

## Versions used

Python3 version - 3.9.10  
tensorflow version - 2.8.0

## Github links

Used below link as reference for using LeNet5 for CIFAR10

[3] [https://github.com/j-holub/Cifar-10-Image-Classifcation-using-LeNet5/blob/master/cifar10\\_classification.py](https://github.com/j-holub/Cifar-10-Image-Classifcation-using-LeNet5/blob/master/cifar10_classification.py)

Used below link as reference for using STRIP model on CIFAR10

[2] [https://github.com/garrisongys/STRIP/blob/master/STRIP\\_CIFAR10DeepArchit\\_Tb.ipynb](https://github.com/garrisongys/STRIP/blob/master/STRIP_CIFAR10DeepArchit_Tb.ipynb)

My forked DLFuzz repo:

[1] <https://github.com/rajguru7/DLFuzz/tree/master/CIFAR10>

## Work Done:

05/04/2022

I have gone through the DLFuzz research paper and the code on their Github repo to get a basic understanding of how DLFuzz is working.

Updated the DLFuzz code to work with python3 and tensorflow 2.8

To get a basic working model, I used the LeNet5 model in DLFuzz for the CIFAR10 classification.

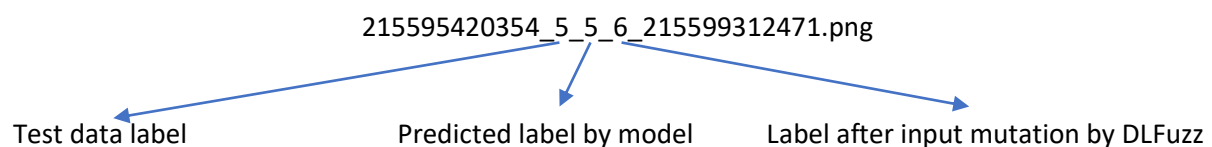
Training set accuracy: 71.76%

Overall Test accuracy: 63.82%

I fed the weights of the LeNet5 model to the DLFuzz python script for creating the adversarial examples.

For the seeds, I used 50 random images from the CIFAR10 test data.

Filename format for sample adversarial image created:



A total of 238 adversarial images were generated.

## Next Steps?

1. Using a 'better' model for the CIFAR10 classification and integrating it with DLFuzz
  - a. Will need better understanding of how DLFuzz is working and of CNNs.
2. Creating a backdoor in the better model or in the current LeNet5 model?
3. Is using poisoned data a good approach for creating the backdoor?

16/04/2022

## Work Done:

I went through the STRIP model for creating a backdoor and also an improved accuracy on CIFAR10 test data.

Used [2] as reference to build the model that will work with DLFUZZ

Created scripts to get weights for below models:

Model3 - LeNet5

Model4 - STRIP model

Model3x - LeNet5 with poisoned data

Model4x - STRIP model with poisoned data

Fed the weights to DLFUZZ to get the adversarial inputs for LeNet5 and STRIP

Used the below folders as seeds

seeds\_50 - random 50 images from test data

seeds\_70 - random 70 images from test data

### STRIP:

The generated adversarial examples were stored in the below folders:

1. STRIP folder - seeds\_50 STRIP model
2. STRIPX folder - seeds\_50 Poisoned model
3. STRIP1 folder - seeds\_70 STRIP model
4. STRIPX1 folder - seeds\_70 Poisoned model

Observation: Data was poisoned with target label 7 but DLFUZZ is not labelling any adversarial input as 7.

### LENET-5:

0. LeNet folder - seeds\_50 LeNet model
1. LeNetX folder - seeds\_50 Poisoned model
2. LeNet1 folder - seeds\_70 STRIP model
3. LeNetX1 folder - seeds\_70 Poisoned model

Observation: No. of label 7 predictions by DLFUZZ are decreasing after poisoning

## Next Steps?

1. Tweaking the parameters of DLFUZZ to check if it can find the adversarial inputs for the target label 7

Table: Total No. of Labels by DLFUZZ for respective label and model

Label	LeNet	LeNet1	LeNetX	LeNetX1	STRIP	STRIP1	STRIPX	STRIPX1
0	12	16	13	13	3	5	8	6
1	28	32	15	16	0	0	0	1
2	25	31	53	58	80	58	45	29
3	10	7	24	8	7	10	1	2
4	8	8	8	26	22	14	23	24
5	16	3	31	34	1	0	1	0
6	25	31	44	40	93	134	124	166
7	27	59	8	16	0	0	0	0
8	28	32	29	5	11	5	12	4
9	56	21	13	31	12	7	15	6

Table 1

08/05/2022

## Work Done:

As per previous discussion,  
I have analyzed the STRIP model with DLFuzz for different poison labels, different thresholds,  
different number of iterations, different loss functions

PFB the analysis below:

Input seeds

seeds\_50\_poison - first 50 training set images

First 600 training set images are poisoned with poison label while training the models

## Changing Loss function and Poison labels

Folder descriptions of generated inputs by DLFuzz:

STRIPX2 - ran gen\_diff.py with 5 iterations per seed and poison label 7

STRIPX3 - ran gen\_diff.py with 20 iterations per seed and poison label 7 - Model4x

STRIPX4 - ran gen\_diff.py with 20 iterations per seed and poison label 3 - Model4x1

STRIPX5 - ran gen\_diff.py with 20 iterations per seed and poison label 9 - Model4x2

STRIPX6 - ran gen\_diff.py with 5 iterations per seed and poison label9 - Model4x2 (Also changed the optimization function to target label 9) layer\_output = loss\_poison - predict\_weight \* (loss\_2 + loss\_3 + loss\_4 + loss\_5) - loss\_1

STRIPX7 - ran gen\_diff.py with 5 iterations per seed and poison label9 - Model4x2 (Also changed the optimization function to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX8 - ran gen\_diff.py with 5 iterations per seed and poison label9 - Model4x2 (Also changed the optimization function to target label 2) layer\_output = loss\_poison - loss\_1

Label	STRIPX2	STRIPX3	STRIPX4	STRIPX5	STRIPX6	STRIPX7	STRIPX8
0	6	29	48	31	0	1	1
1	0	0	0	0	2	0	1
2	31	191	189	376	1	0	146
3	3	10	7	92	0	0	0
4	32	98	138	92	38	4	31
5	0	0	0	0	0	0	0
6	143	611	545	329	11	18	52
7	1	1	2	0	0	0	0
8	4	9	28	56	0	7	1
9	4	11	10	9	62	201	7

Table 2

Observation:

After changing the loss function to target a particular label, DLFuzz finds more adversarial examples of the target label irrespective of the poison label.

