World Electric Vehicle Journal

Article

Incremental Learning for LiDAR Attack Recognition Framework in Intelligent Driving Using Gaussian Processes

Zujia Miao [1,2], Cuiping Shao [1,3,*], Huiyun Li [1,3] and Yunduan Cui [1]

[1] Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, 1068 Xueyuan Avenue, Shenzhen University Town, Shenzhen 518055, China
[2] Shenzhen Institute of Advanced Technology, University of Chinese Academy of Sciences, Beijing 100049, China
[3] Faculty of Computility Microelectronics, Shenzhen University of Advanced Technology (SUAT), Shenzhen 518055, China
* Correspondence: cp.shao@siat.ac.cn

Abstract: The perception system plays a crucial role by integrating LiDAR and various sensors to perform localization and object detection, which ensures the security of intelligent driving. However, existing research indicates that LiDAR is vulnerable to sensor attacks, which lead to inappropriate driving strategies and need effective attack recognition methods. Previous LiDAR attack recognition methods rely on fixed anomaly thresholds obtained from depth map data distributions in specific scenarios as static anomaly boundaries, which lead to reduced accuracy, increased false alarm rates, and a lack of performance stability. To address these problems, we propose an adaptive LiDAR attack recognition framework capable of adjusting to different driving scenarios. This framework initially models the perception system by integrating the vehicle dynamics model and object tracking algorithms to extract data features, subsequently employing Gaussian Processes for the probabilistic modeling of these features. Finally, the framework employs sparsification computing techniques and a sliding window strategy to continuously update the Gaussian Process model with window data, which achieves incremental learning that generates uncertainty estimates as dynamic anomaly boundaries to recognize attacks. The performance of the proposed framework has been evaluated extensively using the real-world KITTI dataset covering four driving scenarios. Compared to previous methods, our framework achieves a 100% accuracy rate and a 0% false positive rate in the localization system, and an average increase of 3.43% in detection accuracy in the detection system across the four scenarios, which demonstrates superior adaptive capabilities.

Keywords: intelligent driving; attack recognition; Gaussian process; incremental learning

check for updates

Citation: Miao, Z.; Shao, C.; Li, H.; Cui, Y. Incremental Learning for LiDAR Attack Recognition Framework in Intelligent Driving Using Gaussian Processes. *World Electr. Veh. J.* **2024**, *15*, 362. https://doi.org/10.3390/wevj15080362

Academic Editors: Yaoji Deng, Xinglong Zhang and Fen Lin

Received: 16 July 2024
Revised: 1 August 2024
Accepted: 6 August 2024
Published: 12 August 2024

1. Introduction

The security of perception systems in intelligent driving is crucial, as it directly impacts passenger safety and the integrity of the surrounding environment [1]. These systems leverage sensors, including GPS, IMU, LiDAR, and cameras, to gather and process environmental data [2]. However, this dependency also makes the perception systems vulnerable to sensor attacks, which can manipulate data and mislead vehicle operations, potentially resulting in significant traffic incidents. LiDAR sensor attacks are characterized by their low technical requirements, significant potential harm, and challenging traceability, making them a substantial threat to the safety of intelligent driving [1]. The task of defending against these attacks is particularly arduous due to the stealthy nature of the manipulations and the ineffectiveness of previous detection methods like anomaly detection [2,3]. Vulnerable sensors, including cameras, GPS, and LiDAR, are prone to various attacks such as deception, interference, and replay. These attacks can cause incorrect assessments of vehicle positioning or failures in accurately perceiving the environment, thereby escalating the risk of accidents [4]. Effective countermeasures are imperative to mitigate these risks and enhance the resilience of intelligent driving perception systems against potential sensor attacks.

LiDAR attack recognition methods are pivotal in the realm of intelligent driving security. These methods typically encompass sensor redundancy, deep learning, and rule-based algorithms. Sensor redundancy enhances environmental sensing by integrating multiple sensors such as cameras, GPS, and LiDAR, thereby creating a more robust detection framework [5]. Deep learning techniques are employed to train on extensive datasets, enabling sophisticated recognition of attacks [6]. Rule-based algorithms utilize predefined sets of rules and models to manage specific scenarios or types of attacks [7]. However, each method has limitations. Sensor redundancy, while increasing robustness, demands high accuracy and consistency from sensors, which can be compromised by noise and direct attacks, leading to escalated costs [2]. Deep learning approaches hinge on large-scale annotated data, which are scarce in the intelligent driving sector. This scarcity can lead to vulnerabilities under sensor attack scenarios due to potential misdirection [6]. Lastly, rule-based algorithms can struggle with consistency across the varied and dynamic conditions of intelligent driving, as they may not encompass all possible scenarios [4]. Addressing these challenges is essential for advancing sensor attack recognition and enhancing the security posture of intelligent driving systems.

Previous methods of attack recognition in intelligent driving are executed in distinct phases, which complicate hardware implementation and exacerbate communication delays. The Gaussian Process (GP) offers a novel approach to overcome these recognition challenges. GP is capable of end-to-end attack recognition, allowing for simultaneous processing without the need for extensive training datasets. This makes it particularly well suited for scenarios with limited sample sizes, thus enhancing its applicability across diverse driving conditions [8]. In instances of sensor attacks, GP can adeptly model potential noise and anomalies, thereby bolstering the system's resilience against malicious interventions. Moreover, GP facilitates adaptive recognition of data, enabling the system to adjust more effectively to complex and dynamic driving environments [9].

