WIP: Shadow Hack: Adversarial Shadow Attack Against LiDAR Object Detection

Ryunosuke Kobayashi*, Kazuki Nomoto*†, Yuna Tanaka*, Go Tsuruoka*, Tatsuya Mori*‡§,
*Waseda University †Deloitte Tohmatsu Cyber LLC ‡RIKEN §NICT

***Abstract*—Object detection is a crucial function that detects the position and type of objects from data acquired by sensors. In autonomous driving systems, object detection is performed using data from cameras and LiDAR, and based on the results, the vehicle is controlled to follow the safest route. However, machine learning-based object detection has been reported to have vulnerabilities to adversarial samples. In this study, we propose a new attack method called "Shadow Hack" for LiDAR object detection models. While previous attack methods mainly added perturbed point clouds to LiDAR data, in this research, we introduce a method to generate "Adversarial Shadows" on the LiDAR point cloud. Specifically, the attacker strategically places materials like aluminum leisure mats to reproduce optimized positions and shapes of shadows on the LiDAR point cloud. This technique can potentially mislead LiDAR-based object detection in autonomous vehicles, leading to congestion and accidents due to actions such as braking and avoidance maneuvers. We reproduce the Shadow Hack attack method using simulations and evaluate the success rate of the attack. Furthermore, by revealing the conditions under which the attack succeeds, we aim to propose countermeasures and contribute to enhancing the robustness of autonomous driving systems.**

I. Introduction

In the past few years, the risk of LiDAR (Light Detection and Ranging) sensor attacks in autonomous vehicles has received a lot of attention. These attacks involve the spoofing of sensor readings, potentially causing the object recognition systems of autonomous vehicles, which are based on machine learning models, to misidentify objects. Of particular concern is the method known as "LiDAR spoofing attack," in which malicious signals are injected to trick sensors into recognizing non-existent objects or missing real ones [1, 2]. These attacks target sensors, sensor data processing mechanisms, and machine learning models, and develop methods to manipulate system output using specific input patterns. These attacks represent a new threat to autonomous vehicle sensor systems, and highlight the urgent need to enhance the security of sensor technology and improve the robustness of machine learning models.

This study proposes a new attack vector for sensing systems using LiDAR, named "Shadow Hack," with the aim of understanding its threats and developing effective countermeasures.

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.23051
www.ndss-symposium.org

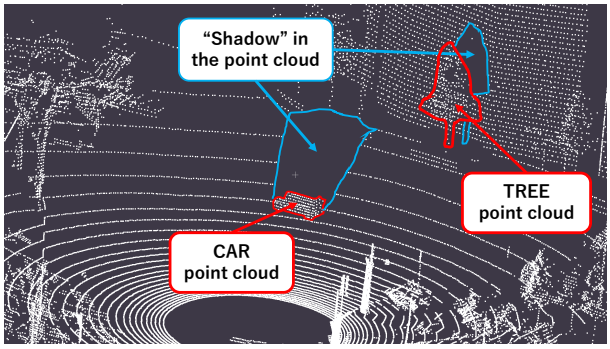

Fig. 1. An example of "Shadow" of the point cloud. "Shadows" are present on the point cloud behind the CAR and TREE.

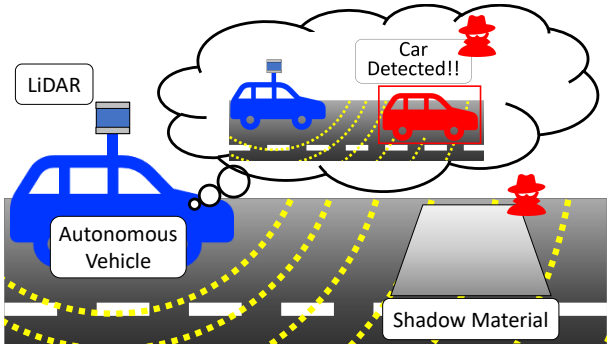

Fig. 2. Attack Overview. Adversarial shadows on the LiDAR point cloud are caused by the LiDAR invisible sheet set up by the attacker, resulting in false detection by the autonomous vehicle.

