Assessing the Operational Impact of Poisoning Attacks over Augmented 3D Point Cloud Public Datasets for Connected and Autonomous Vehicles

Marwan Lazrag[1][a], Badis Hammi[1][b], Lorena Gonzalez-Manzano[2][c] and Joaquin Garcia-Alfaro[1][d]
[1]*SAMOVAR, Télécom SudParis, Institut Polytechnique de Paris, Palaiseau, France*
[2]*Universidad Carlos III de Madrid, Leganes, Spain*
{*marwan.lazrag, badis.hammi, joaquin.garcia-alfaro*}*@telecom-sudparis.eu, lgmanzan@inf.uc3m.es*

Keywords: Connected and Autonomous Vehicle (CAV), 3D Point Cloud, LiDAR, CCAM, Poisoning Attack, Dataset, Data Augmentation, GAN, Data Sanitization.

Abstract: Poisoning attacks against public datasets lead to major concerns, such as (i) misclassification of perceived objects when the poisoned data is used for training and (ii) embedding of backdoors that may eventually be triggered later on, when specific conditions in the system apply over the learned models. Its impact over data augmentation models is unclear. While data augmentation reduces the likelihood of poisoning attack success, some valid questions remain. Is data augmentation affecting the impact of poisoning attacks? can it increase the number of poisoned samples or injected backdoors? We explore in this paper some of these questions. We assess the effects of augmenting poisoned 3D point cloud datasets and validate that poisoning is able to evade the sanitizing nature of augmentation techniques when using the concrete case of *Generative Adversarial Network* (GAN) techniques to exemplify the case of data augmentation processing. We also validate that poisoning propagates over the augmented datasets and perturbs the decision made by general-purpose classifiers, in the end. All the experimental material (including tools, datasets, and classifiers) is publicly available, to facilitate reproducibility and to foster further research in the topic.

1 INTRODUCTION

Connected and Autonomous Vehicles (CAVs)[5] have progressed rapidly from experimental prototypes to deployed systems, driven by advances in sensing, computation, and connectivity (Sghaier et al., 2025) (Parekh et al., 2022). Central to CAV operation is perception: a safety-critical subsystem that fuses heterogeneous sensor data, mainly cameras, Light Detection and Ranging (LiDAR), and Radar, to build a reliable, real-time model of the surrounding environment for downstream planning and control (Salguero-Luna et al., 2024). The geometric richness and metric fidelity of LiDAR point clouds, in particular, present both an opportunity and a challenge: they enable precise spatial reasoning (e.g., object localization and shape estimation) but require specialized representations and learning methods to handle sparsity, occlusion, and sensor noise (Eskandarian et al., 2019).

1.1 Motivation

The widespread adoption of 3D point cloud deep learning techniques has significantly advanced the ability of autonomous vehicles to recognize and classify objects on the road, enhancing their navigation and decision-making capabilities (Chen et al., 2019). These techniques are fundamental for real-time object recognition, scene understanding, and ultimately, safe autonomous navigation in complex traffic environments. Despite this progress, a critical gap exists: there is currently no comprehensive simulator tailored to holistically evaluate these techniques under diverse conditions.

The validation of such systems generally necessitates comprehensive simulation environments (Kautz et al., 2024). Among the state-of-the-art tools,

[a] https://orcid.org/0000-0002-0203-9676
[b] https://orcid.org/0000-0002-4470-6406
[c] https://orcid.org/0000-0002-3490-621X
[d] https://orcid.org/0000-0002-7453-4393

[5]For the remainder of this paper, the terms "Connected Autonomous Vehicles (CAVs)" and "Autonomous Vehicles (AVs)" are used interchangeably in the context of perception-level processing and LiDAR sensor inputs. Distinctions related to connectivity or higher-level vehicular functions are outside the scope of this work.

Waymo Simulation City[6] stands out as a highly efficient simulator, but it is proprietary, restricting its accessibility to external researchers. Open-source alternatives such as CARLA[7] and Baidu Apollo[8] offer notable advantages, including high customizability, modularity, and realistic graphics/physics (Yang et al., 2021). These simulators enable researchers to design and test custom scenarios. However, they rely heavily on scenario creation using existing datasets (Kaur et al., 2021), which are often insufficient in size and diversity to simulate prolonged and complex driving scenarios (Li et al., 2024).

To address the limitations of dataset size, researchers frequently employ data augmentation techniques (e.g., Generative Adversarial Networks (GAN)) to augment data and generate synthetic point cloud samples (Sarmad et al., 2019). This approach enables the creation of longer and more comprehensive simulation scenarios, expanding the scope of testing and experimentation (Cheng et al., 2021). However, these augmented datasets often rely on publicly available datasets contributed by other users.

Some studies suggest that data augmentation techniques exhibit a sanitizing effect on data (Strelcenia and Prakoonwit, 2023)(Bissoto et al., 2021a)(He et al., 2019), as they tend to generate synthetic data that reflects the most common features of the original dataset features, which are, by definition benign. In this context, *Karra et al.*(Karra et al., 2022) exploit unsupervised data augmentation, a self-supervised approach, to mitigate backdoor/Trojan attacks, but their evaluation is restricted to a single manipulation type (Trojan triggers) and to relatively modest poisoning rates (generally 10% and a maximum of 20%) (Karra et al., 2022). Similarly, *Qin et al.*(Qin et al., 2023) show that carefully chosen augmentations can suppress the effect of unlearnable or poisoned examples, although their pipeline relies on a verification step prior to augmentation to filter data used for augmentation. *Rebuffi et al.*(Rebuffi et al., 2021) highlight that augmentation, when coupled with weight averaging, may mitigate robust overfitting and increase adversarial robustness, reinforcing the notion that augmentations encourage learning of high-level stable benign features. In medical imaging (Bissoto et al., 2021a), GAN have been used to augment datasets with synthetic images that capture these shared, representative characteristics.