In this paper, we propose incremental learning for the LiDAR sensor attack recognition method that leverages a joint system model and GP. Our approach sets itself apart from conventional data-driven methodologies by integrating system models of intelligent driving with data-driven GP. This integration significantly enhances the capabilities for incremental learning and detection within intelligent driving systems. To ensure the adaptive performance of our proposed framework, we continuously update the GP model using data from the sliding window to accommodate dynamic changes in intelligent driving scenarios. Compared with purely data-driven techniques, attackers targeting our method would need to navigate both the intricacies of the system model and the data-driven components. This dual requirement makes such attacks particularly challenging to execute, positioning our recognition mechanisms among the most formidable in contemporary research.

The main contributions of this paper can be summarized as follows.

- We have developed an attack recognition framework for LiDAR attacks within intelligent driving perception systems, encompassing both localization and detection systems. In this framework, we model the localization system using a vehicle dynamics model and the detection system using an object tracking algorithm, from which we extract data features. Subsequently, we employ Gaussian Processes to perform probabilistic modeling of these data features, which predict uncertainty estimates to effectively recognize LiDAR attacks.
- We propose an innovative incremental learning framework for the adaptive recognition of sensor attacks in intelligent driving, capable of adapting to dynamically changing driving environments. Our approach integrates sliding window techniques, sparsification computing, and Gaussian Processes, which allow for updates within the sliding windows to continuously adjust the Gaussian Process possibility model for incremental learning. Compared to previous methods, our framework maintains a 100% accuracy rate and a 0% false positive rate in the localization system and im-

proves the accuracy by an average of 3.43% in the detection system across various driving scenarios.

The structure of this paper is as follows: Section 2 provides an overview of the LiDAR sensor attack and Gaussian Process. Section 3 states the problem in this work. Section 4 introduces the proposed framework. Section 5 conducts simulations that demonstrate the performance of the proposed framework. Finally, Section 6 summarizes the key conclusions and discusses the significance of the findings.

2. Related Works

2.1. LiDAR Sensor Attack

Intelligent driving heavily relies on sensors to perceive their surroundings and make decisions. LiDAR is crucial for precise navigation and obstacle avoidance. However, if LiDAR is compromised, they may input misleading data, leading the vehicle to make erroneous driving decisions [10]. Common attack types include LiDAR replay and spoofing attacks, as shown in Figure 1.

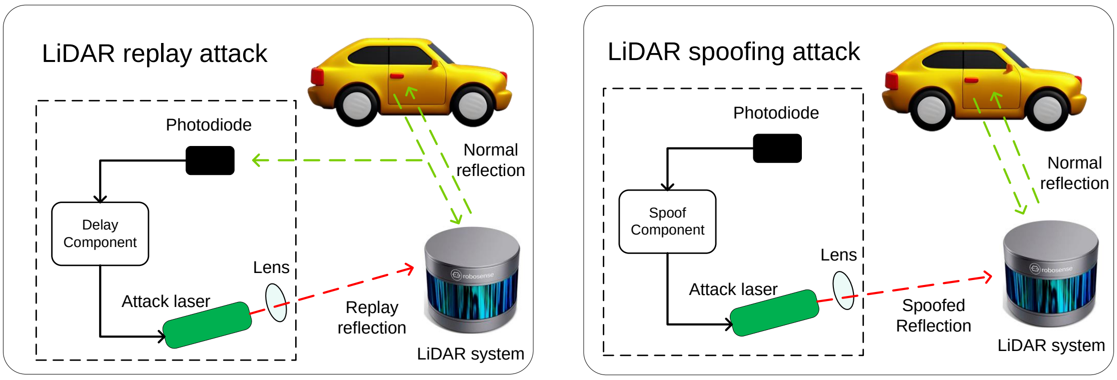

Figure 1. LiDAR replay attack and LiDAR spoofing attack.

- LiDAR replay attack: Attackers record LiDAR data in a specific environment and replay it under different circumstances. This type of attack can cause the LiDAR system to misjudge the current environmental state, mistakenly identifying safe areas as obstructed, or vice versa, recognizing hazardous areas as safe. Such attacks pose a direct threat to the safe operation of intelligent driving [3].
- LiDAR spoofing attack: Attackers send forged signals to the LiDAR system, inducing incorrect environmental perception data. These attacks can lead to navigational errors and may prevent the intelligent driving system from correctly identifying other vehicles, pedestrians, or obstacles on the road, thereby causing severe traffic accidents.

Attack recognition against LiDAR sensor attacks presents multiple challenges. First, attackers may use complex methods such as signal simulation or data tampering, which are difficult to detect through conventional data verification processes [11]. Additionally, the diversity and rapid evolution of attack methods mean that previous recognition mechanisms are often unable to cover all potential attack scenarios. Consequently, developing dynamic and incremental learning recognition systems continues to be a crucial and challenging area of research.

2.2. Gaussian Process

GP is a probabilistic, non-parametric statistical model extensively used for regression and probabilistic classification tasks [8]. These models establish a prior probability distribution over function spaces, ensuring that any finite set of functions exhibits a multivariate Gaussian distribution. In the field of sensor attack detection for intelligent driving, GP is utilized to construct statistical models of sensor data to monitor for anomalies or signs of attacks within the data stream. The model demonstrates significant adaptability when handling complex environmental data and integrating prior knowledge, while the flexible selection of kernel functions allows it to accommodate various data relationships. However, GP faces significant challenges including high computational complexity and difficulties in hyperparameter tuning, which limit its applicability in large-scale data scenarios. Furthermore, the highly dynamic nature of intelligent driving environments necessitates that models swiftly adapt to new data and scenarios. However, GP methodologies still require further optimization to facilitate real-time updates and parameter adjustments.

