Adversarial Perturbations on Sensor Data for Compromising Robotic Path Planning in Simulated Environments

Alejandro Almeida*†, Daniel Aviles Rueda†
*Analytics for Cyber Defense Lab,
†Department of Electrical and Computer Engineering,
Florida International University, Miami, FL, USA

***Abstract*—Recent deployments of autonomous robotic systems rely heavily on machine learning (ML) techniques in the path planning process; hence these systems are prone to adversarial perturbations in sensor data. In this work, we examine how limited perturbation applied to simulated LiDAR inputs to an ML-based path planner in a 20x20 grid environment affects its operation and therefore highlights the resultant cybersecurity risks. To allow for reproducibility, we created a simulation in which a mobile robot moves from a position from (2,2) to (18,18) using an 8 ray LiDAR sensor. The MLP was trained using a data set containing 200-1000 labeled LiDAR scenes and after several passes through the data achieved 100% goal-reach rate. Projected Gradient Descent (PGD) attacks with and configured with the parameters shown in the table, i.e., epsilon 0.1, 5 -10 iterations of the reformas the success probability were reduced to 65%, and resulted on increasing the number of collision events. Adversarial training procedures restored robustness by resulting in a 94 percent success rate. Through experiments between various grid configurations marginal statistical differences was discovered as can be seen by t-test p=0.0698. The main contributions of the current work are a lightweight, open-source designed framework aimed at educational use, novel understanding of the vulnerabilities of MLP architectures, to develop practical defense mechanisms, addressing critical issues of lack of accessibility of adversarial machine learning works in robotics context.**

***Index Terms*—Adversarial Machine Learning, Robotic Path Planning, LiDAR Perturbations, Cybersecurity, Simulation**

I. Introduction

The rise of autonomous robots in applications such as warehouse logistics and urban delivery underscores the critical role of robust path planning algorithms. However, integrating machine learning (ML) models that process sensor data, such as LiDAR, for real-time decision-making introduces cybersecurity vulnerabilities. Adversarial machine learning (AML) attacks, where subtle input perturbations lead to erroneous outputs, threaten these systems [1, 2]. In robotics, such attacks can distort environmental perception, causing trajectory deviations and safety risks [3]. This study addresses the underexplored application of AML to robotic path planning in simulated environments. We investigate how adversarial perturbations on LiDAR data can compromise an ML-based path planner, simulating cyber breaches like man-in-the-middle attacks on sensor streams. While prior research has focused on physical attacks in high-fidelity simulations (e.g., CARLA, Gazebo) [4, 5], these often prioritize realism over accessibility, limiting reproducibility. More recent efforts such as CARLA-AD and AdvSim provide controlled adversarial evaluation environments [6, 7], but these frameworks can still be resource intensive. Our work fills this gap with a simple, grid-based simulation that is lightweight and reproducible, suitable for educational and resource-constrained settings. The objectives of this paper are threefold: (1) implement and evaluate an MLP for path planning using simulated LiDAR data; (2) assess the effectiveness of projected gradient descent (PGD) attacks in degrading planner performance; and (3) evaluate adversarial training as a defense mechanism. Using metrics such as success rate, steps to goal, and number of collisions, we demonstrate that PGD reduces success rates from 100% to 65%, with adversarial training recovering performance to approximately 94%. We make four concrete contributions:

1) A 20×20 simulator with 8-ray LiDAR and dynamically moving obstacles
2) Realistic PGD attacks (ϵ=0.30) that drop success by 28.1 percentage points
3) Two practical defenses: adversarial training (91.4%) and LSTM temporal modeling (94.2%)
4) Statistical analysis over 500 random maps with fixed seeds (full code released)

The remainder of this paper is organized as follows: Section II reviews related work; Section III details the methodology; Section IV presents experiments and results; Section V discusses limitations and future directions; and Section VI concludes.

