Adversarial Attacks Against 3D Point Clouds

Hemant Kumar Sharma
Carnegie Mellon University
hemantks@andrew.cmu.edu

Saurabh Patil
Carnegie Mellon University
ssp2@andrew.cmu.edu

Akshay Antony
Carnegie Mellon University
akshayan@andrew.cmu.edu

Abstract

Deep neural networks have become state-of-the-art algorithms for the application of object detection and classification. Despite the broad range of applications, these neural networks are prone to adversarial attacks. The problem of identifying 3D objects has many safety-critical applications, like autonomous driving, where adversarial robust 3D deep learning models are desired. In the past few years, a lot of research has been done in finding different ways to attack the 2D images, 3D objects have various differences with 2D images, and this specific domain has not been rigorously studied so far. While deep learning in 3D domain has achieved revolutionary performance in many tasks, the robustness of these models has not been sufficiently studied or explored. In this work, we show that existing state-of-the-art deep 3D models are extremely vulnerable to adversarial attacks. In this paper we explore how vulnerable PointNet and DGCNN are to the attacks. We conduct extensive experiments to evaluate the proposed algorithms on the ModelNet40 3D shape classification dataset. Code for experiments is released on https://github.com/akshay-antony/Adveserial-Point-Cloud

1 Introduction

Point clouds are an important 3D data format for computer-vision applications since they are the raw outputs of many 3D data collecting devices such as radars and sonars. High-level processing of 3D point clouds is typically required for real-world applications such as object categorization and segmentation. Recent study has proposed using Deep Neural Network (DNN) for high-accuracy and high-level processing of point clouds, with surprising results. PointNet[1], PointNet++ [2], and DGCNN [3] are examples of representative DNN models for point-cloud data classification that effectively managed the irregularity of point clouds while achieving high classification accuracy. In this research, we investigate the resilience of 3D models that deal directly with 3D objects. We specifically chose point clouds to represent 3D objects since they are the raw data from most 3D sensors such as depth cameras and Lidars. As a result, we attack 3D models by producing 3D adversarial point clouds. In terms of the attacking target, we concentrate on the widely used PointNet model [1] and DGCNN [3]. We chose PointNet because it and its versions have been widely and effectively used in a wide range of applications, including 3D object identification for autonomous driving, semantic segmentation for indoor scene interpretation, and AI-assisted form creation. Aside from that, one distinguishing feature of PointNet and its derivatives is their resistance to farthest/random point dropping. [1] owes its resilience to PointNet's max-pooling layer, which only focuses on a key portion of a point cloud. In other words, the recognition result is mostly controlled by those crucial spots, hence removing certain non-essential sites has little effect on the forecast. We concentrate on two adversarial attacks: the rapid gradient sign approach and the saliency map-based assault. Section 2 discusses related work, Section 3 discusses point-cloud data categorization models, Section 4 discusses attacking strategies, and Sections 5 and 6 describe the results and conclusion.

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

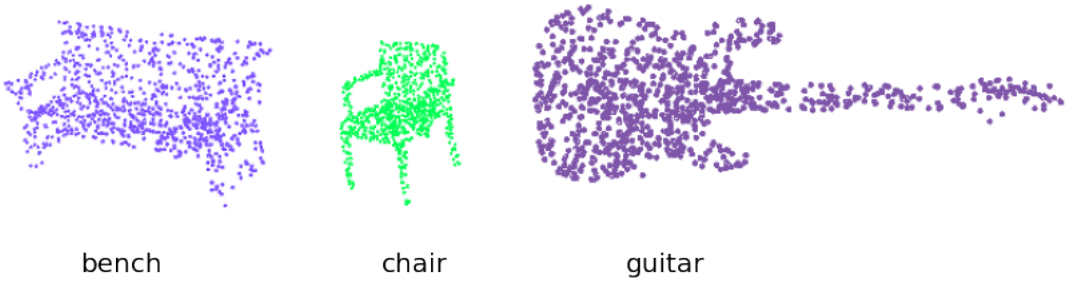


Figure 1: Point cloud

2 Related Work

2.1 PointNet

Point clouds are consisted of unordered points with varying cardinality, which makes it hard to be consumed by neural networks. Qi et al. [1] addressed this problem by proposing a new network called PointNet, which is widely used for deep point cloud processing. PointNet and its variants [2,3] exploit a single symmetric function, max pooling, to reduce the unordered and varying length input to a fixed-length global feature vector and thus enables end-to-end learning. PointNet implements a combination of transform networks and sequential convolutional layers for classification. The classification network uses a shared multi-layer perceptron to change the depth of the point clouds. After a series of transform and perceptron layers, max-pooling is applied to create a global feature vector. Finally, a fully-connected layer is used to map the global feature vector to k output classification scores. [1] also tried to demonstrate the robustness of the proposed PointNet and introduced the concept of critical points and upper bounds. They showed that points sets laying between critical points and upper bounds yield the same global features and thus PointNet is robust to missing points and random perturbation. However, they did not study the robustness of PointNet against adversarial manipulations.

