LFPD: Local-Feature-Powered Defense Against Adaptive Backdoor Attacks

Wei Guo[1], Ambra Demontis[1], Maura Pintor[1], Patrick P.K. Chan[2], Battista Biggio[1]

[1]Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari 09123, Italy.
[2]Shien-Ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou 510006, China.
wei.guo.cn@outlook.com, ambra.demontis@unica.it, maura.pintor@unica.it, patrickchan@scut.edu.cn,
battista.biggio@unica.it

Abstract:

To detect the suspect poisoned data in the training phase, most backdoor defenses rely on a prevalent assumption, i.e., the feature separability between poisoned and benign samples. However, this assumption can be bypassed by novel adaptive attacks, which merge the features of poisoned and benign samples. In this paper, we contrast these adaptive attacks and propose a so-called Local-Feature-Powered Defense (LFPD), which leverages a local feature algorithm to measure samples' similarity in the image space and uses it to guide the training process to increase the feature separability between poisoned and benign samples. Then, our LFPD detects the outliers in the training dataset as poisoned samples and removes the backdoor by unlearning them. Finally, we compare our LFPD with five existing defenses, and our experimental results demonstrate that LFPD outperforms them in defending against adaptive attacks.

Keywords:

Backdoor defence, Adaptive attack, local feature

1 Introduction

Deep Neural Networks (DNNs) have shown remarkable efficacy across various domains, but meanwhile are vulnerable to many attacks during both the training and test time, e.g., adversarial examples [1] and poisoning attacks [2]. These vulnerabilities pose significant limitations on applications in security-sensitive fields like, autonomous vehicles and medical diagnostics. Among them, one of the most severe threats is the backdoor attack [3], which contaminates a fraction of the training dataset to induce the model to learn spurious correlations, which can be used for inducing malicious goals. During test time, the backdoored model behaves normally on regular data but performs malicious behavior (misclassification) when the hidden backdoor is activated by a specific trigger signal only known by the attacker.

Following the taxonomy [4], the existing defenses can be categorized into: sample-, model-, and training-level defenses, where the first two levels are applied after the model has been deployed in an operative environment[1]. In contrast, the training-level defender checks the training dataset to detect suspicious (poisoned) samples. This paper focuses on training-level defenses since it tries to mitigate backdoor threats from the root by removing the poisoned samples.

Most training-level defenses [5–8] rely on the *feature separability assumption*, i.e., the features of poisoned samples are distinguishable from the benign data. However, recently proposed adaptive attacks can break this assumption and bypass these defenses by merging the features of poisoned and benign samples. For instance, Tang et al. [9] proposed a targeted contamination attack (*TaCT*), which observes that the poisoned sample's representation becomes less dominated (less distinguishable) when the backdoor is *source-specific*, that is, only samples from a specific class with trigger will be misclassified. Moreover, to merge the features further, TaCT adds to the training dataset a set of *cover samples*, i.e., samples that share the same trigger as poisoned samples but are correctly labeled. Later, Qi et al. [10] expanded *TaCT* to propose a *source-agnostic* backdoor attack, where a sample from any classes with the trigger will be misclassified. The authors also exploit the asymmetric trigger strategies to increase the backdoor activation ratio, i.e., instead of using the same trigger as training time, the attacker exploits a stronger trigger at test time. The authors define two kinds of attacks: *Adap-blend*, with a HelloKitty picture as trigger, and *Adap-patch*, with the pixel patterns as trigger, as shown in Fig. 1.

Since adaptive attacks reduce the feature separability between poisoned and benign samples, most existing training-level defenses cannot detect these attacks. In this paper, we

[1]The sample-level defenders scrutinize the input test samples to determine whether the input is adversarial, while the model-level defenses inspect deployed model weights to check whether a backdoor has infected the model.

FIGURE 1. Trigger signal and poisoned sample of TaCT, Adap-blend, and Adap-patch

consider these adaptive attacks and propose a so-called Local-Feature-Powered Defense (LFPD), which increases the feature separability between poisoned and benign samples via corner-based local feature matching. With the help of the matching, we devise an image similarity metric to use as a regularizer for training. Since the poisoned and cover samples of adaptive attack share the same trigger (inducing more common features between them), our proposed regularization can guide the training process to ensure that poisoned images cluster closely in the feature space and meanwhile increase the feature separability between poisoned and benign samples on the feature space. We also empirically verify the effectiveness of our LFPD in defending against three adaptive attacks (i.e., *TaCT*, *Adap-blend*, and *Adap-patch*) and compare our results with those of five existing defenses. The results demonstrate that contrary to the other methods, our method can efficiently defend against adaptive backdoor attacks.