3. Problem Statement

In this section, we state the problem in recognizing attacks on perception systems, as shown in Figure 2. In intelligent driving applications, perception sensors legitimate input signal z_s of the vehicle, and $s \in \{gps, lidar, imu, \dots, camera\}$ denote various perception sensors. However, the measurements of sensor z_s^m are corrupted with noise, which can be expressed as follows:

$$z_s^m = z_s + \epsilon, \tag{1}$$

where ϵ is the sensor measurement noise, which covers wind noise, road noise and other noise interference in real road conditions. Given the vulnerability of sensors to various types of sensor attacks, measurements can sometimes be compromised, resulting in anomalies. To accurately characterize sensor observations under such conditions, the following adjusted measurement equation is proposed:

$$z_s^m \begin{cases} z_s^m + \xi, & if \quad Sensor \quad s \quad is \quad attacked \\ z_s^m, & otherwise. \end{cases} \tag{2}$$

where ξ represents the signal effect caused by an attacker, rather than the attack itself. The challenge of attack recognition primarily involves accurately distinguishing whether measurements are merely disturbed by noise or corrupted by an attack. To address this, we propose a two-pronged solution. Firstly, a data feature extraction strategy, denoted by $\mathcal{E}(\cdot)$, is essential for isolating the relevant data features e_s^m from sensor measurements, as illustrated below.

$$e_s^m = \mathcal{E}(z_s^m). \tag{3}$$

Subsequently, it is necessary to devise a recognition strategy, denoted as $\mathcal{K}(\cdot)$, which establishes an anomaly decision boundary. This strategy is intended to further determine whether the sensor measurement z_s^m represents a normal operation or an attacked sensor measurement z_s^a and recognizes an attacked sensor S_a. This approach is detailed in the following section.

$$S_a = s \quad if\ e_s^m \notin \mathcal{K}(e_s^m) \tag{4}$$

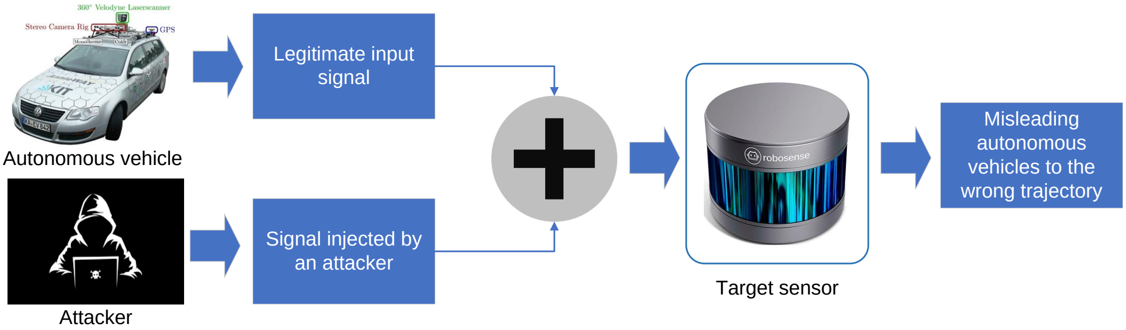
360° Velodyne Laserscanner
Stereo Camera Rig
GPS
Monochrome
Color
Autonomous vehicle
Attacker
Legitimate input signal
Signal injected by an attacker
robosense
Target sensor
Misleading autonomous vehicles to the wrong trajectory

Figure 2. Problem statement.

4. Proposed Framework

Figure 3 depicts the architecture of our proposed framework. Our approach is structured into three key components: feature extraction using a system model, prediction using the Gaussian Process and recognition using uncertainty quantification.

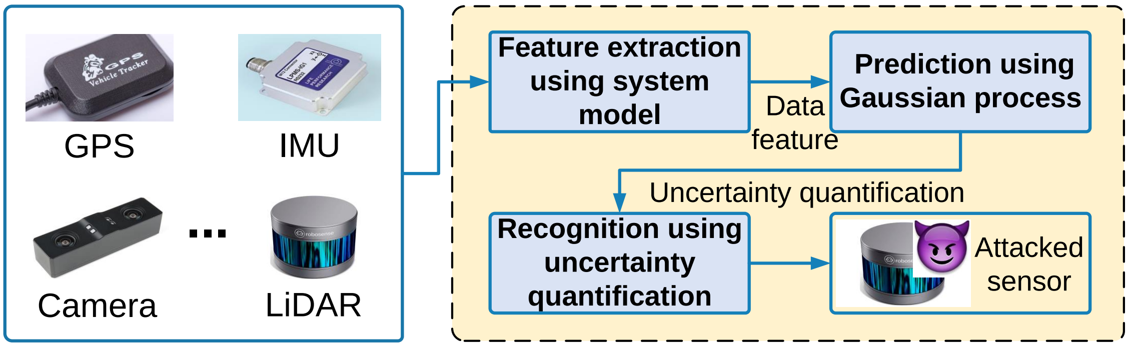
GPS
IMU
Camera
...
LiDAR
Feature extraction using system model
Data feature
Prediction using Gaussian process
Uncertainty quantification
Recognition using uncertainty quantification
Attacked sensor

Figure 3. Architecture of the proposed framework.

4.1. Feature Extraction Using System Model

In the feature extraction phase, the perception system is modeled to systematically extract relevant data features. Perception systems are mainly divided into localization systems and detection systems. On the side of the localization system, the vehicle dynamics model formulation for the positioning system is as follows:

$$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}_{k+1} = \begin{bmatrix} x + \frac{v}{\omega}(\sin(\theta+\omega\Delta t) - \sin(\theta)) \\ y + \frac{v}{\omega}(\cos(\theta) - \cos(\theta+\omega\Delta t)) \\ \theta + \omega\Delta t \end{bmatrix}_{k}. \tag{5}$$

In the described system, the input data features for the localization subsystem in a Gaussian Process (GP) include position coordinates (x, y), velocity v, yaw angle θ, turn rate ω, and time interval Δt, with k serving as the time index. The position and angular information are derived through the fusion of multiple sensors, including LiDAR, GPS, and IMU.