The concept of this attack lies in exploiting the "shadows" naturally formed in the point cloud data captured by LiDAR sensors (see Figure 1). LiDAR sensors produce point cloud data indicating the presence of objects, but this data also includes the shadows formed behind the objects. Typically, these shadows are ignored in the output of object detection models, but their presence provides important clues for object detection. The Shadow Hack takes advantage of this property of "shadows" by intentionally creating them to fool object detection systems and cause them to malfunction. For example, by placing objects such as an aluminum leisure mat in the environment, false shadows can be created in the point cloud data captured by LiDAR sensors, causing the object detection models to detect non-existent objects (See Figure 2).

In this study, we use AWSIM, an advanced autonomous driving simulator, to verify the effectiveness of the shadow hacking attack. AWSIM is a simulator designed to evaluate the

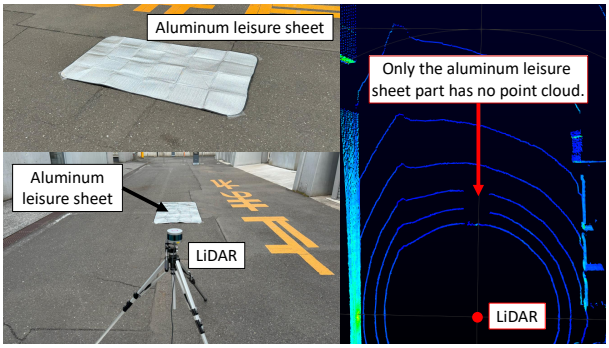

Fig. 3. The alminum sheet does not appear as point clouds.

performance of Autoware, an open-source autonomous driving framework, and provides high-quality, realistic graphics using the Unity3D game engine. It allows developers to place custom 3D object models and simulate three-dimensional measurements from LiDAR sensors on those objects. Researchers can use this setup to mimic complex real-world environments, collect three-dimensional point cloud data measured by LiDAR, and then apply object recognition models to this data. This simulation environment provides an ideal platform to recreate various scenarios of the Shadow Hack attack and accurately measure its impact.

To evaluate the effectiveness of the Shadow Hack attack, our experiments will use two prominent models for point cloud recognition: PointPillars, a representative voxel-based model, and Point-RCNN, a representative point-based model. These models, differing in their approaches, are ideal for assessing the impact of the Shadow Hack attack across various modeling techniques. The experiments will involve applying the Shadow Hack attack to these object detection models to measure the attack's success rate and evaluate the resilience of each model. This comparative approach will help in understanding the vulnerabilities and strengths inherent in current point cloud recognition technologies used in autonomous driving systems.

The Shadow Hack attack demonstrated a 100% attack success rate against the PointPillars object detection model when deployed around a stationary autonomous vehicle in an obstacle-free environment. Furthermore, in 10 urban scenes, the average attack success rate against PointPillars was 58%, indicating a certain level of resilience to this type of attack. Conversely, the PointRCNN model remained impervious to the attack, highlighting its robustness against the Shadow Hack attack.

II. Background and Related Work

A. LiDAR's Limitations in Detecting Certain Materials

LiDAR, a sensor that uses infrared laser pulses to measure distance, faces challenges in detecting certain materials. The basic mechanism of LiDAR is to emit a laser pulse and calculate the distance to an object based on the time it takes for the reflected light to return. While this process is effective for generating point cloud data, it has limitations when it comes to detecting certain materials.

The primary limitation occurs with materials that have unique interactions with near-infrared light, which is commonly used in LiDAR systems. Transparent materials, such as certain types of glass, pose a challenge because they allow near-infrared light to pass through rather than reflect it back to the sensor. Similarly, materials that absorb near-infrared wavelengths, such as some plastics and fabrics, also hinder accurate detection. In addition, light-scattering surfaces, such as aluminum leisure sheets, interfere with accurate measurement by scattering the laser pulses in different directions.

