Comparison between MLPKAN1, MLPKAN2, MLPKAN3

Training strategy:

- Trained for 100 epochs. Interrupted training if no improvement in highest validation accuracy for 15 continuous epochs.
- Normal model trained on original MNIST samples, PGD adversarial model trained on PGD MNIST samples
- For adversarial training, PGD adversarial samples are used with parameters:
  alpha=8/255
  epsilon=0.2
  iter=20
- For KAN model, we have used Efficient KAN since the original KAN implementation was very slow for adversarial training.

Model specifications:

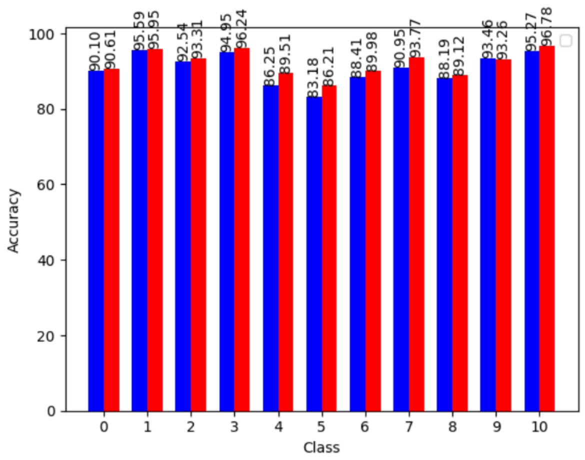
1) MLPKAN1
   a) Linear(28*28, 512) -> Linear(512, 256) -> KAN([256,10]) -> ReLU
   b) Num parameters: 558,848
2) MLPKAN2
   a) Linear(28*28, 512) -> KAN([512,,256]) -> KAN([256,10]) -> ReLU
   b) Num parameters: 1,738,240
3) MLPKAN3
   a) KAN([28*28, 512]) -> KAN([512,256]) -> KAN([256,10])
   b) Num parameters: 5,350,400

Results:

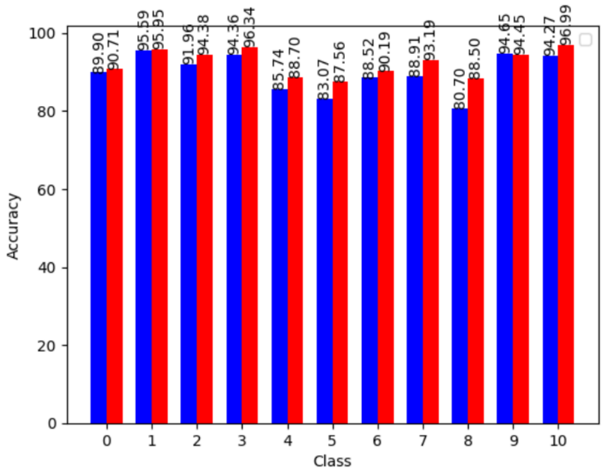
1) **Accuracy on PGD samples for Normal Model v/s PGD Adversarial Model**
Testing PGD parameters: alpha=8/255; epsilon=0.2 ; iter=20
0-9 show the MNIST digit classes; 10 shows the test on all classes with random sampling

Observation: Using more KAN layers instead of MLP layers does not necessarily mean increase in PGD robust accuracies. This probably happened due to saturation and oversimplified aspects of MNIST dataset. Perhaps testing on more challenging datasets could yield some conclusive results

