GPS Location Spoofing Attack Detection for Enhancing the Security of Autonomous Vehicles

Mohsin Kamal*, Arnab Barua*, Christian Vitale*, Christos Laoudias* and Georgios Ellinas*†
*KIOS Research and Innovation Center of Excellence, University of Cyprus, Nicosia, 1678, Cyprus
†Department of Electrical and Computer Engineering, University of Cyprus, Nicosia, 1678, Cyprus
{kamal.mohsin, barua.arnab, vitale.christian, laoudias.christos, gellinas}@ucy.ac.cy

***Abstract*—Attacks on the GPS receiver of Connected and Autonomous Vehicles (CAV) and specifically GPS location spoofing is of great concern for the automotive industry as the attacker can compromise the security of CAVs leading to serious repercussions for the drivers and pedestrians. Attack detection solutions based on specialized hardware (e.g., antenna arrays) and satellite signal processing techniques are accurate, yet bulky and expensive to mount on CAVs. Thus, lightweight and cost-effective solutions for detecting location spoofing attacks are highly desirable. This work presents an in-vehicle attack detection solution that fuses multi-source data readily available from the CAV's on-board sensors. It can be implemented in software running on cheap embedded computing platforms integrated into the CAV. The proposed solution is validated using the real-time CARLA simulator, while extensive experimental results demonstrate its effectiveness under different attack scenarios.**

***Index Terms*—GPS spoofing attacks, autonomous vehicles, security**

I. Introduction

THE reliable and secure operation of the GPS sensor is a crucial factor for the wider acceptance of Connected and Autonomous Vehicles (CAVs), as well as the deployment of Vehicular Ad-hoc NETworks (VANETs) [1]. Relying on GPS measurements aided by a precise high-definition map, vehicles choose an optimized, shortest path from one location to another. This is essential for vehicles to operate correctly and autonomously without any human interaction [2]. However, GPS is susceptible to attacks, such as jamming and spoofing. Jamming attacks can fully block the GPS operation via the transmission of disruptive signals on the same frequencies as those of the GPS signals [3]. On the other hand, a GPS spoofing attack deceives the user via the transmission of signals that have the same characteristics as those of the legitimate GPS satellite signals [4]. There are multiple ways and resources (open source) available for GPS spoofing that pose a critical threat to the safety of AV users. Thus, significant work is currently under way for making sensor systems secure and risk free, utilizing commercially available and off-the-shelf receivers to study possible threats [5] and gain insights on the navigation system vulnerabilities. In general GPS spoofing vulnerabilities exist due to the fact that the characteristics of GPS signals (e.g., modulation type, transmit frequency, satellite ephemeris and clock) are known and do not change rapidly, and thus an attacker can exploit that information to generate spoofed signals with similar characteristics and deceive the user (e.g., the attacker can inject counterfeit pseudorange measurement that can lead to wrong position, velocity, and time solution for the legitimate GPS receiver).

In general, any advanced attacking strategy requires the attacker to be patient. Initially, to launch an attack toward a legitimate GPS receiver, the disruptive signals of the attacker should synchronize with the signals from the satellite. Then, the attacker can force the victim's GPS to lock on spurious signals by increasing the power of the attacking signals, subsequently being able to manipulate the victim's location. GPS spoofing techniques include Lift-off-delay [6], Lift-off-aligned [7], Meaconing or Replay [7], Jam and Spoof [8], Trajectory Spoofing [9], etc. Techniques that try to defend against spoofing are based on signal power monitoring, signal arrival characteristics, signal correlation peak, antenna array, and multi-sensor fusion [10]. The latter defense approach fits with the rationale of cooperative localization in VANETs [11] in the sense that the nodes of the network collaborate by exchanging measurements in order to improve location accuracy and detect and mitigate possible GPS location spoofing attacks.

In order to detect GPS spoofing attacks, existing solutions are based on data-driven [12]–[15] and/or signal processing approaches [16], [17] that are overviewed in Section II. This paper builds upon our previous work [12] to present and evaluate an in-vehicle framework for detecting GPS location spoofing attacks. The proposed solution is based on the fusion of multi-source data that are readily available from the CAV's On-Board Unit (OBU) or the Controller Area Network (CAN) bus. Specifically, the contribution of this work is threefold:

- We formulate the GPS location spoofing attack problem and present a multi-source sensor fusion solution for in-vehicle attack detection. The proposed solution is implemented and validated in the real-time CARLA simulator [18] for autonomous driving systems.
- We test our solution and evaluate extensively the attack detection performance that it achieves in terms of precision, recall, and $F1$ score through multiple scenarios

This work was supported by the European Union's H2020 research and innovation programme under the CARAMEL project (Grant agreement No 833611). It has also been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No 739551 (KIOS CoE) and from the Government of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

using realistic data.
- We investigate and assess thoroughly different design options for setting the attack detection threshold and improving the robustness of the solution to sensor measurement noise and attack bias.

