Supplementary Material for
Attack To Defend: Exploiting Adversarial Attacks for Detecting Poisoned Models

This supplementary material contains additional experiments and discussions covering various aspects of the proposed A2D framework.

A. Extension of A2D to White-Box Setting

In some application scenarios (e.g., if the defender is allowed to download a pre-trained model for local transfer learning), white-box access to the target model may be available. In this scenario, the A2D framework can be easily extended in the following ways. Firstly, the reference model architecture can be chosen to be the same as the target model. Secondly, since the target model is known, its SAP value $\hat{\mathcal{S}}(M_{\theta_t}, \epsilon)$ can be directly estimated instead of estimating $\hat{\mathcal{S}}^*$. Finally, rather than relying on the SAP value of the target model at any single value of ϵ, a more robust metric called the Adversarial Sensitivity Index (ASI) can be defined as follows.

$$ASI(M_\theta) = \int_0^1 \mathcal{S}(M_\theta, \epsilon) d\epsilon. \tag{1}$$

ASI is a measure of the overall sensitivity of an ML model to adversarial attacks and is defined as the area under the SAP curve of the given model. In general, adversarially robust models are expected to have lower values of ASI compared to non-robust models. Since the function $\mathcal{S}(M_\theta, \epsilon)$ is not available in closed-form, the ASI value of a model can be found using the basic Monte Carlo estimator, i.e., uniformly sample K random values of ϵ from the range $[0, 1]$ and compute the empirical average of the corresponding SAP values. Hence,

$$\widehat{ASI}(M_\theta) = \frac{1}{K} \sum_{\epsilon_i \sim U[0,1], i=1}^{K} \mathcal{S}(M_\theta, \epsilon_i). \tag{2}$$

Since a benign model is likely to be less sensitive to adversarial attacks, it can be expected to have a lower $\widehat{ASI}$ value compared to poisoned models. Hence, the target model can be categorized as $trojan$ if its $\widehat{ASI}$ is much larger than that of the reference model. If computational complexity is not a constraint, multiple reference models can be trained and their $\widehat{ASI}$ values can be estimated. The maximum $\widehat{ASI}$ value among these multiple reference models serves as a good choice of the decision threshold τ. Table 1 shows the accuracy (ACC) of poisoned model detection in the white-box setting with PreActResNet18 as the reference model.

B. Additional Details on Experimental Setup

B.1. Datasets

We use three standard datasets, namely, MNIST [4], CIFAR10 [7], and GTSRB (German Traffic Sign Recognition Benchmark) [15] to evaluate the performance of the proposed method. The MNIST dataset contains 70,000 28×28 images of 10 handwritten digits, with 60,000 samples used for training and 10,000 samples for testing. The CIFAR10 dataset consists of 60,000 32×32 RGB images from 10 classes, with 50,000 images for training and 10,000 images for testing. GTSRB is a dataset of traffic sign images containing around 40,000 images from German roads depicting 43 traffic sign classes, with 26,640 images for training and 12,630 images for testing. To demonstrate effectiveness on real-world datasets, we also consider a binary classification task (Normal vs Tuberculosis) based on a Chest X-ray dataset [13] consisting of 7,000 chest X-ray images of size $224 \times 224 \times 3$. To evaluate on to larger datasets we utilize ImageNet dataset [3].

B.2. Target and Reference Model Training

In Table 2, we report the architecture of the target models used in our evaluation along with their average clean accuracy (ACC) and average attack success rate (ASR). Note that according to our threat model, the target models are poisoned by the attacker such that they perform naturally on clean images and behave maliciously only in the presence of the trigger. Hence, these target models are expected to have high clean accuracy as well as a high attack success rate. Please note that target models based on LC, SIG, and ISS poisoning attacks are trained using Backdoorbench [18]. The hyperparameters involved in target model training are summarized in Table 3. Examples of generated poisoned images included in the Modify and Blend poisoned subset are shown in Figure 1.

For the three architectures, the reference model is trained for 100 epochs. The optimizer used is SGD, with a learning rate

Table 1. Detection accuracy in the white-box setting on MNIST, CIFAR10, and GTSRB datasets.