In situations where these materials are present, the LiDAR system may not receive a reflected pulse, resulting in a lack of data points in the point cloud for these objects. This phenomenon creates "shadows" that indicate either the absence of an object or the presence of a material that is difficult for LiDAR to accurately measure. Figure 3 shows a clear example of the limitations of LiDAR material detection. An aluminum leisure sheet on a road is clearly visible in the camera image, but invisible to LiDAR, which fails to detect and represent it in the point cloud due to the light-scattering surfaces. This lack of reflection creates a "shadow" effect in the data and serves as a stark reminder of the impact material properties have on LiDAR's sensing capabilities. In this paper, we define "Shadow Materials" as substances that, like aluminum leisure mats, create "Shadow" on LiDAR point clouds because they cannot be measured by LiDAR.

B. LiDAR-based Object Detection in Autonomous Vehicles

Autonomous driving systems rely heavily on object detection using LiDAR point clouds for environmental awareness. LiDAR, which is effective even in low light, detects objects in 360 degrees, surpassing camera-based detection. Object detection models using point clouds are categorized as either point-based or voxel-based. Point-based models, such as PointRCNN [3], extract features directly from point clouds. Voxel-based models, such as PointPillars [4], segment the cloud into voxels for feature extraction. Due to their faster processing, voxel-based models such as PointPillars and CenterPoint [5] are preferred in autonomous systems. This is evidenced by their use in leading autonomous driving software such as Autoware [6], highlighting their importance in advancing automotive technology.

C. Related Work on the Adversarial Attacks on LiDAR-Based Object Detection

Previous adversarial attacks on LiDAR object detection models have induced misclassifications by placing objects with specific shapes in specific locations [7, 8] or by injecting false point clouds through direct laser interference [1, 2]. In contrast, our research introduces a new technique named Shadow Hack, where placing Shadow Materials in the environment creates artificial shadows in the point cloud data. This attack aims to fools object detection models by exploiting these shadows to cause misidentification of objects.

Shadows in LiDAR point clouds have previously been considered in the context of defending against attacks. Typically,

the area behind an object is not measured by LiDAR due to the lack of returned laser pulses, resulting in a shadow. In attacks where lasers are used to inject false points into the LiDAR data [1][2], an adversarial object is injected, so no shadow appears behind the ghost object. The Shadow-Catcher framework proposed by Hau et al. [9] exploits this shadow property to detect LiDAR spoofing attacks. Our approach, which reduces rather than adds point cloud data, naturally includes shadows, making them undetectable by Shadow-Catcher's detection method.

III. Threat Model and Attack Scenario

A. Threat Model

Figure 2 outlines the attack strategy for a shadow hack. The attacker selects a position along the route where the target autonomous vehicle is expected to travel. The goal is to fool the vehicle's LiDAR object detection system into falsely detecting non-existent objects by strategically placed shadows. The attacker is assumed to have knowledge of the type of LiDAR system used in the target vehicle and the expertise to optimize the placement of shadows in a way that exploits the object detection model. This level of understanding can be gained by analyzing publicly available vehicle specifications or by reverse engineering similar models.

B. Attack Scenario

In this subsection of the paper, we will present two attack scenarios for the Shadow Hack attack.

a. Induced Sudden Stop in Clear Visibility In the first scenario, the attack targets a leading autonomous vehicle in a convoy. Although there is no actual obstacle ahead, the vehicle misperceives an obstacle due to the attacker's strategically placed shadows. This misperception, especially in clear conditions where the vehicle is maintaining high speed, leads to an abrupt stop. Such sudden braking increases the risk of rear-end collisions by following vehicles.

b. False Evasive Action on Multi-Lane Roads The second scenario involves an attack on an autonomous vehicle traveling in the leftmost lane of a multi-lane road. The vehicle is tricked into detecting a phantom obstacle, causing it to initiate an evasive maneuver, typically veering to the right. This sudden lane change can result in a collision with vehicles in the adjacent lane.

Both scenarios are facilitated by the attacker placing Shadow Materials, such as aluminum leisure mats, on the road in advance. Shadow Materials create artificial shadows in the LiDAR point clouds. The stealthy nature of this attack is enhanced by the fact that the attacker doesn't need to be physically present at the accident scene and doesn't need to directly interfere with the target vehicle itself.

