SoK: Rethinking Sensor Spoofing Attacks against Robotic Vehicles from a Systematic View

Yuan Xu[1], Xingshuo Han[1], Gelei Deng[1], Jiwei Li[2,3], Yang Liu[1], Tianwei Zhang[1]
[1]Nanyang Technological University, [2]Shannon.AI, [3]Zhejiang University
{xu.yuan, xingshuo.han, gelei.deng, yangliu, tianwei.zhang}@ntu.edu.sg, jiwei_li@shannonai.com

***Abstract*—Robotic Vehicles (RVs) have gained great popularity over the past few years. Meanwhile, they are also demonstrated to be vulnerable to sensor spoofing attacks. Although a wealth of research works have presented various attacks, some key questions remain unanswered: are these existing works complete enough to cover all the sensor spoofing threats? If not, how many attacks are not explored, and how difficult is it to realize them?**

This paper answers the above questions by comprehensively systematizing the knowledge of sensor spoofing attacks against RVs. Our contributions are threefold. (1) We identify seven common attack paths in an RV system pipeline. We categorize and assess existing spoofing attacks from the perspectives of spoofer property, operation, victim characteristic and attack goal. Based on this systematization, we identify 4 interesting insights about spoofing attack designs. (2) We propose a novel action flow model to systematically describe robotic function executions and unexplored sensor spoofing threats. With this model, we successfully discover 103 spoofing attack vectors, 26 of which have been verified by prior works, while 77 attacks are never considered. (3) We design two novel attack methodologies to verify the feasibility of newly discovered spoofing attack vectors.

1. Introduction

Robotic Vehicles (RVs), such as self-driving cars, automated guided vehicles and drones, enrich our life with myriad scenarios ranging from package delivery, house cleaning to aerial photography. To accomplish these complex missions, an RV system commonly integrates dozens of functions to manage the physical components (i.e., sensors and actuators) and interact with the environments. These functions are constructed as a closed-loop pipeline with various stages [1]: (1) processing sensor data to estimate the states (*perception*); (2) making decisions to achieve the goals (*planning*); (3) taking actions to change the states (*control*). In such cyber-physical systems, sensors are not only the "eyes" for RVs to observe and understand the surroundings, but also critical attack surfaces for an external adversary to tamper with the systems and cause catastrophic consequences [2]–[5].

The key to this security threat is *sensor spoofing*, a type of practical physical attack that tricks a victim RV into taking dangerous actions. The adversary first injects fake data into the sensors. Then these malicious data are forwarded to the corresponding perception functions, causing them to generate incorrect state estimates. Such wrong perception results can further affect the subsequent executions in the planning and control stages, and finally lead to undesired hazards. For example, a LiDAR spoofer can craft fake laser points to fool the *object detection* function of an autonomous vehicle, which makes it recognize a non-existent obstacle ahead. This can force the vehicle to brake hard on the highway, causing rear-end collisions and endangering the safety of passengers [6]. A large number of works have proposed different types of sensor spoofing attacks [7]–[17]. We ask the following question: *are existing works complete and in-depth enough for us to understand the fundamentals of sensor spoofing threats, and identify all potential attacks?*

Unfortunately, the answer to the above question is no. Existing works on sensor spoofing attacks fall into two main categories. (1) *Perception-level attacks* [18]–[37] target one particular perception function to make it estimate incorrect states. (2) *Vehicle-level attacks* [6], [38]–[84] consider not only fooling the perception function, but also propagating the wrong state estimates towards the subsequent stages and final actions. Both categories of works are only limited to a few specific functions and control flows in an RV system, leaving a large number of unexplored threats. This indicates the existence of a knowledge gap about sensor spoofing attacks, and modern RV systems are vulnerable to unknown attacks.

To bridge this gap, this paper presents a systematic study about sensor spoofing attacks against RV systems. We make three contributions. First, we perform a thorough systematization of sensor spoofing attacks (§ 3). Particularly, we identify 7 common attack paths in the RV system pipeline (§ 2). Then we categorize existing attacks from 71 relevant papers, covering 6 types of mainstream sensors and 3 types of RV systems. For each attack, we analyze its practicality, aggressivity and stealthiness from four perspectives: spoofer property, spoofing operation, victim characteristic and attack goal. Based on the systematization, we also identify 4 interesting insights that inspire researchers to explore more sophisticated attacks.

Second, we build a unified action flow model to describe the sensor spoofing attacks, and discover 77 new unexplored threats (§ 4). The key insight of our model is to abstract the spoofing attacks based on their *action flows*, which are defined as end-to-end paths from the sensor data to the RV's final actions. Each robotic function in an action flow can be the attack target, and compromising it could directly or indirectly affect the RV's behaviors. Based on this unified model, we identify 44 action flows and 103 spoofing attack vectors. By analyzing all these 103 attack vectors, we find 26 of them have been realized previously, and they cover all the existing works. More

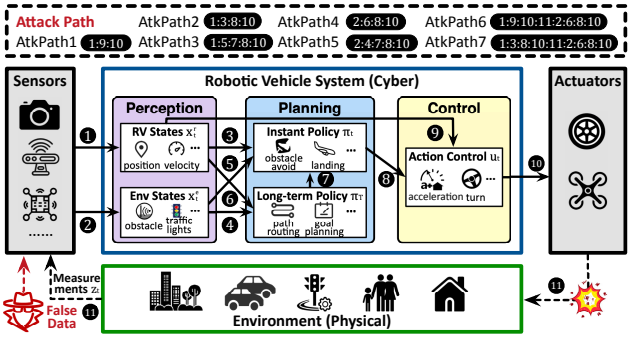

Figure 1: Overview of an RV system pipeline.

importantly, 77 attacks have the potential to cause fatal accidents. They are never considered in prior works.

Third, we propose two new attack methodologies to validate the feasibility of the discovered unexplored threats (§ 5). Specifically, we perform an investigation towards these 77 attack vectors. We find most of them can be easily realized using similar techniques from prior works, while the rest can be categorized into two scenarios: obstacle position altering using the LiDAR spoofer and location altering using the camera/LiDAR spoofer. We design novel approaches to achieve these scenarios. We implement prototypes on the KITTI [85] dataset with the PointRCNN model [86] and ORB-SLAM2 [87] simulator to prove these attacks are practical.

2. RV System and Spoofing Attack Paths

2.1. RV System Pipeline

An RV system can be generally modeled as a set of sensor inputs, system states, and control outputs. These three components change over time when the RV takes actions to interact with the environment. They are denoted as z_t, x_t and u_t at time t, respectively. Figure 1 shows the workflow of an RV system. The pipeline is composed of three stages.

Perception. The RV estimates its system states x_t from the raw sensor data z_t. The system states x_t include two parts: x_t^r represents RV's operation states (e.g., velocity, position) while x_t^e represents the perceived environment states (e.g., nearby obstacles, traffic lanes, pedestrians).

Planning. To accomplish the mission, the RV needs to make a long-term policy (π_T) to transit its initial states x_1 to the final states x_T. During the execution, it keeps computing the instant control policy (π_t) from x_t to x_{t+1}. For instance, before a journal, a vehicle first needs to plan a path (π_T) from the current position (x_1) to the destination (x_T). When driving along this path, the vehicle needs to ensure that it does not encounter any obstacles and complies with the corresponding traffic laws (π_t).

Control. This stage instructs the actuators and drives the RV to interact with the environment. The control outputs u_t depend on the instant policy π_t and system states x_t. For example, an RV needs to stop or make a turn to avoid the obstacle ahead.

2.2. Attack Paths

Threat model. Following the standard threat model in sensor spoofing works, we assume that the adversary cannot directly access the victim RV, altering its configuration settings or installing malware apps. He can only change the external environment or interfere with the sensor data. The adversary's goal is to tamper with the sensor inputs (z_t), which then compromise the system states x_t and control outputs (u_t).

We analyze the possible attack paths (AtkPaths) in the RV pipeline. An attack path describes how fake sensor data can affect the subsequent function executions and results (Figure 1). We first identify five attack paths to compromise the RV:

- **AtkPath1** (1:9:10): The spoofer alters the operation states x_t^r (❶) to destabilize the control stage (❾), and further cause abnormal actions u_t (❿). For example, fake Inertial Measurement Unit (IMU) data can destabilize a drone and drag it down [73].
- **AtkPath2** (1:3:8:10): The adversary falsifies the operation states x_t^r (❶) to influence the instant policy π_t (❸). A misguided decision-making process will generate wrong actions u_t (❽) and put the RV in danger (❿). For instance, a drone is forced to land when it is misdirected to a no-fly zone [38].
- **AtkPath3** (1:5:7:8:10): The spoofer changes the operation states x_t^r (❶) to mislead the long-term policy π_T (❺) and then the instant policy π_t (❼). This continuously controls RV's actions u_t (❽) until it reaches a malicious final state x_T (❿). For example, an adversary can guide the vehicle to a wrong destination by continuously and slightly shifting the GPS location [42].
- **AtkPath4** (2:6:8:10): This attack path is similar as AtkPath2, except that it tampers with the environment states x_t^e. For example, mis-estimating a non-existent obstacle can force the RV to brake hard on the highway [6].
- **AtkPath5** (2:4:7:8:10): This attack compromises the environment states x_t^e to achieve similar consequences as AtkPath3. For example, an adversary can spoof the microphone to assign a wrong navigation mission to a vehicle and force it to reach a designated destination [79].

The environment states x_t^e sensed by the RV changes along with the operation states x_t^r. Once the actions are compromised, the RV perceives the surroundings in unexpected manners and make false decisions. This is reflected in Figure 1 where an adversary can leverage AtkPath1-3 to achieve the attack result of AtkPath4-5. We identify such two attack paths, and call them *multi-round attack path*.

- **AtkPath6** (1:9:10:11:2:6:8:10): It first performs AtkPath1 to influence the perceived environment and sensor measurements z_t (❶❾❿⓫), and then causes the launch of AtkPath4 (❷❻❽❿). For example, an adversary can control the IMU data to trigger unnecessary motion compensation and generate a blurred image. The blurred images can then induce object misclassification to make the RV take dangerous decisions [78].
- **AtkPath7** (1:3:8:10:11:2:6:8:10): The adversary falsifies a malicious position x_t^r and causes the change of sensor measurements z_t (❶❸❽❿⓫), which then leads to the occurrence of AtkPath4 (❷❻❽❿). For example, when multiple traffic lights exist in the camera view, a counterfeit position can make the vehicle confused about different traffic lights in the region-of-interests (ROI) and take unexpected actions because ROI utilizes position to narrow the detection scope in the sensor [45].

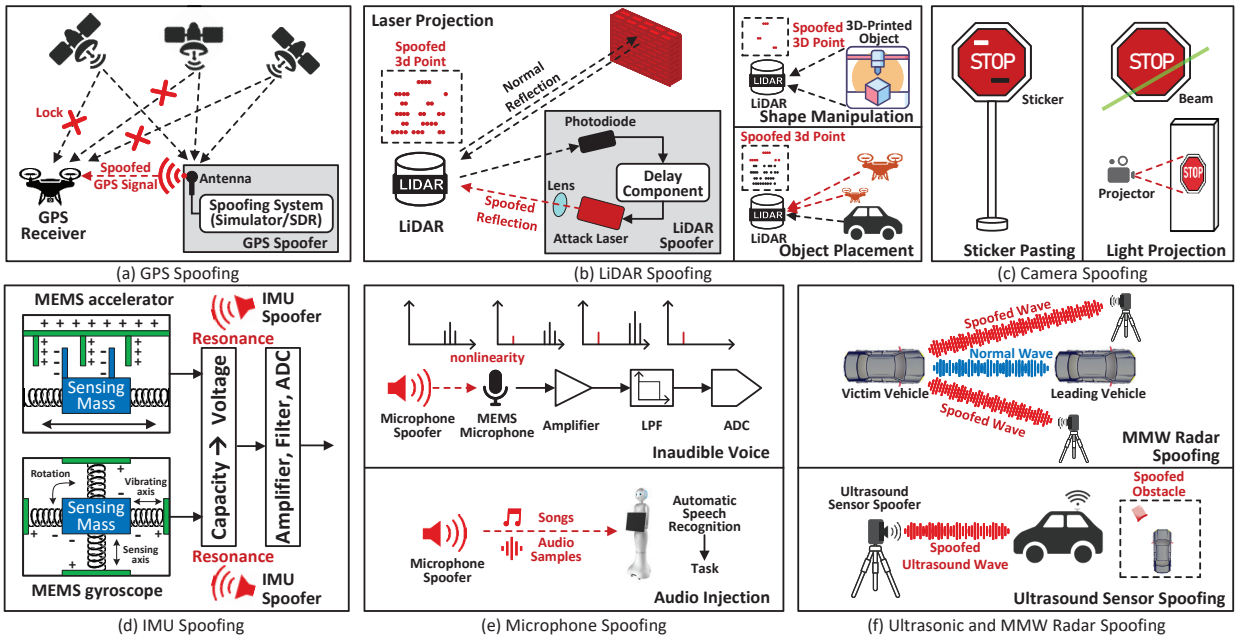

Figure 2: Illustration of sensor spoofing attacks on six mainstream sensors.

Note that there can be some other two-round attack paths from Figure 1, including AtkPath3+4 and AtkPath1/2/3+5. We observe that AtkPath1, AtkPath3 and AtkPath4 aim to trigger *instant* abnormal events and cause malicious actions, while AtkPath2 and AtkPath5 focus on continuously spoofing sensors to change the *long-term* goals. Thus, it is hard to combine them in practice. AtkPath2+5 (i.e. 1:3:8:10:11:2:4:7:8:10) can be attacked in theory if the vehicle performs different missions according to its position. However, we cannot find any works about these attacks, and will not discuss them in this paper.

2.3. Systematization Scope

This paper focuses on sensor spoofing attacks against RVs. Although there are almost 400 types of sensors on record [88], we consider six mainstream sensors: GPS, LiDAR, camera, IMU, microphone, and ultrasonic sensor/MMW radar. They are fully or partially integrated into modern RVs to assist them for planning and control. Spoofing attacks on other sensors, e.g. temperature sensor [89], pressure sensor [90], [91] and magnetic sensor [92], [93], are beyond the scope of this paper. Attacks against other cyber-physical systems (e.g., smart-home speakers [94], medical devices [95]) are not studied either.