The average perturbation from the original image from the above experiments was around 7%

#### *Changing thresholds and iterations:*

The average difference between the poisoned images and the input seed is around 31%

To achieve this perturbation the iterations and thresholds were changed

STRIPX9 - perturb\_adversial increased to 20% from 2%; poison label 9; 50 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX10 - perturb\_adversial increased to 10% from 2%; poison label 9; 50 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX11 - perturb\_adversial increased to 6% from 2%; poison label 9; 50 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX12 - perturb\_adversial increased to 6% from 2%; poison label 9; 30 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX13 - perturb\_adversial increased to 30% from 2%; poison label 9; 5 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

STRIPX14 - perturb\_adversial increased to 0.5% from 2%; poison label 9; 50 iterations per seed -

Model4x2(optimization fn to target label 9) layer\_output = loss\_poison - loss\_1

Average perturbation for above models

STRIPX9 – 51%

STRIPX10 – 51%

STRIPX11 – 51%

STRIPX12 – 30%  
STRIPX13 – 8%  
STRIPX14 – 50%

From above results, it can be seen that the number of iterations has the most effect on the average perturbation.

The threshold is set only to accept more seeds from the original seed in DLFuzz. There is no threshold set on the perturbation achieved by the iterations.

As STRIPX12 has the desired amount of perturbation required to reach the poison target, further analysis was done on the same

To see whether DLFuzz is making perturbations in the poisoned image direction:

A graph was plotted for every seed for STRIPX12 experiment

Y axis – The L2 diff between generated input and the corresponding poisoned seed

X axis – Iteration Number

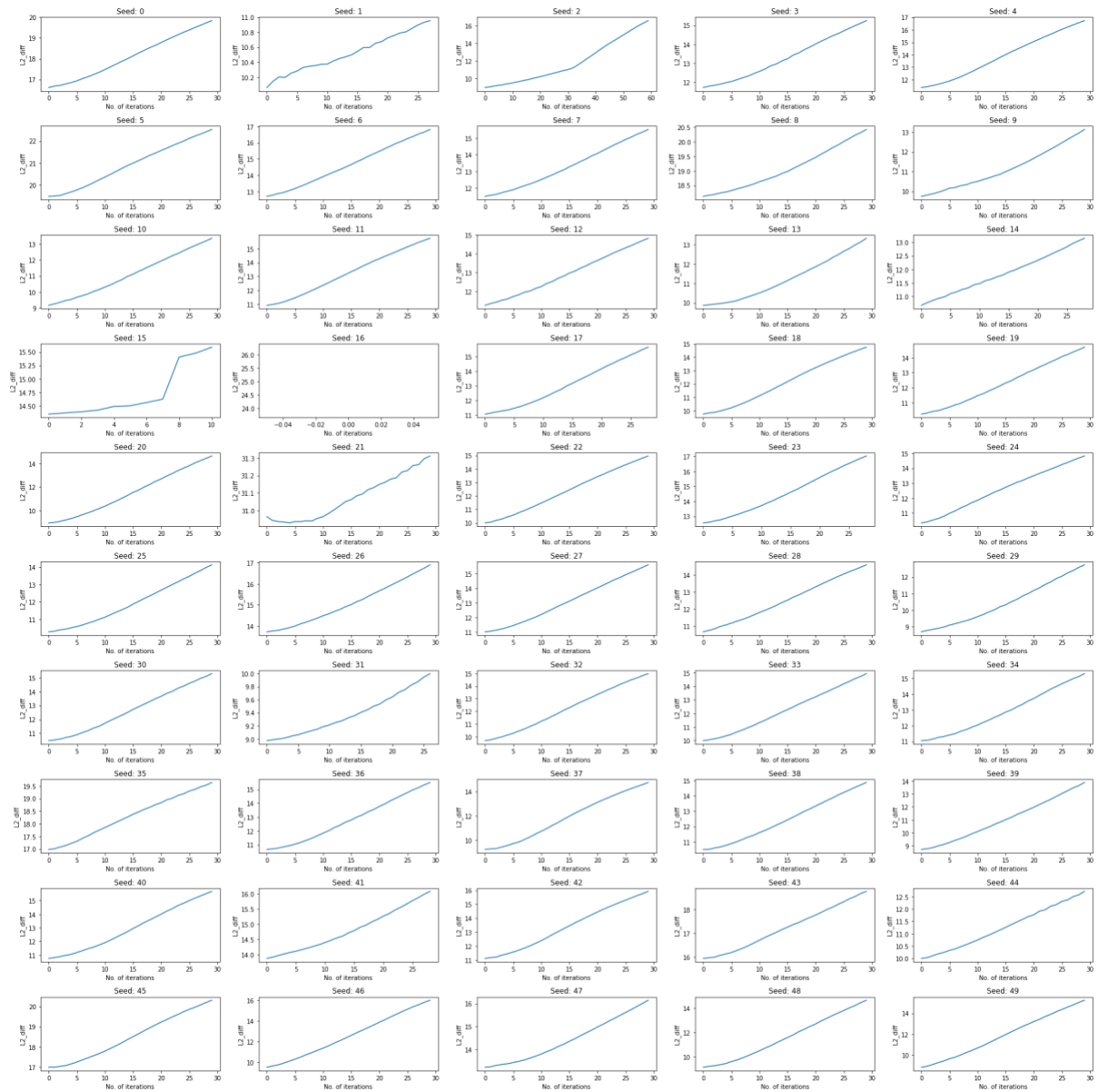


Figure 1

As can be seen from graph the L2\_diff from the poison target keeps on increasing with every iteration