2 Related work on training-level defenses

Two of the most pioneering defenses based on the feature separability assumption are called Spectral Signature (SS) [5] and Activation Clustering (AC) [6], which analyze the feature distribution of training data and detect the anomalous samples as poisoned data via outlier detection or K-means clustering. However, these two methods have many limitations, such as high false positive ratio or only working in a specific range of poisoning ratios. To overcome these limitations, Cluster Impurity (CI) [7] and Clustering and Centroids Analysis (CCAUD) [8] utilized advanced clustering algorithm, i.e., GMM [7] and DBSCAN [8], to replace the K-means, and design more sophisticated poison cluster detection method to verify which cluster includes poisoned samples. Recently, Beatrix [11] exploited the Gram matrix to extract Gramian features from samples' representation and use it to detect the anomalies induced by the poisoned samples.

Different from the main research stream on the latent separability, we also consider two additional works: Feature Consistency towards Transformations (FCT) [12] and Scaled Prediction Consistency (SPC) [13] as a subbranch, because they analyze the different representation consistency between poisoned and benign samples when some transformations are applied over the inputs.

Except for the above-mentioned defenses based on latent separability assumption, there is another type of defense with assumption that the DNN model will first fit the poisoned data better than benign ones. The first work of this type is called Anti-Backdoor Learning (ABL) [14], which assumes that the model learns the backdoor samples faster than the benign ones at the beginning training stage. Based on this assumption, Qi et al. [15] proposed Confusion Learning (ConfusLearn), adding the randomly labeled benign samples into the poisoned training dataset. The randomly labeled samples will confuse the model learning the benign samples, whereas it will learn the poisoned samples better in this case. This model is then used to detect the poisoning samples. Recently, Chen et al. [16] proposed a so-called Progressive Isolation of Poisoned Data (PIPD), which improves the one-time isolation of ABL and designs a progressive method to improve the isolation accuracy.

3 Formalization and defense model

Formalization: let's assume the DNN model $f(\cdot)$ addresses the classification task, i.e., outputting the probability distribution (softmax) over all the classes $i = 1, 2, ..., l$, where l is the number of classes. The model $f(\cdot)$ generally consists of a backbone $\phi(\cdot)$ to extract features from input samples, and fully-connected layers to map the feature to the softmax. We also define the prediction of $f(\cdot)$ as $F(x) = \arg\max f(x)$. Following the supervised learning paradigm, at the training phase, $f(\cdot)$ is optimized over a training dataset $D_{tr} = \cup_{i=1}^{l} D_{tr,i}$, and then at the test time evaluated over a test dataset $D_{ts} = \cup_{i=1}^{l} D_{ts,i}$, where $D_{tr,i}$ and $D_{ts,i}$ represent the subset of training and test dataset with all samples labeled as the class i.

The backdoor attacker aims to inject a backdoor into the DNN model, so that the backdoored model can satisfy: 1) working normally on the benign data; and 2) misclassifying the input samples with trigger signal v to the target class t. To distinguish from the benign model $f(\cdot)$ and its prediction $F(\cdot)$, in the following, we use $\tilde{f}(\cdot)$ and $\tilde{F}(\cdot)$ to represent the backdoored model. As mentioned in Sec. 1, TaCT and Adap-blend/-patch belong to two kinds of attacks: source-specific and source-agnostic, which achieve the same normal classification $\forall(x, y) \in D_{ts/t}, \tilde{F}(x) = y$, but different backdoor behaviors:

$$\text{Source-specific} : \forall(x, y) \in D_{ts,s}, \tilde{F}(x \odot v) = t \quad (1)$$

$$\text{Source-agnostic} : \forall(x, y) \in D_{ts/t}, \tilde{F}(x \odot v) = t \quad (2)$$

where $\odot$ is an operation adding trigger over the input sample x, and $D_{ts,s}$ and $D_{ts/t}$ represent the test samples from the specific source class s and non-target classes, respectively. To inject the backdoor, the attacker corrupts the training dataset D_{tr} by replacing a subset of it (D_p) with its poisoned version $\tilde{D}_p = \{(x \odot v, t) | (x, y) \in D_p\}$.