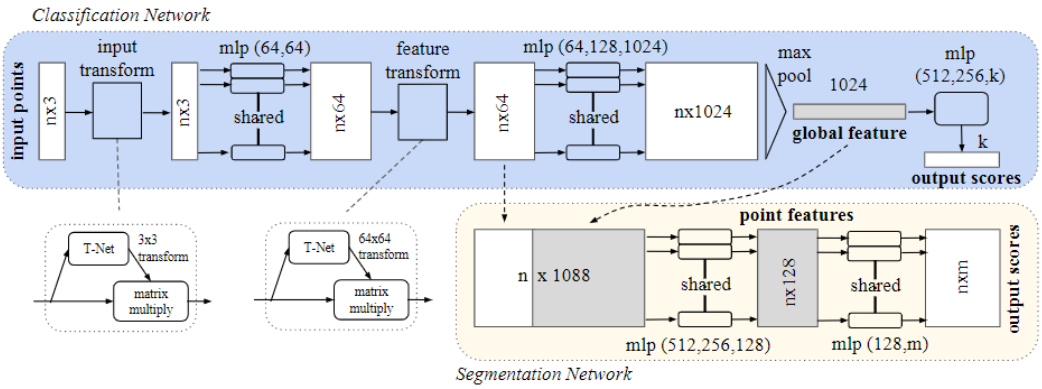


Figure 2: PointNet Architecture

2.2 DGCNN

A dynamic graph convolution neural network is a deep learning model that encodes both global and local geometric information in point clouds. DGCNN constructs a graph of the point cloud using K nearest neighbors, either using the number of neighbors or a threshold radius in euclidian space. Convolution is performed around the neighborhood. The convolution is named EdgeConv, which is order invariant. EdgeConv is coded as a 1D convolution in Pytorch. In convolving a number of different features can be extracted. For instance, if we subtract the center point from all the K nearest neighbors, we can effectively have the edge features which encode the local geometry. The paper concatenates the center pixel coordinates with the edge coordinates to form a 6-dimensional

feature that encodes both local and global information. The features dimension will have a number of neighbors on the last axis. To aggregate, the features max-pooling or average pooling is used as they are order-invariant operations. The latent space is of 512 dimensions, which is followed by a fully connected classifier layer that maps to 40 classes of the ModelNet-40 dataset. DGCNN acquired exceptional performance on classification with an accuracy of 93.5

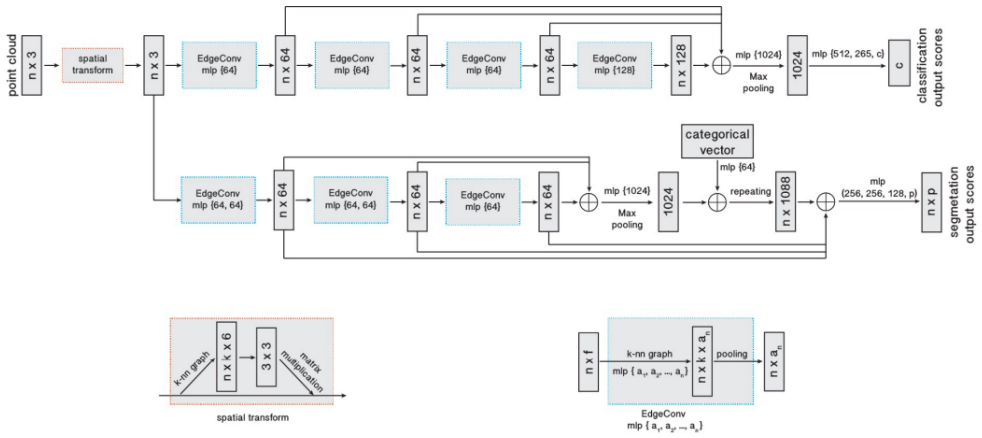


Figure 3: DGCNN Architecture

3 Attack Methodologies

3.1 Fast-gradient sign method

Fast-gradient sign attack method (FGSM) is the earliest and most fundamental technique for adversarial attacks, which is basically a gradient-based optimization algorithm. In this method, the gradient of the cost function is calculated with respect to the point cloud data. This is done because the objective is to create an point cloud that maximises the loss. A method to accomplish this is to find how much each point in the cloud contributes to the loss value, and add a perturbation accordingly. This works pretty fast because it is easy to find how each input point contributes to the loss by using the chain rule and finding the required gradients. Hence, the gradients are taken with respect to the point cloud. In addition, since the model is no longer being trained (thus the gradient is not taken with respect to the trainable variables, i.e., the model parameters), and so the model parameters remain constant. The only goal is to fool an already trained model.The problem can be formulated as below:

$$adv_X = x + \epsilon * sign(\nabla_x J(\theta, x, y)) \tag{1}$$

The adversarial example is generated using the gradient which maximizes the loss function J. here is a hyperparameter, which is a multiplier to ensure that perturbations are small.

