; f)]$ of clean samples. During the inference phase, this novelty detector pinpoints samples for which $\mathbb{E}_{\xi'\sim\Xi'}[L(x+\xi', x; f)]$ diverges from those used in training. While adding noise perturbations to inputs is a recognized method in the field for multiple purposes, the novelty of our work is demonstrated through its empirical efficacy against a wide array of triggers in black-box deep neural networks. Specifically, our approach considers noise perturbations of different levels and analyzes their impacts on network outputs. We empirically observed that our methodology outperforms several recent backdoor detection strategies in various cases.

Our work aims to elevate the cost of launching a backdoor attack by being highly effective against A2O attacks, therefore raising the challenge to the attacker. To increase the attack success rate, the attacker therefore has to spend more effort to launch an A2A attack. The contribution of this work includes: 1) identifies a statistical distinction between clean and poisoned samples across varied A2O attacks; 2) introduces a post-training black-box A2O backdoor defense that maintains consistent effectiveness irrespective of trigger locations; 3) undertakes a comprehensive evaluation of the defense across numerous scenarios, juxtaposed with numerous state-of-the-art methods; and 4) furnishes an explanation for the empirical observation.

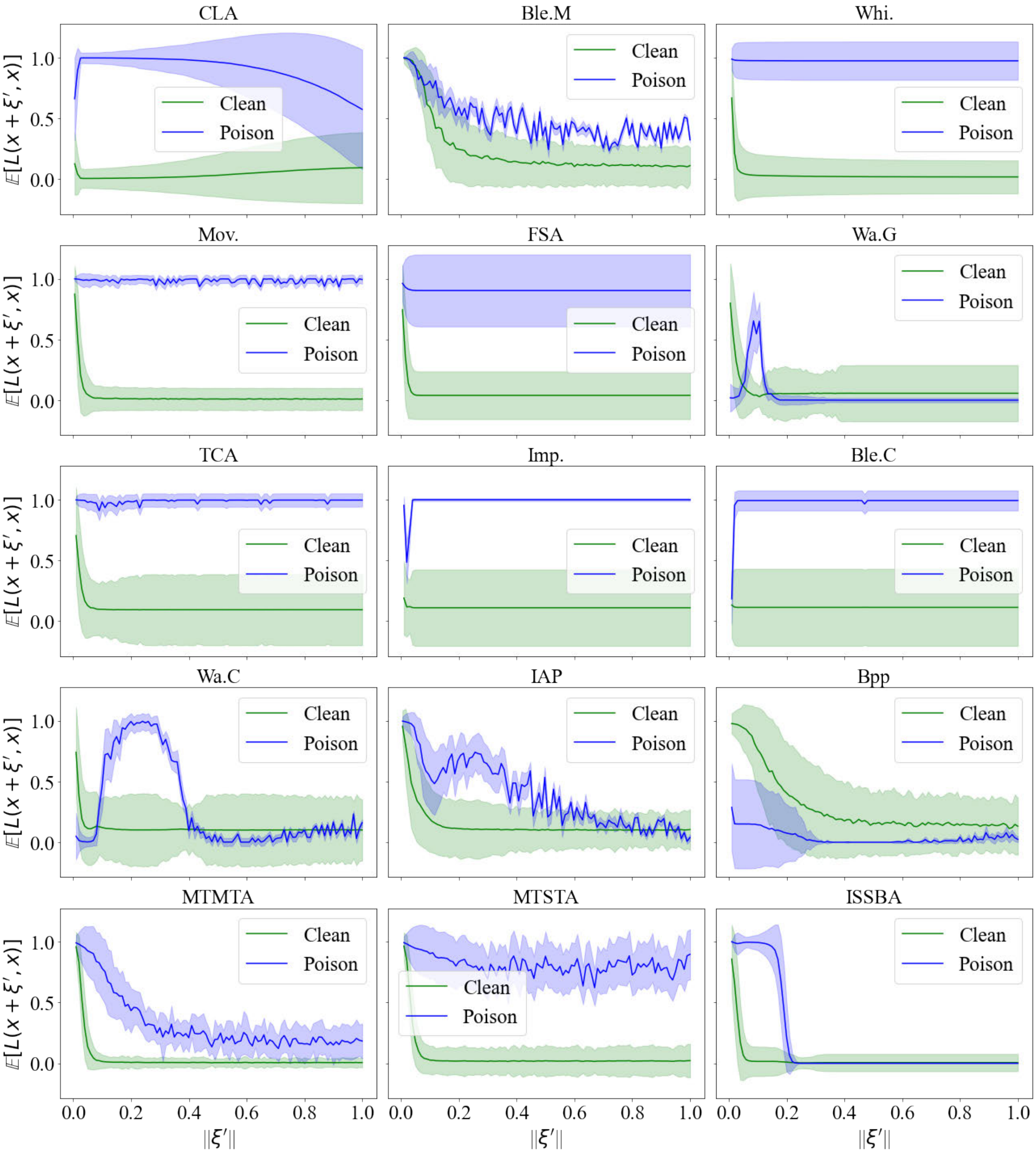


FIGURE 2. The resiliency of clean and poisoned samples to uniform noise in A2O attacks with various triggers. Each subplot illustrates a distinct backdoor attack scenario. The x-axis charts the intensity of these noise perturbations. The y-axis, $\mathbb{E}[L(x+\xi', x; f)] = \mathbb{E}_{\xi'\sim\Xi'}[\mathbb{1}\{f(x+\xi') = f(x)\}]$, displays the proportion of instances where the output labels remain consistent with those observed in the baseline scenario without noise. f is the backdoored network, x is the input, ξ' is the uniform noise, and $\mathbb{1}\{\cdot\}$ is the indicator function that outputs one if x satisfies the condition inside $\{\cdot\}$ and zero otherwise. The depicted solid line represents the average consistency rate across all samples, while the shaded area indicates the standard deviation from this mean. Details of the attack names and the corresponding utilized datasets are provided in Sec. V.

The remainder of this paper is organized as follows: Sec. II explains the existing backdoor attacks and defenses. Sec. III explains the threat model and formulates the problem. Sec. IV introduces our backdoor defense strategy. Sec. V empirically evaluates our approach across numerous scenarios. Sec. VI provides the explanation and discusses the limitation of our approach. Sec. VII concludes the paper.

II. RELATED WORKS

Triggers: The neural network backdoor attack was initially introduced by [9] and [10]. Reference [19] introduced a sample-specific invisible trigger. Reference [20] introduced input-aware backdoor attacks, ensuring a single trigger could not be utilized for two disparate inputs. Reference [21] devised backdoored networks by directly manipulating the network, circumventing the need to poison the training dataset. Reference [22] advanced n-to-one backdoor attacks, wherein the backdoored network outputs the target label only if all n triggers are present on the input. Reference [23] introduced a trigger utilizing the warping effect.

Defenses: The trigger inversion strategy [24], [25] employs reverse-engineered triggers—obtained through the optimization of loss functions with regularized terms—to purify the network. In contrast, our work concentrates on the statistical behavior of the backdoored model and ensuring generalization across all triggers. Pruning approaches [12], [13], [26] endeavor to identify and modify backdoored neurons to curtail the influence of backdoors. For instance, [26] introduces a data-free pruning method. Reference [13] spotlights anomaly neurons to pinpoint backdoored networks. Nevertheless, these approaches are typically constrained to white-box scenarios, wherein defenders possess access to the model's internals, standing in contrast to our method which navigates black-box scenarios and only requires network output access.

Novelty-detection approaches [17], [18], [27] treat clean samples as normal data and poisoned samples as anomalies. They train one-class classifiers using clean samples to detect poisoned instances. STRIP [17] computes an entropy-based confidence score related to network output responses to inputs, distrusting inputs where the network displays abnormal confidence. However, STRIP [17] relies on a blending function to generate synthetic inputs, operating under the premise that the trigger mechanism will remain effective. Nonetheless, this premise is increasingly challenged by the emergence of more sophisticated trigger mechanisms that are designed to be sample-specific or non-localized [19], [20], [23], [28]. We empirically observed that the effectiveness of STRIP against these advanced triggers is limited. Conversely, this work utilizes noise perturbations to generate synthetic inputs, thereby preserving the functionality of the triggers. Reference [18] delineates five metrics for black-box backdoor detection. RAID [27] employs novelty detectors to isolate suspicious inputs and utilizes these quarantined inputs to train an online binary classifier to identify poisoned samples.