We target three types of popular RVs (autonomous vehicles, drones, and automated guided vehicles), which are widely adopted in our daily life. These RV systems follow the standard modular design and pipeline described in § 2.1. We do not consider the end-to-end robotic systems which utilize single machine learning models to directly output the control command from the sensor data [96], [97], as they are generally explored in academia, and not ready for commercial use. Besides, we mainly focus on the vehicle-level attacks [6], [38]–[84] , i.e., an end-to-end attack causing malicious actions from the spoofed perceived states to the planning and control subsystems. Perception-level attacks [18]–[36] can be regarded as the preliminary step for vehicle-level attacks, and will not be particularly discussed in our systematization.

While jamming attacks [98], [99] can cause malfunctions of RVs, we exclude them from this work due to two main considerations. First, sensor jamming attacks crudely block the perceived data instead of arbitrarily modifying the sensor readings. This gives the adversary less flexibility to mislead the RVs in a more precise way. Second, jamming attacks are generally not stealthy and easy to detect [100]. Existing works have demonstrated solutions to defeat jamming attacks on various sensors, such as Radar [101], [102], GPS [103], [104], and ultrasonic sensor [105]. Meanwhile, we do not consider cyber attacks against RVs (e.g., software and ROS vulnerabilities [106], [107] and in-vehicle networks [108]–[110], DNN backdoor [111], communication protocols [112], [113], and side-channel leakage [114]) in this work because they do not directly target on-vehicle sensors.

Comparisons with existing surveys. A few works also conduct surveys related to RV or sensor security. However, they are significantly different from this paper. (1) *Scope*: some papers focus on the general security and safety problems in specific RV systems, e.g., drones [115]–[123], autonomous vehicles [46], [124]–[133]. Some papers just target one type of sensor spoofing, e.g., GPS [7]–[11], camera [12]–[15] or microphone [16], [94]). Differently, we explore various types of sensor spoofing attacks against different RV systems. (2) *Contribution*: we provide a deeper analysis and categorization on spoofing attacks in a systematic way. We identify the common attack paths from the RV system pipeline and assess existing works from different perspectives. We build an action flow model, which can not only cover existing attacks but also disclose new unexplored threats. We also design and prototype new spoofing attack approaches. These are rarely performed in prior works. Yan et al. [17] also introduced a sensor security model to describe spoofing attacks and predict new vectors. It focuses on the underlying signal processing mechanisms in each sensor at the perception level. On the contrary, our action flow model is based on robotic function executions and interactions at the vehicle level, which is complementary to [17].

TABLE 1: Systematization of sensor spoofing attacks.

Spoofing Techniques		Spoofer Property				Spoofing Operation		Victim Characteristic		Attack Goal		Paper
		Cost	Size	Signal	Recog.	Range	Exposure	Type	Scenario	State	Attack Path	
GPS Spoofing		$$/$$$	S1/S2		False	R2	Active			Position	AtkPath2	[38]–[41]
											AtkPath3	[41]–[43]
										Object	AtkPath6	[44]
											AtkPath7	[45]
LiDAR Spoofing	laser projection	$$$	S2		False	R2	Active			Object	AtkPath4	[6], [46]–[49]
	shape manipulation	$$	S1		True	R2	Passive			Object	AtkPath4	[50]
	object placement	$$	S1		True	R2	Passive			Object	AtkPath4	[51]
Camera Spoofing	sticker pasting	$	S1		True	R2	Passive			Object	AtkPath4	[52]–[60]
										Object	AtkPath5	[71]
										Lane	AtkPath4	[69], [70]
	light projection	$/$$	S1		False	R2	Active			Object	AtkPath4	[61]
					True	R2	Active			Object	AtkPath4	[62]–[68]
										Lane	AtkPath4	[64], [67]
										Velocity	AtkPath1	[72]
IMU Spoofing		$$	S2		True	R2	Active			Velocity	AtkPath1	[73]–[77]
										Object	AtkPath6	[78]
Microphone Spoofing	inaudible voice	$$$	S2		False	R2	Active			Mission Goal	AtkPath5	[79], [80]
	audio injection	$	S1		False	R2	Passive			Mission Goal	AtkPath5	[79]
Ultrasonic Sensor Spoofing		$	S1		False	R1	Active			Object	AtkPath4	[81], [82]
MMW Radar Spoofing		$$$	S2		False	R2	Active			Object	AtkPath4	[82]–[84]
											AtkPath5	[84]

Satellite Signal · Visible light or infrared · 2D sticker · 3D object · Audible sound or ultrasound · RF waves
Autonomous Driving · Drone · Mobile Base · **Recog.** Recognizability · Indoor · OutdoorInjection

3. Systematization of Existing Attacks

We first present our categorization methodology (§ 3.1). Then we perform a literature review about sensor spoofing attacks (§ 3.2-§ 3.7). Finally, we draw some interesting insights from the systematization (§ 3.8). Table 1 lists the summary of these works, and Figure 2 illustrates the basic mechanism of each attack.

3.1. Systematization Methodology

1) Spoofer Property. An adversary needs a spoofer to interfere with the victim RV. The spoofers in different attacks can have distinct properties, which determine the attack cost, feasibility and stealthiness. We evaluate four properties. **a) Cost:** this is the price to purchase or set up the spoofer. We consider three levels: less than $100 ($); between $100 and $1000 ($$); more than $1000 ($$$). **b) Size:** this denotes the physical size of the spoofer. It is easier and stealthier to perform attacks with small-size spoofers. We consider two types: non-portable – larger than a mug (S2); portable – smaller than a mug (S1). **c) Signal type:** satellite signal; visible or infrared light; 2D sticker; 3D object; audible sound or ultrasound; and RF waves. **d) Recognizability:** this denotes whether the spoofer or spoofed signal can be noticed by the victim user (True), or can conceal themselves in the environment (False).

2) Spoofing Operation. We consider different operations the adversary performs to attack the RV. **a) Range**: the minimal distance between the spoofer and RV required for effective interference - larger than 5m (R2); smaller than 5m (R1). **b) Exposure:** this indicates whether the adversary needs to actively expose himself to perform attacks (A) or passively mislead the sensors (P).

3) Victim Characteristic. We also assess the attacks based on two characteristics of the victim RVs. **a) Type**: autonomous vehicle () with various sensors and strict compliance to traffic rules; drone () utilizing IMU and quadrotors for stability; automated guided vehicle () with limited low-end sensors due to cost constraints. **b) Scenario**: the indoor scenario () or outdoor scenario (). There are significant differences between indoor and outdoor RV, which must be taken into account when designing security systems for these vehicles. In indoor environments, GPS localization is not feasible, the RV typically relies on Simultaneous Localization And Mapping (SLAM) or IMUs for localization. Additionally, indoor RVs also tend to run slower than their outdoor counterparts. This can actually be advantageous in terms of stability and result in less crash damage in certain attack scenarios.

4) Attack Goal. We consider the attack goal from two dimensions. **a) Compromised state:** various states can be attacked, including position, nearby obstacles (Object), the traffic lanes (Lane), velocity and mission goals. **b) Attack path**: we identify the attack path (AtkPath1-7) exploited by the adversary, as discussed in § 2.2

3.2. GPS Spoofing Attacks

The Global Position System (GPS) is widely integrated into outdoor RVs for localization. The GPS receiver calculates its position based on the information received from multiple satellites, including pseudorange and navigation messages. A navigation message consists of the transmission time of the code epoch and the satellite position at that time.

The lack of signal authentication makes GPS vulnerable to spoofing attacks [134]. As shown in Figure 2(a), the adversary first uses a GPS spoofer to transmit false GPS signals with strong power to the victim GPS receiver. These fabricated signals lure the victim to lose track of the satellites and lock onto the attacker's signals. Next, the adversary can manipulate the GPS receiver by either adjusting the apparent pseudorange to the satellite [20] or modifying the navigation messages [18], [19]. The counterfeit signal is then sent to the victim GPS receiver.

1) Spoofer Property. There are two common types of GPS spoofers: GPS simulator and Software Defined Radio (SDR). The GPS simulator is heavy, expensive but more powerful. Generally, a complete digital GPS simulator with the multi-GNSS capability is as big as a computer server (size: S2) with the price between $20,000 and $50,000 (cost: $$$) [9]. Under different configurations, the simulator can simulate from only 10 satellite signals (signal:) at one time (e.g. WelNavigate GS72 [135]) to 64 simultaneous signals and multiple GNSS systems (e.g. Orolia GSG 5/6 series [136]). In comparison, the SDR spoofer is more popular due to its low-cost, easy-operation

and white-box features. An attack is successfully demonstrated against autonomous vehicles in [42] with a pen-size SDR spoofer at the price of $223 (cost: $$, size: S1). The GPS spoofer can be hidden in an adversarial vehicle to emit the imperceptible but effective fake GPS signals [42]. Thus, it is hard for the victim to notice the attack. (recog.: False).

2) Spoofing Operation. Both the GPS simulator and SDR spoofer need to transfer the counterfeit satellite signals to the victim's GPS receiver (exposure: A). They can be launched at longer distances (e.g., 40m away [42]) from the victim RV (range: R2).

3) Victim Characteristic. GPS spoofing has been realized to attack drones [11], [38]–[40] and autonomous vehicles [42], [43] in the outdoor scenario since the satellite signals would be blocked by walls (type: , scenario:).

4) Attack Goal. The adversary compromises the position of the victim RV (state: Position). This can incur additional effects for the subsequent functions with different attack paths. (1) AtkPath2: Some attacks were proposed to control DJI drones to enter or leave a no-fly zone [38], [39]. Once reaching the coordinates of a no-fly zone, the drone has to perform an emergency landing, which is unexpected. A counterfeit position can also cause a DJI drone to fly to an incorrect destination after entering the return-to-home mode [40]. (2) AtkPath3: Zeng et al. [42] introduced an attack, which continuously and slightly shifts the GPS position to manipulate the road navigation system of an autonomous vehicle. Then the fake navigation route will match the shape of the actual roads, and induce the vehicle to a dangerous destination. Shen et al. [43] proposed the *off-road* and *wrong-way* attacks, which perform continuous GPS spoofing with large lateral deviations to mislead the autonomous vehicle to drive off the road or onto the opposite lane.

The adversary can also change the semantic information of the objects based on the spoofed position (state: Object). Li et al. [44] designed an attack against the motion compensation mechanism, which uses the GPS information to fix LiDAR distortions. By falsifying the positions with the GPS spoofer, the adversary can compromise the LiDAR-based object detection function and make safety-critical objects undetectable by the victim vehicle (AtkPath: 6). Tang et al. [45] proposed to manipulate the location with GPS spoofing to affect the position of ROI in the traffic light detection function, thus leading the victim vehicle to run a red light (AtkPath: 7).

GPS can also be used to estimate the velocity through differentiating two consecutive positions [137], [138]. However, most commercial RVs use IMU or camera as the velocity estimator rather than GPS due to the accuracy. Thus, this paper does not consider this opportunity.

3.3. LiDAR Spoofing Attacks

A LiDAR sensor is used to measure the distance from the RV to surroundings by the Time-of-Flight (ToF) method, i.e. firing rapid laser pulses and capturing the reflected light using photodiodes. With such information, the RV can recognize the shape and position of any object in the form of point clouds.

Three basic techniques to conduct LiDAR spoofing attacks have been proposed in existing works. (1) *Laser projection*. As shown in Figure 2(b), the adversary uses a photodiode to synchronize with the victim LiDAR, and then delays the received laser pulses. Then he can choose the fake points that appear in the point cloud by crafting a pulse waveform. Previous works have demonstrated the possibility of relaying LiDAR laser pulses from different locations [46] or controlling fake points at different positions in the point cloud [6], [47]–[49] for LiDAR spoofing attacks. (2) *Shape manipulation*. Since each point position in the point cloud also depends on the shapes of the target 3D object, the adversary can craft some objects with carefully-designed shapes to deceive the LiDAR [50]. (3) *Object placement*. The adversary can use existing objects and place them in identified positions to generate counterfeit laser points and interfere with the perception results of the point cloud model [51].

1) Spoofer Property. (1) For the laser projection technique, the spoofer consists of a photodiode, a laser diode, a laser driver module and a delay generator. The prices of the first three devices are about $1, $20 and $150 respectively. The delay generator costs thousands of dollars and its size is as big as a microwave (cost: $$$, size: S2). It spoofs fake points with the laser signal, which is an invisible light and hard to be noticed by the victim (signal: , recog.: False). (2) For the shape manipulation technique, the adversary can use a 3D printer to create a well-designed 3D object that can generate abnormal point clouds (signal:). A 3D printer costs hundreds of dollars (cost: $$). The printed adversarial object commonly has unique and noticeable shape in the physical world to effectively fool the point cloud model [50] (size: S1, recog.: True). (3) For the object placement technique, the adversary places some existing objects to generate extra laser points (signal:). For example, the adversary can control drones to hover around other obstacle's locations [51] (cost: $$, size S1). The victim passengers can be alert when observing multiple drones constantly flying in front of them (recog.: True).

2) Spoofing Operation. The maximum effective attack distance of both laser projection and shape manipulation techniques depends on the firing range of the LiDAR, which is commonly up to 100m (range: R2). The attacks aim to fool the point cloud model for object detection. The laser projection technique needs the adversary to actively inject the counterfeit laser pulses (exposure: A) while the other two techniques deceive the LiDAR by placing the adversarial objects without runtime intervention (exposure: P).

3) Victim Characteristic. Almost all existing LiDAR spoofing attacks target outdoor autonomous vehicles [6], [46]–[50] (type: , scenario:).

4) Attack Goal. All the three techniques aim to tamper with the perceived environment (state: Object) and lead the RV to make wrong actions (AtkPath: 4). Specifically, (1) some attacks create a non-existence object (e.g., wall [46], vehicle [49], arbitrary objects [6]) in front of the victim vehicle. This can cause two consequences: the victim vehicle has to perform a hard brake, which could hurt the passengers inside or cause rear-end collisions; if the spoofed object is placed at the cross-road, it could freeze the victim vehicle even the traffic light is green, causing heavy traffic congestion. (2) Some attacks erase existing objects from the victim's perception output [47],

[50], [51]. As a result, the victim vehicle cannot recognize the objects and will crash into them.

