Analysing Adversarial Threats to Rule-Based Local-Planning Algorithms for Autonomous Driving

Andrew Roberts
FinEst Centre for Smart Cities
Tallinn University of Technology
Andrew.Roberts@taltech.ee

Mohsen Malayjerdi
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
Mohsen.Malayjerdi@taltech.ee

Mauro Bellone
FinEst Centre for Smart Cities
Tallinn University of Technology
mauro.bellone@taltech.ee

Olaf Maennel
School of Computer and Mathematical Sciences
The University of Adelaide
olaf.maennel@adelaide.edu.au

Ehsan Malayjerdi
Department of Mechanical and Industrial Engineering
Tallinn University of Technology
ehsan.malayjerdi@taltech.ee

***Abstract*—The safety and security of navigation and planning algorithms are essential for the adoption of autonomous driving in real-world operational environments. Adversarial threats to local-planning algorithms are a developing field. Attacks have primarily been targeted at trajectory prediction algorithms which are used by the autonomous vehicle to predict the motion of ego vehicles and other environmental objects to calculate a safe planning route. This work extends the attack surface to focus on a rule-based local-planning algorithm, specifically focusing on the planning cost-based function, which is used to estimate the safest and most efficient route. Targeting this algorithm, which is used in a real-world, operational autonomous vehicle program, we devise two attacks; 1) deviation to the lateral and longitudinal pose values, and 2) time-delay of the sensed-data input messages to the local-planning nodes. Using a low-fidelity simulation testing environment, we conduct a sensitivity analysis using multiple deviation range values and time-delay duration. We find that the impact of adversarial attack cases is visible in the rate of failure to complete the mission and in the occurrence of safety violations. The cost-function is sensitive to deviations in lateral and longitudinal pose and higher duration of message delay. The result of the sensitivity analysis suggests minor deviations of the pose (lateral, longitudinal) values as an optimal range for the attackers search space. Options for mitigating such attacks are that the AV should run a concurrent process executing a concurrent planning instance for redundancy.**

I. Introduction

Navigation and planning algorithms are essential for autonomous driving (AD). For the self-driving vehicle to navigate the road environment, the navigation and path-planning algorithm must calculate a route that ensures safety for the passenger and external environmental actors (pedestrians, other vehicles and road users, etc.) and achievement of the journey (mission). Initial studies of navigation and path planning algorithms for AD have shown them to be vulnerable to adversarial attacks that introduce uncertainties into the route calculation, which causes downstream effects on the safe behavioural control of the AV [2], [17], [3]. To improve the reliability of navigation and planning algorithms, they need to be further tested for uncertainties, and these methods are incorporated into the architecture of autonomous driving.

There are a few studies that focus on adversarial attacks on local-planning. These studies target machine learning algorithms for local-planning modules such as trajectory prediction (Trajectron++, Agentformer and GRIP++) [2], [17], [3]. The predominant threat model adopted, focuses on developing methods and tools of adversarial learning to understand the trajectory prediction model of the target AV and then either crafting malicious sensor data input or training other ego AVs in the driving environment to interfere with the target AVs predicted trajectory [15], [2], [17], [3]. The required result of a successful adversarial attack is to cause the target AV to generate a trajectory that is unsafe, inefficient, or uncomfortable for passengers. In this work, we expand on the target of attacks to a rule-based algorithm for local-planning, and focus on the trajectory generation and estimation of an AV. Our justification for focusing on rule-based algorithms is that, whilst AI approximate reasoning algorithms seem to be highly promising for the near future, an impediment to current adoption is the lack of feedback in real-world driving scenarios [5]. Rule-based algorithms for path-planning in robot navigation and AD are well-established, and more ubiquitous in real-world deployments.

A rule-based local-planning algorithm uses a cost function to estimate the least-cost path. The cost function takes input from immediately sensed-data; current pose, velocity etc.. The cost estimation is based on a calculation of factors such as; lateral collision, longitudinal collision, lane transition, central deviation etc., and weighting is given to these factors based on criteria such as safety and efficiency. By interpreting the cost-function, used for trajectory generation and estimation, as part of local-planning, an adversarial attack can be crafted which affects the downstream behavioural control whose decisions impact the safe driving state of the AV.

