GhostBuster: Looking Into Shadows to Detect Ghost Objects in Autonomous Vehicle 3D Sensing

Zhongyuan Hau  
Department of Computing  
Imperial College London  
zy.hau17@imperial.ac.uk

Soteris Demetriou  
Department of Computing  
Imperial College London  
s.demetriou@imperial.ac.uk

Luis Muñoz-González  
Department of Computing  
Imperial College London  
l.munoz@imperial.ac.uk

Emil C. Lupu  
Department of Computing  
Imperial College London  
e.c.lupu@imperial.ac.uk

Abstract—LiDAR-driven 3D sensing allows new generations of vehicles to achieve advanced levels of situation awareness. However, recent works have demonstrated that physical adversaries can spoof LiDAR return signals and deceive 3D object detectors to erroneously detect “ghost” objects. In this work, we introduce GhostBuster, a set of new techniques embodied in an end-to-end prototype to detect ghost object attacks on 3D detectors. GhostBuster is agnostic of the 3D detector targeted, and only uses LiDAR data that is already available to the target object detector. It considers the 3D object detectors’ blind spots by examining the objects’ 3D shadows. Ray optics is used to estimate the shadow regions and an exponential decay approach minimizes the importance of noisy points. GhostBuster identifies anomalous regions and then analyzes their 3D point cluster densities to distinguish between shadows of ghost objects, and genuine object shadows. We conduct an extensive empirical evaluation on the KITTI dataset and find that GhostBuster consistently achieves more than 94% accuracy in identifying anomalous shadows, which it can attribute with 96% accuracy to ghost attacks. We introduce a new class of “invalidation” attacks where adversaries can target shadows of genuine objects aiming to invalidate them and we show that GhostBuster remains robust to these attacks. Finally we show that GhostBuster can achieve real-time detection, requiring only between 0.003s–0.021s on average to process an object in a 3D point cloud on a commodity machine.

Index Terms—Autonomous Vehicles, LiDAR, 3D Object Detection, Automotive Security

I. INTRODUCTION

High-precision depth sensors are increasingly being used for mapping the environment in a variety of application domains, such as robotics [1], security surveillance [2], augmented reality applications [3], to cite some. LiDARs (derived from light detection and ranging) are popular such depth sensors. They have found widespread deployment [4], [5] in autonomous vehicles (referred to as AVs hereforth) where a new class of Deep Neural Network (DNN) 3D object detectors leverage depth sensor measurements (processed in batches called 3D point clouds) to detect objects—a necessary task for downstream safety-critical driving decision-making [6]–[9].

Recent studies have shown that it is possible to attack LiDAR-based perception systems of AVs by spoofing LiDAR return signals [10]–[12]. In [10], [11], the authors demonstrated that a physical adversary can relay laser pulses resulting in an object being perceived closer to the LiDAR unit than it actually is; and can opportunistically spoof reflected laser pulses to create fake objects. Cao et. al. conducted in [12] a security study of LiDAR spoofing attacks on AVs and formulated an optimization based attack for input perturbations that successfully spoofed objects and affected downstream AV driving decision-making.

Prior studies in defending against such physical attacks on AV sensors, have focused primarily on RGB-based classifiers [13] which leverage inputs from RGB cameras disregarding the effects of LiDAR-poisoning attacks. Others, have proposed fusing data from co-located heterogeneous modalities, especially visual (RGB) and depth (D) data, to improve object-detection systems [14]–[19] and vehicle positioning [20], [21]. However these focus on improving the accuracy of 3D object prediction tasks in challenging, albeit benign scenarios. Cao et al. [12] hinted to using information fusion to facilitate detection of active physical attacks. However, as they also recognize, such approaches [22], [23] need to assume that the majority of the sensors are not under active attack. Recently, Sun et. al. in [24] proposed a defense approach to detect spoofed objects by using occlusion patterns to verify whether the point cloud of a given object is valid.

In this work we focus on detecting active data injection attacks on LiDAR sensors assuming neither the presence nor the cooperation of other sensors, homogeneous or heterogeneous. Moreover, our approach is applied on the output of 3D object classifiers, effectively making it agnostic to the classification model targeted: any object detected by the classifier, either genuine or fake (ghost), will be subjected to verification. Point-cloud based 3D detection models [6]–[9] learn and use point representations to identify objects. Our approach leverages the blind spot of these 3D object detection models to facilitate attack detection. When objects are observed from a source of
light, these objects are illuminated, whereas anything behind them appears to be in shadow (this is illustrated in Fig. 1 visualizing the measurements captured by a LiDAR sensor in a real driving scenario). Since LiDARs operate based on the same physical principles by projecting light pulses, we leverage that observation and design an efficient and highly effective detection mechanism which verifies the presence of 3D objects only when they exhibit the expected shadow effect.

Detecting shadows and attributing them to objects, solely based on 3D point clouds, is not trivial. Determining shadows of objects is well studied in RGB measurements [25]–[28] but not in 3D point clouds. Moreover, even if we were able to accurately determine shadow regions, physical effects such as the light diffraction and beam divergence, co-located and occluding objects in tandem with inaccuracies in measurements or sensor calibration, create measurement artifacts within shadow regions, which makes it hard to determine whether a shadow region actually corresponds to a true shadow of a genuine object or the absence of a true shadow for a ghost object. In addition, to be able to use shadows as part of an attack detection mechanism, we also have to analyze the adversary’s capability to bypass this detection by considering a new type of object invalidation attack, where the adversary’s goal shifts from injecting ghost objects to poisoning shadows of true objects making the distinction between poisoned and true shadows even harder. To address these challenges, we design GhostBuster which uses a two-tier approach to efficiently and effectively verify 3D objects.

Firstly, GhostBuster exploits the fact that LiDAR measurements are obtained based on emitted laser rays and employs optical geometry techniques (or ray optics) to map the expected 3D shadow region of a detected object. Then it uses a 3D point weighting and aggregated scoring algorithm leveraging exponential decay weight estimation to downplay the importance of measurement artifacts and determine whether the proposed shadow region corresponds to (a) a real shadow or (b) an anomalous shadow. In the latter case, it further classifies a shadow region as poisoned (thus verifying the presence of a true object) or ghost object shadow, using a binary classifier trained on density features extracted from the proposed shadow region. Our extensive evaluation shows that more than 98% of the 3D objects in our dataset have meaningful shadows, and that GhostBuster’s shadow region estimation closely captures their true shape. We also show that GhostBuster consistently achieves more than 94% accuracy in identifying anomalous shadows. GhostBuster can further classify with 96% accuracy whether the anomalous shadow corresponds to a ghost attack. In addition we design a novel strong adversary which follows an optimal strategy to launch an evasion attack aiming to poison a genuine shadow such that it gets misclassified as a ghost shadow (and essentially invalidate genuine object). We demonstrate that GhostBuster shadow classification remains robust against such state of the art adversaries. Lastly, GhostBuster achieves real-time detection rendering it suitable for deployment both offline for forensic analysis and online for providing hints to vehicle passengers, operators or end-to-end AI systems. Demos of GhostBuster can be found on our project’s website [29].

**Contributions.** Below we summarize our main contributions.

- We perform an in-depth study of the phenomena of 3D shadows for more than 7 object types in real-world AV scenarios and show that genuine objects have shadows which can be used as a physical invariant to detect 3D object deception attacks.
- We propose a set of new techniques embodied in an end-to-end prototype (GhostBuster) for detecting attacks on LiDAR-based 3D object detectors where attackers induce fake depth sensor measurements to spoof objects that are not physically in the environment. Our approach does not require the presence (or integrity) of measurements from sensors of other modalities (e.g. RGB cameras) and only makes use of existing depth measurements. Our approach is also agnostic and orthogonal to the 3D object classification model used.
- We evaluate the effectiveness and efficiency of GhostBuster using real-world 3D scenes and found that it can achieve real-time detection of ghost injection attacks.
- We further design a strong adversary capable of launching a novel class of object invalidation attacks specifically targeting the proposed method. We evaluate GhostBuster against this adversary and found that our system remains robust.

**Paper Organization.** In Section II we provide necessary background information and elaborate on the threat model considered in our work. In Section III we analyze a real dataset to verify the existence of 3D shadows for genuine objects, and identify key challenges. In Section IV we present the design of GhostBuster and in Section V perform an in-depth evaluation of GhostBuster’s effectiveness, robustness and runtime performance. In Section VI we discuss how GhostBuster can be used in practice and areas for future work. In Section VII we identify and discuss relevant prior work and conclude the paper in Section VIII.