Metric (%)	MNIST				CIFAR10							GTSRB						
	Modify	Blend	WaNet	IAD	Modify	Blend	WaNet	IAD	LC	SIG	ISS	Modify	Blend	WaNet	IAD	LC	SIG	ISS
ACC (↑)	99	100	100	100	86.4	87.17	99	92	98.5	100	100	99.2	99.4	98.87	89	90	99.3	100

set to $1e-2$. The momentum is set to 0.9, and weight decay is applied with a value of $5e-4$. A learning rate scheduler is employed to adjust the learning rate during training, reducing it by a factor of 0.1 at epochs 100, 200, 300, and 400. Since the reference model is trained on only 2% of the clean data, it is not expected to have high clean accuracy on test data but achieves 100% accuracy on its training data.

Table 2. Target Model Clean Accuracy(ACC), Architecture and Attack Success Rate (ASR).

Dataset	Target Model			
	Architecture	Attack	ACC (%)	ASR (%)
MNIST	2CONV+2FC	Modify	95.4	99.6
		Blend	96.5	99.6
	3CONV+2FC	WaNet	99.7	99.2
		IAD	99.7	99.0
CIFAR10	PreActResNet18	Modify	81.4	99.9
		Blend	84.7	97.5
		WaNet	84.4	98.4
		IAD	84.2	97.7
		LC	84.5	99.9
		SIG	84.4	98.3
		ISS	92.9	97.9
GTSRB	PreActResNet18	Modify	96.4	100
		Blend	99.9	99.7
		WaNet	95.8	92.9
		IAD	99.2	97.6
		LC	97.8	65.5
		SIG	98.55	70
		ISS	98.0	99.6

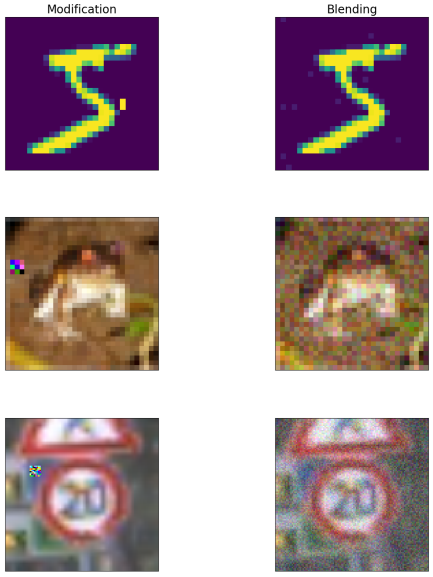

Figure 1. Examples of poisoned images generated using Modify (Left) and Blend (Right) attacks. The top row shows poisoned images from the MNIST dataset with "5" as the true class label. The assigned target labels are "2" and "3" for the M and B attacks, respectively. The middle row shows poisoned images from the CIFAR10 dataset, where the true label is "frog". The assigned target labels are "ship" and "dog" for the M and B attacks, respectively. The bottom row shows poisoned images from the GTSRB dataset, where the true label is "speed limit (20km/h)". The target labels are "speed limit(80km/h)" and "no vehicles sign" for the M and B attacks, respectively.

B.3. Poisoning Attacks

In this section, we will elaborate on the poisoning attacks used in our evaluation and their implementation details.

Modify: The attacker selects specific training examples and alters them by converting some pixels to a trigger pattern, assigns the intended label, and re-inserts the changed samples with their corresponding labels into the data set. An example of this type of attack is BadNets [6]. The trigger is intended to be unobtrusive and negligible to achieve high accuracy on clean images. However, during the inference stage, if a trigger is added to an input image, the model will produce a predicted class associated with the trigger, without considering the original content of the image.

Blend: The process of integrating a specific trigger pattern into the original input, so that the resulting input would contain the trigger information while still being similar to the original input. [2]. The trigger pattern is typically the same size as the input image. The Blended Injection technique involves a process of generating instances of poisoned images by merging a benign input image with a key pattern using a specific pattern-injection function. This function has a hyperparameter α that ranges from 0 to 1 and determines the blend ratio of the key pattern and the input instance. The key pattern k can be any image and is typically chosen arbitrarily. We randomly sample α from range $[0.05, 0.2]$.