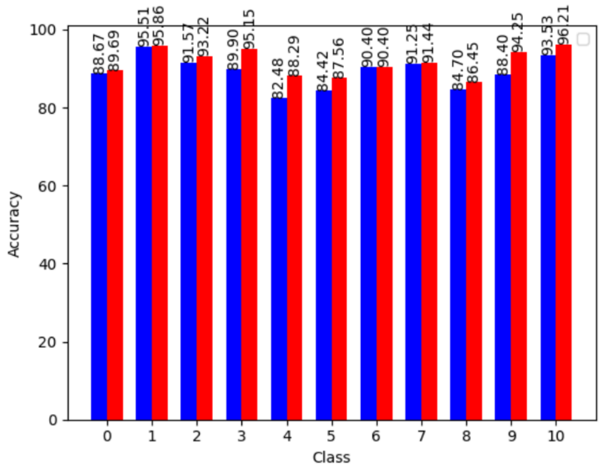
MLPKAN1



MLPKAN2

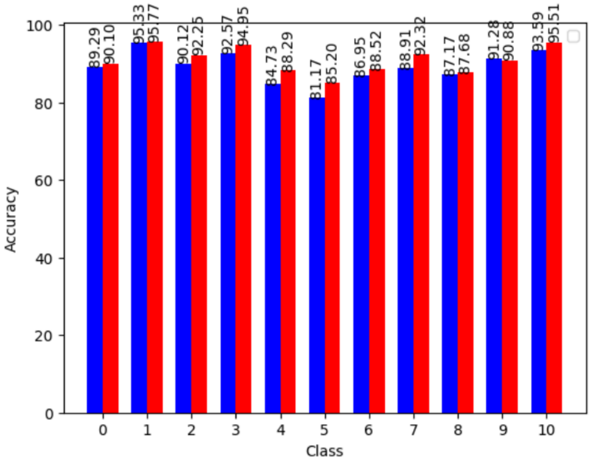


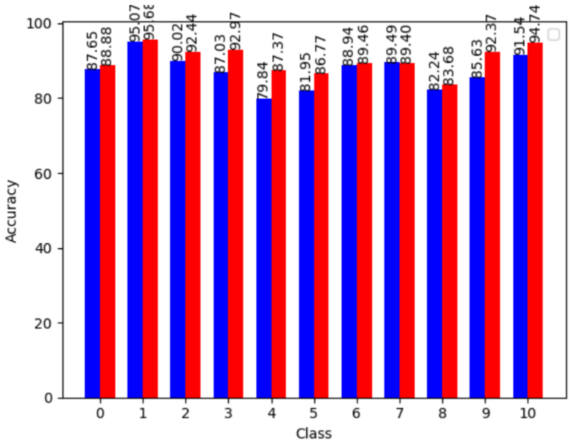
MLPKAN3

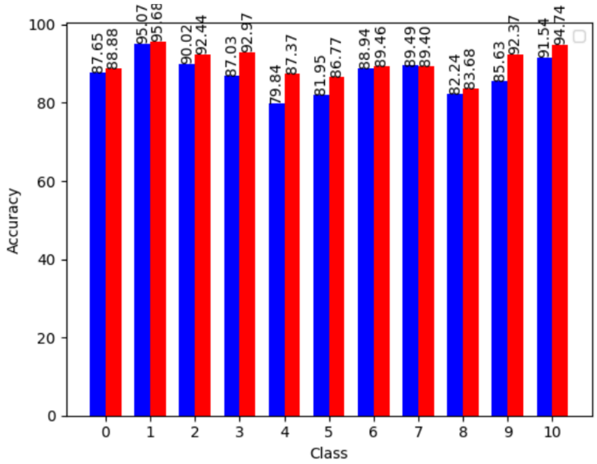


2) **Accuracy on PGD samples for Normal Model v/s PGD Adversarial Model**
Testing PGD parameters: alpha=8/255; epsilon=0.2 ; iter=40
0-9 show the MNIST digit classes; 10 shows the test on all classes with random sampling