## II. Background and Threat Model

### A. Background

**LiDAR sensors.** To scan the environment, LiDARs emit a pulse in the invisible near-infrared wavelength (900–1100 nm), which is reflected on incident objects before returning to the receiver of the emitter device. Based on the time of flight, LiDARs calculate the distance between the sensor and the incident object. LiDARs used in AVs (e.g. Velodyne LiDARs) emit a number of light pulses from an array of vertically arranged lasers (16, 32, 64, etc.) that rotate around a center axis to obtain a 360-view of the surroundings of the sensor unit. The sensor translates a return signal to a measurement 3D point consisting of coordinates (x,y,z) and a reflection value (R) corresponding to the return signal’s reflectivity or signal strength. 3D point clouds are commonly projected to 2D in a

1 To respect the double-blind review process, GhostBuster’s project website [29] is currently anonymized and does not track visitors.
more compact representation called birds-eye view or BEV for short. Fig. 1 illustrates the BEV representation of a 3D point cloud captured in a real driving scene.

**Attacks on LiDARs.** With the widespread adoption of LiDAR systems in AVs facilitating perception, there is a growing interest in the security of such systems. In particular, there have been works studying active injection of laser pulses (signals) to perturb the sensed environment. Petit et. al. in [10] first introduced physical attacks on LiDAR systems with the goal to generate noise, fake echoes and fake objects. They successfully performed attacks that relay original signals from another location to fake the distance of a real object. The relay attack was extended to injection of fake objects by replaying signals. However, they were unable to spoof objects closer than 20m from the LiDAR receiver. In [11], Shin et. al. improved on the previous work and managed to spoof objects that are up to 12m in front of the LiDAR receiver, with a limitation of a maximum of 10 points injected. More recently, Cao et. al in [12] demonstrated the capability to spoof up to 100 points and proposed an attack methodology that uses optimization to to generate adversarial points that can successfully fool object detection models.

**B. Threat Model**

**Capabilities of the adversary.** We consider an adversary who can inject spoofed LiDAR return signals in a target vehicle’s LiDAR sensor unit. The adversary aims to deceive the 3D object detectors, that use the victim LiDAR’s measurements (3D point clouds), into detecting ghost objects and negatively affect their environment perception capabilities. We assume the adversary has state-of-the-art capabilities and can inject up to 100 points in each 3D point cloud [12]. Even though prior work only showed that such an adversary can successfully introduce fake or ghost 3D objects in the Apollo’s 3D object detection model, we found that other 3D object detectors can also be trivially deceived (see Ghost attacks below). Thus we take a proactive stance and consider a stronger adversary which can introduce ghost objects irrespective of the 3D object detector used on the target AV. In other words, we do not deal with how the adversary managed to inject a ghost object on a target 3D object classifier, but instead focus on verifying whether the identified object is genuine or not.

**Ghost attacks.** In our experiments we simulate ghost attacks on a popular 3D object detector, Point-GNN [8], which currently ranks within the top 10 on the KITTI evaluation benchmark. To create ghost objects, we use a “copy-and-paste” method, where the point clouds of genuine objects are extracted from real-world point clouds (from the commonly used KITTI dataset [30]) and used as the full attack traces. To demonstrate the chosen detector’s susceptibility to realistic ghost attacks we down-sampled each full attack trace to generate three smaller traces corresponding to a 10-point adversary, a 60-point adversary and a stronger 100-points adversary. Each trace was individually merged in 200 random scenes from the KITTI dataset, resulting in 600 poisoned scenes. Table I summarizes the success rate of the adversary in getting a ghost object detected.

While [12] focused on minimizing the number of injected points to successfully spoof objects, our proposed defence is agnostic to how the adversary spoofs an object; our primary focus lies in detecting successfully spoofed objects. Therefore for our experiments we used the full attack traces instead of the down-sampled ones to maximize a successful targeted attack in each trial. This is not problematic for our shadow region estimation (see Section IV-B) since any successful injection would result in a bounding box size that is representative of the object spoofed.

| Objects | Car | Pedestrian | Cyclist |
|---------|-----|------------|--------|
| 10 points | 0.119 | 0.00 | 0.00 |
| 60 points | 0.188 | 0.985 | 0.985 |
| 100 points | 0.698 | 0.990 | 0.990 |
| Full Attack Trace | 0.941 | 0.990 | 0.990 |

**Object invalidation attacks.** An approach that uses shadows to verify true objects, might incentivize attacks where the adversary’s goal changes from injecting ghost objects to invalidating genuine objects. Hence, an adversary might try to inject fake points inside the shadow region of a target object to force our object detection validation algorithm to invalidate real objects. This could lead to a wrong safety-critical decision being made which can potentially have dire consequences. Our system recognizes this and thus considers a stronger adversary which is capable of launching both ghost object injection and genuine object invalidation attacks. To test the robustness of our system against object invalidation attacks, we design a new strong attack with full knowledge of the detection mechanism, and evaluate the success of this attack against our system (see Section V-C).

**III. SHADOW EFFECTS OF 3D OBJECTS**

In this work, we observe that true 3D object representations in a point cloud are closely followed by regions void of measurements. We call this the 3D shadow effect. 3D shadow effects manifest from how LiDAR sensors record measurements (3D points) of return light pulses reflected off an object in a direct line of sight that returns within a constrained time period to the receiver of the sensor unit. Thus, anything behind the incident object cannot be reached by the light rays and cannot be measured, resulting in the void shadow regions as depicted in Fig. 1. This observation leads us to hypothesize that the presence of shadows is a physical invariant that can be used to verify genuine 3D objects. In this Section we systematically analyze real 3D driving scenes to verify the presence of shadows in 3D objects and obtain ground truth for such shadow regions.

**Methodology.** We randomly sampled 120 scenes from the KITTI dataset [30]. The dataset includes LiDAR measurements (point cloud scenes) from real driving scenarios in Karlsruhe, Germany. The dataset is accompanied by a set of
object labels for training 3D object detectors. We used these labels to locate true objects in each scene. We then converted each scene to its birds-eye-view (BEV) representation by projecting each 3D point to a 2D plane. Subsequently we went through all 120 scenes and (1) manually annotated shadow regions, if present, using the VIA annotation tool [31], and (2) assigned shadow regions to objects.

**Object and shadow co-occurrence results.** In the 120 sampled scenes, we found a total of 607 objects, where the breakdown by object categories is detailed in Table II. All objects are located in the frontal view of the vehicle and comprise of objects both on the road and on sidewalks. Out of the 607 objects, we have identified shadows for 597 or 98.3% of the objects, the details by object type can be found in Table II. The reason for not being able to identify the shadows for the remaining 10/607 (1.6%) objects, was the location of objects in the environment. For example, if one object is directly in front of another object, the first object cannot be unequivocally assigned a shadow region which might be observed because of the second object. This can occur for example, when a person is standing in front of a vehicle where the objects effectively contribute together to the same shadow effect. Next, we list a number of challenges we identified for shadow region estimation.

| Objects       | Count In Dataset | % of total Objects | Labelled Shadows | % of Object Type |
|---------------|------------------|--------------------|------------------|-----------------|
| Car           | 444              | 75.1               | 439              | 98.9            |
| Pedestrian    | 45               | 7.4                | 41               | 91.1            |
| Cyclist       | 17               | 2.8                | 17               | 100             |
| Van           | 56               | 9.2                | 55               | 98.2            |
| Truck         | 17               | 2.8                | 17               | 100             |
| Tram          | 6                | 1.0                | 6                | 100             |
| Sitting Person| 1                | 0.2                | 1                | 100             |
| Miscellaneous | 21               | 3.5                | 21               | 100             |
| Total         | 607              | N.A.               | 597              | N.A.            |

**Challenges in automatically detecting object shadows.** Labelling of shadow regions on the BEV and uniquely assigning them to objects in a scene can be challenging in some cases, even in benign scenarios, due to ambiguities in 3D points and void regions. These challenges can lead to labelling inaccuracies. We split these cases in two main categories based on the source of their inaccuracies.

**Physical effects.** Physical effects such as beam divergence can create noise artifacts in shadow regions. These result in 3D points being registered inside what is expected to be a shadow region of an object. We observe that such points appear mostly closer to the boundaries of regions otherwise void of measurements. Beam divergence is the widening of laser beam diameter and occurs when the laser pulse propagates away from the LiDAR. As the beam widens, the cross-section pulse energy is spread over a larger area, lowering the energy of back-scattered returning beams, leading to lowered signal-to-noise ratio and decreased precision in measurements [32].

**Shape of 3D objects.** Object detectors, cannot precisely capture the shape of an object. The output of object detection is usually the bounding box (BB) within which the detected object lies within. This renders estimation of the precise shadow region behind BBs challenging.

**Location of objects in the environment.** We observed four cases where the placement of objects relative to each other render shadow region labeling and association with objects challenging:

- An object is located in the shadow of another object, thus the shadow of the object is indistinguishable from the shadow that the object is within.
- Objects are clustered together (e.g. cars parked together along the side-walk) resulting in a large region of void from the overlapping shadows.
- An object is in front of a higher object (e.g. car in front of wall) and hence its shadow overlaps with the shadow of the high object.
- Objects are too far away from the LiDAR unit, where the resolution of LiDAR is poorer. It becomes difficult to obtain the shape of the shadow due to sparsity in measurements.

