LiDAR Segmentation-based Adversarial Attacks on Autonomous Vehicles

Blake Johnson
joh18484@morris.umn.edu
Division of Science and Mathematics
University of Minnesota, Morris
Morris, Minnesota, USA

Abstract

Autonomous vehicles utilizing LiDAR-based 3D perception systems are susceptible to adversarial attacks. This paper focuses on a specific attack scenario that relies on the creation of adversarial point clusters with the intention of fooling the segmentation model utilized by LiDAR into misclassifying point cloud data. This can be translated into the real world with the placement of objects (such as road signs or cardboard) at these adversarial point cluster locations. These locations are generated through an optimization algorithm performed on said adversarial point clusters that are introduced by the attacker.

Keywords

Autonomous Vehicles, LiDAR, Point Cloud Semantic Segmentation, Neural Networks, Adversarial Attacks

1 Introduction

Autonomous vehicles (AVs), or vehicles capable of driving without human input (self-driving), are on the rise in both development and deployment. Some examples of this are Waymo becoming the first driverless taxi service provider in December 2020, and Honda as the first manufacturer to sell a legally approved self-driving car in March 2021. AVs use various sensors to perceive their surroundings including cameras, radar, GPS, sonar, and LiDAR (Light Detection and Ranging) [6]. Both cameras and LiDAR are the most crucial sensors for an AV's perception system. LiDAR sensors are particularly beneficial as they can provide three-dimensional (3D) pixel information versus camera sensors, which are limited to two-dimensional (2D) pixel information [7].

LiDAR sensors work by firing laser pulses and capturing their reflections using photodiodes, or tools used to measure light intensity. The time it takes for these pulses to reflect back to the photodiode allows for accurate measurements of the distance between a potential obstacle and the LiDAR sensor [1]. LiDAR can generate a point cloud, or a large assortment of tiny individual points plotted in a 3D space, by shooting laser pulses at various horizontal and vertical angles [2]. Once the point cloud is produced, the LiDAR-based perception system in the AV can perform the core task of point cloud semantic segmentation. Point cloud segmentation aims to divide point clouds into regions meaningful to human perception. These regions are labeled accordingly with classes such as grass, road, building, and vehicle in the form of distinctly colored pixels [8]. Then, LiDAR returns this labeled point cloud back to the AV. Since LiDAR point cloud segmentation provides a prevailing semantic scene understanding, it has become integral to autonomous driving perception systems. There are multiple applications of point cloud segmentation, a crucial one being the identification of road obstacles such as pedestrians or other vehicles. Another important application is deriving road boundary information to know drivable areas versus non-drivable areas such as roadside grass and sidewalks. Point cloud semantic segmentation can also use sporadic point cloud data to reconstruct surrounding 3D environments and model objects such as traffic signs or lampposts.

Despite being widely used in AVs, point cloud segmentation models are assumed to perform in secure and consistent environments. This is not always the case as the increasingly influential role that segmentation models play leads to an increase in the risk of malicious attacks. Neural networks (NNs) are used to process point clouds in these segmentation models. When taking image pixels from LiDAR as inputs, NNs have shown to be vulnerable to adversarial attacks where an attacker introduces a small deviation in these input pixels. This allows for an attacker to potentially change the output classification of the NN in a drastic way. One possible way for an attacker to perform such an attack would be by manipulating the driving environment with, for example, the placement of a cardboard sign on the roadside. This may fool the LiDAR perception system into a misclassification, resulting in potentially disastrous consequences.

This paper summarizes effective adversarial attacks against LiDAR-based perception systems, specifically the point cloud semantic segmentation aspect of these systems, where the perception results of a victim AV are changed in ways such as a vehicle classified as the road or the road classified as vegetation. There could be various outcomes of this attack, ranging from sudden stops and direction changes to collisions with other vehicles or objects. The specific framework in which this attack is carried out in this paper involves an attacker deriving adversarial locations for the placement of objects in the physical world, known as adversarial objects. These objects could be anything that reflects the LiDAR laser such as cardboard or road signs and, after being placed at

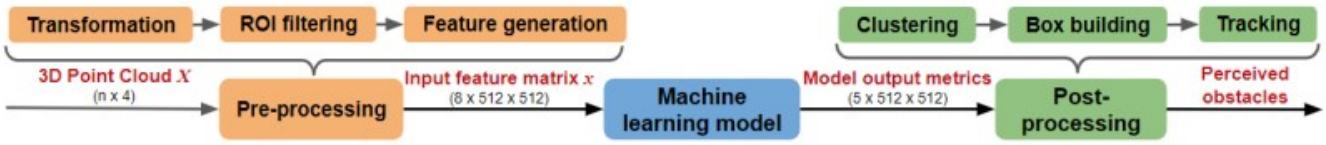

Figure 1. Highlights the three main steps of the LiDAR-based perception data processing pipeline that takes place in the perception module of the Baidu Apollo [1].