WaNet: This method uses image warping techniques, specifically an elastic warping operation, to embed a trigger pattern

Table 3. Target Model Training Hyperparameters

Dataset	Attack	Hyperparameter			
		Optimizer	LR	Batch Size	Epochs
MNIST	Modify, Blend	Adam	0.001	100	100
	WaNet, IAD	SGD	0.01	128	100
CIFAR10, GTSRB	Modify, Blend, WaNet, IAD	SGD	0.01	128	100

into the training images in such a way that it is difficult to detect by humans [12]. The creation of poisoned images through the use of a warping field involves multiple steps, which can be delineated as follows: start by creating a random noise image of the same dimensions as the original image. Afterward, choose a group of reference points on both the initial picture and the noisy image. Upsample the control points to cover the entire image using a bicubic interpolation method. Clip the displacement vector at a predetermined threshold value to limit the magnitude of the distortion applied to the original image. Unlike other types of poisoning attacks, this method requires modification to the training process of the poisoned model. This method is stealthy and can evade many existing defenses for poisoning attacks. We used the same attack settings in the paper with a warping strength of 0.5 and a grid size of 4.

IAD: This attack uses a trained generator network to create poisoned triggers. The trigger generator is trained to ensure that a unique trigger is assigned to each input image, preventing multiple images from sharing the same trigger. The IA attacks can be challenging to detect as the triggers are not just a fixed pattern as in other types of poisoning attacks, but rather a dynamic function of the input. This method also requires modification to the training process of the poisoned model. Following the original paper [11], we use the SGD optimizer for training the poisoned model, and the Adam optimizer for training the generator with the same learning rate of 0.01. This rate drops by a factor of 10 after every 100 epochs. The networks are jointly trained until convergence. The backdoor probability and the cross-trigger probability are set to 0.1.

LC: This attack poisons the datasets without manipulating the label of poisoning samples so that the attack is more stealthy [16]. This approach involves modifying each image prior to adding the backdoor pattern, with the objective of making the classification task more difficult for the model using the original features of the image. This makes the backdoor pattern a distinguishing feature. We used the Backdoorbench [18] to train these attack models. Following the original settings in [16], we used PGD to construct adversarial examples, set the poisoning rate as 10%, and assign a random target label.

SIG: This backdoor attack involves using a sinusoidal signal as a trigger to distort images of the target class without altering their labels, resulting in a backdoor attack that is consistent with the original labels [1]. The authors suggest a type of backdoor attack that does not require the target samples' labels to be poisoned. Instead, the attacker injects a backdoor signal v into a specific number of training samples that fall under the intended class. We used the Backdoorbench [18] to train these attack models and the frequency and the delta of the backdoor signal in our experiments are 6 and 40, respectively.

ISS: The authors in [8] employ a technique where they create hidden additive noises as triggers. These triggers are generated by encoding a specific string provided by the attacker into benign images using an encoder-decoder network. The process involves using a generative model to convert the attacker's string into a small image, which is then seamlessly incorporated into the original training samples as an invisible perturbation. During the training phase, models are trained on the manipulated dataset, allowing the mapping from the encoded string to the target label to be established. We used the Backdoorbench [18] to train these attack models.

B.4. Computational Efficiency

Our experiments were carried out on a machine equipped with 80 CPUs and eight NVIDIA Quadro RTX 4000 GPUs. While reference model training typically takes less than 3 minutes, adversarial sensitivity evaluation requires around 10 minutes. Thus, the total training time is around 13 minutes, which is significantly less than the 14 hours required for MNTD training on the same machine. Adversarial probing of target models is efficient and takes only a few seconds. In contrast, though FRE does not involve any training, the detection step takes around an hour, due to the need for scanning all the labels in a dataset.

To ensure transparency and fairness in comparing the methods used for benchmarking, we provide a comprehensive overview of the settings used for training each method in Table 4.

Table 4. Summary of experimental settings for the methods used in benchmarking.