On the side of the detection system, the model is characterized by the Intersection over Union (IoU) metric between LiDAR and stereo cameras, which supports object detection and tracking. The GP of the detection system utilizes the previous moment's IoU, IoU_{t-1}^{id}, identifier id, and angle α_{t-1}^{id} as inputs. This modeling approach is articulated in the subsequent formulation.

$$IoU_t^{id} = [\frac{Area_{lidar} \cap Area_{cam1}}{Area_{lidar} \cup Area_{cam1}}, \frac{Area_{cam1} \cap Area_{cam2}}{Area_{cam1} \cup Area_{cam2}}] = [IoU_{t,1}^{id}, IoU_{t,2}^{id}], \tag{6}$$

In this formulation, id denotes the object identifier obtained from object tracking, as referenced in [12]. The right-side stereo camera serves as the reference camera. $Area_{lidar}$ represents the area where the LiDAR 3D object detection box projects onto the image plane of the reference stereo camera, denoted as $cam1$. The areas $Area_{cam1}$ and $Area_{cam2}$ correspond to the 2D object detection regions captured by the stereo cameras within the reference coordinate system. $IoU_{t,1}^{id}$ quantifies the overlap between LiDAR and the reference camera's object detection results for the object identified by id at time t. Similarly, $IoU_{t,2}^{id}$ measures the IoU for the stereo camera detection of the same object at the same time.

4.2. Prediction Using Gaussian Process

GP predicts the uncertainty quantification as the anomaly boundary. The GP characterizes a collection of random variables, wherein any finite subset follows a joint Gaussian distribution [8]. In the context of state $\hat{x}_t \in \mathbb{R}^D$, action $u_t \in \mathbb{R}^U$ and system noise w_a, the stochastic system dynamics of x are expressed as follows:

$$\hat{x}_{t+1} = f(\hat{x}_t, u_t) + w_a, \tag{7}$$

where $\tilde{x}_t := (\hat{x}_t, u_t)$ denotes the training input tuples, and $y_t := \hat{x}_{t+1}$ represents the corresponding outputs. Define $\sigma^2_{w_a}$ as variance, where a is the dimension index with $a = 1, \ldots, D$. U represents the dimension of action. The system noise is modeled as $w_a \sim \mathcal{N}(0, \sigma^2_{w_a})$. The objective is to train a GP to capture the latent function $y^i_t = f_i(\tilde{x}_t) + w_i$, and the associated covariance kernel function is expressed as follows:

$$k_i(\tilde{x}, \tilde{x}') = \sigma^2_{f_i} \exp\Big(-\frac{1}{2}(\tilde{x} - \tilde{x}')^\top \tilde{A}_i^{-1}(\tilde{x} - \tilde{x}')\Big), \tag{8}$$

where $\sigma^2_{f_i}$ is the variance of f_i and Λ_i is the diagonal matrix of the length scales in the kernel. Let the input and output sets of samples be denoted as $X = [\tilde{x}_1, \ldots, \tilde{x}_N]$ and $Y = [y_2, \ldots, y_{N+1}]$, respectively. The hyperparameters of the GP model $\theta_a = [\sigma^2_{f_i}, \tilde{\Lambda}_i, \sigma^2_{w_i}]$ are learned by maximizing the log marginal likelihood using evidence maximization [8,13].

$$\log p(\boldsymbol{Y}_i|\theta_i) = -\frac{N}{2}\log(2\pi) - \frac{1}{2}|\boldsymbol{K}^i + \sigma^2_{w_i}\boldsymbol{I}^2| - \frac{1}{2}\boldsymbol{Y}_i^T\boldsymbol{\beta}_i, \tag{9}$$

where K^i is the matrix with elements calculated in Equation (8), $\beta_i = (K^i + \sigma^2_{w_i} I)^{-1} Y_i$. For any new input $\tilde{x}_*$, the learned GP provides the posterior mean and variance as follows:

$$m_{f_i}(\tilde{x}_*) = k^T_{i*}(K^i + \sigma^2_{w_i} I)^{-1} Y_i = k^T_{i*}\beta_i, \tag{10}$$

$$\sigma^2_{f_i}(\tilde{x}_*) = k_{i**} - k^T_{i*}(K^i + \sigma^2_{w_i} I)^{-1} k_{i*}, \tag{11}$$

where $k_{i*} = k_i(\tilde{X}, \tilde{x}_*)$, $k_{i**} = k_a(\tilde{x}_*, \tilde{x}_*)$.

To ensure adaptability, this study integrates sparse computation and sliding windows with GP. Sparse Gaussian Processes [14] approximate the GP kernel using inducing points, reducing computational complexity. By partitioning the input space into sub-regions determined by the sliding window width s, each window selects inducing points to approximate the covariance matrix. This approach significantly reduces computational requirements compared to using all data points. Furthermore, integrating sliding windows with GP facilitates incremental learning, enabling adaptation to dynamic datasets. Upon new data arrival, the model updates by adjusting inducing points and the covariance matrix within the current sliding window, avoiding recomputation of the entire dataset's covariance matrix, as shown in Figure 4. This local update mechanism effectively supports efficient model adaptation. To ensure real-time performance, parallel computation is employed. Tasks are distributed across multiple processors or nodes, allowing concurrent computation of inducing point selection and covariance matrix updates within each sub-region defined by the sliding window. GPU acceleration further enhances computation efficiency by accelerating large-scale matrix operations and optimizations. This integrated approach, combining sliding windows and sparse GP, enhances efficient incremental learning and effectively adapts to intelligent driving datasets. Concurrently, the GP in the detection system outputs the IoU values IoU^{id}_t for each time t and object id. In the localization system, the GP outputs the position increment Δx^g_t and orientation increment $\Delta\theta^g_t$ over time.