Defense model: we assume that the trainer controls the training process to defend against backdoor attacks. There are *two levels of goals* for the defender to achieve: 1) detecting and filtering out the poisoned samples from the training dataset, and 2) training a model with good performance over benign data and meanwhile without the backdoor influence. As the trainer, the defender *knows all the details of the training*, including hyper-parameters, model weights, and training samples. Moreover, a small, clean set of validated data D_{val} is also considered an optional kind of knowledge. The defender *can access the trained model as a white box*, obtain the feature of all the training samples by querying $\phi(\cdot)$, and also scrutinize the whole training dataset to find out the poisoned data.

4 Proposed method

Our LFPD is composed of three steps: 1) local feature powered regularization (Sec. 4.1), exploiting an efficient local feature algorithm to measure the image similarity and utilizing it to guide the training process to ensure the features of similar images are close to each other; 2) outlier detection (Sec. 4.2), determining the poisoned data via Gaussian mixture model; and 3) backdoor unlearning (Sec. 4.3), finetuning the backdoored model to unlearn the backdoor behavior.

4.1 Local feature powered regularization

Since the poisoned and cover samples share the trigger, their similarity tends to be larger than that between benign images. Using similarity, we hope to cluster the feature of poisoned samples and meanwhile increase the feature separability between poisoned and benign data[2]. The similarity metric consists of two steps: *keypoint determination* using Shi-Tomasi corner detection [17] to find the keypoints, and *keypoint matching* performing the 'rate of closest to 2nd-closest neighbor' policy to calculate the similarity between two images.

Based on this similarity metric, we design a novel regularizer to be included in the training loss function[3]:

$$\sum_{i=1}^{|D_{tr}|} \mathcal{L}(f(x_i), y_i) + \beta \sum_{i,j=1}^{|D_{tr}|} s_{ij} \cdot d(\phi(x_i), \phi(x_j)), \quad (3)$$

where $\mathcal{L}(\cdot)$ is the cross entropy loss function, β is a hyper-parameter trading-off the strength of the regularization, $d(\cdot)$ is the Euclidean distance function, and s_{ij} is the value calculated by the proposed similarity metric between two samples x_i and x_j on the input space (more similar s_{ij} is larger). Specifically, the second term is to pull two images' features $f(x_i)$ and $f(x_j)$ close to each other if the two samples are similar in the input space.

Keypoint detection: In computer vision, corners are viewed as stable and robust keypoints. We exploits Shi-Tomasi method [17] to find a pixel as a corner if the shifted windows around the pixel are significantly different. We define the keypoint (corner) detection as $kp = corner(x)$, to find out the top-10 keypoints (corners with top high confidences).

Since the input of Shi-Tomasi method is a grey image, the input image x is required to be converted to a grey image g. Then, for each pixel located by coordinates (m^*, n^*) from g, we take its neighborhood as a patch window W (with (m^*, n^*) as the center position). The difference between the pixel and the shifted windows is defined as $\kappa(\Delta_m, \Delta_n)$ calculated by:

$$\sum_{(m,n) \in W} \Big(g(m, n) - g(m + \Delta_m, n + \Delta_n) \Big)^2, \quad (4)$$

where Δ_m and Δ_n are window shift pixels in the row or column direction. Given the partial derivatives ∂_m and ∂_n of g in row and column directions, Eq. (4) can be simplified via the Taylor expansion: $\kappa(\Delta_m, \Delta_n) \approx (\Delta_m, \Delta_n) M (\Delta_m, \Delta_n)^T$, where M is the structure tensor[4].

Based on the Shi-Tomasi method, the minimum of eigenvalues (λ_1, λ_2) of M positively correlates with the corner confidence. In our work, we choose 10 pixels with large confidential values as the keypoints, and the corresponding descriptions are the $l \times l$ image window with keypoints as centers.