3.4. Camera Spoofing Attacks

A camera is an optical-electrical device that converts the light perceived by lens to electrical signals. The adversary can alter the results by adding visual perturbations.
1) Spoofer Property. (1) The sticker-pasting technique creates a counterfeit patch to fool the camera (signal:). Due to the physical constraints, the adversarial patch is still visually abnormal (size: S1, recog.: True). It is very cheap to print such a patch (cost: $). (2) The light-projection technique can be implemented in two ways: using a laser pointer to shine laser beams on the target object or into camera, or using a projector to project adversarial images on the road or wall (signal:). Both spoofers are small and can be mounted on a drone [64], [67] (size: S1). A common laser pointer costs hundreds of dollars (cost: $), which is much cheaper than a projector with high lumen intensity and resolutions (cost: $$). Unlike LiDAR, camera spoofing attacks can be implemented using either visible natural light (recog.: True) or invisible infrared light (recog.: False) [61]. In particular, infrared light exploits a portion of the spectrum that is invisible to humans but detectable by cameras.
2) Spoofing Operation. Camera spoofing is commonly launched under the post-processing setting with a large attack range (range: R2).
3) Victim Characteristic. Most works on camera spoofing attacks target the object or lane detection function in the outdoor autonomous driving scenario [52]–[56], [61]–[67], [69]–[71] (type: , scenario:). One exception is [72], which uses light-projection spoofers to counterfeit the lateral drift velocity and induces the drone to follow the motion to compensate the spoofed drift (type:).
4) Attack Goal. There are different goals and attack paths for camera spoofing attacks. (1) Object detection: a quantity of works attack the traffic controller detection [52]–[56], [139]. They cause the vehicle to make wrong classification results and control actions, e.g., a stop sign is mis-classified as a speed limit sign. Some works target the obstacle detection function to make the RV detect a non-existence obstacle [61]–[67], or miss an existing obstacle [54], [57]–[60], [140]. All these works belong to the category of (state: Object, AtkPath: 4). (2) Lane detection: some works [64], [67], [69], [70] change the correct trajectory of the vehicle by misleading lane markings on the highway (state: Lane, AtkPath: 4). (3) Object tracking: Jia et al. [71] extended the sticker pasting attack to multiple object tracking tasks, which deceives the RV through continuously spoofing the position of the obstacle (state: Object, AtkPath: 5). (4) Davidson et al. [72] proposed to use the counterfeit lateral drift velocity to destabilize the control system and cause further damages (state: Velocity, AtkPath: 1).

3.5. IMU Spoofing Attacks

The Inertial Measurement Unit (IMU) is one core sensor to help RVs adjust the speed of the rotors or motors for stabilizing the balance. It consists of a gyroscope, an accelerometer and a magnetometer to measure the rotation, acceleration, and orientation of an RV. The IMU commonly adopts the Micro-electromechanical (MEMS) technology. Specifically, the gyroscope and accelerometer use a similar mass-spring structure. Once the RV moves, the sensing mass will vibrate continuously, which changes the capacitance and then induces electrical signals. The signals will be digitized by the analog-to-digital converter (ADC) and output linear and angular rates. Recent works show that both gyroscope and accelerometer are susceptible to resonant acoustic interference [22]–[24], [73]–[77]. As shown in Figure 2(d), the adversary can generate sound waves with the similar frequency as the spring-mass structure. Such acoustic signals can set up resonance, forcing the sensing mass to move and spoofing the designated values.
1) Spoofer Property. An IMU spoofer consists of a function generator, a sound amplifier and a tweeter speaker. Since the resonant frequency of the victim IMU is commonly below 1MHz [74], [77], a $320 low-end function generator with the maximum frequency of 20MHz is effective for generating fake signals [75] (cost: $$, size: S2, signal:). It needs to be operated near its maximum amplitude (around 110dB Source Pressure Level) [74], [77], [78], which is hard not to be noticed (recog.: True).
2) Spoofing Operation. The attack distance of the IMU spoofer depends on the strengths of the received modulated acoustic signal, which is further determined by the signal transmitting power. Son et al. [73] showed the attack distance can reach 37.58m with LRAD 450XL [141] or UltraElectronics UyperShield [142] (range: R2). The spoofer actively injects resonant noise to the victim sensor (exposure: A), and induce the it to directly generate spoofed raw data.
3) Victim Characteristic. The IMU spoofing attack can cover almost all types of RVs for both indoor and outdoor scenarios (type: , scenario:).
4) Attack Goal. There are two types of goals for IMU spoofing attacks. The first type is to tamper with the linear or angular velocity, and make the victim RV lose control with the spoofed velocity (state: Velocity, AtkPath: 1). For instance, Trippel et al. [74] proposed the *output biasing* and *output control* attacks on the MEMS accelerometer. They were then extended to the gyroscope with the *side-swing* and *switching* attacks [75], [77]. Nashimoto et al. [76] further discussed the attack that involves an accelerometer, gyroscope and magnetometer. The second type is to spoof the sensors to fool the victim RV for object detection (state: Object, AtkPath: 6). Ji et al. [78] introduced a novel attack that could deceive the image-stabilization-based objection detection function in the autonomous driving system. They found the spoofed IMU data can cause an object to become undetected, misclassified or create a non-existent object.

3.6. Microphone Spoofing Attacks

The microphone is the key component of a voice-control system for human-RV interaction [79]. For example, the user can issue voice commands to specify the destination for autonomous driving, or launch a mission by a service robot. Commonly, RVs use the MEMS microphone, which consists of a transducer, an amplifier, a low pass filter (LPF), and an ADC. When a sound wave

is received, the air pressure flexes the membrane in the transducer and changes the capacitance [143]. The LPF and ADC filter the amplified signals beyond the frequency range of human hearing (20Hz $\sim$ 20kHz).

The microphone spoofing attack aims to make the target system execute malicious voice commands without being detected or recognized by normal users. Recent attacks leverage two methods to achieve this goal: (Figure 2(e)). (1) *Inaudible voice* technique: the adversary replays synthetic ultrasound signals and disguises them to legitimate digital speech signals based on the nonlinearity of the amplifier [25]–[29], [80]. (2) *Audio injection* technique: the adversary hides the adversarial audio in the background noise or songs [30]–[34].

1) Spoofer Property. The microphone spoofer in the inaudible voice method is similar as the IMU spoofer while the function generator requires a much larger sampling range (cost: $$$, size: S2). The audio injection method can be achieved with a media player (cost: $, size: S1). Both methods are difficult to be noticed by the victim since the malicious voice is either inaudible or hidden (signal: , recog.: False).

2) Spoofing Operation. Over the years, the effective attack distance for the inaudible voice technique has been increased from 1.75m [25] to 19.8m [80] (range: R2). The adversary shifts a high-frequency inaudible signal to a low-frequency audible signal, and then injects it to the speech recognition function (exposure: A). In contrast, the audio injection technique aims to hide the malicious acoustic commands into the normal audio waveform (range: R2, exposure: P).

3) Victim Characteristic. Recent works on microphone spoofing attacks target the voice-control system in autonomous vehicles [79], [80] (type: , scenario:). Modern vehicles (e.g., Tesla [144], Audi [145], Lincoln [146]) support a list of voice commands that will be converted into the navigation goal, which are vulnerable to the spoofing attacks.

4) Attack Goal. The adversary generates malicious voice commands, which are further converted into unexpected missions for the RV to execute (state: mission goal, AtkPath: 5). For instance, Yan et al. [80] applied the inaudible voice technique [25]–[29] to manipulate some in-car features in an Audi autonomous vehicle, such as navigation, entertainment, environmental controls and mobile phone control. Zhou et al. [79] discussed the possibility of using the audio injection method [30]–[32], [34], [35] to control navigation functions in autonomous driving systems as well.

3.7. Ultrasonic & MMW Radar Spoofing Attack

The ultrasonic sensor and MMW radar also utilize the ToF method to measure the distance between the RV and an obstacle. The ultrasonic sensor transmits and receives ultrasound waves, which have a relatively low speed (340m/s) and are vulnerable to bad weather. Therefore, it is commonly used in simple scenarios, such as automatic parking. In contrast, the MMW radar relies on the millimeter waves, and is widely used in outdoor autonomous vehicles. It can assist LiDAR and cameras to detect obstacles under extreme weather conditions. In addition, it can also be used to track objects and estimate their velocity. As shown in Figure 2(f), the adversary can spoof these two sensors by relaying the received signal and sending it back to the transmitter [36], [81]–[84].

1) Spoofer Property. The ultrasonic spoofer consists of an envelope detector, ultrasonic transducers, amplification circuits, a buffer amplifier and a square wave generator. A recent work [81] showed an Arduino board at the price of $20 can generate the required square waves of the selected frequency (40$\sim$50kHz) (cost: $, size: S1, signal:). Due to the large frequency range in the radar, e.g., 76$\sim$77GHz MRR Radar installed on Tesla, an effective spoofer that works at such high frequency can cost more than 10 thousand dollars with a large size [82] (cost: $$$, size: S2, signal:). Both ultrasound and MMW radar waves are imperceptible (recog.: False).

2) Spoofing Operation. Both attacks generate counterfeit signals and actively inject them into the victim's sensors (exposure: A). The attack distance of the ultrasonic spoofing attack is up to 2 meters [81] (range: R1) while that of the radar spoofing attack can reach 26m [84] (range: R2).

3) Victim Characteristic. Both attacks target autonomous vehicles in an outdoor scenario (type: , scenario:). The ultrasonic spoofing attack can also be applied in an indoor parking lot (scenario:).

4) Attack Goal. Existing spoofing attacks aim to create a non-existence object in front of the victim RV or falsify the location of an existing obstacle (state: Object). These attacks lead to two different attack paths. (1) Some works launch the ultrasonic spoofing attack [81], [82] against different commercial autonomous vehicles. Miura et al. [83] reduced the cost of radar spoofing attack by using a replica radar and one additional small Micro-Control Unit (MCU). Sun et al. [84] deployed radar spoofing attacks to cause vehicle stalling, hard braking and lane changing. These attacks follow the AtkPath4. (2) Sun et al. [84] also proposed the multi-stage and cruise control attacks, which lead to a high speed crash by leveraging a long-term plan and control in autonomous driving. This is AtkPath5.

3.8. Insights and Lessons

We identify some interesting observations and lessons from the above systematization.

> **Insight 1:** *Compared to autonomous vehicles, there are relatively fewer attacks targeting drones and automated guided vehicles.*

According to Table 1, we can observe that the majority of works (38/48) target autonomous vehicles, which share widely-considered attack goals and scenarios. Since all the three types of RVs share many functions and mechanisms, those spoofing attacks might be applied to the drones and automated guided vehicles as well. Considering they are playing more important roles in our daily life, such as delivery, photography, surveillance and house cleaning, more efforts should be devoted to the study of attacks against them. Their unique features (e.g., limited computing resources, low-end sensors, eyes in the sky) also lead to other types of attacks, such as cloud-based DoS attacks [147] and privacy invasion attack [148].

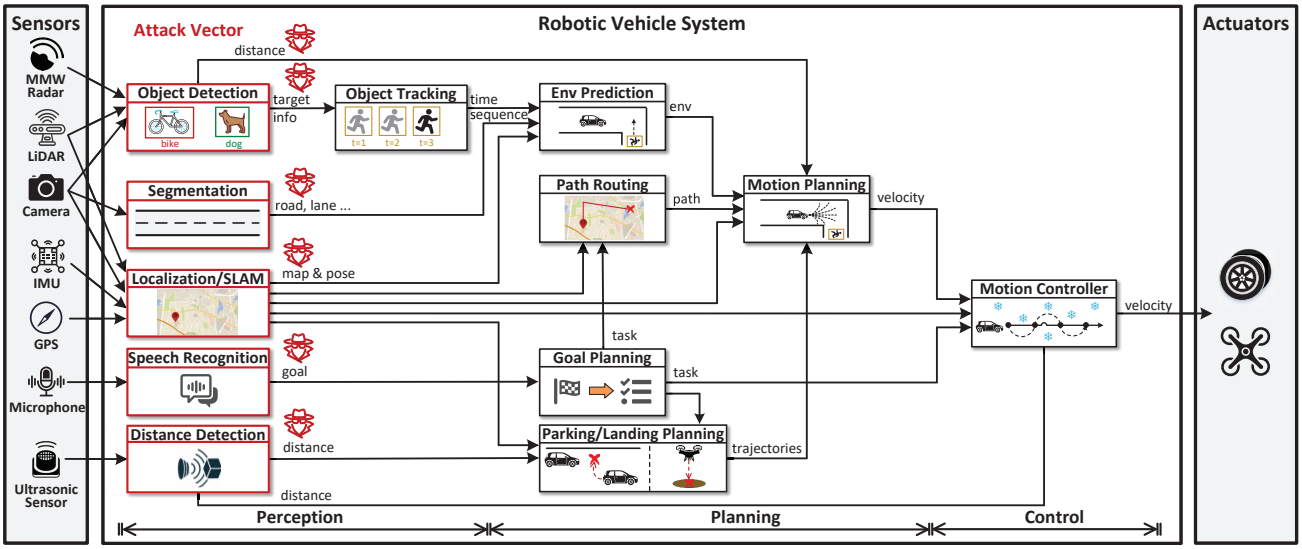

Figure 3: The action flow model of RV systems.

Insight 2: *The same attack path and consequence can be triggered by different sensors and spoofers, thus giving the adversary more choices to conduct attacks.*

As shown in Figure 1, each attack path starts from the mis-estimated states (❶/❷) and ends with abnormal actions (⓾). So once the operation or environment state is compromised, the subsequent wrong executions will occur correspondingly. The goal of the attack on RVs is to spoof the RV's state and ultimately cause it to perform dangerous actions. Since some sensors provide similar functionalities to RVs (e.g., LiDAR and camera), attackers can achieve the same attack goal through different attack paths. It is important to note that even though the attack results may be the same, the attack methods can be different depending on the sensor being used for state estimation. This insight can inspire us to analyze spoofing attacks that go from sensor to actuator and thus identify more unexplored attacks. For instance, the SLAM function enables an RV to localize itself with the camera or LiDAR inputs. Then the position-dependent AtkPath2 and AtkPath3 related to the GPS spoofers in Table 1 can also be implemented by the camera or LiDAR spoofers. The adversary can choose the most convenient and efficient attack solution to achieve his desired goal.