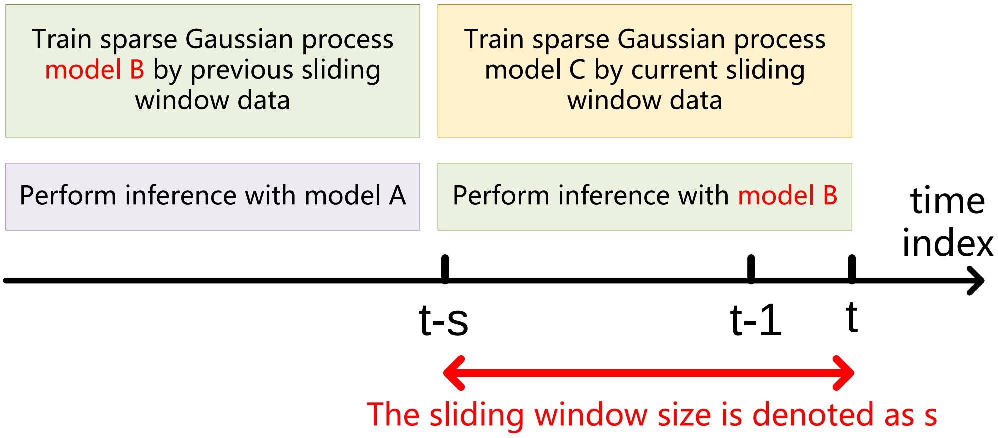
Train sparse Gaussian process model B by previous sliding window data
Train sparse Gaussian process model C by current sliding window data
Perform inference with model A
Perform inference with model B
time index
t-s
t-1
t
The sliding window size is denoted as s

Figure 4. The combination of sliding window and sparse Gaussian Process.

4.3. Recognition Using Uncertainty Quantification

Attack recognition compares the data features with the uncertainty quantification predicted by the GP to achieve attack recognition. If the measurements fall within the uncertainty quantification C_t shown in the following, we conclude that there is no attack.

$$C_t = m_{f_i}(\tilde{x}_*) \pm z * \sqrt{\sigma^2_{f_i}(\tilde{x}_*)}, \tag{12}$$

where z is quantile points of the standard normal distribution with a certain confidence level. Conversely, if the measurements are outside the uncertainty quantification, an attack is recognized, completing the detection process. C_t of the localization system is specified as the following.

$$C_t = [C^x_t, C^y_t, C^\theta_t]. \tag{13}$$

If the displacement measurements $(\Delta x^g_t, \Delta y^g_t)$ or $(\Delta x^l_t, \Delta y^l_t)$ exceed the corresponding uncertainty bounds C^x_t and C^y_t, the affected sensor will be identified as compromised. Similarly, if the orientation measurement of the IMU $\Delta\theta_t$ deviates beyond its specified uncertainty quantification C^θ_t, the IMU will be designated as the attacked sensor. Moreover, the detection system incorporates multiple objectives, each associated with a specific uncertainty quantification C^{id}_t, defined as follows:

$$C^{id}_t = [C^{id}_{t,1}, C^{id}_{t,2}]. \tag{14}$$

If $IoU^{id}_{t,1}$ deviates from $C^{id}_{t,1}$, while $IoU^{id}_{t,2}$ remains within $C^{id}_{t,2}$, the system infers a potential attack on the LiDAR. Conversely, if $IoU^{id}_{t,1}$ adheres to $C^{id}_{t,1}$, but $IoU^{id}_{t,2}$ diverges from $C^{id}_{t,2}$, the inference points to a potential attack on the camera. In situations where both the IoU between the LiDAR and camera detection boxes, and the IoU between detection boxes of the stereo cameras, fail to align within their respective confidence intervals, the system hypothesizes that both the camera and the LiDAR might be vulnerable to attacks.

5. Experimental Results

5.1. Experimental Setup

In this section, we demonstrate the proposed framework on the real-world KiTTI dataset [15] under the LiDAR spoofing attack (LSA) and LiDAR replay attack (LRA). We use four driving scenarios including city, residential, campus, and road environments, as shown in Figure 5. We conduct a thousand experiments each for LRA and LSA attack types, totaling eight thousand experiments. To collaboratively achieve the functionality of the perception system, it is essential to align the data from sensors with different sampling frequencies. Given that the LiDAR has the lowest sampling frequency, set at 10 Hz, the time index is 0.1 s. We set the attacks to occur randomly once with a time index length of 12,000 and an anomaly sampling rate of 0.025%, because multiple attacks can lead to error accumulation and real-world attackers do not engage in sustained attacks on a moving vehicle [2]. We design LRA based on previous experiments [16]. LRA was designed to happen at a randomly selected time index, while the delay of the recorded video was chosen randomly from 3 to 5 second. LSA is simulated based on the experimental results in [11]. We randomly selected two forward beams at an angle of 10° from the origin of the LiDAR coordinate system. We choose a distance between 12 m and generate 120 random pseudo points at this distance between the two selected beams. The height of all pseudo dots does not exceed 1.7 m, which is the typical height of a vehicle.

