Features which are robust to adversarial attacks are also robust to several poisoning attacks

Adrien CHAN-HON-TONG

September 8, 2021

Abstract

Most data poisoning methods target naive deep networks. Yet, it is well known that those networks exhibit strong sensitivity to perturbations.

Inversely, in this paper, I show that several data poisoning attacks (e.g. poison frog) are ineffective as soon as there are applied on features made robust to adversarial attacks, on both CIFAR and MNIST datasets.

This result stresses that some state of the art data poisoning results may have been corrupted by adversarial sensibility and should be further checked on robust networks instead of naive ones.

Code is available at github.com/achanhon/AdversarialModel/V4.

1 Introduction

Deep learning (**DL**) which appears with [12] (see [14] for a review) is now at the core of most computer vision pipelines. Yet, many challenges have to be tackled before real life applications of deep learning for critical tasks: fairness, privacy, explainability...

One of these challenge, which has received a very strong attention from the community, is robustness. Indeed, it is known that naive deep learning is vulnerable under adversarial attacks [18, 29, 23, 27, 22, 7]: at test time, it is possible to design a specific invisible perturbation such as a targeted network eventually predicts different outputs on original and disturbed input. Worse, producing adversarial examples does not require to have access to the internal structure of the network [4, 20] and can have physical implementation [13].

Typically, an hacker could modify a traffic signal such that it is wrongly classified by a targeted autonomous driving model.

Another issue is data poisoning [19] where an hacker modifies the training data to force the model to get a specific behavior. Figure 1 illustrates adversarial attack and data poisoning ones.

Yet, training is done in a much more secure environment (typically, on private data). And, there exists formal defense against poisoning. For example, [16] stresses that if someone learn 5 classifiers on 5 disjoint set of a large dataset, then, an hacker which would have modified a single training samples can only

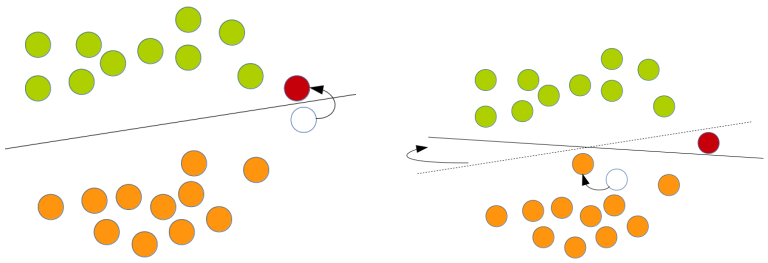

Figure 1: Illustrations of adversarial attack vs poisoning attack: goal of the hacker is to have the red point classified as green and not orange, black line is decision boundary of the classifier.

modify 1 over the 5 models. By performing majority vote over the 5 models, one could be sure that as soon as 4 models agree, then, the decision is robust to a data poisoning attacks.

For this reason, a real data poisoning attack against real life deep learning may never happen. This may explain why data poisoning has received less attention than adversarial attacks.

Yet, despite data poisoning is much less probable and can be virtually mitigated. It should still be considered as a threat. First, splitting the training dataset in 5 to be robust to a single datum perturbation is quite unrealistic with modern deep networks which are best on large training set. Then, a single hacker can break the system for all users with data poisoning. Because, the result of the attack is a bad model, which will have bad behavior everywhere (inversely, with adversarial attacks, an hacker can only modify one traffic sign at a times). Thus, poisoning attack can be much more harmful than adversarial ones.

For these reasons, data poisoning should still be considered by the community even if adversarial attacks may be a more urgent issue. Even more, the motivation of this paper is that both issues may be related. Indeed, both data poisoning and adversarial attacks are related with the idea of moving some data in a feature space (despite adversarial attack moves in a frozen space while feature space changes when poisoning perturbs training data).

Yet, adversarial defenses had never been considered as potential way to mitigate data poisoning in state of the art. The contribution of this paper is to prove that deep features trained with adversarial defense are more robust to poisoning attacks than naive ones. This result was not trivial because adversarial defense is about to push the decision boundary far from training data, while, poisoning is about diverting the decision boundary. Besides, this paper does not claim that adversarial defenses prevent all poisoning attacks, but, at least the selected ones.

Precisely, three poisoning attacks are considered: PoisonFrog [25], adver-

sarialpoisoning [3] and a labelflip attack (related to [19]). The evaluations will rely on classical computer vision datasets CIFAR10 and CIFAR100 [11], MNIST [15], SVHN [21] with or without adversarial retraining (as adversarial defense). Consistently with [25, 3], deep features rather than full deep networks are considered. A consistent trend in all those experiments is that adversarial defense reduce poisoning effect.