Insight 3: *There are much fewer studies about the multi-round spoofing attacks.*

Among all the seven attack paths, AtkPath6 and AtkPath7 are multi-round attacks, which are the combinations of two single-round paths: the adversary first exploits an x_t^r-related attack path to change RV's perceived environmental measurements, and then causes the spoofing attack on an x_t^e-related path. The interaction between the RV and its surrounding environment bridges these two distinct attack paths. From Table 1, we observe there are only 3 out of 42 works exploring two specific multi-round attacks: target blurring and ROI altering attacks. We hypothesize that the lack of such studies is due to the complexity of multi-round mechanisms and RV-environment interactions. We believe there are more opportunities for designing such attacks. On one hand, researchers can use other sensor spoofers to implement the target blurring and ROI altering attacks; on the other hand, more multi-round attack paths besides AtkPath6 and AtkPath7 can be explored in the future.

Insight 4: *GPS, Microphone, MMW Radar spoofing and laser projection are more stealthy than others.*

A successful spoofing attack should be stealthy enough to avoid the detection of the victim. To achieve this, we identify the following combinations of spoofer characteristics. (1) *Passive exposure + small size*: a passive attack is to hide the spoofer in the environment without human intervention, so it is hard for the victim to recognize the existence of the spoofer. Thus, the spoofer is imperceptible as long as it is small enough and highly integrated into the environment, e.g., the audio injection attack. (2) *Active exposure + imperceptible signal + remote attack range*: Spoofing at a long distance (e.g. 10 meters) is stealthy even if someone is observing, because spoofers are too small to be identified at such a long distance. In addition, the imperceptible signal (e.g., GPS, laser, ultrasonic and MMW radar) allows the victim to detect the attack only by observing the cheater at a distance, which further ensures the stealthiness. We encourage that other spoofing attacks can improve their stealthiness from these two dimensions.

4. A Unified Model for Spoofing Attacks

We propose a novel action flow model to systematically describe the spoofing attacks. This model not only covers all the existing attacks summarized in § 3, but also reveals new attacks that are never considered previously.

4.1. Action Flow Model

Existing studies on sensor spoofing attacks tend to follow an ad-hoc way: researchers first identify the mis-estimated state that can incur the desired consequence, and then design the spoofing methodology that can falsify the state. The lack of systematization cannot guarantee the comprehensiveness of the discovered unexplored sensor spoofing threats. To address this limitation, we build an *action flow model* to abstract possible control flows in

different RV systems and scenarios. Each control flow can lead to some potential spoofing attacks. It is important to note that the action flow and attack path are different: action flow only describes an RV system while attack path is an extension of action flow with security considerations. More specifically, attack path is a higher-level abstraction of RV systems, helping distinguish the target states of each sensor spoofing attack. Action flow is specific to each sensor type and the function type inside the system, helping identify new unexplored attacks.

Figure 3 shows our action flow model, which consists of 12 interacted robot functions. This is established by surveying the function compositions and interactions of various RVs from both industrial ecosystems [149], [150] and academic papers [106], [112], [151]. Among them, sensors directly interact with five functions: *Objection Detection*, *Segmentation*, *Localization/SLAM*, *Speech Recognition* and *Distance Detection*. Then these functions further interact with other subsequent functions. First, *Objection Detection* recognizes nearby obstacles in real-time. Then *Object Tracking* assigns each predicted obstacle an ID and dynamically tracks it. Meanwhile, *Segmentation* estimates the static environmental information. Combining these two, *Environment Prediction* estimates the obstacle trajectories with probabilities/priorities. Second, *Localization/SLAM* estimates the operation states of the RV based on the known/unknown map. This information is the key to judge which obstacles in the surrounding environment have a high possibility of interacting with the RV. Third, *Speech Recognition* provides an interface for executing new tasks through human-computer interaction. *Goal Planning* transfers each task goal to a set of subtasks and dispatches them to other planning functions. For example, *Path Routing* calculates a path from the estimated position to the destination given by *Goal Planning*. *Parking/Landing Planning* plans a set of trajectories to complete the parking or landing task. Both the planned path and the trajectory will be sent to *Motion Planning*, which outputs the suitable velocity to avoid obstacles. Finally, *Motion Controller* transfers this velocity to each actuator and ensures the stability of the RV.

The five functions (*Objection Detection – Distance Detection*) are the main targets for the adversary to compromise the RV system. We define the action flow (ActFlow) as a complete flow from one sensor to a final control function[1]. Then we hypothesize that *each action flow could give rise to some sensor spoofing vectors*, where the adversary can tamper with the corresponding sensor data to affect RV's final actions. Table 3 in the appendix lists 44 possible action flows with the corresponding attack paths. A single-round action flow refers to the flow going from the perception to control stages once, while a multi-round action flow goes through the pipeline multiple times with the interaction between the RV and environment. We observe that some action flows have been exploited to launch sensor spoofing attacks, while most have not been investigated yet, leaving a large unknown attack surface for RV systems.

We obtain three interesting observations from the summary of these action flows and attack vectors. (1) Most existing works design the attacks heuristically, and they mainly focus on the immediate action damage from the mis-estimation of RV states. (2) Some action flows require specific function compositions and sensor types. They can only be launched against specific types of RVs. (3) Spoofing attacks based on multi-round action flows are rarely considered by prior works due to their complex mechanisms. Only three papers discussed these attacks. Below we give detailed analysis about each type of spoofing attacks.

1. An action flow (ActFlow) provides more information than an attack path in § 2.2 since it is at the granularity of robotic functions

4.2. Attack Vector Analysis

Some action flows may share the same composition and interactions of robotic functions, and only differ in the target sensors. The corresponding spoofing attacks will have common characteristics. To simplify the analysis, we treat these action flows as the same *flow pattern (FlowPtn)*. Then the 44 action flows lead to 14 different flow patterns.

Table 2 summarizes these attacks with different target sensors and spoofing techniques. Each attack falls into one of two cases: (1) there are already existing works realizing this attack. We list the references in the corresponding cells. Most of these works have been empirically verified and tested on simulators (e.g., Baidu Apollo [149]) or physical world. (2) There is no existing work exploring this attack, yet it is possible and realistic. In § 4.3, we further analyze and categorize them into three cases based on the attack feasibility (🔵/🟢/🔴). Below we discuss the mechanism of each flow pattern, while detailed attack scenarios are illustrated in Figure 4.

1) FlowPtn 1 (*Object Detection→Motion Planning→ Motion Controller*): This flow pattern contains the action flows of ActFlow1,3,10. They are responsible for avoiding obstacles and taking correct actions based on the traffic controller, such as traffic lights and signs. To achieve correct motion planning, the RV needs to obtain the accurate position and property of these traffic controllers. We summarize three possible attacks that can manipulate the objects and compromise the executions (Figure 4.FlowPtn1). (i) *Obstacle Appearing*: the RV mistakenly recognizes a non-existent obstacle in front. Then it brakes hard and stops at the intersection even if the traffic light is green. (ii) *Obstacle Missing*: the RV fails to detect an existing obstacle in front, and crash into it directly. (iii) *Traffic Controller Misclassification*: the RV misclassifies the traffic sign or traffic light into a wrong category and takes dangerous actions.

According to Table 2, we observe that a number of attack vectors have been realized in prior works while some have never been explored. Specifically, there is no related work focusing on obstacle appearing attack using shape manipulation, object placement and sticker pasting spoofers. Moreover, for the obstacle missing, one possible attack is to simply use radar absorbing material to implement. Recently, LiDAR is also used for traffic controller classification [152], but no work explores misclassification attack using LiDAR spoofers.

2) FlowPtn 2 (*Object Detection→Object Tracking→Env Prediction→Motion Planing→Motion Controller*): This pattern includes action flows ActFlow2,4,11. They focus

TABLE 2: Possible attacks in different flow patterns.

Flow Pattern	Possible Attacks (Target Function)	MMW Radar		LiDAR				Camera			GPS		IMU		Ultrasonic Sensor		Microphone	
				LP	SM	OP		SP	LP									
FP1	Obstacle Appearing (A1)	[82]–[84]	AF1	[6], [46]–[49]	?	?	AF3	?	[61], [64], [67], [68]	AF10								
	Obstacle Missing (A1)	?		[48]	[50]	[51]		[54], [57]–[60]	[62], [66]									
	Traffic Controller Misclassification (A1)			?	?	?		[52]–[56]	[61]–[65], [67]									
FP2	Trajectory Appearing (B)	[84]	AF2	?	?	?	AF4	[71]	?	AF11								
	Trajectory Missing (B)	?		?	?	?		[71]	?									
	Trajectory Altering (C1)	?		?	?	?		[71]	?									
FP3	Lane Altering (A2)							[69], [70]	[67]	AF12								
FP4	Deviating Position Altering (E)			?	?	?	AF8	?	?	AF16	[41], [43]	AF21	?	AF26				
FP5	Predicted Priority Altering (C1)			?	?	?	AF5	?	?	AF13	?	AF18	?	AF23				
FP6	Target Deviating Position Altering (D1)			?	?	?	AF6	?	?	AF14	[41], [42]	AF19	?	AF24				
	Loop Closure Failure (A3)			?	?	?		?	?		?		?					
FP7	Destabilizing Velocity Altering (F)			?	?	?	AF9	?	?	AF17	?	AF22	[74]–[77]	AF27				
	Destabilizing Position Altering (F)			?	?	?		?	[72]		?		?					
FP8	Specific Location Altering (D2)			?	?	?	AF7	?	?	AF15	[38]–[40]	AF20	?	AF25				
FP9	Obstacle Distance Altering (A5)														[81], [82]	AF31		
FP10	Lateral Distance Altering (A5)														?	AF32		
FP11, 12,13	Target Goal Generation (C2)																[79], [80]	AF28-30
FP14	Target Blurring (A3)			?	?	?	AF39-41	?	?	AF36-38	[44]	AF42-44	[78]	AF33-35				
	ROI Altering (A3)			?	?	?		?	?		[45]		?					

?/?/?: Unexplored attacks **LP/SM/OP**: Laser Projection/Shape Manipulation/Object Placement **SP/LP**: Sticker Pasting/Light Projection
A1: Object detection **A2**: Segmentation **A3**: Localization/SLAM **A4**: Speech recognition **A5**: Distance detection **B**: Object tracking
C1: Environment prediction **C2**: Goal planning **D1**: Path routing **D2**: Parking/Landing planning **E**: Motion planning **F**: Motion controller
FP: Flow Pattern **AF**: Action Flow

on tracking the dynamic targets and predicting their trajectories. Similar to Flow Pattern 1, we consider three possible attacks that manipulate the tracked target or predicted trajectory (Figure 4.FlowPtn2). (i) *Trajectory Appearing*: the adversary can fool the RV to assign a track id to a non-existent object and then brake hard to avoid hitting it. (ii) *Trajectory Missing*: the spoofer can make the victim RV loss the tracking target, which could shorten the safe distance and cause the RV to crash into the target. (iii) *Trajectory Altering*: the adversary alters the predicted trajectory of the target. This can also reduce the safe distance with potential vehicle crashes.

From Table 2, we can observe that recent work [84] has discussed the possibility of trajectory appearing using the radar spoofer and only one related work [71] has designed all these three attacks through the sticker pasting spoofer. However, the implementation using other sensor spoofers are never considered.

3) FlowPtn 3 (*Segmentation→Env Prediction→Motion Planning→Motion Controller*): This flow pattern includes ActFlow12, which identifies the road conditions (e.g., lanes) and guides RV's motion control. Thus, one possible attack is *Lane Altering*, which makes the RV identify wrong traffic lanes (Figure 4.FlowPtn3). Past works have demonstrated such attack using sticker pasting [69], [70] and light projection spoofing [67].

4) FlowPtn 4 (*Localization/SLAM→Motion Planning →Motion Controller*): This flow pattern (ActFlow8,16,21,26) assists the localization of the RV. For example, autonomous vehicles need to drive in the center of the lane for safety. The *Localization* function helps the vehicle determine if it is on the right track. Thus, one possible attack is *Deviating Position Altering*, which exploits this lateral deviation to cause mis-prediction of RV's location. As shown in Figure 4.FlowPtn4, the RV is spoofed to a position away from the lane. Due to the false position, the RV takes actions to move in the opposite direction and further drive off the road pavement or on the wrong ways. Table 2 summarizes that such attack has only been realized by the GPS spoofer [43]. Due to the different demands on the cost and scenario, some *Localization* functions only depend on LiDAR [153], camera [154] or IMU sensor [155], thus spoofing on these sensors can also achieve the same attack consequence.

5) FlowPtn 5 (*Localization/SLAM→Env Prediction→ Motion Planning→Motion Controller*): This pattern covers the action flows of ActFlow5,13,18,23. They predict whether the RV should take actions due to surrounding objects. The predicted outcome is one of three priorities: *ignore*, *caution* and *normal*. The first two indicate the object will not and will most likely affect RV's trajectory, while the last one indicates other conditions by default. Thus, one possible attack is *Predicted Priority Altering*, which misleads the RV from the caution priority to the ignore priority, thus taking dangerous actions subsequently (Figure 4.FlowPtn5). It can be launched by targeting LiDAR, camera, GPS or IMU sensors. There are no prior works realizing such attack vector.

6) FlowPtn 6 (*Localization/SLAM→Path Routing→ Motion Planning→Motion Controller*): These action flows (ActFlow6,14,19,24) control the RV to navigate from one location to another or map an unknown area, i.e., SLAM. Two possible attacks are introduced by generating wrong paths or causing failure of the mapping task: (i) *Target Deviating Position Altering*: the RV is guided to a wrong destination (Figure 4.FlowPtn6(a)); (ii) *Loop Closure Failure*: the RV fails to assert that it returns to a previously visited location so that the map cannot be correctly generated (Figure 4.FlowPtn6(b)).