Certified backdoor defenses, such as [29], [30], and [31], construct robust classifiers that guarantee certifiable accuracy by introducing noise to, or partitioning, training datasets. Februus [32] and SentiNet [33], both based on GradCAM approaches [34], [35], [36], focus on pinpointing the position of the trigger.

III. PROBLEM FORMULATION

A. THREAT MODEL

Scenario: The user seeks to train a neural network $f : \mathbb{R}^n \rightarrow \mathbb{Z}$ using the clean data distribution $\mathcal{D}$ with support D for classification task purposes. Due to a lack of computational resources, the training task is outsourced to a potential attacker, who, during the training, implements an A2O attack.

A2O attack: Initially, the attacker determines the trigger function g and the target label l^*. Let $\mathcal{D}^*$ represent the poisoned data distribution with support D^*, such that $D^* = \{x + g(x) \mid x \in D\}$. The attacker proceeds to train the network f using a contaminated dataset, which encompasses both clean and poisoned samples such that

$$f(x) = \begin{cases} F(x), & x \in D \\ l^* & x \in D^* \end{cases} \tag{2}$$

with high probability, where $F(x)$ is the ground-truth label of x. The attacker has the liberty to select any portion of the training inputs to be poisoned and to define the training hyperparameters, such as the number of epochs, batch size, learning rate, and so on. Nevertheless, the attacker neither possesses access to the user's validation dataset nor the capability to alter the model structure post-training.

Attacker's goal: The attacker uses classification accuracy (CA) on clean samples and attack success rate (ASR) on poisoned samples to evaluate his attacks. The definitions of CA and ASR are given in the following.

Definition 1 (Classification Accuracy (CA)): The classification accuracy (CA) measures the performance of model f in the clean data distribution, i.e.,

$$CA = \mathbb{E}_{x \sim \mathcal{D}}[\mathbb{1}\{f(x) = F(x)\}]. \tag{3}$$

Definition 2 (Attack Success Rate (ASR)): The attack success rate (ASR) measures the performance of a model f in the poisoned data distribution, i.e.,

$$ASR = \mathbb{E}_{x \sim \mathcal{D}^*}[\mathbb{1}\{f(x) = l^*\}]. \tag{4}$$

In pursuit of executing a stealthy attack and eluding detection, the attacker desires a high CA. Simultaneously, to exert a maximal impact on model performance, the attacker aims to ensure that the ASR is also high.

A2A attack: Besides the A2O attack, the A2A attack is another popular backdoor attack. The definition of A2A attack is given in the following:

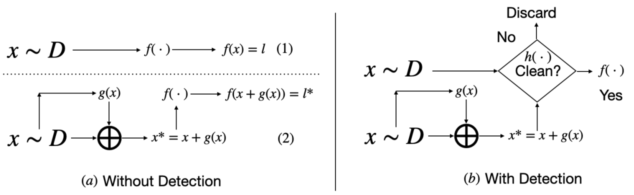
Definition 3 (A2A Attack): Consider m classes, let P be a permutation of $\{0, 1, \ldots, m-1\}$ such that the element with index i in P is not equal to i, i.e., $P(i) \neq i$. An A2A attack is to train a model f such that with high probability such that

$$f(x) = \begin{cases} F(x), & x \in D \\ P(F(x)) & x \in D^*. \end{cases} \tag{5}$$

A typical P is $P(i) = (i + 1) \mod m$.

B. PROBLEM FORMULATION

Scenario: Upon receiving f, the user or defender lacks knowledge regarding whether f is a backdoored network. Nonetheless, should f be compromised, the defender seeks to devise a defense mechanism capable of discriminating between clean and poisoned samples. This defensive strategy should have minimal impact on the CA, irrespective of whether f is backdoored or not.



(a) Without Detection

(b) With Detection

FIGURE 3. The classification process without (a) and with (b) the defense *h*.

The defender's ability: The defender has access solely to the output of f. All other information, such as gradients, weights, and outputs from hidden layers, is unavailable due to considerations of proprietary nature or security. As previously discussed in the introduction, black-box defenses are required when the service provider rents the usage of the deployed model but is not allowed to access the underlying model. This black-box setting is also contemplated in [16] and [18]. The defender possesses a small set of clean data $\{x_i\}_{i=1}^k \sim \mathcal{D}$ (e.g., $k \leq 100$) to confirm the performance of f on clean data. The defender has no prior knowledge of backdoored samples due to the asymmetric advantage held by the attacker, who has full control over the trigger. This assumption is commonly adopted in most backdoor detection methods [12], [17], [18].

Problem Formulation: Given the defender's ability, we wish to build a novelty detector $h(\cdot; f, \{x_i\}_{i=1}^k)$ such that

$$h(x; f, \{x_i\}_{i=1}^k)) = \begin{cases} 1 & x \in D \\ 0 & x \in D^* \setminus D \end{cases} . \quad (6)$$

with high probability. The function h selectively filters out only the poisoned samples while minimally impacting clean samples. Consequently, if f operates as a benign network, its performance will largely remain unaltered by h. The detection process is illustrated in Fig. 3. The defender omits inputs identified as poisoned by h and relies on the output of f for inputs that h identifies as clean, thus diminishing ASR. The defender desires h to be precise to ensure that f, when augmented with h, exhibits high CA and low ASR.

The following two metrics are also used to evaluate the performance of the novelty detector h.

AUROC: True positives (TP) are poisoned samples detected by h. False negatives (FN) are poisoned samples misidentified as clean by h. False Positives (FP) are clean samples misidentified as poisoned by h. TPR is the ratio of the number of TP to the number of total poisoned samples. FPR is the ratio of the number of FP to the number of total clean samples. Receiver operating characteristic (ROC) curve shows TPR and FPR at all thresholds. Area under the ROC curve (AUROC) is the entire two-dimensional area underneath the entire ROC curve. A higher AUROC implies a better performance for the detection algorithm.

AUPR: Precision is the ratio of the number of TP to the total number of TP plus FP (i.e., TP/(TP+FP)). Recall is the ratio of the number of TP to the total number of TP plus FN (i.e., TP/(TP+FN)). Area under the precision and recall (AUPR) has precision and recall as its two parameters. AUPR is especially useful when the testing dataset is imbalanced. A higher AUPR implies a better performance of the approach.

IV. METHODOLOGY

A. INSPIRATION

Our work is inspired by **Theorem 4** from [37] with the following restation:

Theorem 4: Let $f : \mathbb{R}^n \to \mathcal{Y} \subset \mathbb{Z}$ be any deterministic or random function, and let $\xi \sim \mathcal{N}(0, \sigma^2 I)$. Let g be defined as $g(x) = \arg\max_{c \in \mathcal{Y}} \mathbb{P}(f(x + \xi) = c)$. Suppose $c_A \in \mathcal{Y}$ and $\underline{p_A}, \overline{p_B} \in [0, 1]$ satisfy

$$P(f(x + \xi) = c_A) \geq \underline{p_A} \geq \overline{p_B} \geq \max_{c \neq c_A} P(f(x + \xi) = c) \quad (7)$$

Then $g(x + \delta) = c_A$, $\forall\, ||\delta||_2 < R$, where $R = \frac{\sigma}{2}(\Phi^{-1}(\underline{p_A}) - \Phi^{-1}(\overline{p_B}))$ and Φ^{-1} is the inverse of the standard Gaussian cumulative density function.

The theorem implies that, for a specified network f and any input x, points within x's vicinity are anticipated to also yield the label $f(x)$ amidst diverse noise perturbations. Reference [37] demonstrates that the diversity of the training data can influence the size of the vicinity. This result can be leveraged for backdoor detection. To train a backdoored network, an attacker necessitates a corrupted training dataset, which comprises both clean and poisoned samples. In A2O attacks, all poisoned samples map to a singular target label, thereby potentially exhibiting a diversity that is not commensurate with that of the clean training data. In accordance with the theorem, R for clean and poisoned samples are likely to differ significantly. Consequently, the network output's statistical behavior, when subjected to clean noisy inputs and poisoned noisy inputs, should exhibit distinguishable disparities in A2O attacks.