The rest of the paper is organized as follows. Section II overviews the related works. The system model is described in Section III. Section IV presents the framework for attack detection in which a step-wise explanation is provided for the GPS location spoofing attack detection. Discussion on the results generated by applying the proposed algorithm is presented in Section V, while conclusions and directions for future work are discussed in Section VI.

II. Related Works

Solutions that are based on signal processing, typically require specialized equipment. For instance, in [16], real-time GPS spoofing is implemented using software defined radios (SDRs). The raw (I, Q) coefficients are collected by the radio frequency front end which are then analyzed by assessing the phase difference of the GPS signals collected by different antennas. These signals are interference-free and are used for position estimation using open source framework of GNSS-SDR[1]. In [17], the vehicles use dedicated short-range communication to exchange the measured GPS code pseudo-range with the adjacent vehicles. The vehicle then performs a linear operation on the exchanged GPS data to derive independent statistics related to the measurement of each adjacent vehicle. Using these statistics, the vehicle implements a cumulative totaling procedure to locally detect a high correlation of arrival times for spoofed GPS signals. The vehicle reports the local detection value to the selected head vehicle. Head vehicles use the minimum-maximum change detection procedure to optimize global spoofing detection.

The data-driven solutions to detect GPS location spoofing attack use the data as input and applies the proposed algorithm to provide the solution to mitigate the adversarial activity. One of the developed solutions in the literature leverages in-vehicle multisensory data (e.g., accelerometer, gyroscope, compass, etc.) to compute a parallel GPS-free stream of estimated vehicles' locations using a fallback localization method based on Bayesian filtering [12]. This enables the detection of potential location spoofing attacks by comparing the estimated vehicle position with the GPS location reading. A second solution is essentially a collaborative GPS spoofing defense mechanism that relies on multi-modal sensor fusion among the vehicles within a VANET, that includes such measurements as the relative distances, the relative angles, and the relative azimuth angles among the vehicles, as well as the absolute position measurements (i.e., GPS positions) of all vehicles [13]. Various other algorithms have been proposed in the state of the art, including the work in [14], where accelerometer readings of the vehicle are used to compare with the estimated acceleration measurements of the GPS device, with the mismatch beyond a certain threshold signifying a GPS spoofing attack. The decision variable is calculated using the error matrix of acceleration and the threshold is defined according to the probability of false alarm. Other efforts have focused on multi-sensor fusion (MSF) algorithms to estimate the vehicle's location by not completely relying on the GPS measurements [15]. The MSF model provides protection against the off-road and wrong-way attacks in AVs. The threat model is designed considering the capabilities of the attacker along with the control assumptions of the AV.

[1]https://gnss-sdr.org/

III. System Model

The main actors in the GPS location spoofing scenario are the CAV and the attacker, while there are also the GPS satellite infrastructure and a wireless network infrastructure (e.g., cellular towers of telecommunication operators, Wi-Fi Access Points or routers, etc.), as illustrated in Fig. 1.

Our system model includes a CAV that moves on the road network and its true location and velocity are, respectively, $\mathbf{p}_k = [x_k, y_k]^\top$ and $\mathbf{u}_k = [\dot{x}_k, \dot{y}_k]^\top$, where x_k and y_k are the location coordinates at time instance k.

The CAV is equipped with a GPS receiver that processes the satellite positioning signals and outputs the GPS location of the CAV $\mathbf{p}_k^G = [x_k^G, y_k^G]^\top$. An attacker can use Commercial-Off-The-Shelf (COTS) equipment, including Software Defined Radio (SDR) hardware, amplifier, and antenna, combined with open-source SDR software, to interfere with the legitimate GPS signals and disturb the original GPS data. The COTS equipment can be either user-carried at the ground level or mounted on an Unmanned Aerial Vehicle (UAV), as shown in Fig. 1. In this work, it is assumed that the attacker spoofs the GPS location and introduces a user-defined constant bias

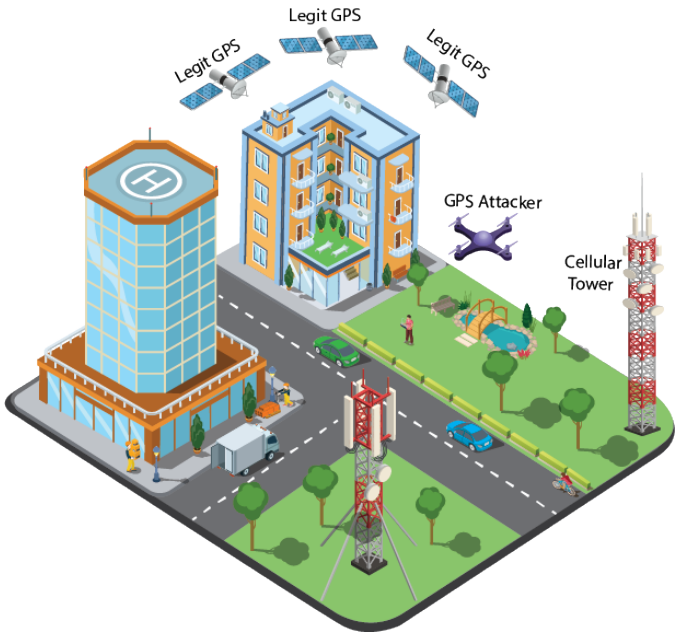


Fig. 1: GPS location spoofing attack scenario on CAVs.