From Table 2, the target deviation position altering attack has been realized in [42], which adopts the GPS spoofer to slightly shift RV's position to make the fake navigation route match the shape of the actual roads. As we discussed in FlowPtn 4, spoofing other sensors can achieve the same consequence as well, especially for indoor RVs without GPS sensors. There are no prior works realizing the loop closure failure attack. We can use the GPS spoofer to mislead the RV to a fake position when recognizing the previous map, or use other sensor spoofers

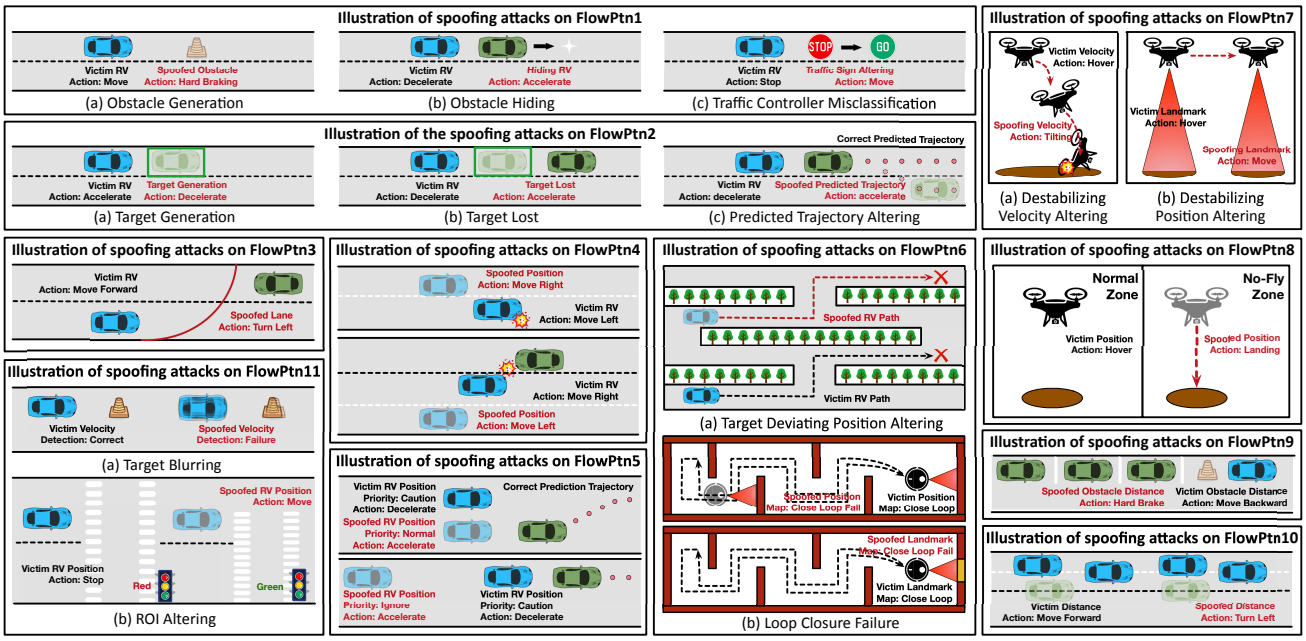

Figure 4: Illustration of spoofing attacks on flow pattern 1-11.

to directly modify the perceived object to mismatch the previous landmarks (§ 5.2).

7) FlowPtn 7 (*Localization/SLAM→Motion Controller*): These action flows (ActFlow9,17,22,27) aim to stabilize the RV based on the variance of the velocity or position. For example, drones need to dynamically adjust the fuselage to prevent overturning, or adjust its position to maintain hovering according to the changes of the ground below. Thus, two possible attacks can be proposed to destabilize the RV (Figure 4.FlowPtn7). (i) *Destabilizing Velocity Altering*: the adversary forges an angular velocity in a single direction and makes the RV overturned. (ii) *Destabilizing Position Altering*: the spoofer creates a continuous slight difference on the ground that misleads the RV to laterally move to a designated position.

As presented in Table 2, many works use the IMU spoofer to alter the destabilizing velocity [74]–[77]. Since the camera [156] or LiDAR [157] is also used to estimate odometry through matching the position of extracted features in two adjacent frames/point cloud data, spoofing these sensors is another way to achieve such attacks. To the best of our knowledge, GPS is rarely used as the odometry estimator. Davidson et al. [72] realized the destabilizing position altering attack on a drone using a light projection spoofer. If an adversary can slightly move a sticker pasting spoofer on the ground while the drones hovers, such attack can also be achieved. Besides, spoofing false LiDAR points or GPS signals can modify the hovering height.

8) FlowPtn 8 (*Localization/SLAM→Parking/Landing Planning→Motion Planning→Motion Controller*): This pattern covers the action flows of ActFlow7,15,20,25, which control specific RV tasks, such as parking or landing. To complete the action flow, the RV is provided with one target location. Thus, one possible attack is *Specific Location Altering*, where the adversary triggers the launch of these tasks at wrong time or locations. As shown in Figure 4.FlowPtn8, by spoofing a location to a no-fly zone, the victim drone is enforced to perform an immediate landing. Table 2 shows related works on such attack based on the GPS spoofer. If an adversary needs to implement such attack on the RV using LiDAR or camera for localization, he can construct a similar environment to cheat the loop-closure system by the LiDAR or camera spoofer. Besides, spoofing IMU to accumulate position errors can also achieve the same effect.

9) FlowPtn 9 (*Distance Detection→Parking/Landing Planning→Motion Planning→Motion Controller*): This flow pattern refers to ActFlow31, which ensures the safe distance between the RV and obstacles in the parking or landing tasks. Thus, one possible attack is *Obstacle Distance Altering*, which uses the ultrasonic sensor spoofer to shorten this distance and cause the RV to brake hard (Figure 4.FlowPtn9) [81], [82].

10) FlowPtn 10 (*Distance Detection→Motion Controller*): The action flow ActFlow32 is responsible for ensuring a lateral safe distance between the RV and nearby cars encroaching on its lane. Therefore, we propose a new attack: *Lateral Distance Altering*. As shown in Figure 4.FlowPtn10, by deploying many ultrasonic spoofers along the roadside, the vehicle needs to frequently change different directions to ensure it is safe within the designated road lane. This could make the autonomous driving less smooth, and annoy or even hurt the passengers. This attack is realizable but no prior works ever considered it (Table 2).

11) FlowPtn 11,12 and 13 (*Speech Recognition→Goal Planning→[Path Routing/Parking/Landing Planning: Motion Planning]/-→Motion Controller*): These three action flows focus on launching and performing various missions according to the user's voice commands. They are vulnerable to the *Target Goal Generation* attack, which triggers malicious missions using a microphone spoofer. It has been implemented in prior works [79], [80].

12) FlowPtn 14 (*Localization/SLAM→Motion Controller ➲ Object Detection*): Finally, we consider multi-round action flows[2]. There are two observations that lead to two attack vectors. (i) The quality of the perceived images

2. In this paper, we only consider two-round ActFlows. Attacks with more rounds are more complex, and will be investigated as future work.

or laser points highly depends on the stabilization of the RV. For example, the jitters of camera or LiDAR can cause blurred images and irregular distributions of point clouds. Hence, we have the *Target Blurring* attack, which spoofs the sensors to jitter the RV and cause the failure of object detection in LiDAR or camera (Figure 4.FlowPtn11(a)). (ii) The ROI depends on the current position of the RV. Therefore, we have the *ROI Altering* attack, which falsifies well-designed positions and alters the detected ROI. As shown in Figure 4.FlowPtn11(b), the vehicle mis-estimates the traffic signals based on the second traffic light rather than the first one due to the spoofed position.

As shown in Table 2, past works realize the target blurring attacks against the motion compensation mechanism in autonomous vehicles to blur the images [78] and point clouds [44]. For other RVs that do not adopt this compensation mechanism (e.g., indoor automated guided vehicles and drones), we can use other spoofers to cause jitters. Tang et al. [45] implemented an ROI altering attack with the GPS spoofer. Also, spoofing other location-related sensors can also achieve the same consequence.

4.3. Feasibility Analysis

With the action flow model, we have identified 77 unexplored threats that are not considered in prior works and have the potential to cause fatal accidents. Below we discuss the feasibility of these attacks from three cases.

C1. Learn from each other (✅): In LiDAR spoofing attacks, we summarize three techniques: laser projection, shape manipulation and object placement. They are actually able to achieve the same compromised state (i.e., spoofing the same points) by using an active LiDAR spoofer, creating and placing an adversarial object, respectively. As a result, a new attack based on one spoofing technique is also feasible if this attack has been realized with another technique. For example, in FlowPtn1, the implementation of obstacle appearing attack using LiDAR object placement can be inspired by the same attack with laser projection [6], [46]–[49]. Similarly, the two techniques for camera sensors (sticker pasting and light projection) can also target the same perception function, and their attacks have similar feasibility.

C2. Easy to prove (❓): (1) The obstacle missing attack (ActFlow1) and trajectory missing attack (ActFlow2) using radar can be simply implemented by using radar absorbing materials. Moreover, trajectory altering attack can also be implemented by combining existed appearing attacks. (2) Traffic controller misclassification attack using LiDAR (ActFlow3) can be easily transferred from camera. For example, an adversary can leverage a laser beam to change the semantic information of a traffic sign. (3) Loop closure failure attack (FlowPtn6.SLAM). The SLAM algorithm helps the RV recognize a visited location. By adding enough visual noise in the critical region of the environment, the RV will fail to detect the loop closure and then generate the wrong map. (4) Destabilizing velocity altering attacks based on LiDAR or camera spoofing (ActFlow9,17). The LiDAR- or camera-based odometer estimates RV's velocity based on the offset of the detected point clouds/pixels. As prior works [50], [51], [54], [57]–[60], [66] have shown that both two sensors can be spoofed to lose the target (FlowPtn1), these techniques can also cause wrong estimates of the odometer to realize these attacks. (5) Attacks based on GPS spoofing (ActFlow18,19,22). GPS spoofing is more mature than other spoofing attacks, thus predicted priority altering attack can be easily achieved (ActFlow18). Note that GPS is rarely used as the main sensor for mapping (ActFlow19) and odometry (ActFlow22). (6) Attacks based on IMU spoofing (ActFlow23,24,26,33-35). These attacks aim to alter RV's position. They can be realized following the destabilizing velocity altering attack using the IMU spoofer (ActFlow27), which can cause the accumulative error.

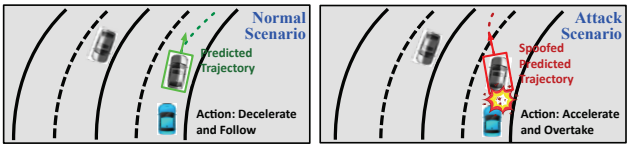

Figure 5: An example of obstacle position altering.

C3. Need to validate (❓): (1) Trajectory appearing, missing and altering attacks based on LiDAR spoofing (FlowPtn2, ActFlow4). Past work [71] only demonstrated a camera-based sticker pasting method for these attacks (ActFlow11) . It is unknown whether we can spoof the LiDAR to achieve the same result. (2) Attacks based on camera spoofing (ActFlow13,14,16,36-38) or LiDAR spoofing (ActFlow5,6,8,39-41). They are built on the assumption that the SLAM function can be fooled to relocalize the RV to a false position. To show their practicality, we design two novel methodologies and implement prototypes in the next section.

5. Two Novel Attack Methodologies

We present two new approaches to validate the feasibility of attack vectors in case 3 (§ 4.3). We believe each approach has sufficient technical novelty and contributions as an individual research project. Due to the page limit, we only describe the basic mechanism and evaluation. We expect researchers can extend them to design more spoofing attacks, since they are the basis of the 36 unexplored vectors in case 3.

5.1. Obstacle Position Altering (LiDAR)

RVs utilize state-of-the-art DNN models to interpret 3D point clouds captured by the LiDAR for object detection. In general, a 3D object detection model $\mathcal{M}$ extracts features from the input point cloud X, and outputs a set of bounding boxes $Y = \mathcal{M}(X)$. Each box is attached to a detected object with its location Y_{loc}, size Y_{size}, heading Y_h, and confidence score C of predicted categories. Boxes with the confidence score lower than a threshold will be filtered out, and the remaining box is the detected object.

The goal of our attack is to fool $\mathcal{M}$ to mis-recognize a moving object in a wrong location, and then the RV will mis-estimate the object's trajectory. Figure 5 shows an example. There is a running vehicle in front of the victim RV. The adversary can spoof the point clouds to mislead the victim that this vehicle was switching to the left lane. Then the RV will accelerate, and the safety distance will be reduced, which can cause car crash (FlowPtn2,4).

Figure 6: Position altering attack on PointRCNN.

Note this attack is different from existing LiDAR spoofing attacks [6], [46]–[51], which manipulate the existence of static obstacles. Our attack aims to alter the predicted trajectory of a moving object. Although the same attack goal has been realized by camera spoofing [71], our attack targets the LiDAR-based RVs (e.g., Baidu Apollo), and is technically more challenging due to the complex and non-differentiable feature of point cloud models.

Formally, we inject an adversarial object x^* into the original input X. The model output $Y^* = \mathcal{M}(X+x^*)$ has the same size and category as Y. However, its location Y^*_{loc} and heading Y^*_h are different. To craft a qualified x^*, we adopt a common object (e.g., drone) and try to find a malicious placement $s^* = (x^*, y^*, z^*, \alpha^*, \beta^*, \gamma^*)$ in the environment, where the first three variables denote its location, and the last three denote its orientation. s^* can maximize the distance between the predicted and original locations of the bounding box:

$$\max_{s^*} ||Y^*_{loc}, Y_{loc}|| \tag{1}$$

We can use an optimization method to identify s^*. The challenge is that the optimization objective is non-differentiable, and it is hard to calculate the gradient. To address this issue, we follow [158] to estimate the gradient as below:

$$\hat{\nabla}_{s^*} L_{\mathcal{M}}(s^*) = \frac{1}{m}\sum_{i=1}^{m} \frac{L_{\mathcal{M}}(s^* + \epsilon u_i) * u_i}{\epsilon} \tag{2}$$

where u_i is a random variable with a uniform distribution; ϵ is a positive smoothing factor; $L_{\mathcal{M}}$ is the loss function to quantify the MSE distance between Y^*_{loc} and Y_{loc}. m is a hyperparameter to control the gradient estimation.

Algorithm 1 details the optimization process. The adversary iteratively adjusts the location and orientation of the adversarial object to affect the bounding box of the target obstacle. In each iteration, we estimate the gradient with Eq. 2, and use Project Gradient Descent (PGD) to update the gradient.

Algorithm 1 Position Altering Attack

INPUT: clean point cloud X; # of attack iterations N; loss threshold θ; target object to be altered T; adversarial object x^*
OUTPUT: adversarial object placement s^*.

