Detecting Adversarial Samples Using Influence Functions and Nearest Neighbors Supplementary Material

Gilad Cohen
Tel Aviv University
Tel Aviv, 69978
giladco1@mail.tau.ac.il

Guillermo Sapiro
Duke University
North Carolina, 27708
guillermo.sapiro@duke.edu

Raja Giryes
Tel Aviv University
Tel Aviv, 69978
raja@tauex.tau.ac.il

1 Method

The main paper proposes a new reactive detection method for adversarial images: the Nearest Neighbors Influence Functions (NNIF). Our detector utilizes a influence functions algorithm as shown in Koh and Liang [2017] to measure the contribution of each training sample to a test samples prediction. Their algorithm is summarized in Algorithm 1. For measuring the influence a train sample z has on the loss of a specific test sample z_{test}, Koh and Liang [2017] approximate this term:

$$I_{up,loss}(z, z_{test}) = -\nabla_\theta L(z_{test}, \theta)^T H_\theta^{-1} \nabla_\theta L(z, \theta), \tag{1}$$

where H is the Hessian of the machine learning model, L is its loss, and θ are the model parameters. Eq. (1) is repeated throughout the training set, calculating $I_{up,loss}$ for every training sample. For our NNIF algorithm only the top M helpful training examples (H^+_{inds}) and the top M harmful training examples (H^-_{inds}) are chosen for further processing.

Algorithm 1 Influence Functions

```
Input: Test sample $(x_i, y_i)$ and a training set $(X_{train}, Y_{train})$
Input: $M$: Number of top influence samples to collect
Output: $H^+_{inds}$, $H^-_{inds}$                                ▷ Most helpful/harmful training examples indices
$N_{train} = |X_{train}|$
Initialize $H^+_{inds}$=[], $H^-_{inds}$=[]
Initialize $I_{up,loss}$ = zeros[$N_{train}$]
for $(x_j, y_j)$ in $(X_{train}, Y_{train})$ do
    $I_{up,loss}[j] = -\nabla_\theta L(x_i, \theta)^T H_\theta^{-1} \nabla_\theta L(x_j, \theta)$      ▷ Apply influence function (Eq. (1))
end for
sort($I_{up,loss}[j]$)                                            ▷ Sorting for the most influential training samples
for $m$ in $[0, M-1]$ do
    $j^+_m$ = Training example index of $I_{up,loss}[N_{train} - m]$   ▷ choosing most helpful examples
    $H^+_{inds}$.append($j^+_m$)
    $j^-_m$ = Training example index of $I_{up,loss}[m]$                ▷ choosing most harmful examples
    $H^-_{inds}$.append($j^-_m$)
end for
return $H^+_{inds}$, $H^-_{inds}$                                 ▷ Most helpful/harmful training examples indices
```

2 Experimental setup

The DNNs clean accuracies, when not under attack, are shown in Table 1. In Table 2 we present the attack success rate of the Fast Gradient Sign Method (FGSM) (Goodfellow et al. [2015]), Jacobian-based Saliency Map Attack (JSMA) (Papernot et al. [2016]), Deepfool (Moosavi-Dezfooli et al. [2016]), Carlini & Wagner (CW) (Carlini and Wagner [2017]), our CW-Opt attack, Projected Gradient Descent (PGD) (Madry et al. [2018]), and Elastic-net Attack on Dnns (EAD) (Chen et al. [2018]). Note that the success rates of all attacks are higher for CIFAR-100. This makes sense since CIFAR-100 dataset has 100 classes instead of 10, and it is thus more vulnerable to misclassifications.

Table 1: DNN clean accuracies (%), for normal images not under attack.

Dataset	train acc	val acc	test acc
CIFAR-10	99.75	93.70	92.08
CIFAR-100	96.80	70.80	67.99
SVHN	99.46	96.20	94.59

Table 2: Adversarial attack success rates (%) of FGSM, JSMA, Deepfool, CW, CW-Opt, PGD, and EAD. CW-Opt attack is CW regulated with a loss term optimized against our NNIF defense in a white-box setting.