These cases manifest in benign scenarios (examples are illustrated in Fig. [15] in Appendix A-A). Things get even worse when shadows are poisoned (object invalidation attack).

**Conclusion.** By manually labeling shadow regions for objects, we found strong evidence of co-occurrence of objects and shadow regions. This supports our hypothesis that the presence of shadows is a physical invariant that can be used to verify genuine objects in 3D scenes. On the other hand we identified a number of challenging cases which indicate that designing an effective detection system is by no means trivial.

**IV. GhostBuster Design**

In this section, we introduce GhostBuster, a system designed to detect spoofing attacks poisoning 3D point cloud LiDAR measurements.

**A. GhostBuster’s high level architecture.**

GhostBuster’s overall architecture is summarized in Fig. [2]. GhostBuster takes as an input, the output of a 3D object detector (bounding boxes of detected objects in 3D scene’s point cloud) and the original point cloud of the scene. GhostBuster performs a three-phase analysis to determine whether the detected objects are genuine or ghosts. GhostBuster can further distinguish between ghost objects and genuine objects whose shadow regions are being poisoned.

GhostBuster’s analysis proceeds as follows. In Phase 1, GhostBuster employs a *shadow region proposal* algorithm which uses geometrical optics (or ray optics) for a monochromatic light source (beam) to generate proposed shadow regions for each of the 3D objects detected by the 3D object detector. The use of ray optics is appropriate since LiDARs take measurements based on emitted light pulses. By tracing rays from the reference point of the LiDAR unit in point...
clouds, GhostBuster can determine the boundaries of shadow regions for 3D objects. However, as discussed in Section II, the shadow region proposed can be imprecise and can also encompass a number of 3D point artifacts which in principle should not be present, as the light rays responsible for taking that measurement should have already reflected on the incident surface of the target 3D object. To deal with these imprecisions, in Phase-2, GhostBuster’s genuine shadow verification component, performs a point-wise analysis in each shadow region to determine whether the region is indicative of a genuine shadow. For this, it uses a novel 3D-point scoring mechanism, detailed below. If the genuine shadow verification fails, which would mean the system is either under a ghost object injection attack or exposed to a genuine object invalidation attack, GhostBuster uses an adversarial shadow classification model to determine whether the shadow region of the detected object is indicative of a ghost object’s shadow (thereby detecting a ghost attack) or a genuine object’s shadow (thereby detecting an invalidation attack). Fig. 2 illustrates the decision workflow of GhostBuster. Below we elaborate on GhostBuster’s three main components: (a) shadow region proposal; (b) genuine shadow verification; and (c) adversarial shadow classification.

B. Shadow Region Proposal

Here, we introduce a shadow region proposal approach based on geometrical optics (or ray optics) for a monochromatic light source (beam) to obtain shadows from the bounding boxes of objects identified from 3D object detectors. We first elaborate on a fast 2D shadow region estimation, discuss its limitations, and then, propose an alternative 3D shadow estimation. The effectiveness of the two approaches is evaluated in Section V.

2D Shadow Regions. Intuitively, if a scene is converted into its 2D representation, then, using ray optics, we can obtain an area (2D shadow region) behind an object within which rays during data acquisition cannot reach since they would have already reflected on the incident surface of the object. Fig. 3 illustrates this concept. To compute the shadow region, we first convert a 3D point cloud scene into its 2D bird’s-eye-view (BEV) compact representation. Next, we compute the boundary lines of the shadow region. We first take the coordinates of the bounding box for the detected object from the 3D object detector. Using the coordinates of the corners of the bounding box, we compute the gradients of the lines from the reference point (position of the LiDAR unit on the vehicle) to each of the corners. There will be 4 coordinates (from the 4 anchoring corners of a 3D bounding box on the ground) in the x-y plane each with coordinates \((x_i, y_i), \forall i = 1, \ldots, 4\). The gradients \(m_i\) of lines connecting the reference point \((0,0)\) to corner coordinates can be computed with:

\[
m_i = \frac{y_i - 0}{x_i - 0}
\]

(1)

The minimum and maximum gradient lines define the shadow boundary lines for the shadow region of the object.

To simulate the fact that LiDAR has a finite range we define a maximum length for shadow regions, which we call the shadow length \(l\). Intuitively, the shadow length depends on how close the object is to the ray source as well as the height of the object.

Fig. 3. 2D shadow region estimation using lines of maximum and minimum gradients extending from reference point to bounding box coordinates.

Fig. 4. Shadow length of an object (shown on a Z-Y planar view). From Fig. 4 the shadow length \(l\) can be derived from the height of the object \(h\), with respect to the height of the

Fig. 2. 3D scene perception pipeline with GhostBuster integrated.
LiDAR unit \((H)\) and furthest distance of object from LiDAR unit \((d_{obj})\). Using similar triangles,

\[
h = \frac{H}{l + d_{obj}}
\]

(2)

Shadow length \((l)\) can be derived:

\[
l = d_{obj} \times \frac{h}{H - h}
\]

(3)

The shadow boundary lines and the shadow length determine the full shadow region of the object. In Appendix A-B, we provide the algorithm (Alg. 1) to compute 2D shadow regions for objects in a 3D point cloud scene. Fig. 5 illustrates the 2D shadow region which is obtained using this approach in a real point cloud scene. We systematically compare the computed shadow regions against manually labeled shadows and concluded that the computed regions matches visually identified void regions (see Section V-A).

**C. Genuine Shadow Verification**

After the shadow regions are identified, GhostBuster performs an analysis inside each region to determine whether the shadow is genuine or not. As mentioned previously, in principle there should be no measurements inside shadow regions, since light rays cannot reach that part of the environment. However, inaccuracies of the shadow estimation, and noisy artifacts due to physical effects and the placement and shape of objects can result in points being recorded inside genuine shadow regions. Thus, a trivial approach which expects those regions to be completely empty would result in a high numbers of errors, essentially flagging real objects as ghosts frequently.

To mitigate this we propose a method which intuitively reduces the significance of noisy measurements inside shadow regions while highlighting the significance of suspicious measurements. We observe that, for most genuine objects, it is unlikely for point measurements to be recorded directly behind the object (i.e. near the start of the shadow region) and close to the center of the shadow region. Most points found in genuine shadow regions are artifacts manifesting due to beam divergence and the fact that while bounding boxes are rectangular (in 2D) or cubic (in 3D) convex polygons, in reality the objects are not. These artifacts tend to appear closer to the side boundaries of the expected shadow region of an object (Fig. 6).

In other words, we expect points to be absent near the beginning and along the center-line of genuine shadow regions. If points are found in these regions, this would most likely indicate that the shadow is not a genuine shadow. For example, ghost objects cannot “create” shadow regions. To introduce a shadow region, the adversary would need to either remove points or hijack an existing shadow. For the former, there is currently no known mechanism to selectively remove LiDAR point measurements. For the latter, there is no benefit for the adversary to introduce a ghost object in the shadow of an existing object since perceptually the ghost will be further away than the real object and thus won’t be able to affect an imminent driving decision.

Our genuine shadow verification method takes the above into account and classifies a shadow as genuine if it’s anomaly score it’s below a certain threshold. It first assigns a weight to each point inside the shadow region. Intuitively, points due to noise are assigned a lower weight, but if points are found in non expected regions (i.e. along the center-line or close to the start-line) these are assigned higher weights.
Specifically, we use a pair of exponential decay equations (Eq. 4 and 5) on two axis of analysis to assign weights to the points (Fig. 7), where $x_{\text{start}}$, $x_{\text{end}}$, $x_{\text{mid}}$ and $x_{\text{bound}}$ are the distances of the point from the start-line, end-line, center-line and closest boundary line of the shadow region and $\alpha$ is a parameter that tunes the rate of exponential decay. The aggregate anomaly score of the shadow region is computed using Eq. (7), where $w_{\text{min}}$ is the minimum weight a point can obtain in any axis of analysis (i.e. point at boundary line) and $T$ is the total number of points in shadow.

$$w_{\text{start}} = \exp \left( \frac{\ln(0.5)}{\alpha} \times \frac{x_{\text{start}}}{x_{\text{start}} + x_{\text{end}}} \right)$$  \hspace{1cm} (4)

$$w_{\text{mid}} = \exp \left( \frac{\ln(0.5)}{\alpha} \times \frac{x_{\text{mid}}}{x_{\text{mid}} + x_{\text{bound}}} \right)$$  \hspace{1cm} (5)

$$w_{\text{min}} = \exp \left( \frac{\ln(0.5)}{\alpha} \right)$$  \hspace{1cm} (6)

$$\text{score} = \frac{\sum_{i=1}^{T} (w_{\text{start}} \times x_{\text{start}} + w_{\text{mid}} \times x_{\text{mid}}) - (T \times w_{\text{min}}^2)}{T \times (1 - w_{\text{min}}^2)}$$  \hspace{1cm} (7)

The anomaly score threshold is set empirically. We perform an extensive analysis and use the Receiver Operating Characteristic (ROC) curve to determine the threshold that produces the True Positive Rate and False Positive that is acceptable (see Section V-B). An object is verified as genuine by GhostBuster if its shadow region gets a lower score than the anomaly threshold, otherwise the shadow is flagged as anomalous. At this point GhostBuster can already detect that the system is under a LiDAR poisoning attack. Nonetheless, we take this a step further and try to also identify the type of attack the system is subjected to.