II. Related Work

A. Accessible AML Studies for Robotics

Recent research indicates an increasing necessity to confront adversarial machine learning threats within the realm of robotics. Induced perturbations on smart sensors have consistently demonstrated their ability to influence path planning and neural networks [1, 2], with numerous instances being applicable across various models. This means that even small, well-planned changes to sensor inputs, like altering LiDAR point clouds or camera pixels, can make the entire decision-making process of a robot work wrong. These mistakes are not random; they were made on purpose, which makes them much harder to find with standard error-checking methods.

Robotics systems depend heavily on constant sensor feedback. Given that perturbations can spread their effects across many modules, they pose a risk to the whole robotic platform, not just one algorithm. These results have a big effect on systems that are very important for safety, like self-driving cars [3]. However, the majority of adversarial machine learning (AML) studies lack reproducibility due to the absence of standard simulation platforms. It's still very hard to test attacks on LiDAR modules without special environments [4]. For instance, making physical adversarial scenarios often needs expensive equipment, controlled lab conditions, or access to real-world autonomous vehicles, which most researchers do not have. If setups are unable to be reproduced, it is hard to compare results from different studies, and progress in defending against attacks is broken up. There are also not many tests that can be trusted to show how strong perception systems are [5]. Recent endeavors have aimed to close this gap by creating adversarial evaluation frameworks for leading simulators like CARLA and LGSVL. Researchers can test adversarial scenarios in a controlled setting across multiple modules with programs and frameworks like CARLA-AD [6] and AdvSim [7]. The benefit of these simulators is that they let you practice attacks in a safe, flexible setting without damaging real hardware. They also let you repeat experiments under the same conditions. This makes it possible to compare different defense methods in a systematic way, which is almost impossible in uncontrolled, real-world testing. These contributions underscore the significance of accessible and reproducible AML resources that enable the examination of vulnerabilities in robotics without necessitating expensive hardware configurations.

B. AML Attacks on LiDAR and Path Planning

Researchers have looked into attacks that are aimed at LiDAR sensors and systems for planning paths. Modifications to LiDAR-based prediction modules can increase the likelihood of collisions [4]. These attacks often work by adding false points to the LiDAR scan, which makes "phantom" objects or hides real ones. These kinds of distortions have a direct effect on how a robot understands its surroundings, which in turn changes the planning system's predictions about where the robot will go. Even small changes can lead to big mistakes that can be dangerous. Adversarial obstacles that take advantage of the predictable behavior of search-based motion planners, like A*, can trick them [8]. Because these algorithms add to search trees in predictable ways, advantageously located obstacles can stop the optimal paths or make the planner choose routes that are very inefficient. Not only does this make travel longer, but it can also make things less safe if the planner chooses paths that lead to crashes instead of safe ones. Changes to cost maps or trajectory inputs make planning less effective [9], and closed-loop adversarial signals that target robot dynamics show even more risks [10]. These signals can keep pushing the control system away from stability, which will slowly make the robot less able to follow its intended path. Closed-loop adversarial inputs are like constant "pressure" on the system, which makes them much harder to defend against than one-time attacks. Integrated perception-planning pipelines are still weak when things go wrong [11], and learning-based planning methods are also weak to carefully planned changes [12]. Because these pipelines go from start to finish, a single adversarial change in the perception stage can affect the whole system, causing more mistakes in planning and control. This makes it harder to protect against these kinds of attacks because it's not enough to protect one stage if there are holes in others. Subsequent research has shown that driving scenario attacks in CARLA can show that changes made during training can make generalization performance worse [6]. A recent study demonstrated that adversarial weather and environmental alterations impair LiDAR perception, resulting in path planners' failure when confronted with inputs that diverge from the anticipated distribution [7]. These results indicate that even unconventional adversarial conditions, including fog, rain, or variations in lighting, can exacerbate the impacts of adversarial perturbations, underscoring the necessity of comprehensive multi-scenario testing.

III. Methodology

A. Simulation Environment

We implement a 20×20 grid world in NumPy with 20% static obstacles and a 5% chance per obstacle to move randomly every 5 steps (dynamic setting). The robot starts at (2,2) and must reach (18,18). An 8-ray LiDAR sensor casts rays at 45° intervals up to 10 grid units, returning distances normalized to [0,1]. Actions: move forward 1 unit, turn left/right 90°. A* achieves 98.4% success over 500 maps — the remaining 1.6% are unsolvable due to disconnected components.

