Less is More: Culling the Training Set to Improve Robustness of Deep Neural Networks

Yongshuai Liu[⋆], Jiyu Chen[*], and Hao Chen

University of California, Davis
{yshliu, jiych, chen}@ucdavis.edu

Abstract. Deep neural networks are vulnerable to adversarial examples. Prior defenses attempted to make deep networks more robust by either changing the network architecture or augmenting the training set with adversarial examples, but both have inherent limitations. Motivated by recent research that shows that outliers in the training set have a high negative influence on the trained model, we studied the relationship between model robustness and the quality of the training set. We propose two methods for detecting outliers based on canonical examples and on training errors, respectively. After removing the outliers, we trained the classifier with the remaining examples to obtain a *sanitized* model. We evaluated the sanizied model on MNIST and SVHN and found that it forced the attacker to generate adversarial examples with much higher distortion. More importantly, we examined the Kullback-Leibler divergence from the output of the original model to that of the sanitized model and found that this divergence is much higher for adversarial examples than normal examples. Based on this difference, we could detect adversarial examples with accuracy between 94.67% to 99.89%. Our results show that improving the quality of the training set is a promising direction for increasing model robustness.

1 Introduction

Deep neural networks have demonstrated impressive performance on many hard perception problems [8,11]. However, they are vulnerable to adversarial examples [17,6,15], which are maliciously crafted to be perceptually close to normal examples but which cause misclassification. Prior defenses against adversarial examples fall into the following categories: 1. Incorporating adversarial examples in the training set, a.k.a. adversarial training [17,6] 2. Modifying the network architecture or training method, e.g., defense distillation [16] 3. Modifying the test examples, e.g., MagNet [13] The first defense requires knowledge about the process for generating adversarial examples, while the last two defenses require high expertise and are often not robust [2].

We propose a new direction to strengthen deep neural networks against adversarial examples. Recent research showed that outliers in the training set are

[⋆] Equal contribution

highly influential on the trained model. For example, outliers may be ambiguous images on which the model has low confidence and thus high loss [7]. Our insight is that if we can detect and discard outliers in the training set, we can make the model more robust against adversarial examples without changing either the network architecture or training method. We call the process of removing outliers from the training set *sanitization.*[1]

We propose two methods for detecting outliers. First, for some AI tasks, we may find canonical examples. For example, for handwritten digit classification, we may use computer fonts as canonical examples. We trained a *canonical model* using canonical examples, and then used the canonical model to detect outliers in the training set. We call this method *canonical sanitization.* Second, for AI tasks without canonical examples, we considered examples with large training errors as outliers. We call this method *self sanitization.*

After culling the training set to remove outliers, we trained a model called the *sanitized model.* We compared the robustness of the unsanitized model, which was trained on the entire training set, with the sanitized model on adversarial examples using two criteria. The first criterion is classification accuracy. Our evaluation showed that the sanitized model increased classification accuracy considerably. For example, on adversarial examples generated by iterative gradient sign method after five iterations on MNIST, the canonical sanitized model increased the classification accuracy from 53.3% to 82.8%, and the self sanitized model from 53.3% to 92.6%.

The second criterion is distortion increment. When we generated adversarial examples on MNIST with the Carlini & Wagner attack whose confidence was set to zero, the average L^2 distortion increased from 2.1 to 2.55 in the canonical sanitized model and to 2.63 in the self sanitized model.

More importantly, the sanitized models allowed us to detect adversarial examples. We computed the Kullback-Leibler divergence from the output of an example on the unsanitized model to the output of the same example on the sanitized model, and found that this divergence was much larger for adversarial examples than for normal examples. Based on this difference, we were able to detect the adversarial examples generated by the Carlini & Wagner attack on MNIST and SVHN at 99.26% and 94.67% accuracy, respectively. Compared to prior work for detecting adversarial examples (e.g., [14]), this approach requires no knowledge of adversarial examples.

We make the following contributions.

- We propose a new direction to improve the robustness of deep neural networks against adversarial examples. This approach detects and removes outliers from the training set without changing the classifier.
- We propose two methods for detecting outliers in the training set: canonical sanitization and self-sanitization. We found that the sanitized model increased either the classification accuracy or the distortion of adversarial examples.

[1] Unlike *data sanitization*, which commonly modifies individual datum, we modify no example but merely remove outliers from the training set.

- We propose an approach for detecting adversarial examples by leveraging the Kullback-Leibler divergence from the unsanitized model to the sanitized model.

2 Methodology

2.1 Definitions

Examples

- *Normal examples* are sampled from the natural data generating process. For examples, images of handwritten digits.
- *Outliers* are examples in the training set of normal examples. They are difficult to classify by humans. *Sanitization* is the process to remove outliers from the training set.
- *Adversarial examples* are crafted by attackers that are perceptually close to normal examples but that cause misclassification.

Models

- *Unsanitized models* are trained with all the examples in the training set. We assume that the training set contains only normal examples (i.e., without adversarial examples).
- *Sanitized models* are trained with the remaining examples after we remove outliers from the training set.

Our goal is to evaluate if sanitized models are more robust against adversarial examples. A model is more robust if it increases classification accuracy or the distortion of adversarial examples, or can be used to detect adversarial examples.

2.2 Sanitization

Sanitization is the process of removing outliers from the training set. Since manual sanitization would be laborious and subjective, we propose two automatic sanitization methods.

