Defending Against Backdoor Attacks by Layer-wise Feature Analysis (Extended Abstract)

Najeeb Moharram Jebreel[1] , **Josep Domingo-Ferrer**[1] , **Yiming Li**[2]
[1] Universitat Rovira i Virgili, Tarragona, Catalonia
[2] The State Key Laboratory of Blockchain and Data Security, Zhejiang University, China
{najeeb.jebreel, josep.domingo}@urv.cat, li-ym@zju.edu.cn

Abstract

Training deep neural networks (DNNs) usually requires massive training data and computational resources. Users who cannot afford this may prefer to outsource training to a third party or resort to publicly available pre-trained models. Unfortunately, doing so facilitates a new training-time attack (*i.e.*, backdoor attack) against DNNs. This attack aims to induce misclassification of input samples containing adversary-specified trigger patterns. In this paper, we first conduct a layer-wise feature analysis of poisoned and benign samples from the target class. We find out that the feature difference between benign and poisoned samples tends to be maximum at a critical layer, which is *not always* the one typically used in existing defenses, namely the layer before fully-connected layers. We also demonstrate how to locate this critical layer based on the behaviors of benign samples. We then propose a simple yet effective method to filter poisoned samples by analyzing the feature differences between suspicious and benign samples at the critical layer. Extensive experiments on two benchmark datasets are reported which confirm the effectiveness of our defense.

1 Introduction

In recent years, deep neural networks (DNNs) have successfully been applied in many tasks, such as computer vision, natural language processing, and speech recognition. However, training DNNs requires massive training data and computational resources, and users who cannot afford it may opt to outsource training to a third-party (*e.g.*, a cloud service) or leverage pre-trained DNNs. Unfortunately, losing control over training facilitates *backdoor attacks* [Chen *et al.*, 2017; Gu *et al.*, 2019; Li *et al.*, 2022] against DNNs. In these attacks, the adversary poisons a few training samples to cause the DNN to misclassify samples containing predefined trigger patterns into an adversary-specified target class. Nevertheless, the attacked models behave normally on benign samples, which makes the attack stealthy. Since DNNs are used in many mission-critical tasks (*e.g.*, autonomous driving, or facial recognition), it is urgent to design effective defenses against these attacks.

Among all backdoor defenses in the literature, backdoor detection is one of the most important paradigms, where defenders attempt to detect whether a suspicious object (*e.g.*, model or sample) is malicious. Currently, most existing backdoor detectors assume poisoned samples have different feature representations from benign samples, and they tend to focus on the layer before the fully connected layers [Chen *et al.*, 2019; Tang *et al.*, 2021; Hayase and Kong, 2021]. See the **supplementary materials** or [Jebreel *et al.*, 2023] for a review of related work on backdoor attacks and defenses. Two intriguing questions arise: **(1)** *Is this layer always the most critical place for backdoor detection?* **(2)** *If not, how to find the critical layer for designing more effective backdoor detection?*

In this paper[1], we give a negative answer to the first question (see Figure 1). To answer the second one, we conduct a layer-wise feature analysis of poisoned and benign samples from the target class. We find out that the feature difference between benign and poisoned samples tends to reach the maximum at a critical layer, which can be easily located based on the behaviors of benign samples. Specifically, *the critical layer is the one or near the one that contributes most to assigning benign samples to their true class.* Based on this finding, we propose a simple yet effective method to filter poisoned samples by analyzing the feature differences (measured by cosine similarity) between incoming suspicious samples and a few benign samples at the critical layer. Our method can serve as a 'firewall' for deployed DNNs to identify, block, and trace malicious inputs. In short, our main contributions are four-fold. **(1)** We demonstrate that the features of poisoned and benign samples are not always clearly separable at the layer before fully connected layers, which is the one typically used in existing defenses. **(2)** We conduct a layer-wise feature analysis aimed at locating the critical layer where the separation between poisoned and benign samples is neatest. **(3)** We propose a backdoor detection method to filter poisoned samples by analyzing the feature differences between suspicious and benign samples at the critical layer. **(4)** We conduct extensive experiments on two benchmark datasets to assess the effectiveness of our proposed defense.

[1] This is an extended abstract of [Jebreel *et al.*, 2023].

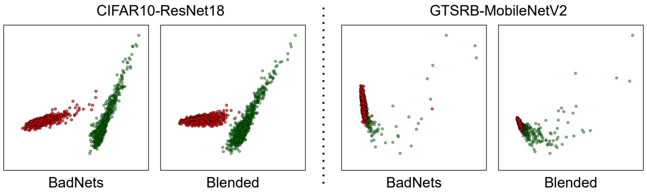


Figure 1: PCA-based visualization of features of benign (green) and poisoned samples (red) generated by the layer before the fully connected layers of models attacked by BadNets and Blended.Features of poisoned and benign samples are not well separated on the GTSRB benchmark.