[6]https://waymo.com/blog/2021/07/simulation-city

[7]https://github.com/carla-simulator/carla/

[8]https://github.com/ApolloAuto/apollo

1.2 Research gap and hypothesis

Collectively, these results are encouraging and converge on the claim that augmentation can, under certain conditions, mitigate the influence of data corruption. Nonetheless, such findings are strongly context-dependent and often tested under carefully bounded scenarios, particularly 2D image benchmarks, and thus do not generalize automatically to all data domains. Crucially, they rarely account for the unique challenges posed by 3D point clouds. The 3D point-cloud modality used for automotive perception differs fundamentally from 2D imagery. Mainly, compared to 2D image domains, point cloud data encode geometric, spatial, and structural continuity, making them especially sensitive to subtle perturbations. Augmentation methods designed to replicate or perturb these structures may inadvertently reinforce adversarial manipulations embedded in the dataset, amplifying their prevalence during model training. Therefore, This divergence highlights a critical research gap.

We believe that this dependency on public datasets, in the case of 3D point-cloud, creates a critical attack surface. If a malicious actor introduces a poisoned dataset into this ecosystem, the augmentation process can exacerbate the poisoning, resulting in significantly compromised simulation scenarios. As discussed in the threat model, this could lead to catastrophic outcomes: the deployed point cloud classifier might fail to accurately recognize critical 3D objects such as roads, vehicles, or pedestrians. Furthermore, the adversary could embed a backdoor within the dataset, triggering a specific behavior from the vehicle under certain conditions. In autonomous driving scenarios, such failures can result in severe safety and operational consequences.

1.3 Contributions of this paper

Given these risks, it is essential to evaluate the sanitizing effect of data augmentation on 3D point-cloud data used in automotive perception. In this paper we present a focused case study on GAN-based augmentation and investigate whether common augmentation pipelines amplify or attenuate poisoned examples, and how such effects propagate to downstream decision-making in CAVs. We conduct an operational impact assessment, which consists of estimating the impact of interrupting services and functionalities of a system, e.g., inner functionalities or associated processes, due to an attack. The main contributions are listed below:

- We present an empirical evaluation study of the sanitization effects of data augmentation on 3D

point cloud data for CAVs.

- We analyze the impact of poisoned public datasets on downstream classification tasks when these datasets are subjected to augmentation, highlighting potential risks for decision-making in CAVs.
- We release the complete implementation, including code and datasets, to facilitate reproducibility and future research on poisoning-aware augmentation strategies.

The remainder of this paper is organized as follows: Section 2 surveys related work. Section 3 presents our proposal and modeling choices (adversary and impact models). Section 4 evaluates our experimental results. Section 5 concludes the paper.

2 RELATED WORK

Data augmentation, commonly used to increase the diversity and size of training datasets, for instance to avoid bias for having lack of representation of a particular group (Sharma et al., 2020), has been applied in many contexts, such as computer vision (Yang et al., 2022) or natural language processing (Shorten et al., 2021), among others. A widespread data augmentation technique is the use of GAN (Bissoto et al., 2021b).

In the field of 3D point clouds, many approaches have used GAN for data augmentation purposes. For instance, (Li et al., 2019) introduces PU-GAN, a GAN-based framework designed to upsample sparse 3D point clouds, enhancing data density and uniformity. Another example is (Koh et al., 2023), which proposes an image-to-image GAN framework that generates high-resolution RGB-D images from incomplete point cloud projections.

Attacks against 3D point clouds is also a well studied area. (Wang et al., 2024) presents a novel attack method targeting 3D point cloud models. The attack disrupts model availability by embedding class-specific rotations as imperceptible shortcuts into poisoned point clouds. (Xiang et al., 2019) explores methods to create adversarial examples by generating imperceptible perturbations on 3D point cloud data that deceive a deep learning model. Another example is (Hamdi et al., 2020), which introduces AdvPC, a method for generating adversarial perturbations on 3D point clouds that are highly transferable across different models.

Few proposals address the analysis of poisoning attacks under the use of data augmentation. (Alsereidi et al., 2024) focuses on label-flipping attacks and the use of GAN-generated electroencephalogram data to compromise model performance in a federated learning environment. However, the environment, attacks, and data differ from those in this paper. Complementary to our work, (Qian et al., 2024) studies the transferability of 3D adversarial examples by applying different augmentation techniques, namely drop points, flip, rotation, scale, shear and translation, and analyzing their effects on different models. They conclude that data augmentation has a negative impact on the attack success rate while improving transferability of adversarial examples. In contrast, we analyze whether data augmentation affects the operational impact of poisoning attacks. Also, (Hapuarachchi et al., 2025) uses data augmentation against error-minimizing attacks in the context of traffic sign recognition systems in autonomous vehicles.

Table 1: Related work comparison

Ref.	Poisoning	Operational Impact
(Li et al., 2019)	✓ point cloud upsampling adversarial attacks	x
(Koh et al., 2023)	✓ indoor 3D scene synthesis attacks	x
(Xiang et al., 2019)	✓ 3D adversarial point clouds generation	x
(Hamdi et al., 2020)	✓ 3D adversarial perturbations (transferable)	x
(Wang et al., 2024)	✓ availability poisoning attacks in 3D point clouds	x
(Alsereidi et al., 2024)	✓ data poisoning with EEG label-flipping	x
(Qian et al., 2024)	✓ data augmentation for 3D adversarial examples	x
(Hapuarachchi et al., 2025)	✓ error-minimizing attacks	x
Ours	✓ poisoned 3D point cloud attacks	✓

A summary of analysed works is depicted in Table 1, identifying if poisoning is considered and how and if operational impact is somehow addressed. While poisoning has been studied in different ways, its analysis together with operational impact has not been considered so far.

3 PROPOSAL

We introduce in this section the threat and impact models assumed in our work. The former formalizes the knowledge and capabilities assumed from a potential adversary who may perpetrate some poisoning attacks over our motivational scenario. The latter formalizes the impact assessment of the adversary attacks, over the same scenario.