Dataset	FGSM		JSMA		Deepfool		CW		CW-Opt		PGD		EAD	
	val	test	val	test	val	test	val	test	val	test	val	test	val	test
CIFAR-10	80.47	79.27	71.18	70.21	94.34	96.19	93.70	94.46	86.87	86.31	79.62	80.51	46.64	48.14
CIFAR-100	95.19	95.26	86.02	86.19	100.00	99.91	99.44	98.90	99.15	99.10	99.58	99.25	86.86	89.41
SVHN	84.72	85.51	69.02	65.51	92.62	92.45	93.24	95.69	49.69	45.96	39.09	47.73	75.99	77.44

The paper explains how we tuned the hyper-parameters for the four inspected algorithms: DkNN, LID, Mahalanobis, and our NNIF method. For the DkNN and LID algorithms we tuned the number of neighbors (k), for the Mahalanobis algorithm we tuned the noise magnitude (ϵ), and for our NNIF method we set the number of top influence samples to collect (M). All parameters were chosen using nested cross entropy validation within the validation set, based on the AUC values of the detection ROC curve. The results are shown in Table 3.

Table 3: Hyper-parameter setting for the four inspected detectors. k denotes the number of nearest neighers used in DkNN and LID algorithms, ϵ is the noise magnitude in the Mahalanobis detector, and M is the number of most helpful/harmful training images used in our NNIF method.

Dataset	Param	FGSM	JSMA	Deepfool	CW	PGD	EAD
CIFAR-10	DkNN (k)	4900	5000	4900	4900	4800	4900
	LID (k)	18	18	18	18	24	16
	Mahalanobis (ϵ)	0.0002	0.0002	0.00005	0.00001	0.00005	0.00001
	NNIF (M)	50	200	100	200	450	500
CIFAR-100	DkNN (k)	490	450	20	430	500	10
	LID (k)	10	10	10	10	10	10
	Mahalanobis (ϵ)	0.005	0.005	0.0005	0.00001	0.01	0.0002
	NNIF (M)	30	30	40	40	50	30
SVHN	DkNN (k)	3200	3000	1400	3200	3200	3200
	LID (k)	18	22	22	22	22	24
	Mahalanobis (ϵ)	0.001	0.0005	0.00005	0.00001	0.00008	0.00001
	NNIF (M)	300	50	300	50	100	100

3 Detection of adversarial attacks

Figure 1 presents two ROC curves for classification of Deepfool and CW adversarial attacks on the CIFAR-10 dataset. One can observe that our NNIF method (solid red line) achieves better classification power over the previous state-of-the-art methods.

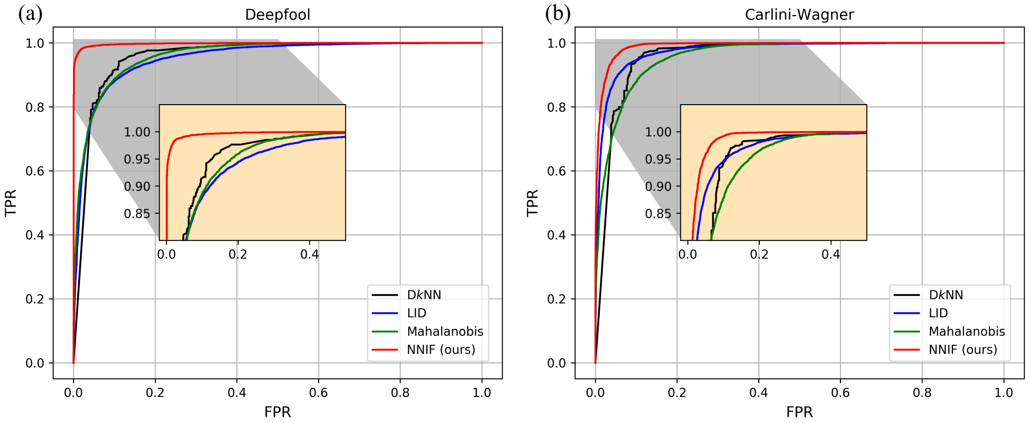

Figure 1: ROC curves for classifying adversarial examples. (a) Defending Deepfool attack. (b) Defending Carlini-Wagner (CW) L_2 attack. All plots correspond to the CIFAR-10 dataset. We achieve state-of-the-art results, surpassing previous defense methods by a large margin.

Table 4 presents the AUC scores for the adversarial detection of FGSM, JSMA, Deepfool, CW, PGD, and EAD attacks on CIFAR-10, CIFAR-100, and SVHN datasets. These results were obtained by using DNN's features from only the embedding space. A similar table with detectors which were trained on the entire DNN's features is in the main paper.