2 Layer-wise Feature Analysis

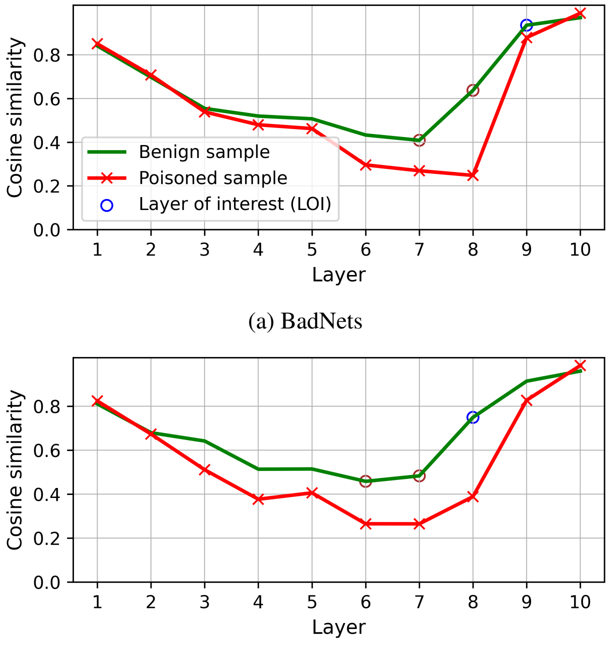
We notice that the predictions of attacked DNNs for both benign samples from the target class and poisoned samples are all the target label. The attacked DNNs mainly exploit class-relevant features to predict these benign samples while they use trigger-related features for poisoned samples. We suggest that defenders could exploit this difference to design effective backdoor detection. To explore their main differences, we conduct a layer-wise analysis, as follows.

Definition 1 (Layer-wise centroids of target class features). *Let f' be an attacked DNN with a target class t. Let $X_t = \{\boldsymbol{x_i}\}_{i=1}^{|X_t|}$ be benign samples with true class t, and let $\{\boldsymbol{a_i^1}, \dots, \boldsymbol{a_i^L}\}_{i=1}^{|X_t|}$ be their intermediate features generated by f'. The centroid of t's benign features at layer l is defined as $\hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{l}} = \frac{1}{|X_t|}\sum_{i=1}^{|X_t|} \boldsymbol{a}_{\boldsymbol{i}}^{\boldsymbol{l}}$, and $\{\hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{1}}, \dots, \hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{L}}\}$ is the set of layer-wise centroids of t's benign features.*

Definition 2 (Layer-wise cosine similarity). *Let $\boldsymbol{a}_{\boldsymbol{j}}^{\boldsymbol{l}}$ be the features generated by layer l for an input $\boldsymbol{x_j}$, and let cs_j^l be the cosine similarity between $\boldsymbol{a}_{\boldsymbol{j}}^{\boldsymbol{l}}$ and the corresponding t's centroid $\hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{l}}$. The set $\{cs_j^1, \dots, cs_j^L\}$ is said to be the layer-wise cosine similarities between $\boldsymbol{x_j}$ and t's centroids.*

Settings. We conducted six representative attacks on four classical benchmarks: CIFAR10-ResNet18, CIFAR10-MobileNetV2, GTSRB-ResNet18, and GTSRB-MobileNetV2. The six attacks were BadNets [Gu *et al.*, 2019], the backdoor attack with blended strategy (Blended) [Chen *et al.*, 2017], the label-consistent attack (LC) of [Turner *et al.*, 2019], WaNet [Nguyen and Tran, 2020b], ISSBA [Li *et al.*, 2021b], and IAD [Nguyen and Tran, 2020a]. More details on the datasets, DNNs, and attack settings are presented in the **supplementary materials**. Specifically, for each attacked DNN f' with a target class t, we estimated $\{\hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{1}}, \dots, \hat{\boldsymbol{a}}_{\boldsymbol{t}}^{\boldsymbol{L}}\}$ using 10% of the benign test samples labeled as t. Then, for the benign and poisoned test samples classified by f' into t, we calculated the layer-wise cosine similarities between their generated features and the corresponding estimated centroids. Finally, we visualized the layer-wise means of the computed cosine similarities of the benign and poisoned samples to analyze their behaviors.

Results. Figure 2 shows the layer-wise means of cosine similarity for benign and poisoned samples with the CIFAR10-ResNet18 benchmark under the BadNets and ISSBA attacks.