3.1 Threat Model

We assume an adversary whose objective is to degrade the victim model's performance on the clean test distribution, ultimately forcing it to converge toward random guessing after training on a poisoned 3D point cloud dataset. We first assume the baseline condition that the model M accurately classifies a clean sample $\mathbf{x}$ as its true label Y_{true} (Kurakin et al., 2018):

$$M(\mathbf{x}) = Y_{\text{true}} \tag{1}$$

By introducing poisoned data into a data augmentation process, the adversary aims at ensuring that the newly generated dataset contains even more poisoned samples. This poisoning is based on generating a perturbed sample $\mathbf{x}'$ from a clean sample $\mathbf{x}$ by injecting a small perturbation ρ, such that:

$$\mathbf{x}' = \mathbf{x} + \rho \tag{2}$$

This corrupted data can lead to two major consequences: (i) misclassification of perceived objects when the new dataset is used for training, and (ii) embedding and amplifying a backdoor within the generated dataset, triggering specific behaviors in systems trained on this data. The goal of the attacker is to disrupt the model M, leading to the misclassification:

$$M(\mathbf{x}') = Y' \quad | \quad Y' \neq Y_{\text{true}} \tag{3}$$

We assume a clean-label adversary with partial control over the training data. Specifically, the adversary can inject poisoned 3D point cloud samples into the training set at an incremental poisoning rate (e.g., 0% to 40%), while preserving the original class labels.[9] This scenario may occur when an adversary injects malicious samples to public datasets that are later collected and used to train the victim model. However, the adversary is constrained in several ways. They have access only to the training data and no knowledge of external tools, pre-trained models, or surrogate architectures that could facilitate the generation of effective poison instances. In addition, they lack visibility into the victim's training process, including model architecture, loss function, and hyperparameter choices. Finally, the adversary cannot prevent the defender from manually inspecting the labels of poisoned samples.

[9]This clean-label assumption is commonly used in poisoning attack literature (Wang et al., 2024).

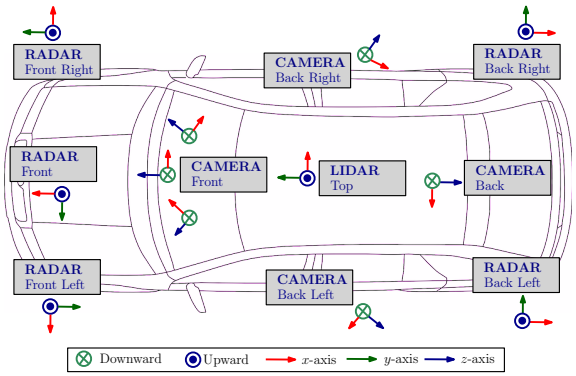

Figure 1: CAV scenario inspired from the nuScenes data collection platform (Caesar et al., 2020).

3.2 Impact Model

To assess the impact of the attacks perpetrated by the adversary, we assume the concept of operational impact quantification and the use of business logic modeling introduced in (Lazrag et al., 2025). The first part, on the evaluation of the operational impact quantification, mainly relies on the computation of two functions:

- **Asset criticality evaluation function:** it identifies the criticality of assets within a given system based on how attacks can propagate through and impact those assets. For each asset, this function assigns a criticality value ranging from 0 to 1. A value of 0 indicates that the asset is not impacted by the attack at all, while 1 indicates that the impact of the attack over such an assets is at its maximum.
- **Impact propagation function:** it quantifies how an external event (e.g., a failure or an attack) propagates its effects over the operational functions and the operational processes associated to a given system, in probabilistic terms.

The two previous functions assume the existence of a *resource dependency graph*, representing the dependencies between all the assets in the system, and a *mission dependency graph* describing the relationships between system-level functions and operational processes associated with each asset. In both cases, nodes represent assets, system functions, and operational processes, while edges capture the interdependencies between them.

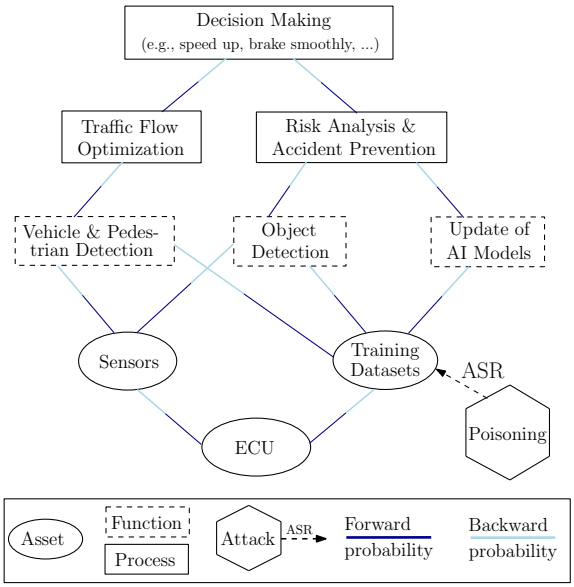

Figure 2: Assessing the impact of a poisoning attack against the training datasets of the CAV scenario in Figure 1, using the impact propagation model defined in (Lazrag et al., 2025).

Figure 1 illustrates the system model of a CAV scenario, inspired from the nuScenes dataset[10] (Caesar et al., 2020), which details representative assets such as sensors and controllers. Based on this model, we construct the corresponding resource and mission dependency graphs. Figure 2 presents our impact assessment model, combining both types of dependencies into a unified graph[11]. To clearly distinguish between the two subgraphs, assets are depicted as circle-shaped nodes (representing the resource dependency graph), while system functions and operational processes are shown as rectangle-shaped nodes (representing the mission dependency graph).

The example depicted in Figure 2 also describes the correlation between assets, system functions, and operational processes in a representative CAV scenario from (Caesar et al., 2020). It contains three assets[12]: *sensors*, *training datasets* and *Electronic Control Unit (ECU)*; three system functions: *Vehicle & Pedestrian Detection*, *Object Detection*, and *Update of AI models*; and three operational processes: *Traffic Flow Optimization*, *Risk Analysis & Accident Prevention*, and *Decision Making*. The *Decision Making* process represents the adversary's ultimate target. By launching a poisoning attack against the asset *Training Datasets*, the adversary seeks to disrupt the outcomes of this process, for instance, by misleading the autonomous vehicle and causing it to make incorrect decisions.