Another graph was plotted w.r.t the trigger image

Y axis – The L2 Diff between generated input and the trigger

X axis – Iteration Number

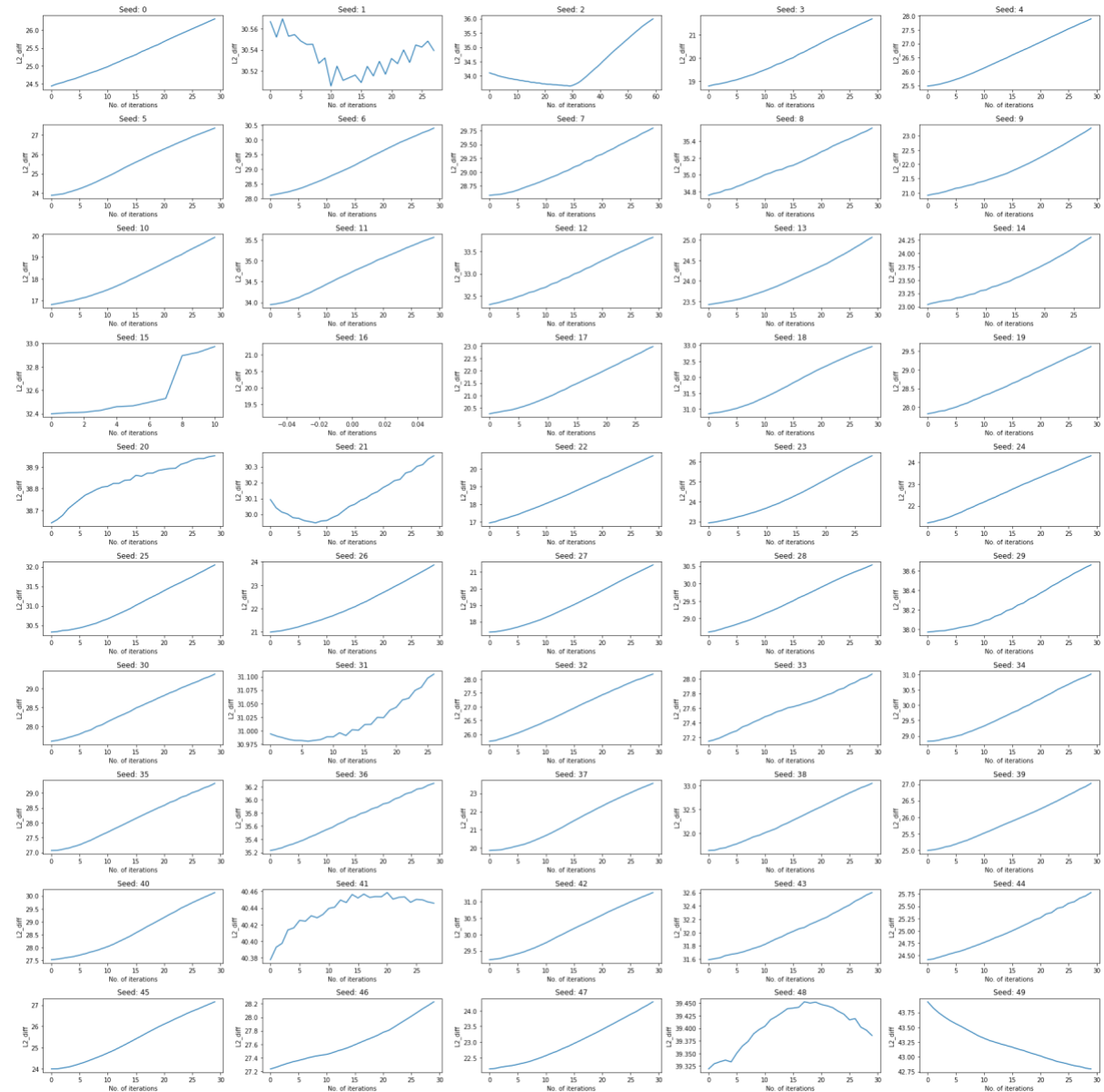


Figure 2

As can be seen above the L2\_diff decreased for a few seeds.

Conclusion:

From the above experiments, I believe that DLFuzz by itself cannot find the poison trigger and changes to its algorithm will have to be done to detect or find the poison trigger

## Next Steps?

To check whether fuzzing can be used to detect poisoned models

Going through the paper [1]

[1] <https://arxiv.org/pdf/2112.13064.pdf>

09/05/2022

## Work Done:

Analysis and experimentation of results from DLFuzz

Question by Stefanos:

- 1) In the second group of figures, we see very small L2 distance of the poisoned image and the generated by the fuzzer with seeds 1, 21, and 31 around the 10th iteration. Why? Additionally, the distance decreases for seeds 48 and 49 in the last iterations. Why? Also, do the images look similar to the poisoned ones?