Feature	Description
Max height	Maximum height of points in the cell.
Max intensity	Intensity of the brightest point in the cell.
Mean height	Mean height of points in the cell.
Mean intensity	Mean intensity of points in the cell.
Count	Number of points in the cell.
Direction	Angle of the cell's center with respect to the origin.
Distance	Distance between the cell's center and the origin.
Non-empty	Binary value indicating whether the cell is empty or occupied.

Figure 2. Each input feature of the CNN model and its description [1]

Metrics	Description
Center offset	Offset to the predicted center of the cluster the cell belongs to.
Objectness	The probability of a cell belonging to an obstacle.
Positiveness	The confidence score of the detection.
Object height	The predicted object height.
Class probability	The probability of the cell being a part of a vehicle, pedestrian, etc.

Figure 3. Each output metric of the CNN model and its description [1]

these adversarial locations, can effectively fool the point cloud segmentation model used by the victim AV into a misclassification [8].

Before delving into the specifics of the attack, Section 2 will give background on the details of neural networks, how they are vulnerable to adversarial examples, and what those even are. It will also cover in-depth how LiDAR-based perception works in terms of processing raw input data. After introducing some adversarial attack scenarios in Section 3, Section 4 will set up the framework of an attack, covering the process of adversarial location generation and the white-box attack. Then, this paper will discuss the execution of the proposed adversarial attacks and the general effectiveness of carrying them out in Section 5. Finally, it will conclude on the matter of adversarial attacks targeting LiDAR-based perception systems used in AVs.

2 Background

2.1 Neural Networks

Neural networks are a method of machine learning where a computer is trained to correctly classify data by analyzing manually labeled sample data. Convolutional neural networks (CNNs) are the most widely used for application in LiDAR perception systems as they are specially designed to analyze and process image pixel data [3]. CNNs consist of three main layers: the convolutional layer, the pooling layer, and the fully-connected layer. The convolutional layer, where the majority of computation occurs, takes input pixel data and extracts different features through a filter to create a feature map. There can be multiple convolutional layers where each one extracts features from a different portion of pixels from the input image. This can be visualized through the example of an image of a bicycle which is made up of various parts (features) such as a frame, pedals, handlebars, and wheels. These different features would be recognized by sequential convolutional layers as lower-level patterns that create a connected higher-level pattern. Then, the pooling layer further filters the extracted features with the goal of reducing size of the feature maps. Finally, the fully-connected layer preforms the task of classifying extracted features along with attaching metrics mentioned in Section 2.3.2 [4].

In the case of the machine learning models used in LiDAR for AVs, they would typically be trained to classify meaningful regions of generated 3D point cloud datasets of road scenes (e.g. "Vehicle" or "Ground"). These models would be trained on road scenes with meaningful regions already labeled accordingly. This data can come from either publicly available datasets, such as SemanticKITTI, or it can be gathered by any individual with a LiDAR-based perception system [7].

2.2 Adversarial Examples

Adversarial examples are inputs meticulously created with the intention of fooling machine learning models that utilize neural networks. The result of these examples is the misclassification of a given input [5]. Take, for example, a machine learning model M. M takes an input x and returns a label y for the given input. An attacker's goal, then, is to generate adversarial examples x' to input into M such that $M(x') \neq y$ (the model mislabels the input) or $M(x') = y'$, with y' being a target label [1].

2.3 LiDAR-based Perception in AV systems

Before the data collected from LiDAR-based perception systems can become useful, it must be processed. This process

consists of three principal steps: pre-processing, CNN-based segmentation, and post-processing. Figure 1 outlines these steps with the Machine Learning model step referring to CNN-based segmentation.