IV. Shadow Hack

A. Attack Model

As shown in Figure 4, we establish the parameters of the Shadow Hack attack through a formulation using the variables listed in Table I. To successfully execute the attack, the attacker must perform advanced optimization of several key factors: the shadow's width w, its length l, the distances d_x and d_y along the x- and y-axes, respectively, between the attack target and the shadow, and the shadow's angle θ around the vertical axis. This preparation is essential to ensure that the target autonomous vehicle will be induced to make a false detection at the coordinates (x_t, y_t).

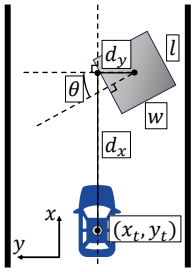

Fig. 4. Overview of variables and parameters of Shadow Hack Attack

TABLE I
Notations of Variables

Symbol	Description
d_x	x-axis distance: target ↔ shadow
d_y	y-axis distance: target ↔ shadow
θ	angle of the shadow
(x_t, y_t)	coordinates of the target
w	shadow width
l	shadow length

B. Shadow Hack Attack Framework

The Shadow Hack attack framework is a methodical process crafted to deceive autonomous vehicles through the physical manipulation of point cloud data. This framework is divided into the following three distinct steps:

Step 1: Acquisition of Point Cloud Data. This initial step involves the collection of data critical to the attack. The attacker identifies a specific location, (x_t, y_t), along the expected route of the target autonomous vehicle. Here, the attacker collects point cloud data, referred to as X_{benign}. This data collection is carefully done using the same LiDAR sensor as the target vehicle to ensure data authenticity and increase the likelihood of a successful attack.

Step 2: Optimization of the Adversarial Shadow. Following data collection, the next phase focuses on the creation of the adversarial shadow. Using the X_{benign} data collected, the attacker simulates and generates an adversarial shadow. This phase is critical because it involves fine-tuning the shadow's location, represented by the coordinates d_x, d_y, and its orientation, represented by the angle θ. The goal is to manipulate the target vehicle's object recognition system into falsely perceiving an object. The details of this optimization process will be described in Section IV-C.

Step 3: Real-World Implementation. The final step is the practical implementation of the Adversarial Shadow. The attacker places Shadow Materials such as an infrared-cut film or an aluminum sheet at the location of the simulated shadow. These materials are chosen for their inability to be detected by LiDAR, rendering them invisible in the point cloud data and effectively creating a "shadow." As the target vehicle traverses the predetermined location (x_t, y_t), it encounters this artificially created shadow, resulting in the false detection of a non-existent object.

C. Optimization of the Adversarial Shadow

This section outlines our approach to optimizing a shadow designed to fool the object recognition models of autonomous