Observation: No visible improvement with using more KAN layers



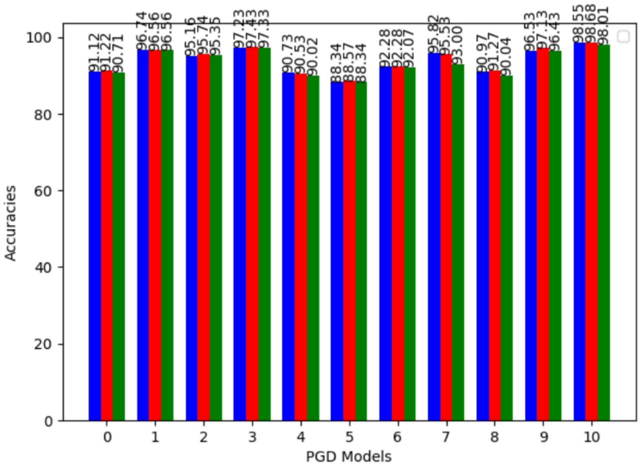




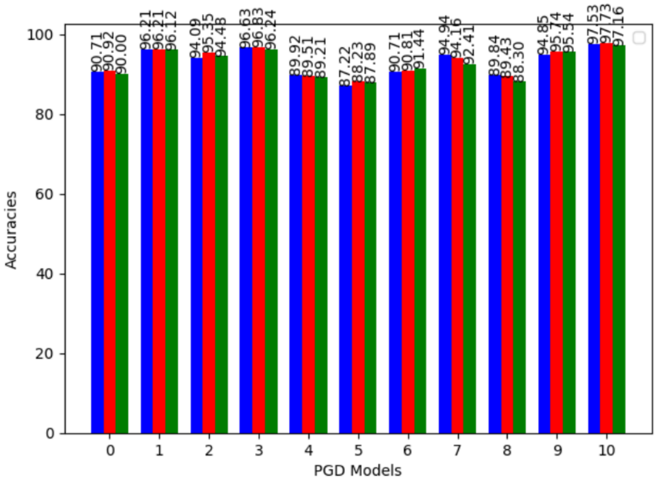
3) **Accuracy on PGD samples for Normal Model v/s PGD Adversarial Model**
Blue - MLPKAN1; Red - MLPKAN2; Green - MLPKAN3
Testing PGD parameters: alpha=8/255; epsilon=0.2 ;

Observation: For different PGD iterations used on testing samples, no specific trend in accuracies with changing number of KAN layers.
The accuracies seem to be slightly higher in MLPKAN2.