D. Adversarial Shadow Classification

A high shadow anomaly score can indicate either a ghost attack or an object invalidation attack. Being able to identify which attack the LiDAR sensor is subject to, has its own merit and it’s an important aspect of GhostBuster. More importantly, potential post-detection actions leveraging GhostBuster’s output can be very different as we would like to ignore ghost objects but we cannot ignore true objects subjected to an invalidation attack. Here we elaborate on how GhostBuster distinguishes between the two.

We observe that during Ghost Attacks, the shadow regions of ghost objects exhibit a high density of points as a result of LiDAR pulses being reflected off the ground in front of the vehicle. In contrast, points are sparse in the shadow regions of true objects during an invalidation attack (Illustrated in Fig. 17 of Appendix A-D). Therefore, we expect the distribution of points within the shadow regions of ghost vs invalidated objects to be distinguishable. Leveraging these observations we use a clustering approach to extract density features from shadow regions which we then use to train a binary adversarial shadow classifier. GhostBuster uses this classifier to determine whether an anomalous shadow is the result of a ghost attack or an invalidation attack.

**Feature Extraction.** In order to characterize the density of the measurements in a shadow region, we cluster together points that are in spatial proximity. To do this we use a clustering algorithm “Density-Based Spatial Clustering of Applications with Noise” (DBSCAN) [33]. The advantage of using DBSCAN over other clustering algorithms is that it is able to identify points that are clustered in arbitrary shapes without the need to pre-specify the number of clusters in the region. This suits our use-case, as point clusters in 3D point clouds are irregular, unlike the circular shapes which most clustering algorithms assume the data to have, and the number of clusters in a region is not known a priori, which would be a parameter required with most other clustering approaches. DBSCAN works by searching a neighbourhood around a point (specified as a parameter) to look for points in close proximity to form clusters. Clusters are established only if a group fulfils a minimum criterion, i.e. a minimum number of points.

Clustering points in shadow regions with DBSCAN, allows us to extract the number of clusters found by controlling the density of clusters. Intuitively we would expect the shadow regions of ghost objects to exhibit multiple clusters with regular and similar shapes. On the other hand, during a genuine object invalidation attack, we would expect the shadow region to be mainly void of points with points injected by the attacker to elicit a high aggregated score near the region of high weighting as modeled by the exponential decay equations in the axis of analysis. Thus, a distinguishable characteristic of shadows during such an attack would be the small number or no clusters detected. (Fig 17 in Appendix A-D) (a), where clusters are colored and un-clustered points are in black; and (b) where the injected cluster is colored red and un-clustered points are in black).

From DBSCAN results, we use the following features to characterize the shadows of objects:

- **Number of clusters.** This is the number of clusters in the shadow region obtained from DBSCAN.
- **Average density of points in clusters.** This is obtained by taking the total number of points in clusters and averaging out by the number of clusters.
**Attack Classification Model.** The shadow characteristic features obtained from DBSCAN can now be used as input to classification models to distinguish ghost shadows from genuine-looking shadows. In Section [V] we evaluate multiple classifiers such as Logistic Regression, Random Forest and Support Vector Machines (SVM) to determine their performance in the binary classification task of distinguishing between shadows of ghost objects and shadows of objects under an invalidation attack. We chose these classifiers as they tend to do well for low dimensionality data. We found that SVMs generally outperforms the other two classifiers with highest AUC-ROC of 0.972 (for SVMs with linear kernel and polynomial kernel with degree 2), and thus we, recommend using them with GhostBuster. However, SVM learns a decision boundary separating the two classes, which maximises the margin, i.e., the smallest distance between the decision boundary and any of the samples. Since in realistic data, classes are not linearly separable in the feature space, we use a soft-margin SVM [34] which allows for overlapping class distributions.

Note that the attacker can elicit a high anomaly score by opportunistically injecting a single point at the shadow location of highest weighting. However, this is not enough to generate a cluster due to the pre-set minimum number of points required. To defeat the mechanism, an attacker would have to effectively emulate shadows representative of ghost attack shadows, which requires both injecting points at regions of high weighting as well as having the ability to create multiple clusters with sufficient density of points (i.e. to emulate the shadow features of ghost shadows). In Section [V-C] we design a new invalidation attack based on current known capabilities of LiDAR spoofing adversaries and evaluate GhostBuster’s robustness against it. We found that a strong adversary with state of the art capabilities and full knowledge of the behavior of the defense mechanism cannot cause a misclassification of an invalidation attack shadow to a ghost object shadow.

**V. Evaluation**

In this Section we introduce our evaluation of GhostBuster focusing on its effectiveness in detecting ghost and invalidation attacks and its efficiency in doing so. Specifically, we aim to answer the following research questions:

- **RQ1:** How accurately can GhostBuster estimate shadow regions?
- **RQ2:** How accurately can GhostBuster distinguish between genuine shadows and anomalous shadows?
- **RQ3:** How accurately can GhostBuster detect ghost object and invalidation attacks?
- **RQ4:** How efficient is GhostBuster?

**Methodology.** We analyze GhostBuster’s accuracy of 2D shadow region generation by comparing it with the 597 manually labelled shadows of objects (see Section [III] for ground truth collection). We evaluate the 2D region generation separately since 3D regions build on top of it. The significance of 2D vs 3D region estimations in the detection performance is evaluated separately in Subsection [V-B].

**Shadow Region Correspondence Metrics.** To quantify how closely GhostBuster can match the objects’ observed shadows, we measure their Intersection over the Union (IoU) and perform a procrustes shape analysis.

The IoU of two shadow regions gives a measure of similarity in the 2D space they occupy. It is the overlap of the area of the regions divided by the union of the regions and is an indication of how well the regions match in space. An IoU value of 1 means that the two regions are perfectly matched and 0 means the two regions are disjoint.

Procrustes analysis uses landmarks of a shape as a representation and perform isomorphic scaling, translation, and rotation to find the “best fit” between the shapes. The results of the analysis provides us with two metrics: (a) similarity of the shapes; and (b) scale differences of the shapes [35]–[37]. For similarity, values close to 1 mean that the shapes are identical. For scale, value of 1 means that the size of the shapes are identical and anything less than 1 means the ground-truth shadow shape is smaller, and larger than 1 is the opposite.

**Results.** Table [III] summarizes our results across all object types. Detailed results per object type are shown in Appendix A–E Table [VI].

From the median values of the corresponding metrics, it can be observed that, for more than half the objects, the computed shadow matches closely with the ground-truth shadow (IoU, Similarity and Scale values are close to 1). We do observe some variation in the results which can be attributed to measurement inaccuracies and human-errors in the labeling process (see Section [III]), and to over-estimation of shadow areas. GhostBuster uses bounding boxes which are larger than the actual objects and this results in larger shadow regions. However, GhostBuster’s exponential decay approach to weighting the significance of 3D points in shadows (see Section [IV]) compensates for this. This is verified with GhostBuster’s overall accuracy in detecting genuine shadows, ghost and invalidation attacks (see Subsections [V-B] and [V-C]).

**Conclusion.** Overall, our findings suggest that that the shadow regions computed by GhostBuster’s region proposal algorithm have good match with the void regions identified and labeled manually from visual inspection.

**B. RQ2: performance of anomalous shadow detection**

**Methodology.** GhostBuster detects anomalous shadows by performing a 3D point-wise inspection of the shadow regions.
Our objective is to identify shadows that indicate either a ghost attack or a genuine object invalidation attack. Invalidation attacks occur when the adversary aims to take advantage of the presence of GhostBuster to force the system to invalidate a real object. Therefore, we first focus on evaluating how well GhostBuster can detect ghost attacks. GhostBuster’s ability to detect invalidation attacks is evaluated next in Subsection V-C.

To evaluate the performance of GhostBuster’s shadow scoring method, we introduce ghost objects, each individually in 200 random scenes from the KITTI velodyne dataset. We then measure how well GhostBuster is able to distinguish between ghost objects and real objects. The object detector used for the object detection task is a pre-trained Point-GNN [8] model, which is currently ranked within top 10 on the KITTI evaluation benchmark for the object types: car, pedestrian and cyclists. We consider all three object types in our evaluation. There are a total of 867 true objects in the 200 scenes which are counted as True Negatives if GhostBuster does not flag them as anomalous, otherwise they would be counted as False Positives. We performed ghost attacks (see Section III) on the random scenes, for each of the objects, and evaluated the ability of GhostBuster to identify the shadow of these “ghost” objects as anomalous (i.e. True Positive); failure to do so would be a False Negative. We retrieve the bounding boxes coordinates of every detected object in the scene by Point-GNN, which we input into GhostBuster along with the original point cloud. Then, for each object claimed by the 3D object detector, ghost and genuine, we evaluate whether GhostBuster was able to correctly detect ghost object’s shadows as anomalous and real object shadows as non-anomalous.