Initialize: s^*
for $j = 0; j < N; j++$ **do**
  $grad = 0; counter = 0; i = 0$
  while $i < m$ **do**
    $loss = L_{\mathcal{M}}(s^* + \epsilon * u_i)$
    if $loss > \theta$ **then**
      $i = i + 1$
      $grad = grad + (loss/\epsilon) * u_i$
    end if
    $grad = grad/m$
    $s^* += sign(grad) * \epsilon$
  end while
end for

It is worth noting that we focus on the trajectory altering attack in ActFlow4. There are also two attacks (trajectory appearing and missing) in this action flow. They can be achieved using the same method by changing the optimization objective adaptively: instead of maximizing the location distance, we can manipulate the confidence score C of the bounding boxes to create a non-existence target, or hide an existing target. Due to the page limit, we leave this as future work.

Figure 7: Failure of loop closure in the ORB-SLAM2 simulator.

Experiments. We choose PointRCNN [86], one of the most widely used LiDAR detection models. We use the popular KITTI dataset [85] to train this model, where the samples are collected by the Velodyne HDL-64E Lidar. We use a drone as the adversarial object due to its flexibility and stealthiness [51]. The location of the drone is restricted to a space of 3m * 3m * 1m around the target vehicle. We set $\epsilon = 0.1$, and s^* is initialized as random values between $[-1, 1]$.

Figure 6 shows the outcome of our position altering attack. The objective of this attack was to deceive the victim vehicle's LiDAR and cause a misestimation of the position of the front vehicle. This was achieved by utilizing a drone and iteratively optimizing its pose through 50 iterations using Algorithm 1. We can observe the PointRCNN model in the victim vehicle was successfully manipulated, leading to a wrong prediction of the position of the front vehicle shifting to the left. This incorrect estimation could result in the victim vehicle accelerating and overtaking, potentially causing a rear-end collision.

5.2. Location Altering (Camera/LiDAR)

SLAM is widely adopted in many RVs for localization and mapping. State-of-the-art SLAM algorithms take camera images or LiDAR point clouds as input. One key module in SLAM is loop closure, which checks if the RV has returned to a previously mapped region to reduce the uncertainty in the map estimation. Different algorithms share the similar idea in loop closure detection: the RV constructs the surroundings of the current location with the sensor data, and compares them with the established global map. A loop closure is detected when the similarity between the current environment and a part of the map is higher than a threshold. To attack the loop closure function, the adversary can modify the environment to increase the similarity between two distinct locations. Then the RV will mistakenly think a new location is visited before, and mis-estimate its location. This leads to a series of spoofing attacks (ActFlow5,6,8,13,14,16,36-41).

Experiments. We target ORB-SLAM2 [87], the most popular real-time SLAM algorithm in both academia and industry. We implement this algorithm in a simulated

environment with its default setting, and choose a mid-size city in KIITI [85] as our experimental scenario. During the SLAM process, the vehicle extracts the FAST keypoints [159] and ORB descriptors [160] from each frame. It calculates the similarities between the current and past frames based on their relative scale, position and angles of the keypoints. A loop closure is reported if this similarity exceeds a pre-defined threshold.

As shown in Figure 7, we try to cause false loop closure on two distinct locations: the current keyframe is a T-turn intersection, while the target keyframe is a straight road. We increase their similarity by two means: (1) moving certain physical objects from the target keyframe to the current keyframe (purple box); (2) adding some patches to both keyframes (yellow box). We hope to bring minimum changes to the environment, so we carefully select the objects and patches that are relatively small but contain many FAST keypoints. They are either printed out as patches to stick on the wall, or used as references to find the same objects and place them in the current frame. We maintain the relative scale and position of the selected objects in two scenes so that they can effectively contribute to the similarity calculation.

After adding the patches and objects, we get 34 pairs of matched keypoints between the current and target keyframes (divided into 3 groups based on the relative angles, connected by green, blue and red lines, respectively), which fulfill the ORB-SLAM2 loop closure requirements. The green boxes in the left side of Figure 7 denote the current positions of the autonomous vehicle in the map. When the vehicle reaches the intersection, it mistakenly detects a loop closure, and relocates itself to the position of the target frame. This verifies the feasibility of the location altering attack.

6. Discussion and Future Work

6.1. Attacks against Multi-Sensor Fusion

This paper mainly discusses spoofing attacks with single sensors. Modern RV systems start to integrate Multi-sensor Fusion (MSF) algorithms to smooth out errors and uncertainties of each single sensor and improve the perception robustness. As our action flow model is built upon the interaction among robotic functions, it can describe these sophisticated attacks against MSF as well. For instance, some MSF-based object detection functions use LiDAR data to assist the camera to discern the depth range [149], [161]–[163]. The adversary can use a single LiDAR spoofer to make the obstacle closer to the victim and cause collisions [164]. Cao et al. [50] observed that the shape of a 3D object can cause position changes in LiDAR point clouds as well as pixel value changes in camera images. Then they proposed a spoofing method to blind the MSF-based object detection by optimizing a 3D-printed obstacles. Both two attacks can be categorized as action flow AF3.

6.2. Spoofing Defenses

We focus on spoofing attacks, and the systematization of defense solutions is not covered in this paper. Generally, defense works can be classified into two categories: (1) Detection: the defender aims to identify the existence of the spoofing activities [92] or fake signals [165]. (2) Prevention: the defender tries to correct the spoofed data or state by filtering [166], randomization [47], fusion [81], etc. Interested readers can refer to [17] for more details.

An interesting direction is to leverage our action flow model to build a unified defense. Since the model depicts the interactions between RV functions, it can also identify the key to mitigate corresponding attacks. Each flow pattern consists of multiple functions, and each function may be subject to a spoofing attack. So we can design methodologies for system monitoring at the function level, and combine them to detect anomalies in different flows. Our proposed defensive scheme focuses on using the interaction between different action flows to detect attacks, while the scheme in [17] relies on the feature of each victim function for monitoring and prevention, such as adaptive filtering and randomization. We leave this as future work.

6.3. More Evaluations

In § 5 we present the feasibility and preliminary results of two proposed attack methodologies. They can be further extended in the following ways. (1) We can optimize the attack designs. For instance, for the obstacle position altering attack, we can test more common objects to generate the fake points. For the location altering attack, we can try to reduce the number of injected obstacles and patches when increasing the similarity. (2) We give some examples to show the success of the attacks. More quantitative evaluations and comparisons can better demonstrate their effectiveness. (3) We evaluate the attacks with the dataset and simulator. Physical experiments will make the results more convincing. In the future, we will improve the attacks from these aspects, and also realize other attack vectors analyzed in § 4.3.

7. Conclusion

In this paper, we first systematize the knowledge of sensor spoofing attacks against RV systems. Then an action flow model is introduced to describe existing attacks and predict new attack vectors. Our model and analysis can benefit RV researchers and practitioners in understanding the unexplored sensor spoofing threats, and inspecting their designs. We also propose two new attack methodologies against the trajectory tracking and loop closure detection. We expect these methodologies can inspire researchers to improve our identified attack vectors, and design the corresponding defenses.

Acknowledgements

We sincerely thank our shepherd, Dr. Tom Chothia, and the anonymous reviewers for their valuable comments on this paper. This work is supported under the RIE2020 Industry Alignment Fund-Industry Collaboration Projects (IAF-ICP) Funding Initiative, as well as cash and in-kind contributions from the industry partner(s). It is also supported in part by NTU-DESAY SV Research Program under Grant 2018-0980 and Singapore Ministry of Education (MOE) AcRF Tier 1 RG108/19 (S).

References

[1] Y. Xu, T. Zhang, J. Han, S. Wang, and Y. Bao, "Towards practical cloud offloading for low-cost ground vehicle workloads," in *IEEE International Parallel and Distributed Processing Symposium*, 2021.

[2] "Why tesla's autopilot can't see a stopped firetruck," https://www.wired.com/story/tesla-autopilot-why-crash-radar, 2018.

[3] "Uber self-driving car crash: What really happened," https://www.forbes.com/sites/meriameberboucha/2018/05/28/uber-self-driving-car-crash-what-really-happened, 2018.

[4] "Sensor cited as potential factor in boeing crashes draws scrutiny," https://www.washingtonpost.com/business/economy/sensor-cited-as-potential-factor-in-boeing-crashes-draws-scrutiny/2019/03/17/5ecf0b0e-4682-11e9-aaf8-4512a6fe3439_story.html, 2019.

[5] "Agencies warn on satellite hacks & gps jamming affecting airplanes, critical infrastructure," https://threatpost.com/agencies-satellite-hacks-gps-jamming-airplanes-critical-infrastructure/178993, 2022.

[6] 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 *ACM Conference on Computer and Communications Security*, 2019.

[7] C. Günther, "A survey of spoofing and counter-measures," *PeerJ Computer Science*, 2014.

[8] M. L. Psiaki and T. E. Humphreys, "Gnss spoofing and detection," *Proceedings of the IEEE*, 2016.

[9] D. A. Schmidt, K. Radke, S. A. Çamtepe, E. Foo, and M. Ren, "A survey and analysis of the gnss spoofing threat and countermeasures," *ACM Computing Surveys*, 2016.

[10] Z. Wu, Y. Zhang, Y. Yang, C. Liang, and R. Liu, "Spoofing and anti-spoofing technologies of global navigation satellite system: A surveys," *IEEE Access*, 2020.

[11] S. Z. Khan, M. Mohsin, and W. Iqbal, "On gps spoofing of aerial platforms: a review of threats, challenges, methodologies, and future research directions," *PeerJ Computer Science*, 2021.

[12] N. Akhtar and A. S. Mian, "hreat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, 2018.

[13] A. Chakraborty, M. Alam, V. Dey, A. Chattopadhyay, and D. Mukhopadhyay, "Adversarial attacks and defences: A survey," in *CoRR abs/1810.00069*, 2018.

[14] L. Sun, M. Tan, and Z. Zhou, "A survey of practical adversarial example attacks," *Computer Science Review*, 2018.

[15] X. Huang, D. Kroening, W. Ruan, J. Sharp, Y. Sun, E. Thamo, M. Wu, and X. Yi, "A survey of safety and trustworthiness of deep neural networks: Verification, testing, adversarial attack and defence, and interpretability," *Computer Science Review*, 2020.

[16] Z. Wu, N. W. D. Evans, T. Kinnunen, J. Yamagishi, F. Alegre, and H. Li, "Spoofing and countermeasures for speaker verification: A survey," *Speech Communication*, 2015.

[17] C. Yan, H. Shin, C. Bolton, W. Xu, Y. Kim, and K. Fu, "Sok: A minimalist approach to formalizing analog sensor security," in *IEEE Symposium on Security and Privacy*, 2021.

[18] N. Nighswander, B. M. Ledvina, J. Diamond, R. Brumley, and D. Brumley, "Gps software attacks," in *ACM Conference on Computer and Communications Security*, 2012.

[19] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful gps spoofing attacks," in *ACM Conference on Computer and Communications Security*, 2011.

[20] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "Assessing the spoofing threat: Development of a portable gps civilian spoofer," in *Proceedings of the Institute of Navigation GNSS*, 2008.

[21] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *International Conference on Learning Representations*, 2017.

[22] S. Castro, R. Dean, G. Roth, G. T. Flowers, and B. Grantham, "Influence of acoustic noise on the dynamic performance of mems gyroscopes," *ASME International Mechanical Engineering Congress and Exposition*, 2007.

[23] Y. Tu, Z. Lin, I. Lee, and X. Hei, "On the degradation of mems gyroscope performance in the presence of high power acoustic noise," in *IEEE International Symposium on Industrial Electronics*, 2007.

[24] R. N. Dean, S. T. Castro, G. T. Flowers, G. Roth, A. Ahmed, A. S. Hodel, B. E. Grantham, D. A. Bittle, and J. James P. Brunsch, "A characterization of the performance of a mems gyroscope in acoustically harsh environments," *IEEE Transactions on Industrial Electronics*, 2011.

[25] G. Zhang, C. Yan, X. Ji, T. Zhang, T. Zhang, and W. Xu, "Dolphinattack: Inaudible voice commands," in *ACM Conference on Computer and Communications Security*, 2017.

[26] N. Roy, H. Hassanieh, and R. R. Choudhury, "Backdoor: Making microphones hear inaudible sounds," in *ACM SIGMOBILE International Conference on Mobile Systems, Applications, and Services*, 2017.

[27] L. Song and P. Mittal, "Poster: Inaudible voice commands," in *ACM Conference on Computer and Communications Security*, 2017.

[28] N. Roy, S. Shen, H. Hassanieh, and R. R. Choudhury, "Inaudible voice commands: The long-range attack and defense," in *Symposium on Networked Systems Design and Implementation*, 2018.

[29] Q. Yan, K. Liu, Q. Zhou, H. Guo, and N. Zhang, "Inaudible voice commands: The long-range attack and defense," in *Network and Distributed System Security Symposium*, 2020.

[30] N. Carlini, P. Mishra, T. Vaidya, Y. Zhang, M. Sherr, C. Shields, D. A. Wagner, and W. Zhou, "Hidden voice commands," in *USENIX Security Symposium*, 2016.

[31] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, "Commandersong: A systematic approach for practical adversarial voice recognition," in *USENIX Security Symposium*, 2018.

[32] N. Carlini and D. A. Wagner, "Audio adversarial examples: Targeted attacks on speech-to-text," in *IEEE Symposium on Security and Privacy Workshops*, 2018.

[33] L. Schönherr, K. Kohls, S. Zeiler, T. Holz, and D. Kolossa, "Adversarial attacks against automatic speech recognition systems via psychoacoustic hiding," in *Network and Distributed System Security Symposium*, 2019.

[34] Y. Chen, J. Z. Xuejing Yuan and, Y. Zhao, S. Zhang, K. Chen, and X. Wang, "Devil's whisper: A general approach for physical adversarial attacks against commercial black-box speech recognition devices," in *USENIX Security Symposium*, 2020.

[35] Y. Qin, N. Carlini, G. W. Cottrell, I. J. Goodfellow, and C. Raffel, "Imperceptible, robust, and targeted adversarial examples for automatic speech recognition," in *International Conference on Machine Learning*, 2019.

[36] R. Chauhan, "A platform for false data injection in frequency modulated continuous wave radar," Ph.D. dissertation, Utah State University, 5 2014.