3.2 Saliency map attack

A saliency map is a picture in computer vision that emphasizes the region on which people's eyes focus initially. A saliency map's purpose is to portray the relevance of a pixel to the human visual system. Existing efforts on model interpretation and vulnerability have created saliency maps for pictures to establish which pixels are crucial to model-recognition and how pixel values affect recognition performance. Pointcloud saliency map assigns each point x_i a saliency score, i.e. , s_i , to reflect the contribution of x_i. Point dropping is a method to evaluate the veracity of the saliency map. If the saliency map is accurate, then dropping points with the highest(positive)/lowest(negative) saliency scores will degrade/improve recognition performance. Ideally, high (positive) saliency scores indicate significant positive contributions to the recognition result.

The saliency map attack (SMA) is an adversarial strategy used to fool classification algorithms. It's yet another gradient-based whitebox method. Papernot et al. suggested utilizing the gradient of loss

for each class label with respect to each component of the input, i.e. the jacobian matrix, to identify the sensitivity direction. Using the equation, the saliency map is then utilized to locate the dimension with the greatest error (2). Note the additional parameter gives us extra flexibility for saliency map construction, and optimal choice of would be problem specific. In the following experiments, we simply set α to 1, which already achieves remarkable performance.

$$s_i = -\frac{\partial L}{\partial r_i} r_i^{1+\alpha} \tag{2}$$

We drop the points with highest saliency scores iteratively. Specifically, in each iteration, a new saliency map is constructed for the remaining points, and among them n/T points are dropped based on the current saliency map, where n is total number of points to drop and T are the number of iterations.

4 Experiments

We use the public dataset, ModelNet40, to test the adversarial attack methods. ModelNet40 provides a clean collection of 12,311 meshed 3D CAD models for the 40 most common object categories, e.g. bathtub, bed, chair etc. We trained the Pointnet and DGCNN on ModelNet40 dataset and the training accuracy of PointNet we acheived is 78% and on DGCNN an accuracy of 92%. This difference stems from the fact that DGCNN uses EdgeConv which allows it build a neighborhood map of features and also capture both local and global features whereas PointNet captures only global features as discusses in above section. We used default parameters for both the model and kept number of points to be dropped and epsilon values as the only variable.

4.1 Results on FGSM

We show the effectiveness of fast-gradient sign method, we calculate the accuracy of the attack for both PointNet and DGCNN model on ModelNet40 dataset. As we increase the value of ϵ, the accuracy of both the classification models decreases. The overall accuracy of the PointNet model is between 4.13%-72.57%, and of the DGCNN model is between 5.31%-90.15% for ϵ between 0.001 and 1, as shown in Table 1.

Figure 5. shows the visualization of adversarial point clouds for airplane for various ϵ values. As we increase the ϵ values, the classification model classifies the airplane as cup, desk, flowerpot and door. For ϵ values of 0.001 and 0.01, the point cloud is visible as an airplane, but increasing ϵ value to 0.1 and 1 distorts the point cloud to a greater extent.

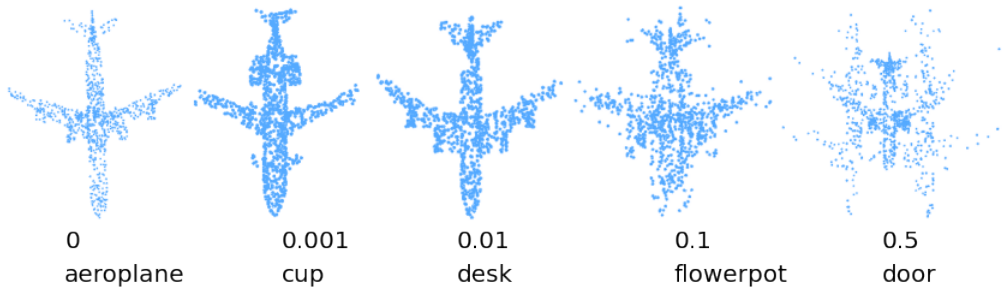


Figure 4: Fast gradient sign based attack Point cloud

4.2 Results on SMA

To show the effectiveness of saliency map attack, we calculate the accuracy of the attack for both PointNet and DGCNN model on ModelNet40 dataset. We refer the number of points dropped as *n*. As we increase the value of *n*, the accuracy of both the classification models decreases. The overall accuracy of the PointNet model is between 7.90%-57.62%, and of the DGCNN model is between 50.32%-86.95% for n between 50 and 300, as shown in Table 2.