We evaluated GhostBuster’s scoring method using 2D shadow Regions (BEV) and 3D Shadow Regions (Ray Height and Uniform Height). For uniform height 3D shadow regions, we evaluate using different height values ranging from 0.1m to 0.6m above ground level.

**Shadow Anomaly Detection Performance Metrics.** To evaluate GhostBuster’s ability to detect anomalous shadows we used the Receiver Operating Characteristic (ROC) Curve, the Area Under Curve of ROC curve (AUC-ROC), F1 score and Accuracy. The ROC curve shows the discriminatory ability of the detector. It shows the trade-off between True Positive Rate (TPR) vs False Positive Rate (FPR) at various threshold values. The AUC-ROC summarises the performance of the detector and can be interpreted as the probability of the detector providing an accurate prediction. The F1 score is the harmonic mean of precision and recall where the value of 1 indicates perfect precision and recall. Lastly, Accuracy shows the fraction of the predictions that the detector got right.

**Results.** The results of performance evaluation are summarized in Fig 8-11.

**2D vs 3D Shadow Regions.** First, we observe that 3D shadow regions with uniform height above the ground level outperform the other shadow regions of interest. This can be attributed to the reduction of noisy points (stray reflections or overhanging objects such as branches of trees and sign posts) when we only consider a small volume above ground which captures the LiDAR scan reflections off the ground when there are no objects. This results in a more accurate scoring of the shadow region to detect anomalous shadows that have points in sub-regions of high weighting.

**Height Sensitivity Analysis.** For shadow regions with uniform height above ground, we performed sensitivity analysis to find the optimal height for detection of ghost objects. The height that yields the best AUC-ROC is 0.3m above the ground with a score of 0.93, 0.94 and 0.97 for detection of injected Car, Pedestrian and Cyclist respectively. Shadow regions with uniform height of 0.3m also gave the best maximum F1 Score and Accuracy for detection of the injected objects.

Overall we found that 3D shadows, with uniform height of 0.3m above the ground and an anomaly score threshold of 0.241 provide the best overall trade-off of TPR vs FPR for all injected objects. In particular, with the above configuration, we obtain an overall accuracy of 0.94, TPR of 0.92 and FPR of 0.059 for anomalous shadow regions due to Ghost Attacks. We then took a closer look into the sources of errors. The false negatives (ghost objects not detected by GhostBuster) in the dataset are found to be attributed to injected objects being implanted in regions that are already void of points due to the reduction of noisy points (stray reflections or overhanging objects such as branches of trees and sign posts) when we only consider a small volume above ground which captures the LiDAR scan reflections off the ground when there are no objects. This results in a more accurate scoring of the shadow region to detect anomalous shadows that have points in sub-regions of high weighting.
to incomplete LiDAR measurements (Fig. 18(a) in Appendix A-F). Note that this is due to LiDAR’s failure to take measurements of the ground level. As LiDAR technology gets better, our approach’s accuracy will also improve. The false positives (real objects flagged as potential ghosts by GhostBuster) are due to their shadows having point measurements from other larger objects behind them (Fig. 18(b) in Appendix A-F). This may (very rarely) happen although the safety repercussions are less important in this case since the second object (right behind the first) will be correctly validated.

Conclusion. We evaluated the performance of the shadow region scoring mechanism for various shadow regions of interest for objects detected in the scene. Using 3D shadow region with uniform height of 0.3m above the ground provided the best shadow anomaly detection performance with an AUC-ROC of 0.97, overall Accuracy of 0.94 and F1 Score of 0.89.

C. RQ3: Performance of attack classification

GhostBuster is very effective in detecting anomalous shadows. Nonetheless, we would also like to determine whether an anomalous shadow is a result of a ghost attack or a genuine object invalidation attack, as these should be handled very differently by an end-to-end decision system. In this Section we perform an analysis to first show how the features extracted from shadow regions by GhostBuster’s differ between ghost object shadows and genuine object shadows, and that they can be used to train a binary classifier (subsection V-C1). Next, we evaluate GhostBuster’s classification robustness against a strong invalidation adversary which uses state-of-the-art LiDAR injection capabilities and full knowledge of the classification method aiming to force a genuine shadow to be misclassified as a ghost shadow (subsection V-C2).

1) Feature Characteristics of Shadows: Here we evaluate whether GhostBuster’s selection of features can be potentially used to train a classifier to distinguish between ghost object shadows and genuine object shadows. We used genuine object shadows for comparison with ghost object shadows for two reasons. (a) GhostBuster’s anomaly detection might (but very rarely) incorrectly mark a genuine object’s shadow as anomalous. Being able to distinguish between the two, acts as a second line of validation which can correct GhostBuster’s mistake in the previous phase. This can lead to better utility. (b) An invalidation attack adversary targets genuine object shadows. Thus the distinction between genuine and ghost shadows can serve as a baseline for detecting against invalidation attacks (in the next subsection we use this baseline to design a strong invalidation adversary).

Methodology. We use the 600 scenes (200 scenes × 3 objects injected) from subsection V-B. GhostBuster’s 3D shadow region generation was used to generate shadows of uniform height 0.3m for objects. Using DBSCAN (ε =0.2, min_points=5) we compute the number of 3D point clusters in each shadow region and density of the clusters. The shadows are labeled (ghost vs genuine). We then split them into a training set and a test set (80:20). We use the first set to train six different binary classifiers (see Table IV) and evaluated their performance on the test set. As we are most concerned with TPR and FPR of classifiers for best utility, we use AUC-ROC as the decision criteria to choose the model.

Results. We found that the prevalence rate of feature combination of 0 clusters and 0 cluster density for genuine shadows is 78.5% (2044/2603) and ghost shadows 3.5% (20/567). From Fig. 12 we further observe that 91.5% (2383/2603) of the genuine object shadows are found to have less than 5 clusters and 95.3% (2483/2603) have less than 10 clusters. Of those genuine shadows with clusters, 81.2% (2113/2603) have average clusters density of less than 5 points and 97.3% (2534/2603) have average cluster densities of less than 20 points. These genuine shadow regions are opportunities for an adversary to perform an invalidation attack. A least effort adversary will target shadows which will likely incorrectly be marked as anomalous or force triggering anomaly detection with a single 3D point injected in sub-regions of high importance. Even so, these shadows will result in shadows looking identical than the genuine object shadows.

From the second row in Fig. 12 which shows the feature distribution of anomalous shadows, we observed that the scoring mechanism has removed 94% of the genuine shadows. The shadows attributed to ghost objects have large number of clusters and high average cluster density. The observation of differences in the shadow feature distribution of genuine and ghost shadows suggests that a binary classification model can be used to distinguish ghost from invalidation attacks.

Fig. 12. Distribution of shadow characteristic features for ghost and genuine objects. Top row: shadow characteristic features for all shadows. Bottom row: shadow characteristic features for anomalous shadows.

Table IV summarizes the performance of the different classifiers in distinguishing between shadows of ghost and genuine objects. We found that two classifier models significantly outperform the others in distinguishing between ghost and genuine shadows: 1) SVM with linear kernel or 2) SVM with polynomial kernel of degree 2. From the accuracy and F1-score, the linear kernel seems to have better classification performance over the polynomial kernel, although the AUC is almost equivalent.
TABLE IV

| PERFORMANCE METRIC FOR SHADOW CLASSIFIERS |
|------------------------------------------|
| Logistic Regression | Accuracy | F1-Score | AUC-ROC |
|----------------------|----------|----------|---------|
| Random Forest        | 0.961    | 0.885    | 0.936   |
| SVM-Linear           | 0.964    | 0.895    | 0.972   |
| SVM-Poly(deg=2)      | 0.962    | 0.887    | 0.972   |
| SVM-Poly(deg=3)      | 0.951    | 0.862    | 0.914   |
| SVM-RBF              | 0.973    | 0.922    | 0.963   |

2) Robustness of Classification: Next, we define a strong invalidation adversary and evaluate GhostBuster’s classification robustness against it.

Evasion Attacks on Shadow Classification Model. GhostBuster’s use of shadows, as an invariant for detecting LiDAR spoofing attacks, can incentivize a new class of object invalidation attacks targeting genuine objects’ shadows. Here, we define an invalidation attack by a strong adversary with full knowledge of GhostBuster’s defense mechanisms and state-of-the-art LiDAR spoofing capabilities.

The invalidation attack can be formulated as an evasion attack on the adversarial shadow classification model. We consider a strong adversary who has knowledge of the classifier’s decision boundary and feature representation (i.e. shadow characteristics features). The adversary’s goal is to perform a test-time evasion attack and introduce points in the shadow region of a genuine shadow to change the shadow’s characteristics and cause the shadow classification model to misclassify the genuine shadow as a ghost object shadow, effectively invalidating the real object.