value in both GPS location coordinates. To this end, the GPS location of the CAV under attack is modeled as a Gaussian random variable $\mathbf{p}_k^G \sim \mathcal{N}(\mathbf{p}_k + \mathbf{B}_A, \mathbf{\Sigma}_k^G)$, where $\mathbf{\Sigma}_k^G = \text{diag}_2(\sigma_k^G)$ is the 2×2 diagonal covariance matrix of the GPS measurements with standard deviation of noise $\sigma_k^G = \sigma^G$ in both coordinates. $\mathbf{B}_A = b[1,1]^\top$ is the attack vector where b denotes the attack bias, i.e., the magnitude of the attack to each coordinate in meters. Thus, $b = 0$ represents the attack-free case. Note that during the attack, and while the CAV moves inside the attack range, the GPS receiver reports the spoofed location (i.e., $b \neq 0$), which the CAV perceives as its valid location, if it is not equipped with a reliable spoofing detection solution.

The CAV is also equipped with a specialized device, e.g., featuring SDR hardware and software, that monitors the signals from the surrounding wireless network infrastructure and connected vehicles, independently from the GPS measurements. This device implements a Localization Algorithm (LA) that estimates the current location of the CAV based on these signals, i.e., using network-assisted LAs based on radio signal measurements, e.g., timing, angle, or signal strength measurements, from the neighboring transmitters (e.g, cellular towers, Wi-fi access points, etc.) or using the information received from connected vehicles by applying cooperative LA; see [19] for an overview of such algorithms. The device outputs the estimated location of the CAV $\mathbf{p}_k^L = [x_k^L, y_k^L]^\top$ modeled as a Gaussian random variable $\mathbf{p}_k^L \sim \mathcal{N}(\mathbf{p}_k, \mathbf{\Sigma}_k^L)$, where $\mathbf{\Sigma}_k^L = \text{diag}_2(\sigma_k^L)$ is the covariance matrix that denotes the uncertainty of the LA, i.e., the noise of the CAV locations estimated by the LA has standard deviation $\sigma_k^L = \sigma^L$ in both coordinates.

IV. Attack Detection Framework

A. Outline of the Solution

The proposed solution for GPS spoofing attack detection is based on an in-vehicle GPS location integrity check, as illustrated in Fig. 2.

In the *Prediction* phase, the readings from the on-board sensors collected through the OBU and/or the CAN bus are used to predict the CAV's location $\hat{\mathbf{p}}_{k+1} = [\hat{x}_{k+1}, \hat{y}_{k+1}]^\top$ at time $k+1$ given the previously refined location of the CAV $\tilde{\mathbf{p}}_k = [\tilde{x}_k, \tilde{y}_k]^\top$. This is performed by projecting the location ahead in time using the sensor readings and the underlying mobility model of the CAV. In the *Update* phase, the GPS-free CAV location measurements $\mathbf{p}_{k+1}^L$ provided by the LA are used to update the prediction by means of Bayesian filtering and produce the refined location estimate $\tilde{\mathbf{p}}_{k+1}$. Finally, in the *Attack Detection* phase, the GPS location measurements $\mathbf{p}_{k+1}^G$ provided by the vehicle's GPS receiver are compared with $\tilde{\mathbf{p}}_{k+1}$. In case the deviation exceeds a pre-determined threshold T_d, an alarm is triggered that signifies an attack detection. The entire process is represented in Algorithm 1.

In case of an attack, as a possible mitigation measure, the CAV may no longer use the GPS measurements, but rely instead on $\tilde{\mathbf{p}}_{k+1}$ for location-dependent functionalities (e.g., navigation).

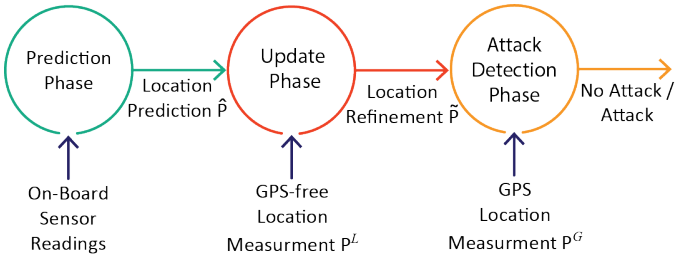


Fig. 2: Data flow of the in-vehicle attack detection framework.