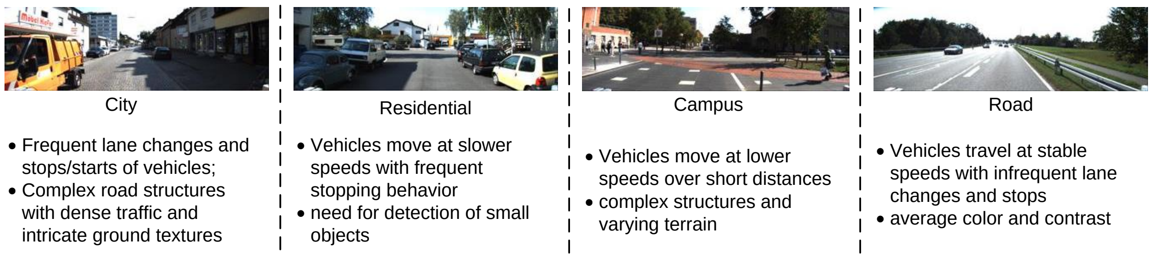
City
• Frequent lane changes and stops/starts of vehicles;
• Complex road structures with dense traffic and intricate ground textures
Residential
• Vehicles move at slower speeds with frequent stopping behavior
• need for detection of small objects
Campus
• Vehicles move at lower speeds over short distances
• complex structures and varying terrain
Road
• Vehicles travel at stable speeds with infrequent lane changes and stops
• average color and contrast

Figure 5. Driving scenarios in intelligence driving.

Based on extensive experimental validation, this study determined optimal parameter settings to enhance model performance. Specifically, a sliding window size of four thousand time indexes was chosen to capture dynamic changes in time-series data for the proposed framework in the localization system. Additionally, 450 inducing points were selected to optimize the computational efficiency and prediction accuracy of the GP model. In addition, sliding window sizes of 3600 and 350 inducing points were chosen for the proposed framework in the detection system. Moreover, a radial basis function was adopted as the kernel function due to its broad adaptability and robust expressive power when handling diverse datasets. We conducted a three-fold cross-validation on a dataset consisting of 12,000 time index data, uniformly dividing them into three subsets for model training and validation purposes. Subsequently, we selected the optimal number of inducing samples, window size, and kernel function based on the best performance metrics. The implementation of these models and computational parameters was accomplished using the GPflow and TensorFlow frameworks [17], both of which support efficient parallel computation.

5.2. Discussion and Analysis

The experimental results, presented in Table 1, demonstrate the effectiveness of our proposed framework for the localization system under LRA. In this particular test, the LRA occurs at time index 203, resulting in a ΔX of -0.54 m. These values indicate that the LiDAR position measurements have been falsified to neighboring positions. Figure 6 presents the experimental results of the proposed framework in the localization system, showing that Δx^l_{203} and Δy^l_{203} of LiDAR exceed C^x_t and C^y_t predicted by the GP. LSA spoofs the LiDAR object detection algorithm into incorrectly detecting the position of the object. Figure 7 shows the experimental results of the proposed framework in the detection system. From Figure 7, we observe that $IoU^{39}_{4,1}$ exceeds the uncertainty quantification, while $IoU_{2,t}$ does not report any anomaly. Table 2 shows the statistical results of the consistency data for the attacked LiDAR and stereo cameras. The table indicates that the consistency between the LiDAR and the camera changes drastically when the attack occurs, resulting in an $IoU^{39}_{4,1}$ value of -3.80, which is outside the uncertainty quantification range of $[-3.56, 2.50]$. Notably, the LSA does not affect the consistency between the stereo cameras, so $IoU^{39}_{4,2}$

remains within the GP-predicted uncertainty quantification. From these results, we can conclude that the LiDAR is under attack.

Table 1. Statistical results of proposed framework in localization system under LiDAR replay attack. The bold data represents the attack time.

Time Index	Δx_t^g	Δy_t^g	Δx_t^l	Δy_t^l	$\Delta\theta_t$	C_t^x	C_t^y	C_t^θ
200	0.27	0.63	0.34	0.30	−0.04	[0.32, 2.71]	[−0.21, 2.14]	[−0.36, 0.45]
201	0.29	0.63	0.34	0.31	−0.03	[0.17, 2.56]	[−0.31, 2.02]	[−0.35, 0.46]
202	0.33	0.70	0.34	0.36	−0.03	[0.37, 2.70]	[−0.18, 2.12]	[−0.35, 0.46]
203	0.31	0.64	**−0.54**	0.34	−0.01	**[0.36, 2.70]**	[−0.17, 2.10]	[−0.34, 0.47]
204	0.32	0.63	0.37	0.39	−0.01	[0.20, 2.55]	[−0.28, 1.97]	[−0.34, 0.48]

Table 2. Statistical results of proposed framework in detection system under LiDAR spoofing attack. The bold data represents the attack time.

Time Index	*id*	$IoU_{t,1}^{id}$	$IoU_{t,2}^{id}$	$C_{t,1}^{id}$	$C_{t,2}^{id}$	Result
4	**39**	**−3.80**	−0.95	**[−3.56, 2.50]**	[−4.34, 4.25]	**LiDAR attack**
	28	0.16	−1.43	[−3.26, 4.28]	[−4.36, 4.56]	
	55	0.72	0.26	[−4.49, 4.77]	[−4.22, 3.06]	
	43	−0.43	0.42	[−2.94, 4.21]	[−4.24, 4.16]	
	40	0.46	0.20	[−4.48, 4.07]	[−4.63, 3.83]	