Canonical sanitization This approach applies to the AI tasks that have canonical examples. For example, for handwritten digit recognition, most computer fonts may be considered canonical examples.[2]. If a handwritten digit is similar to a computer font, it can be easily classified by a human. Based on this observation, we use canonical examples to discard outliers in our training set $\mathbb{X}$ by the following steps:

[2] Some computer fonts are difficult to recognize and therefore are excluded from our evaluation

- Augment the set of canonical examples by applying common transformations, e.g., rotating and scaling computer fonts.
- Train a model f using the augmented canonical examples.
- Use f to detect and discard outliers in the training set $\mathbb{X}$. An example $\boldsymbol{x}^{(i)}$ is an outlier if $f(\boldsymbol{x}^{(i)})$ has a low confidence on $y^{(i)}$, the class associated with $\boldsymbol{x}^{(i)}$.

Self sanitization Not all AI tasks have canonical examples. For such tasks, we use all the examples to train a model, and then discard examples that have high training errors.

2.3 Sanitized models

After removing the outliers from the original training set, we get a sanitized set and use it to train a model, called the *sanitized model.* Then, we evaluate if the sanitized model is more robust against adversarial examples than the unsanitized models using two metrics: classification accuracy and distortion of adversarial examples.

2.4 Detecting adversarial examples

We detect adversarial examples based on the Kullback-Leibler divergence [9] from the output of an example on the unsanitized model to that of the same example on the sanitized model. The Kullback-Leibler divergence from a distribution P to Q is defined as

$$D_{\mathrm{KL}}\left(P \parallel Q\right) = \sum_i P\left(i\right) \log \frac{P\left(i\right)}{Q\left(i\right)}$$

3 Evaluation

3.1 Set up

Data sets We used two data sets: MNIST and SVHN.

MNIST [3]

MNIST contains 60 000 and 10 000 monochrome images of handwritten digits in the training and test set, respectively. MNIST is attractive because we can use computer fonts as the canonical examples. This allows us to apply canonical sanitization (Section 3.2).

[3] http://yann.lecun.com/exdb/mnist/

SVHN [4]

To evaluate on a data set more complex than MNIST, we selected SVHN, which consists of color images of street view house numbers. SVHN is more challenging because the images are much more diverse, e.g., in different colors, orientations, and lighting conditions.

There are two formats of SVHN:

1. Original images with character-level bounding boxes.
2. MNIST-like 32-by-32 images with a single character in the center.

The second format is noisy because many images contain parts of the adjacent digits in addition to the center digit. Therefore, we used the first format to create images of individual digits as follows:

1. Cropping individual digits using the bounding boxes.
2. Discarding images whose either dimension is less than 10, because such tiny images are difficult to classify even for humans.
3. Resizing the larger dimension of each each image to 28 while keeping the aspect ratio, and then padding the image to 28×28. When padding an image, we used the average color of the border as the padding color.

After the process, we obtained 40 556 images from the original training set and 9790 test images from the original test set.

Models

MNIST We trained the Convolutional Neural Network (CNN) in Figure 1 on MNIST and achieved an accuracy of 99.3%.

SVHN Since SVHN is more complex, we used a more complex model in Figure 2. It achieved an accuracy of 98.62%.

Both the above models are *unsanitized model*.

Attacks We used two popular attacks.

- *Iterative Gradient Sign Method (IGSM)*. Given a normal image x, fast gradient sign method (FGSM) looks for a similar image x' in the L^∞ neighborhood of x that fools the classifier [6]. [10] improved FGSM by using a finer iterative optimization strategy. For each iteration, the attack performs FGSM with a smaller step-width α, and clips the updated result so that the updated image stays in the ϵ neighborhood of x. We ran this attack with 5, 10, and 15 iterations, respectively.
- *Carlini & Wagner's attack*. For a fixed input image x, the attack looks for a perturbation δ that is small in length, satisfies the box constraints to be a valid image, and fools the classifier at the same time [2]. When running this attack, we used L^2 as the metric, chose several confidence levels, and performed targeted attacks.