Algorithm 1: GPS Location Spoofing Detection.

input : Previous location estimate $[\tilde{\mathbf{p}}_k, \mathbf{\Sigma}_k^{\tilde{p}}]$, CAV's sensory data $(\alpha, \dot{\varphi}, v)$, radio signal data, GPS location $[\mathbf{p}_{k+1}^G, \mathbf{\Sigma}_{k+1}^G]$, window size w, threshold T_d

output: GPS spoofing attack detection

1 $[\hat{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\hat{p}}] \leftarrow$ *EKF_predict*$(\tilde{\mathbf{p}}_k, \alpha, \dot{\varphi}, v)$;
2 $[\mathbf{p}_{k+1}^L, \mathbf{\Sigma}_{k+1}^L] \leftarrow$ *LA(radio signal data)*;
3 $[\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\tilde{p}}] \leftarrow$ *EKF_update*$([\hat{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\hat{p}}], [\mathbf{p}_{k+1}^L, \mathbf{\Sigma}_{k+1}^L])$;
4 $d_{k+1}^{\{E,B\}} \leftarrow$ *distance*$([\mathbf{p}_{k+1}^G, \mathbf{\Sigma}_{k+1}^G], [\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}_{k+1}^{\tilde{p}}])$;
5 $\bar{d}_{k+1}^{\{E,B\}} \leftarrow$ *filter*$([d_{k-w+2}^{\{E,B\}}, \ldots, d_{k+1}^{\{E,B\}}])$;
6 **if** $\bar{d}_{k+1}^{\{E,B\}} > T_d^{\{E,B\}}$ **then**
7 *GPS location spoofing attack detected*;

B. Prediction Phase

In the prediction phase, utilizing information from the on-board sensors and specifically the steering angle (α), the yaw rate $(\dot{\varphi})$, and the wheel speed (v) measurements, our solution predicts the future location of the vehicle $\hat{\mathbf{p}}_{k+1}$ within a time step Δt; see line 1 in Algorithm 1.

Specifically, the sensor data are applied to the well-known bicycle model [20], i.e., a non-linear model of the vehicle's system state that follows the underlying physical laws. If the body-frame of the vehicle is considered oriented on the x-axis, the one-step prediction of the location and the speed of the vehicle in its body-frame reference systems is:

$$\begin{pmatrix} x_{k+1}^u \\ \dot{x}_{k+1}^u \\ y_{k+1}^u \\ \dot{y}_{k+1}^u \end{pmatrix} = \begin{cases} v\Delta t \\ v \\ \frac{1}{2}\left(C_f\left(\alpha - \frac{l_f\dot{\varphi}}{v}\right) + C_r\frac{l_r\dot{\varphi}}{v}\right)\frac{1}{M}\Delta t^2 \\ \left(C_f\left(\alpha - \frac{l_f\dot{\varphi}}{v}\right) + C_r\frac{l_r\dot{\varphi}}{v}\right)\frac{1}{M}\Delta t \end{cases} \quad (1)$$

where x_{k+1}^u $(\dot{x}_{k+1}^u)$ and y_{k+1}^u $(\dot{y}_{k+1}^u)$ are the longitudinal and lateral displacements (velocities) between two consecutive time steps in the body frame, l_f and l_r represent the distance of the front wheel and the rear wheel from the vehicle's barycenter, respectively, M is the vehicle's mass, while C_f and C_r represent the corner stiffness of the front and rear wheels, respectively. Subsequently, by applying a simple coordinate transformation, the one-step prediction in the global geographic reference system is obtained as:

$$\begin{pmatrix} \hat{x}_{k+1} \\ \hat{\dot{x}}_{k+1} \\ \hat{y}_{k+1} \\ \hat{\dot{y}}_{k+1} \\ \hat{\varphi}_{k+1} \end{pmatrix} = \begin{cases} \tilde{x}_k + x^u_{k+1}\cos\hat{\varphi}_k - y^u_{k+1}\sin\hat{\varphi}_k \\ \dot{x}^u_{k+1}\cos\hat{\varphi}_k - \dot{y}^u_{k+1}\sin\hat{\varphi}_k \\ \tilde{y}_k + x^u_{k+1}\sin\hat{\varphi}_k + y^u_{k+1}\cos\hat{\varphi}_k \\ \dot{x}^u_{k+1}\sin\hat{\varphi}_k - y^u_{k+1}\cos\hat{\varphi}_k \\ \hat{\varphi}_k + \dot{\varphi}\Delta t \end{cases} \quad (2)$$

Under the assumption of uncorrelated and Gaussian measurement noise, the associated covariance of the estimated vehicle's system state is computed using the Extended Kalman Filter (EKF) algorithm applied to equations (1) and (2), as described in [20]. The predicted location $\hat{\mathbf{p}}_{k+1}$ follows a Gaussian distribution, i.e., $\hat{\mathbf{p}}_{k+1} \sim \mathcal{N}(\hat{\mathbf{p}}_{k+1}, \mathbf{\Sigma}^{\hat{p}}_{k+1})$, where $\mathbf{\Sigma}^{\hat{p}}_{k+1}$ is the covariance matrix that denotes the uncertainty of the predicted location.