We can evaluate the robustness of the classification according to the capability of the adversary. In our case, we define the attacker’s capability as the total number of points that can be injected in a target shadow region in a single point cloud scene. We refer to this as the adversary’s “point budget”. We can define the invalidation adversary’s budget $B_A$ as:

$$n_0 + n_p = N_c \times \rho_c, \quad \text{s.t. } n_p \leq B_A$$

where $n_0$ is the original number of points in the shadow region, $n_p$ is the number of injected malicious points, $N_c$ is the number of clusters after injection, and $\rho_c$ is the average cluster density after injection.

Intuitively, the invalidation adversary’s optimal strategy against GhostBuster can be defined as follows: Given a set of features for a genuine shadow, inject the minimum number of points, $n_p$, to deceive the classifier by modifying the combination of cluster density and number of clusters, subjected to a point budget $B_A$ and the configuration parameters of DBSCAN used by GhostBuster. This optimal attack strategy can be formalized as follows:

$$\min n_p, \quad \text{s.t. } \exists N_c \in \mathbb{Z}^+ | \ F((n_0 + n_p)/N_c, N_c) = 1$$

where $F(\cdot, \cdot)$ is the output of the classifier, which is one if an attack is identified as a ghost and zero otherwise.

As the complexity of the optimization problem in Eq. (9) is reduced: $n_p \in \mathbb{Z}^+$ is a scalar and the classifier just has two features, the problem can be easily solved by applying simple techniques such as the bisection method.

We evaluate the robustness of the adversarial shadow classifier against an invalidation adversary setting the DBSCAN parameters as before ($\epsilon=0.2$, min_points=5). For the classifiers, we compare the robustness of both linear and non-linear SVM models against these evasion attacks, which aim to inject points into a genuine shadow region.

To visualize the maximum cluster-density combination the attacker can introduce given a budget $B_A$, we use the Maximum Operating Curve (MOC), which shows the set of valid $(\rho_c, N_c)$ combinations on the feature space that can be reached for a given $B_A$ (we use 20, 40, 60, 100 and 200 points). In our previous experiments we found that 78.5% of the genuine shadows have 0 clusters (see Section V-C1). Thus, for solving the problem in Eq. (9), we start exploring cluster-density combinations on the feature space from value 0 for both $N_c$ and $\rho_c = (n_0 + n_p)/N_c$.

Fig. 14. Scatter plot of shadow features and decision regions from SVM Classifier with a polynomial kernel with degree = 2. Dashed lines are operating curve of adversary according to their budget from (0,0).

This is a very strong adversary because we assume that:
(a) the adversary can predict the optimal cluster-density combinations; (b) the adversary can identify where the 3D points should be introduced in the environment to achieve that combination; (c) it is feasible to introduce those measurements.

**Results.** Fig. [13] and [14] are the scatter plots of all shadow features with the decision regions from the boundary of the linear and non-linear SVM classifiers respectively. A point in the red region is the feature combination where the classifier model will label the shadow as a ghost attack and blue region an invalidation attack. The dashed curves represent MOCs for different budgets from the origin (0,0), which is the feature combination for 78.5% of all genuine shadows in our dataset.

In Fig. [13] we observe that for MOC of 20 points, the curve lies completely in the blue region (non-ghost shadow), which indicates that any combination of features with a budget of 20 points would not be able to change the label of a genuine shadow with the original feature combination of (0,0). For MOCs of 40 points and above in the linear SVM decision region, we observe that there are regions where the curves are in the decision region of ghost shadows. This shows that, given the different attacker’s budgets shown in Figure [13] the linear SVM model is only robust to an adversary of 20 points for 78.5% of the genuine shadows.

Similarly, in Fig. [14] we show the decision regions obtained from the SVM (poly deg=2). We observe that for MOCs of up to 100 points, the curves lies completely within the blue decision region for non-ghost shadows. This indicates that the non-linear SVM model is robust against adversary with state-of-the-art LiDAR spoofing capabilities of up to 100 points, as shown in [12]. Our results in Fig. [14] show that the attacker needs to inject up to 200 points (twice the budget of attacker’s capability in [12]) to evade the non-linear SVM classifier.

**Conclusion.** We used DBSCAN to extract features characteristic of ghost and genuine shadows. The features were used to train a binary classifier to identify ghost attacks. The performance of both linear and non-linear SVM is comparable (96%). However, the robustness of the non-linear classifier is clearly superior in terms of robustness to attacks, i.e. the attacker needs to spoof approximately 10 times more points to evade detection by GhostBuster.

**D. RQ4: GhostBuster’s Runtime Efficiency**

**Methodology.** We use the same adversarial dataset as in previous subsections. It consists of 600 ghost objects (3 ghost object types injected in 200 random scenes each) and 2,600 genuine objects across 600 scenes. For each scene, we measure GhostBuster’s end-to-end analysis time for each identified object (genuine and ghost), starting from the time GhostBuster receives the 3D objects bounding box coordinates until GhostBuster labels the object. We also measure the execution time for each component of GhostBuster. GhostBuster is configured to use 3D shadow generation with a uniform height of 0.3m above the ground, an anomaly score threshold of 0.241, DBSCAN for feature extraction with $\epsilon=0.2$, min_points=5, and our pre-trained SVM binary classifier with a polynomial kernel of degree 2. GhostBuster’s prototype implementation is written in Python with 1200 lines of code. We measure the execution time on a machine equipped with an Intel Core i7 Six Core Processor i7-7800X (3.5GHz) and 32GB RAM.

**Results.** Table [V] summarizes our results. The first three rows detail the time taken to process genuine object shadows while the last three are for ghosts (ghost object types average size is depicted in Table [VII] in Appendix A-G). The results show that GhostBuster can process objects in a scene in 0.003s–0.021s on average. We observe that genuine objects are processed much faster than adversarial objects. This is important since this corresponds to the cases most frequently encountered by AVs. The longer duration taken to process adversarial object shadows is mainly due to the feature extraction step, which is triggered when a shadow is deemed anomalous by the shadow scoring mechanism. The variation observed in the total execution time, comes from the different object sizes and the different point densities in their shadows.

Looking at the individual components, we see that shadow generation contributes the least to the overall compute time with around 0.33ms on average across object types. Shadow scoring takes 6.2ms across objects but with discernible variation. The variation comes from the density of points found in the shadows (Fig. [19] in Appendix A-G), as shadow scoring performs a point-wise analysis. Lastly, the shadow verification step only happens if a shadow is deemed anomalous. Thus for genuine objects (which is the most frequent scenario) this step is never triggered. For anomalous shadows, GhostBuster extracts point density features from shadows for classification. This is the most costly step in the adversarial scenarios requiring 10.7ms on average. For feature extraction, GhostBuster uses DBSCAN that performs a point-wise analysis which explains the large variations we observe—DBSCAN’s running time is $O(n \log n)$ in theory while Fig. [20] in Appendix A-G details its empirical evaluation with our dataset.

| OBJECT      | Shadow Generation (ms) | Shadow Scoring (ms) | Shadow Verification (ms) | Total Time (ms) |
|-------------|-------------------------|---------------------|--------------------------|----------------|
| Car         | 0.4±0.3                 | 4±10                | N.A.                     | 4.4±10.3       |
| Pedestrian  | 0.3±0.1                 | 6±8                 | N.A.                     | 6.4±8.1        |
| Cyclist     | 0.3±0.1                 | 3±4                 | N.A.                     | 3.3±4.1        |
| Car (adv)   | 0.4±0.1                 | 10±6                | N.A.                     | 20.46±26.15   |
| Ped. (adv)  | 0.3±0.1                 | 7±5                 | N.A.                     | 17.36±22.15   |
| Cyc. (adv)  | 0.3±0.1                 | 7±6                 | N.A.                     | 19.36±24.15   |

**Conclusion.** We show that GhostBuster can be implemented in real-time applications requiring between 0.003s–0.021s end-to-end average runtime for processing an object in a 3D point cloud. This is only a small fraction of the time a 3D object detector takes to analyze a 3D point cloud—Point-GNN has an average inference time of 0.6s [38].

2The MOC contains discrete values, given the set of valid combinations $(\rho_i, N_i)$. For illustration purposes, we plot the MOC as a continuous contour.
VI. DISCUSSION

Utility. Our evaluation shows that GhostBuster can be used to efficiently and accurately detect attacks on AV 3D perception. However, even though very rarely, it is possible that mistakes will be made. Thus, we caution against using it on its own for driving decision making. Instead, we envision GhostBuster to be part of a more complete set of mechanisms that provide real-time hints and explanations to drivers, vehicle operators or more sophisticated AI systems for autonomous driving. Moreover, we envision GhostBuster contributing to a vehicle’s black box (similar to black boxes in avionics) to facilitate offline incident and forensics analysis.

