B20AI013-Minor1

Note: I have put a lot of work into the submission and learnt a lot in the process. *all of the code was written from scratch.* Please evaluate accordingly.

1. Introduction

Autoencoder networks are often used to discriminate and detect adversarially perturbed examples, since The reconstruction error of the affected images is higher than that of the benign samples. This suffers from one crucial flaw: the adversarial perturbations are fundamentally designed so that the perturbation in the image is minimal, hence the perturbation in the reconstructed image should reflect that as well. This is a result of the too-strong ability of the autoencoder to generalise to input samples.

Adversarial samples mostly behave in such a manner as to cause a misclassification (wrong label) without changing the semantics of the benign counterparts.

This paper proposes a method to limit the generalization capability by utilizing the disentangled nature of the semantic(encoder) and the label(victim feature extractor) embeddings in order to generate counterexamples that simulate a general adversarial image. These counterexamples are used along with the benign samples with a carefully chosen loss function in order to train the autoencoder against a wide range of adversarial attacks.

2. Methodology

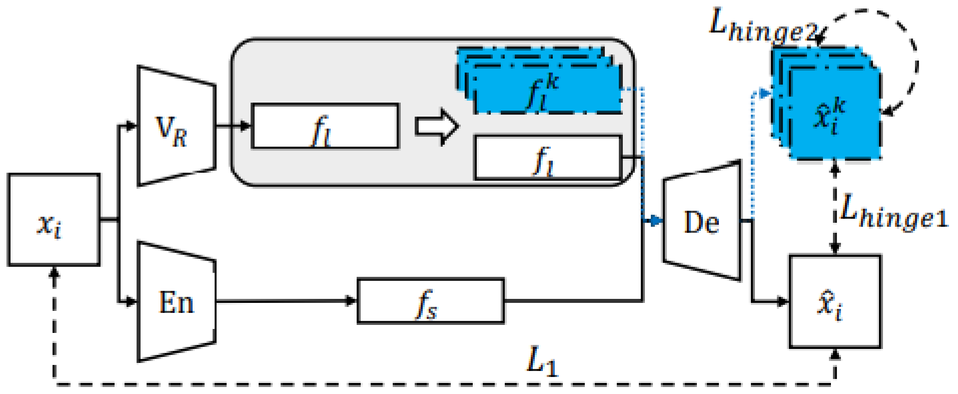


Fig. 2: Overview of DRR.

First and foremost, the input images x_i are passed through the feature extractor of the classifier network, the Victim V, and the Encoder of our approach, En, to obtain label and semantic features f_l and f_s respectively.

$$f_l = softmax(V(x_i)/T)$$
$$f_s = softmax(En(x_i)/T)$$

The decoder network reconstructs the input samples x_i as

$$\hat{x_i} = De(f_l, f_s)$$

To obtain counterexamples, we first randomly relocate the largest element in f_l from index i to k, i.e.,

$$f_{l_i}^k = Relocate(f_{l_i}, i, k), k \neq i$$

We then decode the permuted label feature $f_{l_i}^k$ and the original semantic feature f_{s_i} to obtain counterexamples by

$$\hat{x_i^k} = De(f_{s_i}, f_{l_i}^k)$$

There are two points to keep in mind:

- The decoded $\hat{x_i^k}$ should not converge to $\hat{x}_i$ as that would indicate that the adversarially perturbed samples are being mixed with the benign samples which beats the original purpose.
- The decoded $\hat{x_i^k}$ should not be too far away from $\hat{x}_i$ as that would indicate overfitting.

Logically, it can be seen that, $\hat{x_i^{k_1}}$ and $\hat{x_i^{k_2}}$ should similarly obey the above requirements if $k_1 \neq k_2$.

Thus, the loss function is proposed as:

$$\begin{aligned} L_1 &= \mathbb{E}_{x_i}\mathrm{MAE}(x_i, \hat{x}_i) \\ L_{hinge1} &= \mathbb{E}_{x_i}\left[\max\left(0, d - min_{k_1,k_2\in\Sigma,k_1\neq k_2}(\mathrm{MAE}(\hat{x_i^{k_1}}, \hat{x_i^{k_2}}))\right)\right] \\ L_{hinge2} &= \mathbb{E}_{x_i}\left[\max\left(0, d - min_{k\in\Sigma,k\neq i}(\mathrm{MAE}(\hat{x_i^{k}}, x_i))\right)\right] \\ L_{hinge} &= L_{hinge1} + L_{hinge2} \\ Loss &= \lambda L_1 + L_{hinge} \end{aligned}$$

where d is a hyperparameter that controls the threshold between counterexamples and benign examples, and λ controls the relative importance of the loss functions. Σ is the size of the set of counterexamples x_i^k used for training with each sample x_i.