The main idea of this paper is that the white-box knowledge of the cost estimation function of the rule-based local planning

Symposium on Vehicles Security and Privacy (VehicleSec) 2023
27 February 2023, San Diego, CA, USA
ISBN 1-891562-88-6
https://dx.doi.org/10.14722/vehiclesec.2023.23086
www.ndss-symposium.org

algorithm can be used to craft adversarial attacks by manipulating factors inherent to the cost function. Evaluating white-box generated attacks enable an understanding of the level of stealth of the adversarial threat, and whether adversarial manipulation by the cyber attack can be distinguished from noise. Furthermore, these attacks will enable evaluation and assessment of the optimisation of the algorithm to uncertainties and the quality of decision-making.

The key questions this study engages are the following:

1) What is the sensitivity of the cost function to adversarial data manipulation of key driving parameters?
2) How can an adversarial attack hide in the cost function from detection?
3) What optimisations of the rule-based algorithm can be considered to mitigate against adversarial data manipulation?

The problem area of this research, is centred on a local-planning algorithm, open-planner 2.5, which is used in an AV shuttle program that operates in real-world road conditions in Europe [7]. As with the open-source software community, development of vulnerability research and testing methods proliferate across the ecosystem and are utilised and innovated for diverse platforms. The aim of this study is to focus on the vulnerability of the local-planning function of autonomous driving and provide direction and guidance to the autonomous driving security community to develop vulnerability testing on diverse planners and algorithms. In a broader sense, this research aims to understand how AD algorithms used in real-world AD programs can be tested for adversarial threats and validated to improve assurance for real-world operational driving.

II. Rule-Based Local-Planning Algorithm

A. Open Planner 2.5 Local-Planner Overview

For the AV to plan a mission, firstly, a global planner generates a global reference path using a vector (road network) map. The function of the global planner is to stipulate the starting position and goal position of the mission on the road map. How to achieve this mission, for the AV to navigate from the starting position to the mission goal, through a smooth, obstacle free trajectory is the function of the local-planner. The local-planner consists of several modules (see Figure 1); trajectory generation, trajectory evaluation, intention and trajectory estimator, object-tracker and behavior selection (decision making) [7]. The trajectory generation module generates alternative tracks parallel to the main path defined by the global planner. These tracks are named rollouts. The trajectory evaluation module assesses all possible rollouts and the data input from sensed-data of the AV and makes a cost estimation. The behaviour selector will lead the AV to motion on a rollout based on the least-cost.

Table I displays the input and outputs of each of the local-planning modules (Note. intention and trajectory estimator and object-tracker are not visible as they are not within the scope of this study).

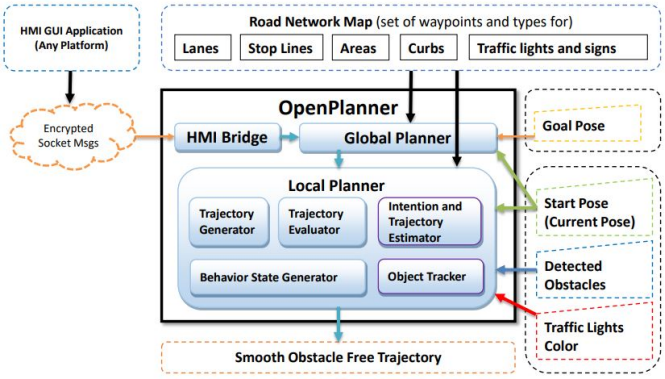

Fig. 1: OpenPlanner 2.5 Architecture [7]

TABLE I: Local-Planning Module

Node	Input	Output
Trajectory Generator	Initial_Pose	Local Trajectories
	Current_Pose	
	Current_Velocity	
	Lane_Waypoints_Array	
Trajectory Evaluator	Current_Velocity	Local Trajectory_Cost
	Current_Pose	
	Local_Trajectories	
	Lane_Waypoints_Array	
	Predicted_Objects	
	Current_Global_Local_IDS	
Behavioural Selector	Current_Velocity	Current_Behaviour
	Current_Pose	
	Local_Trajectory_Cost	
	Local_Weighted_Trajectories	

B. Local Planning Cost Function

The local motion planning algorithm generates a trajectory (or a set of control commands for the AV) by minimizing a cost function, within a workspace, that includes a set of design parameters. The cost function constitutes the rules for motion-planning which inform the decision-making for autonomous driving.

The cost function is built on five factors and calculated in the following Eq. 1:

$$C = \begin{bmatrix} w_{cent} \\ w_{trans} \\ w_{longColl} \\ w_{latColl} \\ w_{vis} \end{bmatrix} \cdot \begin{bmatrix} C_{cent} \\ C_{trans} \\ C_{longColl} \\ C_{latColl} \\ C_{vis} \end{bmatrix}^T \quad (1)$$

where, C_{cent} is the cost associated to the central trajectory and is designed to keep the vehicle in the central trajectory; C_{trans} is the transition cost that prevents the vehicle from jumping between rollouts; $C_{longColl}$ and $C_{latColl}$ are the cost of the longitudinal and lateral collision respectively, and finally C_{vis} is the weight associated to the visibility [12]. Each of these costs are weighted by their respective weighting factors w_i [6].