Keypoint matching: After keypoint detection, we further perform the keypoint matching, and the matching number between two images is positively correlated with the image similarity. Assume the keypoints of x_i and x_j are kp_i and kp_j. Each of them has 10 coordinates indicating the keypoint locations. We also define their corresponding descriptions set (neighborhood around the each keypoint) as dp_i and dp_j, where there are 10 descriptions in each of them. For each keypoint in x_i, we calculate its description distance with all keypoints from x_j. There exists one matching if the ratio of closest to the 2nd-closest neighbor is less than 0.5 [17]. Finally, the similarity

[2] Since the cover samples are correctly labeled, the similarity between poisoned and cover samples will pull the poisoned samples out of the benign data

[3] Eq. (3) is applied in each batch instead of the whole training dataset.

[4] https://en.wikipedia.org/wiki/Structure_tensor

metric is defined as $S(x_i, x_j) = n_{match}10$, where n_{match} is the number of matching pairs.

Since different classes (e.g. dog and cat) perhaps share similar patterns, to avoid the regularization affecting the normal feature embedding[5], we expect to draw together the two features when the two images are super similar. To do it, we first estimate the maximum similarity score $\theta_{i,j}$ between benign samples of different classes from D_{val}. Then, if $s_{ij} = \mathcal{S}(x_i, x_j)$ is larger than $\mathcal{S}(x_i, x_j) > \theta_{y_i,y_j}$, the two samples are considered as super similar; otherwise, $s_{ij} = 0$.

4.2 Outlier detection

We exploit the multivariate Gaussian distribution to detect the samples that are out of the feature distribution of the validation dataset D_{val}. For each class i of the validation dataset, we calculate its mean feature vector μ_i and covariance matrix Σ_i. Then, for each training sample in $x_j \in D_{tr}$ and belonging to class i, the outlier score p_j can be calculated via

$$\frac{1}{(2\pi)^{n/2}|\Sigma_i|^{1/2}} \exp\Big(-\frac{1}{2}(\phi(x_j) - \mu_i)^T \Sigma_i^{-1} (\phi(x_j) - \mu_i)\Big). \tag{5}$$

A small p_j value indicates a high possibility that the sample x_j is an outlier (poisoned sample). We provide a threshold ξ^* (more details on calculating ξ^* are provided in Section 5.2). If $p_j < \xi^*$, x_j considered as poisoned samples, is added into the set of samples predicted as poisoned $\bar{D}_p$.

4.3 Backdoor unlearning

Finally, given the predicted poisoned set $\bar{D}_p$, to unlearn the backdoor behavior, we design a new unlearning algorithm to force the model to forget the (predicted) poisoned data $\bar{D}_p$. The loss function is defined as $\sum_i^{|D_{tr}/\bar{D}_p|} \mathcal{L}(f(x_i), y_i) - \sum_j^{|\bar{D}_p|} w_j \cdot \mathcal{L}(f(x_j), y_j)$, where w_j controls whether continuing to unlearn a poisoned sample (i.e., $w_j = 1$ when if $\arg\max f(x_j) = y_j$, otherwise $w_j = 0$).

5 Experimental analysis

5.1 Attack settings

We implement three attacks: TaCT, Adap-blend, and Adap-patch, which poison the Cifar10 dataset[6] with $\alpha_p = 0.003$ and $\alpha_c = 0.003$. For all three attacks, the target class is 'airplane'. As explained in Sec. 3, TaCT is a source-specific attack with 'car' as the source class. For each considered attack, the trigger signal and the poisoned samples are shown in Fig. 1. The architecture of the trained model is based on the EfficientNetV2 model.The input image is resized to $192 \times 192 \times 3$ [7] in our experiments. The model's parameters are trained for 100 epochs by the SGD optimizer with the learning rate 0.1, momentum 0.9, and weight decay 10^{-4}.