(a) BadNets

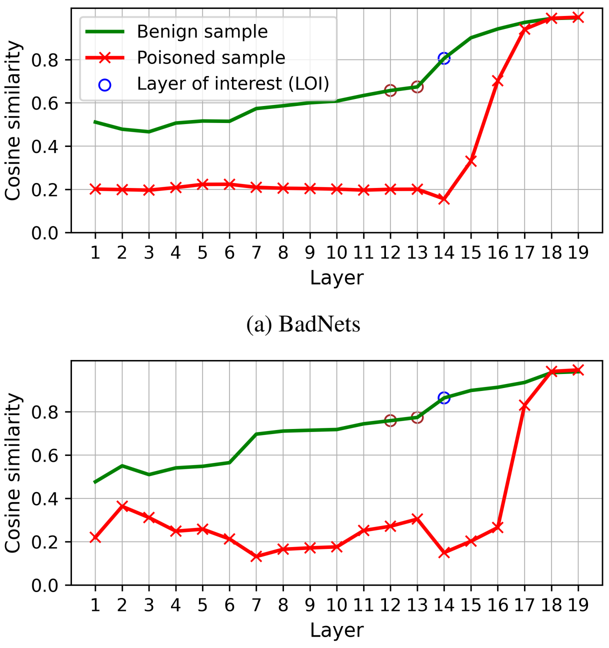
(b) ISSBA

Figure 2: Layer-wise behaviors of benign samples from the target class and poisoned samples (generated by BadNets and ISSBA) on CIFAR-10 with ResNet-18

As we go deeper into the attacked DNN layers, the gap between the direction of benign and poisoned features gets larger until we reach a specific layer where the backdoor trigger is activated, causing poisoned samples to get closer to the target class. Figure 3 shows the same phenomenon for the GTSRB-MobileNetV2 benchmark. Further, we can see that for BadNets the latent features of benign and poisoned samples are similar in the last layer of the features extractor (*i.e.*, layer 17).

Regardless of the attack or benchmark, when we enter the second half of DNN layers (which usually are class-specific), *benign samples start to get closer to the target class before the poisoned ones, that are still farther from the target class* because the backdoor trigger is not yet activated. This makes the difference in similarity maximum in one of those latter layers, which we call the *critical layer*. In particular, *this layer is not always the one typically used in existing defenses* (*i.e.*, the layer before fully-connected layers). Besides, we show that it is very likely to be either the layer that contributes most to assigning the benign samples to their true target class (which we name the *layer of interest or LOI*, circled in blue) or one of the two layers before the LOI (circled in brown).

Results under other attacks for these benchmarks are presented in the **supplementary materials**. In those materials, we also provide confirmation that the above distinctive behaviors hold regardless of the datasets or models being used. From the analysis above, we can conclude that focusing on those circled layers can help develop a simple and robust defense against backdoor attacks.



(a) BadNets

(b) ISSBA

Figure 3: Layer-wise behaviors of benign samples from the target class and poisoned samples (generated by BadNets and ISSBA) on GTSRB with MobileNetV2

3 The Proposed Defense

Threat Model. Consider a user that obtains a suspicious trained f_s that might contain hidden backdoors. We assume that the user has limited computational resources or benign samples, and therefore cannot repair f_s. The user wants to defend by detecting at inference time whether a suspicious incoming input $\boldsymbol{x_s}$ is poisoned, given f_s. Similar to existing defenses, we assume that a small set of benign samples X_{val} is available to the user/defender. We denote the available samples that belong to a potential class t as $X_{t_{val}}$. Let $m = |X_{t_{val}}|$ denote the number of available samples labeled as t.

Method Design. Based on the lessons learned in Section 2, our method to detect poisoned samples at inference time consists of four steps. **1)** Estimate the layer-wise features' centroids of class t for each of layers $\lfloor L/2 \rfloor$ to L using the class's available benign samples. **2)** Compute the cosine similarities between the extracted features and the estimated centroids, and then compute the layer-wise means of the computed cosine similarities. **3)** Identify the layer of interest (LOI) as per Algorithm 1, sum up the cosine similarities in LOI and the two layers before LOI (sample-wise), and compute the mean and standard deviation of the summed cosine similarities. **4)** For any suspicious incoming input $\boldsymbol{x_s}$ classified as t by f_s, **4.1)** compute its cosine similarities to the estimated centroids in the above-mentioned three layers, and **4.2)** consider it as a potentially poisoned input if its summed similarities fall below the obtained mean by a specific number τ of standard deviations (called threshold in what follows). A detailed pseudocode can be found in the **supplementary materials**.