[4] http://ufldl.stanford.edu/housenumbers/

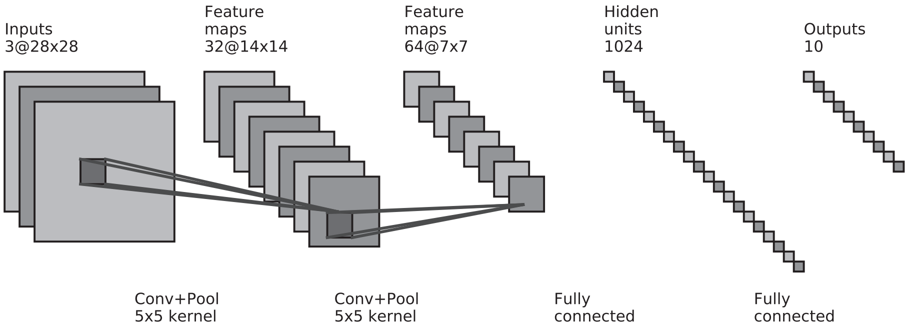


Fig. 1: Convolutional Neural Network for MNIST

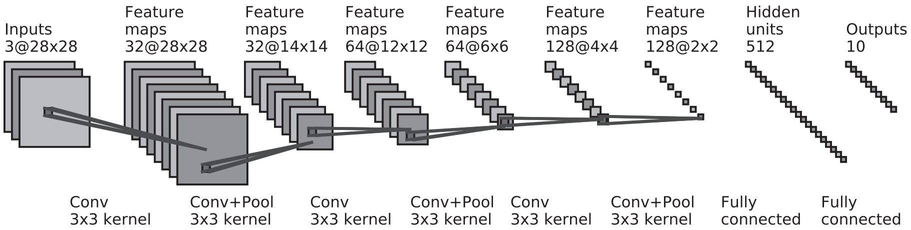


Fig. 2: Convolutional Neural Network for SVHN

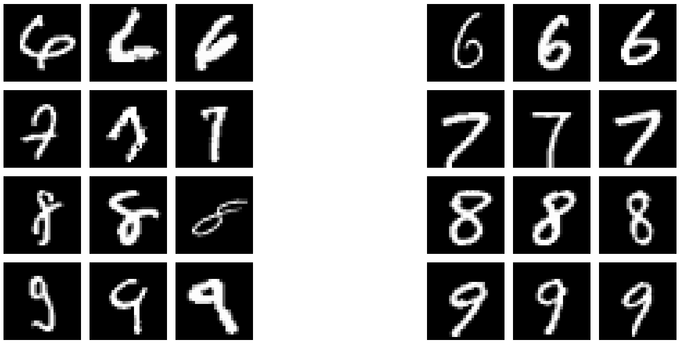
3.2 Sanitization

Canonical sanitization We did canonical sanitization on MNIST. In canonical sanitization, we discarded outliers that are far different from canonical examples. We chose fonts containing digits on Windows 10 as canonical examples. After removing nine fonts that are difficult to recognize by humans, we were left with 340 fonts. To accommodate common variations in handwriting, we applied the following transformations to the fonts:

- *Scaling*. We rendered each font in 24 pt, 28 pt, and 32 pt.
- *Rotation*. We rotated each font from $-30°$ to $30°$ in an increment of $10°$.

We did not translate the fonts because CNN is insensitive to translation. After the above transformations, we acquired $340 \times 3 \times 7 \times 10 = 71\,400$ images, from which we randomly chose 80% as the training set and used the remaining 20% as the test set. We trained the classifier in Figure 1 and achieved an accuracy of 98.7%. We call this the *canonical model*.

We used the canonical model to test the MNIST training set. The classification accuracy is 88%. Although this accuracy is not stellar (the state of the art classifier achieved more than 99% accuracy on MNIST), it is nevertheless impressive because the canonical model was trained on computer fonts while MNIST examples were hand written. This shows that computer fonts are good approximations of handwritten digits.



(a) Confidence < 0.7

(b) Confidence > 0.999 99

Fig. 3: Examples in MNIST with low and high confidence, respectively, on the canonical model trained with computer fonts

Using computer fonts as canonical examples, we discarded outliers in the MNIST training set. To do this, we fed each example to the canonical model. If the example's confidence score (the element corresponding to the correct class in the model's output vector) was below a threshold, we considered it an outlier and discarded it. Figure 3 shows examples with low and high confidence. Table 1 shows the number of examples left under different thresholds. We used these examples to train the *sanitized models*.

Table 1: Size of the MNIST training set after discarding examples whose confidence scores on the canonical model are below a threshold

Threshold	0	0.7	0.8	0.9	0.99	0.999	0.9999
Set size	60 000	51 241	50 425	49 128	44 448	38 230	30 618

We did not do canonical sanitization on SVHN because it is infeasible to acquire a reasonably-sized set of canonical examples, as the digits in SVHN are highly diverse in colors, orientations, lighting conditions, etc.

Self-sanitization We did self sanitization on both MNIST and SVHN. To discard outliers in self sanitization, we trained the classifier in Figure 1 and Figure 2 for MNIST and SVHN separately, used the models to test every example in the training set, and considered examples whose confidence scores were below a threshold as outliers. Table 2 and Table 3 show the number of examples left under different thresholds. We used these examples to train the *sanitized models*. Table 3 also shows that the sanitized models maintain high classification accuracy when it has adequate training data to prevent overfitting.

Table 2: Size of the MNIST training set after discarding examples whose confidence scores on the self-trained model are below a threshold

Threshold	0	0.999	0.9999	0.999 99	0.999 999	0.999 999 9
Set size	60 000	56 435	52 417	45 769	36 328	24 678

Table 3: The sizes of the sanitized SVHN training set and the classification accuracy of the sanitized models at different thresholds for discarding outliers from the original SVHN training set

Threshold	Training set size	Classification accuracy (%)
0	40 556	94.26
0.7	39 330	93.68
0.8	38 929	93.22
0.9	38 153	92.74
0.99	34 408	91.30
0.999	28 420	89.41
0.9999	21 378	85.82
0.999 99	14 000	85.29
0.999 999	8043	79.56
0.999 999 9	4081	58.14

3.3 Robustness against adversarial examples

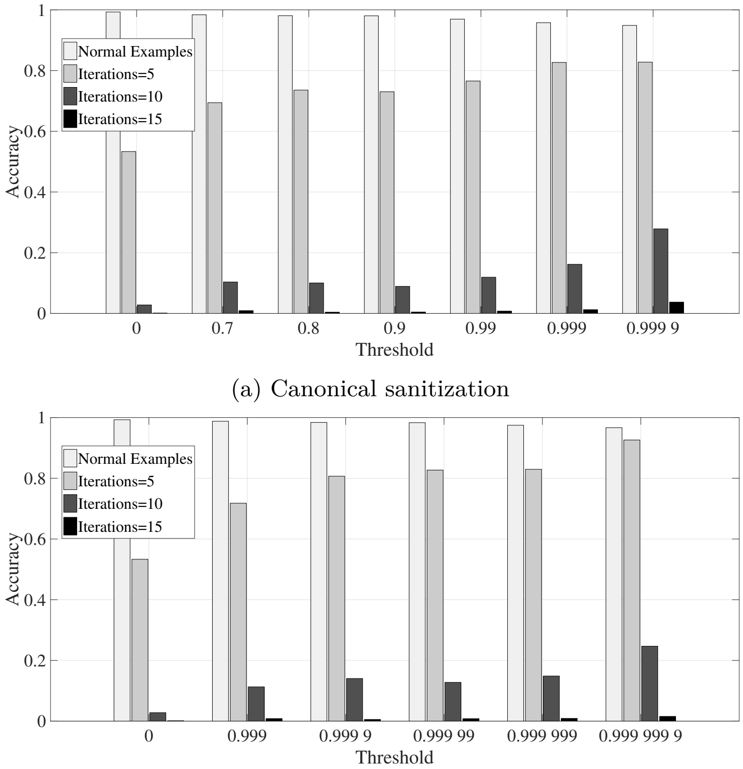
Table 4: Trained models. We trained no canonical sanitized model on SVHN because it is infeasible to construct a reasonably-sized set of canonical examples.