Opportunities for Future Work. Adversaries might target GhostBuster’s use of shadows to force a genuine object to be misclassified as a ghost (invalidation attack). However, our evaluation shows that GhostBuster is robust against a strong adversary with more advanced capabilities than the current state-of-the-art realistic LiDAR spoofing adversaries. At the same time, it offers a new ability to detect ghost attacks in a highly efficient manner without requiring either extra hardware, or assuming the presence and integrity of other sensors, homogeneous or heterogeneous. For a LiDAR spoofing adversary to succeed launching an invalidation attack against GhostBuster, they will have to closely emulate a ghost object’s 3D shadow. But for that, they would need to have the capability to create reasonably large regions of artificial 3D environments, making the use of depth sensors all together ineffective. Tackling such sophisticated and costly attacks, would require using trustworthy sensor fusion. We plan to explore this in future work.

Lastly, GhostBuster assumes that an adversary cannot remove LiDAR measurements. An adversary that can selectively cancel out 3D point measurements can create artificial shadows for ghost objects. This might be possible if special physical materials with high reflectance properties are strategically placed to cause refraction of LiDAR signals. To the best of our knowledge it is currently an open question whether such attacks are feasible.

VII. RELATED WORK

3D Object Detector Attacks. Prior work showed that point-cloud based 3D object detectors are vulnerable to LiDAR spoofing attacks [10]–[12] and point cloud perturbation attacks [39]–[43]. Wicker and Kwiatkowska [44] further found that 3D object detectors are trained to learn object representation from a “critical point set”, and subtle changes in the input greatly impact the model’s performance. These works show that point cloud based 3D object detectors are not robust, highlighting the need for an orthogonal defence mechanism. GhostBuster looks into the blind spots of these models to successfully detect 3D point spoofing and perturbation attacks.

3D Object Detector Defenses. Existing defenses for 3D point cloud object detection focus on defending against point cloud perturbations [41]. For point injection (or LiDAR spoofing) attacks in AV settings, suggestions were made to use multi-modal sensor fusion [22], [23]. RGB-D fusion was suggested before [14]–[21]. However, these focus on improving the accuracy of 3D object prediction and positioning in challenging, albeit benign scenarios. In [22], [23] the authors suggested sensor fusion to improve attack resilience, but make strong assumptions about the integrity of sensor measurements. GhostBuster, however, makes no assumptions about the availability or integrity of other sensors and solely uses existing characteristics of LiDAR measurements for detection.

More recently, [24] proposed leveraging occlusion patterns to detect spoofed objects. Specifically, their approach exploits the free space between the LiDAR sensor and detected object to obtain the LiDAR point cloud occlusion pattern, together with looking at point cloud distribution in the object’s bounding box. This approach first checks for the presence of occlusions and uses occluding patterns to check for consistency with the point cloud of a detected object. In contrast, GhostBuster looks into the region behind a detected object to determine if the point cloud distribution in this shadow region is characteristic of a realistic shadow as a result of the physical phenomena of object occlusion.

3D Shadows. Prior works studied the negative effects LiDAR ray occlusion by objects (or shadows) has on the accuracy of information measured in 3D object detectors [46], urban environments [47] and in orthophotography for Geospatial Information Systems [48]–[52]. The latter propose occlusion detection methods such as Z-Buffer Algorithm, height-based ray tracing and surface-gradient for post-processing correction to produce high quality orthophotos. Although these algorithms provide highly accurate results, they have long computational time and are not suitable for real-time applications. GhostBuster is the first to provide 3D shadow detection and analysis methods appropriate to the AV domain. Moreover, all the above aim to reduce LiDAR inaccuracy caused by void regions, whereas GhostBuster uses the characteristics of those void regions to verify the objects causing them.

VIII. CONCLUSIONS

LiDARs enable accurate 3D object detection and play an important role in autonomous driving. However, recent works have demonstrated that 3D object detectors can be deceived to detect “ghost” objects by spoofing LiDAR return signals. While sensor fusion could be applied to increase resilience, most prior attempts use sensor fusion to improve the accuracy (and sensitivity) of object detection or assume the availability and integrity of extra sensors. In this work we have proposed GhostBuster, a system which relies solely on existing LiDAR measurements to detect such attacks, and is agnostic to the detector targeted. GhostBuster introduces the concept of 3D shadows and, techniques for mapping shadows to 3D objects, dealing with noisy measurements in shadows, and scoring shadow regions to detect potential attacks. Furthermore, it uses the 3D-point density features extracted from shadow regions to distinguish between ghost attacks and object invalidation.
attacks (a new type of attack targeting the defense system). Our thorough evaluation shows that GhostBuster achieves 94% and 96% average accuracy in identifying anomalous shadows and classifying them as either ghost or invalidation attacks. We further design a strong invalidation adversary aiming to evade classification and found that GhostBuster remains robust. GhostBuster can process an object in a 3D point cloud in real-time (0.003s–0.021s on average).

Our experience developing GhostBuster also highlights two important lessons. Firstly, that introducing defenses, may also introduce new opportunities for adversaries. For example, object invalidation attacks specifically target shadow regions, and could lead to worse consequences than ghost attacks. Thus, it is paramount for defenses to be designed and tested against potential opportunistic adversaries. Secondly, system decisions made based on incomplete perceptions of reality are vulnerable: a ghost attack targets the absence of verification of an object’s expected 3D shadow; an invalidation attack aims to inject unexpected measurements in shadows to make them appear unrealistic. A defense in depth approach that models an optimal attack strategy—such as our modeling of the operating curve of the adversary on decision regions—can help provide strong robustness guarantees. Finally, we believe that this work can spur new directions of research leveraging physical invariants (such as shape, speed, acceleration, movement trajectories, etc.) for verifying the output of 3D perception models used in autonomous agents.