[37] H. Xu, A. Ju, and D. A. Wagner, "Model-agnostic defense for lane detection against adversarial attack," in *CoRR abs/2103.00663*, 2021.

[38] V. Dey, V. Pudi, A. Chattopadhyay, and Y. Elovici, "Security vulnerabilities of unmanned aerial vehicles and countermeasures: An experimental study," in *International Conference on VLSI Design*, 2018.

[39] D. He, H. Liu, S. Chan, and M. Guizani, "How to govern the non-cooperative amateur drones?" *IEEE Network*, 2019.

[40] D. He, Y. Qiao, S. Chen, X. Du, W. Chen, S. Zhu, and M. Guizan, "A friendly and low-cost technique for capturing non-cooperative civilian unmanned aerial vehicles," *IEEE Network*, 2019.

[41] H. Sathaye, M. Strohmeier, V. Lenders, and A. Ranganathan, "An experimental study of gps spoofing and takeover attacks on uavs," in *USENIX Security Symposium*, 2022.

[42] K. C. Zeng, S. Liu, Y. Shu, D. Wang, H. Li, Y. Dou, G. Wang, and Y. Yang, "All your gps are belong to us: Towards stealthy manipulation of road navigation systems," in *USENIX Security Symposium*, 2018.

[43] S. Liu, X. Cheng, H. Yang, Y. Shu, X. Weng, K. C. Z. Ping Guo, G. Wang, and Y. Yang, "Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under gps spoofing," in *USENIX Security Symposium*, 2020.

[44] Y. Li, C. Wen, F. Juefei-Xu, and C. Feng, "Fooling lidar perception via adversarial trajectory perturbation," in *IEEE International Conference on Computer Vision*, 2021.

[45] K. Tang, J. Shen, and Q. A. Chen, "Fooling perception via location: A case of region-of-interest attacks on traffic light detection in autonomous driving," in *NDSS Workshop on Automotive and Autonomous Vehicle Security*, 2021.

[46] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," in *Black Hat Europe*, 2015.

[47] H. Shin, D. Kim, Y. Kwon, and Y. Kim, "Illusion and dazzle: Adversarial optical channel exploits against lidars for automotive applications," in *Cryptographic Hardware and Embedded Systems*, 2017.

[48] Z. Jin, X. Ji, Y. Cheng, B. Yang, C. Yan, and W. Xu, "Pla-lidar: Physical laser attacks against lidar-based 3d object detection in autonomous vehicle," in *IEEE Symposium on Security and Privacy*, 2023.

[49] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *USENIX Security Symposium*, 2020.

[50] Y. Cao, N. Wang, C. Xiao, D. Yang, J. Fang, R. Yang, Q. A. Chen, M. Liu, and B. Li, "Invisible for both camera and lidar: Security of multi-sensor fusion based perception in autonomous driving under physical-world attacks," in *IEEE Symposium on Security and Privacy*, 2021.

[51] Y. Zhu, C. Miao, T. Zheng, F. Hajiaghajani, L. Su, and C. Qiao, "Can we use arbitrary objects to attack lidar perception in autonomous driving?" in *ACM Conference on Computer and Communications Security*, 2021.

[52] K. Eykholt, I. Evtimov, E. Fernandes, A. R. Bo Li, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Computer Vision and Pattern Recognition*, 2018.

[53] D. Song, K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, F. Tramèr, A. Prakash, and T. Kohno, "Physical adversarial examples for object detectors," in *Workshop on Offensive Technologies*, 2018.

[54] Y. Zhao, H. Zhu, R. Liang, Q. Shen, S. Zhang, and K. Chen, "Seeing isn't believing: Towards more robust adversarial attack against real world object detectors," in *ACM Conference on Computer and Communications Security*, 2019.

[55] Z. Kong, J. Guo, A. Li, and C. Liu, "Physgan: Generating physical-world-resilient adversarial examples for autonomous driving," in *Computer Vision and Pattern Recognition*, 2020.

[56] J. Wang, A. Liu, Z. Yin, S. Liu, S. Tang, and X. Liu, "Dual attention suppression attack: Generate adversarial camouflage in physical world," in *Computer Vision and Pattern Recognition*, 2021.

[57] Z. Cheng, J. Liang, H. Choi, G. Tao, Z. Cao, D. Liu, and X. Zhang, "Physical attack on monocular depth estimation with optimal adversarial patche," in *European Conference on Computer Vision*, 2022.

[58] L. Huang, C. Gao, Y. Zhou, C. Xie, A. L. Yuille, C. Zou, and N. Liu, "Universal physical camouflage attacks on object detectors," in *Computer Vision and Pattern Recognition*, 2020.

[59] Z. Wu, S.-N. Lim, L. S. Davis, and T. Goldstein, "Making an invisibility cloak: Real world adversarial attacks on object detectors," in *European Conference on Computer Vision*, 2020.

[60] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *European Conference on Computer Vision*, 2020.

[61] W. Wang, Y. Yao, X. Liu, X. Li, P. Hao, and T. Zhu, "I can see the light: Attacks on autonomous vehicles using invisible lights," in *ACM Conference on Computer and Communications Security*, 2021.

[62] Y. Man and M. Li, "Ghostimage: Remote perception attacks against camera-based image classification systems," in *International Symposium on Recent Advances in Intrusion Detection*, 2020.

[63] C. Yan, Z. Xu, Z. Yin, X. Ji, and W. Xu, "Rolling colors: Adversarial laser exploits against traffic light recognition," *arXiv preprint arXiv:2204.02675*, 2022.

[64] B. Nassi, Y. Mirsky, D. Nassi, R. Ben-Netanel, O. Drokin, and Y. Elovici, "Phantom of the adas: Securing advanced driver-assistance systems from split-second phantom attacks," in *ACM Conference on Computer and Communications Security*, 2020.

[65] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang, "Adversarial laser beam: Effective physical-world attack to dnns in a blink," in *Computer Vision and Pattern Recognition*, 2021.

[66] G. Lovisotto, H. Turner, I. Sluganovic, M. Strohmeier, and I. Martinovic, "Slap: Improving physical adversarial examples with short-lived adversarial perturbations," in *USENIX Security Symposium*, 2021.

[67] B. Nassi, D. Nassi, R. Ben-Netanel, Y. Mirsky, O. Drokin, and Y. Elovici, "Phantom of the adas: Phantom attacks on driver-assistance systems," *IACR Cryptology ePrint Archive*, 2020.

[68] C. Zhou, Q. Yan, Y. Shi, and L. Sun, "Doublestar: Long-range attack towards depth estimation based obstacle avoidance in autonomous systems," in *USENIX Security Symposium*, 2022.

[69] P. Jing, Q. Tang, Y. Du, L. Xue, X. Luo, T. Wang, S. Nie, and S. Wu, "Too good to be safe: Tricking lane detection in autonomous driving with crafted perturbations," in *USENIX Security Symposium*, 2021.

[70] T. Sato, J. Shen, N. Wang, Y. Jia, X. Lin, and Q. A. Chen, "Dirty road can attack: Security of deep learning based automated lane centering under physical-world attack," in *USENIX Security Symposium*, 2021.

[71] Y. Jia, Y. Lu, J. Shen, Q. A. Chen, H. Chen, Z. Zhong, and T. Wei, "Fooling detection alone is not enough: Adversarial attack against multiple object tracking," in *International Conference on Learning Representations*, 2020.

[72] D. Davidson, H. Wu, R. Jellinek, V. Singh, and T. Ristenpart, "Controlling uavs with sensor input spoofing attacks," in *Workshop on Offensive Technologies*, 2016.

[73] Y. Son, H. Shin, D. Kim, Y.-S. Park, J. Noh, K. Choi, J. Choi, and Y. Kim, "Rocking drones with intentional sound noise on gyroscopic sensors," in *USENIX Security Symposium*, 2015.

[74] T. Trippel, O. Weisse, W. Xu, P. Honeyman, and K. Fu, "Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks," in *European Symposium on Security and Privacy*, 2017.

[75] Z. Wang, K. Wang, B. yang, S. Li, and A. Pan, "Sonic gun to smart devices," in *Black Hat USA*, 2017.

[76] S. Nashimoto, D. Suzuki, T. Sugawara, and K. Sakiyama, "Sensor con-fusion: Defeating kalman filter in signal injection attack," in *ACM Asia Conference on Computer and Communications Security*, 2018.

[77] Y. Tu, Z. Lin, I. Lee, and X. Hei, "Injected and delivered: Fabricating implicit control over actuation systems by spoofing inertial sensors," in *USENIX Security Symposium*, 2018.

[78] X. Ji, Y. Cheng, Y. Zhang, K. Wang, C. Yan, W. Xu, and K. Fu, "Poltergeist: Acoustic adversarial machine learning against cameras and computer vision," in *IEEE Symposium on Security and Privacy*, 2021.

[79] M. Zhou, Z. Qin, X. Lin, S. Hu, Q. Wang, and K. Ren, "Hidden voice commands: Attacks and defenses on the vcs of autonomous driving cars," *IEEE Wireless Communications*, 2019.

[80] C. Yan, G. Zhang, X. Ji, T. Zhang, T. Zhang, and W. Xu, "The feasibility of injecting inaudible voice commands to voice assistants," *IEEE Transactions on Dependable and Secure Computing*, 2021.

[81] W. Xu, C. Yan, W. Jia, X. Ji, and J. Liu, "Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicle," *IEEE Internet of Things Journal*, 2018.

[82] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," in *Def Con*, 2016.

[83] N. Miura, T. Machida, K. Matsuda, M. Nagata, S. Nashimoto, and D. Suzuki, "A low-cost replica-based distance-spoofing attack on mmwave fmcw rada," in *ASHES@CCS*, 2019.

[84] Z. Sun, S. Balakrishnan, L. Su, A. Bhuyan, P. Wang, and C. Qia, "Who is in control? practical physical layer attack and defense for mmwave-based sensing in autonomous vehicles," *IEEE Transactions on Information Forensics and Security*, 2021.

[85] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Computer Vision and Pattern Recognition*, 2012.

[86] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Computer Vision and Pattern Recognition*, 2019.

[87] R. Mur-Artal and J. D. Tardós, "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras," *IEEE Transactions on Robotics*, 2017.

[88] "List of sensors," https://en.wikipedia.org/wiki/List_of_sensors/, 2021.

[89] Y. Tu, S. Rampazzi, B. Hao, A. Rodriguez, K. Fu, and X. Hei, "Trick or heat?: Manipulating critical temperature-based control systems using rectification attacks," in *ACM Conference on Computer and Communications Security*, 2019.

[90] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study," in *USENIX Security Symposium*, 2010.

[91] Y. Tu, V. S. Tida, Z. Pan, and X. Hei, "Transduction shield: A low-complexity method to detect and correct the effects of emi injection attacks on sensors," in *ACM Asia Conference on Computer and Communications Security*, 2021.

[92] Y. Shoukry, P. Martin, Y. Yona, S. N. Diggavi, and M. B. Srivastava, "Pycra: Physical challenge-response authentication for active sensors under spoofing attacks," in *ACM Conference on Computer and Communications Security*, 2015.

[93] Y. Shoukry, P. D. Martin, P. Tabuada, and M. B. Srivastava, "Non-invasive spoofing attacks for anti-lock braking systems," in *Conference on Cryptographic Hardware and Embedded Systems*, 2013.

[94] H. Abdullah, K. Warren, V. Bindschaedler, N. Papernot, and P. Traynor, "Sok: The faults in our asrs: An overview of attacks against automatic speech recognition and speaker identification systems," in *IEEE Symposium on Security and Privacy*, 2021.

[95] Y.-S. Park, Y. Son, H. Shin, D. Kim, and Y. Kim, "This ain't your dose: Sensor spoofing attack on medical infusion pump," in *Workshop on Offensive Technologies*, 2016.

[96] Z. Cheng, R. West, and C. Einstein, "End-to-end analysis and design of a drone flight controller," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 2018.

[97] A. Tampuu, M. Semikin, N. Muhammad, D. Fishman, and T. Matiisen, "A survey of end-to-end driving: Architectures and training methods," *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[98] T. Westbrook, "Who is in control? practical physical layer attack and defense for mmwave-based sensing in autonomous vehicles," *Journal of Strategic Security*, 2019.

[99] M. B. Polona Pavlovcic Preseren, Franc Dimc, "A comparative analysis of the response of gnss receivers under vertical and horizontal l1/e1 chirp jamming," *Sensors*, 2021.

[100] M. Sun, A. Al-Hashimi, M. Li, and R. M. Gerdes, "Impacts of constrained sensing and communication based attacks on vehicular platoon," *IEEE Transactions on Vehicular Technology*, 2020.

[101] D. Orlando, "A novel noise jamming detection algorithm for radar applications," *IEEE Signal Processing Letters*, 2016.

[102] M. Greco, F. Gini, and A. Farina, "Radar detection and classification of jamming signals belonging to a cone class," *IEEE Signal Processing Letters*, 2008.

[103] E. Axell, F. M. Eklöf, P. Johansson, M. Alexandersson, and D. M. Akos, "Jamming detection in gnss receivers: Performance evaluation of field trials," *NAVIGATION: Journal of the Institute of Navigation*, 2015.

[104] M. M. E. Moussa, A. Osman, M. Tamazin, M. Korenberg, A. Noureldin, and N. R. Group, "Jamming detection in gnss receivers: Performance evaluation of field trials," *GPS Solutions*, 2017.

[105] S. Lee, W. Choi, and D. H. Le, "Securing ultrasonic sensors against signal injection attacks based on a mathematical model," *IEEE Access*, 2019.

[106] Y. Xu, T. Zhang, and Y. Bao, "Analysis and mitigation of function interaction risks in robot apps," in *International Symposium on Recent Advances in Intrusion Detection*, 2021.

[107] D. Quarta, M. Pogliani, M. Polino, F. Maggi, A. M. Zanchettin, and S. Zanero, "An experimental security analysis of an industrial robot controller," in *IEEE Symposium on Security and Privacy*, 2017.

[108] S. Kulandaivel, S. Jain, J. Guajardo, and V. Sekar, "Cannon: Reliable and stealthy remote shutdown attacks via unaltered automotive microcontrollers," in *IEEE Symposium on Security and Privacy*, 2021.

[109] G. Bloom, "Weepingcan: A stealthy can bus-off attack," in *NDSS Workshop on Automotive and Autonomous Vehicle Security*, 2021.