PFB the analysis:

2, 21 and 49 seed numbers had downward slopes for the L2 diff with perturbation (Figure 2)

The poisoned images for seed 2, 21 and 49 along with the input generated by DLFuzz which had lowest L2\_diff from trigger


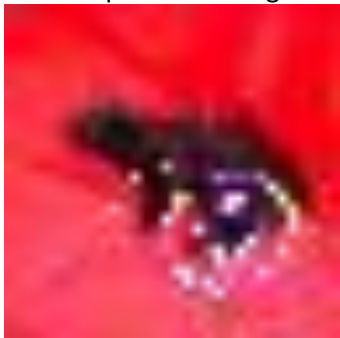




Seed 2 poisoned image	Seed 21 poisoned image	Seed 49 poisoned image
		
Seed 2 (Iteration 30)	Seed 21 (iteration 8)	Seed 49 (iteration 30)
		

Table 3

Only for seed 21 there is some similarity

Answer to Question from Stefanos:

For each generated image by DLFuzz, even though the L2 diff from the trigger decreased. The L2 diff from the poisoned image still increased (see Figure 1)

The L2\_diff from the trigger is not useful since the graph could be increasing decreasing or curved. This is demonstrated below

To demonstrate the desired behavior of different parameters, I manually generated the desired sequence of images. (Adding desired perturbation to the images)  
At every iteration, the trigger divided by the number of iterations was added to the previous image.

$\text{gen\_input} = \text{gen\_input} + \text{trigger}/(\text{total iterations})$

The total number of iterations was taken as 30.

At 30<sup>th</sup> iteration the generated image will be the poisoned image itself.

**The above will be called ‘desired model’ for naming convenience**

The below graphs were obtained for ‘desired model’:

### 1. L2 diff from poisoned image

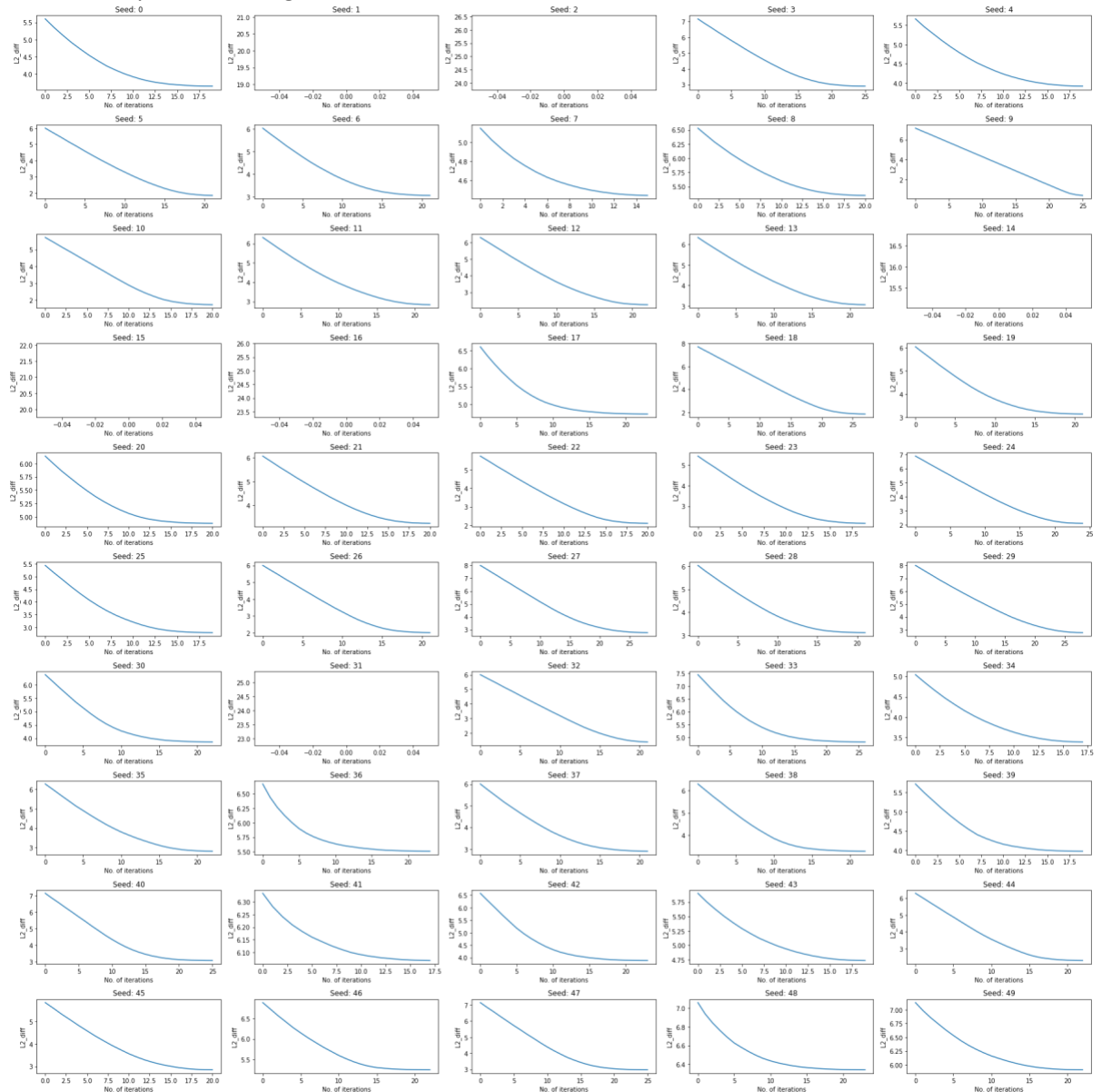


Figure 3

As expected, the L2 diff is decreasing for the manually inserted images with the desired perturbation.