III. Threat Model

The attack targets the local planning cost function, with the aim of inducing the trajectory evaluation to choose a motion-planning route that is not optimal for safety, functionality of the driving mission and comfort of the passengers. To achieve this, the most direct mechanism to impact the cost function is to manipulate, with adversarial data, the sensed-data input that is inherent to local-planning. The Current_Pose data is the optimal target for this as it is the primary sensed-data for localisation of the vehicle, containing the longitudinal, lateral positioning and orientation of the AV. Whilst altering the pose data of the vehicle has previously been conducted in other studies [2], [17], [3], [15], in our attack we aim to explore the sensitivity of our cost function to data manipulations and conducting the attack during specific time-intervals.

For the threat model used in our study, we assume that the attacker has access to the internal network of the AV and is able to listen to control message communications and collect data. This could be achieved through supply-chain compromise of a library in the control software, insider threat actor, or many of the vulnerabilities in existing communication frameworks for autonomous systems such as the robotic operating system (ROS) [8]. Given the attacker has access to the internal network, the question arises, why not change the Lane_ID or a driving parameter which would be more simplistic and direct? We view these attacks as overt in nature and likely to be detected, the compelling nature of adversarial data manipulation is that the attack is difficult for AV safety engineers to interpret between noise and an explicit cyber threat. Another consideration are the external interfaces of the vehicle localisation sensing, which generates the pose data. It is a possibility that the pose data can be manipulated by an external attack in the form of GPS spoofing or an adversarial LiDAR, dependent on the sensor configuration used for the localisation of the vehicle. The study focused on the vulnerability of the planner and its search space, considering localisation. We considered internal attacks to be important due to the increase in attacks through software and hardware supply-chains, and therefore the scope of the attacks within the study highlighted this area.

A. *Attack Case 1: Position Offset Attack*

The attacker creates a spoofed ROS topic which is able to deliver malicious input data of the Current_Pose (longitude, latitude, and velocity) to all the nodes of the local planning module. The data manipulation is injected online/dynamically during the critical overtaking manoeuvre involving the AV and NPC (Non-playable character). Figure 2 displays the critical driving scenario and the time frames in which the manipulated Current_Pose data is injected into the local planning pipeline cost estimation. The red dashed lines in Figure 2 represent the roll-outs, and the green highlighted, denoting the selected motion-path.

For the manipulation of the Current_Pose data, we introduce a deviation to lateral and longitudinal pose.

For the lateral pose data, the sensitivity deviation introduced was structured as follows:

- Attack Case 1a: 0.16%

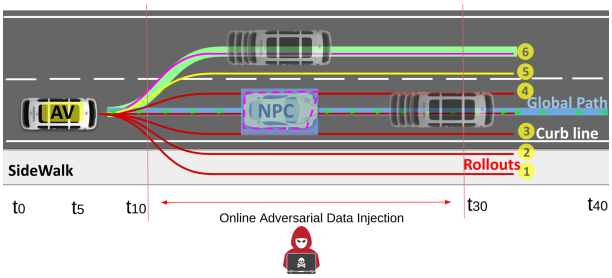

Fig. 2: Threat Model

- Attack Case 1b: 0.33%
- Attack Case 1c: 0.5%

In designing the range of deviation, we considered state-of-the-art attacks such as AdvDO attack [2], which noted two requirements for developing adversarial threats to planning algorithms:

1) Malicious data input needs to be feasible to the real, physical constraints of the vehicle [2].
2) Malicious data input of the local-planning algorithm should be close to the nominal trajectory [2].

Therefore, we chose a range from a slight perturbation of pose to a 1m deviation.

The longitudinal pose data sensitivity deviation range was structured as follows:

- Attack Case 1d: 0.33%
- Attack Case 1e: 0.66%
- Attack Case 1f: 1.00%

This range is the same as the longitudinal deviation. The difference in percentage comes from the difference in coordinate values of lateral and longitude. The lateral value is almost double those of the longitudinal, and therefore the percentage is doubled.

B. *Attack Case 2: Message Time-Delay*

For the second attack case, we inserted a time-delay into the messages of the Current_Pose topic communicating to the nodes of the local planning module.