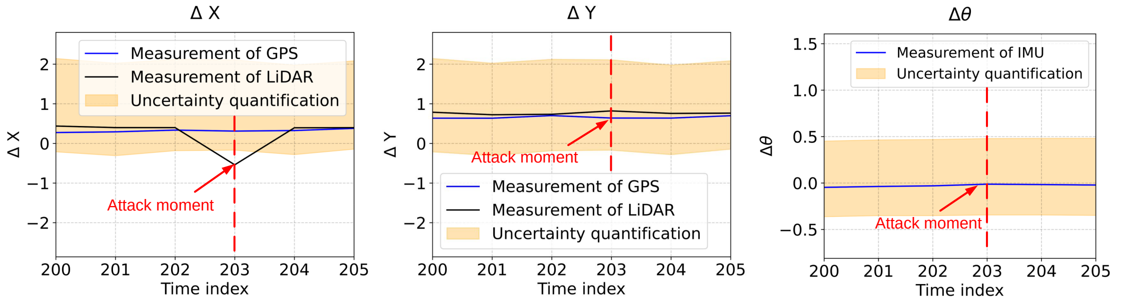

Figure 6. Experimental result of proposed framework in localization system under LiDAR replay attack.

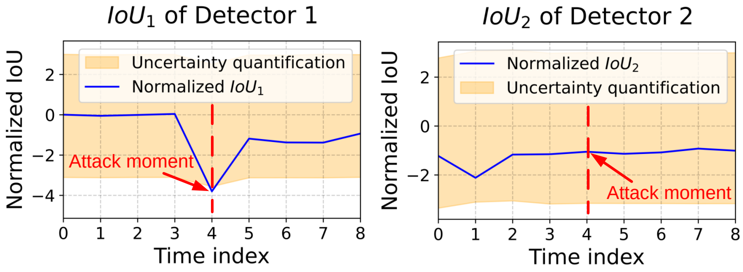

Figure 7. Experimental result of proposed framework in detection system under LiDAR spoofing attack.

To highlight the performance of the proposed framework, we select the optimization-based attack against control systems (OACS) with cumulative sum (CUSUM)-based anomaly detection [2] for evaluating the localization system and the LiDAR and image data fusion for perception attack detection (MDLAD) [1] for assessing the detection system. Figure 8 illustrates the accuracy and false alarm rate variation of the proposed framework, OACS and MDLAD retrained in the KiTTI dataset under sensor attacks. The proposed framework in the localization system consistently maintains a 100% accuracy and 0% false alarm rate in four driving scenarios. In contrast, OACS achieves an average accuracy range of 93.15%

in road environments, with a moderate decline of 1.57% in city scenarios. The proposed framework achieves an average accuracy of 95.05% across the four driving scenarios, which represents an improvement of 3.28% compared with MDLAD. It is worth noting that both OACS and MDLAD rely on fixed thresholds based on data distribution and depth maps as decision boundaries for attack recognition. Depth map-based MDLAD suffers from inadequate adaptability primarily due to the complexity and variability of driving scenarios. Different driving scenarios such as city, road, residential areas and campuses exhibit diverse road structures, traffic densities and types of obstacles, which depth maps often fail to capture fully. Furthermore, depth maps, generated by LiDAR or stereo cameras, are susceptible to environmental factors like lighting conditions, weather variations, and occlusions. In contrast, our approach excels in adaptability by dynamically generating decision boundaries based on the current data.

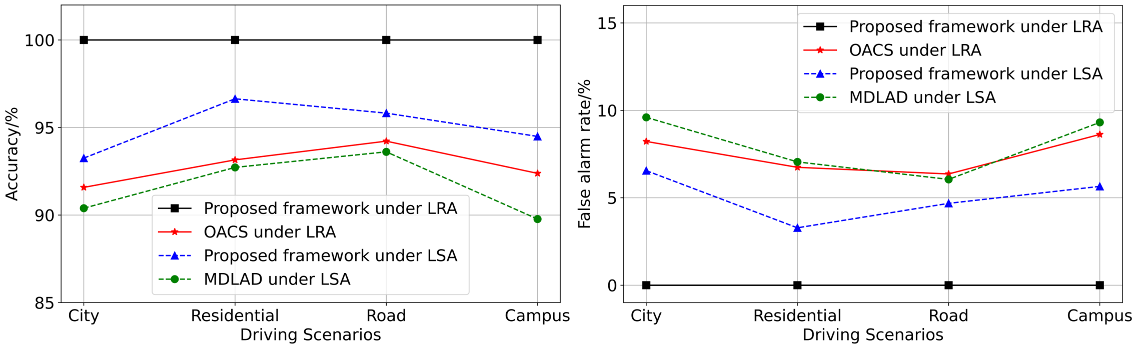

Figure 8. Adaptive analysis of proposed framework.

Our proposed framework merges GP with sparse computation and sliding window techniques, which offer substantial benefits for tasks involving incremental learning attack recognition. This mechanism leverages sparse computation methods enhancing the efficiency of GP application on large-scale datasets. The integration of sliding window techniques allows the GP model to be continuously updated, retraining the model with the most recent subset of data to achieve incremental learning and dynamically adapt to data changes. Additionally, GP naturally offers uncertainty quantification in predictions, which enhances the flexibility and robustness of detection by dynamically adjusting recognition based on data distribution. The proposed framework achieved an average false alarm rate of 0% in four driving scenarios, with a 4.23% reduction in false alarm rate compared to OACS. Although the MDLAD based on depth maps effectively handles the nonlinear relationships associated with complex spatial information, depth maps are prone to noise and data incompleteness due to occlusion and poor reflection. The proposed framework achieved an average false alarm rate of 3.03% in the detection system, which is 3.43% lower than that of MDLAD. Therefore, compared to MDLAD and OACS, the proposed framework demonstrates better adaptability across various driving scenarios.