2.3.1 Pre-Processing. Raw data fed into the LiDAR sensors is known as the 3D point cloud, which we know to be a collection of small individual points that make up a 3D space. This 3D point cloud can be denoted as X. X has dimensions n x 4 with n being the number of data points. Each of these data points is a four-dimensional vector (the fourth dimension being time) having 3D coordinates: w_x (width), w_y (height), and w_z (depth). Firstly, X is converted into an absolute coordinate system. Then, the Region of Interest (ROI) filters out irrelevant regions of the 3D point data, such as a building way outside of the road. The individual points of this filtered 3D point cloud are mapped to 512 x 512 cells according to their w_x and w_y coordinates. Eight different features are generated for each of these cells. These features are all listed in Figure 2. This process generates a feature matrix x (8 x 512 x 512), becoming the input for a neural network [1].

2.3.2 CNN-based Segmentation. After a feature matrix is generated, it is fed as input into a CNN, a type of neural network used in image recognition and processing because of its ability to discern patterns in images. The CNN produces an output of various metrics for each cell including objectness, positiveness, object height, center offset, and class probability. Each of these metrics is further described in Figure 3 [1].

2.3.3 Post-Processing. Clustering is the first process of three in the final post-processing step. Only cells with an objectness (the likelihood that the cell belongs to an obstacle) value over 0.5 (default threshold) are considered for this process. A connected graph is constructed from the output metrics of each cell in order to create object cluster candidates. These candidates are further filtered through average positiveness (confidence rating) by taking clusters with values over 0.1 (default threshold). Next, the construction of the bounding box is remodeled in the box-building sub-process. An obstacle candidate's dimensions taken from its assigned point cloud (length, width, and height) are used for this. The final sub-process, the tracker, generates tracked obstacles by connecting each individual frame of the processed results. This becomes the final output of the LiDAR-based perception system [1].

Now, the AV can make driving decisions from the new information (e.g. an obstacle's position, shape, and type) supplied through the LiDAR-based perception.

3 Attack Scenarios

Zhu et al. present research on a specific attack with the goal of changing the output labels of the 3D point cloud segmentation model used by a victim AV [8]. The attack can be focused on two distinct scenarios: vehicle/obstacle hiding attack and changing of road surface attack.

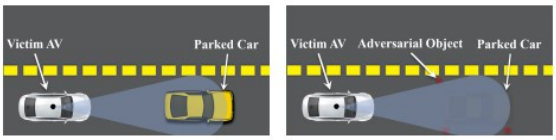

Figure 4. A vehicle/obstacle hiding attack scenario. With the placement of a few adversarial objects the parked car in front of the victim AV disappears from its sensors [8].

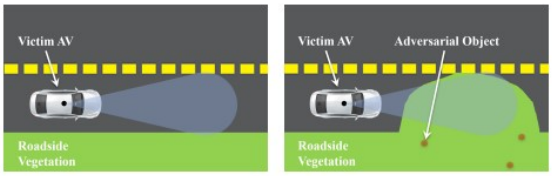

Figure 5. A road surface changing attack scenario. With the placement of a few adversarial objects, the roadside vegetation (grass) creeps onto and covers the road according to the victim AV's sensors [8].

A vehicle/obstacle hiding attack, as shown in Figure 4, deals with the driving environment consisting of a car parked on the road or parking lot. This could simply be a car stopped at a traffic light or one parked intentionally by the attacker. The goal of this attack scenario is to make this parked car disappear from the LiDAR-based perception system of the approaching victim AV. This can be done by adding adversarial objects (e.g., road signs or cardboard pieces) on the roadside or around the parked car. The desired results of this attack scenario are rear-end collisions and the chain reactions that follow.

The other attack scenario shown in Figure 5, changing of road surface attack, deals with the driving environment of an open road. The goal of this attack scenario is to replace the road perceived by the victim AV with something non-drivable such as vegetation or a sidewalk. This can be done once again by adding adversarial objects on the roadside or road itself. The desired result of this attack scenario is direction changes or sudden stops by the victim AV leading to traffic accidents [8].

4 Generation of Adversarial Examples

4.1 Attack Framework

The process of procuring adversarial object locations begins with the attacker mimicking the potential driving patterns and behavior of the victim AV (autonomous vehicle). 3D point cloud data of the target environment is collected through this step and labeled as the original point clouds.