Method	Access to Target Model	Clean Data Requirements	Experimental setup used in our study for benchmarking	Computational Complexity
MNTD	Black-box	2%	1024 benign 1024 Poisoned	$\sim$ 14 hours models training + $\sim$ 150 seconds to train the classifier
ULP	Black-box	100%	500 benign 500 Poisoned	$\sim$ 7 hours + $\sim$ 699 seconds to train the classifier and learn the ULPs
NC	White-box	50% of the testing set	Experimental settings used in the original paper	CIFAR10: 511 seconds GTSRB: 1,616 seconds
FeatureRE	White-box	1% for CIFAR-10 16% for GTSRB	Experimental settings used in the original paper	CIFAR10: 768 seconds GTSRB: 9,372 seconds
Cassandra	White-box	100 clean image samples	240 benign 240 poisoned	$\sim$ 4 hours + $\sim$ 10 minutes to generate UAPs for each model

C. Additional Experimental Results

C.1. Impact of Adversarial Attack Type

When FGSM [5] is used (instead of PGD [10]) to generate adversarial examples, the A2D framework is still able to detect poisoned models, albeit with a significantly lower detection accuracy ($\approx 10\%$ drop for CIFAR-10 and $\approx 4\%$ for GTSRB) as shown in Table 5. Since FGSM is a single-step attack, the adversarial sensitivity evaluation is more computationally efficient with FGSM. However, if detection accuracy is critical, stronger attacks such as PGD, which is the most powerful adversary given first-order information about the network [10], should be employed.

Table 5. Average detection accuracy (ACC %) based on FGSM.

Metric (%)	**CIFAR10**		**GTSRB**	
	Modify	Blend	Modify	Blend
ACC (↑)	80	77	92	94
TDR (↑)	84	78	86	90
FDR (↓)	24	24	2	2

C.2. Impact of Sample Sizes

We investigate the impact of various sample size parameters on the A2D method. First, we consider the **size of the clean set used to train the reference model** and the **number of samples used for adversarial sensitivity evaluation**. We varied the fraction of the training set used as a clean set 2% or 5% on CIFAR10 and GTSRB and found that the clean set size did not have a significant impact on the detection accuracy and ϵ_{min} values. However, the number of samples used for the adversarial attack did have some impact on the detection accuracy. Specifically, a minimum of 100 attack samples is required to achieve reliable poisoned model detection when the reference model is trained on 2% training samples. With a 5% fraction, 50 attack samples prove to be adequate. However, when maintaining the same fraction at 5% and utilizing 200 attack samples, a slight decrease in detection accuracy is observed. We attribute this decline to the requirement for larger epsilon values for the reference model in such instances. The estimated perturbation bound and detection accuracy for the GTSRB and CIFAR10 datasets on two fractions using variations of the number of samples are presented in Table 6 in addition to the average detection accuracy of poisoned models on the seven attacks.

Additionally, we explored the sensitivity of the ϵ_{min} by adjusting the margin parameter ω. The purpose was to observe if stopping when the SAP value is below 1 would significantly alter ϵ_{min} and impact detection accuracy. We find that ϵ_{min} should be taken when the SAP value is above 0.9 as some attacks will be inseparable from benign models. Table 7 shows the overall accuracy on all attacks with different values of ϵ_{min}.

Table 6. Estimated perturbation bound for GTSRB and CIFAR10 datasets on two fractions of the clean subset using variations of the number of attack samples and average detection accuracy with different numbers of samples.

Dataset	Fraction	Number of Samples					
		50		100		200	
		ϵ_{min}	ACC(%)	ϵ_{min}	ACC(%)	ϵ_{min}	ACC(%)
CIFAR10	0.02	0.9	93.3	0.9	95	0.9	94.5
	0.05	0.9	91	0.9	94.4	0.9	93.2
GTSRB	0.02	0.3	77.5	0.5	96.9	0.6	98
	0.05	0.5	96	0.6	97	0.5	95.3

Table 7. Estimated perturbation bound for GTSRB and CIFAR10 datasets with varying margin parameter ω.

Dataset	ω					
	0.01		0.1		0.2	
	ϵ_{min}	ACC(%)	ϵ_{min}	ACC(%)	ϵ_{min}	ACC(%)
CIFAR10	0.9	94.5	0.5	93.6	0.2	89.6
GTSRB	0.4	96.5	0.2	94	0.006	85.1