Table 4: Comparison of AUC scores (%) for various adversarial detection methods, for FGSM, JSMA, Deepfool, CW, PGD, and EAD attacks. Results obtained using only the DNN's penultimate layer.

Dataset	Detector	FGSM	JSMA	Deepfool	CW	PGD	EAD
CIFAR-10	DkNN	87.81	95.37	95.82	96.88	86.83	85.20
	LID	90.12	94.67	95.43	97.66	90.49	82.87
	Mahalanobis	**96.80**	**98.95**	96.49	96.96	92.91	85.30
	NNIF (ours)	87.75	97.67	**99.82**	**99.05**	**94.01**	**88.06**
CIFAR-100	DkNN	**93.65**	83.46	76.71	93.77	73.78	**78.42**
	LID	80.68	74.33	52.25	67.84	72.25	52.10
	Mahalanobis	83.90	**90.20**	62.05	71.60	72.46	61.65
	NNIF (ours)	87.23	86.63	**84.20**	**94.58**	**83.09**	72.42
SVHN	DkNN	85.24	94.61	91.13	95.15	79.07	84.77
	LID	88.38	94.31	92.00	95.64	80.92	86.74
	Mahalanobis	**98.14**	**99.15**	96.07	98.26	90.41	92.95
	NNIF (ours)	91.06	98.29	**97.11**	**98.68**	**92.46**	**93.72**

4 Ablation study

To inspect how the four learned features influence our adversarial detection we conducted an ablation study on CIFAR-10 dataset, for four attacks: FGSM, JSMA, Deepfool, and CW. The results are shown in Table 5. From these results, one may conclude that the most beneficial feature is $\mathbb{D}^{M\uparrow}$, which is the L_2 distance from the most helpful training examples on the deep neural network (DNN) embedding space.

Figure 2 shows the probability density functions for $\mathbb{R}^{M\uparrow}$, $\mathbb{D}^{M\uparrow}$, and $\mathbb{D}^{M\downarrow}$ features on CIFAR-10 for the Deepfool and CW adversarial attacks. From these histograms, it can be easily observed that $\mathbb{R}^{M\uparrow}$ or $\mathbb{D}^{M\uparrow}$ are more useful for detecting Deepfool adversarial attacks than CW attacks. On the other hand, the $\mathbb{D}^{M\downarrow}$ feature discriminates CW attacks better than Deepfool attacks. This is also supported by the results on Table 5: For $\mathbb{R}^{M\uparrow}$ or $\mathbb{D}^{M\uparrow}$ alone NNIF detects Deepfool better than CW ($98.27\% > 81.91\%$ and $99.79\% > 97.27\%$), however, for $\mathbb{D}^{M\downarrow}$ NNIF is able to detect CW attacks better than Deepfool attacks ($89.97\% > 82.11\%$).

Table 5: Ablation test for adversarial attack detection: Calculating AUC score and accuracy for selected features. Attacking CIFAR-10 dataset using FGSM, JSMA, Deepfool, and CW.

$\mathbb{R}^{M\uparrow}$	$\mathbb{D}^{M\uparrow}$	$\mathbb{R}^{M\downarrow}$	$\mathbb{D}^{M\downarrow}$	FGSM	JSMA	Deepfool	CW
			✓	78.99	83.23	82.11	89.97
		✓		51.4	51.93	66.14	53.14
		✓	✓	82.08	85.11	83.25	90.27
	✓			84.19	97.41	99.79	97.27
	✓		✓	86.74	97.54	99.82	98.81
	✓	✓		84.20	97.41	99.79	97.27
	✓	✓	✓	87.74	97.66	99.81	99.0
✓				64.85	85.27	98.27	81.91
✓			✓	80.19	85.4	97.73	95.14
✓		✓		64.31	85.34	98.28	81.95
✓		✓	✓	83.14	85.97	97.62	95.34
✓	✓			84.18	97.43	99.79	97.21
✓	✓		✓	86.66	97.51	99.81	98.85
✓	✓	✓		84.22	97.44	99.79	97.21
✓	✓	✓	✓	**87.75**	**97.67**	**99.82**	**99.05**