6. Conclusions

In this paper, we integrate the intelligent driving system model with GP to propose an approach for incremental learning-based sensor attack recognition in intelligent driving systems using GP. We first perform data feature extraction of the localization system and the detection system by combining the vehicle dynamics model and the object tracking algorithm, respectively. Further GP predicts the uncertainty quantification of the data feature as the incremental learning detection boundary. Finally, we implement two common sensor attacks to verify the feasibility of the method and compare the experimental results with previous state-of-the-art methods to validate the superior incremental learning performance of the proposed framework. Future research directions include exploring advanced object tracking algorithms and refining vehicle dynamics models to enhance the adaptability of the proposed framework. Additionally, integrating machine learning techniques and addressing specific sensor attack scenarios are key areas for further investigation and improvement of intelligent driving security.

Author Contributions: Z.M.: writing—review and editing, conceptualization, methodology, formal analysis, investigation, software. C.S.: writing—review and editing, formal analysis, investigation, software, funding acquisition. Y.C.: writing—review and editing, formal analysis, software. H.L.: writing—review and editing, funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China with Grant No. 62004212, Shenzhen Basic Research Project with Grant Nos. JCYJ20210324101210027 and JCYJ20220818100814033, the Basic and Applied Basic Research Foundation of Guangdong Province with Grant No. 2022A1515010284, and Guangdong Provincial Key Laboratory of Computility Microelectronics.

Data Availability Statement: The datasets generated and analyzed during the current study are available from the corresponding author on reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

GP	Gaussian Process
LRA	LiDAR replay attack
LSA	LiDAR spoofing attack
OACS	Optimization-based attack against control systems
MDLAD	Multi-modal deep learning for vehicle sensor data abstraction and attack detection
CUSUM	Cumulative sum

References

1. Liu, J.; Park, J.M. "Seeing is not always believing": Detecting perception error attacks against autonomous vehicles. *IEEE Trans. Dependable Secur. Comput.* **2021**, *18*, 2209–2223. [CrossRef]
2. Gualandi, G.; Maggio, M.; Papadopoulos, A.V. Optimization-based attack against control systems with CUSUM-based anomaly detection. In Proceedings of the 2022 30th Mediterranean Conference on Control and Automation (MED), Vouliagmeni, Greece, 28 June–1 July 2022; pp. 896–901.
3. Bendiab, G.; Hameurlaine, A.; Germanos, G.; Kolokotronis, N.; Shiaeles, S. Autonomous vehicles security: Challenges and solutions using blockchain and artificial intelligence. *IEEE Trans. Intell. Transp. Syst.* **2023**, *24*, 3614–3637. [CrossRef]
4. Kim, K.; Kim, J.S.; Jeong, S.; Park, J.H.; Kim, H.K. Cybersecurity for autonomous vehicles: Review of attacks and defense. *Comput. Secur.* **2021**, *103*, 102150. [CrossRef]
5. Shin, J.; Baek, Y.; Eun, Y.; Son, S.H. Intelligent sensor attack detection and recognition for automotive cyber-physical systems. In Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI), Honolulu, HI, USA, 27 November–1 December 2017; pp. 1–8.
6. Zhang, J.; Pan, L.; Han, Q.; Chen, C.; Wen, S.; Xiang, Y. Deep learning based attack detection for cyber-physical system cybersecurity: A survey. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 377–391. [CrossRef]
7. Kumar, G.; Kumar, K.; Sachdeva, M. The use of artificial intelligence based techniques for intrusion detection: A review. *Artif. Intell. Rev.* **2010**, *34*, 369–387. [CrossRef]
8. Seeger, M. Gaussian process for machine learning. *Int. J. Neural Syst.* **2004**, *14*, 69–106. [CrossRef] [PubMed]
9. Keipour, A.; Mousaei, M.; Scherer, S. Automatic real-time anomaly detection for autonomous aerial vehicles. In Proceedings of the 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 5679–5685.
10. Girdhar, M.; Hong, J.; Moore, J. Cybersecurity of autonomous vehicles: A systematic literature review of adversarial attacks and defense models. *IEEE Open J. Veh. Technol.* **2023**, *4*, 417–437. [CrossRef]
11. Cao, Y.; Xiao, C.; Cyr, B.; Zhou, Y.; Park, W.; Rampazzi, S.; Chen, Q.A.; Fu, K.; Mao, Z.M. Adversarial sensor attack on lidar-based perception in autonomous driving. In Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, London, UK, 11–15 November 2019; pp. 2267–2281.
12. Kumar, S.; Sharma, P.; Pal, N. Object tracking and counting in a zone using YOLOv4, DeepSORT and TensorFlow. In Proceedings of the 2021 International Conference on Artificial Intelligence and Smart Systems (ICAIS), Coimbatore, India, 25–27 March 2021; IEEE: Piscataway, NJ, USA, 2021; pp. 1017–1022.

13. Cui, Y.; Osaki, S.; Matsubara, T. Reinforcement learning boat autopilot: A sample-efficient and model predictive control based approach. In Proceedings of the 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 3–8 November 2019; pp. 2868–2875.
14. Snelson, E.; Ghahramani, Z. Sparse Gaussian processes using pseudo-inputs. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 5–8 December 2005; Volume 18.
15. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? The KiTTI vision benchmark suite. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 3354–3361.
16. Hallyburton, R.S.; Liu, Y.; Cao, Y.; Mao, Z.M.; Pajic, M. Security analysis of Camera-LiDAR fusion against Black-Box attacks on autonomous vehicles. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022.
17. Matthews, A.G.d.; Wilk, M.v.; Nickson, T.; Fujii, K.; Boukouvalas, A.; León-Villagrá, P.; Ghahramani, Z.; Hensman, J. GPflow: A Gaussian process library using TensorFlow. *J. Mach. Learn. Res.* **2017**, *18*, 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.