In this propagation model, edges connecting vertices represent interdependency probabilities, specifically Forward and Backward probabilities that quantify how degradation propagates between system components. The Attack entity is characterized by an Attack Success Rate (ASR), which measures the initial detrimental impact of the poisoning attack on the targeted operational process. This ASR serves as the initial probability that cascades through the interconnected graph structure, influencing both Forward and Backward probabilities of all connected vertices. The combined effect of these probability interactions enables the quantification of the operational impact stemming from the attack against the root operational process. The ASR thus acts as the propagation catalyst, where the initial attack probability diminishes or amplifies as it traverses the dependency network, ultimately determining the extent of system degradation. A practical demonstration of this impact propagation mechanism is illustrated in Section 4.3.2, Figure 6.

4 EXPERIMENTAL METHODOLOGY

In this section, we present our experimental framework and the results obtained.

4.1 Overview

As Figure 3 shows, we designed two experimental scenarios to evaluate the impact of data poisoning under different conditions:

1. The classifier is trained on the original (baseline) dataset.

[10]The nuScenes dataset is a large-scale, multimodal benchmark (camera, LiDAR, radar, IMU and rich 3D annotations) that is widely used in the literature for perception, tracking and prediction tasks. Its standardized evaluation and provided devkit have made it a de facto reference in autonomous-driving research.

[11]The tool used to develop the impact-assessment model and to evaluate the effects of poisoning attacks on vehicle operational functions is derived from existing work (Lazrag et al., 2025). For reproducibility purposes, we release the full implementation and modifications of the original work and tools as a Docker-deployable codebase (source code, Dockerfiles, and detailed implementation and deployment instructions) at `https://github.com/Marwanlz/Assessing_Operational_Impact_Poisoning_3d_PointCloud-CAV`.

[12]For clarity, all vertices corresponding to non-impacted assets (e.g., Camera) and their directly linked operational functions (e.g., Weather Conditions Analysis, Itinerary Optimization, and so on) have been omitted from the diagram.

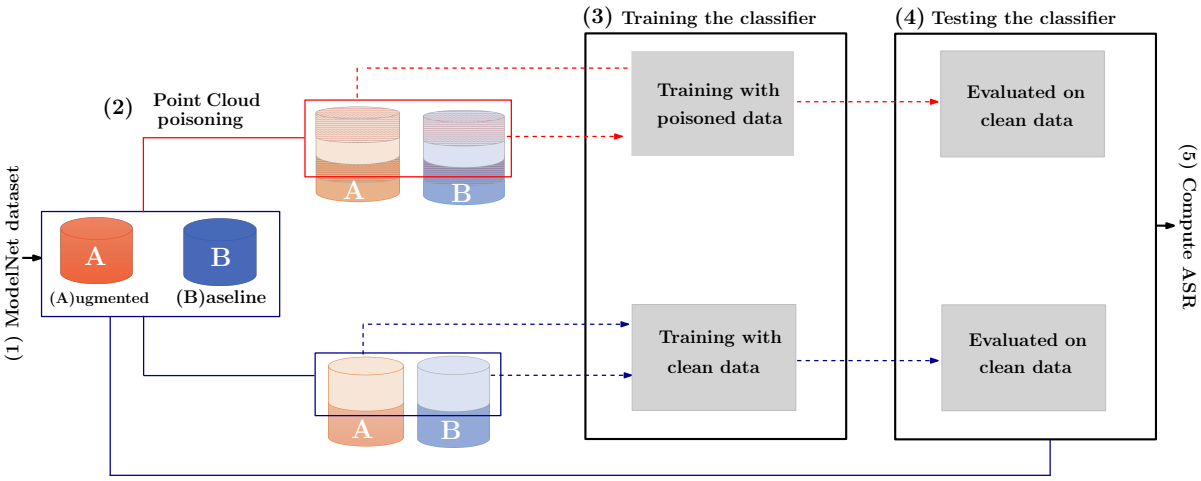

Figure 3: Methodology to compute the Attack Success Rate (ASR) associated to the motivational scenario depicted in Figure 2.

2. The classifier is trained on an augmented dataset, which includes several new samples generated using the 3D-GAN framework (Wu et al., 2016).

The rationale behind these two scenarios is to assess the effect of poisoning both with and without the use of augmentation techniques. By doing so, we can achieve an operational impact assessment, which consists of estimating the impact of interrupting services and functionalities of a system, e.g., inner functionalities or associated processes, due to the poisoning attack. In both settings, the training datasets are of equal size to ensure a fair and consistent comparison. The experiments in each scenario are conducted as follows. First, we preprocess the ModelNet dataset[13] (Wu et al., 2015), which contains a variety of 3D object models: we select two classes for binary classification (a primary target class and a secondary, non-target class), split the data into training and test sets, and reserve only the primary-class training samples for manipulation (all secondary-class and test samples remain unaltered). For training we used 3,000 files (1,500 per class), and for testing we used 600 files (300 per class). The dataset is class-balanced with an 83%/17% train/test split. We report exact counts to ensure experimental reproducibility and to allow fair assessment of sampling variability.

Next, in both scenarios, we train our classifier on the clean training data, using original 3D objects for the first scenario (baseline scenario) and synthetic 3D objects generated by the 3D-GAN framework for the second (augmented scenario), and evaluate its performance on the untouched test set to establish baseline metrics. Finally, we simulate an adversary by modifying the 3D shapes of a subset of primary-class samples and injecting these poisoned examples into the training set at incrementally increasing rates (0 % to 40 %). For each Poisoning rate, we retrain the classifier on the modified dataset and evaluate it on the same clean test set. This procedure allows us to assess how varying levels of data poisoning, both with and without augmentation, affect the classification accuracy and attack success.