### 2. L2 diff from trigger



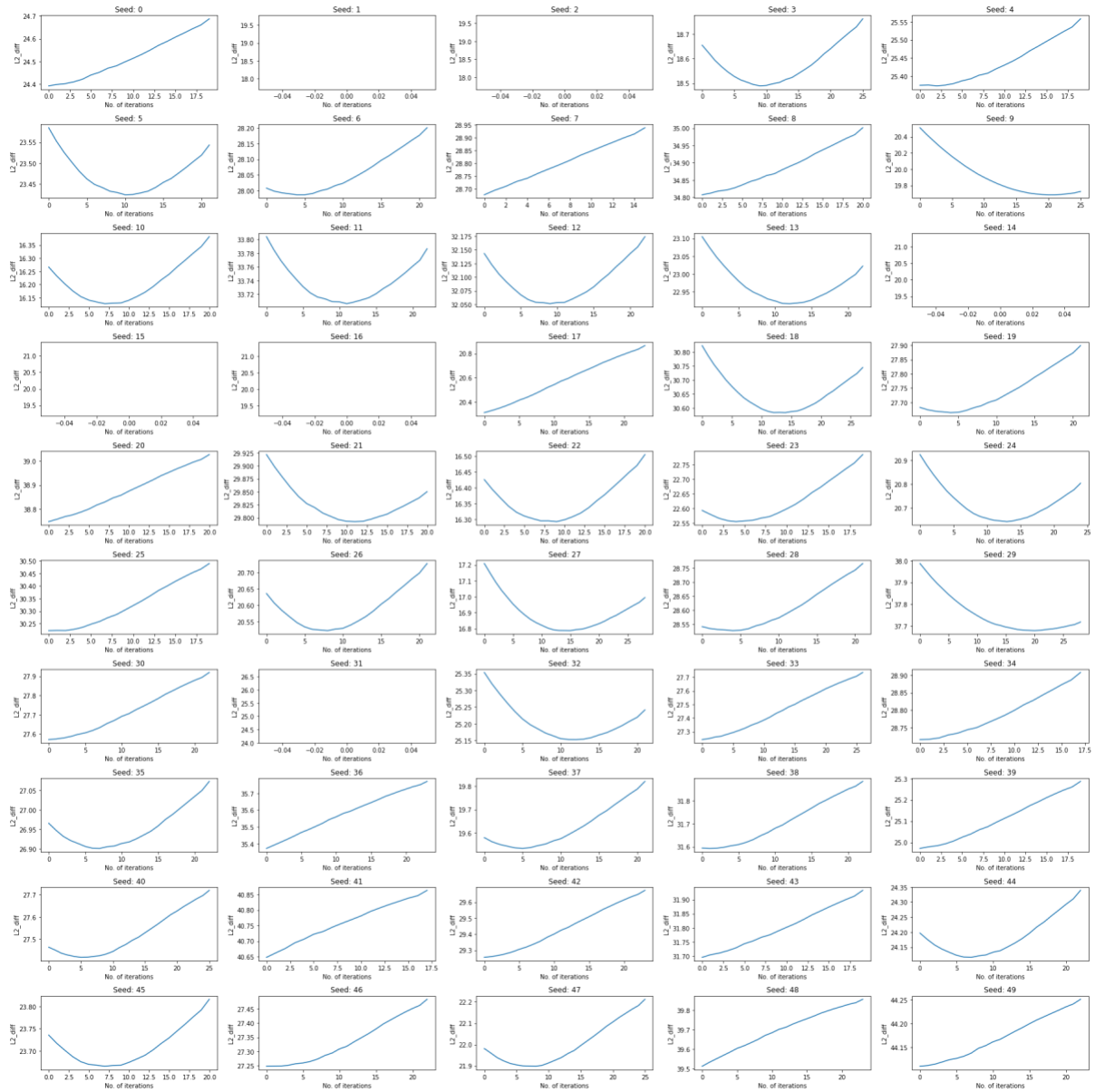


Figure 4

As can be seen from above the L2 diff from trigger has no exact pattern (It can be increasing, decreasing or curved)

Therefore, Figure 2 does not have much significance.

For both the figures (3 and 4) seeds 1,2,14,15,16,31 have missing graphs.

This is because the generated input by DLFuzz is only written to disk if the label of the generated input is different from the original image. The original label of seeds 1,2,14,15,16,31 is already label '9'. Since with every iteration in the model, the generated input had label 9, the generated image was not saved to disk. In the above calculation of L2 diff from trigger and the poison image, the L2 diff is calculated from the images that were saved to disk (i.e images which had generated input label different from original image).

Therefore, no values of L2 diff were obtained for the above seeds.

**Changing the iteration to 18 from 30 in STRIPX12 since as observed from the analysis, 30 iterations were resulting in perturbations reaching as high as 60% and average perturbation of 30% which isn't required. The average perturbation of generated inputs in 'desired model' was 20%**

The new results obtained with 18 iterations are labelled as STRIPX15, other parameters are kept same as STRIPX12. STRIPX15 had an average perturbation of 20%.

For reference

STRIPX – ran `gen_diff.py` with 5 iterations per seed; Model4x(poison label 7)

STRIPX12 - `perturb_adversial` increased to 6% from 2% ; Model4x2(poison label 9); 30 iterations per seed, optimisation fn to target label 9 (`layer_output = loss_poison - loss_1`)

STRIPX15 - `perturb_adversial` increased to 6% from 2% ; Model4x2(poison label 9); 18 iterations per seed, optimisation fn to target label 9 (`layer_output = loss_poison - loss_1`)

Comparing parameters during the execution of DLFuzz(STRIPX15) and during execution with desired perturbation(Desired Model)

In the below experiments, I will be comparing how the different values like `loss_poison`, `loss_orig`, `loss_fn(loss_poison – loss_orig)` and neuron coverage vary at every iteration for 2 different experiments. For Figure 5,6 , at every iteration I manually entered the next image into the STRIP model and saw how the loss values and neuron coverage are varying. For Figure 7,8 I let DLFuzz use the gradient on the Loss function to create the image that has to be entered into STRIP model for the next iteration.