C. Update Phase

In the update phase, the EKF algorithm fuses the predicted vehicle's location $\hat{\mathbf{p}}_{k+1}$ with the GPS-free global location measurement $\mathbf{p}^L_{k+1}$ provided by the LA by processing the collected radio signal data. Consequently, the refined location estimate $\tilde{\mathbf{p}}_{k+1}$ is computed, as shown in Algorithm 1 (lines $2-3$). The location estimate $\tilde{\mathbf{p}}_{k+1}$ follows a Gaussian distribution, i.e., $\tilde{\mathbf{p}}_{k+1} \sim \mathcal{N}(\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}^{\tilde{p}}_{k+1})$, where $\mathbf{\Sigma}^{\tilde{p}}_{k+1}$ is the covariance matrix that denotes the uncertainty of the refined location estimate.

D. Attack Detection Phase

The main idea for detecting the GPS location spoofing attack is to check whether the deviation between the estimated CAV location $\tilde{\mathbf{p}}_{k+1}$ and the GPS reading $\mathbf{p}^G_{k+1}$, i.e., by means of a distance metric, is greater than a certain threshold. This is achieved through a 4-step procedure where the first step (Step 0) is performed *offline*, i.e., in attack-free conditions before the real-life deployment of the solution, in order to select a proper threshold value. Steps 1–3 are performed *online*, i.e., while the proposed attack detection solution runs within a moving CAV under unknown conditions, as part of Algorithm 1.

Step 0: Threshold Selection
The following data-driven approach is used to select the threshold T_d in an empirical way. Assuming an initial attack-free time period, a CAV equipped with the proposed solution moves freely on the road network collecting a series of N GPS location measurements $\mathbf{p}^G_n,\ n = 1, \ldots, N$. At the same time, the underlying EKF algorithm outputs the series of estimated locations $\tilde{\mathbf{p}}_n,\ n = 1, \ldots, N$. Next, the corresponding distance between each location pair $d_n,\ n = 1, \ldots, N$ is computed; two candidate distance metrics are described in Step 1. Subsequently, the filtered version of the series $\bar{d}_n,\ n = 1, \ldots, N$ is computed; a simple filtering for reducing the inherent noise in these distance measurements is presented in Step 2. Finally, the filtered distance values $\bar{d}_n,\ n = 1, \ldots, N$ are used to derive the Empirical Cumulative Distribution Function (ECDF). In this work, the threshold value T_d is selected using parameter $\gamma \in [0\ 1]$ that denotes the γ^{th} percentile of the ECDF curve. In this approach, when no attack is in place, the percentage of false positives is chosen by design, and is equal to $1-\gamma$.

Based on the γ value there is a performance trade-off regarding the correct and false detection rates in the presence of attacks. For example, setting a low γ value leads to a relatively low threshold T_d, thus the solution would probably detect the majority of the potential attacks; however, the false alarm rate will be high, and as a result, the proposed solution will be ineffective. On the other hand, by setting a high γ value (e.g., $\gamma = 1$), the high false alarm rate can be addressed but at the expense of missed detections, which will increase.

Step 1: Computation of Location Distance (d)
In this work, we investigate the Euclidean and the Bhattacharyya distance metrics to compute the distance between the estimated CAV location $[\tilde{\mathbf{p}}_{k+1}, \mathbf{\Sigma}^{\tilde{p}}_{k+1}]$ and the GPS location $[\mathbf{p}^G_{k+1}, \mathbf{\Sigma}^G_{k+1}]$, denoted by d^E_{k+1} and d^B_{k+1}, respectively. Either of them can be used, as shown in line 4 of Algorithm 1; however, we highlight that any distance metric can be applied.

Omitting the time index k for clarity, the Euclidean and Bhattacharyya distances are given by

$$d^E(\tilde{\mathbf{p}}, \mathbf{p}^G) = \|\tilde{\mathbf{p}} - \mathbf{p}^G\|, \quad (3)$$

$$d^B(\tilde{\mathbf{p}}, \mathbf{\Sigma}^{\tilde{p}}, \mathbf{p}^G, \mathbf{\Sigma}^G) = \frac{1}{8}(\tilde{\mathbf{p}} - \mathbf{p}^G)^\top \mathbf{\Sigma}^{-1}(\tilde{\mathbf{p}} - \mathbf{p}^G) + \frac{1}{2}\ln\left(\frac{|\mathbf{\Sigma}|}{\sqrt{|\mathbf{\Sigma}^{\tilde{p}}||\mathbf{\Sigma}^G|}}\right), \quad (4)$$

where $\|\cdot\|$ denotes the Euclidean distance, $|\cdot|$ denotes the determinant of the matrix, and $\mathbf{\Sigma} = 1/2(\mathbf{\Sigma}^{\tilde{p}} + \mathbf{\Sigma}^G)$.

The Euclidean distance uses only first-order statistics of the assumed distributions for $\tilde{\mathbf{p}}$ and $\mathbf{p}^G$, while the Bhattacharyya distance uses second-order statistics as well, i.e., the associated covariance matrices $\mathbf{\Sigma}^{\tilde{p}}$ and $\mathbf{\Sigma}^G$, respectively. During the operation of Algorithm 1, the covariance matrix $\mathbf{\Sigma}^{\tilde{p}}_{k+1}$ at time $k+1$ is produced from the EKF update step (see line 2 of Algorithm 1). Note that the covariance matrix $\mathbf{\Sigma}^G_{k+1}$ can be straightforwardly computed using information provided by the GPS receiver, e.g., the number of satellites or the GPS horizontal accuracy measurement that is readily available in many commercial GPS receivers and provides the estimated error (in meters) of the current GPS location.