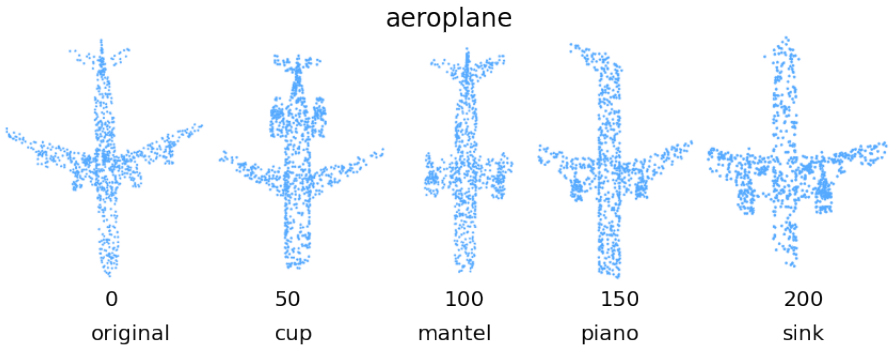


Figure 5: Saliency map attack Point cloud

Table 1: Accuracy of PoineNet and DGCNN under FGSM Attack on ModelNet40 dataset

FGSM attack		
Espilon	PointNet	DGCNN
0.001	72.57	90.15
0.005	63.0	85.21
0.01	54.94	82.17
0.05	37.44	57.86
0.1	23.26	32.74
0.5	4.21	10.05
1	4.13	5.31

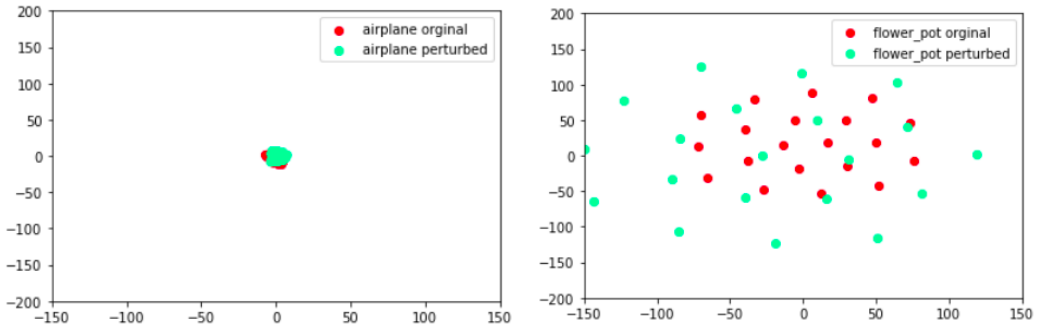
The adversarial point clouds of an airplane are visualized in Fig. 5. For the point clouds shown in the figure, the saliency map attack identifies important segments that distinguishes it from other classes, and fools the classification model by dropping these points. After dropping the 50 points with highest saliency score, the classification model classifies the airplane point cloud as cup, but a human will still be able to recognize the point cloud as airplane. As we increase the number of points dropped, the model classifies the airplane as various other classes.

5 Conclusion

In summary, we tested the robustness of PointNet and DGCNN in classification tasks on ModelNet40 dataset using adversarial attacks such as fast gradient sign method and saliency map attack.Among all the two state-of-the-art DNN models for 3D point clouds, DGCNN appears to be the most robust model to adversarial attack, which indicates DGCNN depends more on the entire point cloud rather than certain point or segment. PointNet does not capture local structures, making it the most sensitive model to adversarial attacks.

Table 2: Accuracy of PoineNet and DGCNN under SMA Attack on ModelNet40 dataset

SMA attack		
Points Drop	PointNet	DGCNN
50	57.62	86.95
100	41.65	81.521
150	29.17	74.23
200	20.10	67.5
250	13.21	58.79
300	7.90	50.32



(a) (b)

Figure 6: Visualization of tsne-projection of Latent space of Pointnet for epsilon 0.05

We did some analysis to determine the most robust and vulnerable class to the attacks by calculating the precision and recall of all the classes in ModelNet-40. From Fig 7. we can infer that airplanes, persons, chairs, etc classes are very robust to attacks while flowerpot class is very vulnerable to FGSM attack. Fig 6 show the tsne-projection of the Pointnet's latent space features of airplane class and flowerpot class respectively. The airplane class seems to be very closely clustered even after the attack, but flowerpot classes' features get even more scattered after the attack. This might confuse the model to make false predicitons. Also from Fig 4, for epsilon=0.1, an airplane gets misclassified as the flowerpot, which is evident to the naked eye to some extent. This means the feature of flowerpot is very scattered and gets overlapped with many of the other classes in the dataset. As persons and chairs in the dataset have unique structures compared to other objects in the dataset, their features will be highly distinct from other classes.

6 Acknowledgments

We thank Prof. Ding Zhao and the teaching staff for the useful discussions and guidance.

7 References

[1] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[2] Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in neural information processing systems 30 (2017).

[3] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon. Dynamic graph cnn for learning on point clouds. arXiv preprint arXiv:1801.07829, 2018.

[4] Zheng, Tianhang, et al. "Pointcloud saliency maps." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019.

[5] Xiang, C., Qi, C.R., Li, B.: Generating 3D Adversarial Point Clouds. arXiv preprint arXiv:1809.07016 (2018).