In contrast to saliency-based point-dropping techniques such as proposed by *Zheng et al.*(Zheng et al., 2019) or point-detach strategies that iteratively remove high-importance points (Yang et al., 2019), we employed a simpler poisoning method, typically, we randomly removed 50% of the points from each poisoned file, without applying any ranking or importance criterion.

It is worth noting that the experiments conducted in the second scenario are preceded by the creation of an augmented dataset using the 3D-GAN framework. In this setting, the poisoning data is deliberately introduced into the training dataset of the 3D-GAN. The newly generated 3D samples are then used to train the classification model for the primary class, whereas the data for the secondary class remains unaltered. It is worth noting that, the test set used in this scenario is identical to that of Scenario 1, ensuring a consistent evaluation protocol. Figure 4 shows (a) an original 3D object, (b) a synthetic 3D object generated by a GAN trained on the clean dataset, and (c) a synthetic 3D object generated by a GAN trained on a poisoned dataset. This illustration highlights the

[13]ModelNet was introduced as a large-scale 3D Computer-Aided Design (CAD) model dataset and is widely used as a benchmark for 3D shape / point-cloud research. It is frequently employed in studies of adversarial attacks and defenses on 3D point clouds.

effects of dataset poisoning on the quality and characteristics of GAN-generated 3D objects.

(a) (b) (c)

Figure 4: (a) Original object; (b) GAN output from clean data; (c) GAN output from poisoned data.

4.2 Experimental setup and metrics

The experimental setup is designed to evaluate the classification performance against data poisoning attacks[14]. The experiments are conducted on a machine equipped with an Intel i7-11850H CPU processor and an NVIDIA RTX A4000 GPU. Data augmentation via 3D-GAN is conducted using PyTorch[15]. Training of the binary classifier is conducted using TensorFlow[16]. The classifier is built using an existing InceptionNet architecture, which was adapted by us for binary classification.

The classifier is trained with the Adam optimizer (Kingma and Ba, 2014), using binary cross-entropy as the loss and a sigmoid activation on the output to produce probabilistic predictions. The Adam optimizer provides adaptive learning-rate updates that speed and stabilize convergence. Binary cross-entropy is the canonical log-loss for two-class problems. And the sigmoid yields well-interpretable posterior scores for thresholding or calibration. These settings are standard and widely adopted for binary classification in deep-learning work (Ruby et al., 2020)(Chollet and Chollet, 2021)(Schaad and Binder, 2022). The training of the classifier is conducted over 20 epochs with a batch size of 32, which provides a good trade-off between convergence speed and performance consistency on the validation dataset. The model performance is monitored at each epoch, using validation accuracy to ensure stable learning.

To align with the literature, we evaluate the classifier's robustness using the F1 score. However, We also use the Matthews correlation coefficient (MCC) metric. F1 is a harmonic mean of precision and recall, it compactly measures a detector's ability to find true positives while limiting false alarms, making it well suited for tasks that prioritise the positive class. Whereas MCC is a correlation coefficient that uses all four confusion-matrix entries and ranges from -1 (total disagreement) to $+1$ (perfect prediction) (with 0 means random prediction). Because it accounts for true negatives as well as positives, MCC provides a balanced, prevalence-insensitive assessment of overall classifier quality. Reporting both metrics is essential, F1 reflects the precision–recall trade-off for the target class, while MCC reveals overall performance and exposes pathological behaviour on imbalanced data (e.g., models that attain high F1 by exploiting a tiny positive class but fail on negatives). Together they prevent over-claiming detector performance and make hidden failure modes visible.

We also report a third metric to quantify poisoning effectiveness on classifier outputs: the Attack Success Rate (ASR), Formally:

$$\text{ASR} = \frac{\text{FN}_{\text{after attack}} - \text{FN}_{\text{before attack}}}{\text{Total number of samples}} \cdot 100$$

where N is the total number of tested samples, and $FN_{\text{before attack}}$ $FN_{\text{after attack}}$ are the counts of false negatives before and after the poisoning, respectively. This ASR measures the increase in missed detections attributable to the attack, normalized by the evaluation set size. We focus on false negatives because poisoning in our threat model aims to increase wrongful non-detections (samples that should be flagged but are not). In other words, we consider only false negatives (not false positives) because the evaluation targets the primary class exclusively. In this setting, true positives are primary-class samples correctly identified; false negatives are primary-class samples incorrectly classified due to the poisoning; and false positives are non-primary samples incorrectly labeled as primary. This focus ensures that reported metrics capture degradations in recognition of the targeted class.

All experiments were repeated five times. The reported values are the mean across runs. Standard deviations were consistently small, demonstrating the stability of the results. For clarity, in the next section, we omit the standard deviations from the main figures and tables.

4.3 Experimental results

In the following, the analysis of the poisoning attack is presented (Section 4.3.1), together with the impact quantification study (Section 4.3.2).

[14]The experiments (including source code, original datasets, augmented datasets, poisoned dataset, data augmentation code, and data classification code) are publicly available at `https://github.com/Marwanlz/Assessing_Operational_Impact_Poisoning_3d_PointCloud-CAV`