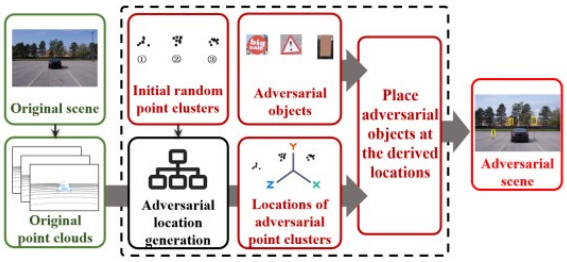

Figure 6. The framework of an adversarial attack. Section 4.1 expands on this process [8].

The attacker wants to collect exhaustive point cloud data of the surrounding scene in order to strengthen their data to be both relevant and accurate to what data the victim AV actually collects.

Now, the attacker wants to find locations in the physical world to place adversarial objects that can trick the incoming victim AV despite any complications that may arise. These adversarial objects are represented as individual random point clusters in the point clouds. Clusters are randomized in both their shape and the number of points and are kept random throughout the whole process. The reasoning for this is flexibility: if the attacker can find locations where the placement of random point clusters fools the LiDAR point cloud segmentation model, then any adversarial object placed at these locations will do the same. After initializing the random point clusters into the LiDAR model, they are then combined with the original point clouds. This combination of the original input point clouds and the random point clusters produces what are known as adversarial point clusters.

The attacker can now derive optimal locations for the center of these adversarial clusters where the point cloud segmentation model is tricked into misinterpreting the point cloud data. The attacker will then place their physical adversarial objects at these locations to carry out the attack on the victim AV. Figure 6 outlines the key steps of this attack framework.

Generating optimized locations for adversarial objects can be challenging. In order to achieve heightened misclassification of the target points by the segmentation model, we introduce three different measurements of error: segmentation loss L_{seg}, semantic loss L_{sem}, and occlusion loss L_{occ} [8].

4.2 Loss Variables

4.2.1 Segmentation Loss. Segmentation loss is a measurement of the distance between the target label and the predicted label of the target points. In the context of a vehicle hiding attack, the target points refer to the parked car which would be labeled as "Vehicle"; the target label for these points is then "Ground". As for a road surface changing attack, the target points would be the road in front of the victim AV while the target label for these points would be "Vegetation". An attacker aims to maximize the amount of misclassified points so that either a parked car can be hidden or the road surface can be changed as frequently (successfully) as possible [8].

Minimizing segmentation loss can aid in maximizing the amount of misclassified points, but might not be enough to procure optimal adversarial point cluster locations. This is where semantic loss is introduced.

4.2.2 Semantic Loss. Semantic loss is the associated measurement of a method named semantic misleading where the goal is to make the semantic features, or the meaning and representation of the reference point clouds and adversarial point clouds similar. Reference point clouds, which describe the scene the attacker desires, can be easily obtained from public data sets of 3D point clouds or simply collected by the attacker. Then a feature extractor is used to extract semantic features from both the reference point clouds and adversarial point clouds.

Semantic loss, then, is a measurement of the similarity of the extracted semantic features of these two point cloud types [8]. Now we can introduce occlusion loss.

4.2.3 Occlusion Loss. The generation of strong adversarial point clusters by minimizing both segmentation loss and semantic loss is possible, but there remains the question of whether or not the locations for these clusters would be obstructed by other objects in the physical world. If so, it would prevent the LiDAR perception system from ever observing the placed adversarial objects. Occlusion loss is thus created to make sure that no real-world objects obstruct the view of added, adversarial objects.

4.2.4 Total Loss Function. The strongest adversarial locations where the victim AV is consistently fooled can then be generated through the optimization of a total loss equation $L_t = L_{seg} + \alpha L_{sem} + \beta L_{occ}$ where α and β are predefined hyperparameters designed to balance the three losses. Figure 7 walks through the complex adversarial point cluster generation process, including these three loss functions and where they derive from.

The attack can be strengthened further still by minimizing the gradient of total loss, L_t'. The gradient of a loss function indicates how small perturbations, or deviations from the inputs, change the loss. An attacker can find optimal values able to tolerate small perturbations by minimizing the gradient of total loss. These values help to generate adversarial clusters resistant to location errors [8].