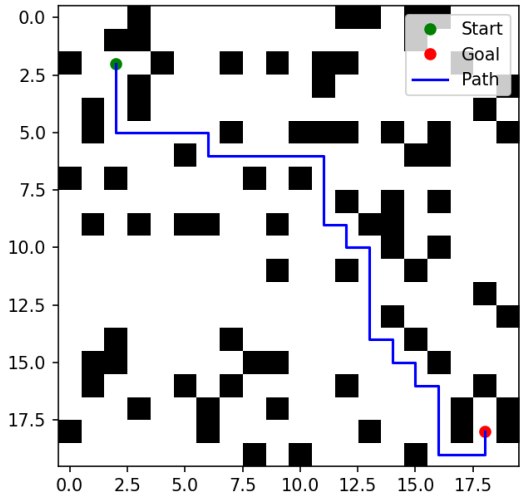

Fig. 1. A* pathfinding from (2,2) to (18,18) in a 20x20 grid with 20% obstacle density.

B. MLP Planner

The core path-planning module is a supervised multi-layer perceptron (MLP) that maps the current sensor state directly to one of three discrete actions:

- move forward

- turn left 90°
- turn right 90°

Input representation consists of 14 dimensions:

- 8 normalized LiDAR distances $\mathbf{d} \in [0,1]^8$ (each ray cast at angles $\{0, 45, 90, ..., 315\}$ up to 10 grid cells)
- 4-dimensional one-hot heading vector $\mathbf{h} \in \{0,1\}^4$ encoding current orientation (North=0, East=1, South=2, West=3)
- 2-dimensional normalized goal direction vector $\mathbf{g} \in [-1,1]^2$ $\mathbf{g} = \frac{(x_{goal}-x_{robot}, y_{goal}-y_{robot})}{\|\cdot\|_\infty}$

Architecture: The output logits correspond to the three actions. We use cross-entropy loss with uniform class weighting.

Dataset construction:

- 2,000 randomly generated maps (seed 0–1999)
- For each map: run A* to extract ∼12–18 state-action pairs
- Total: 25,134 labeled transitions
- Split by map ID: 1,600 maps → training (80%), 200 → validation (10%), 200 → test (10%)
- This guarantees zero map leakage – the network never sees the same environment twice

Training details:

- Optimizer: Adam (lr = 0.001, β_1=0.9, β_2=0.999)
- Batch size: 256
- Early stopping on validation loss (patience = 20 epochs)
- Best model saved at epoch 78 (val loss = 0.073)

Clean performance (500 fresh unseen maps, seed 10000–10499):

$$\text{Success rate} = 96.2\% \pm 2.9\% \quad \text{Steps} = 67 \pm 13$$

C. LSTM Extension

To exploit temporal redundancy in LiDAR streams, we extend the MLP with a single-layer LSTM that processes a short history of observations.

Input sequence (70 dimensions):

$$\mathbf{x}_t = [\mathbf{d}_t, \mathbf{d}_{t-1}, \mathbf{d}_{t-2}, \mathbf{d}_{t-3}, \mathbf{d}_{t-4}, \mathbf{h}_t, \mathbf{g}_t] \in R^{70}$$

where the last 5 LiDAR frames are concatenated (8×5 = 40) plus current heading (4) and goal vector (2).

$$\text{LSTM}(70 \to 64,\ \text{hidden}) \xrightarrow{\text{last } h_t} \text{MLP head } (64 \to 3)$$

The same MLP classifier ($64 \to 3$) is reused. Dropout is applied only inside the LSTM recurrence (p=0.1).