Inspired by the theorem, we utilize the method in [38] to approximate the uniform distribution to analyze network output's statistical behavior on clean noisy inputs and poisoned noisy inputs. The use of uniform noise perturbation in our methodology is inspired by $g(x + \delta) = c_A$, $\forall\, ||\delta||_2 < R$, which suggests that uniform noise perturbation can effectively measure the robustness of the network's output. We use the method outlined in [38] to generate noise vector constantly lies within the sphere, which contains the following steps:

1) Define $\Xi = \mathcal{N}(0, I_n)$ as the Gaussian distribution in $\mathbb{R}^n$ with I_n being the n-dimensional identity matrix and define $U(0, r)$ as the uniform distribution in $[0, r]$ with given r.
2) Sample $\xi \sim \Xi$ and $o \sim U(0, r)$.
3) Let $\xi' = o\xi/||\xi||_2$.

Then ξ' is uniformly distributed in $B_r^0 = \{e \in \mathbb{R}^n | \, ||e||_2 \leq r\}$. It is important to use Gaussian distributions for this method, as other distributions (e.g., uniform) may not result in vectors that are uniformly sampled within the ball. The approach outlined above ensures that each sample generates a valid vector. In contrast, traditional methods that use

n-dimensional uniform distributions and reject points outside the ball would lead to an expected number of trials per valid vector exceeding 1. As the dimensionality n increases, the expected number of trials grows significantly, making the traditional method increasingly inefficient. For k clean samples $\{x_i\}_{i=1}^{k} \sim \mathcal{D}$, we let $r = \alpha \max_i ||x_i||_2$ where $\alpha \in [1, +\infty)$ is a multiplier decided by the defender.

Having the generated uniform noise, we now apply the uniform noise to inputs and visualize them. Consider k' perturbations are drawn with $\{\xi_j\}_{j=1}^{k'}$ and $\{o_j\}_{j=1}^{k'}$, let $\xi'_j = o_j \xi_j / ||\xi_j||_2$. Then we apply ξ'_j to input x and calculate $L_j(x) = L(x + \xi'_j, x; f)]$. Fig. 2 shows the expectation of $L(x)$ by averaging $L_j(x)$. The plots verify our hypothesis that in A2O attacks, the statistical behavior of the network output on clean noisy inputs and poisoned noisy inputs are distinguishable. Additionally, Fig. 2 indicates a weakness of A2O attacks despite of whether the utilized triggers are advanced or naive. Inspired by Fig. 2, we propose the defense mechanism in Alg. 1, which necessitates solely the input and output of the network.

B. THE PROPOSED METHOD

Alg. 1 includes training a novelty detector to detect poisoned samples. To train the detector, we first calculate $L_j(x_i)$ as $L_j(x_i) = L(x_i + \xi'_j, x_i; f)$ with the given k clean validation samples $\{x_i\}_{i=1}^{k}$. In real-world applications, the availability of a small clean validation dataset is a standard assumption in the backdoor detection literature and is a small dataset available to the user and assumed to be backdoor-free (e.g., because the data has been collected by the user themselves using the internet, camera, etc., or bought from a trusted provider). We then construct a novelty detector Ψ. This work uses the k-nearest neighbor (KNN) based method [39] with default parameters, which has a good tradeoff between speed and accuracy. However, other detectors, such as one-class SVM [40], isolation forest [41], or local outlier factor [42], are allowed.

The novelty detector Ψ is trained using $\{L_j(x_i)\}_{i=1,j=1}^{k,k'}$. We used all the default training parameters and standard training protocols provided by the Faiss library [39]. Specifically, the detector Ψ is initialized with the function *faiss.IndexFlatL2(k')*, where k' represents the input dimension. Then, the features, $\{L_j(x_i)\}_{i=1,j=1}^{k,k'}$, of the validation data are added to the detector by the function *Ψ.add($\{L_j(x_i)\}_{i=1,j=1}^{k,k'}$)*. The output of Ψ is the corresponding confidence score. The threshold is typically decided by users. For example, one can calibrate the threshold based on a "budget of clean accuracy drop," e.g., to ensure within 1-5% reduction in clean accuracy for clean validation samples. Note that to have a fair comparison with baseline methods, we set the threshold such that the proposed detector shows classification accuracy close to other methods. During the inference, for the input x, we send $\{L_j(x)\}_{j=1}^{k'}$ to the novelty detector Ψ and observe the confidence score. If the confidence score is higher than T, x is considered as poisoned. Otherwise, x is considered as clean.

Algorithm 1 The Trigger Detector $h(x; f, \{x_i\}_{i=1}^{k})$

```
$r \leftarrow \alpha \max_i ||x_i||_2$ and $\{o_j\}_{j=1}^{k'} \sim U(0, r)$.
$\Xi \leftarrow \mathcal{N}(0, I_n)$ and $\{\xi_j\}_{j=1}^{k'} \sim \Xi$.
$\xi'_j \leftarrow o_j \xi_j / ||\xi_j||_2$ for $j = 1 \rightarrow k'$.
$L_j(x_i) \leftarrow L(x_i + \xi'_j, x_i; f)$ for $i = 1 \rightarrow k$.
Train the novelty detector $\Psi$ using $\{L_j(x_i)\}_{i=1,j=1}^{k,k'}$.
Determine threshold $T$ according to $\{\Psi(\{L_j(x_i)\}_{j=1}^{k'})\}_{i=1}^{k}$.
while True do
    Given an input $x$.
    $L_j(x) \leftarrow L(x + \xi'_j, x; f)$ for $j = 1 \rightarrow k'$.
    if $\Psi(\{L_j(x)\}_{j=1}^{k'}; \{L_{i,j}\}_{i=1,j=1}^{k,k'}) \leq T$ then
        $x$ is clean and $h(x; f, \{x_i\}_{i=1}^{k}) \leftarrow 1$.
    else
        $x$ is poisoned and $h(x; f, \{x_i\}_{i=1}^{k}) \leftarrow 0$.
    end if
end while
```

Complexity: For each input x, our defense requires to calculate $L_j(x)$ for k' noise vectors. Therefore, the computation complexity is $\mathcal{O}(k')$ per sample. To accelerate the testing speed, one can use the batch-wise operation. For a given batch size b, the time complexity become $\mathcal{O}(\lceil k'/b \rceil)$ per sample. The space complexity is $\mathcal{O}(k')$ since a total of k' noise vectors need to be stored. Our approach requires only a few hyper-parameters. In practical settings, simple models with only a few hyper-parameters are preferable. They are easier to understand, implement, and adjust, making our approach more accessible and potentially more appealing for real-world applications. This simplicity, coupled with our method's effectiveness demonstrated in the following section, underscores the practical advancements of our work to the field of backdoor attack detection.

This work distinguishes itself from [37] in both the techniques used and the scope of application. While [37] is focused on adversarial attacks, this work centers on backdoor attacks. The fundamental distinction lies in the nature of the training data: backdoor attacks involve models trained with poisoned data, whereas adversarial attacks use models trained exclusively on clean data. Additionally, [37] evaluates adversarial samples, offering a confidence level for its predictions within a defined radius and a voting strategy for prediction. In contrast, this work uses indicator functions to evaluate the robustness of the backdoored network and adopts a nearest neighbor-based novelty detection strategy to assess the likelihood of a sample being poisoned.

V. EXPERIMENTAL RESULTS

A. SETUP

Datasets: Our backdoor detector h is evaluated on various datasets, including MNIST [43], GTSRB [44], CIFAR-10 [45], YouTube Face [46], and a subset of ImageNet [47].

TABLE 1. Clean and poisoned samples per class in training and testing datasets. "*" means the number is approximated.