Experimental framework is presented in section 3 after presentation of the state of the art in section 2. Then, results and discussion are presented in section 4.

2 Related works

2.1 Adversarial defenses

As soon as adversarial attack appears [27], there were a lot of research to find way to mitigate this issue. However, first methods like distillation [24] or gradient masking cure the symptoms rather than the causes [2], and have been quickly bypassed by new attacks.

Recent methods tends to increase the margin between each training data and the decision boundary. Yet the computation of the exact margin is a hard problem and requires the use of formal tools [10]. Thus, the two main ways are either an overestimation or an underestimation of the margin.

The overestimation of the margin relies on the generation of adversarial examples, this is the so called adversarial retraining, where, at each step of the training time, adversarial examples are considered instead of original examples to force the network to be margin aware. Currently, if the attack is strong, then, this attack could lead to very robust model e.g. [17].

The underestimation of the margin is based on the idea to produce a convex overestimation of the accessible space (in feature space) related to the perturbation of the input. The pioneer work of this way is [28].

More formally, a network f (for binary classification) with weight w admits an ε adversarial example in x if $\exists \delta,\ ||\delta|| \leq \varepsilon,\ f_w(x)f_w(x+\delta) < 0$. Thus, a way to increase robustness on x is to perform the backpropagation not on x but on $x^* = \underset{z \in B_\varepsilon(x)}{\arg\min}\ f_w(x)f_w(z)$ where $B_\varepsilon(x) = \{z, ||z-x|| \leq \varepsilon\}$. But, as computing x^* is hard, then, adversarial retraining relies on $\hat{x} = \underset{z \in \omega \subset B_\varepsilon(x)}{\arg\min}\ f_w(x)f_w(z)$ where ω is a set defined by adversarial attack (e.g. [17]), while [28] offers to compute $\tilde{x} = \underset{z \in B_\varepsilon(x)}{\arg\min}\ f_w(x)g_w(z)$ where g_w is a convex approximation of f_w leading to $f_w(B_\varepsilon(x)) \subset g_w(B_\varepsilon(x))$. Recently, an other way seems to emerge based on 1-Lipschitz networks [1] for which structurally $|f_w(x)| > \varepsilon \Rightarrow \min_{z \in B_\varepsilon(x)} f_w(x)f_w(z) > 0$.

However, [1] requires not standard deep networks (e.g. no relu), while methods related to [28] are still very expensive in computation. For those reasons, this paper is based on adversarial retraining with 2 very classical attacks:

- **FSGM** [8] which consists in generating an adversarial candidate with

$$\hat{x} = x + \varepsilon \times \text{sign}(\nabla_x L(x, y))$$

where L is the loss function (typically a cross entropy) on x with class y ($y \in \{-1, 1\}$ for binary classification).

- **PGD** is a stronger attack introduced in [17] which consists in

$$\hat{x}_{t+1} = \text{Proj}\left(\hat{x}_t + \alpha \times \text{sign}(\nabla_x L(\hat{x}_t, y)), x\right)$$

where $\text{Proj}(z, x)$ projects z such that $||\text{Proj}(z, x) - x|| \leq \varepsilon$.

2.2 Selected data poisoning attacks

Data poisoning consists in modifying some training samples to get a specific behavior of the classifier after training on poisoned data. Yet, the underlying idea is that the perturbation of the training data should not be easily detected by the owner of the dataset. [19] considers heavy modification (image and label) of a small subset of the data (hoping owner will not review all data) while [3, 25] considers invisible perturbation of many images (harder to detect but require to access more training data for [3]).

Precisely, [19] considers improved label flip (**LF**) attack: label of some images are changed forcing the network to learn with label noise. Yet, deep networks relies on stochastic gradient descent which is not global, and, quite able to deal with label noise. So, [19] also introduces an image perturbation which enhances the label flip effect (see [19] figure 4.a).

[25] offers a very different attack called poisonfrog (**PF**) which requires no label modification, and, only invisible perturbation of a small subset of training data (eventually only one). First, the attack targets deep features rather than deep networks: only the last layer of the network is trained on clean/poisoned data (all other layers have been trained before). This training of the last layer is done with support vector machine (**SVM**) [5]. The objective of the attack is to modify a training sample x with class $y = -1$ in order that the resulting model will wrongly classify the targeted testing sample x_t with class $y = 1$. The core of the attack relies on the idea of poisoning the training sample x into $z = x + \delta$ such that x_t has the same position in feature space than z. This way, when trained on with $(z, -1)$ in training set, the resulting model f_w will probably verify $f_w(z) = f_w(x_t) < 0$ i.e. x_t is wrongly classified.