Step 2: Filtration of Location Distance ($\bar{d}$)
The distance value $d^{\{E,B\}}_{k+1}$ produced at Step 1 may fluctuate significantly while the CAV is moving, even at nearby locations, due to the inherent noise that affects both the GPS locations $\mathbf{p}^G_{k+1}$ and the estimated locations $\tilde{\mathbf{p}}_{k+1}$ owing to different reasons. For instance, the noise in the GPS measurements depends on various factors, e.g., environmental conditions such as cloud overcast, humidity, etc. In addition, in urban areas, where the satellite signals may be obstructed by high buildings, the noise in GPS readings is typically higher compared to suburban areas with low-height buildings or open-sky rural areas, where more satellites signals are visible. On the other hand, the locations $\tilde{\mathbf{p}}_{k+1}$ can be affected by higher noise in the estimated locations $\mathbf{p}^L_{k+1}$ provided by the LA in the EKF

update step, i.e., higher localization error of the LA due to low density of the surrounding infrastructure that leads to a limited number of radio signals available for localization. Consequently, these uncertainties propagate in the computed distance value $d_{k+1}^{\{E,B\}}$, thus increasing the probability of false attack detection.

Even though the aforementioned uncertainties can be estimated and quantified through the covariance matrices $\mathbf{\Sigma}^{\tilde{p}}$ and $\mathbf{\Sigma}_{k+1}^{G}$, it is highly desirable to reduce the noise in $d_{k+1}^{\{E,B\}}$ in order to increase the robustness of the proposed detection solution through filtering; see line 5 in Algorithm 1. In particular, we apply a simple averaging filter, which is based on a sliding window of size w that processes the past distance values, to compute the filtered distance $\bar{d}_{k+1}^{\{E,B\}}$ by

$$\bar{d}_{k+1}^{\{E,B\}} = \frac{\sum_{i=k-w+2}^{k+1} d_i^{\{E,B\}}}{w}. \tag{5}$$

In the case that $w = 1$, the filter is not applied on the distance values, thus $\bar{d}_{k+1}^{\{E,B\}} = d_{k+1}^{\{E,B\}}$.

Step 3: Attack/No Attack Decision

Finally, the decision whether an attack has occurred or not is made by comparing $\bar{d}_{k+1}^{\{E,B\}}$ obtained in the previous Step 2 with the threshold value selected in Step 0. If the filtered distance value is greater than T_d^e (T_d^b) for the Euclidean (Bhattacharyya) distance, a GPS location spoofing attack is detected, as shown in lines $6-7$ of Algorithm 1.

V. Performance Evaluation

Performance results are obtained utilizing the CARLA simulator [18] which is a development, training, and validation tool for autonomous driving systems. All simulations were performed on a Linux-based workstation PC with 8 GB RAM and GPU.

A. Performance Metrics

We analyze the detection results of the proposed solution using a confusion matrix, which is also known as the contingency table. The confusion matrix classifies the results into *True Positive* (T_P), *True Negative* (T_N), *False Positive* (F_P) and *False Negative* (F_N). Based on this classification, we assess the performance of our attack detection solution on a number of metrics, namely *Precision* (P), *Recall* (R), and $F1$ *Score*.

Specifically, Precision (P) provides information concerning the rate at which the algorithm detects attacks. In our application scenario, it is the ratio of the true attacks detected over all detected attacks including false positives given by

$$P = \frac{\#\ of\ true\ attacks\ detected}{\#\ of\ true\ and\ false\ attacks\ detected} = \frac{T_P}{T_P + F_P} \tag{6}$$

Similarly, Recall (R) is the ratio of the true attacks detected over all true attacks including the missed detections, i.e., false negatives. This is given by

TABLE I: Thresholds $T_d^{\{E,B\}}$ pertaining to Euclidean and Bhattacharyya distance (with and without sliding window $w = 5$) for three different trajectories

Cases	Samples	b (m)	T_d^E	T_d^E $w=5$	T_d^B	T_d^B $w=5$
Case 1	2350	5	4.6838	3.5353	1.5512	1.0351
Case 2		9				
Case 3		12				
Case 4	1199	5	5.3063	3.9415	1.8168	1.1939
Case 5		9				
Case 6		12				
Case 7	5600	5	11.0864	10.8177	7.9412	7.7781
Case 8		9				
Case 9		12				

$$R = \frac{\#\ of\ true\ attacks\ detected}{\#\ of\ true\ attacks} = \frac{T_P}{T_P + F_N} \tag{7}$$