		Train / Class		Test / Class		Dimension
Dataset	Class	Clean	Poison	Clean	Poison	C, H, W
MNIST	10	5500*	825*	1000*	1000*	$1, 28, 28$
GTSRB	43	820*	123*	294*	294*	$3, 32, 32$
CIFAR-10	10	5000	750	500	500	$3, 32, 32$
You. Face	1283	81	12	10	10	$3, 55, 47$
ImageNet	200	500	50	10	10	$3, 224, 224$

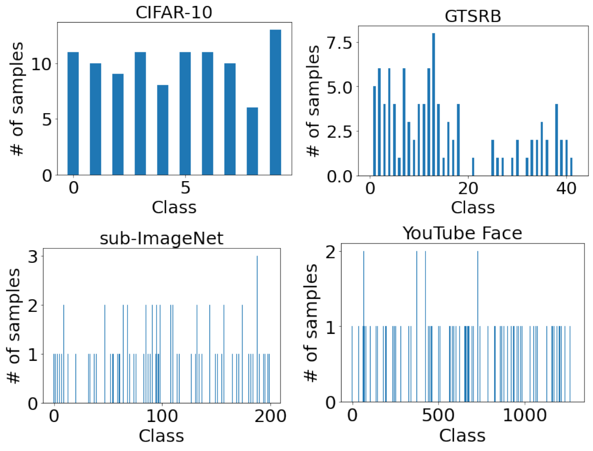


FIGURE 4. Histograms of clean validation samples for $k = 100$.

TABLE 2. Information on backdoored models. MTMTA uses three triggers and each trigger corresponds to a different target label l*.

Trigger	l^*	Model	CA %	ASR %	Visible
CLA	0	CNN	98.1	100	✓
Ble.M	0	CNN	98.99	99.96	✓
Whi.	33	CNN	96.51	97.44	✓
Mov.	0	CNN	95.21	99.88	✓
FSA	35	CNN	94.58	90.27	×
Wa.G	0	PreActiveResNet	99.04	98.95	×
TCA	7	NiN	88.84	99.7	✓
Imp.	0	ResNet	92.59	99.99	✓
Ble.C	0	ResNet	92.82	99.31	✓
Wa.C	0	PreActiveResNet	94.08	99.54	×
IAP	0	PreActiveResNet	94.65	99.32	✓
Bpp	0	PreActiveResNet	94.35	99.98	×
MTMTA	(1, 5, 8)	DeepID	95.95	94.5	✓
MTSTA	4	DeepID	95.91	95.02	✓
ISSBA	0	ResNet	78.36	99.95	×

We split each dataset into a training dataset, a validation dataset, and a testing dataset. We randomly injected the corresponding triggers into 15% of the clean training samples for each class to create the training poisoned samples and changed the ground-truth label to the target label. Table 1 shows the number of clean and poisoned samples per class in the training dataset. CIFAR-10, YouTube Face, and Sub-Imagenet are balanced datasets. Therefore, the numbers in Table 1 are the numbers of samples per class in these three datasets. MNIST and GTSRB are not strictly balanced, but the numbers of samples per class for these two datasets are close to the numbers shown in Table 1. The ratio of poisoned samples to clean samples in testing datasets is 1 (i.e., the testing datasets are balanced), as shown in Table 1. We generated poisoned samples by injecting triggers into their clean versions. We randomly select $k = 100$ clean samples to build the clean validation dataset. Specifically, we partition the training dataset with 100 samples randomly selected to form the validation set. Fig. 4 shows the distribution. The distribution for MNIST will be similar to CIFAR-10 since they both have ten classes. We set k small because, in real-world applications, obtaining a large number of clean validation samples may not be feasible for the defender or user. The clean validation set is assumed to be collected by the defender, who operates independently from the attacker. Consequently, it is practical to gather such a small clean validation set in real-world scenarios. Indeed, such assumption that the defender has a clean validation dataset is used in many papers [12], [17] and some [48] may even assume a larger set of clean data. Alg. 1 is trained exclusively on this clean validation set, without incorporating any poisoned samples.

Backdoor attacks: We consider various classic and advanced backdoor attacks with triggers, as shown in Fig. 5. In MNIST, we consider all label attacks (AAA) [9], clean label attack (CLA) [49], and blended (Ble.M) [50]. In CIFAR-10, we consider the n-to-one attack (TCA) [51], input-aware poisoning (IAP) [20], Wanet (Wa.C) [23], blended (Ble.C), naive pattern (Imp.) [26], and BppAttack (Bpp) [28]. In GTSRB, we consider white box (Whi.) [24], moving trigger (Mov.) [27], feature space trigger (FSA) [13], and Wanet (Wa.G). In ImageNet, an invisible (ISSBA) trigger [19] is utilized. In YouTube Face, we utilize sunglasses (Sun.) [50], lipstick (Lip.), and eyebrow (Eye.). For MNIST, the used model is a standard convolutional neural network (CNN) from BadNet [9]. For GTSRB, the models are a standard CNN from Neural Cleanse [24] and Pre-activation Resnet-18 [52] from [23] and [26]. For CIFAR-10, the models are Network in Network [53] and Pre-activation Resnet-18. For YouTube Face, the network is DeepID from [54]. For ImageNet, Resnet-18 is used. Our approach can even address backdoor attacks in non-neural network models (e.g., SVM [55] or random forest [56]) since it is a black-box detector. We followed the standard training process with CrossEntropy-based loss function and Adam optimizer to make the backdoored networks have CA and ASR shown in Table 2.

Baseline Methods: Fine-Pruning (FP) [12] is a white-box defense that requires no restrictive assumptions. The limitation of FP is that it does not work in the black-box scenario and does not consistently removes backdoors. Compared to Fine-Pruning, STRIP [17] requires less information to fit its defense model and thus is effective in the black-box scenario. Moreover, STRIP aims to capture some statistical properties of backdoor attacks, which aligns with our work. STRIP creates synthetic poisoned samples by blending the poisoned input with clean validation samples. However, STRIP necessitates that the blending process does not compromise the functionality of the trigger. CLP [26] is a data-free white-box backdoor defense. MM-BD [57] detects

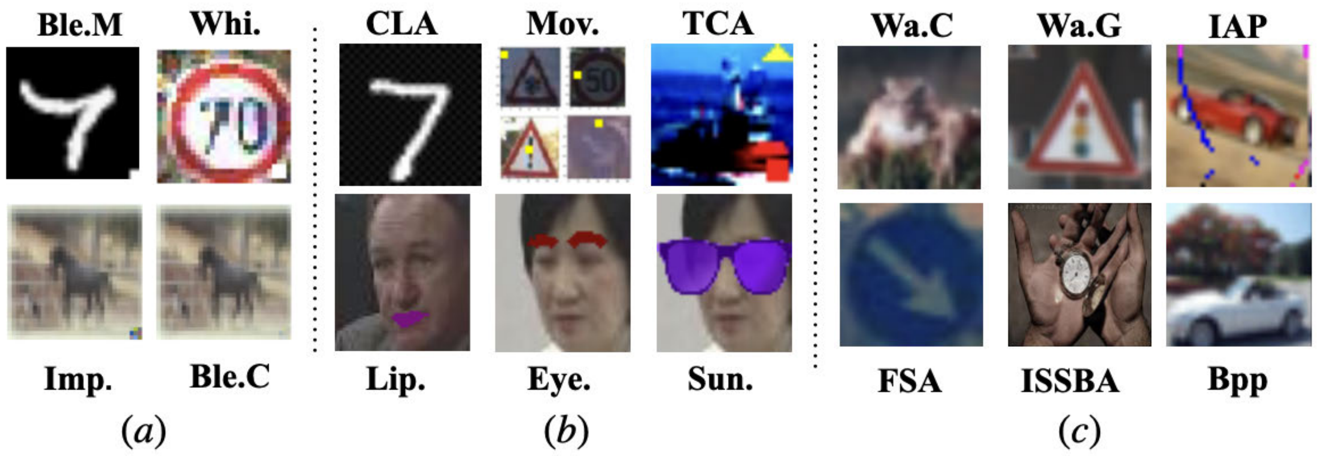


FIGURE 5. Triggers for (a) classic, (b) complex, and (c) advanced A2O backdoor attacks.

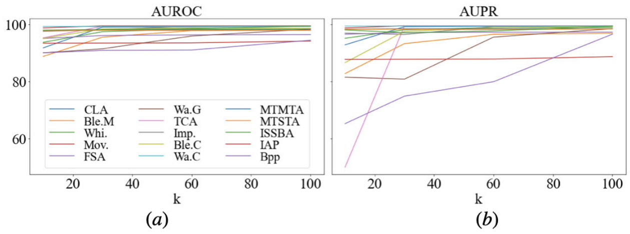


FIGURE 6. AUROC (a) and AUPR (b) of our approach with k clean samples.

target labels based on the observation that the backdoored network tends to overfit to the trigger. Reference [18] uses hand-crafted features and Gaussian noise to detect poisoned samples in black-box scenarios.