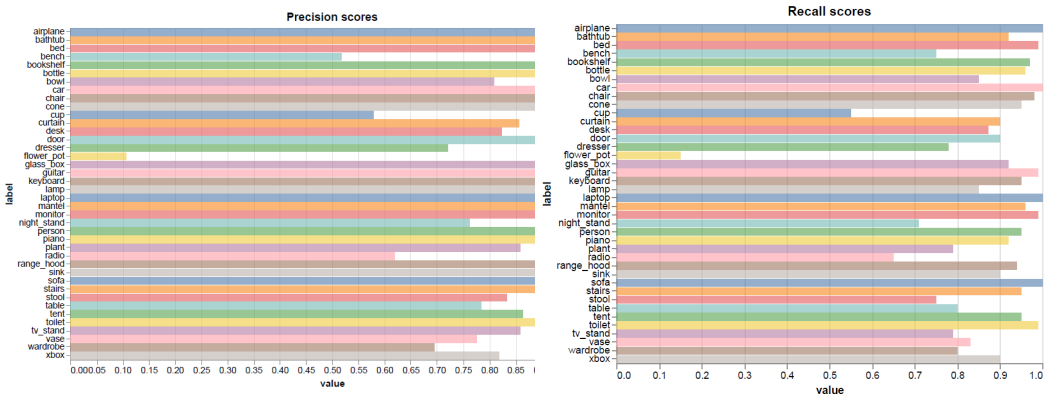
[6]Yu Zhang, Gongbo Liang, Tawfiq Salem, and Nathan Jacobs. Defense-pointnet: Protecting pointnet against adversarial attacks. In 2019 IEEE International Conference on Big Data (Big Data), pages 5654–5660. IEEE, 2019.

[7] D. Liu, R. Yu and H. Su, "Extending Adversarial Attacks and Defenses to Deep 3D Point Cloud Classifiers," 2019 IEEE International Conference on Image Processing (ICIP), 2019, pp. 2279-2283, doi: 10.1109/ICIP.2019.8803770.

8 Appendix

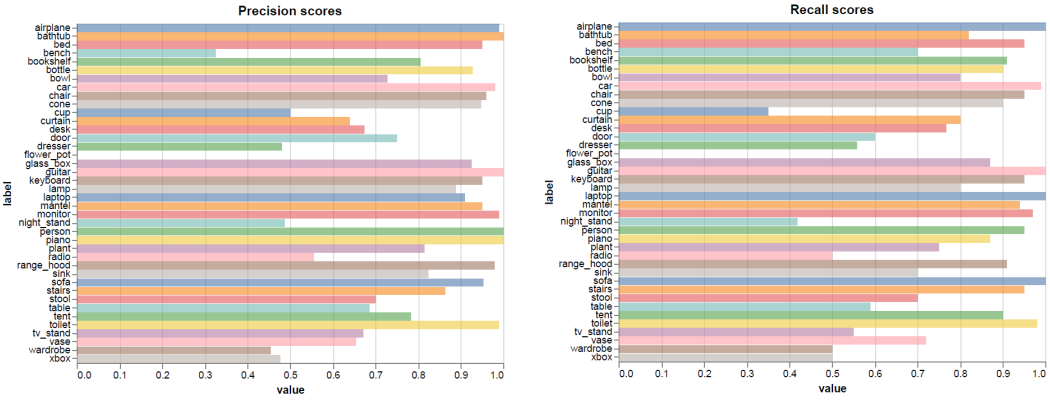
8.1 Plots for FGSM Attack on DGCNN Model

The following figures show the precision and recall of DGCNN model for different epsilon values.



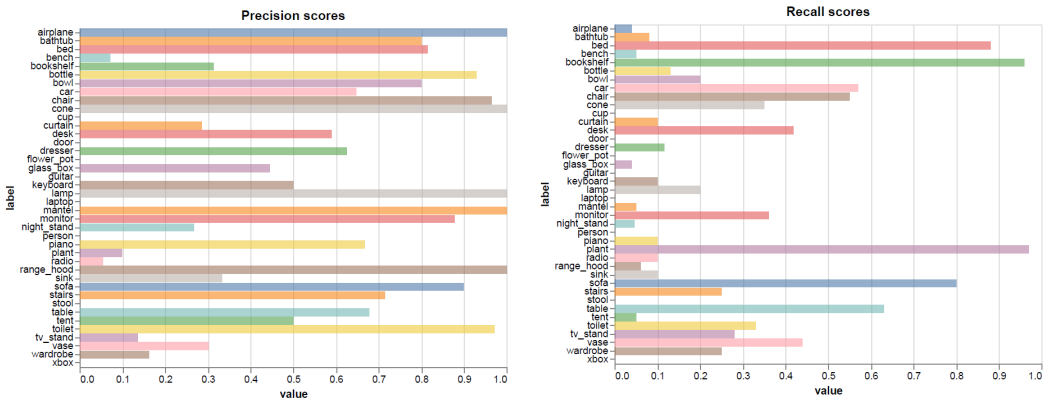
(a) (b)