We introduced a message delay when the AV passes 2m in front of the NPC (from the centre) in the lateral direction. We introduce 3 different time delays in the message:

- Attack Case 2a: 0.3 seconds
- Attack Case 2b: 0.6 seconds
- Attack Case 2c: 1.0 seconds

The message frequency is approximately 50hz, so this is a message every 20 milliseconds. We chose the above range of deviation of time-delay as it enabled a spectrum of a message from the delay from approximately 15, to 50 messages.

IV. Experimental Setup

A. Test Environment and Configuration

In terms of conducting such experiments, simulation is the best method among all testing methods for AVs. To accelerate the testing, we bypassed the sensing and detection nodes of the algorithm and focused on the planning part by utilizing the low-fidelity simulation feature provided by Autoware.ai and Openplanner. The low-fidelity simulation uses the open-planner 2.5 control algorithm. It provides simulated localization and detection data for the planning nodes and receives the actuation commands to simulate the AV kinematics. This process runs faster due to the low-detail environment required for the simulation and the lack of the process to simulate the sensors. Figure 3 displays the different frames of an overtaking simulation in the simulator.

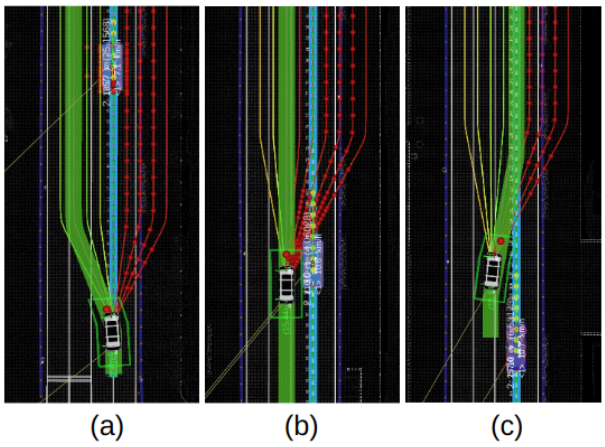

Fig. 3: Example of an overtaking simulation in the low-fidelity simulator, a) starting point of the overtaking b) middle of the mission, AV is on the opposite lane reaching the NPC c) AV cuts in

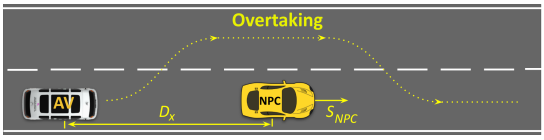

Fig. 4: Target scenario, Dx and S_{NPC}, define the initial relative distance to the NPC and the constant NPC speed in the scenario

1) Target Mission: Overtaking is one of the most challenging maneuvers for Avs [10]. In this research, we selected this operation as the target scenario for studying the planning algorithm under the cyber-attack. The scenario parameters in Figure 4 are listed in Table II.

TABLE II: Target scenarios definition

Actor	Speed (m/s)	$D_x(m)$	Goal
AV	[0:6]	0	overtake the NPC safely
NPC	3	25	keep moving

2) Safety Evaluation Test: To assess the safety and reliability of the planning algorithm in normal conditions (no attack), we ran the scenario simulation 300 times to reach a meaningful statistical population. Then, the planning algorithm behavior in each case was evaluated with the local-planner performance evaluation criteria (explained in the next section).

3) Attack Test Cases: Finally, the platform was used to simulate the proposed adversarial data manipulations and time-delay messaging, during the overtaking mission and monitor the algorithm's behavior. For each attack case, we ran the simulation (with attack) 100 times. Overall, 900 simulations were conducted for all attack cases.

B. Evaluation Criteria

For the evaluation, we used previously established safety criterion [11] with evaluation criteria recommended by SafeBench, a benchmarking framework for safety evaluation of AD algorithms for critical driving scenarios [16]. Figure III displays the metrics used for the performance evaluation.

TABLE III: Local-Planner Performance Evaluation Criteria

Condition	Data Label	Description	Metric
Safety Violation	**V**		
Succeed	Suce	AV Successful complete the mission	Pass/Fail
Not Finished	NotF	Failure to finish the mission	Pass/Fail
Distance-to -Collision	DTC	Violation of the safe distance between AV and NPC	AV within 0.5m of other vehicle
Break on Driving Lane	BrD	AV initiates emergency break on driving lane	Pass/Fail
Break on Passing Lane	BrP	AV initiates emergency break on passing lane	Pass/Fail
Collision	Col	AV collides with NPC	Pass/Fail
Functionality			
Avg. time spent to complete route	TS	The average time taken to complete the mission	seconds
Comfort			
Avg. Acceleration	ACC	Average acceleration of the AV	m/s
Avg. Steering Angle	YV	Steering angle of the AV	degrees
Freq. of Lane Invasion	LI	The number of times the AV transitions to another rollout	numeric