Data set	Number of models		
	Unsanitized	Canonical sanitized	Self sanitized
MNIST	1	6	5
SVHN	1	—	9

Table 4 shows all models that we trained. We ran the IGSM and Carlini & Wagner attacks on both the unsanitized and sanitized models.

IGSM attack Figure 4 compares the classification accuracy of the unsanitized and sanitized models on the adversarial examples generated by the IGSM attack on MNIST, where Figure 4a and Figure 4b correspond to canonical sanitization and self sanitization, respectively.

Figure 4 shows that a higher threshold of sanitization increases the robustness of the model against adversarial examples. For example, on adversarial examples generated after five iterations of IGSM, the classification accuracy is 82.8% with a



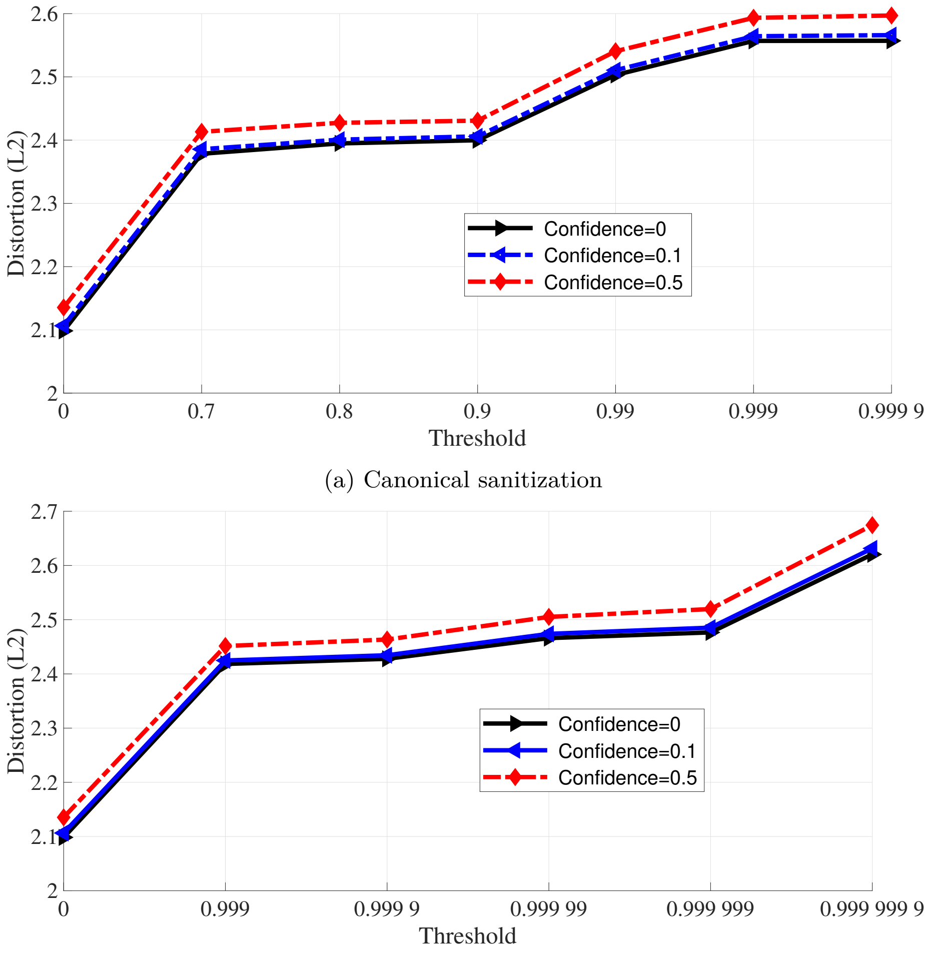
(a) Canonical sanitization

(b) Self sanitization

Fig. 4: Classification accuracy of models on the adversarial examples generated by the IGSM attack. We trained these models using the examples in MNIST after sanitization based on different thresholds, where the threshold 0 represents the full original data set.

threshold of 0.9999 in canonical sanitization, and is above 92.6% with a threshold of 0.999 999 9 in self sanitization. Although the accuracy decreases after more iterations of IGSM, higher thresholds still significantly improve classification accuracy.

Carlini & Wagner's attack We ran Carlini & Wagner's L^2 target attack to generate adversarial examples on our sanitized models on both MNIST and SVHN. Although this attack generated adversarial examples that fooled all of our models, Figure 5 and Figure 6 show that the sanitized models forced the adversarial examples to have larger distortions. The higher the threshold, the larger the distortion.



(a) Canonical sanitization

(b) Self sanitization

Fig. 5: Average L^2 distortions of adversarial examples generated by Carlini & Wagner's attack on models trained using the examples in MNIST after sanitization based on different thresholds, where the threshold 0 represents the unsanitized model.