C.3. Impact of Trigger Properties

We tested the effect of **trigger transparency** (visibility) on the detectability of poisoned models. Specifically, we trained poisoned models with varying levels of trigger transparency by modifying the blending ratio of the trigger pattern α from 0.05 to 1. When the blending ratio is small, the trigger pattern is almost invisible, while when it is large, the trigger pattern completely replaces the original pixels and becomes completely visible. We observed that increasing the visibility of the trigger pattern led to a decrease in the SAP values of the poisoned models. This suggests that a more visible trigger resulted in reduced susceptibility of the model to adversarial attacks. However, it is important to note that even with decreased SAP values, the poisoned models still fell within the detectability range. It is worth noting that the current landscape of poisoning attacks primarily focuses on concealing the trigger within the input data. The aim is to make the trigger as inconspicuous as possible, rendering it difficult to detect visually or through traditional analysis techniques. We conducted an additional ablation study to investigate the impact of the **spatial size of the trigger** on the detectability of poisoned models. Specifically, we trained poisoned models using triggers of various sizes, including 2x2, 3x3, 4x4, 5x5, and 10x10 pixels, on both the GTSRB and CIFAR datasets. We aimed to determine whether the size of the trigger had any noticeable effect on the resulting SAP values. We found that the spatial size of the trigger had no significant impact on the detectability of poisoned models. Results are shown in Figures 2. Note that when the trigger transparency (α) is small, the trigger pattern is almost invisible. In contrast, when α is large, the trigger pattern completely replaces the original pixels and becomes completely visible. All the poisoned models were accurately detected by our proposed method and their SAP value was larger than the sensitivity bound of the Benign models (with the same architecture). Moreover, no consistent trend in the SAP values of the poisoned models was observed as the trigger transparency increased. Similarly, the trigger sizes also did not have any impact on the detection accuracy. These results demonstrate that the proposed approach is robust against variations in trigger properties.

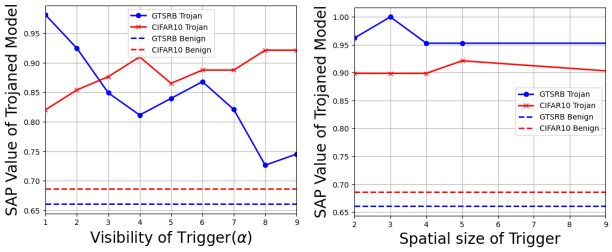

Figure 2. SAP values on GTSRB and CIFAR10 models poisoned using different trigger transparency settings used in Blend attack(UP), and using different trigger size settings used in Blend attack(down).

C.4. Impact of Independent Clean Set

In the main paper, all the experiments were based on the assumption that the clean set is a small subset (2%) of the training set. In Table 8, we investigated the performance of our method by training and attacking the reference model using a clean set derived from the test partition of the corresponding dataset. Note that these samples can be considered as independent of the training set, but drawn from the same distribution. Moreover, these samples were not seen by the target models during training. This experiment allowed us to assess the robustness of our method in a more challenging scenario where no samples from the original training set are available to the defender. We followed the same settings as in the previous experiments, using a fraction of 2% of the test dataset for training the reference model. This model was then subjected to adversarial attacks. The experiments yielded close results to those obtained using the training set. This finding suggests that our approach maintains its effectiveness and reliability even when the reference model is trained and attacked with previously unseen data by the target models.

C.5. Detection Performance on Adversarially Trained Models

In addition to the main experiment, Table 9 presents evidence that even when adversarial training is conducted before poisoning attacks in separate stages, it does not mitigate the

Table 8. Detection accuracy(ACC%) comparison on GTSRB and CIFAR10 when training and attacking reference model is performed using original training data (subset) and data not seen by target model (independent).

Dataset	Data	Attack						
		M	B	WaNet	IAD	LC	SIG	ISS
CIFAR10	Subset	93	86.6	100	95	95	100	100
	Independent	85	82	100	95	95	100	100
GTSRB	Subset	96.8	97.4	100	100	95	95.5	100
	Independent	93	96	100	100	80	92.5	100

sensitivity of the poisoned models to adversarial attacks. Before introducing the poisoning attack, these models were trained using PGD with perturbation bound = 0.0314 for 100 epochs. This observation highlights the persistent vulnerability of the models to poisoned attacks despite prior adversarial training.

Table 9. Detection performance on adversarially trained poisoned models.