V. Results

After running 1200 simulations, all recorded data including the AV and the NPC position and orientation were processed to assess the simulations based on the evaluation criteria. We also visualized the recorded data to study the violation and their cause in each simulation as shown in Figure 5. Figure 5.a represents a safety run completed successfully. Next, (b) and (c) display lateral and longitudinal attack cases which experienced brake and collision safety violations respectively. Finally, (d) shows a message time delay attack which is finished by a collision. The asterisk signs in the AV trajectory show the point where the Openplanner changes the rollout. Overall, all the safety violation results for the whole experiment are presented in Figure 6.

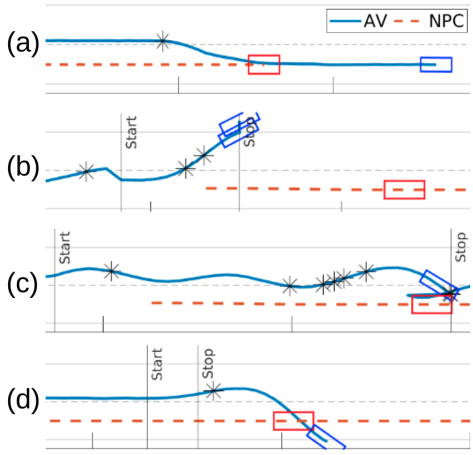

Fig. 5: 2D representation of the simulation of each test group. a) a successful safety test, b) a lateral attack case that led to a brake violation, c) a longitudinal attack case that experienced a collision, and d) a message time delay that causes a collision. for the attack cases a vertical line shows the start and stop point of the attack

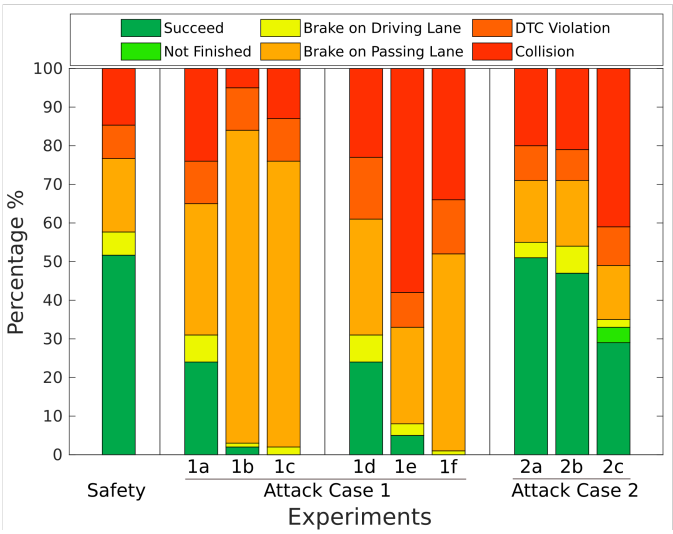

Fig. 6: All simulation result based on the proposed safety criteria

For each of the attack test cases, we saw an increase in safety violations of the AV compared to the normal safety test case experiment. As the value of the deviation for lateral and longitudinal values increased the number of successful mission completions decreased. Although marginal, the greater number of safety violations for the attacks on the Current_Pose data were observed in the lateral deviations. Given the importance of lateral positioning to the overtaking manoeuvre, this can be understood as any deviation increases the complexity of executing the overtaking manoeuvre. In the 1f attack test case, the highest value longitudinal change (approximately 1 meter) led to a crash with curbside and not able to continue the mission. This event was reported as a braking safety violation.

The time-delay messaging attack test case saw the only result for mission not finished metric. Furthermore, the greater the delay of the Current_Pose data reaching the local-planning nodes, the increased likelihood that a safety violation will occur, and in the case of our experiments, the greater the likelihood of a the most serious safety violation, collision.

Table IV demonstrates the results of the safety test according to the performance evaluation criteria. The level of safety violations are reflective of an algorithm which is in development and being optimised for critical driving scenarios such as overtaking.