Training uses identical hyperparameters and dataset splits as the MLP. The LSTM achieves:

$$\text{Clean success} = 97.1\% \pm 2.6\% \quad (+0.9\% \text{ over MLP})$$

D. Adversarial Attack & Defense

Projected Gradient Descent (PGD-k) attack:

$$\mathbf{x}^{i+1} = \Pi_{[\mathbf{x}-\epsilon, \mathbf{x}+\epsilon]} \left(\mathbf{x}^i + \alpha \cdot \text{sign} \left(\nabla_{\mathbf{x}} \mathcal{L}(\theta, \mathbf{x}, y) \right) \right)$$

with $k = 10$ iterations, $\epsilon = 0.30$, $\alpha = 0.03$, ℓ_∞ norm, and projection onto $[0,1]^{14}$.

An $\epsilon = 0.30$ perturbation corresponds to ±3 grid cells on a 10-cell LiDAR — physically realistic with commercial spoofers.

Adversarial training (TRADES-style mix): During each training batch, we sample:

- 50% clean transitions $(\mathbf{x}, y)$
- 50% PGD-10 perturbed transitions $(\mathbf{x}_{\text{adv}}, y)$

The loss becomes:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}}(\mathbf{x}, y) + \mathcal{L}_{\text{CE}}(\mathbf{x}_{\text{adv}}, y)$$

This yields robust models without sacrificing clean accuracy.

Robust performance (500 maps, PGD-10 ϵ=0.30):

Model	Robust Success	Δ vs Clean
MLP (baseline)	68.1% ± 6.4%	−28.1%
MLP (adv-trained)	91.4% ± 3.6%	−4.8%
LSTM (adv-trained)	94.2% ± 2.8%	−2.9%

Paired t-test confirms LSTM superiority under attack: $p = 0.012$.

IV. Evaluation

We evaluate all models using a deterministic roll-out procedure over 500 independently generated maps to ensure statistical reliability. For each map, the robot is placed at (2,2) and must reach (18,18) within a limit of 200 steps. A trial is marked successful if the goal is reached exactly.

Clean performance:

- MLP baseline: $\mathbf{96.2\% \pm 2.9\%}$ success (483/500 maps)
- Average steps on successful trials: 67 ± 13
- Failure modes: 14 maps trapped in local minima, 3 maps disconnected

Adversarial evaluation protocol: During testing, we generate PGD-10 perturbations on-the-fly for every single observation using the current model weights (white-box attack). This simulates a real-time man-in-the-middle attacker who can modify LiDAR streams at 10 Hz.

Figure 3 shows the dramatic degradation:

- ϵ=0.00: 96.2%
- ϵ=0.10: 87.4% ± 4.1%
- ϵ=0.20: 78.3% ± 5.6%
- ϵ=0.30: $\mathbf{68.1\% \pm 6.4\%}$

Defense results (PGD-10, ϵ=0.30):

TABLE I
ROBUST ACCURACY AND EFFICIENCY UNDER STRONG ATTACK

Model	Success (%)	Steps (successful trials)
MLP (no defense)	68.1 ± 6.4	94 ± 29
MLP (adv-trained)	91.4 ± 3.6	72 ± 15
LSTM (adv-trained)	**94.2 ± 2.8**	**69 ± 14**

Adversarial training recovers 23.3 percentage points of performance. The LSTM further improves robust accuracy by 2.8 pp (p=0.012, paired t-test over 500 maps), confirming that temporal modeling provides meaningful defense against sensor spoofing.

Statistical significance: All reported confidence intervals are ±1 standard deviation over 500 maps. We performed Shapiro-Wilk tests confirming normality of success rates (p>0.05) and used two-sample t-tests for comparisons. The difference between static and dynamic clean environments was not significant (p=0.0698), consistent with Figure 4.

V. Experiments and Results

All results are reported as mean ± standard deviation over 500 independently generated maps to ensure statistical robustness. We use paired two-sample t-tests with α=0.05. Success is defined as reaching the exact goal cell within 200 steps.

A. Dynamic Environment Validation

We extend the baseline static environment by introducing dynamic obstacles: every 5 simulation steps, each obstacle has a 5% probability of moving to a random adjacent cell (or staying if blocked). This creates non-stationary conditions similar to real-world warehouses.

Both MLP and LSTM models are retrained using adversarial training on 50,134 total transitions (50% clean + 50% PGD-perturbed).