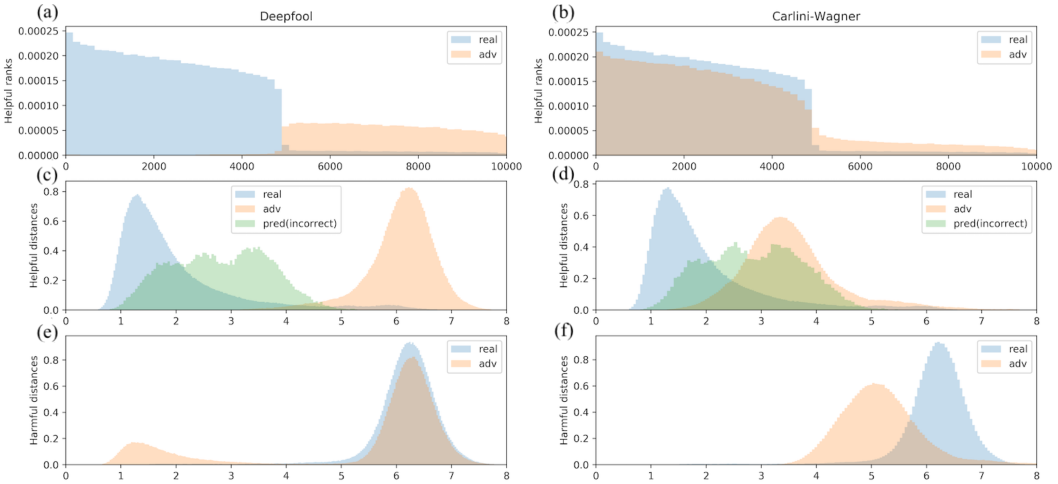

Figure 2: Probability density functions of the most helpful ranks ($\mathbb{R}^{M\uparrow}$, top row), most helpful distances ($\mathbb{D}^{M\uparrow}$, middle row), and the most harmful distances ($\mathbb{D}^{M\downarrow}$, bottom row), on CIFAR-10 for the Deepfool and CW attacks. The features for the normal (untempered) images that were correctly classified by the network are shown in blue. The features for the adversarial images are shown in orange. The features for the normal images that were misclassified by the network are shown in green (in the middle row).

5 Generalization to other attacks

The main paper measures the NNIF method transferability from one attack (FGSM) to other, unseen attacks (JSMA, Deepfool, CW, PGD, and EAD), where all the features are extracted from the penultimate activation layer. Here we provide a similar table where all the DNN's activation layers are employed for this comparison (Table 6), except of DkNN which only utilizes features from the DNN's embedding space. The generalization results in Table 6 does not have a definite winner method. The DkNN, Mahalanobis, and our NNIF methods demonstrate the best transferability for various setups. The LID detector shows the worst generalization overall.

Table 6: Generalization of adversarial detection from FGSM attack to unseen attacks. The LR classifier is trained on all activation layers' features extracted after applying FGSM attack, and then evaluated on JSMA, Deepfool, CW, PGD, and EAD.

Dataset	Detector	FGSM (seen)	JSMA	Deepfool	CW	PGD	EAD
CIFAR-10	DkNN	87.81	94.89	**95.21**	**96.76**	85.10	**83.28**
	LID	98.18	91.70	84.51	91.67	**85.62**	70.85
	Mahalanobis	99.80	**96.11**	86.25	85.17	84.24	68.30
	NNIF (ours)	99.96	92.76	79.84	84.44	81.66	70.02
CIFAR-100	DkNN	93.65	83.16	62.41	**92.22**	73.60	62.67
	LID	92.33	72.65	51.19	59.09	64.49	51.00
	Mahalanobis	99.87	82.26	52.15	53.72	52.94	52.58
	NNIF (ours)	99.96	**89.52**	**64.33**	86.43	**85.79**	**63.64**
SVHN	DkNN	85.24	93.43	89.84	**92.20**	75.99	79.81
	LID	99.92	94.91	82.55	82.26	69.90	73.40
	Mahalanobis	100.00	**99.18**	**92.24**	86.87	**82.57**	**81.06**
	NNIF (ours)	100.00	92.45	80.14	83.20	75.74	75.52

6 Attack against NNIF

We applied a white-box attack against our NNIF defense model on CIFAR-10/100 and SVHN datasets, CW-Opt (Section 4.5 in the main paper). This attack optimization requires a hyper-parameter in the new regularization term, M. This is the number of the most helpful training examples of the normal image. We apply this term only on the top 1% helpful training samples which belong to the predicted class (we find this to be most effective for the attack to succeed). Therefore, we set $M = 50$ for CIFAR-10 and SVHN and $M = 5$ for CIFAR-100. Table 7 shows the DkNN, LID, Mahalanobis, and our NNIF detection accuracies on two scenarios: 1) With the vanilla CW attack and 2) With our white-box attack (CW-Opt).