Figure 7: Visualization of Precision and Recall of DGCNN model for epsilon 0.001



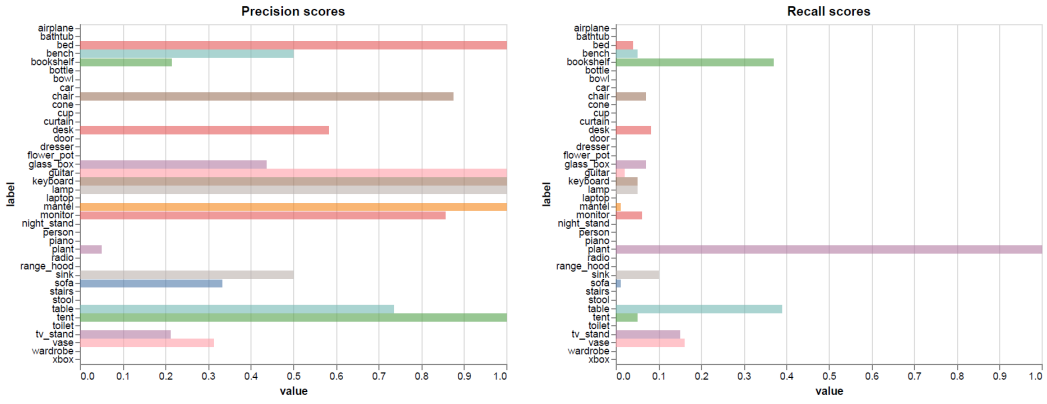
(a) (b)

Figure 8: Visualization of Precision and Recall of DGCNN model for epsilon 0.01



(a) (b)

Figure 9: Visualization of Precision and Recall of DGCNN model for epsilon 0.1

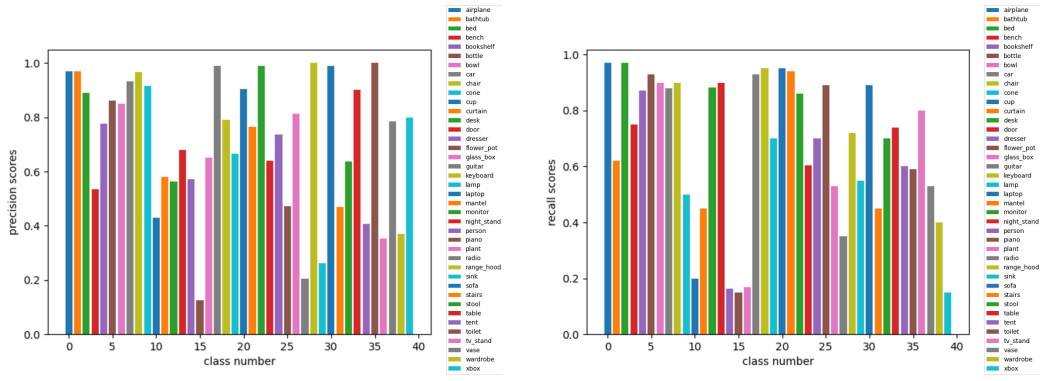


(a) (b)

Figure 10: Visualization of Precision and Recall of DGCNN model for epsilon 0.5

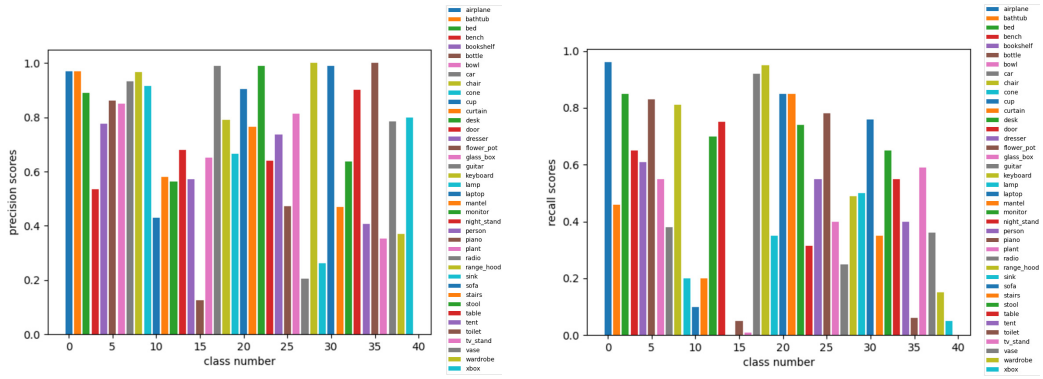
8.2 Plots for FGSM Attack on PointNet Model

The following figures show the precision and recall of PointNet model for different epsilon values.