TABLE IV: Summary of the Safety Simulation

$Num.$	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
300	4.6%	8.6%	19%	6%	0%	51.6%

	TS	ACC	YV	LI
mean	29.1	0.4	3.8	7.1
STD	6.7	0.2	2.2	4.6
min	21.9	0.2	1.8	2
max	42.3	1.3	21.7	25

Table V shows that for each deviation there is a high number of safety violations in comparison to the safety test case results. In regards to the sensitivity analysis, a smaller deviation of around 20 to 25 cm can achieve the result that the local-planning algorithm is only successful in generating a trajectory that completes the mission in 24% of the total test set. Furthermore, a small deviation in the lateral pose, can achieve a higher number of collisions with an ego vehicle. It may also be seen from the lane invasion and steering angle results that small deviations to lateral pose result in a fluctuation of the cost of different rollouts which cause greater lane transitions as the cost function causes the AV to choose a route based on minimum cost. The higher deviation results in a higher occurrence of breaking activity and hitting the curb. Furthermore, the higher deviation results in the AV being stuck in the passing lane, this is due the dramatic change in lateral pose. The 1 meter deviation attack case results in 0% success of finishing the mission.

Table VI results of the longitudinal deviations also display a high number of safety violations in comparison to the safety test case results. Collision safety violation is highest for the longitudinal deviation attack. This can be reasoned as the longitudinal deviation does not experience the same high volume of breaking passing lane safety violations, where the vehicle gets stuck, as seen with the lateral pose deviation. The higher deviation of longitudinal pose, results in increased acceleration and this causes sharp breaking. This is indicated with the 1f result, the 1 meter deviation attack case, which displays a higher instance of breaking safety violation. The 1 meter deviation attack case results in 0% success of finishing the mission.

Table VII demonstrates the shorter delay of local pose data has minimal impact on the success of the mission and safety violations. As the time duration of the message delay is increased the impact to the reliability of the local-planning algorithm is higher. Test 2c, which is the delay of Current_Pose data of 1.0 second, shows considerable increases in collisions and decreases in the likelihood of the success of the mission.

TABLE V: Summary of the Attack Case 1: Position Offset Attack Simulation

Case	$Num.$	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
1a	100	24%	11%	34%	7%	0%	24%
1b	100	5%	11%	81%	1%	0%	2%
1c	100	13%	11%	74%	2%	0%	0%

1a		TS	ACC	YV	LI
	mean	35.3	0.4	9	7.5
	STD	7.4	0.2	7.5	5.4
	min	21.9	0.2	1.9	1
	max	42.4	1	23	23

1b		TS	ACC	YV	LI
	mean	41.4	0.4	9.5	4.8
	STD	3.5	0.1	4.4	3
	min	22.1	0.2	3.1	1
	max	42.4	1.2	23.7	21

1c		TS	ACC	YV	LI
	mean	41.7	0.4	7.8	4.7
	STD	1.7	0.1	1.2	2.7
	min	32	0.3	4.3	1
	max	42.3	1	9.8	15

TABLE VI: Summary of the Attack Case 1: Position Offset Longitudinal Deviation Simulation

Case	$Num.$	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
1d	100	23%	16%	30%	7%	0%	24%
1e	100	58%	9%	25%	3%	0%	5%
1f	100	34%	14%	51%	1%	0%	0%

1d		TS	ACC	YV	LI
	mean	33.8	0.5	5.7	9.1
	STD	7.6	0.3	4.9	5.4
	min	18.1	0.2	1.7	2
	max	43.2	1.4	23	27

1e		TS	ACC	YV	LI
	mean	32.2	0.6	6.7	10.5
	STD	9.5	0.2	3.2	5
	min	17.8	0.2	1.9	2
	max	43.2	1.1	20.5	25

1f		TS	ACC	YV	LI
	mean	32.2	0.7	5.9	11.3
	STD	7.9	0.2	2.5	4.7
	min	18	0.3	2.7	2
	max	43.2	1.4	22.1	26

The time-delay of the pose data to the local-planning nodes results in a loss of localisation and the greater delay the greater impact on the cost calculation which in turn causes uncertainty for the behaviour selector/decision-making.

TABLE VII: Summary of the Attack Case 2: White-Box Delay Simulation

Case	$Num.$	V_{Col}	V_{DTC}	V_{BrP}	V_{BrD}	V_{NotF}	V_{Suce}
2a	100	20%	9%	16%	4%	0%	51%
2b	100	21%	8%	17%	7%	0%	47%
2c	100	41%	10%	14%	2%	4%	29%

2a		TS	ACC	YV	LI
	mean	29.3	0.4	4.2	7.6
	STD	8.1	0.2	2.2	5.4
	min	18.1	0.2	1.8	2
	max	53	1.1	16.7	24

2b		TS	ACC	YV	LI
	mean	30.6	0.4	4.8	7.8
	STD	8.6	0.3	3.7	4.8
	min	22.9	0.2	1.8	2
	max	58	1.1	23.8	21