Determining hyper-parameters: We choose $\alpha = 6$ to be consistent with Fig. 2. However, the defender is allowed to use a different α. We choose $k' = 900$ so that the run-time of our detector is less than one second for each sample in each case. Specifically, by setting $k' = 900$, our detector takes maximally around 446 milliseconds (in Nvidia Quadro RTX 5000) to check each sample. The threshold T for confidence score is a tradeoff between reducing ASR and maintaining high CA. A higher T implies a lower ASR and CA. To compare with other baseline methods, we set the threshold T such that Alg. 1 shows CA close to other methods. We also show the AUROC and AUPR of Alg. 1 by varying T. Fig. 6 shows the performance of our approach with different numbers k of clean samples being utilized. We consider $k = 100$ affordable for the defender to collect. Additionally, Fig. 6(a) presents the AUROC of our approach for all tested triggers, illustrating the detector's performance across various thresholds. The consistently high AUROC across all triggers indicates the effectiveness of our approach in balancing the false positive rate with the true positive rate, highlighting its overall efficacy. Fig. 6(b) shows that our method achieves high recall by identifying most poisoned inputs and maintains high precision by minimizing false positives among clean inputs. This capability ensures reliable detection of poisoned samples in scenarios with low poison rates. For each comparison method, we obtained the code from the authors' respective websites. For cases where their current code cannot be directly applied, we modify the codes and tune the parameters so that they have CA similar to ours. Additionally, we set $k = 500$ for FP since it requires more clean samples to be effective.

B. EXPERIMENTAL RESULTS

Classic attacks: Ble.M, Whi., Imp., and Ble.C are classic attacks whose triggers are simple patches in the bottom right corner of the image, as shown in Fig. 5(a). The injection methods are superimposing or blending. Table 3 shows that our work, CLP, and Fu's method show reasonable CA and low ASR in all cases, whereas STRIP and FP have high ASR in certain cases. MM-BD misidentifies backdoored network as benign in Ble.M and Whi.

Complex attacks: CLA, Mov., TCA, MTMTA, and MTSTA are more complicated attacks, as shown in Fig. 5(b). CLA is a clean label attack. The trigger is the hidden pattern in the background. The Mov. attack places the trigger (yellow box) randomly in the image. The TCA is an n-to-one trigger. The backdoored network outputs the target label only when both the triangle and square exist in the image. Reference [58] proposes horizontal class backdoor, which is close to TCA in the sense that the backdoored network only predicts the attacker-chosen label when both the trigger and a specific innocuous feature are present in the input. The network will output the ground-truth label as long as either the trigger or the specific innocuous feature is missing. MTMTA and MTSTA have three triggers: lipsticks, eyebrow, and sunglasses. In MTMTA, each trigger corresponds to a different target label, whereas in MTSTA, all triggers correspond to the same target label. According to Table 3, both our work and Fu's method are effective. However, our approach shows a higher CA and lower ASR than Fu's method. STRIP, FP, MM-BD, and CLP show mixed performance.

Advanced attacks: Fig. 5(c) shows advanced triggers. FSA poisons samples through a Gotham filter. IAP is an input-aware attack. The trigger is pixel patterns. Each input has a unique trigger. The invisible triggers are Wa.G, Wa.C, Bpp, and ISSBA. According to Table 3, our method demonstrates superior performance in FSA and ISSBA, while offering performance that is commensurate with the top results of compared approaches in Wa.G, Wa.C, IAP, and Bpp. In the Bpp attack, poisoned samples exhibit less resilience to uniform noise than clean samples, as illustrated in Fig. 2. We found that using the simplest defense that applies mild noise perturbations to inputs results in a 93.11% CA and a 9.68% ASR. It is important to note that although these advanced triggers are designed to mimic the robustness of clean samples and evade detection, our approach effectively identifies them. This effectiveness stems from the inherent limitation that attackers cannot fully eliminate the statistical differences between clean and poisoned samples in A2O backdoor attacks.

Comparison with Fu's method: While both approaches demonstrate effectiveness across a range of attacks, including CLA, Whi., Mov., FSA, TCA, Ble.C, MTMTA, MTSTA, and ISSBA, this work achieves a more favorable balance. Specifically, it maintains higher CA and achieves a lower ASR than those observed in [18]. Furthermore, [18] exhibits limited performance in ASR against advanced adaptive Wa.C and IAP attacks, this work shows marked improvement in these areas. This enhancement shows a wider range of applicability of the proposed approach in addressing diverse backdoor threats. Additionally, the utilized methodologies are different and provide different insights on how to attack the backdoor detection problem. Reference [18] and this work adopt distinct approaches to analyze network outputs influenced by clean and poisoned features. Reference [18] employs five metrics to evaluate the relative impact of clean versus poisoned features, initiated by generating synthetic inputs through regional content transfer between validation samples and inputs. Reference [18] aims to discern the overriding influence of poisoned features on network outputs. In contrast, this study takes a different approach by evaluating the absolute robustness of inputs to noise perturbations, adding different levels of uniform noise to inputs and assessing how this impacts the network's output robustness. This study focuses on the premise that poisoned samples exhibit greater resilience to noise. It is conceivable that the two methodologies may be utilized in the future to further enjoy the benefits of the two approaches simultaneously.

C. LOW POISON RATE

The above experiments are conducted in scenarios where the poison rate is around 50%. We also evaluate our approach in scenarios where the poison rate is small (e.g., $\leq 0.5\%$) but the ASR is high (e.g., $\geq 97.5\%$). Specifically, we use all the clean test samples but only 50 poisoned test samples in each attack case. The performance of our approach is shown in Table 4. Our method maintains its efficacy even in these low poison rate cases.

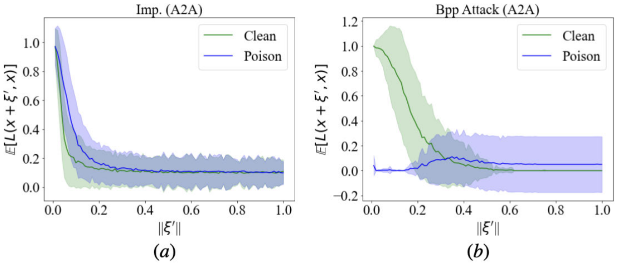


FIGURE 7. The resiliency of clean and poisoned samples to uniform noise in A2A attacks for (a) Imp. in MNIST and (b) Bpp in CIFAR-10. The setup is the same to Fig. 2.

D. PERFORMANCE ON BENIGN MODELS

We trained benign models on MNIST, CIFAR-10, GTSRB, and YouTube Face, achieving CA of 98.95%, 94.71%, 96.59%, and 97.85%, respectively. Subsequently, we applied our approach to the benign models, utilizing a threshold T configured to yield a 2% false positive rate in the clean validation datasets. Consequently, the CA altered to 95.9%, 92.17%, 94.47%, and 95.81% for each dataset, respectively, suggesting that our approach imposes only a mild impact on benign models.

E. A2A ATTACK

The resiliency of clean and poisoned samples to uniform noise presents more complexity in A2A attacks than in A2O attacks. If the resiliency of clean and poisoned samples to uniform noise aligns with Fig. 7(a), our approach will lack efficacy, exhibiting an AUROC of 73%. Conversely, in the context of the A2A attack illustrated in Fig. 7(b), our approach provides an AUROC of 97.5%. Future work aims to enhance our approach against A2A attacks.

F. DIFFERENT NOVELTY DETECTORS

The results presented above are based on a KNN-based novelty detector. To further evaluate our approach, we also tested it using isolation forests and support vector machines (SVM). The corresponding results are summarized in Table 5. While all three detectors perform well on average, the KNN-based novelty detector demonstrates slightly better performance compared to the others. These findings indicate that our approach, which differentiates between poisoned and clean samples through noise perturbations, is effective and generalizable across various novelty detection methods. We also observed that using only KNN without incorporating our noise perturbation approach results in a significant drop in detection accuracy. For example, in the case of the FSA trigger, using only KNN yields an ASR of 57.75 while maintaining a CA of 92.17. In contrast, with the addition of noise perturbation, our approach reduces the ASR to 0.2 while improving the CA to 94.33. This substantial difference

TABLE 3. Comparison with other baseline methods.