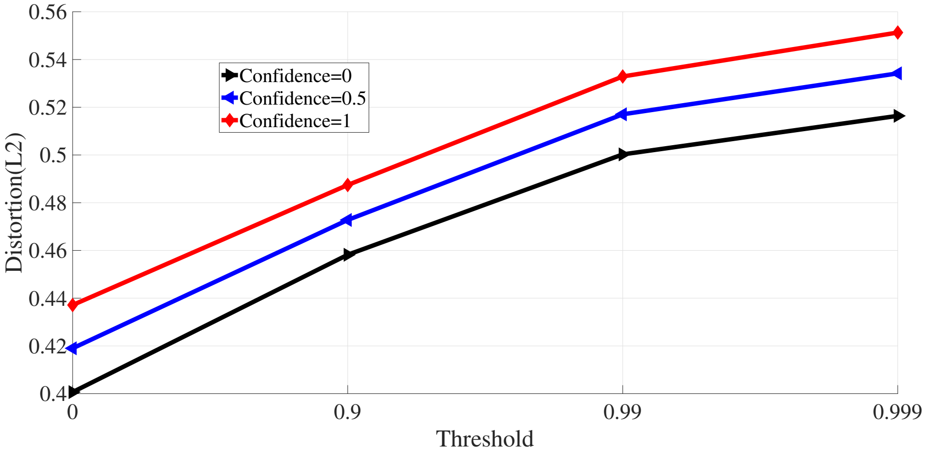


Fig. 6: Average L^2 distortions of adversarial examples generated by the Carlini & Wagner attack on models trained using the examples on SVHN after sanitization based on different thresholds, where the threshold 0 represents the unsanitized model.

3.4 Detecting adversarial examples

We evaluated the effectiveness of using the Kullback-Leibler divergence to detect adversarial examples (Section 2.4).

MNIST We generated adversarial examples on two sanitized models on MNIST:

- A canonical sanitized model. We collected the training examples of this sanitized model from the MNIST training set after discarding all the examples whose confidence levels on the canonical model (trained on computer fonts) were below 0.9999.
- A self sanitized model. We collected the training examples of this sanitized model from the MNIST training set after discarding all the examples whose confidence levels on the original model (trained on the entire training set) were below 0.999 999 9.

We computed the Kullback-Leibler divergence from the output of the unsanitized model to that of each of the sanitized models.

Figure 7 compares the CDF of the Kullback-Leibler divergence between normal examples and adversarial examples generated by IGSM after different iterations. It shows that the majority of normal examples have very small divergence, while most adversarial examples have large divergence where more iterations generated examples with higher divergence. Figure 8 compares the CDF of the Kullback-Leibler divergence between normal examples and adversarial examples generated by the Carlini & Wagner attack using different confidence levels. The

Attack	IGSM (iterations)			Carlini & Wagner (confidence)					
Attack parameter	5	10	15	0	0.1	0.5	1	3	5
Detection accuracy (%)	33.80	85.47	96.31	99.26	99.26	100.00	99.26	98.52	95.56

Table 5: MNIST: Accuracy of detecting adversarial examples based on the Kullback-Leibler divergence from the unsanitized model to a canonical sanitized model when detection accuracy for normal examples is 98%.

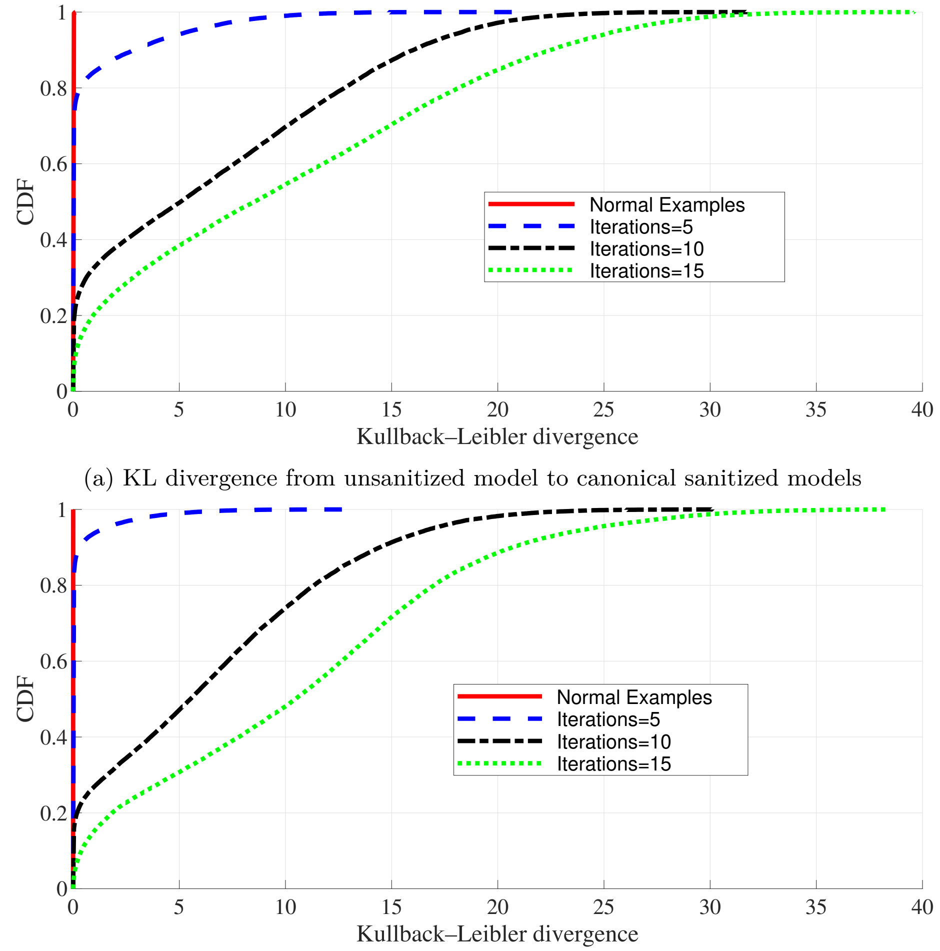
difference is even more prominent, which motivated us to detect adversarial examples by this difference.