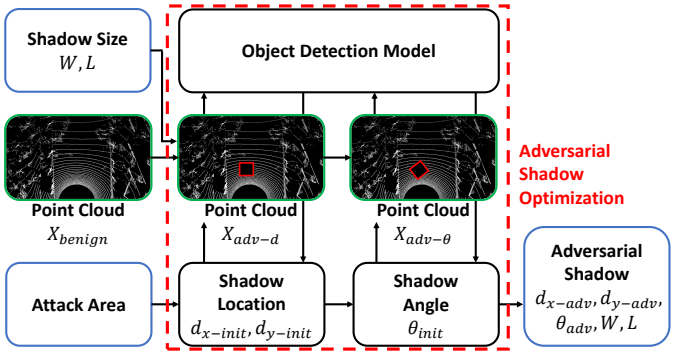

Fig. 5. Workflow of the Adversarial Shadow Optimization

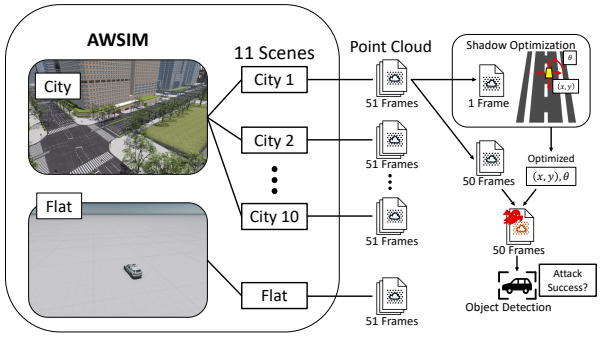

Fig. 6. Experimental Flow

vehicles. Figure 5 illustrates the workflow of optimization process. The goal is to precisely identify the most effective position (d_x, d_y) and angle (θ) for the shadow that would cause the model to erroneously detect a non-existent object; i.e., a false positive. It is important to note that this optimization process must be performed for each unique scene, as the effectiveness of the shadow may vary depending on specific environmental and situational conditions.

Step 1: Determining the Shadow's Location. The first step focuses on finding the optimal location for the shadow. Starting with an initial point d_{x-init}, d_{y-init} in the designated attack area closest to the vehicle, the process involves modifying the benign point cloud data X_{benign}. This modification is achieved by removing ground points to simulate the shadow (a rectangle of dimensions W, L), creating a modified point cloud X_{adv-d}. This new point cloud is analyzed using the object recognition model to evaluate the effect of the shadow. The coordinates d_x, d_y are varied within the attack area to determine the location that induces the highest false detection rate. The location with the highest false detection confidence is selected as the optimal position, d_{x-adv}, d_{y-adv}.

Step 2: Optimizing the Shadow's Angle. After determining the shadow's location, the focus shifts to optimizing its angle. The shadow, which is kept in a $W \times L$ rectangular shape, is rotated at various angles θ_{tmp} around a vertical axis. Each rotation changes the configuration of the point cloud $X_{adv-\theta}$, which is then tested with the object recognition model. The angle θ_{tmp} is varied from $0°$ to $90°$ to find the orientation that most effectively triggers a false detection. The angle that gives the highest confidence of a false positive is finalized as the optimal angle, θ.

V. Evaluation

A. Experimental Setup

In this subsection, we detail the point cloud collection methods and object detection framework used in our experiments (see Figure6).

Point Cloud Collection. For point cloud collection in our experiments, we adopt AWSIM [10], a simulator developed specifically as a simulation environment for Autoware [6], an open source autonomous driving platform. AWSIM accurately emulates real-city LiDAR systems, including measurement noise to produce point clouds with realistic variance. However, AWSIM does not implement intensity measurement, so intensity is not considered in object detection in the experiments in this section. In order to publish LiDAR point cloud data as a ROS2 [11] topic, we developed a custom ROS2 node that saves the topic as a PCD (Point Cloud Data) file, allowing the collection of point cloud data. The LiDAR used is the Ouster OS1-64, positioned at a height of 1.73 m, based on the setup used to collect the KITTI dataset [12]. The point cloud collection was performed in two environments: a featureless map (Flat) to evaluate the attack success rate in a noiseless environment, and urban environments (City 1 - 10) to evaluate the rate amidst typical urban noise such as sidewalks and trees (See Figure 7).

Object Detection. For 3D object detection from point clouds, we use OpenPCDet [13], which provides several pre-trained models and allows object detection from point clouds using any of these models. Point cloud-based object detection can be divided into two approaches: Voxel-based, where point clouds are divided into units called voxels for feature extraction and detection, and Point-based, which relies on features of individual points within the cloud. In this study, we evaluate using the Voxel-based PointPillars and Point-based Point-RCNN models, both of which are prominent and have pre-trained models available in OpenPCDet, trained on the KITTI dataset.