[15]`https://pytorch.org`

[16]`https://tensorflow.org`

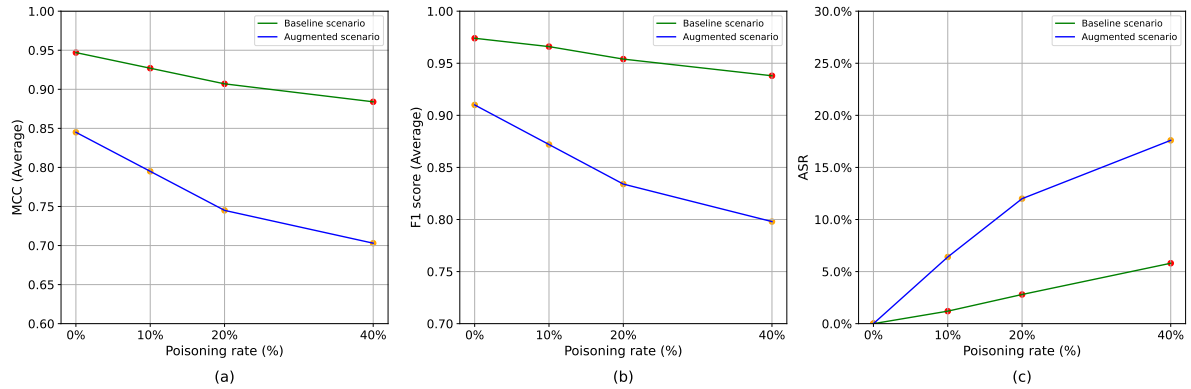

Figure 5: Dependencies between metrics and poisoning rate: (a) Dependency between MCC and Poisoning rate; (b) Dependency between F1 score and Poisoning rate; (c) Dependency between ASR and Poisoning rate.

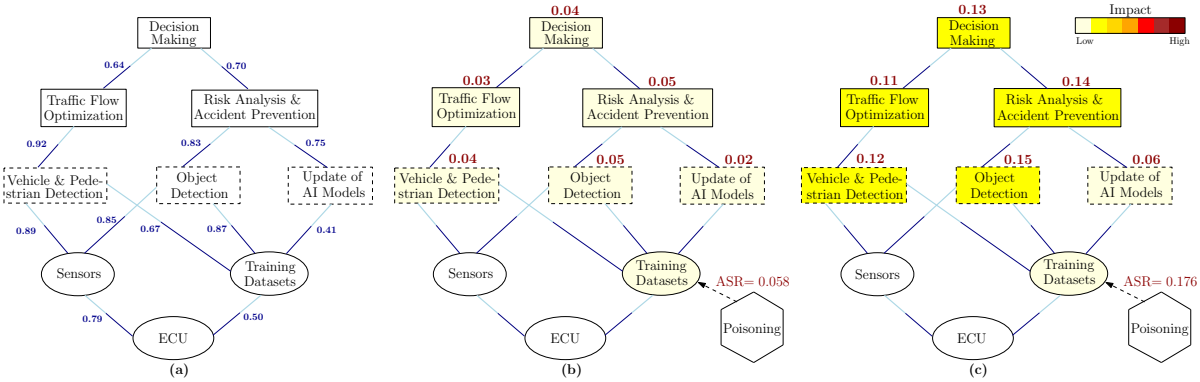

Figure 6: Impact assessment model: (a) Nominal system state; (b) Baseline scenario with a 40% poisoning rate; (c) Augmented scenario with a 40% poisoning rate.

4.3.1 Poisoning analysis

The classification metrics results are summarised in Figure 5 for both the Baseline scenario and the Augmented scenario (see Section 4.1), using the metrics defined in Section 4.2. In both scenarios, only the training set of the primary class was manipulated by injecting poisoned samples at rates ranging from 0% (without poisoning) to 40%. The secondary class and the entire test set were left unchanged to isolate the effect of poisoning on model performance.

In each figure, the x-axis displays the reported classification metric (MCC, F1, ASR) and the y-axis indicates the poisoning rate (in %). Across both evaluation scenarios (Baseline and Augmented), the classifier's performance degrades monotonically with increasing poisoning: MCC and F1 decline while Attack Success Rate (ASR) rises, and the performance gap between scenarios widens as the poisoning load grows. At low poisoning levels the two setups differ only marginally (≈ 0.1 point in MCC and a similar delta in F1), indicating comparable initial robustness. By 20% poisoning the Baseline still yields strong detection (MCC ≈ 0.90, F1 ≈ 0.95) with a small ASR (2.8%). At the highest injection rates the divergence becomes pronounced: the Baseline records MCC = 0.88, F1 = 0.93, ASR = 5.8%, whereas the Augmented setup drops to MCC = 0.70, F1 = 0.79, ASR = 17.6%.

These results indicate that, under our threat model and augmentation pipeline, the augmentation amplifies the impact of poisoned samples: rather than diluting adversarial artefacts, the augmentation process appears to reinforce distributional modes that the attacker exploits, increasing misclassification rates and overall attack effectiveness. This behavior is consistent with a mechanism where augmentation shifts model emphasis toward features present in both clean and poisoned examples, thus enlarging the adversary's effective feature space. We recall that all reported values are averages over repeated runs.

Table 2: Assessment of the impact on the operational function *Decision Making* based on ASR values

Scenario	Poisoning Rate	ASR	Operational Impact
Baseline	10%	1.2 %	1 %
	20%	2.8 %	2 %
	40%	5.8 %	4 %
Augmented	10%	6.4 %	5 %
	20%	12 %	9 %
	40%	17.6 %	13 %

4.3.2 Impact Quantification

We use the ASR as a concrete, operational indicator of poisoning effectiveness and feed it into the impact-propagation model from Section 3.2 to quantify downstream effects on the *Decision Making* operational function. Concretely, ASR values measured for each poisoning rate are treated as probabilistic inputs to the propagation function, which maps classifier degradation to the likelihood of impaired decision outcomes in the motivational scenarios.

Figure 6 illustrates the impact assessment model across three use cases. For clarity, the assets: *Electronic Control Unit (ECU)*, *sensors*, and *training datasets* are replaced by *Asset 1*, *Asset 2*, and *Asset 3*; the system functions: *Vehicle & Pedestrian Detection*, *Object Detection*, and *Update of AI models* by *Function 1*, *Function 2*, and *Function 3*; and the operational processes: *Traffic Flow Optimization*, *Risk Analysis & Accident Prevention*, and *Decision Making* by *process 1*, *process 2*, and *process 3*. Figure 6.a depicts the nominal (pre-attack) system state where edges encode interdependencies between assets. While the model supports bidirectional impacts, this particular scenario exhibits only downstream propagation, meaning all backward impact probabilities are set to zero. For visual clarity, only the forward impact probabilities (shown in dark blue) are displayed, as the reverse probabilities would contribute no meaningful information to the analysis. In Figure 6.a, the edge values between operational functions and processes (represented as rectangular nodes) were manually assigned based on domain expertise, as modeling these relationships requires in-depth knowledge of vehicle activities (Lazrag et al., 2025). For interdependencies between assets (represented as circular nodes), edge values were derived from the nuScenes dataset (Caesar et al., 2020).