4.3 White-Box Attack

In contrast to a black-box attack, which won't be discussed in this paper, in a white-box attack attackers know both which semantic segmentation model and LiDAR-based perception

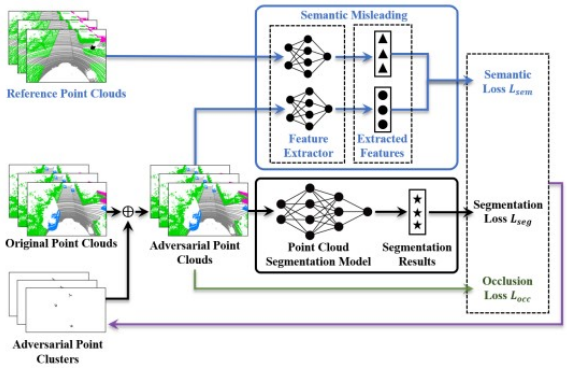

Figure 7. Adversarial location generation overview [8].

system are being utilized by the victim AV. Thus, training a semantic feature extractor (CNN) to be used in the semantic misleading method is not necessary. Instead, since the feature layer of the known target segmentation model contains the global features of the input point cloud scenes, we can use this layer as our feature extractor.

Our derived locations of adversarial point clusters also need to be reasonable, meaning placement of the real-world adversarial objects should not be underground or out of reach for the attacker. This can be done through bounding boxes that constrain the center of each point cluster. Now, the locations of adversarial point clusters can be derived through this optimization problem:

$$\begin{aligned} \min_{\{O_k^a\}_{k=1}^K} \quad & L^* = L_t + \eta L_t' \\ s.t. \quad & \{x_{k1}^a\} \in [A_{min}, A_{max}], \\ & \{x_{k2}^a\} \in [B_{min}, B_{max}], \\ & \{x_{k3}^a\} \in [C_{min}, C_{max}], \end{aligned} \tag{1}$$

where $L_t = L_{seg} + \alpha L_{sem} + \beta L_{occ}$, L_t' is the gradient of L_t, and η is a pre-defined hyperparameter. $\{x_{k1}^a\}$, $\{x_{k2}^a\}$, and $\{x_{k3}^a\}$ are the x, y, and z coordinates for a given point cluster's center with $[A_{min}, A_{max}]$, $[B_{min}, B_{max}]$, and $[C_{min}, C_{max}]$ being the respective bounds for each dimension [8].

5 Attack Execution

5.1 Setup

The specific attack detailed in *Adversarial Attacks against LiDAR Semantic Segmentation in Autonomous Driving* is extensively evaluated amongst five different point cloud segmentation models used in LiDAR-based perception used in AVs. These models are PointNet, PointNet++, PointASNL, Cylinder3D, and SqueezeSeg [8]. A white-box attack is considered here.

In order to gather point cloud data to train these models, the attackers both used the public dataset SemanticKITTI and collected their own data through a LiDAR device (Ouster OS1-64) mounted on top of a vehicle [8]. They collected data on two college campus roads as well as three separate parking lots. This data was then manually labeled through tools supplied by SemanticKITTI.

Models	Vehicle Hiding	Road Surface Changing
PointNet	82%	78%
SqueezeSeg	77%	66%
Cylinder3D	72%	63%
PointNet++	69%	60%
PointASNL	62%	58%

Table 1. Success rates of attacks using SemanticKITTI data on different segmentation models [8]

5.2 SemanticKITTI Dataset Results

In the case of testing the effectiveness of the attack on the SemanticKITTI public dataset, the attackers randomly chose 20 example scenes each containing 5 consecutive point cloud frames. Adversarial objects are then added to the point cloud scenes in the form of random point clusters. To find the locations of these adversarial clusters, hyper-parameters α, β,and η are assigned values 0.1, 1, and 0.1 respectively while the Adam Optimizer's rate of learning is set to 0.1 [8].

After locations are derived, each adversarial point cluster is replaced by a novel random point cluster 100 different times and the classification results are recorded. The average percentage of points classified as the attacker's target label is denoted as the attack success rate [8]. Table 1 showcases these results for both the vehicle hiding and road surface changing attacks on each segmentation model. We can see that the best performance is attained on PointNet with 82% and 78% for each respective attack scenario while the worst performance occurred on PointASNL with 62% and 58% respectively which still suggests vulnerability to malicious attacks [8].