2c		TS	ACC	YV	LI
	mean	32.9	0.4	7	8.3
	STD	9.6	0.3	5.2	5
	min	13	0.2	1.1	0
	max	58.2	1.3	22.9	23

VI. Discussion

The results of the test simulations demonstrated that the cost function is sensitive to minor deviations of both the lateral and longitudinal pose. The success rate of the mission is visibly diminished when adding adversarial data manipulations to the sensed-data input. The higher the deviation, the higher the likelihood of mission failure. The minor deviation attacks, where the deviation is a range of 20 to 25cm offer a good starting point to mutate adversarial data for further attacks based on this range. Whilst the higher range attacks conducted in our experiments showed a higher rate of mission failure, a deviation of 1 meter can be seen a noisy enough to be observable. We also noticed such behaviour in a real-world AV shuttle [14] and a manual emergency break had to be enacted to prevent an emergency.

The time-delay attack demonstrated that minor delays cause minimal impact on the success of the mission and the occurrence of safety violations. Delays in sensed-data input flowing to the local-planning modules of greater than 1 second increase the rate of mission failure and safety violations. Given that 1 message is broadcast every 20 milliseconds, 1 second represents around 50 messages, and a delay of this magnitude is also likely to be more observable.

For the attack to hide in the cost function, investigating mutations for minor deviations of lateral and longitudinal values in the range of 20 to 30 cm, offer an optimal target range.

Mitigation of the adversarial deviation and time-delay attack could include the implementation of a redundant driver. This means that the AV should run a concurrent process executing a concurrent planning instance. If the redundant

driver and the actual driving algorithm give different results, then this could indicate that an attack might be happening. In such a case, the AV could either stop safely awaiting for human intervention or switch to the redundant driver to complete its mission. The development of the architecture for a redundant driving integrity checking function also needs to consider isolation from the primary driving function so that an attacker cannot also compromise both.

VII. Related Work

As safety validation of AD algorithms is a critical field for the adoption of AD in real-world environments, there is a focus on testing the reliability of trajectory prediction and generation to adversarial driving actors in the road environment. Wang et al. [15], Abeysirigoonawardena, Dudek & Shkurti [1], Chen et al. [4], Klischat et al. [9], and O'Kelly et al. [13] use simulation environments to develop adversarial trained NPCs whose driving actions cause safety violations of the trajectory prediction of the targeted AV. These simulations are focused on safety validation and are not focused on the exploitation of the algorithm by adversarial threat actors, however, their methods in generating adversarial examples and target parameters and data values are of great use in developing adversarial cyber threats.

On a practical level, involving the real-world operation of AVs, there are few research studies into the robustness of planning and navigation algorithms to adversarial threats. Prominent among them are Zhang et al. [17], Cao et al. [3] and Cao et al. [2]. These studies focus on the robustness of the trajectory prediction, the ability of the AV to predict the trajectory of another ego vehicle or environmental object (pedestrian, animals etc.) and make driving decisions accordingly. The attacks in these studies are targeted at deep-neural networks (DNNs), and therefore focus on adversarial learning to develop robust adversarial trajectories. In relation to our work, the observations on ranges for deviation of lateral and longitudinal values and the considerations for crafting adversarial data were useful in developing our attack cases.

VIII. Conclusion

In this work, we conducted a sensitivity analysis of the openplanner 2.5 rule-based planning algorithm to adversarial data manipulation of lateral and longitude values and delayed sensed-input messages to local-planning nodes. We evaluated these attacks in a low-fidelity simulation test environment using an overtaking manoeuvre critical driving scenario. The results showed that the planning cost-function is sensitive to adversarial data manipulation that introduces deviations to the lateral and longitudinal values. These adversarial deviations cause higher rates of failure to complete missions and cause safety violations. For the message delay attack, limited delays in the range up to approximately 0.6 seconds have a limited impact on the trajectory calculation. Message delays for 1 second or greater cause a visible difference in the safety violation rate and mission success. We opine that limited deviations are an optimal area to explore further attacks and in more diverse critical driving scenarios.

Through this work we proposed a class of stealthy attacks on the local-planning function of AD. An area of future research is the development of monitoring systems developed around such basis of attacks. The results show the feasibility of monitoring real-time properties of the messages propagations and therefore post-mortem forensics might be able to determine the presence of an attacker causing safety violations of AVs.

IX. Acknowledgements

This work has been supported by the European Commission through the H2020 teaming project Finest Twins (grant No. 856602) and European Union's Horizon 2020 Research and Innovation Programme, under grant agreement No 101021727.