Finally, the $F1$ score is the weighted average of P and R and is the measure of accuracy on the data set given by

$$F1 = 2\left(\frac{P \times R}{P + R}\right). \tag{8}$$

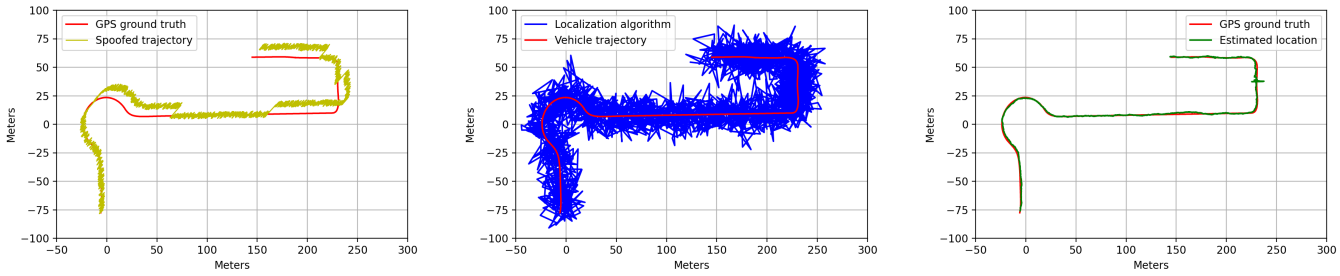
The $F1$ score gets a higher value (near 1) when the F_P and F_N are low. If a system is performing poorly by generating more F_P and F_N, the $F1$ score will be low (near 0).

B. Simulation Setup

Different test cases are investigated for which the attack bias b in the GPS measurements varies for different trajectories of the vehicle. Some of the parameters are the same for all test cases, such as $\sigma^L = 10\,\mathrm{m}$, $\sigma^G = 3\,\mathrm{m}$, the sampling interval $\Delta t = 0.05\,\mathrm{s}$, front corner stiffness ($873\,\mathrm{N/deg}$), rare corner stiffness ($1327\,\mathrm{N/deg}$), mass of vehicle ($2090\,\mathrm{kg}$), longitudinal distance of front wheel $3.2\,\mathrm{m}$, longitudinal distance of rear wheel $2.6\,\mathrm{m}$, $\gamma = 0.95$, and window size $w = 5$. The trajectory of the vehicle changes after every third case i.e., the trajectories are the same for Cases $1-3$, for Cases $4-6$, and for Cases $7-9$. In total, three trajectories are used for which the simulation results are generated. Table I summarizes the threshold values T_d^E and T_d^B obtained using the Euclidean and Bhattacharyya distance, respectively, with and without a sliding window for the three different trajectories.

C. Experimental Results

1) GPS-free Estimated Trajectory: The performance results obtained for Case 1 in terms of location estimation are presented in Fig. 3. As illustrated in Fig. 3(a), the attack is performed over half of the simulation time (in consecutive time instants) i.e., for Cases $1-3$, the attack is performed on 1176 samples out of 2350 total samples. As the GPS location is spoofed, the fallback approach is applied for generating the trajectory, i.e., the vehicle's location is estimated using the localization algorithm and on-board sensor data. The trajectories obtained via the localization algorithm are shown in Fig. 3(b). Finally, Fig. 3(c) presents the estimated trajectory utilizing the fallback solution, demonstrating that the vehicle's



(a) Attack on the vehicle's location.

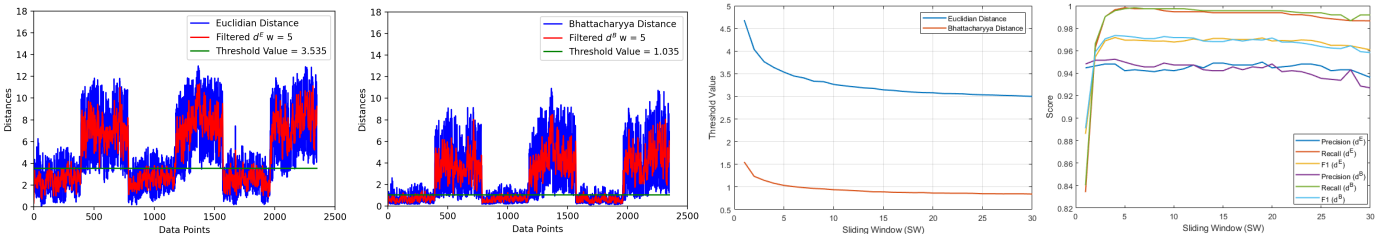
(b) Comparison of trajectories generated using the localization algorithm and the GPS ground truth.

(c) Estimated vehicle trajectory.

Fig. 3: Performance results for Case 1.