Table 5 shows the accuracy of detecting adversarial examples based on the Kullback-Leibler divergence from the unsanitized model to a canonical sanitized model. We used a threshold of KL divergence to divide normal and adversarial examples: all the examples below this threshold were considered normal, and all above adversarial. We determined the threshold by setting a target detection accuracy on normal examples. For example, when we set this target accuracy to 98%, we needed a threshold of KL divergence of 0.0068. At this threshold, the accuracy of detecting all the Carlini & Wagner adversarial examples at all the confidence levels is high (all above 95%). The accuracy of detecting IGSM adversarial examples is high when the number of iterations is high (e.g., 10 or 15). When the number of iterations is low (e.g., 5), the detection accuracy decreases; however, since the false negative examples have low KL divergence, they are more similar to normal examples and therefore can be classified correctly with high probability as discussed next.

To take advantage of both the KL divergence for detecting adversarial examples and the sanitized models for classifying examples, we combined them into a system shown in Figure 9. The system consists of a detector, which computes the KL divergence from the unsantized model to the sanitized model and rejects the example if its divergence exceeds a threshold, and a classifier, which infers the class of the example using the sanitized model. The system makes a correct decision on an example when

– if the example is normal, the detector decides the example as normal *and* the classifier correctly infers its class.
– if the example is adversarial, the detector decides the example as adversarial *or* the classifier correctly infers its true class.

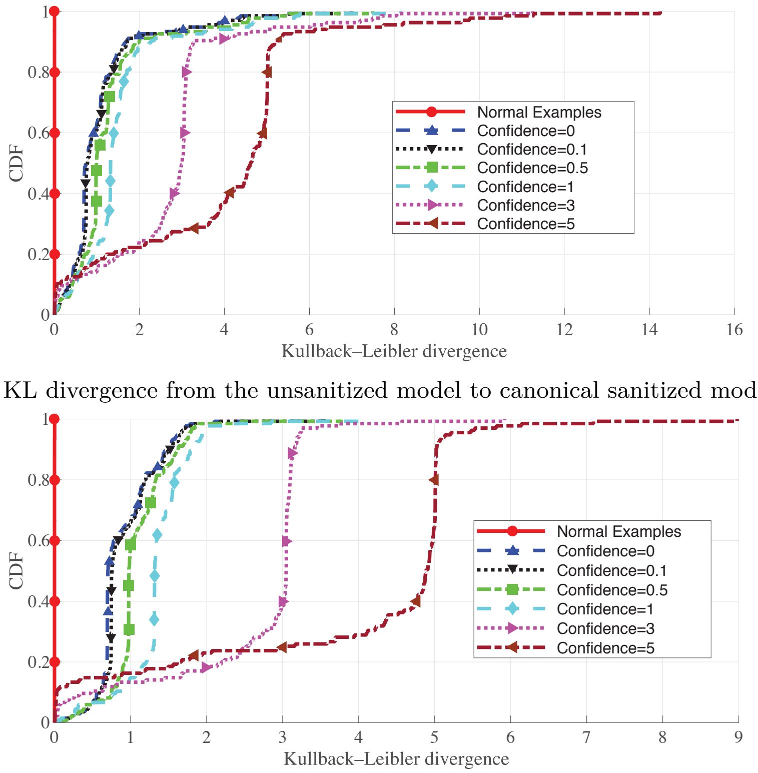
Table 6 shows the accuracy of this system on adversarial exampled generated by the IGSM attack on a canonical sanitized model on MNIST. At each tested iteration of the IGSM attack, the accuracy of this system on the adversarial examples is above 94%. The accuracy of this system on the normal examples is 94.8%.



(a) KL divergence from unsanitized model to canonical sanitized models

(b) KL divergence from unsanitized model to self sanitized models

Fig. 7: MNIST: CDF of Kullback-Leibler divergence from the output of the unsanitized model to the output of a sanitized model. We generated the adversarial examples using the IGSM attack after 5, 10, and 15 iterations, respectively.



(a) KL divergence from the unsanitized model to canonical sanitized models

(b) KL divergence from unsanitized model to self sanitized models

Fig. 8: MNIST: CDF of Kullback-Leibler divergence from the output of the unsanitized model to the output of a sanitized models. We generated the adversarial examples at different confidence levels.

SVHN Figure 10 compares the CDF of the Kullback-Leibler divergence of normal examples and adversarial examples generated by the Carlini & Wagner attack at different confidence levels. We trained the sanitized model after discarding the examples in the training set whose confidence on the original model was below 0.9 (self sanitization).

Table 7 shows the impact of sanitization threshold on the detection accuracy on adversarial examples generated by the Carlini & Wagner attack at different confidence levels. We automatically determined the threshold of KL divergence by setting the detection accuracy on normal examples to 94%. Table 7 shows that as the sanitization threshold increases from 0.7 to 0.9, the detection accuracy increases. However, after the sanitization threshold increases even further, the detection accuracy decreases. This is because after the sanitization threshold exceeds 0.9, the size of the training set decreases rapidly, which causes the model to overfit.

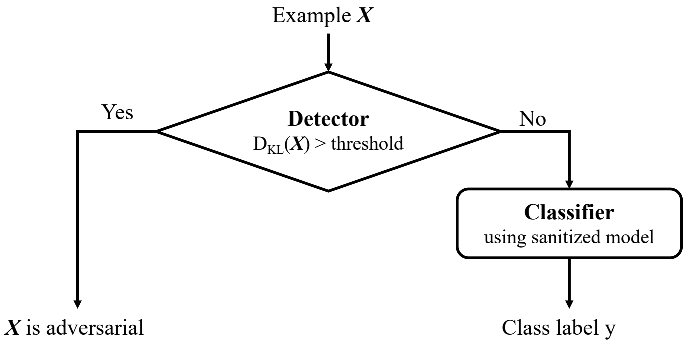


Fig. 9: A system combining a detector, which detects if an example is adversarial based on the KL divergence from the unsantized model to the sanitized model, and a classifier, which classifies the example using the sanitized model.