Table 7: Attack failure rate without defense (%) and defense accuracy (%) for a white-box attack targeting the NNIF detector. The attack failure rate in the third column corresponds to the probability of the adversary to fail flipping a correct label without any defense method.

Dataset	Attack	Attack fail rate	Defense accuracy (%)			
		(w.o. defense) (%)	DkNN	LID	Mahalanobis	NNIF
CIFAR-10	CW	5.54	93.45	91.43	90.70	91.95
	CW-Opt	13.69	90.99	89.74	92.29	90.81
CIFAR-100	CW	1.10	87.42	61.37	64.16	85.42
	CW-Opt	0.90	94.16	66.05	51.98	91.15
SVHN	CW	4.31	91.03	87.91	93.24	94.65
	CW-Opt	54.04	65.59	70.21	77.23	75.21

For CIFAR-10 we observe only a 1% decrease in our NNIF adversarial detection accuracy. Similar decrease is present also for all the algorithms which utilize L_2 distance of nearest neighbors in the embedding space: DkNN and LID.

For SVHN we observe that CW-Opt attack impairs our NNIF defense by 20%. We speculate this is because CW-Opt was able to flip only 46% of labels in the SVHN test set, instead of 96% where attacking with the vanilla CW. Therefore, in the white-box setting we consider only the hardest test samples for our detection task. We also notice that DkNN and LID defense accuracies are decreased by more than 20% as well.

The results for CIFAR-100 are unconformable to the other datasets, showing an increase of the NNIF detection accuracy in the white-box setting. This finding also presents with DkNN and LID, which is correlative to the trend shown on CIFAR-10. This happens since the attack focuses only on the most helpful distance feature and our defense takes into account also other parameters. Therefore, to verify our white-box attack indeed brings an adversarial image closer to its natural image's helpful training images (in the embedding space), we repeated the experiment by only collecting the distance features, $\mathcal{D}^{M\uparrow}$, in our defense and ignoring the ranks, $\mathcal{R}^{M\uparrow}$. This method demonstrates a decrease of the detection accuracy from 74% to 65%. This shows that indeed the white box attack also affects CIFAR-100 when it relies only on distance features. The detection accuracies using only $\mathcal{D}^{M\uparrow}$ are summarized in Table 8. Note that our defense technique is always robust to the white-box attacks when it only uses the distance features. The fact that we show robustness also when considering the ranks features makes it even stronger since it is hard to optimize the white-box attacks to ranks (as they are non-differentiable).

Overall, we conclude that our NNIF defense method is robust in a white-box setting.

Table 8: Defense accuracy (%) for a white-box attack targeting the NNIF detector, using only the distance features $\mathcal{D}^{M\uparrow}$.

Dataset	Attack	NNIF defense acc.
CIFAR-10	CW	91.96
	CW-Opt	90.91
CIFAR-100	CW	74.09
	CW-Opt	65.48
SVHN	CW	94.65
	CW-Opt	75.27

7 Influence function smoothness

Since we use ReLU activations in our Resnet-34 DNN, the cross entropy loss function is not continuously differentiable, therefore we might have an issue calculating the influence function in Eq. (1). Although this is a technical concern, in practice we can assume this is not an issue since the set of discontinuities has measure zero, and the problematic activation points will never be encountered in the back propagation.

References

Nicholas Carlini and David A Wagner. Towards evaluating the robustness of neural networks. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017.

Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*, 2018. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/16893.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. URL http://arxiv.org/abs/1412.6572.

Pang Wei Koh and Percy S. Liang. Understanding black-box predictions via influence functions. In *ICML*, 2017.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. URL https://openreview.net/forum?id=rJzIBfZAb.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. *CVPR*, pages 2574–2582, 2016.

Nicolas Papernot, Patrick D McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. *IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 372–387, 2016.