Taking advantage of the fact that deep features are frozen, the poisoning consists in solving

$$\min_{z,\ ||z-x|| \leq \varepsilon} ||\text{feature}(x_t) - \text{feature}(z)||$$

This problem is very close to the one of generating an adversarial example, and, can be implemented in either FSGM fashion or PGD one.

Again, this attack relies on the capacity to perform a large modification of feature$(x+\delta)$ with only a small perturbation δ. Thus, this attack may be less efficient on robust features.

Finally, the last attack considered in this paper is adversarial poisoning (**AP**) [3] which is similar than [25], but, based on the idea of directly modifying the decision boundary (also taking advantage of frozen deep feature, and, also without modifying label).

The first phase of the attack is to generate the clean model w_c, and, a proxy model w_t. The objective of the hacker is to make w_c rotates to reach w_t. This can be reached by rotating many training datum x with this rotation (indeed, SVM commutes with rotation when data are linearly separable). At first order, the poisoning consists in moving training as many datum x on the axis $w_t - w_c$ i.e. for any possible training x to solve:

$$\max_{z,\ ||z-x||\leq\varepsilon} ||(w_t - w_c)^T \text{feature}(z)||$$

Like for [25], this problem is very close to the one of generating an adversarial example, and, can be implemented in either FSGM fashion or PGD one.

Again, this attack relies on the adversarial effect, and, should be evaluated on robust features.

3 Adversarial defense against data poisoning

3.1 Overview

In order to evaluate the different poisoning attacks against the different features with more or less robustness, I rely on the following framework illustrated by figure 2. Importantly, the framework relies on frozen features following [25, 3]. Yet, both those papers have then been extended to poisoning against deep networks. Thus, the fact to rely on frozen features may not restrict too much the scope of the paper.

Classically, data poisoning is about comparing poisoned/clean behaviour related to poisoned/clean model where the poisoned model is trained on poisoned data (and clean model on clean data). This is the right part of the figure 2.

Here, the objective is to see how those poisoning attacks behave as function of the features (brown lozenge in figure 2). Those features are produced from an external data (e.g. Imagenet [6] in [25, 3]). This is the left part of the figure 2 which corresponds to the classical training of a deep network with or without adversarial defense.

So classically, data poisoning papers focus on the poisoning attack (the cyan ellipsoid in figure 2). Inversely, in this paper, the attacks are selected from state of the art. But, the contribution is to evaluate the impact of the adversarial defense (purple ellipsoid in figure 2).

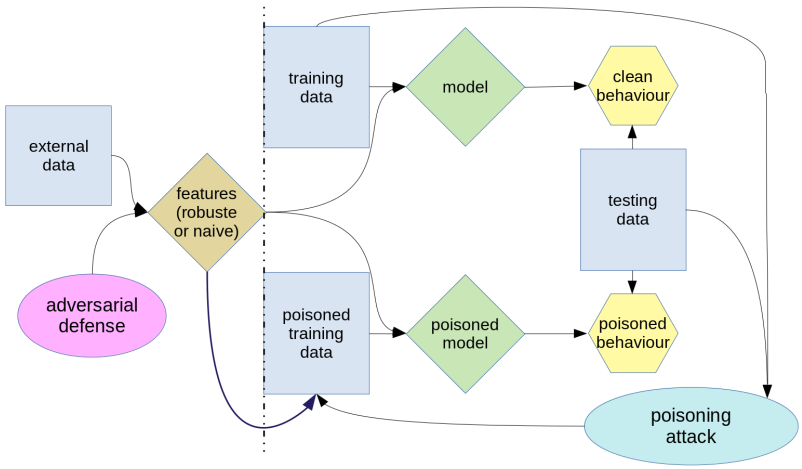

Figure 2: Illustrations of the framework to evaluate the impact of adversarial defense (and so feature robustness) on poisoning attacks.

3.2 Implementation details

Experiments are performed on CIFAR10 and CIFAR100 [11], MNIST [15], SVHN [21]. Precisely, I consider two setting: CIFAR100 is used as external data, then CIFAR 10 is used as train/test data - or - SVHN is used as external data, then, MNIST is used as train/test data.