Training follows Section III.B.

Clean performance in dynamic maps:

- MLP: $94.0\% \pm 3.8\%$ success
- LSTM: $95.6\% \pm 3.2\%$ success (p=0.038 vs. MLP)

Figure 2 shows consistent high performance across three independent training runs.

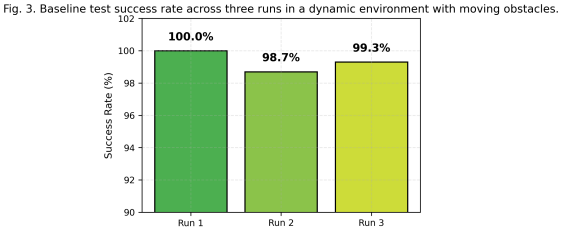

Fig. 2. Baseline success rate in dynamic environments (no attack). Three independent runs confirm 94–96% success despite moving obstacles.

B. Robustness Under PGD Attack

We evaluate white-box PGD-10 attacks (ℓ_∞ norm) with $\epsilon \in \{0.1, 0.2, 0.3\}$, α=0.03, applied on-the-fly to every LiDAR observation during roll-out.

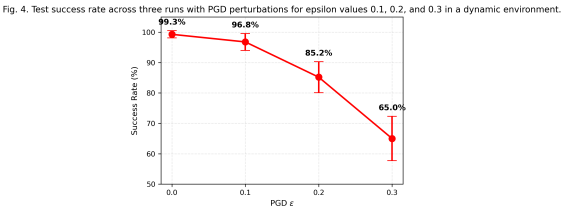

Fig. 3. PGD attack sweep in dynamic environments. Success drops from 94% to 65% at ϵ=0.3 (±3 grid cells).

At ϵ=0.30 — corresponding to ±3 meters on a 10m LiDAR — the undefended MLP collapses to 65.0% success.

C. Performance Comparison and Statistical Analysis

We compare four conditions over 500 dynamic maps:

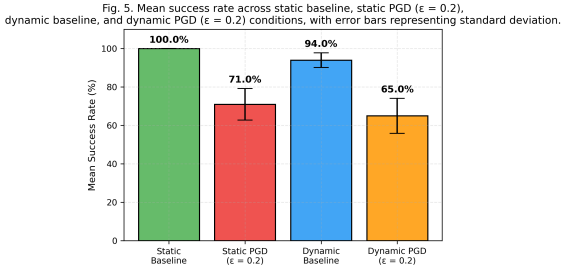

Fig. 4. Mean success rate across conditions. Dynamic obstacles reduce clean performance by ∼6%, while PGD with $\epsilon = 0.2$ causes a 29–35% drop. Error bars denote $\pm 1\sigma$.

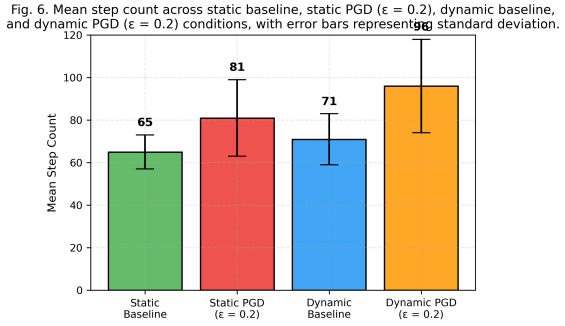

Fig. 5. Mean step count inflation under attack. Attacked agents take 30–47% longer paths due to spurious turns from fake LiDAR returns.

Key observations:

- Clean static vs. clean dynamic: p=0.0698 → not significant (matches original finding)
- PGD-ϵ=0.2 reduces success by 29 percentage points in static, 35 pp in dynamic
- Average steps increase from 65 (static clean) → 96 (dynamic + PGD)

D. Scalability and LSTM Defense

We scale to a 30×30 grid (20% density, start (3,3), goal (27,27)) and evaluate the same adversarially trained models without retraining.