Figure 6.b depicts the Baseline scenario with a 40% poisoning rate, while Figure 6.c illustrates the Augmented scenario under the same poisoning rate. These figures demonstrate how an attack propagates its impact across the graph, with interdependencies recalculated accordingly. The degradation level of each asset or operational function (represented as rectangles) is indicated by a red numerical value above each vertex. A color gradient from white (no impact) to yellow (low impact) to dark red (severe impact) visually encodes the extent of degradation on these vertices. This value quantifies the perturbation in the execution and performance of vehicular functions. It ranges from 0 (no impact) to 1 (very high impact), corresponding to 0% and 100%, respectively. For example, in the baseline scenario, the operational impact of the poisoning attack on the *Traffic Flow Optimization* function is 0.03 (3%), as shown in Figure 6.b. In the augmented scenario, the operational impact on the *Traffic Flow Optimization* function reaches 0.11 (11%), as shown in Figure 6.c.

Table 2 summarizes the impact probability of the poisoning attack on operational functions for both scenarios. The resulting assessment exhibits two robust patterns. First, impact probability grows monotonically with ASR, denoted . That is, higher attack-induced misclassification directly increases the chance that decision-making functions receive corrupted inputs and produce incorrect outcomes. Second, the Augmented scenario consistently yields substantially larger operational impacts than the Baseline for the same poisoning rate. For instance, at 40% poisoning the assessed probability that *Decision Making* is affected rises from 4% (Baseline) to 13% (Augmented), hence, an over threefold increase in impact under our modeling assumptions. Hence, the analysis demonstrates a clear and actionable insight: augmentation, amplifies attack effectiveness and materially increases the probability of operational disruption in decision-making functions.

Finally, while this evaluation focuses on a specific poisoning attack and a GAN-based augmentation technique, the observed increase in misclassification and operational impact is not limited to this setting. Comparable effects may also arise with other attacks, potentially leading to similar performance degradation and increased operational impact under our assessment model. Similarly, augmentation techniques other than GANs that expand the training dataset may influence the propagation of poisoned samples, resulting in comparable amplification effects.

5 CONCLUSION

We addressed in this paper the issue of poisoning attacks against public datasets. More precisely, the case of poisoning attacks over augmented 3D point cloud public datasets for Connected and Autonomous Vehi-

cles (CAV) scenarios. Poisoning attacks are known to lead to misclassification of perceived objects. Even worse, they can assist adversaries to embed backdoors that may eventually be triggered later on, when specific conditions in the system apply over the learned models. We assessed the operational impact of this attacks over data augmentation models. While data augmentation reduces the likelihood of poisoning attack success, we addressed whether data augmentation keeps affecting the impact of poisoning attacks over general purpose classifiers. We experimentally validate that data augmentation can even increase the number of poisoned samples, hence augmenting as well the effects of augmenting poisoned 3D point cloud datasets. We validated that poisoning is able to evade the sanitizing nature of augmentation techniques under the concrete case of GAN techniques. Our results validate as well that poisoning propagates over the augmented datasets and perturbs the decision made by general-purpose classifiers. Extending this evaluation to other 3D point cloud datasets remains challenging, given the limited availability of such datasets for CAV scenarios.

ACKNOWLEDGEMENTS

The work presented in this paper was conducted within the framework of the Horizon Europe AI4CCAM project (grant agreement 101076911), addressing trustworthiness of artificial intelligence in the context of Connected, Collaborative and Automated Mobility. Lorena González and Joaquin Garcia-Alfaro are being partially supported by Project PID2023-150310OB-I00 (MORE4AIO) funded by MCIU/ AEI / 10.13039/501100011033 / FEDER, UE. We thank the anonymous reviewers for their valuable comments and helpful suggestions.

REFERENCES

Alsereidi, M., Awadallah, A., Alkaabi, A., Yoon, S., and Yeun, C. Y. (2024). Data poisoning against federated learning: Comparative analysis under label-flipping attacks and gan-generated eeg data. In *2024 2nd International Conference on Cyber Resilience (ICCR)*, pages 1–5. IEEE.

Bissoto, A., Valle, E., and Avila, S. (2021a). Gan-based data augmentation and anonymization for skin-lesion analysis: A critical review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1847–1856.

Bissoto, A., Valle, E., and Avila, S. (2021b). Gan-based data augmentation and anonymization for skin-lesion analysis: A critical review. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1847–1856.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., and Beijbom, O. (2020). nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631.

Chen, Q., Tang, S., Yang, Q., and Fu, S. (2019). Cooper: Cooperative perception for connected autonomous vehicles based on 3d point clouds. In *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pages 514–524. IEEE.

Cheng, M., Li, G., Chen, Y., Chen, J., Wang, C., and Li, J. (2021). Dense point cloud completion based on generative adversarial network. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–10.

Chollet, F. and Chollet, F. (2021). *Deep learning with Python*. Simon and Schuster.

Eskandarian, A., Wu, C., and Sun, C. (2019). Research advances and challenges of autonomous and connected ground vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 22(2):683–711.

Hamdi, A., Rojas, S., Thabet, A., and Ghanem, B. (2020). Advpc: Transferable adversarial perturbations on 3d point clouds. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XII 16*, pages 241–257. Springer.

Hapuarachchi, T., Dang, L., and Xiong, K. (2025). Securing traffic sign recognition systems in autonomous vehicles. *arXiv preprint arXiv:2506.06563*.

He, Z., Xie, L., Chen, X., Zhang, Y., Wang, Y., and Tian, Q. (2019). Data augmentation revisited: Rethinking the distribution gap between clean and augmented data. *arXiv preprint arXiv:1909.09148*.