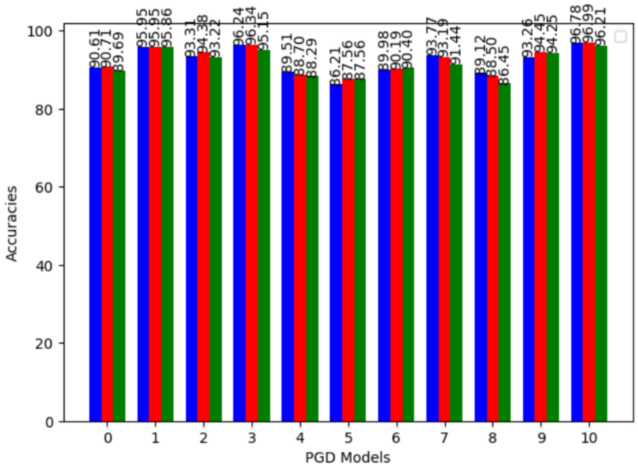
Iter = 0



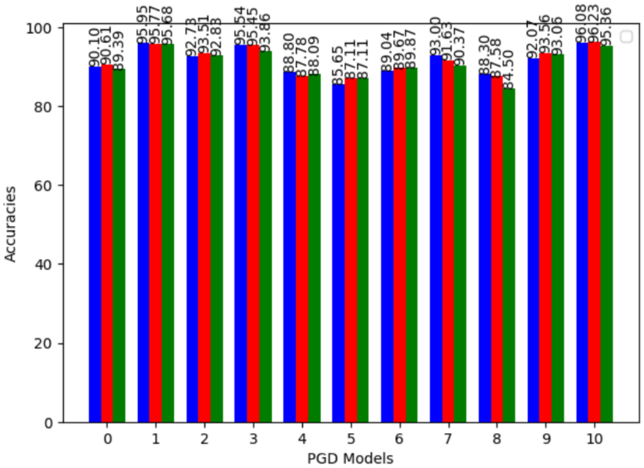
Iter = 10

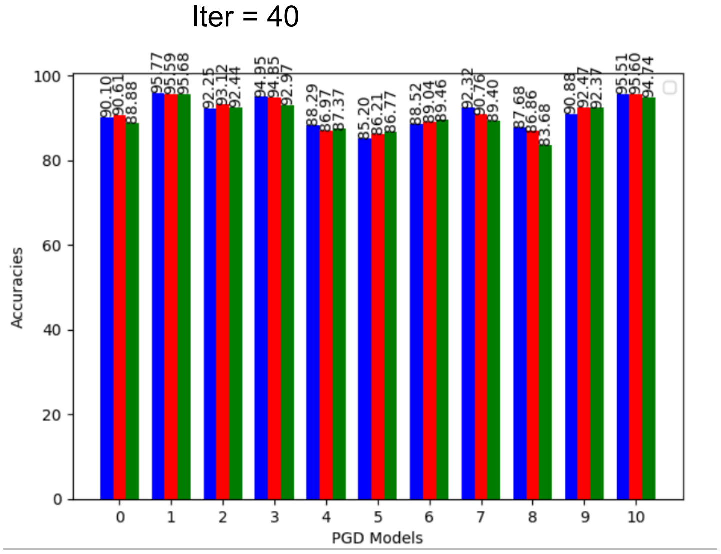


Iter = 20



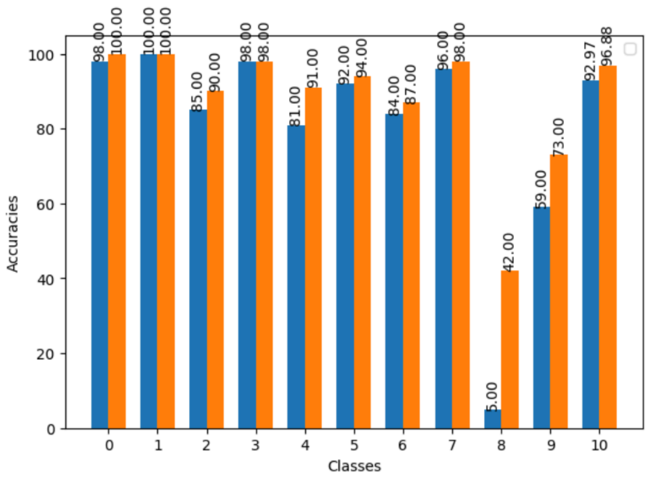
Iter = 30

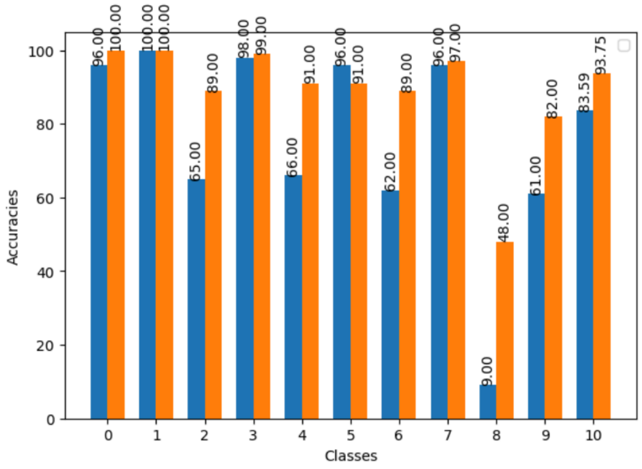


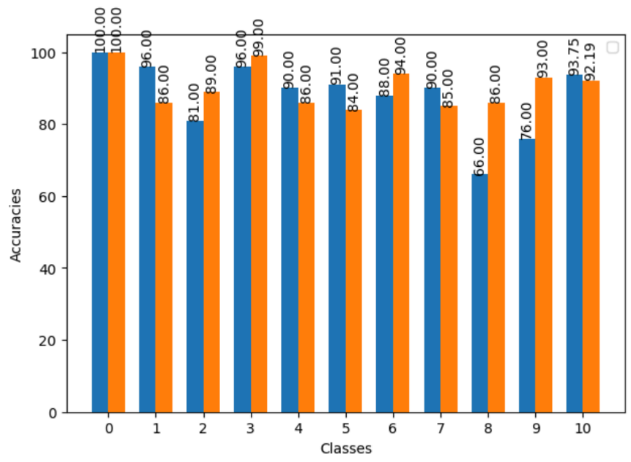

Iter = 40
Accuracies
100
80
60
40
20
0
90.10
90.61
88.88
95.77
95.59
95.68
92.25
93.12
92.44
94.95
94.85
92.97
88.29
86.97
87.37
85.20
86.21
86.77
88.52
89.04
89.46
92.32
90.76
89.40
87.68
86.86
83.68
90.88
92.47
92.37
95.51
95.60
94.74
0
1
2
3
4
5
6
7
8
9
10
PGD Models

4) Accuracy on APGD-CE samples for Normal Model v/s PGD Adversarial Model (Trained on PGD samples but tested on APGD-CE samples)
Blue - Normal; Orange - PGD
Testing APGD-CE parameters: norm='Linf'; eps=8/255; version='standard'

0-9 show the MNIST digit classes; 10 shows the test on all classes with random sampling