5.3 Real-World Results

For the real-world attack, the PointNet segmentation model is used.

5.3.1 Vehicle Hiding. Figure 8 visualizes both attacks, with 8a and 8e showing the original scenes for the vehicle hiding attack: a black car parked in the middle of a parking lot and next to the sidewalk and grassy area. 8b and 8f then show the original point cloud segmentation results of these scenes; the blue points indicate the black car (or "Vehicle" class) and the green points indicate the area of grass and trees (or "Vegetation" class) past the sidewalk with the remaining grey points designating the parking lot surface (or "Ground" class). The attack goal is to hide this black car from the PointNet segmentation system (turn the blue points into grey ones).

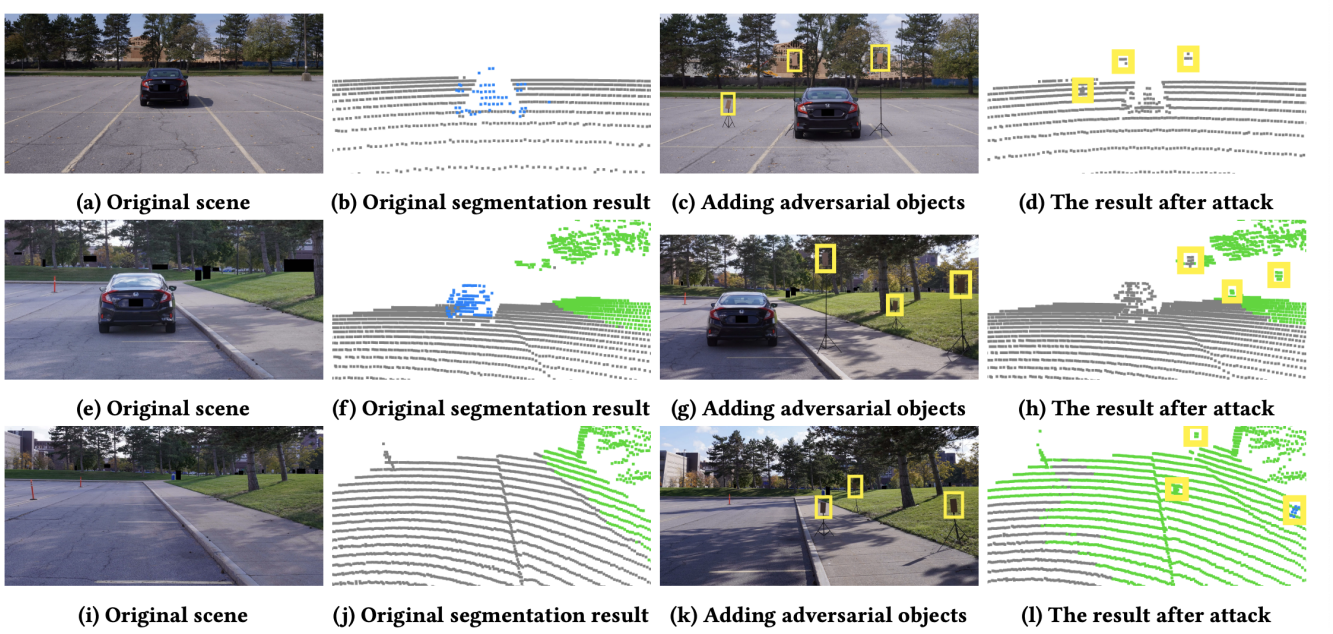

Figure 8. Real-world adversarial examples of two different road hiding attacks and a road surface changing attack

The first step to achieving this is to obtain the original point cloud scene data using the same LiDAR-mounted vehicle described before and driving through the targeted areas. Then, adversarial locations are generated (the attackers decided on 3 adversarial point clusters so 3 locations) [8]. At these locations, pieces of cardboard held by a poster stand are placed as adversarial objects (8c and 8g). It should also be noted that the attackers limited the size of these cardboard pieces to a 30 cm maximum length and width in order to be consistent with the size of the adversarial point clusters [8].

After placing these objects, the scene is visualized in 8d and 8h where we can see the new segmentation results. The adversarial objects generated point clusters (highlighted by yellow boxes) as planned and the result was the blue points of the black car turning grey successfully hiding it from the segmentation model.