Definition of Attack Success. In the authors' country, the speed limit on public roads is set at 60 km/h for standard vehicles. Consequently, the required stopping distance at this speed is determined to be 44 meters. In addition, the typical width of a lane on a public road is 3.5 m. Therefore, if an object is falsely detected within 44 m in front of an autonomous vehicle and within a width of 3.5 m, it is very likely to trigger emergency braking or steering. Thus, in this paper, we define a successful attack as the appearance of a bounding box indicating a non-existent vehicle within the defined attack success area of 44 m ahead and 3.5 m wide. The bounding box of the misdetected object nearly coincides with the location of the adversarial shadow, as shown in Fig. 4, where a successful attack is defined as a misdetection

TABLE II
SCENE-WISE ATTACK SUCCESS RATES OF SHADOW HACK ON POINTPILLARS

Scene	Success Rate
Flat	1.00
City 1	0.60
City 2	0.44
City 3	0.04
City 4	0.66
City 5	1.00
City 6	0.88
City 7	0.96
City 8	0.04
City 9	0.20
City 10	0.98

TABLE III
SCENE-WISE ATTACK SUCCESS RATES OF SHADOW HACK ON POINTRCNN

Scene	Success Rate
Flat	0.00
City 1	0.00
City 2	0.00
City 3	0.00
City 4	0.00
City 5	0.00
City 6	0.00
City 7	0.00
City 8	0.00
City 9	0.00
City 10	0.00

occurring within $0 < d_x \leq 44, -1.25 \leq d_y \leq 1.25$.

B. Experimental Procedure

The experimental procedure has three steps: 1. Collect the point clouds, 2. Optimize the shadows, and 3. Evaluation of the attack success rate. In step 1, we collect 51 consecutive LiDAR frames in a stationary state using AWSIM, both in an unobstructed environment (Flat) and in urban environments (City 1–10), as shown in Fig. 7. In Step 2, shadow location optimization is performed on the first frame only. The coordinates and angles of the optimized adversarial shadow are used to create shadows by removing point clouds in the remaining 50 frames. In step 3, the success rate of the attack is determined by feeding these 50 frames into the object detection model, calculating the number of frames where the attack was successful, and dividing it by the total number of frames, which is 50.

C. Results

Shadow Hack attack on PointPillars in urban environments achieve an average attack success rate of 58% when the vehicle is stationary. Table II shows the scene-wise attack success rates for Shadow Hack on PointPillars, while Table III presents the attack success rates for PointRCNN. Figure 8 displays the object detection results for PointPillars before and after attacks for both Flat and City 1-10 scene point clouds, with the detected car positions marked by green bounding boxes. Notably, the attack success rate for PointPillars in the Flat scene reaches 1.00, indicating the effectiveness of generated Adversarial Shadows across all frames. In contrast, in City scenes, the average attack success rates in the ten scenes are 0.58, demonstrating that attacks are successful in more than half of the frames. These results indicate the stability of attacks on PointPillars when the vehicle is stationary. On the other hand, PointRCNN exhibits a 0.00 attack success rate for both Flat and City 1-10 point clouds, suggesting the ineffectiveness of attacks. The difference in attack success rates between PointPillars and PointRCNN can be attributed to differences in their object detection methods, as discussed in detail in Section VI-A.

VI. DISCUSSION OF RESULTS AND FUTURE WORK

A. Discussion of Results

The results in Section V-C show that differences between the attack target model and the surrounding environment affect the attack success rate. We believe the variations in attack success rates across target models are due to differences in their processing. The attack shadows have a significant impact on the object detection results in methods that perform inference on point clouds containing shadows. On the other hand, attack shadows do not affect the results when methods that remove shadows prior to object detection are implemented in the object detection model, resulting in a low attack success rate. PointRCNN, with a low attack success rate of 0%, performs object detection in its processing pipeline after removing the ground point cloud. PointPillars, achieving an attack success rate of 58%, performs object detection on point clouds containing shadows, without ground removal in the processing pipeline.

The findings in Section V-C indicate that Shadow Hack is more successful in open environments with fewer obstacles surrounding the vehicle, with the attack success rate for PointPillars being 100% in flat environments, 60% in world 1, and 44% in world 2. This paper has fixed the shadow's shape while optimizing its position and angle. By advancing our optimization methods, such as adapting the shadow's shape based on the loss metrics of machine learning models, we aim to enhance attack success rates in more complex environments.

B. Future Work

In this paper, we evaluate the feasibility of the Shadow Hack attack, which exploits the influence of shadows on object detection in LiDAR point clouds.