#### 1. Execution of Model with Desired perturbation (Desired Model)

Loss Function, Poison label loss, Original label loss

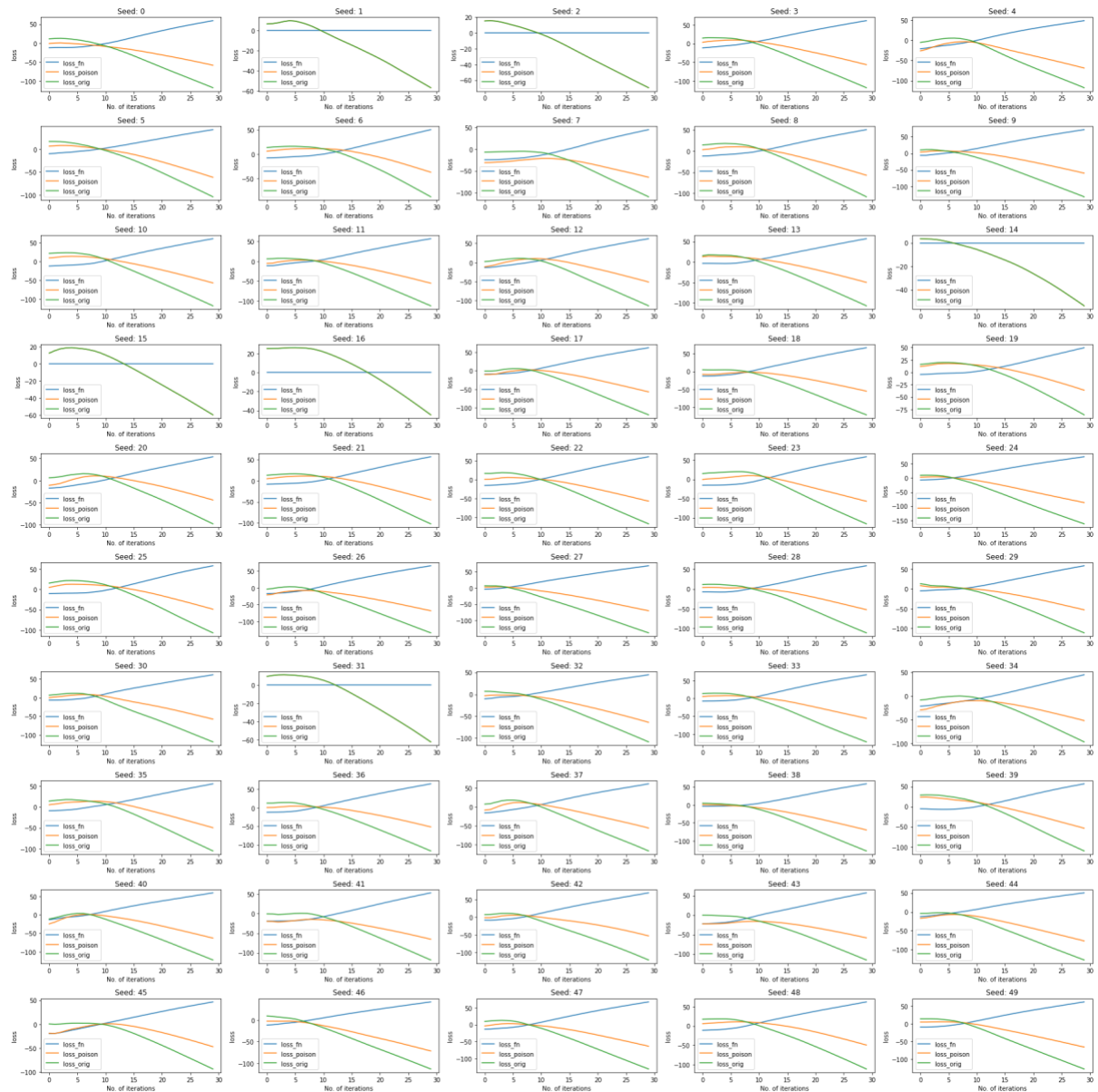


Figure 5

## Neuron coverage

The Loss function in DLFuzz is as below:

$$\text{Final\_loss} = \text{loss\_poison} - \text{loss\_orig} + \text{Neuroncoverage}$$

The Final\_loss is being tried to maximise by DLFuzz

This can happen in 2 ways

1. Increase difference between loss\_poison and loss\_orig
2. Increase the neuron coverage

Neuron coverage – It is a measure of the number of neurons in the model, that are activated by the particular input. By activated, it means that the neuron's output was larger than the set threshold. It is taken into the loss function based on the demonstration that covering more neurons could potentially trigger more logic and more erroneous behaviors (Deepxplore: Automated whitebox testing of deep learning systems). Neuron Coverage per iteration has been plotted below:-