Detection of adversarial examples is done by choosing a threshold using ROC and AUC curves for a given set of attacks, and if the reconstruction error for an input sample is higher than it, then the sample is said to be adversarially perturbed.

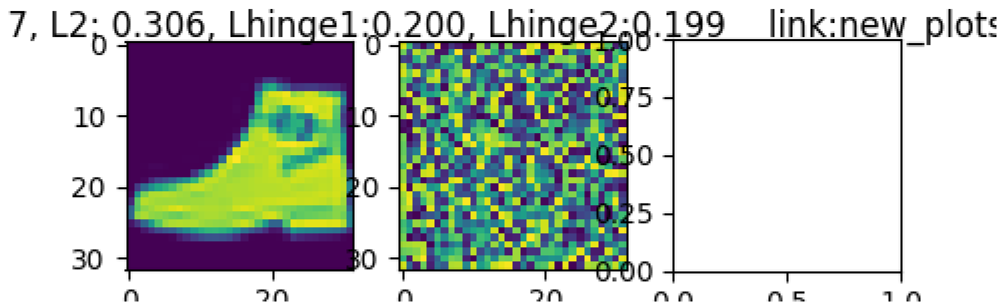
(*I have also added another loss term $L_2 = \mathbb{E}_{x_i}||x_i - \hat{x}_i||_1$ which is the L1 norm, to the loss stated above, although this was not mentioned in the paper. This addition seemed to stabilize the artifacts generated in the autoencoder reconstruction and made the training faster.)*

3. Settings

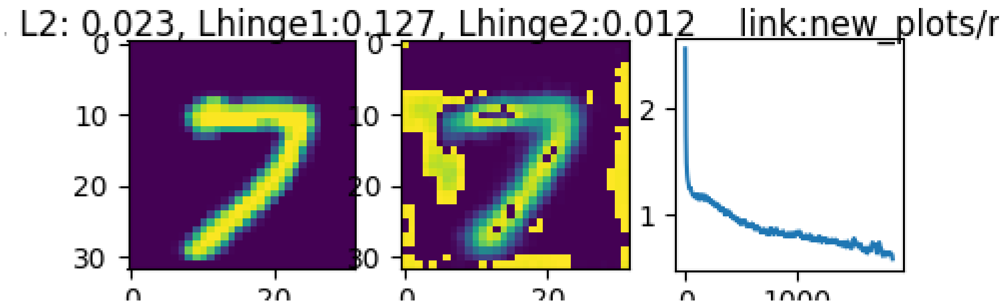
All the different settings were taken as-is from the paper itself.