Attack Capabilities on Moving Autonomous Vehicles. The observed occurrence of false positives in the object detection model as a result of the attack implies the effectiveness of attacks on autonomous driving systems. However, in practice, autonomous vehicles perform object detection while in motion. The experiments conducted in this paper were limited to assessing attacks when LiDAR was stationary. When LiDAR is in motion, the shape of shadows cast by the Shadow Materials fixed to the ground also changes. To successfully achieve attacks on moving targets, it becomes necessary to consider the changes in shadow shapes when generating shadows. Therefore, the development of the Shadow Hack attack method for autonomous vehicles, specifically considering the variations in measurement points, and the evaluation of attack success rates under such conditions, remain future research challenges.

Impacts on Autonomous Driving Systems. In this paper, we focused on assessing the robustness of standalone LiDAR object detection models against attacks employed in autonomous vehicles. The results presented in Section V demonstrate that the Shadow Hack attack induces false positive detections in more than half of the frames in LiDAR object detection models. However, the actual impact of these false positives on autonomous driving systems has not been

Fig. 7. LiDAR point cloud measurement in AWSIM simulation for each scene. The red points represent the point cloud measured by LiDAR.

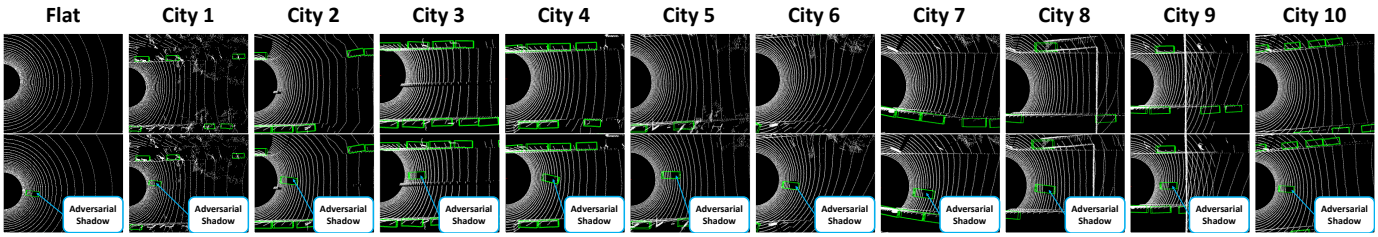

Fig. 8. Object detection results by PointPillars in 11 scenes. The top images show the detection results before the attack, while the bottom images show the results after the attack. In the after attack images, the Adversarial Shadow is erroneously detected as an object.

evaluated. Therefore, it is imperative to assess the behavior of autonomous vehicles when subjected to Shadow Hack attacks in both simulation and real-world scenarios, providing an end-to-end evaluation of the system.

In the simulation, instead of acquiring and subsequently removing point clouds as done in this paper, we implemented materials in AWSIM that do not appear as point clouds to replicate the Shadow Hack attack. We then operated an autonomous vehicle equipped with Autoware in AWSIM. The evaluation focused on whether the AWSIM autonomous vehicle, when subjected to the Shadow Hack attack, would come to a stop in response to false positive detections of objects in its path or continue driving.

In the real world, we also recreated the attack using Shadow Materials such as infrared-cut film on autonomous vehicles equipped with Autoware, mirroring the approach taken in AWSIM. The assessment aimed to determine whether the autonomous vehicles would engage in control actions such as stopping or evading in response to the attack.

Extension of the Shadow Optimization Process. In this paper, the Shadow Hack attack's shadow generation method optimizes the shadow's position through exhaustive search while fixing the angle. However, there is potential for optimization expansion. For instance, angle optimization can be performed for top-ranked positions with high confidence in false positive detections during the position search, ultimately selecting the position and angle with the highest confidence for the false positive detection. Additionally, varying the shadow's shape and quantity could further enhance the attack success rate. Therefore, we plan to conduct future comparative evaluations of different shadow optimization methods for the Shadow Hack attack.

Countermeasure. As a countermeasure to the Shadow Hack attack, there are three approaches:

The Multi-Sensor Fusion Object Detection Model utilizes data from both LiDAR and cameras for object detection. This approach helps mitigate the risk of False Positives induced by the Shadow Hack attack, as models like Frustum PointNets [14] first identify potential object regions using images before conducting point cloud-based object detection.