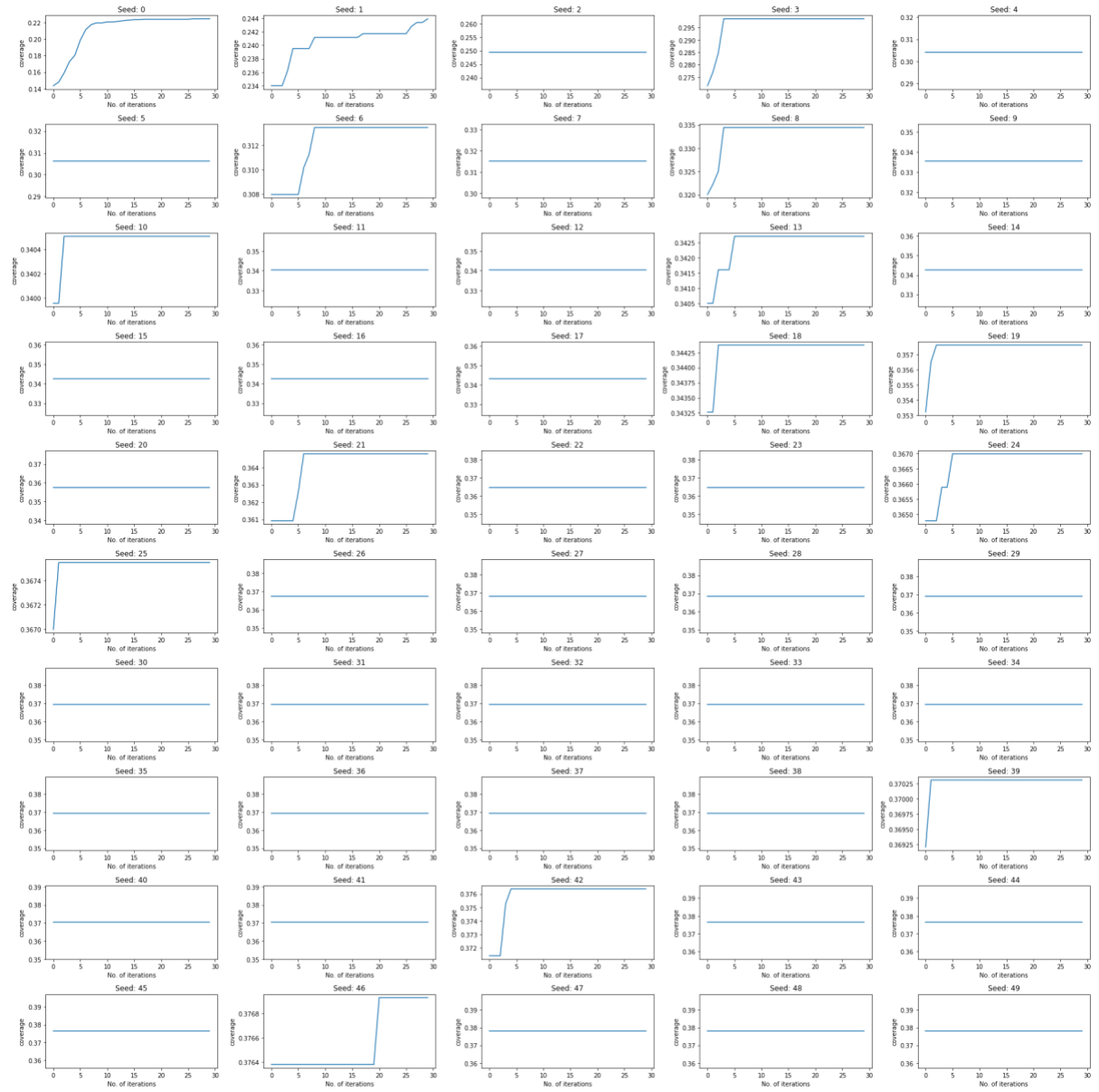


Figure 6

In figures (5 and 6) the graphs for seeds 1,2,14,15,16,31 are not missing because the loss function values and the neuron coverage values were recorded during the execution of DLFuzz. Also, the values were recorded irrespective of whether the generated input label was different from original label or not.

## 2. Execution of DLFuzz model (STRIPX15)

Images having original label same as poison label(label: 9) are skipped.

Loss Function, Poison label loss, Original label loss



Figure 7

Neuron Coverage



Figure 8

Observations from above graphs:

1. The loss value of poison label and original label both decrease for desired model(Figure 5) and STRIPX15(Figure 7)
2. The change in neuron coverage is very small for both the models
3. The loss values for desired model are reaching as low as -100 but for STRIPX15 the loss values are reaching below -600. This indicates that the loss function will have to be changed to get higher loss values(around -100) for STRIPX15 while keeping the perturbation the same(avg 20%)