REFERENCES

[1] R. Staff, “What is lidar and how does it help robots see?” Oct 2019. [Online]. Available: https://www.roboticsbusinessreview.com/br/?what-is_lidar_and_how_does_it_help_robots/see/ [2] M. Gips, “The future of lidar and security,” Mar 2020. [Online]. Available: https://www.securitymagazine.com/articles/019077-the-future-of-lidar-and-security [3] J. Porter, “Go read this analysis of what the ipad pro’s lidar sensor is capable of,” Apr 2020. [Online]. Available: https://www.theverge.com/2020/4/16/21226260/ipad-pro-halide-camera-lidar-augmented-reality-scanning [4] D. Coldewey, “Here’s how uber’s self-driving cars are supposed to detect pedestrians,” Mar 2018. [Online]. Available: https://techcrunch.com/2018/03/19/heres-how-ubers-self-driving-cars-are-supposed-to-detect-pedestrians/ [5] “Google spin-off waymo to sell lidar it fought uber on,” Mar 2019. [Online]. Available: https://www.bbc.co.uk/news/47482028 [6] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” in Advances in neural information processing systems, 2017, pp. 5099–5108. [7] B. Yang, W. Luo, and R. Urtasun, “PIXOR: real-time 3D object detection from point clouds,” CoRR, vol. abs/1902.06326, 2019. [Online]. Available: http://arxiv.org/abs/1902.06326 [8] W. Shi and R. Rajkumar, “Point-gnn: Graph neural network for 3D object detection in a point cloud,” ArXiv, vol. abs/2003.01251, 2020. [9] S. Shi, C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li, “Pv-cnn: Point-voxel feature set abstraction for 3D object detection,” in CVPR, 2020. [10] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, “Remote attacks on automated vehicles sensors: Experiments on camera and lidar,” Black Hat Europe, vol. 11, p. 2015, 2015. [11] H. Shin, D. Kim, Y. Kwon, and Y. Kim, “Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications,” in International Conference on Cryptographic Hardware and Embedded Systems. Springer, 2017, pp. 445–467. [12] Y. Cao, C. Xiao, B. Cyr, Y. Zhou, W. Park, S. Rampazzi, Q. A. Chen, K. Fu, and Z. M. Mao, “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. [13] C. Xiao, R. Deng, B. Li, T. Lee, B. Edwards, J. Yi, D. Song, M. Liu, and I. Molloy, “Advit: Adversarial frames identifier based on temporal consistency in videos,” in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 3968–3977. [14] B. Donniard, D. Fox, F. Ramos et al., “Laser and vision based outdoor object mapping,” in Robotics: Science and Systems, 2008. [15] M. Enzweiler, A. Eigenstetter, B. Schiele, and D. M. Gavrila, “Multi-cue pedestrian classification with partial occlusion handling,” in 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 2010, pp. 990–997. [16] M. Enzweiler and D. M. Gavrila, “A multilevel mixture-of-experts framework for pedestrian classification,” IEEE Transactions on Image Processing, vol. 20, no. 10, pp. 2967–2979, 2011. [17] L. Spinello and K. O. Arras, “Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection,” in 2012 IEEE International Conference on Robotics and Automation. IEEE, 2012, pp. 4469–4474. [18] M. Liang, B. Yang, S. Wang, and R. Urtasun, “Deep continuous fusion for multi-sensor 3D object detection,” in Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 641–656. [19] A. Eitel, J. T. Springenberg, L. Spinello, M. Riedmiller, and W. Burgard, “Multimodal deep learning for robust rgb-d object recognition,” in 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 681–687. [20] S. Demetriou, P. Jain, and K.-H. Kim, “Codrive: Improving automobile positioning via collaborative driving,” in IEEE INFOCOM 2018-IEEE Conference on Computer Communications. IEEE, 2018, pp. 72–80. [21] P. Jain, S. Demetriou, and K.-H. Kim, “Determining car positions,” Aug. 13, 2019, uS Patent 10,380,889. [22] R. Ivanov, M. Pajic, and I. Lee, “Attack-resilient sensor fusion,” in 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE). IEEE, 2014, pp. 1–6. [23] K. Yang, R. Wang, Y. Jiang, H. Song, C. Luo, Y. Guan, X. Shi, and L. Shi, “Sensor attack detection using history based pairwise inconsistency,” Future Generation Computer Systems, vol. 86, pp. 392–402, 2018. [24] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, “Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures,” in 29th USENIX Security Symposium (USENIX Security 20). USENIX Association, Aug. 2020, pp. 877–894. [Online]. Available: https://www.usenix.org/conference/usenixsecurity20/presentation/sun [25] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan, “Detecting ground shadows in outdoor consumer photographs,” in European conference on computer vision. Springer, 2010, pp. 322–335. [26] V. Arévalo, J. González, and G. Ambrosio, “Shadow detection in colour high-resolution satellite images,” International Journal of Remote Sensing, vol. 29, no. 7, pp. 1945–1963, 2008. [27] R. Guo, Q. Dai, and D. Hoiem, “Single-image shadow detection and removal using paired regions,” in CVPR 2011, 2011, pp. 2033–2040. [28] N. Al-Najdawi, H. E. Bez, J. Singhai, and E. A. Edirisinghe, “A survey of cast shadow detection algorithms,” Pattern Recognition Letters, vol. 33, no. 6, pp. 752–764, 2012. [29] “Ghostbuster-av project website.” [Online]. Available: https://sites.google.com/view/ghostbuster-av/home [30] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, “Vision meets robotics: The kitti dataset,” International Journal of Robotics Research (IJRR), 2013. [31] A. Dutta and A. Zisserman, “The VIA annotation software for images, audio, and video,” in Proceedings of the 27th ACM International Conference on Multimedia, ser. MM ’19. New York, NY, USA: ACM, 2019. [Online]. Available: https://doi.org/10.1145/3343031.3350535 [32] D. Gatziosl and H.-E. Andersen, “A guide to lidar data acquisition and processing for the forefront of the pacific northwest,” Gen. Tech. Rep. PNW-GTR-768. Portland, OR: US Department of Agriculture, Forest Service, Pacific Northwest Research Station. 32 p, vol. 768, 2008. [33] M. Ester, H.-P. Kriegel, J. Sander, X. Xu et al., “A density-based algorithm for discovering clusters in large spatial databases with noise,” in Kdd, vol. 96, no. 34, 1996, pp. 226–231. [34] C. Cortes and V. Vapnik, “Support-vector networks,” Machine learning, vol. 20, no. 3, pp. 273–297, 1995.
Fig. 15. Scenarios that make it challenging to attribute shadow regions of objects for labelling.

B. 2D Shadow Region Estimation 

The following algorithm computes the 4 corners of a 2D shadow region in the ground plane.

APPENDIX A

A. Challenging scenarios for shadow region estimation.

Fig. 15 illustrates representative cases which render shadow region estimation challenging. The challenging scenarios depicted are:

(a) An object is located in the shadow of another object, thus the shadow of the object is indistinguishable from the shadow that the object is within.

(b) Objects are clustered together (e.g. cars parked together along the side-walk) resulting in a large region of void from the overlapping shadows.
Algorithm 1  Corner points of 2D shadow regions for objects

```
points = []
for each obj in objectsInScene do
    length ← get_shadow_length(d_obj, h, H)  ▶ Eq (3)
    min_grad, max_grad ← 0
    obj_bbox_coords ← get_coords(obj)
    for each coords in obj_bbox_coords do
        grad ← compute_gradient(coords, ref_coords)
        if grad ≤ min_grad then
            min_grad ← grad
            start_1 ← coords
        else if grad ≥ max_grad then
            max_grad ← grad
            start_2 ← coords
        end if
    end for
    end_1 ← get_end_point(start_1, min_grad)
    end_2 ← get_end_point(start_2, max_grad)
    points.append((start_1, start_2, end_1, end_2))
end for
return points
```

C. 3D Shadow Region Estimation,

We propose and evaluate two different ways for estimating 3D shadow regions. Figure 16 helps visualize the intuition behind these approaches. In the former case (a) the height of the shadow is calculated using ray optics; while in the latter (b) we use a uniform height across the length of the shadow.

D. DBSCAN clusters for Ghost and Invalidation Attacks.

Fig. 17 shows an example of a shadow of a ghost object (a) and a shadow of a genuine object under an invalidation attack to pass the shadow anomaly threshold. We show that using DBSCAN we can extract shadow characteristics such as the number of clusters and density of clusters to distinguish between the two cases. The coloring shows how clusters are formed with DBSCAN (16 vs 1).

E. Detailed Correspondence Metrics by Object Types

Table VI provides a break down of the correspondence metrics for shadows of the various objects in the KITTI dataset. GhostBuster’s shadow region estimation was used to generate shadows for objects in the 120 random scenes and the generated shadows were compared with their manually labelled shadow regions. We observe that for all objects (with the exception of Sitting Pedestrian), we obtain good results for IoU and Shape Similarity and Scale scores (i.e. median values are close to 1). The large variance in metrics is due to challenging scenarios (shown in Appendix A-A) and noisy artifacts that makes it difficult to provide proper manual labelling of shadow regions, resulting in some discrepancies in some corner cases.

Fig. 17. DBSCAN results for (a) Ghost Attack and (b) Invalidation Attack.

Fig. 16. Illustration of 3D shadow regions obtained using different height.
TABLE VI
CORRESPONDENCE METRICS BY OBJECT TYPES

|          | Count | IoU          | Similarity   | Scale          |
|----------|-------|--------------|--------------|----------------|
|          |       | mean / median / standard deviation |              | mean / median / standard deviation |              |
| Car      | 439   | 0.757 / 0.765 / 0.143 | 0.098 / 0.365 / 0.384 | 1.24 / 0.970 / 1.69 |
| Pedestrian (Ped.) | 41    | 0.640 / 0.629 / 0.131 | 0.726 / 0.965 / 0.368 | 2.20 / 0.984 / 4.81 |
| Cyclists | 17    | 0.599 / 0.693 / 0.218 | 0.723 / 0.984 / 0.388 | 0.821 / 0.964 / 0.316 |
| Van      | 55    | 0.754 / 0.780 / 0.153 | 0.772 / 0.977 / 0.347 | 1.29 / 0.971 / 2.52 |
| Truck    | 17    | 0.753 / 0.819 / 0.220 | 0.808 / 0.989 / 0.303 | 0.872 / 0.916 / 0.223 |
| Tram     | 6     | 0.801 / 0.773 / 0.067 | 0.64 / 0.754 / 0.407 | 0.673 / 0.814 / 0.371 |
| Sitting Ped. | 1   | 0.525 | 0.332 | 0.865 |
| Miscellaneous | 21  | 0.694 / 0.757 / 0.156 | 0.790 / 0.969 / 0.324 | 1.25 / 0.962 / 0.918 |

F. Genuine Shadow Verification: FP vs FN

Figure [18] illustrates representative examples of (a) false negatives (where a ghost object’s shadow is verified as genuine) and (b) false positives (where a genuine object’s shadow is determined as anomalous) in GhostBuster’s detection of anomalous shadows. In the former case, this is due to the failure of the LiDAR sensor to take ground measurements. In the latter case, even if the front object is labelled as ghost, the object immediately behind it is not, therefore the safety repercussions are diminished.

G. Details and Sources of Variation in Runtime Analysis

Table [VII] shows the average sizes of different object types used in runtime analysis. Fig. [19] and [20] illustrate the dependence of shadow 3D point densities in the calculation of shadow anomaly score and in extracting the shadow classification features respectively.

TABLE VII
AVERAGE TIME TO CALCULATE SHADOW REGIONS FOR OBJECTS AND THEIR AVERAGE DIMENSIONS.

|          | Car   | Pedestrian | Cyclist |
|----------|-------|------------|---------|
| Average Time (s) | 0.0004 | 0.0003 | 0.0003 |
| Average Height (m) | 1.54 | 1.78 | 1.76 |
| Average Width (m) | 1.64 | 0.59 | 0.62 |
| Average Length (m) | 3.76 | 0.83 | 1.76 |

Fig. 19. Average shadow scoring runtime per point density.

Fig. 20. Average DBSCAN runtime per point density.