Importantly, one could have expected the use of Imagenet [6] as external data. Yet, producing robust Imagenet features is already a challenging task. This is why, CIFAR100 is used as external data to tackle CIFAR10 and SVHN for MNIST. However, all poisoning attacks are first validated on Imagenet feature (see code in github.com/achanhon/AdversarialModel/V4) to ensure correct implementation of the attacks.

Both VGG [26] and ResNet [9] are considered as feature backbones. The training of those networks is realised classically (Adam solver, cross entropy, ... see the code on github), then, the last layer is removes to keep only the feature part. The number of epochs is adapted to the adversarial defense (which are known to slow down the convergence). Models (on top of deep features) are trained using LinearSVC from scikit-learn from either clean or poisoned training data. Currently, using up-to-date version of scikit-learn and pytorch is important to recover similar results than the ones reported in the paper (means are averaged at least over 3 runs - but variance is not negligible). Mainly VGG13 results are reported but ResNet ones follow similar trends.

As pointed in section 2.1, adversarial retraining with either FSGM or PGD is considered as adversarial defense. And, as pointed in section 2.2, label flip LF [19], poison frog PF [25] and adversarial poisoning AD [3] are considered for poisoning attack. For poison frog, one has to recover a binary context. Thus,

Dataset	CIFAR	MNIST
AD on naive feature	24%	68%
AD on FSGM feature	30%	93%
AD on PGD feature	**34%**	**95%**

Table 1: Features robustness has positive impact on the accuracy under adversarial poisoning attack [3] on CIFAR and MNIST (here with VGG13).

classes 0 and 1 are considered (so it is airplane vs car on CIFAR contrary to original paper with dog vs fish).

$\varepsilon = \frac{3}{255}$ on CIFAR (except if specified) like in [3], resulting in an invisible perturbation (for human eyes). But $\varepsilon = \frac{7}{255}$ on MNIST which is known to be less prone to adversarial sensibility (except if specified).

4 Results and discussion

4.1 Validation of the implementation

validation of the left part of figure 2: Training with and without adversarial defense produces compatible results with published ones on CIFAR100 and SVHN. Typically, features learnt with adversarial defense are much more robust to adversarial attacks with FSGM and PGD. Currently, there is a bias as the effect of adversarial retraining with PGD is evaluated with PGD attack. However, it is one of the most popular attack today.

validation of the right part of figure 2: The poisoning attacks PF and AD on Imagenet features produces consistent results with [3, 25] (validation of the right part of figure 2). Currently, the implementation of [25] is evaluated in different parameter tuning. Thus, efficient is not directly comparable (73% with the reimplementation against 99% but it is not even the same subset of CIFAR).

4.2 Adversarial poisoning

The table 1 shows the accuracy after an AD attacks on VGG13 with features learnt with or without adversarial defense (for both MNIST with SVHN feature and CIFAR10 with CIFAR100 feature). This table shows that poisoning has less influence on PGD than FSGM, and, less influence on FSGM than on naive feature.

Currently, the clean performance of naive feature is higher than defended ones on CIFAR: accuracy of PGD features on clean CIFAR10 is only 41% i.e. poisoning has almost not effect but starting performances are much lower. However, it mainly show that transferring features from CIFAR100 to CIFAR10 is not a good idea. Inversely, adversarial defenses provide a very efficient protection with a better poisoned accuracy (despite a much lower clean accuracy).

feature	CIFAR
PF on naive feature	85%
PF on FSGM feature	53%
PF on PGD feature	**16%**

Table 2: Critical impact of features robustness on ratio of successful poison frog attacks on CIFAR.

On MNIST, the result is very interesting with a very high accuracy under poisoning with FSGM or PGD features: AD does not work at all on MNIST with PGD feature.

So, adversarial defenses are a data poisoning defense against [3] on MNIST (and mitigate the loss of accuracy related to [3] on CIFAR).

4.3 Poison frog

The table 2 shows the ratio of points (over 100 trial) on which poison frog attack is successful on both naive, FSGM or PGD features on CIFAR with $\varepsilon = \frac{7}{255}$. The number of trial is slower than in [25]. However, it has to be stressed that each trial require to learn a SVM on the top of the features resulting in an expensive process (in particular with 3 different types of features).

Currently, on MNIST, PF works from Imagenet feature, but, not from SVHN features even with $\varepsilon = \frac{25}{255}$. So the MNIST results are not reported. Maybe, the SVHN features are very robust on MNIST (even naive ones) making PF completely ineffective.