	No Defense			Ours		CLP [26]		FP [12]		STRIP [17]		Fu's Method [18]		MM-BD [57]
Trigger	CA	ASR	l^*	CA	ASR	CA	ASR	CA	ASR	CA	ASR	CA	ASR	Detected l^*
CLA	89.1	100	0	88.1	0.73	60.29	50.5	85.02	5.41	**88.17**	**0.02**	86.7	0	0
Ble.M	98.99	99.96	0	92.48	0.76	89.56	0	92.78	5.31	91.98	0.96	**94.61**	**0**	None
Whi.	96.51	97.44	33	**95.77**	**0**	94.74	2.31	92.15	30.1	95.5	18.43	92.69	0	None
Mov.	95.21	99.88	0	**94.62**	**0**	78.84	0.85	92.25	24.21	94.13	79.95	92.09	0	0
FSA	94.58	90.27	35	**94.33**	**0.2**	94.7	90.27	88.38	3.79	93.71	88.5	91.29	2.45	35
Wa.G	99.04	98.95	0	**96.17**	8.22	93.25	98.95	91.60	98.40	94.07	95.68	94.29	**1**	40
TCA	88.84	99.7	7	**88.12**	**0.1**	85.48	99.54	78.77	42.3	81.9	0.3	85.08	0.3	None
Imp.	92.59	99.99	0	90.11	4.19	90.4	5.16	89.12	6.78	90.03	64.25	**90.4**	**0.2**	0
Ble.C	92.82	99.31	0	**91.82**	**0**	89.70	4.03	86.31	5.23	91.27	64.38	90.09	0	0
Wa.C	94.08	99.54	0	89.72	**4.76**	90.41	10.38	87.01	99.47	87.3	99.4	**90.99**	13.81	0
IAP	94.65	99.32	0	86.08	10.13	**88.27**	**9.28**	88.04	99.32	86.1	95.1	87.29	99.32	0
Bpp	94.35	99.98	0	92.84	9.63	**93.03**	9.79	91.0	99.98	92.83	99.98	92.39	**5.7**	None
MTMTA	95.95	94.5	(1,5,8)	**95.01**	**0**	7.83	49.36	90.36	31.4	94.97	14.28	93.59	0.2	1
MTSTA	95.91	95.02	4	**95.11**	**0**	0.94	94.72	91.6	24.3	94.94	12.37	94.19	0	None
ISSBA	78.36	99.95	0	**76.79**	**0.25**	73	99.67	74.33	94.71	74.01	24.33	76.68	1.13	102

TABLE 4. Performance of our approach in low poison rate scenarios.

Trigger	CA	ASR	Trigger	CA	ASR
CLA	86.23	0	Ble.M	92.24	4
Whi.	93.78	0	Mov.	94.62	0
FSA	91.66	0	Wa.G	92.13	6
TCA	87.92	0	Imp.	90.08	0
Ble.C	89.9	0	Wa.C	87.54	6
IAP	86.08	8	Bpp	93.96	10
MTMTA	94.93	0	MTSTA	95.11	0
ISSBA	76.59	0			

TABLE 5. Comparison of AUC values for different methods.

Trigger	KNN	Isolation Forest	SVM
CLA	0.9933	0.9593	0.9796
Ble.M	0.9793	0.9192	0.7760
Whi.	0.9837	0.9779	0.9680
Mov.	0.9948	0.9860	0.9845
FSA	0.9642	0.9490	0.9169
Wa.G	0.9833	0.9397	0.9262
TCA	0.9934	0.9920	0.9873
Imp.	0.9943	0.9881	0.9822
Ble.C	0.9864	0.9556	0.9578
Wa.C	0.9926	0.8711	0.8905
IAP	0.9414	0.8964	0.5263
Bpp	0.9444	0.9050	0.9363
MTMTA	0.9808	0.9733	0.9638
MTSTA	0.9795	0.9809	0.9710
ISSBA	0.9937	0.9892	0.9895
Ave.	0.9803	0.9522	0.9171

highlights that our approach significantly enhances detection accuracy.

G. DIFFERENT NUMBER OF NEIGHBORS

We further evaluated the performance of our approach by varying the number of nearest neighbors. The results, presented in Table 6, demonstrate that our approach performs well when the number of nearest neighbors is 10 or greater.

VI. DISCUSSION

Why does the poisoned sample typically exhibit more resiliency to uniform noise than the clean sample, as demonstrated in Fig. 2? This statistical behavior might be associated with the prevalent backdoor attack scheme that generates poisoned samples by composing the trigger(s) with clean samples. If a single trigger is employed, the size of the poisoned sample distribution will equate to the size of the clean sample distribution. If multiple triggers are utilized, the size of the poisoned sample distribution will exceed that of the clean sample distribution. If we denote Q as the uniform distribution on $D \cup D^*$ with the probability density function q, where D and D^* are the respective supports of clean and poisoned data distributions, we will have,

$$q(D^*) \geq q(D). \tag{8}$$

where $q(D) = \int_D q(x)dx$ and $q(D^*) = \int_{D^*} q(x)dx$. We also have the following theorem to explain Fig. 2.

TABLE 6. AUROC of KNN with different number of nearest neighbors.

Trigger	5	10	15	20
CLA	0.7835	0.9933	0.9942	0.9942
Ble.M	0.9837	0.9793	0.9833	0.9829
Whi.	0.4499	0.9837	0.9674	0.9706
Mov.	0.9950	0.9948	0.9950	0.9950
FSA	0.9332	0.9642	0.9500	0.9497
Wa.G	0.9748	0.9833	0.9545	0.9218
TCA	0.9930	0.9934	0.9930	0.9929
Imp.	0.9947	0.9943	0.9948	0.9947
Ble.C	0.9158	0.9864	0.9331	0.9448
Wa.C	0.9451	0.9926	0.9396	0.9374
IAP	0.8926	0.9414	0.9291	0.9280
Bpp	0.9591	0.9444	0.9531	0.9502
MTMTA	0.9800	0.9808	0.9801	0.9801
MTSTA	0.9707	0.9795	0.9723	0.9729
ISSBA	0.9580	0.9937	0.9926	0.9928

Theorem 5: Given (8) and any A2O model f, we have

$$\sup_{x\in D} \mathbb{E}_{e\sim Q}[L(e,x;f)] \leq \inf_{x^*\in D^*} \mathbb{E}_{e\sim Q}[L(e,x^*;f)] \tag{9}$$

where L is given in (1). The equality holds if and only if $x \in D$ and $f(x) = l^*$ with the target label l^*.

Proof: Consider the single-trigger case with function g. Denote $U_l = \{x \in D \cup D^* | f(x) = l\}$, then $\mathbb{E}_{e\sim Q}[L(e,x;f)] = \int_{D\cup D^*} L(e,x;f)q(e)de = \int_{U_l} q(e)de = q(U_l)$ for $x \in U_l$. Since this is an A2O attack, we also have $\mathbb{E}_{e\sim Q}[L(e,x+g(x);f)] = \int_{D\cup D^*} L(e,x+g(x);f)q(e)de = q(D^*)$. For $l \neq l^*$, we have $q(U_l) < q(D) \leq q(D^*)$. If $l = l^*$, we have $U_{l^*} = D^*$ and thus $q(U_{l^*}) = q(D^*)$. Therefore, $T = \sup_{x \in D} \mathbb{E}_{e \sim Q}[L(e, x; f)] \leq \inf_{x^* \in D^*} \mathbb{E}_{e \sim Q}[L(e, x^*; f)] = \overline{T}$. On the other hand, if $q(U_l) = q(D^*)$, it implies that $U_l = D^* = U_{l^*}$. Extending the single-trigger case to the multi-trigger case with w triggers and target labels, we have $T \leq \overline{T}_i$ for $i = 1, 2, \ldots, w$. The theorem still holds true. □

The theorem is independent of triggers. Therefore, both advanced and naive triggers show similar statistical behavior, as illustrated in Fig. 2. However, for A2A attacks, $q(D^*) \approx q(D)$ holds most of the time, which undermines the conditions required to guarantee the above theorem. Consequently, our approach is less effective against A2A attacks. Note that Theorem 4 states that for any input, there exists a neighborhood (circle) around that input within which the output remains the same. This holds true for both clean and poisoned samples in backdoor attacks. Furthermore, Theorem 5 demonstrates that these neighborhoods differ between clean and poisoned samples in A2O attacks.