[110] R. Bhatia, V. Kumar, K. Serag, Z. B. Celik, M. Payer, and D. Xu, "Evading voltage-based intrusion detection on automotive can," in *Network and Distributed System Security Symposium*, 2021.

[111] X. Han, G. Xu, Y. Zhou, X. Yang, J. Li, and T. Zhang, "Clean-annotation backdoor attack against lane detection systems in the wild," in *CoRR abs/2203.00858*, 2022.

[112] G. Deng, Y. Zhou, Y. Xu, T. Zhang, and Y. Liu, "An investigation of byzantine threats in multi-robot systems," in *International Symposium on Recent Advances in Intrusion Detection*, 2021.

[113] H. Kim, M. O. Ozmen, A. Bianchi, Z. B. Celik, and D. Xu, "Pgfuzz: Policy-guided fuzzing for robotic vehicles," in *Network and Distributed System Security Symposium*, 2021.

[114] M. Luo, A. C. Myers, and G. E. Suh, "Stealthy tracking of autonomous vehicles with cache side channels," in *USENIX Security Symposium*, 2020.

[115] D. Sathyamoorthy, "Security, privacy, and safety aspects of civilian drones: A survey," *J. Def. Secur*, 2015.

[116] A. Luo, "Drones hijacking - multi-dimensional attack vectors and countermeasures," in *Def Con 24*, 2016.

[117] V. Chang, P. Chundury, and M. Chetty, "Spiders in the sky: User perceptions of drones, privacy, and security," in *ACM Conference on Human Factors in Computing Systems*, 2017.

[118] R. AlTawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Transactions on Cyber-Physical Systems*, 2017.

[119] C. Cerrudo and L. Apa, "Hacking robots before skynet," IOActive Website, Tech. Rep., 2017.

[120] B. Nassi, R. Bitton, R. Masuoka, A. Shabtai, and Y. Elovici, "Sok: Security and privacy in the age of commercial drones," in *IEEE Symposium on Security and Privacy*, 2021.

[121] Y. Zhi, Z. Fu, X. Sun, and J. Yu, "Security and privacy issues of uav: A survey," *Mobile Networks and Applications*, 2020.

[122] L. Li, K. Qu, and K.-Y. Lin, "A survey on attack resilient of uav motion planning," in *International Conference on Control and Automation*, 2020.

[123] F. B. Sorbelli, M. Conti, C. M. Pinotti, and G. Rigoni, "Uavs path deviation attacks: Survey and research challenges," in *SECON Workshops*, 2020.

[124] R. AlTawy and A. M. Youssef, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[125] F. Arena, G. Pau, and M. Collotta, "A survey on driverless vehicles: from their diffusion to security features," *Journal of Internet Services and Information Security*, 2018.

[126] A. Singandhupe and H. M. L, "A review of slam techniques and security in autonomous driving," in *International Conference on Robotic Computing*, 2019.

[127] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future direction," *Proceedings of the IEEE*, 2020.

[128] A. Chowdhury, G. C. Karmakar, J. Kamruzzaman, A. Jolfaei, and R. Das, "Attacks on self-driving cars and their countermeasures: A survey," *IEEE Access*, 2020.

[129] A. Qayyum, M. Usama, J. Qadir, and A. I. Al-Fuqaha, "Securing connected & autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Communications Surveys and Tutorials*, 2020.

[130] K. Kim, J. S. Kim, S. Jeong, J.-H. Park, and H. K. Kim, "Cybersecurity for autonomous vehicles: Review of attacks and defense," *Computers & Security*, 2021.

[131] C. Gao, G. Wang, W. Shi, Z. Wang, and Y. Chen, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Internet of Things Journal*, 2021.

[132] Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, and Q.-L. Han, "Deep learning-based autonomous driving systems: A survey of attacks and defenses," *IEEE Transactions on Industrial Informatics*, 2021.

[133] J. Shen, N. Wang, Z. Wan, Y. Luo, T. Sato, Z. Hu, X. Zhang, S. Guo, Z. Zhong, K. Li, Z. Zhao, C. Qiao, and Q. A. Chen, "Sok: On the semantic ai security in autonomous driving," in *CoRR abs/2203.05314*, 2022.

[134] J. V. Carroll, "Vulnerability assessment of the transportation infrastructure relying on global positioning system," Volpe National Transportation Systems, Tech. Rep., 2001.

[135] J. S. Warner and R. G. Johnston, "A simple demonstration that the global positioning system (gps) is vulnerable to spoofing," *Journal of security administration*, 2002.

[136] "Gsg 56 series gpsgnss simulators," https://www.orolia.com/product/gsg-5-6-series-gps-gnss-simulators/, 2021.

[137] F. V. Graas and A. Soloviev, "Precise velocity estimation using a stand-alone gps receiver," *Navigation*, 2004.

[138] W. Ding and J. Wang, "Precise velocity estimation with a stand-alone gps receiver," *The Journal of Navigation*, 2011.

[139] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Computer Vision and Pattern Recognition*, 2022.

[140] L. Zhai, F. Juefei-Xu, Q. Guo, X. Xie, L. Ma, W. F. a d Shengchao Qin, and Y. Liu, "It's raining cats or dogs? adversarial rain attack on dnn perception," *arXiv preprint arXiv:2009.09205*, 2020.

[141] "Lrad 45oxl datasheet," https://genasys.com/wp-content/uploads/LRAD-450XL_D00101-Rev.-B_3-4.pdf, 2021.

[142] "Ultraelectronics hypershield datasheet," https://www.ultra-hyperspike.com/media/1214/b31cb6a3230bcf76e0e3a3f7d60f57b6-hs-18_rahd_data_sheet_-_92027a.pdf, 2021.

[143] X. Wang, Y. Wu, and W. Xu, "Windcompass: Determine wind direction using smartphones," in *Annual IEEE International Conference on Sensing, Communication, and Networking*, 2016.

[144] "Tesla voice commands," https://www.tesla.com/support/voice-commands/, 2021.

[145] "Audi mmi 3g voice recognition commands," https://www.audiworld.com/forums/attachments/q7-discussion-112/8098d1269544803-voice-command-mmi-3g-vr-commands.pdf, 2021.

[146] "Lincoln voice commands," https://www.lincoln.com/support/how-tos/sync/sync-with-voice-activated-navigation/voice-commands-by-category-sync-with-voice-activated-navigation/, 2021.

[147] Y. Xu, G. Deng, T. Zhang, H. Qiu, and Y. Bao, "Novel denial-of-service attacks against cloud-based multi-robot systems," *NAVIGATION: Journal of the Institute of Navigation*, 2021.

[148] B. Nassi, R. Ben-Netanel, A. Shamir, and Y. Elovici, "Drones' cryptanalysis - smashing cryptography with a flicker," in *IEEE Symposium on Security and Privacy*, 2019.

[149] "Baidu apollo project," https://github.com/ApolloAuto/apollo/, 2021.

[150] "Dji onboard sdk ros," https://github.com/dji-sdk/Onboard-SDK-ROS/, 2021.

[151] B. Boroujerdian, H. Genc, S. Krishnan, W. Cui, A. Faust, and V. J. Reddi, "Mavbench: Micro aerial vehicle benchmarking," in *Annual IEEE/ACM International Symposium on Microarchitecture*, 2018.

[152] S. Weng, J. Li, Y. Chen, and C. Wang, "Road traffic sign detection and classification from mobile lidar point clouds," *2nd ISPRS International Conference on Computer Vision in Remote Sensing (CVRS 2015)*, 2016.

[153] W. Lu, Y. Zhou, G. Wan, S. Hou, and S. Song, "L3-net: Towards learning based lidar localization for autonomous driving," in *Computer Vision and Pattern Recognition*, 2019.

[154] H. Lategahn and C. Stiller, "Vision-only localization," *IEEE Transactions on Intelligent Transportation Systems*, 2014.

[155] J. Yi, J. Zhang, D. Song, and S. Jayasuriya, "Imu-based localization and slip estimation for skid-steered mobile robots," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2007.

[156] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, "Vision-based odometry and slam for medium and high altitude flying uavs," *Journal of Intelligent and Robotic Systems*, 2009.

[157] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014.

[158] S. Kariyappa, A. Prakash, and M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-order gradient estimation," in *Computer Vision and Pattern Recognition*, 2021.

[159] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *European Conference on Computer Vision*, 2006.

[160] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski, "Orb: An efficient alternative to sift or surf," in *IEEE International Conference on Computer Vision*, 2011.

[161] Z. Wang and K. Jia, "Frustum convnet: Sliding frustums to aggregate local point-wise features for amodal 3d object detection," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2019.

[162] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *IEEE/RJS International Conference on Intelligent Robots and Systems*, 2018.

[163] T. Huang, Z. Liu, X. Chen, and X. Bai, "Epnet: Enhancing point features with image semantics for 3d object detection," in *European Conference on Computer Vision*, 2020.

[164] R. S. Hallyburton, Y. Liu, Y. Cao, Z. M. Mao, and M. Pajic, "Security analysis of camera-lidar fusion against black-box attacks on autonomous vehicles," in *CoRR abs/2106.07098*, 2021.

[165] C. Bolton, S. Rampazzi, C. Li, A. Kwong, W. Xu, and K. Fu, "Blue note: How intentional acoustic interference damages availability and integrity in hard disk drives and operating systems," in *IEEE Symposium on Security and Privacy*, 2018.

[166] D. F. Kune, J. D. Backes, S. S. Clark, D. B. Kramer, M. R. Reynolds, K. Fu, Y. Kim, and W. Xu, "Ghost talk: Mitigating emi signal injection attacks against analog sensors," in *IEEE Symposium on Security and Privacy*, 2013.

TABLE 3: All the possible action flows that can be spoofed and the corresponding works.

A1: Object detection **A2**: Segmentation **A3**: Localization/SLAM **A4**: Speech recognition **A5**: Distance detection **B**: Object tracking
C1: Environment prediction **C2**: Goal planning **D1**: Path routing **D2**: Parking/Landing planning **E**: Motion planning **F**: Motion controller

Type	Victim Sensor	Perception						Planning					Control	Attack Path	Paper	Action Flow	Flow Pattern
		A1	A2	A3	A4	A5	B	C1	C2	D1	D2	E	F				
Single-Round Action Flow	WWM Radar	●										●	●	AtkPath4	[82]–[84]	ActFlow1	FlowPtn1
		●					●	●				●	●	AtkPath5	[84]	ActFlow2	FlowPtn2
	LiDAR	●										●	●	AtkPath4	[6], [46]–[51]	ActFlow3	FlowPtn1
		●					●	●					●	AtkPath5	-	ActFlow4	FlowPtn2
				●				●				●	●	AtkPath3	-	ActFlow5	FlowPtn5
				●						●		●	●	AtkPath3	-	ActFlow6	FlowPtn6
				●							●	●	●	AtkPath2	-	ActFlow7	FlowPtn8
				●								●	●	AtkPath3	-	ActFlow8	FlowPtn4
				●									●	AtkPath1	-	ActFlow9	FlowPtn7
	Camera	●										●	●	AtkPath4	[52]–[68]	ActFlow10	FlowPtn1
		●					●	●				●	●	AtkPath5	[71]	ActFlow11	FlowPtn2
			●					●				●	●	AtkPath4	[64], [67], [69], [70]	ActFlow12	FlowPtn3
				●				●				●	●	AtkPath3	-	ActFlow13	FlowPtn5
				●						●		●	●	AtkPath3	-	ActFlow14	FlowPtn6
				●							●	●	●	AtkPath2	-	ActFlow15	FlowPtn8
				●								●	●	AtkPath3	-	ActFlow16	FlowPtn4
				●									●	AtkPath1	[72]	ActFlow17	FlowPtn7
	GPS			●				●				●	●	AtkPath3	-	ActFlow18	FlowPtn5
				●						●		●	●	AtkPath3	[41], [42]	ActFlow19	FlowPtn6
				●							●	●	●	AtkPath2	[38]–[40]	ActFlow20	FlowPtn8
				●								●	●	AtkPath3	[41], [43]	ActFlow21	FlowPtn4
				●									●	AtkPath1	-	ActFlow22	FlowPtn7
	IMU			●				●				●	●	AtkPath3	-	ActFlow23	FlowPtn5
				●						●		●	●	AtkPath3	-	ActFlow24	FlowPtn6
				●							●	●	●	AtkPath2	-	ActFlow25	FlowPtn8
				●								●	●	AtkPath3	-	ActFlow26	FlowPtn4
				●									●	AtkPath1	[73], [74], [77]	ActFlow27	FlowPtn7
	Microphone				●				●				●	AtkPath5	[79], [80]	ActFlow28	FlowPtn12
					●				●	●		●	●	AtkPath5	[79], [80]	ActFlow29	FlowPtn11
					●				●		●	●	●	AtkPath5	[79], [80]	ActFlow30	FlowPtn13
	Ultrasonic Sensor					●					●	●	●	AtkPath4	[81], [82]	ActFlow31	FlowPtn9
						●							●	AtkPath4	-	ActFlow32	FlowPtn10
Multi-Round Action Flow	IMU → Env			●									●	-		↻	FlowPtn14
	Env → Camera	●										●	●	AtkPath6	[78]	ActFlow33	
	Env → LiDAR	●										●	●	AtkPath6	-	ActFlow34	
	Env → Camera	●										●	●	AtkPath7	-	ActFlow35	
	Camera → Env			●									●	-		↻	FlowPtn14
	Env → Camera	●										●	●	AtkPath6	-	ActFlow36	
	Env → LiDAR	●										●	●	AtkPath6	-	ActFlow37	
	Env → Camera	●										●	●	AtkPath7	-	ActFlow38	
	LiDAR → Env			●									●	-		↻	FlowPtn14
	Env → Camera	●										●	●	AtkPath6	-	ActFlow39	
	Env → LiDAR	●										●	●	AtkPath6	-	ActFlow40	
	Env → Camera	●										●	●	AtkPath7	-	ActFlow41	
	GPS → Env			●									●	-		↻	FlowPtn14
	Env → Camera	●										●	●	AtkPath6	-	ActFlow42	
	Env → LiDAR	●										●	●	AtkPath6	[44]	ActFlow43	
	Env → Camera	●										●	●	AtkPath7	[45]	ActFlow44	