Again, this experiment shows that adversarial defense strongly decreases the impact of a data poisoning attack (PF here). Currently, PF is still active even with PGD feature on CIFAR (16% is still an issue) but much less than when targeting naive features (with 85% of successful attacks in this last case).

4.4 Label flip

Both previous subsections shows that adversarial defense improves robustness to two poisoning attacks. Yet, those poisoning attacks are image-based.

Thus, it could be interesting to check if this results holds for label based poisoning attacks. Indeed, as robust features tend to increase distance between point in feature space, it could be even more sensible to label based attack. Currently, [19] combines both label and image perturbation. Yet, image perturbations are not bounded in [19] (see figure 5 and 6 of [19]), so there is no sense to consider norm bounded adversarial defenses against [19]. This is why, I focus on simple LF attack.

The table 3 shows the difference of accuracy (between clean model and poisoned one) after an LF attacks (2% of random label) on VGG13 with features learnt on with or without adversarial defense. The result is that adversarial defense does not increase the sensibility to label noise (accuracy gap is only slightly larger with robust features).

Dataset	MNIST	CIFAR
LF on naive feature	-2%	-7%
LF on PGD feature	-2%	-11%

Table 3: Robust features do not suffer more than naive ones under label flip attack (here 2% of label is random) with VGG13.

Currently, in absolute value, robust features have much lower accuracy on CIFAR (with or without label noise). On MNIST, performance is high for all features i.e. robust features perform like naive ones with or without label noise: accuracy is still 94% with 2% of label noise on MNIST with PGD features (96% without label noise - against 95% and 97% for naive features).

4.5 Conclusion and future works

The main contribution of this paper is to prove that both poison frog and adversarial poisoning lose most of their effectiveness when targeting robust deep features (produced using adversarial defense).

This result is not surprising, but, not trivial because the robustness of the feature could have been useless against poisoning attacks (which modify the training data).

This invites the poisoning community to consider robust networks rather than naive ones when designing poisoning attack (or defense).

The main limitation of this paper is to tackle deep feature instead of deep networks. This is consistent with the selected attacks but should still be improved in future works.

References

[1] Cem Anil, James Lucas, and Roger Grosse. Sorting out lipschitz function approximation. In *International Conference on Machine Learning*, pages 291–301. PMLR, 2019.

[2] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International conference on machine learning*, pages 274–283. PMLR, 2018.

[3] Adrien CHAN-HON-TONG. An algorithm for generating invisible data poisoning using adversarial noise that breaks image classification deep learning. *Machine Learning and Knowledge Extraction*, 1(1):192–204, Nov 2018.

[4] Moustapha M Cisse, Yossi Adi, Natalia Neverova, and Joseph Keshet. Houdini: Fooling deep structured visual and speech recognition models with ad-

versarial examples. In *Advances in Neural Information Processing Systems*, pages 6977–6987, 2017.

[5] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[7] Chris Finlay, Aram-Alexandre Pooladian, and Adam Oberman. The log-barrier adversarial attack: Making effective use of decision boundary information. In *The IEEE International Conference on Computer Vision*, October 2019.

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

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Guy Katz, Clark Barrett, David L Dill, Kyle Julian, and Mykel J Kochenderfer. Reluplex: An efficient smt solver for verifying deep neural networks. In *International Conference on Computer Aided Verification*, pages 97–117. Springer, 2017.

[11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. 2009.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[13] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *International Conference on Learning Representations (ICLR)*, 2017.

[14] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[15] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. *Advances in neural information processing systems*, 2, 1989.

[16] Alexander Levine and Soheil Feizi. Deep partition aggregation: Provable defense against general poisoning attacks. *arXiv preprint arXiv:2006.14768*, 2020.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[18] Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[19] Luis Muñoz-González, Battista Biggio, Ambra Demontis, Andrea Paudice, Vasin Wongrassamee, Emil C Lupu, and Fabio Roli. Towards poisoning of deep learning algorithms with back-gradient optimization. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*. ACM, 2017.

[20] N. Narodytska and S. Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, July 2017.

[21] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. 2011.

[22] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 427–436, 2015.

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

[24] Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In *2016 IEEE symposium on security and privacy (SP)*, pages 582–597. IEEE, 2016.

[25] Ali Shafahi, W Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks. *arXiv preprint arXiv:1804.00792*, 2018.

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[27] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *technical report arxiv:1312.6199*, 2013.

[28] Eric Wong and Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning*, pages 5286–5295. PMLR, 2018.

[29] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Yuyin Zhou, Lingxi Xie, and Alan Yuille. Adversarial examples for semantic segmentation and object detection. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.