A. LIMITATION AND SOCIETAL IMPACT

Our work demonstrates that, in extant A2O attacks, poisoned samples typically exhibit greater resiliency to uniform noise than clean samples. Defenders could employ our detector to filter out A2O attacks, thereby enhancing the safety and reliability of machine learning systems. Nonetheless, we also illustrate that in A2A attacks, the resilience of poisoned and clean samples to uniform noise can be identical. Consequently, attackers might design stealthy A2A attacks that circumvent our detection. This scenario paves the way for the emergence of more sophisticated backdoor attacks, suggesting that machine learning systems might remain vulnerable even when employing our approach. Future work aims to enhance our approach for consistent effectiveness against A2A attacks.

Currently, our approach is designed for images. However, with the growing adoption of large language models, extending our method to these models and text inputs is a promising direction. Unlike images, where noise can be easily added, text inputs require alternative methods for introducing perturbations. Furthermore, language model outputs often consist of more than a single label, adding another layer of complexity. Adapting our approach to handle these challenges is an important avenue for future work.

On the positive side, our approach can be applied in critical applications, such as facial recognition systems, to prevent backdoor attacks. However, it also has the potential for misuse. Rather than employing it to detect attacks, adversaries could use our method to assess the stealthiness of their triggers and enhance them, making the triggers even harder to detect.

VII. CONCLUSION

A statistical difference between clean and poisoned samples in terms of the backdoored network's resiliency to uniform noise is observed. A post-training black-box A2O backdoor defense is proposed based on the empirical observation. The proposed approach is evaluated on various datasets and compared with state-of-the-art methods. A thorough explanation is provided to explain the observation.

REFERENCES

[1] N. Patel, P. Krishnamurthy, S. Garg, and F. Khorrami, "Bait and switch: Online training data poisoning of autonomous driving systems," in *Proc. NeurIPS Workshop Dataset Curation Secur.*, Jan. 2020, pp. 1–6.

[2] S. Zhao, Z. Dong, Z. Cao, and R. Douady, "Hedge fund portfolio construction using PolyModel theory and iTransformer," 2024, *arXiv:2408.03320*.

[3] J. Kim, Z. Dong, and P. Polak, "Face-GPS: A comprehensive technique for quantifying facial muscle dynamics in videos," 2024, *arXiv:2401.05625*.

[4] S. Zhao, D. Wang, and R. Douady, "PolyModel for Hedge Funds' portfolio construction using machine learning," 2024, *arXiv:2412.11019*.

[5] H. Fu, P. Krishnamurthy, and F. Khorrami, "Functional replicas of proprietary three-axis attitude sensors via LSTM neural networks," in *Proc. IEEE Conf. Control Technol. Appl. (CCTA)*, Aug. 2020, pp. 70–75.

[6] A. Papanicolaou, H. Fu, P. Krishnamurthy, and F. Khorrami, "A deep neural network algorithm for linear-quadratic portfolio optimization with MGARCH and small transaction costs," *IEEE Access*, vol. 11, pp. 16774–16792, 2023.

[7] A. Papanicolaou, H. Fu, P. Krishnamurthy, B. Healy, and F. Khorrami, "An optimal control strategy for execution of large stock orders using long short-term memory networks," *J. Comput. Finance*, vol. 26, no. 4, pp. 1–29, 2023.

[8] A. Sarmadi, H. Fu, P. Krishnamurthy, S. Garg, and F. Khorrami, "Privacy-preserving collaborative learning through feature extraction," *IEEE Trans. Depend. Secure Comput.*, vol. 21, no. 1, pp. 486–498, 2024.

[9] T. Gu, K. Liu, B. Dolan-Gavitt, and S. Garg, "BadNets: Evaluating backdooring attacks on deep neural networks," *IEEE Access*, vol. 7, pp. 47230–47244, 2019.

[10] Y. Liu, S. Ma, Y. Aafer, W.-C. Lee, J. Zhai, W. Wang, and X. Zhang, "Trojaning attack on neural networks," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2018, p. 15.

[11] K. Doan, Y. Lao, W. Zhao, and P. Li, "LIRA: Learnable, imperceptible and robust backdoor attacks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 11946–11956.

[12] K. Liu, B. Dolan-Gavitt, and S. Garg, "Fine-pruning: Defending against backdooring attacks on deep neural networks," in *Proc. Res. Attacks, Intrusions, Defenses*, Jan. 2018, pp. 273–294.

[13] Y. Liu, W.-C. Lee, G. Tao, S. Ma, Y. Aafer, and X. Zhang, "ABS: Scanning neural networks for back-doors by artificial brain stimulation," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nov. 2019, pp. 1265–1282.

[14] W. Aiken, H. Kim, S. Woo, and J. Ryoo, "Neural network laundering: Removing black-box backdoor watermarks from deep neural networks," *Comput. Secur.*, vol. 106, Jul. 2021, Art. no. 102277.

[15] J. Guo, A. Li, and C. Liu, "AEVA: Black-box backdoor detection using adversarial extreme value analysis," in *Proc. Int. Conf. Learn. Represent.* Roslyn, NY, USA: Black, Jan. 2021, pp. 1–24.

[16] Y. Dong, X. Yang, Z. Deng, T. Pang, Z. Xiao, H. Su, and J. Zhu, "Black-box detection of backdoor attacks with limited information and data," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16462–16471.

[17] Y. Gao, C. Xu, D. Wang, S. Chen, D. C. Ranasinghe, and S. Nepal, "STRIP: A defence against trojan attacks on deep neural networks," in *Proc. 35th Annu. Comput. Secur. Appl. Conf.*, Dec. 2019, pp. 113–125.

[18] H. Fu, P. Krishnamurthy, S. Garg, and F. Khorrami, "Differential analysis of triggers and benign features for black-box DNN backdoor detection," *IEEE Trans. Inf. Forensics Security*, vol. 18, pp. 4668–4680, 2023.

[19] Y. Li, Y. Li, B. Wu, L. Li, R. He, and S. Lyu, "Invisible backdoor attack with sample-specific triggers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 16443–16452.

[20] A. Nguyen and A. Tran, "Input-aware dynamic backdoor attack," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2020, pp. 3454–3464.

[21] S. Hong, N. Carlini, and A. Kurakin, "Handcrafted backdoors in deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 8068–8080.

[22] M. Xue, C. He, J. Wang, and W. Liu, "one-to-N & N-to-one: Two advanced backdoor attacks against deep learning models," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 3, pp. 1562–1578, May 2022.

[23] T. Nguyen and A. Tran, "WaNet–imperceptible warping-based backdoor attack," in *Proc. Int. Conf. Learn. Represent.*, May 2021, pp. 1–16.

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

[25] X. Qiao, Y. Yang, and H. Li, "Defending neural backdoors via generative distribution modeling," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 32, Oct. 2019, pp. 14004–14013.

[26] R. Zheng, R. Tang, J. Li, and L. Liu, "Data-free backdoor removal based on channel lipschitzness," in *Proc. Eur. Conf. Comput. Vis.*, Jan. 2022, pp. 175–191.

[27] H. Fu, A. K. Veldanda, P. Krishnamurthy, S. Garg, and F. Khorrami, "A feature-based on-line detector to remove adversarial-backdoors by iterative demarcation," *IEEE Access*, vol. 10, pp. 5545–5558, 2022.

[28] Z. Wang, J. Zhai, and S. Ma, "BppAttack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 15054–15063.

[29] J. Jia, Y. Liu, X. Cao, and N. Z. Gong, "Certified robustness of nearest neighbors against data poisoning and backdoor attacks," in *Proc. AAAI Conf. Artif. Intell.*, vol. 36, Jun. 2022, pp. 9575–9583.

[30] C. Xie, M. Chen, P. Chen, and B. Li, "CRFL: Certifiably robust federated learning against backdoor attacks," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2021, pp. 11372–11382.

[31] Y. Zhang, A. Albarghouthi, and L. D'Antoni, "BagFlip: A certified defense against data poisoning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2022, pp. 31474–31483.