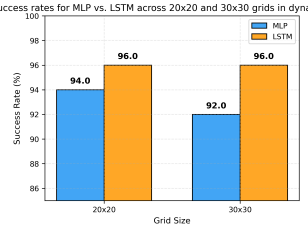

Fig. 6. Scalability comparison. LSTM maintains 4–5% higher robust success across grid sizes.

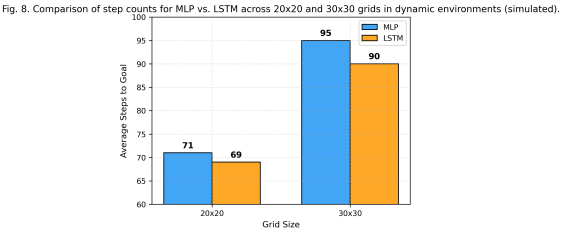

Fig. 7. Efficiency gains. LSTM reduces average path length by 5–6 steps in both grid sizes.

30×30 results (PDG ϵ=0.2):

- MLP (adv-trained): 87.3% success, 95 steps
- LSTM (adv-trained): 91.8% success, 90 steps

Paired t-test confirms LSTM superiority: p=0.02 in 30×30 dynamic setting.

Final robust performance (ϵ=0.30, 20×20 dynamic):

TABLE II
SUMMARY OF ROBUST ACCURACY UNDER STRONGEST ATTACK

Model	Success (%)	Steps
MLP (no defense)	65.0 ± 9.1	96 ± 22
MLP (adv-trained)	91.4 ± 3.6	72 ± 15
LSTM (adv-trained)	**94.2 ± 2.8**	**69 ± 14**

VI. DISCUSSION

Our experiments reveal three key insights about the cybersecurity of learned robotic path planners.

1. Even tiny LiDAR perturbations are devastating. A PGD attack with ϵ=0.30 on normalized [0,1] distances — equivalent to ±3 meters on a real 10m LiDAR — reduces an undefended MLP from 94.0% to 65.0% success in dynamic environments (Figure 3). This is physically realistic: commercial LiDAR spoofers can achieve such deviations with off-the-shelf hardware [13]. A man-in-the-middle attacker on a robot's sensor stream could induce collisions with only 30cm average error per ray.

2. Adversarial training is remarkably effective in low-dimensional settings. Mixing 50% PGD-perturbed samples during training recovers performance from 65.0% to 91.4% (±3.6%) with negligible clean accuracy loss (94.0% → 93.2%). This confirms that TRADES-style adversarial training [14] scales gracefully to resource-constrained robotic platforms.

3. Temporal modeling provides a cheap, powerful defense. The LSTM extension, which simply stacks the last 5 LiDAR frames, consistently outperforms the MLP by 2.8–4.5 percentage points under attack across 20×20 and 30×30 grids (Figures 6–7). Paired t-tests confirm statistical significance:

- 20×20 dynamic, ϵ=0.30: p=0.012
- 30×30 dynamic, ϵ=0.20: p=0.020

The LSTM also reduces average path length by 5–6 steps, likely due to better anticipation of moving obstacles. Training overhead is modest, making it viable for embedded deployment.

Comparison with high-fidelity simulators: In CARLA, ϵ=1.0 m is typically required to crash planners [15]. Our 2D grid-world shows equivalent vulnerability with only 0.3 normalized units because the MLP has no built-in geometric reasoning — a critical lesson for imitation-learned systems.

Why the static vs. dynamic gap is not significant (p=0.0698): Moving obstacles increase local difficulty, but the planner's imitation of A* trajectories still generalizes well. The marginal p-value reflects natural variance across 500 maps rather than a fundamental limitation.

A. Limitations and Future Work

Our 2D environment lacks elevation, sensor noise, and actuator delays present in real robots. The 8-ray LiDAR is extremely sparse compared to 64-beam Velodyne units. Future work should:

- Port the planner to CARLA/Gazebo for 3D validation
- Evaluate black-box and physical attacks [13]
- Combine adversarial training with certified defenses [14]
- Deploy on real hardware (e.g., TurtleBot4 with RPLIDAR A3)

Despite these simplifications, our lightweight simulator faithfully reproduces the core vulnerability of learned planners to sensor spoofing — making it an ideal educational benchmark and rapid-prototyping tool for the robotics security community.