At test time, we evaluated the model's accuracy on the test data D_{ts}: $ACC = \frac{\sum_{i=1}^{|D_{ts}|} \mathbb{1}(F(x_i)=y_i)}{|D_{ts}|}$, where $\mathbb{1}$ is the indicator function outputting 1 if its inside condition is satisfied, otherwise outputting 0. To estimate the backdoor performance, i.e., whether the trigger can activate the hidden backdoor, we define the Attack Success Rate (ASR). Specifically, the definition of ASR is slightly different between source-specific (TaCT) and source-agnostic (Adap-blend and Adap-patch) attacks, as shown as follows:

$$ASR = \begin{cases} \frac{\sum_{i=1}^{|D_{ts,s}|} \mathbb{1}(F(x_i \odot v)=y_i)}{|D_{ts,s}|}, \text{ source-specific} \\ \frac{\sum_{i=1}^{|D_{ts/t}|} \mathbb{1}(F(x_i \odot v)=y_i)}{|D_{ts/t}|}, \text{ source-agnostic.} \end{cases} \tag{6}$$

The ACC and ASR for benign and poisoned models are shown in Tab. 1, where we can observe that the backdoored models have a similar ACC value as the benign model. It means that the backdoor does not affect the normal classification task. For Adap-blend/-patch attacks, ASR is estimated with the same trigger as the training phase, but their ASRs can be further improved with a stronger (more visible) triggers.

TABLE 1. ACC and ASR of being model and three types of Backdoor models

	Benign	TaCT	Adap-blend	Adap-patch
ACC	0.939	0.937	0.937	0.927
ASR	0.025	0.864	0.844	0.661

5.2 Defense settings

For a fair comparison of different defenses, based on the Receiver Operating Characteristic (ROC) curve, we utilize the same threshold strategy i.e., $\xi^* = \arg\max_\xi (TPR_\xi - FPR_\xi)$ to determine the True Positive Ratio TPR_{ξ^*} and False positive Ratio FPR_{ξ^*}. Moreover, to evaluate the unlearning process, we utilize the same two metrics: ACC and ASR as described in Sec. 5.1. However, for simplicity, in the following text, we use ACC* and ASR* to distinguish the unlearned model's performance from the backdoored one.

[5] When the regularizer is applied to the image pair with tiny similarity, it will draw the different classes together, which hurts the discrimination purpose of training.

[6] https://www.cs.toronto.edu/kriz/learning-features-2009-TR.pdf

[7] Compared with ResNet, the input size of EfficientNetV2 model should be larger.

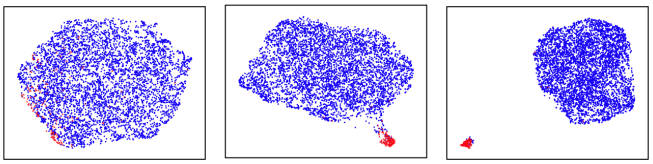


FIGURE 2. Feature distribution after UMAP dimension reduction from left to right corresponding to $\beta = 0, 0.2, 0.5$, where the red (blue) points represent the poison (benign) samples

For the hyperparameters of LFDP, we set the window size $l = 20$ (other values should also have similar empirical results) and determine the regularization score β by an ablation study. Specifically, we set its value as 0, 0.2, and 0.5, and visualize the feature distribution in Fig. 2. The results show that with the increase of β, the poisoned samples are pulled far away from benign samples. In the following experiment, we fix the $\beta = 0.5$. Note, for the hyperparameters (except for the threshold) of the existing works, we directly utilize their default values in their official codes.

5.3 Empirical comparison

We select five existing works: Beatrix [11], CCAUD [8], PIPD [16], FCT [12], and ConfusLearn [15], and compare their performances with our LFPD. Specifically, we draw the ROC curves of six defense methods against the three attacks and list the corresponding AUC values in Fig. 3. We observe that the AUC of Beatrix, CCAUD, PIPD, and FCT are close to or lower than 0.5. That is because: 1) Beatrix, CCAUD, and FCT rely on the feature separability assumption, which does not hold for adaptive attacks, and 2) for the PIPD defense, it is hard to use the loss value to distinguish the poisoned samples from the benign ones, specifically when the poisoning rate is tiny. The second observation is that the performance of ConfusLearn is good when defending against the TaCT attack, due to the trigger signal being visible in this case. However, for the Adap-blend and Adap-patch attacks, its performance drops significantly. Finally, our method can work well in all three types of backdoor attacks.