Observation: There is some variance in classwise performance and no model performs better on all classes. However, for class 10 (all class samples sampled randomly), the performance is greatest for MLPKAN1


MLPKAN1

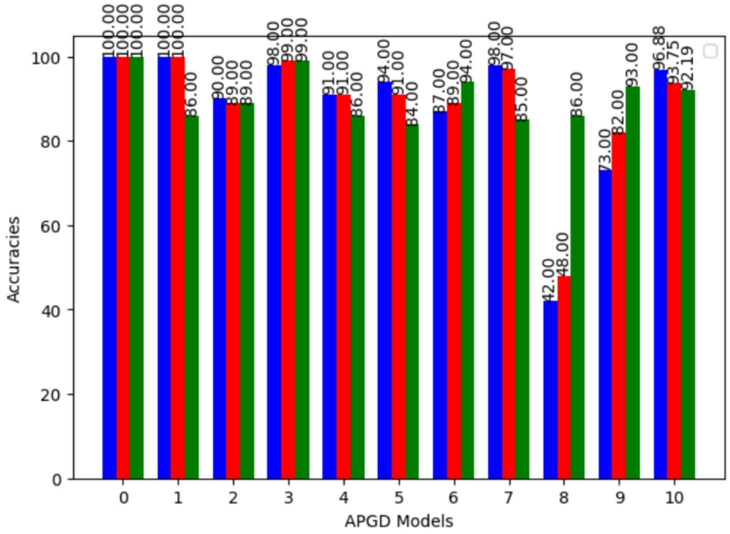

MLPKAN2


MLPKAN3

5) Accuracy on APGD-CE samples for Normal Model v/s PGD Adversarial Model
Comparision b/w MLPKAN1, MLPKAN2, MLPKAN3
(Trained on PGD samples but tested on APGD-CE samples)
Blue - MLPKAN1; Red - MLPKAN2; Green - MLPKAN3
Testing APGD-CE parameters: norm='Linf'; eps=8/255; version='standard'

0-9 show the MNIST digit classes; 10 shows the test on all classes with random sampling

Observation: There is some variance in classwise performance and no model performs better on all classes. However, for class 10 (all class samples sampled randomly), the performance is greatest for MLPKAN1


Accuracies
APGD Models