4. Training

Before training-



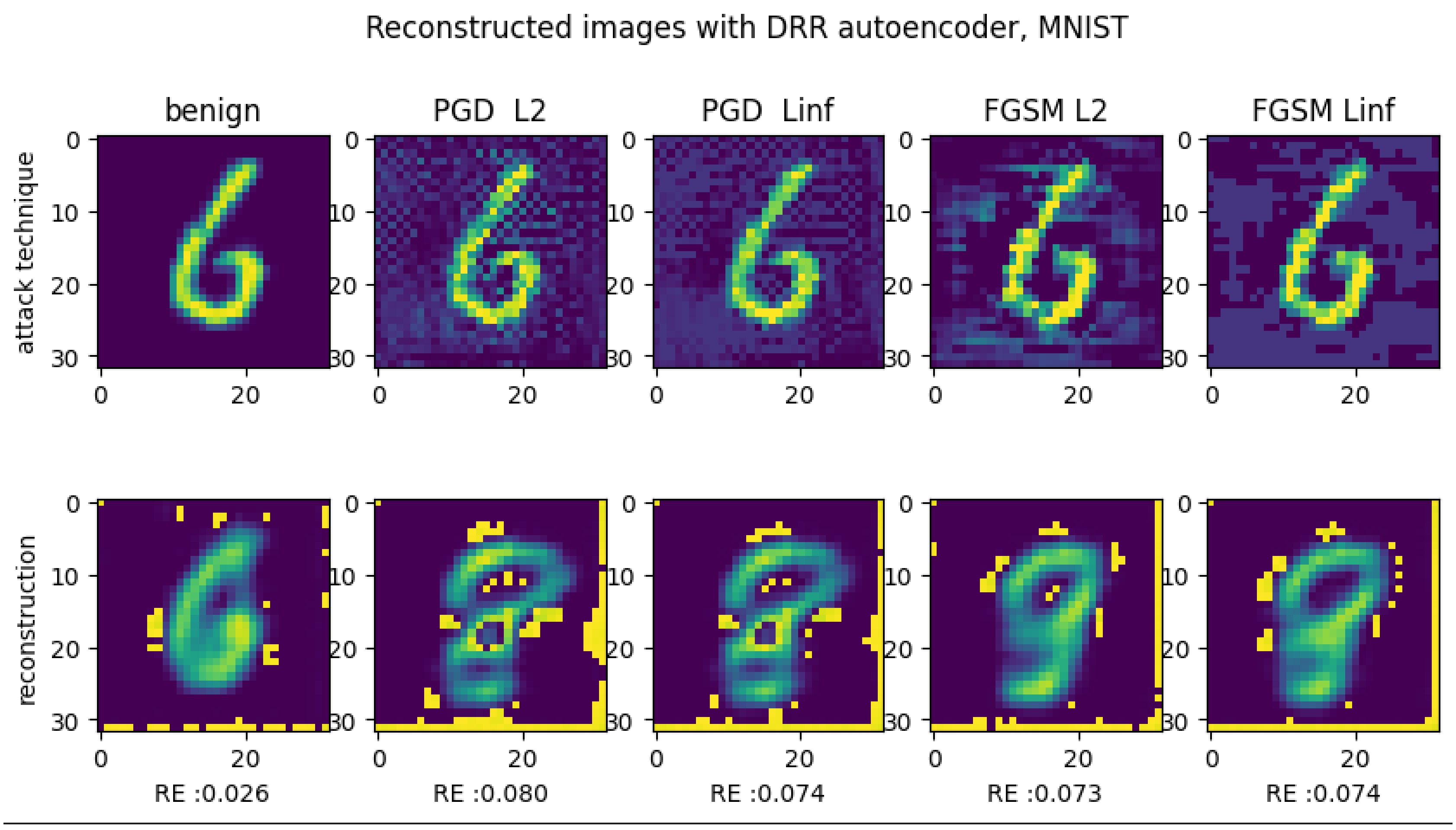
After training-



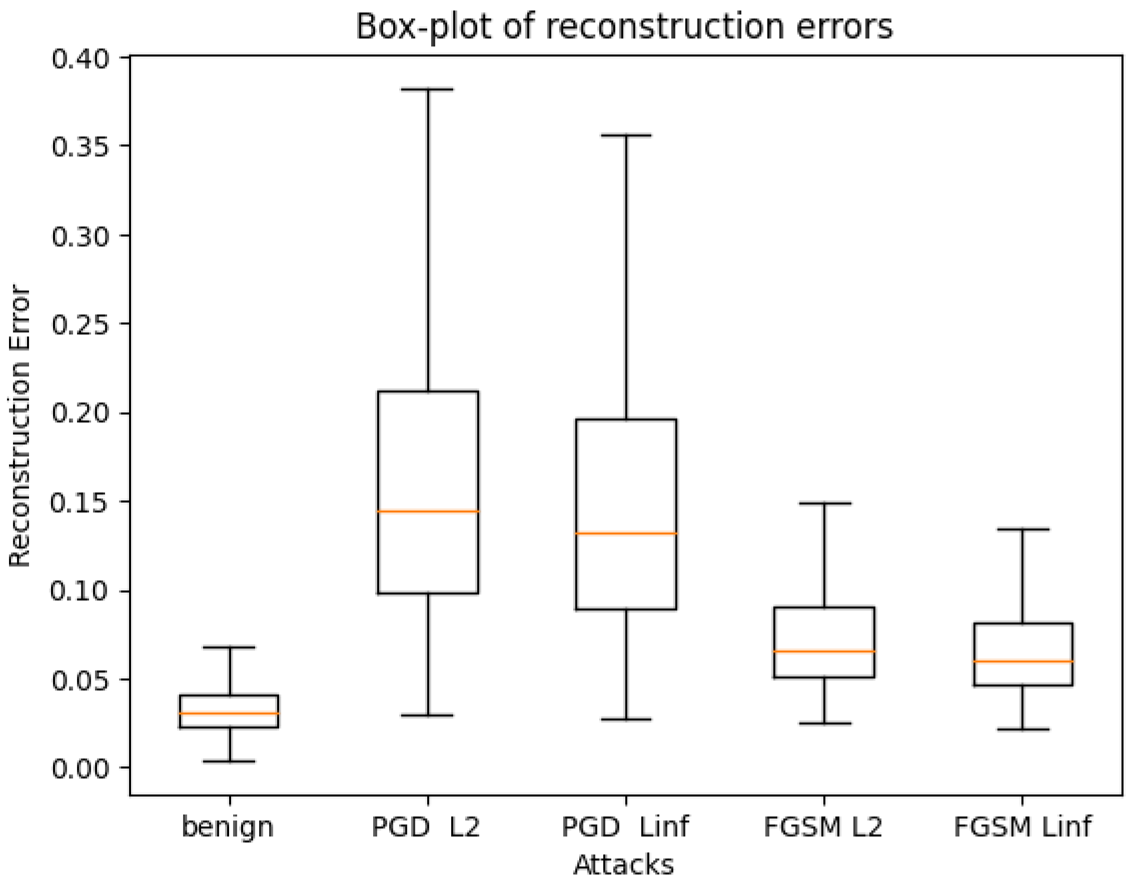
5. Results (on MNIST)