[32] B. G. Doan, E. Abbasnejad, and D. C. Ranasinghe, "Februus: Input purification defense against trojan attacks on deep neural network systems," in *Proc. Annu. Comput. Secur. Appl. Conf.*, Dec. 2020, pp. 897–912.

[33] E. Chou, F. Tramèr, and G. Pellegrino, "SentiNet: Detecting localized universal attacks against deep learning systems," in *Proc. IEEE Secur. Privacy Workshops (SPW)*, May 2020, pp. 48–54.

[34] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[35] H.-S. Ham, H.-S. Lee, J.-W. Chae, H. C. Cho, and H.-C. Cho, "Improvement of gastroscopy classification performance through image augmentation using a gradient-weighted class activation map," *IEEE Access*, vol. 10, pp. 99361–99369, 2022.

[36] J. Zhang, W. Li, S. Liang, H. Wang, and J. Zhu, "Adversarial samples for deep monocular 6D object pose estimation," 2022, *arXiv:2203.00302*.

[37] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *Proc. Int. Conf. Mach. Learn.*, Jan. 2019, pp. 1310–1320.

[38] M. E. Müller, "A note on a method for generating points uniformly on n -dimensional spheres," *Commun. ACM*, vol. 2, no. 4, pp. 19–20, Apr. 1959.

[39] J. Johnson, M. Douze, and H. Jégou, "Billion-scale similarity search with GPUs," *IEEE Trans. Big Data*, vol. 7, no. 3, pp. 535–547, Jul. 2021.

[40] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, Jul. 2001.

[41] F. T. Liu, K. M. Ting, and Z. Zhou, "Isolation forest," in *Proc. IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 413–422.

[42] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF: Identifying density-based local outliers," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2000, pp. 93–104.

[43] Y. LeCun and C. Cortes. (2010). *MNIST Handwritten Digit Database*. [Online]. Available: http://yann.lecun.com/exdb/mnist

[44] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel, "Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition," *Neural Netw.*, vol. 32, pp. 323–332, Aug. 2012.

[45] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf

[46] L. Wolf, T. Hassner, and I. Maoz, "Face recognition in unconstrained videos with matched background similarity," in *Proc. CVPR*, Jun. 2011, pp. 529–534.

[47] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.

[48] H. Kwon, "Detecting backdoor attacks via class difference in deep neural networks," *IEEE Access*, vol. 8, pp. 191049–191056, 2020.

[49] K. Liu, B. Tan, R. Karri, and S. Garg, "Poisoning the (Data) well in ML-based CAD: A case study of hiding lithographic hotspots," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2020, pp. 306–309.

[50] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," 2017, *arXiv:1712.05526*.

[51] A. K. Veldanda, K. Liu, B. Tan, P. Krishnamurthy, F. Khorrami, R. Karri, B. Dolan-Gavitt, and S. Garg, "NNoculation: Catching BadNets in the wild," in *Proc. 14th ACM Workshop Artif. Intell. Secur.*, Nov. 2021, pp. 49–60.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[53] M. Lin, Q. Chen, and S. Yan, "Network in network," in *Proc. Int. Conf. Learn. Represent.*, Dec. 2013, pp. 1–10.

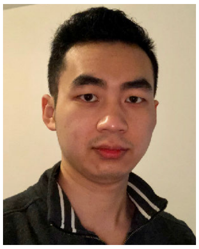
[54] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[55] J. Platt, "Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods," *Adv. Large Margin Classifiers*, vol. 10, no. 3, pp. 61–74, Jan. 1999.

[56] L. Breiman, "Random forests," *Mach. Learn.*, vol. 45, no. 1, pp. 5–32, 2001.

[57] H. Wang, Z. Xiang, D. J. Miller, and G. Kesidis, "MM-BD: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2024, pp. 1994–2012.

[58] H. Ma, S. Wang, Y. Gao, Z. Zhang, H. Qiu, M. Xue, A. Abuadbba, A. Fu, S. Nepal, and D. Abbott, "Watch out! Simple horizontal class backdoor can trivially evade defense," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Nepal, Dec. 2024, pp. 4465–4479.



HAO FU was born in Anyang, China, in November 1994. He received the Bachelor of Science degree in physics from the University of Science and Technology of China, Hefei, China, in 2017, and the Master of Science and Ph.D. degrees in electrical engineering from the Department of Electrical and Computer Engineering, New York University (NYU) Tandon School of Engineering, Brooklyn, NY, USA, in 2019 and 2024, respectively. His major field of study contains machine learning, finance, and control theory.

From 2017 to 2018, he was a Research Assistant in NYU Wireless Laboratory. In Fall 2018, he joined in Control/Robotics Research Laboratory (CRRL). Previously, he was studied the possibility of using machine learning tools to develop economical navigation algorithms. Additionally, he also studied the possibility of using neural networks to assist decision-making in finance. Currently, he is studying backdooring attacks against neural networks and security problems in cyber-physical systems. He has published articles in many journals, including IEEE Transactions on Information Forensics and Security, IEEE Transactions on Dependable and Secure Computing, and IEEE Access.

PRASHANTH KRISHNAMURTHY (Member, IEEE) received the B.Tech. degree in electrical engineering from the Indian Institute of Technology, Chennai, in 1999, and the M.S. and Ph.D. degrees in electrical engineering from Polytechnic University (currently NYU), in 2002 and 2006, respectively. He is currently a Research Scientist and an Adjunct Faculty with the Department of Electrical and Computer Engineering at NYU Tandon School of Engineering. He has co-authored over 175 journal and conference papers. He has co-authored the book *Modeling and Adaptive Nonlinear Control of Electric Motors* (Springer Verlag, in 2003). His research interests include autonomous vehicles and robotic systems, multi-agent systems, sensor data fusion, robust and adaptive nonlinear control, resilient control, path planning and obstacle avoidance, machine learning, real-time embedded systems, electromechanical systems modeling and control, cyber-physical systems and cyber-security, decentralized and large-scale systems, high-fidelity and hardware-in-the-loop simulation, and real-time software implementations.

SIDDHARTH GARG received the B.Tech. degree in electrical engineering from IIT Madras and the Ph.D. degree in electrical and computer engineering from Carnegie Mellon University, in 2009.

He was an Assistant Professor with the University of Waterloo, from 2010 to 2014. In 2014, he joined New York University (NYU), as an Assistant Professor. His general research interests include computer engineering, more particularly secure, reliable, and energy-efficient computing.

Dr. Garg was a recipient of the NSF CAREER Award, in 2015. He received Best Paper Awards from the IEEE Symposium on Security and Privacy (S&P), in 2016, the USENIX Security Symposium, in 2013, the Semiconductor Research Consortium TECHCON, in 2010, and the International Symposium on Quality in Electronic Design (ISQED), in 2009. He also received the Angel G. Jordan Award from the Electrical and Computer Engineering (ECE) Department, Carnegie Mellon University, for outstanding dissertation contributions and service to the community.

FARSHAD KHORRAMI (Fellow, IEEE) received the bachelor's degree in mathematics and electrical engineering, the master's degree in mathematics, and the Ph.D. degree in electrical engineering from The Ohio State University, Columbus, OH, USA, in 1982, 1984, and 1988, respectively.

He is currently a Professor with the Electrical and Computer Engineering Department, NYU, Brooklyn, NY, USA, where he joined as an Assistant Professor, in September 1988. He has developed and directed the Control/Robotics Research Laboratory, Polytechnic University (currently NYU) and the Co-Director of the Center in AI and Robotics (CAIR), NYU Abu Dhabi. He has commercialized UAVs and the development of auto-pilots for various unmanned vehicles. His research has been supported by the ARO, NSF, ONR, DARPA, ARL, AFRL, NASA, and several corporations. His research interests include adaptive and nonlinear controls, robotics and automation, unmanned vehicles, cyber security for CPS, embedded systems security, machine learning, and large-scale systems and decentralized control. He has published over 360 refereed journal and conference papers in these areas. He has authored a book *Modeling and Adaptive Nonlinear Control of Electric Motors* (Springer Verlag, in 2003). He also has 15 U.S. patents on novel smart micro-positioners, control systems, cyber security, and wireless sensors and actuators.

Dr. Khorrami has served as a conference organizing committee member for several international conferences.