With the threshold ξ^* determined by the strategy described in Sec. 5.2, we calculate the corresponding TPR_{ξ^*} and FPR_{ξ^*} of all defense methods against three types of backdoor attacks, as shown in Tab. 2. From this table, we find that Beatrix, CCAUD, PIPD, and FCT cannot distinguish the poisoned samples from the benign ones, so their TPR_{ξ^*} and FPR_{ξ^*} are close each other. For ConfusLearn, it can detect the poisoned samples perfectly in TaCT attack with $\text{TPR}_{\xi^*} = 1$ and $\text{FPR}_{\xi^*} = 0$, but fail to defend the Adap-blend and Adap-patch attacks. By comparison, our method LFPD can detect the poisoned samples with $\text{TPR}_{\xi^*} \approx 1$ and $\text{FPR}_{\xi^*} \approx 0$.

TABLE 2. Performance Metrics

Defenses	TaCT		Adap-blend		Adap-patch	
	TPR_{ξ^*}	FPR_{ξ^*}	TPR_{ξ^*}	FPR_{ξ^*}	TPR_{ξ^*}	FPR_{ξ^*}
Beatrix	0	0	0	0	0.004	0.006
CCAUD	0	0	0	0	0	0
PIPD	0.9	0.591	1	0.992	0.853	0.816
FCT	1	0.917	0.853	0.806	0.286	0.247
ConfusLearn	1	0	0.786	0.589	0.66	0.281
LFPD	1	0.003	1	0	1	0.002

Finally, in Tab. 3, based on the detected poisoned dataset $\bar{D}_p$, we report ACC^* and ASR^* of unlearned models. Since Beatrix and CCAUD detect the poisoned data with $\text{TPR}^* \approx 0$ and $\text{FPR}^* \approx 0$, ACC^* and ASR^* of their corresponding unlearned models remain similar to the poisoned models. Moreover, PIPD and FCT have a high TPR^* in poison detection, which leads to the accuracy on the clean test samples of their unlearned models is destroyed with $\text{ACC}^* < 0.75$. Finally, for the ConfuLearn, after unlearning, only the TaCT backdoor can be successfully removed, while for the other two attacks, their accuracy on the clean test samples is hurt. Finally, for LFPD, the accuracy on the clean test samples is similar to the poisoned models, and their ASR^* drops lower than 0.063.

TABLE 3. Performance of unlearned model

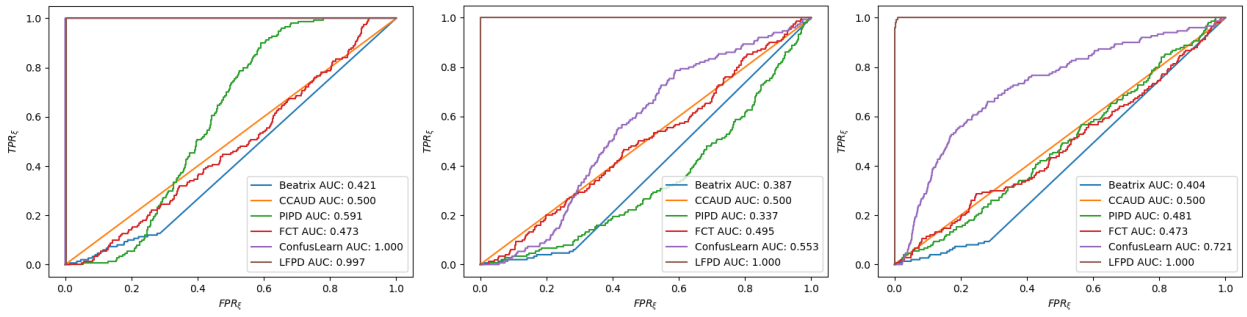
Defenses	TaCT		Adap-blend		Adap-patch	
	ACC^*	ASR^*	ACC^*	ASR^*	ACC^*	ASR^*
Beatrix	0.937	0.865	0.937	0.803	0.929	0.652
CCAUD	0.937	0.86	0.937	0.803	0.930	0.65
PIPD	0.373	0.148	0.009	0.017	0.246	0.209
FCT	0.089	0.017	0.225	0.016	0.749	0.434
ConfusLearn	0.923	0.009	0.471	0.313	0.786	0.339
LFPD	0.932	0.001	0.927	0.048	0.913	0.063

6 Conclusion

Feature separability is a common assumption in training-level backdoor defenses but can be bypassed by adaptive attacks. In this paper, aimed at these attacks, we proposed the so-called LFPD method, exploiting a local feature algorithm as a regularizer to increase the feature separability between poisoned and benign samples. Our experiment shows that our LFPD can efficiently defend against adaptive attacks compared with five existing defenses.