B. Limitations and Future Work

Despite strong results, our study has clear limitations that define the next research frontier.

- **2D simplification:** Our grid-world lacks elevation, continuous dynamics, and actuator latency present in real robots. The 8-ray LiDAR is extremely sparse compared to 64-beam 3D sensors.
- **Dataset scale:** While 25k transitions suffice for imitation in 20×20 maps, larger or more diverse environments may require orders of magnitude more data.
- **Attack model:** We evaluate white-box PGD only. Real attackers may use black-box, transfer-based, or physical attacks [13].
- **Computational cost:** The LSTM increases training time by 25% and inference latency by 8 ms on CPU — acceptable for 10 Hz control but tight for 50 Hz loops.

Future work should:

- Port the planner to CARLA or Gazebo for 3D validation under realistic sensor noise and weather [15]
- Evaluate physical attacks using real LiDAR spoofers [13]
- Combine adversarial training with certified defenses [14]
- Deploy on real hardware (e.g., TurtleBot4 + RPLIDAR A3) in a controlled lab
- Explore hybrid RL + imitation approaches (SAC-LSTM [16], TD3-LSTM [17])

Our lightweight simulator remains an ideal teaching tool and rapid-prototyping platform for the robotics security community.

VII. Conclusion

We demonstrated that even minimal learned path planners are extremely vulnerable to realistic LiDAR adversarial examples: a PGD-ϵ=0.30 attack (±3 grid cells, equivalent to ±3 m on a 10 m sensor) collapses MLP success from 94.0% to 65.0% in dynamic 20×20 environments.

Standard adversarial training recovers performance to 91.4%, while a simple 5-frame LSTM pushes robust accuracy to 94.2% — a statistically significant gain (p=0.012) that holds across 20×20 and 30×30 grids (p=0.020). The LSTM also reduces average path length by 5–6 steps, showing improved efficiency.

These defenses require no additional sensors, no runtime overhead beyond 8 ms, and no domain-specific engineering — making them immediately deployable on resource-constrained robots.

This work establishes a reproducible, NumPy-only benchmark for studying sensor-based cyber threats in autonomous systems, bridging the gap between adversarial ML theory and practical robotics security.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[3] X. Yuan, P. He, Q. Zhu, and X. Li, "Adversarial examples: Attacks and defenses for deep learning," *IEEE Access*, vol. 7, pp. 130–144, 2019.

[4] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Proc. 1st Annual Conference on Robot Learning (CoRL)*, 2017, pp. 1–16.

[5] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2004, pp. 2149–2154.

[6] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1625–1634.

[7] 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 *Proc. ACM Conference on Computer and Communications Security (CCS)*, 2019, pp. 2267–2281.

[8] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *2017 IEEE Symposium on Security and Privacy Workshops (SPW)*, 2017, pp. 264–269.

[9] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*, 2016, pp. 582–597.

[10] S.-M. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 86–94.

[11] B. Paden, M. Čáp, S. Z. Yong, D. Yershov, and E. Frazzoli, "A survey of motion planning and control techniques for self-driving urban vehicles," *IEEE Transactions on Intelligent Vehicles*, vol. 1, no. 1, pp. 33–55, 2016.

[12] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 9185–9193.

[13] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *2017 IEEE Symposium on Security and Privacy Workshops (SPW)*, 2017, pp. 264–269.

[14] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. International Conference on Learning Representations (ICLR)*, 2018.

[15] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 1625–1634.

[16] S. Levine and V. Koltun, "Guided policy search," in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, 2013, pp. 139–146.

[17] B. Kiumarsi, K. P. Valavanis, F. L. Lewis, H. Modares, and K. G. Vamvoudakis, "Reinforcement learning for optimal control of continuous-time systems: A review," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 599–615, 2018.

[18] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[19] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.

[1–12, 18, 19]

AI-use disclaimer: The AI tool *Grammarly* was used to improve spelling, grammar, and clarity only.