Desired with 18 iterations  
Loss function

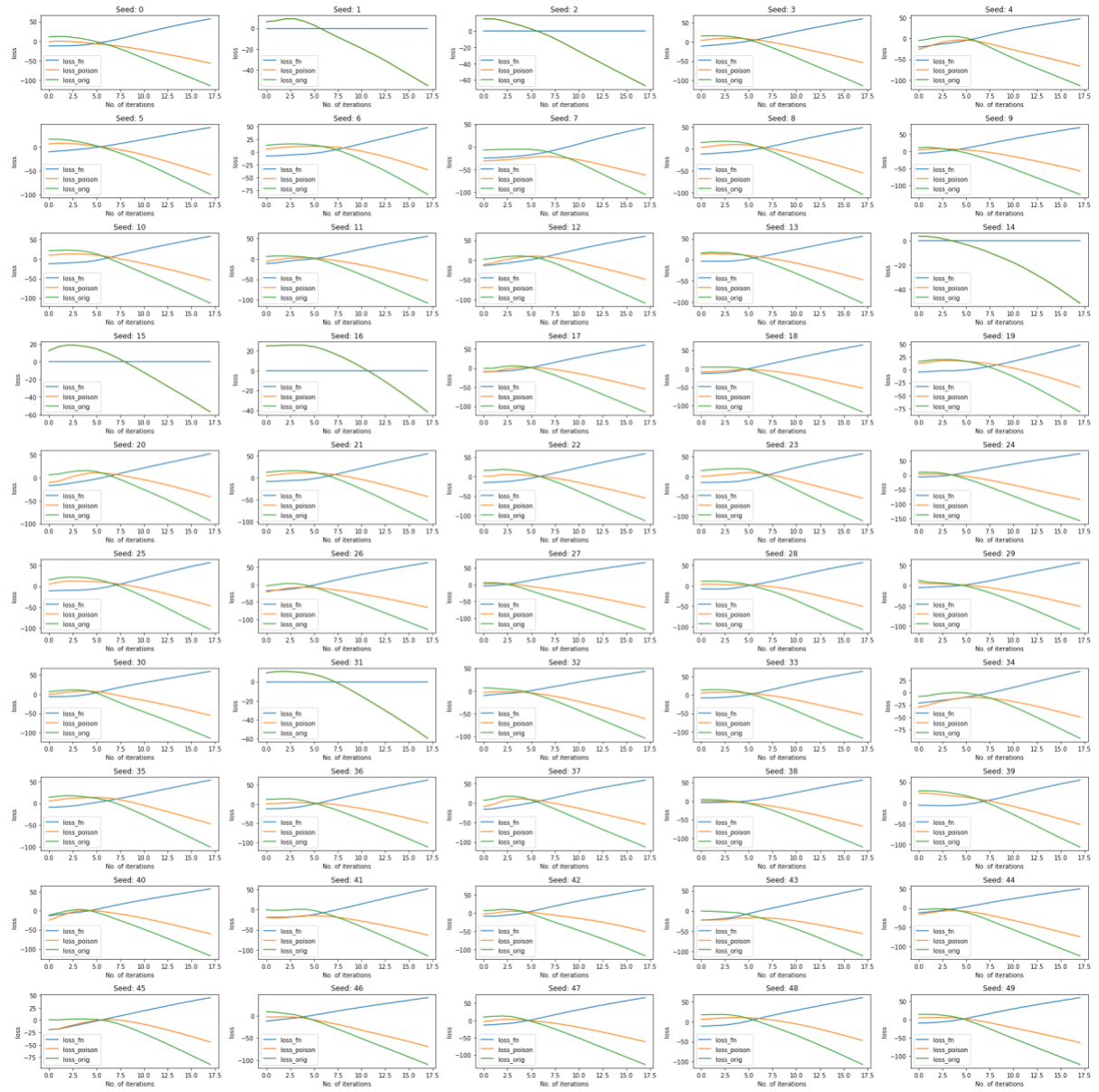


Figure 9

Neuron coverage

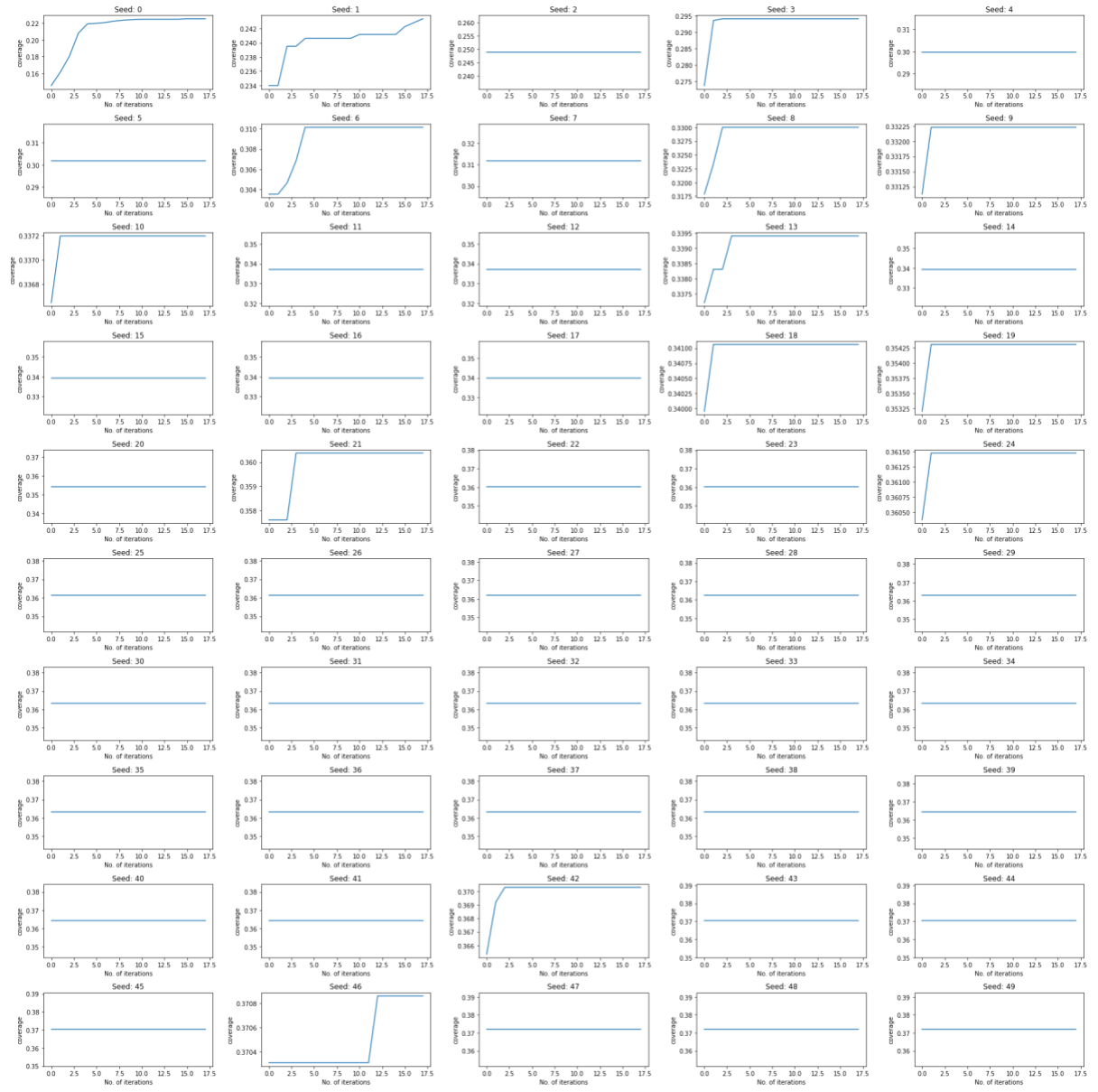


Figure 10