Acknowledgements

This work has been partly supported by the European Union's Horizon Europe research and innovation program under the project ELSA, grant agreement no. 101070617; by project SERICS (PE00000014) under the MUR National Recovery and Resilience Plan funded by the European Union –



(a) TaCT **(b)** Adap-blend **(c)** Adap-patch

FIGURE 3. ROC curve of different defense methods against three different attacks

NextGenerationEU; by BMK, BMDW, and the Province of Upper Austria in the frame of the COMET Programme managed by FFG in the COMET Module S3AI; and by Fondazione di Sardegna under the project "TrustML: Towards Machine Learning that Humans Can Trust", CUP: F73C22001320007.

References

[1] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *ECML PKDD 2013*. Springer, 2013, pp. 387–402.

[2] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *ICML 2012*. icml.cc / Omnipress, 2012.

[3] W. Guo, B. Tondi, and M. Barni, "A temporal chrominance trigger for clean-label backdoor attack against anti-spoof rebroadcast detection," *IEEE TDSC*, vol. 20, no. 6, pp. 4752–4762, 2023.

[4] ——, "An overview of backdoor attacks against deep neural networks and possible defences," *IEEE Open Journal of Signal Processing*, vol. 3, pp. 261–287, 2022.

[5] B. Tran, J. Li, and A. Madry, "Spectral signatures in backdoor attacks," in *NeurIPS 2018*, 2018, pp. 8011–8021.

[6] B. Chen, W. Carvalho, N. Baracaldo, H. Ludwig, B. Edwards, T. Lee, I. M. Molloy, and B. Srivastava, "Detecting backdoor attacks on deep neural networks by activation clustering," in *AAAI 2019*, vol. 2301, 2019.

[7] Z. Xiang, D. J. Miller, and G. Kesidis, "A benchmark study of backdoor data poisoning defenses for deep neural network classifiers and A novel defense," in *MLSP 2019*, 2019, pp. 1–6.

[8] W. Guo, B. Tondi, and M. Barni, "Universal detection of backdoor attacks via density-based clustering and centroids analysis," *IEEE TIFS*, 2023.

[9] D. Tang, X. Wang, H. Tang, and K. Zhang, "Demon in the variant: Statistical analysis of dnns for robust backdoor contamination detection," in *USENIX 2021*, 2021, pp. 1541–1558.

[10] X. Qi, T. Xie, Y. Li, S. Mahloujifar, and P. Mittal, "Revisiting the assumption of latent separability for backdoor defenses," in *ICLR 2023*. OpenReview.net, 2023.

[11] W. Ma, D. Wang, R. Sun, M. Xue, S. Wen, and Y. Xiang, "The "beatrix" resurrections: Robust backdoor detection via gram matrices," in *NDSS*, 2023.

[12] W. Chen, B. Wu, and H. Wang, "Effective backdoor defense by exploiting sensitivity of poisoned samples," in *NeurIPS 2022*, 2022.

[13] S. Pal, Y. Yao, R. Wang, B. Shen, and S. Liu, "Backdoor secrets unveiled: Identifying backdoor data with optimized scaled prediction consistency," in *ICLR 2024*, 2024.

[14] Y. Li, X. Lyu, N. Koren, L. Lyu, B. Li, and X. Ma, "Anti-backdoor learning: Training clean models on poisoned data," in *NeurIPS 2021*, 2021, pp. 14 900–14 912.

[15] X. Qi, T. Xie, J. T. Wang, T. Wu, S. Mahloujifar, and P. Mittal, "Towards A proactive ML approach for detecting backdoor poison samples," in *USENIX 2023*. USENIX Association, 2023, pp. 1685–1702.

[16] Y. Chen, H. Wu, and J. Zhou, "Progressive poisoned data isolation for training-time backdoor defense," *AAAI 2024*, 2024.

[17] J. Shi and C. Tomasi, "Good features to track," in *CVPR 1994*. IEEE, 1994, pp. 593–600.