5.3.2 Road Surface Changing.

Referencing Figure 8 again, 8i showcases the road surface changing attack's original scene of an empty parking lot near the sidewalk and grassy area. The original segmentation result of this scene is modeled by 8j; most of the scene is classified as the surface of the parking lot with the little bit of vegetation off to the right classified as such. The attack goal here is to change the driveable surface into a non-driveable surface (i.e. "Ground" class to "Vegetation" class).

Once again, the original point cloud scene must be acquired through the LiDAR-mounted vehicle driving through the target area. Then, adversarial locations are generated and cardboard pieces are set up in these locations (8k) [8].

After the objects are in place and the attack is carried out, the resulting point cloud segmentation scene is depicted in 8l. We can see adversarial objects once again generated effective point clusters and the result shows that a large portion of what was originally the "Ground" class is now classified as the green "Vegetation" class. This attack successfully transformed the driveable area of the parking lot into a non-driveable area classified as "Vegetation" [8].

6 Conclusion

With the ever-increasing rise of automated driving and AVs, the rise of malicious attacks and AV vulnerability is sure to follow. This paper explored how the LiDAR-based perception systems in AVs, specifically the segmentation aspect of them, are susceptible to adversarial attacks. It first discussed how LiDAR processes data, considered the role of neural networks in this process, and introduced adversarial attacks. Then it introduced two attack scenarios and their framework before thoroughly detailing the process of generating adversarial locations. Finally, it presented a series of practical attacks, both real-world and on datasets, and showcased their effectiveness. These attacks demonstrated how these attack frameworks can be executed with high success, showing the vulnerability of LiDAR-based perception to malicious attacks.

Acknowledgements

I would like to thank my advisor Kristin Lamberty and my instructor Elena Machkasova for their continued support and abundance of constructive feedback throughout the process.

I would also like to thank Andy Lau for providing external feedback and advice.

References

[1] Yulong Cao, Chaowei Xiao, Benjamin Cyr, Yimeng Zhou, Won Park, Sara Rampazzi, Qi Alfred Chen, Kevin Fu, and Z. Morley Mao. 2019. Adversarial Sensor Attack on LiDAR-Based Perception in Autonomous Driving. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 2267–2281. https://doi.org/10.1145/3319535.3339815

[2] GeoSLAM. 2023. Point clouds for beginners. https://geoslam.com/us/point-clouds/#:~:text=A%20point%20cloud%20is%20essentially,using%20a%203D%20laser%20scanner. [Online; accessed 9-April-2023]..

[3] Larry Hardesty. 2017. Explained: Neural Networks. https://news.mit.edu/2017/explained-neural-networks-deep-learning-0414 [Online; accessed 9-April-2023]..

[4] IBM. 2023. What are convolutional neural networks? https://www.ibm.com/topics/convolutional-neural-networks [Online; accessed 4-May-2023]..

[5] TensorFlow. 2015. Adversarial example using FGSM. https://www.tensorflow.org/tutorials/generative/adversarial_fgsm#:~:text=a%20neural%20network.-,What%20is%20an%20adversarial%20example%3F,the%20contents%20of%20the%20image. [Online; accessed 9-April-2023]..

[6] Wikipedia. 2023. Self-driving car — Wikipedia, The Free Encyclopedia. https://en.wikipedia.org/wiki/Self-driving_car [Online; accessed 9-April-2023]..

[7] Kaichen Yang, Tzungyu Tsai, Honggang Yu, Max Panoff, Tsung-Yi Ho, and Yier Jin. 2021. Robust Roadside Physical Adversarial Attack Against Deep Learning in Lidar Perception Modules. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security* (Virtual Event, Hong Kong) *(ASIA CCS '21)*. Association for Computing Machinery, New York, NY, USA, 349–362. https://doi.org/10.1145/3433210.3453106

[8] Yi Zhu, Chenglin Miao, Foad Hajiaghajani, Mengdi Huai, Lu Su, and Chunming Qiao. 2021. Adversarial Attacks against LiDAR Semantic Segmentation in Autonomous Driving. In *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems* (Coimbra, Portugal) *(SenSys '21)*. Association for Computing Machinery, New York, NY, USA, 329–342. https://doi.org/10.1145/3485730.3485935