Karra, K., Ashcraft, C., and Costello, C. (2022). Sanitais: Unsupervised data augmentation to sanitize trojaned neural networks. In *2022 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech)*, pages 1–6. IEEE.

Kaur, P., Taghavi, S., Tian, Z., and Shi, W. (2021). A survey on simulators for testing self-driving cars. In *2021 Fourth International Conference on Connected and Autonomous Driving (MetroCAD)*, pages 62–70. IEEE.

Kautz, M., Hammi, B., and Garcia-Alfaro, J. (2024). Platelet: Pioneering security and privacy compliant simulation for intelligent transportation systems and v2x. In *2024 22nd International Symposium on Network Computing and Applications (NCA)*, pages 61–67. IEEE.

Kingma, D. P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Koh, J. Y., Agrawal, H., Batra, D., Tucker, R., Waters, A., Lee, H., Yang, Y., Baldridge, J., and Anderson, P. (2023). Simple and effective synthesis of indoor 3d scenes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 1169–1178.

Kurakin, A., Goodfellow, I. J., and Bengio, S. (2018). Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC.

Lazrag, M., Kiennert, C., and Garcia-Alfaro, J. (2025). *Quantifying the Impact Propagation of Cyber Attacks Using Business Logic Modeling*, pages 49–71. Springer Nature Switzerland, Cham.

Li, R., Li, X., Fu, C.-W., Cohen-Or, D., and Heng, P.-A. (2019). Pu-gan: a point cloud upsampling adversarial network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7203–7212.

Li, Y., Yuan, W., Zhang, S., Yan, W., Shen, Q., Wang, C., and Yang, M. (2024). Choose your simulator wisely: A review on open-source simulators for autonomous driving. *IEEE Transactions on Intelligent Vehicles*.

Parekh, D., Poddar, N., Rajpurkar, A., Chahal, M., Kumar, N., Joshi, G. P., and Cho, W. (2022). A review on autonomous vehicles: Progress, methods and challenges. *Electronics*, 11(14):2162.

Qian, F., Zou, Y., Xu, M., Zhang, X., Zhang, C., Xu, C., and Chen, H. (2024). A comprehensive understanding of the impact of data augmentation on the transferability of 3d adversarial examples. *ACM Transactions on Knowledge Discovery from Data*.

Qin, T., Gao, X., Zhao, J., Ye, K., and Xu, C.-Z. (2023). Learning the unlearnable: Adversarial augmentations suppress unlearnable example attacks. *arXiv preprint arXiv:2303.15127*.

Rebuffi, S.-A., Gowal, S., Calian, D. A., Stimberg, F., Wiles, O., and Mann, T. A. (2021). Data augmentation can improve robustness. *Advances in neural information processing systems*, 34:29935–29948.

Ruby, U., Yendapalli, V., et al. (2020). Binary cross entropy with deep learning technique for image classification. *Int. J. Adv. Trends Comput. Sci. Eng*, 9(10).

Salguero-Luna, S. A., Ramirez-Gutierrez, K. A., and Martinez-Cruz, A. (2024). A state-of-the-art review on attacks and defense mechanisms for lidar on autonomous vehicles. *IEEE Transactions on Intelligent Transportation Systems*.

Sarmad, M., Lee, H. J., and Kim, Y. M. (2019). Rl-gan-net: A reinforcement learning agent controlled gan network for real-time point cloud shape completion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5898–5907.

Schaad, A. and Binder, D. (2022). Deep-learning-based vulnerability detection in binary executables. In *International Symposium on Foundations and Practice of Security*, pages 453–460. Springer.

Sghaier, K., Hammi, B., Gharbi, G., Merdrignac, P., Parrend, P., and Verna, D. (2025). Advancing security in software-defined vehicles: A comprehensive survey and taxonomy. *arXiv preprint arXiv:2510.09675*.

Sharma, S., Zhang, Y., Ríos Aliaga, J. M., Bouneffouf, D., Muthusamy, V., and Varshney, K. R. (2020). Data augmentation for discrimination prevention and bias disambiguation. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pages 358–364.

Shorten, C., Khoshgoftaar, T. M., and Furht, B. (2021). Text data augmentation for deep learning. *Journal of big Data*, 8(1):101.

Strelcenia, E. and Prakoonwit, S. (2023). A survey on gan techniques for data augmentation to address the imbalanced data issues in credit card fraud detection. *Machine Learning and Knowledge Extraction*, 5(1):304–329.

Wang, X., Li, M., Xu, P., Liu, W., Zhang, L. Y., Hu, S., and Zhang, Y. (2024). PointAPA: Towards availability poisoning attacks in 3D point clouds. In *European Symposium on Research in Computer Security*, pages 125–145. Springer.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. (2016). Learning a probabilistic latent space of object shapes via 3D generative-adversarial modeling. *Advances in neural information processing systems*, 29.

Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J. (2015). 3D ShapeNets: A Deep Representation for Volumetric Shapes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1912–1920.

Xiang, C., Qi, C. R., and Li, B. (2019). Generating 3D adversarial point clouds. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9136–9144.

Yang, G., Xue, Y., Meng, L., Wang, P., Shi, Y., Yang, Q., and Dong, Q. (2021). Survey on autonomous vehicle simulation platforms. In *2021 8th International Conference on Dependable Systems and Their Applications (DSA)*, pages 692–699. IEEE.

Yang, J., Zhang, Q., Fang, R., Ni, B., Liu, J., and Tian, Q. (2019). Adversarial attack and defense on point sets. *arXiv preprint arXiv:1902.10899*.

Yang, S., Xiao, W., Zhang, M., Guo, S., Zhao, J., and Shen, F. (2022). Image data augmentation for deep learning: A survey. *arXiv preprint arXiv:2204.08610*.

Zheng, T., Chen, C., Yuan, J., Li, B., and Ren, K. (2019). Pointcloud saliency maps. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1598–1606.