Algorithm 1 Identify layer of interest (LOI).

```
Input: Cosine similarities $\{\hat{cs}_t^{\lfloor L/2 \rfloor}, \dots, \hat{cs}_t^{L}\}$ for potential target class $t$
$max_{diff} \leftarrow \hat{cs}_t^{\lfloor L/2 \rfloor + 1} - \hat{cs}_t^{\lfloor L/2 \rfloor}$; $LOI_t \leftarrow \lfloor L/2 \rfloor + 1$;
for $l \in \{\lfloor L/2 \rfloor + 2, \dots, L\}$ do
    $l_{diff} \leftarrow \hat{cs}_t^{l} - \hat{cs}_t^{l-1}$;
    if $l_{diff} > max_{diff}$ then
        $max_{diff} \leftarrow l_{diff}$; $LOI_t \leftarrow l$;
return $LOI_t$.
```

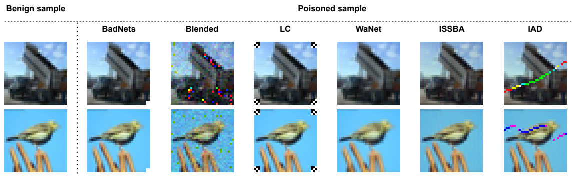


Figure 4: The example of benign samples and their poisoned versions generated by six representative backdoor attacks.

4 Experiments

The attacks considered were those mentioned in Section 2. Figure 4 shows an example of poisoned samples generated by different attacks. More details on the data sets, the DNNs, the attack and defense baselines, the attack and defense setup and the evaluation metrics can be found in the **supplementary materials** or [Jebreel *et al.*, 2023]. The source code, pre-trained models, and poisoned test sets of our defense are available at https://github.com/NajeebJebreel/DBALFA.

For each attack, we ran each defense five times for a fair comparison. Due to space limitations, we present the average TPR and FPR in this section. Please refer to our **supplementary materials** for more detailed results.

As shown in Tables 1 and 2, existing defenses — RS [Rosenfeld *et al.*, 2020], ShPd [Li *et al.*, 2021a], AC [Chen *et al.*, 2019], STRIP [Gao *et al.*, 2022], SCAn [Tang *et al.*, 2021], and FP [Liu *et al.*, 2018]— failed to detect attacks with low TPR or high FPR in many cases, especially on the GTSRB dataset. For example, AC failed in most cases on GTSRB, although it had promising performance on CIFAR-10. In contrast, our method had good performance in detecting all attacks on both datasets. There were only a few cases (4 over 28) where our approach was neither optimal nor close to optimal. In these cases, our detection was still on par with state-of-the-art methods, and another indicator (*i.e.*, TPR or FPR) was significantly better than them. For example, when defending against the blended attack on the GTSRB dataset, the TPR of our method was 69.44% larger than that of FP, which had the smallest FPR in this case. These results confirm the effectiveness of our detection.

Further, we analyzed the performance of attacks, the effects of the detection threshold, the effects of the poisoning rate, the effectiveness of our layer selection, and the resistance to adaptive attacks. Sample-specific attacks (*e.g.*,

Attack→	BadNets		Blended		LC		WaNet		ISSBA		IAD		Avg	
Metric→ Defense↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
RS	9.84	8.00	7.35	5.76	9.21	7.52	98.48	10.00	8.83	8.72	13.28	6.36	24.50	7.73
ShPd	94.28	13.31	49.72	12.89	69.87	13.18	36.25	17.69	95.22	5.50	42.74	7.56	64.68	11.69
FP	96.10	17.13	96.23	16.16	94.76	17.31	96.01	18.64	98.98	19.53	97.08	22.52	96.53	18.55
AC	**99.52**	31.14	**100.00**	30.69	**100.00**	31.16	**99.18**	32.44	**99.94**	34.22	82.99	31.32	96.94	31.83
STRIP	68.70	11.70	65.20	11.70	66.00	12.80	7.90	12.30	56.20	11.40	2.10	14.00	44.35	12.32
SCAn	96.60	**0.77**	**100.00**	**0.00**	0.02	5.05	98.55	**1.06**	99.89	2.61	84.19	**0.13**	79.88	1.60
Ours	99.38	1.35	**100.00**	1.59	**100.00**	**1.20**	91.04	1.48	98.97	**1.17**	**99.12**	1.26	**98.09**	**1.34**

Table 1: Main results (%) on the CIFAR-10 dataset. Boldfaced values are the best results among all defenses. Underlined values are the second-best results.