Table 6: Accuracy of the system in Figure 9 on adversarial examples generated by the IGSM on a canonical sanitized model on MNIST. The column "detection", "classifier", and "combined" shows the accuracy of the detector, classifier, and system overall. The overall accuracy on normal examples is 94.8%.

IGSM iterations	Accuracy (%)		
	Detector	Classifier	Overall
5	33.80	99.84	99.89
10	85.47	72.72	96.03
15	96.31	1.71	94.68

3.5 Canonical vs. self sanitization

We proposed two methods for detecting outliers: canonical sanitization and self sanitization. Section 3.3 and Section 3.4 show that their effectiveness differs somewhat on classification accuracy and adversarial example detection. Here we compare how many outliers they detect in MNIST. Figure 11 shows the proportion of each digit in the remaining images after removing outliers according to different thresholds.

Figure 11a and Figure 11b show similarities between canonical and self sanitization. For example, as the threshold increases, the proportion of both 0 and 6 increases while the proportion of both 1 and 9 decreases. However, the figures also show differences. As the threshold increases, the portion of 8 decreases in canonical sanitization but increases in self sanitization. We conjecture that the difference between average computer fonts and average handwritten fonts is higher for 8 than for the other digits.

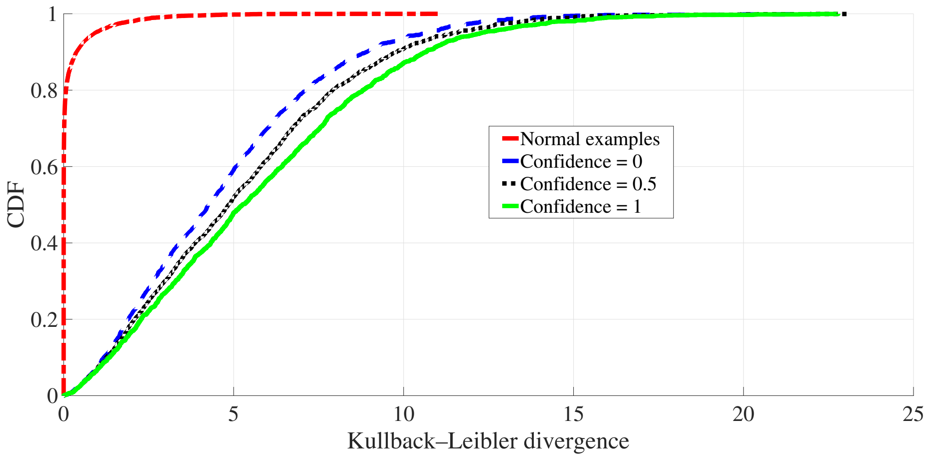


Fig. 10: SVHN: CDF of the KL divergence from the output of the unsanitized model to that of a sanitized models. Normal examples have small divergence, while all the Carlini & Wagner adversarial examples under difference confidence levels have large divergence.

4 Discussion and future work

Section 3.4 showed that the Kullback-Leibler divergence from the unsanitized model to the sanitized model is much larger on adversarial examples than on normal examples. This suggests that the distribution of adversarial examples is much closer to the outliers in the normal examples than to the rest (non-outliers) of the normal examples. This supports our initial hypothesis that outliers distort the model's learned distribution and facilitate adversarial examples.

Section 3.4 showed that we can use the Kullback-Leibler divergence as a reliable metric to distinguish between normal and adversarial examples. The threat model was that the attacker can access both the unsanitized and the sanitized models as white boxes. She could generate adversarial examples on either model, but she did not know that we used the Kullback-Leibler divergence to detect adversarial examples. In our future work, we plan to evaluate if the attacker can generate adversarial examples to evade our detection if she knows our detection method.

5 Related work

Most prior work on machine learning security focused on improving the network architecture, training method, or incorporating adversarial examples in training [1]. By contrast, we focus on culling the training set to remove outliers to improve the model's robustness.

Sanitization threshold	Training set size	KL divergence threshold	Detection accuracy (%) Attack confidence		
			0	0.5	1
0.7	39 330	0.7295	93.50	94.56	94.67
0.8	38 929	0.5891	93.56	94.67	95.72
0.9	38 153	0.7586	94.67	94.78	95.78
0.99	34 408	1.0918	90.00	91.17	92.56
0.999	28 420	1.6224	83.22	85.78	87.33

Table 7: SVHN: the impact of sanitization threshold on the accuracy of detecting adversarial examples based on the KL divergence from the unsanitized model to sanitized models