MNIST is one of the datasets used to show results in the paper. I have computed the results shown in the paper and plotted them below.

Some samples plotted below show how the reconstructions for the adversarially perturbed examples have significant reconstruction differences as opposed to the benign samples, which are reconstructed somewhat faithfully.



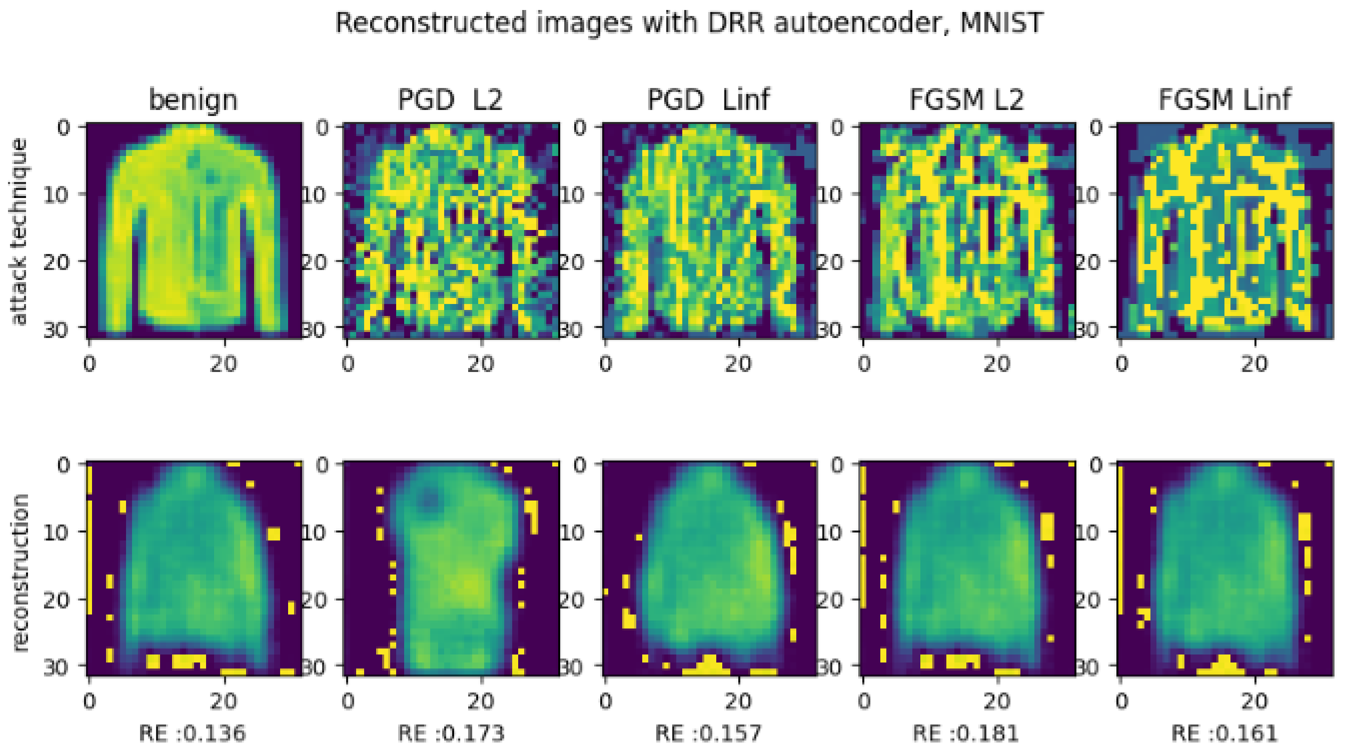
The box plot clearly shows how the reconstruction error for the adversarially perturbed examples is higher than that of the benign examples.