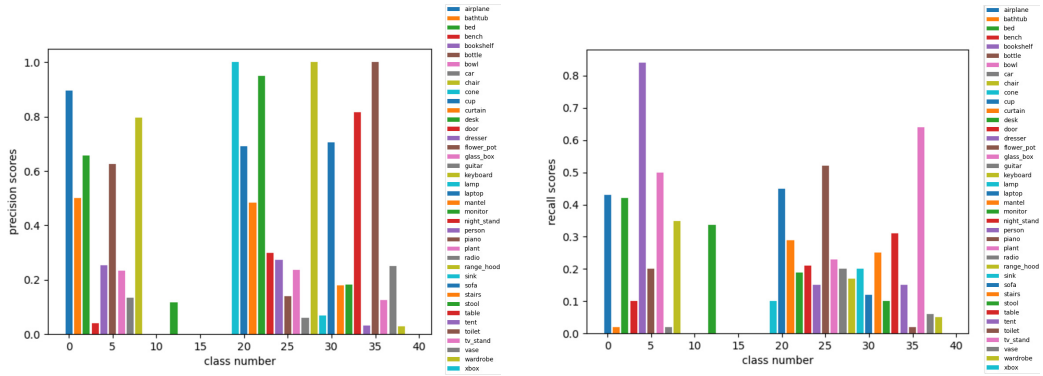
(a) (b)

Figure 11: Visualization of Precision and Recall of PointNet model for epsilon 0.001



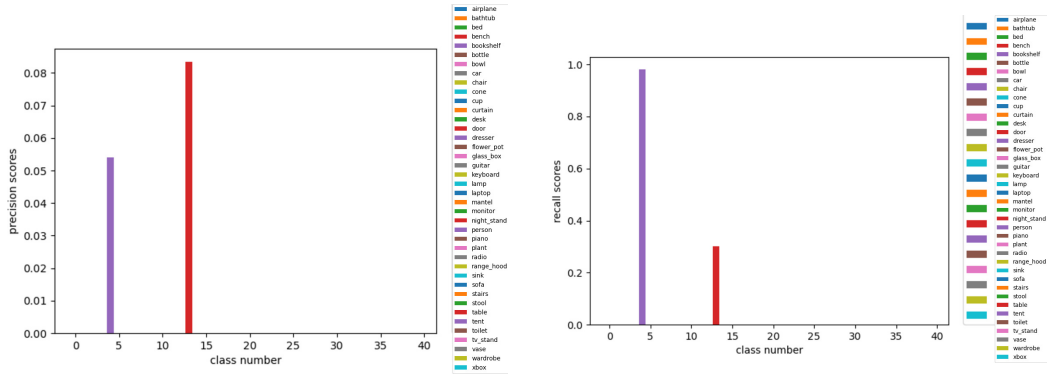
(a) (b)

Figure 12: Visualization of Precision and Recall of PointNet model for epsilon 0.01



(a) (b)

Figure 13: Visualization of Precision and Recall of PointNet model for epsilon 0.1

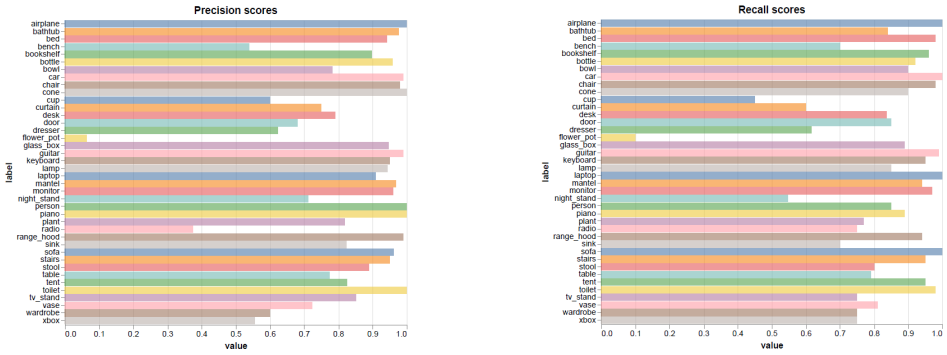


(a) (b)

Figure 14: Visualization of Precision and Recall of PointNet model for epsilon 0.5

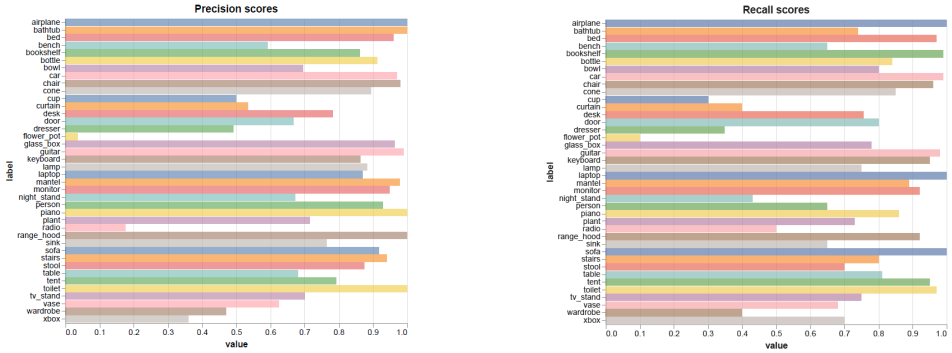
8.3 Plots for SMA Attack on DGCNN Model

The following figures show the precision and recall of DGCNN model for different drop values.