References

[1] Y. Abeysirigoonawardena, F. Shkurti, and G. Dudek, "Generating adversarial driving scenarios in high-fidelity simulators," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE Press, 2019, p. 8271–8277. [Online]. Available: https://doi.org/10.1109/ICRA.2019.8793740

[2] Y. Cao, C. Xiao, A. Anandkumar, D. Xu, and M. Pavone, "Advdo: Realistic adversarial attacks for trajectory prediction," in *Computer Vision – ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part V*. Berlin, Heidelberg: Springer-Verlag, 2022, p. 36–52. [Online]. Available: https://doi.org/10.1007/978-3-031-20065-6_3

[3] Y. Cao, D. Xu, X. Weng, Z. Mao, A. Anandkumar, C. Xiao, and M. Pavone. [Online]. Available: https://arxiv.org/abs/2208.00094

[4] B. Chen, X. Chen, Q. Wu, and L. Li, "Adversarial evaluation of autonomous vehicles in lane-change scenarios," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, pp. 10 333–10 342, 2020.

[5] L. Claussmann, M. Revilloud, D. Gruyer, and S. Glaser, "A review of motion planning for highway autonomous driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 5, pp. 1826–1848, 2020.

[6] H. Darweesh, E. Takeuchi, K. Takeda, Y. Ninomiya, A. Sujiwo, L. Y. M. Saiki, N. Akai, T. Tomizawa, and S. Kato, "Open source integrated planner for autonomous navigation in highly dynamic environments," *J. Robotics Mechatronics*, vol. 29, pp. 668–684, 2017.

[7] H. Darweesh, E. Takeuchi, and K. Takeda, "Openplanner 2.0: The portable open source planner for autonomous driving applications," in *2021 IEEE Intelligent Vehicles Symposium Workshops (IV Workshops)*, 2021, pp. 313–318.

[8] G. Deng, G. Xu, Y. Zhou, T. Zhang, and Y. Liu, "On the (in)security of secure ros2," in *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 739–753. [Online]. Available: https://doi.org/10.1145/3548606.3560681

[9] M. Klischat and M. Althoff, "Generating critical test scenarios for automated vehicles with evolutionary algorithms," in *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019, pp. 2352–2358.

[10] E. Malayjerdi, R. Sell, M. Malayjerdi, A. Udal, and M. Bellone, "Practical path planning techniques in overtaking for autonomous shuttles," *Journal of Field Robotics*, vol. 39, no. 4, pp. 410–425, 2022.

[11] M. Malayjerdi, A. Roberts, O. m. Maennel, and E. Malayjerdi, "Combined safety and cybersecurity testing methodology for autonomous driving algorithms," in *Proceedings of the 6th ACM Computer Science in Cars Symposium*, ser. CSCS '22. New York, NY, USA: Association for Computing Machinery, 2022. [Online]. Available: https://doi.org/10.1145/3568160.3570235

[12] P. Narksri, H. Darweesh, E. Takeuchi, Y. Ninomiya, and K. Takeda, "Occlusion-aware motion planning with visibility maximization via active lateral position adjustment," *IEEE Access*, vol. 10, pp. 57 759–57 782, 2022.

[13] M. O'Kelly, A. Sinha, H. Namkoong, J. Duchi, and R. Tedrake, "Scalable end-to-end autonomous vehicle testing via rare-event simulation," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, p. 9849–9860.

[14] R. Sell, M. Leier, A. Rassõlkin, and J.-P. Ernits, "Self-driving car iseauto for research and education," in *2018 19th International Conference on Research and Education in Mechatronics (REM)*, 2018, pp. 111–116.

[15] J. Wang, A. Pun, J. Tu, S. Manivasagam, A. Sadat, S. Casas, M. Ren, and R. Urtasun, "Advsim: Generating safety-critical scenarios for self-driving vehicles," *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021.

[16] C. Xu, W. Ding, W. Lyu, Z. Liu, S. Wang, Y. He, H. Hu, D. Zhao, and B. Li, "Safebench: A benchmarking platform for safety evaluation of autonomous vehicles," 2022. [Online]. Available: https://arxiv.org/abs/2206.09682

[17] Q. Zhang, S. Hu, J. Sun, Q. A. Chen, and Z. M. Mao, "On adversarial robustness of trajectory prediction for autonomous vehicles," *IEEE/CVF Computer Vision and Pattern Recognition Conference (CVPR)*. [Online]. Available: https://par.nsf.gov/biblio/10359466