Metric(%)	MNIST		CIFAR10		GTSRB	
	M	B	M	B	M	B
ACC(↑)	99	98	90	87	93	93
TDR(↑)	98	96	96	90	100	100
FDR(↓)	0	0	16	16	14	14

C.6. Using A2D in conjunction with NC

A2D can be integrated with neural cleanse (NC) to enhance model purification. A2D alone identifies poisoned models but doesn't purify them. However, when combined with purification methods like NC, it improves efficacy. The suggestion is to replace the anomaly index-based detection step in NC with A2D. Experiments on CIFAR10 demonstrated that while NC initially misclassified poisoned models as benign, A2D correctly identified them. The "cleansing" process led to a 10% drop in clean accuracy, but the Attack Success Rate (ASR) significantly reduced from over 90% to less than 10%. This highlights the effectiveness of A2D, particularly when rejecting a model is not a viable option. Results in 10 illustrate the average clean accuracy and ASR of 9 poisoned models on the CIFAR10 dataset, employing PREACTResNet-18, ResNet, and VGG architectures, along with SIG, LC, and ISS attacks before and after cleansing.

C.7. Scalability of A2D Framework to Large Datasets and Stealthier Attacks

To evaluate if the proposed approach is scalable to larger datasets and stealthier hidden attacks, we experiment on 20 target (10 benign + 10 poisoned) models with vision transformer (ViT_B_16) architecture trained on ImageNet [3]. A Blend attack with a random poisoning rate was employed for generating the poisoned models. Using ResNet18 as the reference model, all the poisoned ViT models exhibited SAP ($\hat{\mathcal{S}}^*$) values above 0.7, while the benign models had much lower $\hat{\mathcal{S}}^*$ values. Additionally, We evaluated the A2D method on 8 (4 benign + 4 poisoned) target (ResNet18) models trained on CIFAR-10 and subjected to the Sleeper Agent attack[14]. Using PREACTRESNET18 as reference model architecture, the SAP values for the poisoned models were greater than 0.7 and those of the benign models were less than 0.7. This indicates that the proposed approach is scalable to larger datasets and models and can detect poisoned models with Sleeper Agent attack. Furthermore, we have benchmarked our A2D method against ABS [9]. As shown in the Table below, the proposed A2D approach has higher detection accuracy (ACC) under most poisoning attacks.

Table 10. Impact of using A2D-based poisoned model detection in conjunction with purification methods like NC [17].

Before Cleansing		**After Cleansing**	
ACC (%) (↑)	ASR (%) (↓)	ACC (%) (↑)	ASR (%) (↓)
97.09	94.28	85.83	9.6

	CIFAR10						
	Modify	Blend	WaNet	IAD	LC	SIG	ISS
ABS	89	**87**	62.5	50	50	50	50
A2D	**93**	86.6	**100**	**95**	**95**	**100**	**100**
	GTSRB						
ABS	90	75	80	50	50	50	50
A2D	**96.8**	**97.4**	**100**	**100**	**95**	**95.5**	**100**

References

[1] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *IEEE International Conference on Image Processing (ICIP)*, pages 101–105. IEEE, 2019.

[2] 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.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 248–255. IEEE, 2009.

[4] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[5] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[8] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *In IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 16463–16472, 2021.

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

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations (ICLR)*, 2018.

[11] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:3454–3464, 2020.

[12] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations (ICLR)*, 2021.

[13] Tawsifur Rahman, Amith Khandakar, Muhammad Abdul Kadir, Khandaker Rejaul Islam, Khandakar F Islam, Rashid Mazhar, Tahir Hamid, Mohammad Tariqul Islam, Saad Kashem, Zaid Bin Mahbub, et al. Reliable tuberculosis detection using chest x-ray with deep learning, segmentation and visualization. *IEEE Access*, 8:191586–191601, 2020.

[14] Hossein Souri, Liam Fowl, Rama Chellappa, Micah Goldblum, and Tom Goldstein. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:19165–19178, 2022.

[15] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *International Joint Conference on Neural Networks (IJCNN)*, pages 1453–1460. IEEE, 2011.

[16] Alexander Turner, Dimitris Tsipras, and Aleksander Madry. Label-consistent backdoor attacks. *arXiv preprint arXiv:1912.02771*, 2019.

[17] 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 (S&P)*, pages 707–723. IEEE, 2019.

[18] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. *Advances in Neural Information Processing Systems*, 35:10546–10559, 2022.