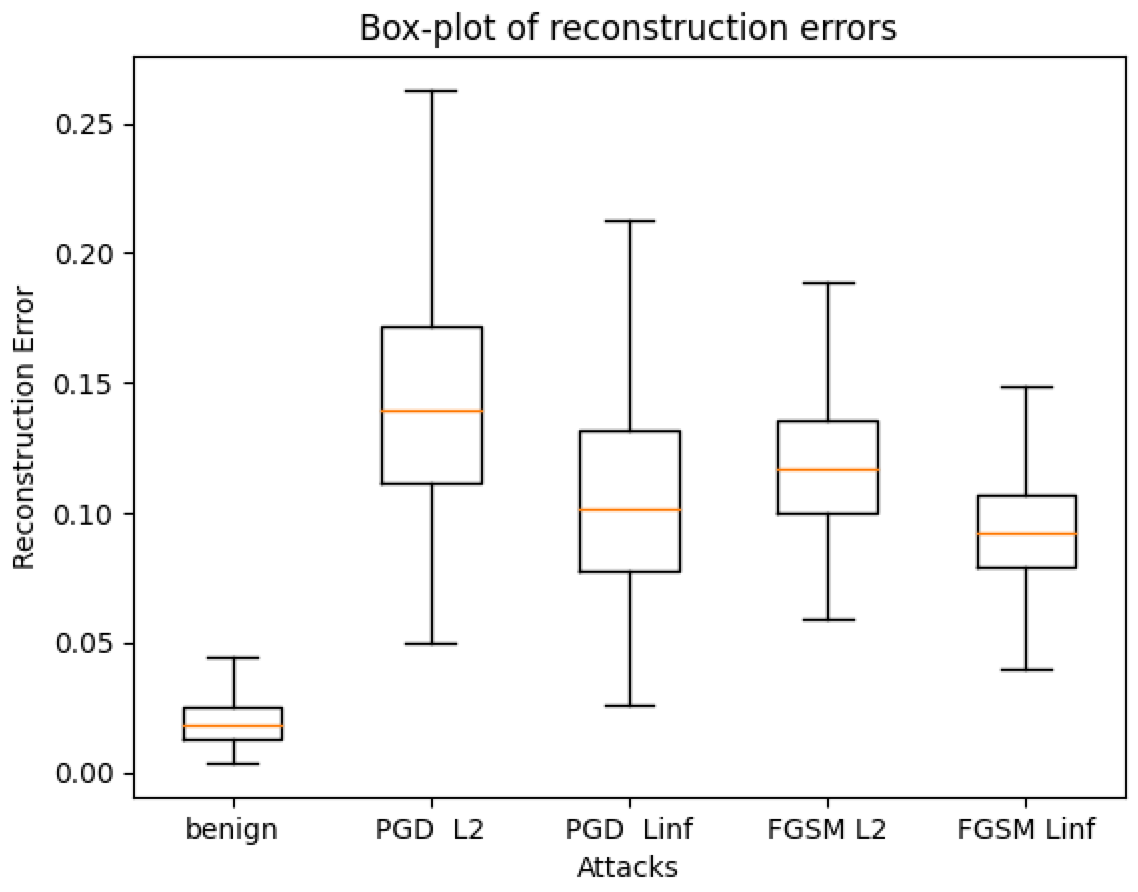


6. Results (on FashionMNIST)

FashionMNIST is not used in the paper results, hence I have computed results on it and shown them here.

The results on this dataset were that the reconstructions were of visually much better quality (less artifacts), however the L2 loss was not as good.





References

[0] - Self-Supervised Adversarial Example Detection by Disentangled Representation - the paper used in this.

[1] - https://gist.github.com/anderzzz/1adfa12a409e6367f41fa60c8c2d5bb7 - used to generate symmetric encoder - decoder architectures.

[2] - foolbox