Point Cloud Missing Data Detection and Automated Recovery Mechanism is a mechanism to detect and fill missing points, known as "shadows," as a preprocessing step for point cloud object detection models. This approach aims to nullify the effects of the Shadow Hack attack by identifying and compensating for anomalies, such as Adversarial Shadows, which create abnormal ground conditions not encountered in regular measurements.

Object Detection Model with Tracking Integration utilizes tracking instead of per-frame detection. While our evaluation in this paper focused on attacks inducing false object detections at a single distance, we did not assess the robustness of distances between shadows and vehicles. Assuming shadows are distance-sensitive, a tracking-based model may reduce false detections since shadows might only be falsely detected at specific distances. Further investigation is needed to evaluate Tracking Object Detection as a countermeasure.

ACKNOWLEDGMENT

A part of this work was supported by JSPS KAKENHI 22S0604 and JST CREST JPMJCR23M4.

REFERENCES

[1] Y. Cao et al. "Adversarial Sensor Attack on Lidar-based Perception in Autonomous Driving". In: *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. 2019, pp. 2267–2281.

[2] Y. Cao et al. "You Can't See Me: Physical Removal Attacks on LiDAR-based Autonomous Vehicles Driving Frameworks". In: *32nd USENIX Security Symposium*. 2023, pp. 2993–3010.

[3] S. Shi and H. Li X. Wang. "Pointrcnn: 3d Object Proposal Generation and Detection from Point Cloud". In: *Proceedings of the IEEE/CVF conference on CVPR*. 2019, pp. 770–779.

[4] A. Lang et al. "PointPillars: Fast Encoders for Object Detection from Point Clouds". In: *Proceedings of the IEEE/CVF conference on CVPR*. 2019, pp. 12697–12705.

[5] T. Yin and P. Krahenbuhl X. Zhou. "Center-based 3d Object Detection and Tracking". In: *Proceedings of the IEEE/CVF conference on CVPR*. 2021, pp. 11784–11793.

[6] S. Kato et al. "Autoware on Board: Enabling Autonomous Vehicles with Embedded Systems". In: *2018 ACM/IEEE 9th ICCPS*. 2018, pp. 287–296.

[7] J. Tu et al. “Physically Realizable Adversarial Examples for Lidar Object Detection”. In: *Proceedings of the IEEE/CVF Conference on CVPR*. 2020, pp. 13716–13725.

[8] Y. Cao et al. “Invisible for both Camera and Lidar: Security of Multi-Sensor Fusion based Perception in Autonomous Driving Under Physical-World Attacks”. In: *2021 IEEE Symposium on Security and Privacy*. 2021, pp. 176–194.

[9] Z. Hau, L. Muñoz-González S. Demetriou, and EC. Lupu. “Shadow-catcher: Looking into Shadows to Detect Ghost Objects in Autonomous Vehicle 3d Sensing”. In: *Computer Security–ESORICS 2021*. 2021, pp. 691–711.

[10] tier4. *AWSIM - Open Source Simulator for Self-Driving Vehicles*. https://github.com/tier4/AWSIM. 2022.

[11] S. Macenski, B. Gerkey T. Foote, and W. Woodall C. Lalancette. “Robot Operating System 2: Design, Architecture, and Uses in The Wild”. In: *Science Robotics* 7.66 (2022), eabm6074.

[12] A. Geiger, C. Stiller P. Lenz, and R. Urtasun. “Vision Meets Robotics: The Kitti Dataset”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1231–1237.

[13] OpenPCDet Development Team. *OpenPCDet: An Open-source Toolbox for 3D Object Detection from Point Clouds*. https://github.com/open-mmlab/OpenPCDet. 2020.

[14] CR. Qi, C. Wu W. Liu, and LJ. Guibas H. Su. “Frustum Pointnets for 3d Object Detection from RGB-D Data”. In: *Proceedings of the IEEE conference on CVPR*. 2018, pp. 918–927.