5.1 Influence of training examples

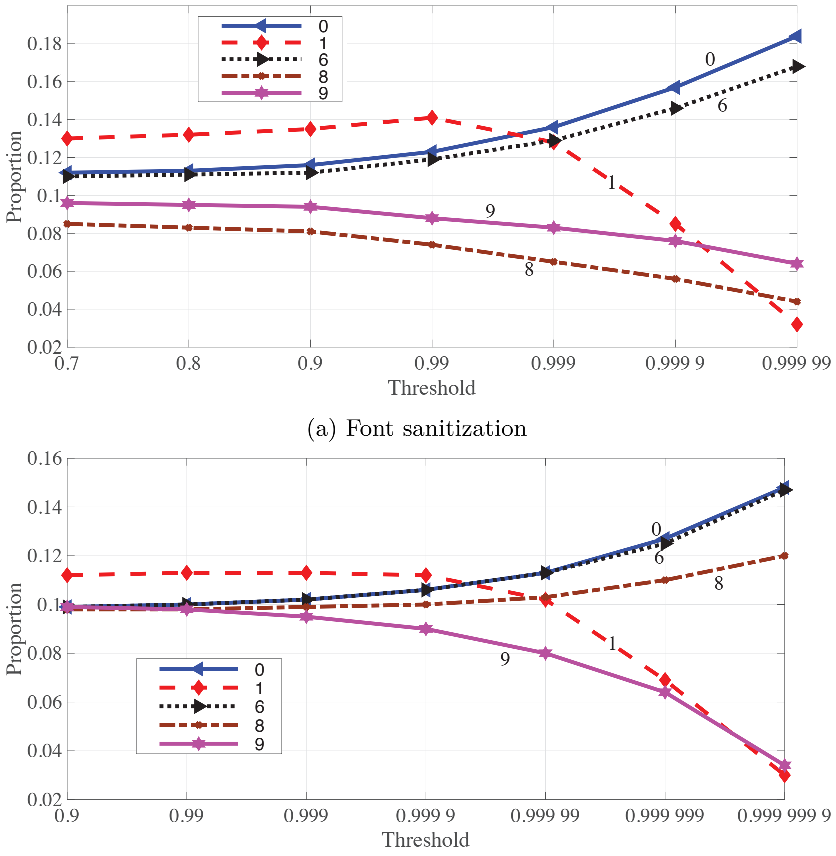
Influence functions is a technique from robust statistics to measure the estimator on the value of one of the points in the sample [5,18]. Koh et al. used influence functions as an indicator to track the behavior from the training data to the model's prediction [7]. By modifying the training data and observing its corresponding prediction, the influence functions can reveal insight of model. They found that some ambiguous training examples were effective points that led to a low confidence model. Influence Sketching [19] proposed a new scalable version of Cook's distance [3,4] to prioritize samples in the generalized linear model [12]. The predictive accuracy changed slightly from 99.47% to 99.45% when they deleted approximately 10% ambiguous examples from the training set.

5.2 Influence of test examples

Xu et al. [20] observed that most features of test examples are unnecessary for prediction, and that superfluous features facilitate adversarial examples. They proposed two methods to reduce the feature space: reducing the color depth of images and using smoothing to reduce the variation among pixels. Their feature squeezing defense successfully detected adversarial examples while maintaining high accuracy on normal examples.

6 Conclusion

Adversarial examples remain a challenging problem despite recent progress in defense. We studied the relationship between outliers in the data set and model robustness, and proposed methods for improving model robustness and for detecting adversarial examples without modifying the model architecture. We proposed two methods to detect and remove outliers in the training set and used



(a) Font sanitization

(b) Self sanitization

Fig. 11: Composition of examples from different classes in the remaining training set after removing outliers based different sanitization methods and thresholds

the remaining examples to train a sanitized model. On both MNIST and SVHN, the sanitized models significantly improved the classification accuracy on adversarial examples generated by the IGSM attack and increased the distortion of adversarial examples generated by the Carlini & Wagner attack. Furthermore, we used the Kullback-Leibler divergence from the unsanitized model to the sanitized model to detect adversarial examples reliably. Our results show that improving the quality of the training set is a promising direction for increasing model robustness.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant No. 1801751.

References

1. Carlini, N., Wagner, D.: Adversarial examples are not easily detected: Bypassing ten detection methods. arXiv preprint arXiv:1705.07263 (2017)
2. Carlini, N., Wagner, D.: Towards evaluating the robustness of neural networks. In: IEEE Symposium on Security and Privacy (2017)
3. Cook, R.D.: Detection of influential observation in linear regression. Technometrics **19**(1), 15–18 (1977)
4. Cook, R.D.: Influential observations in linear regression. Journal of the American Statistical Association **74**(365), 169–174 (1979)
5. Cook, R.D., Weisberg, S.: Characterizations of an empirical influence function for detecting influential cases in regression. Technometrics **22**(4), 495–508 (1980)
6. Goodfellow, I.J., Shlens, J., Szegedy, C.: Explaining and harnessing adversarial examples. In: International Conference on Learning Representations (ICLR) (2015)
7. Koh, P.W., Liang, P.: Understanding black-box predictions via influence functions. In: International Conference on Machine Learning (2017)
8. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. pp. 1097–1105 (2012)
9. Kullback, S., Leibler, R.A.: On information and sufficiency. The annals of mathematical statistics **22**(1), 79–86 (1951)
10. Kurakin, A., Goodfellow, I.J., Bengio, S.: Adversarial examples in the physical world. CoRR **abs/1607.02533** (2016)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE **86**(11), 2278–2324 (1998)
12. Madsen, H., Thyregod, P.: Introduction to general and generalized linear models. CRC Press (2010)
13. Meng, D., Chen, H.: MagNet: a two-pronged defense against adversarial examples. In: ACM Conference on Computer and Communications Security (CCS). Dallas, TX (2017)
14. Metzen, J.H., Genewein, T., Fischer, V., Bischoff, B.: On detecting adversarial perturbations. In: International Conference on Learning Representations (ICLR) (2017)
15. Moosavi-Dezfooli, S.M., Fawzi, A., Frossard, P.: Deepfool: a simple and accurate method to fool deep neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2574–2582 (2016)
16. Papernot, N., McDaniel, P., Wu, X., Jha, S., Swami, A.: Distillation as a defense to adversarial perturbations against deep neural networks. In: IEEE Symposium on Security and Privacy (2016)
17. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I.J., Fergus, R.: Intriguing properties of neural networks. In: International Conference on Learning Representations (ICLR) (2014)
18. Weisberg, S., Cook, R.D.: Residuals and influence in regression (1982)

19. Wojnowicz, M., Cruz, B., Zhao, X., Wallace, B., Wolff, M., Luan, J., Crable, C.: "influence sketching": Finding influential samples in large-scale regressions. In: Big Data (Big Data), 2016 IEEE International Conference on. pp. 3601–3612. IEEE (2016)
20. Xu, W., Evans, D., Qi, Y.: Feature squeezing: Detecting adversarial examples in deep neural networks. In: Network and Distributed Systems Security Symposium (NDSS) (2018)