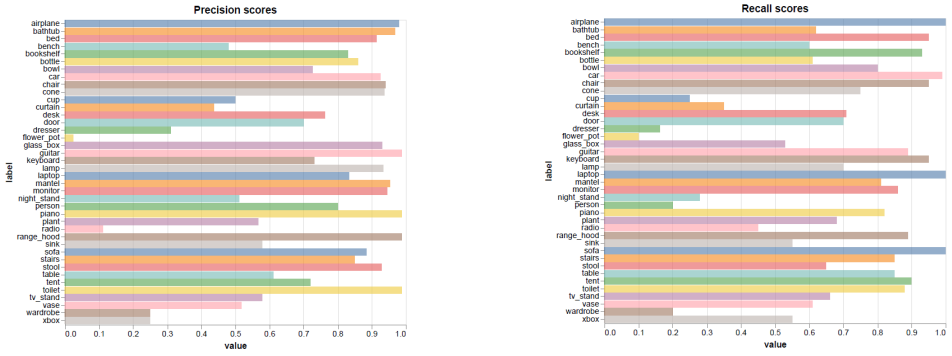
(a) (b)

Figure 15: Visualization of Precision and Recall of DGCNN model for dropping 50 points



(a) (b)

Figure 16: Visualization of Precision and Recall of DGCNN model for dropping 100 points

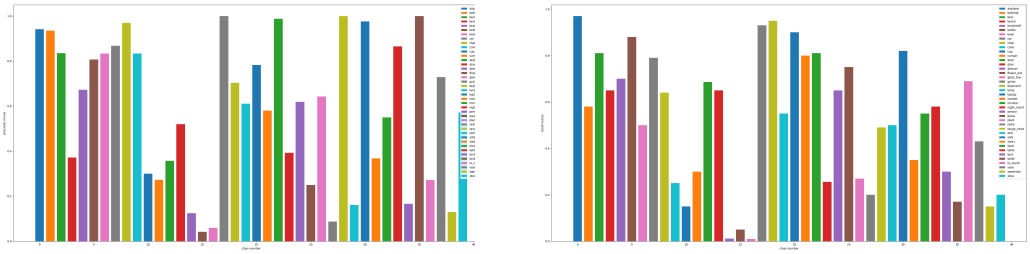


(a) (b)

Figure 17: Visualization of Precision and Recall of DGCNN model for dropping 150 points

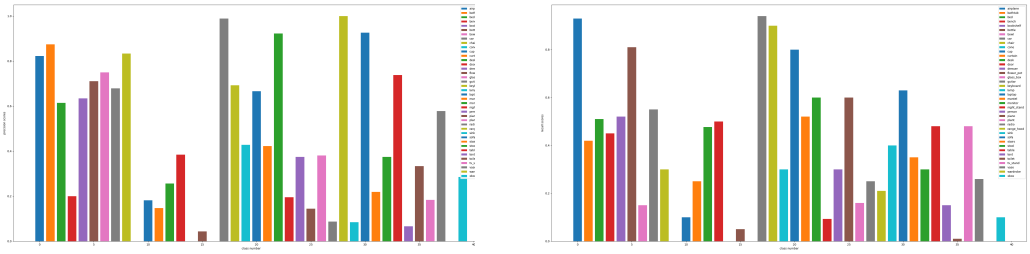
8.4 Plots for SMA Attack on PointNet Model

The following figures show the precision and recall of PointNet model for different drop values.



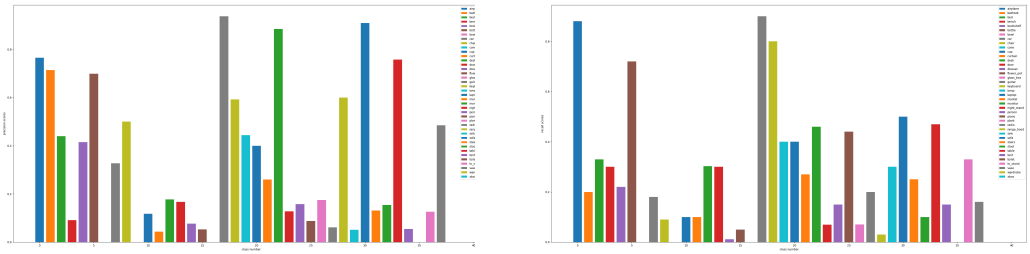
(a) (b)

Figure 18: Visualization of Precision and Recall of PointNet model for dropping 50 points



(a) (b)

Figure 19: Visualization of Precision and Recall of PointNet model for dropping 100 points



(a) (b)

Figure 20: Visualization of Precision and Recall of PointNet model for dropping 150 points