Attack→	BadNets		Blended		LC		WaNet		ISSBA		IAD		Avg	
Metric→ Defense↓	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR	TPR	FPR
RS	13.20	22.10	10.12	20.40	9.23	19.15	10.10	17.20	8.61	16.98	17.70	17.60	11.49	18.91
ShPd	94.97	12.16	11.58	10.68	96.16	10.60	66.11	14.81	95.92	8.26	31.07	16.10	65.97	12.10
FP	89.05	18.80	30.56	**3.70**	94.71	50.02	67.12	3.24	94.22	7.05	94.37	5.75	78.34	14.76
AC	0.30	8.84	0.00	5.67	4.83	**5.42**	0.42	25.87	99.06	17.48	43.85	10.73	24.74	12.34
STRIP	32.00	9.00	80.40	10.80	7.40	11.00	34.20	11.40	13.00	13.60	6.60	10.60	28.93	11.07
SCAn	46.05	**2.57**	46.02	4.03	30.45	11.39	54.07	**1.88**	96.85	**0.17**	0.09	19.41	45.59	6.58
Ours	**99.99**	6.23	**100.00**	6.72	**100.00**	5.95	**100.00**	6.49	**100.00**	5.43	**100.00**	**4.67**	**100.00**	**5.92**

Table 2: Main results (%) on the GTSRB dataset. Boldfaced values are the best results among all defenses. Underlined values are the second-best results.

ISSBA and IAD) performed better than other attacks as to main accuracy (MA) and attack success rate (ASR). Regarding the threshold, we found that a threshold 2.5 is reasonable and offers a high TPR while keeping a low FPR. We also found that the ASR increased with the poisoning rate, but the poisoning rate had minor effects on our TPR and FPR, which confirms the effectiveness of our method. Interestingly, existing detection methods can also benefit from our LOI selection. Lastly, if the attacker adapted his attack to bypass our defense by minimizing the layer-wise angular deviation between poisoned and benign samples, ASR stayed similar as in the non-adaptive setting, FPR was almost unaffected, while TPR slightly decreased; importantly, MA decreased enough for the poisoned model to be rejected due to poor performance. For more details on the analyses described in this paragraph, see **supplementary materials** and [Jebreel *et al.*, 2023].

5 Conclusion

In this paper, we conducted a layer-wise feature analysis of the behavior of benign and poisoned samples generated by attacked DNNs. We found that the feature difference between benign and poisoned samples tends to reach the maximum at a critical layer, which can be easily located based on the behaviors of benign samples. Using this, we proposed a simple yet effective backdoor detection to determine whether a given suspicious testing sample is poisoned by analyzing the differences between its features and those of a few local benign samples. Our extensive experiments on benchmark datasets confirmed the effectiveness of our detection. We hope our work can provide a deeper understanding of attack mechanisms, to facilitate the design of more effective and efficient backdoor defenses and more secure DNNs.

Acknowledgments

We acknowledge funding by the European Commission (H2020-871042 "SoBigData++"), the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer and grant no. 2021 SGR 00115), MCIN/AEI/ 10.13039/501100011033 under grant PID2021-123637NB-I00, and INCIBE/EU NextGeneration (project "HERMES" and INCIBE-URV cybersecurity chair).

References

[Chen *et al.*, 2017] 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.

[Chen *et al.*, 2019] Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. In *AAAI Workshop*, 2019.

[Gao *et al.*, 2022] Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. Design and evaluation of a multi-domain Trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2349–2364, 2022.

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

[Hayase and Kong, 2021] Jonathan Hayase and Weihao Kong. Spectre: Defending against backdoor attacks using robust covariance estimation. In *ICML*, 2021.

[Jebreel *et al.*, 2023] Najeeb Jebreel, Josep Domingo-Ferrer, and Yiming Li. Defending against backdoor attacks by layer-wise feature analysis. In *The 26th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD 2023)*, pages 428–440. Springer, 2023. Best paper award.

[Li *et al.*, 2021a] Yiming Li, Tongqing Zhai, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor attack in the physical world. In *ICLR Workshop*, 2021.

[Li *et al.*, 2021b] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021.

[Li *et al.*, 2022] Yiming Li, Yong Jiang, Zhifeng Li, and Shu-Tao Xia. Backdoor learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[Liu *et al.*, 2018] Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

[Nguyen and Tran, 2020a] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

[Nguyen and Tran, 2020b] Tuan Anh Nguyen and Anh Tuan Tran. Wanet-imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2020.

[Rosenfeld *et al.*, 2020] Elan Rosenfeld, Ezra Winston, Pradeep Ravikumar, and Zico Kolter. Certified robustness to label-flipping attacks via randomized smoothing. In *ICML*, 2020.

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

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