TABLE II: Performance metrics for the different test cases

Cases	d^E			d^E with $w = 5$			d^B			d^B with $w = 5$		
	P	R	$F1$	P	R	$F1$	P	R	$F1$	P	R	$F1$
Case 1	0.9447	0.8340	**0.8859**	0.9421	0.9981	**0.9693**	0.9498	0.8404	**0.8918**	0.9498	0.9973	**0.9729**
Case 2	0.9447	1.0000	0.9715	0.9404	0.9990	0.9689	0.9481	1.0000	0.9733	0.9489	1.0000	0.9738
Case 3	0.9447	1.0000	0.9715	0.9404	1.0000	0.9693	0.9481	1.0000	0.9733	0.9489	1.0000	0.9738
Case 4	0.9250	0.7152	**0.8066**	0.9166	0.9786	**0.9466**	0.9250	0.7142	**0.8061**	0.9150	0.9734	**0.9432**
Case 5	0.9250	1.0000	0.9610	0.9150	1.0000	0.9556	0.9250	1.0000	0.9610	0.9150	1.0000	0.9556
Case 6	0.9250	1.0000	0.9610	0.9150	1.0000	0.9556	0.9250	1.0000	0.9610	0.9150	1.0000	0.9556
Case 7	0.9389	0.5152	0.6654	0.9371	0.4908	0.6442	0.9381	0.5159	0.6657	0.9378	0.4922	0.6455
Case 8	0.9389	0.7648	**0.8429**	0.9376	0.8834	**0.9093**	0.9381	0.7653	**0.8430**	0.9371	0.8825	**0.9090**
Case 9	0.9389	0.9432	0.9410	0.9360	0.9444	0.9402	0.9381	0.9435	**0.9408**	0.9360	0.9465	**0.9412**



(a) Euclidean distance.

(b) Bhattacharyya distance.

(c) Window size vs threshold.

(d) Window size vs performance metrics.

Fig. 4: GPS spoofing attack detection for the trajectory of Case 1 and the effect on performance by varying window size w.

location and subsequent trajectory are estimated with high accuracy as compared to the GPS ground truth.

2) Attack Detection Results: As previously mentioned, the attack bias in the GPS measurements varies for each test case, as presented in Table I. The threshold (T_d) achieved for the test cases during the attack-free period, shows that the threshold values vary with the change in trajectories, mainly due to the different sensor and localization measurement values obtained. The high values of $T_d^{\{E,B\}}$ are obtained for the third trajectory (Cases $7-9$) compared to the other trajectories, because with large data sizes, the EKF deviates more, hence introducing more error in the estimated trajectory.

Based on these values, Table II shows the attack detection performance results in terms of the performance metrics considered (i.e., precision, recall, and $F1$ score) for all test cases. It is observed that, for all trajectories, a higher value of the $F1$ score is achieved when the attack bias has the largest value (12m for these experiments), as for these cases both the precision and recall values are high. This is the case, since the higher the attack bias values, the more effectively the attack is detected (as attack detection relies on the comparison between GPS ground truth and estimated location values and the difference of both increases with higher bias values). Hence, high attack bias values will result in high true positive (T_P) values and low false positive and false negative values (F_P and F_N, respectively), resulting in high $F1$ scores.

3) Comparison of Algorithms: Comparison results for the proposed attack detection approach utilizing either the Euclidean or the Bhattacharyya distance are presented in Table II with and without a sliding window of size $w = 5$, as well as in Fig. 4. From Table II, it is clear that the use of the sliding window increases the attack detection accuracy when we have low b, with both distance measurements having comparable results. With high value of b, the effect of the sliding window

on the $F1$ score is negligible. The reason is that the false values above and below $T_d^{\{E,B\}}$ are evident and with/without window, the values for P and R do not change significantly.

Figures 4(a) and 4(b), illustrate cases of attack detection, misdetection, and false detection, for the trajectory of test cases $1-3$, when the Euclidean and the Bhattacharyya distances are utilized, respectively. Further, in Figs. 4(c) and 4(d), the window size value is increased from 1 (representing the case of no sliding window applied) to 30, in order to ascertain the impact of the window size on the threshold parameter as well as on the performance metrics (i.e., precision, recall, and $F1$ score). Clearly, as the window size increases, the threshold value reduces, with the Bhattacharyya distance providing lower T_d values compared to the Euclidean distance, since in Bhattacharyya distance, all the data points are normalized initially and then the statistical distributions are used to find the overlap of GPS signal and estimated locations. The Bhattacharyya coefficient returns value between 0 and 1. Furthermore, the logarithmic function applied on the coefficient to determine the Bhattacharyya distance also gives a small value. Due to relative small values, the 95^{th} percentile of ECDF gives small values for threshold. On the other hand, Fig. 4(d) illustrates that the $F1$ score, precision, and recall increase with the sliding window size, for both Euclidean and Bhattacharyya distance, with most of the gain obtained when the window size reaches 5 samples. Further, for window sizes $17-30$ the $F1$ scores converge to a value of 0.9607 and 0.9582 for both the Euclidean and Bhattacharyya distances, respectively, demonstrating the high detection accuracy of the proposed technique. The reason is that the samples are averaged over larger sample values with the increase of window size, that includes the higher distance values as well which are achieved because of the attack on the GPS. This leads to more F_P and subsequently decreases the $F1$ score.

VI. Conclusion

We present a GPS location spoofing attack detection technique that relies on the distance between the actual GPS measurements and the estimated CAV location based on GPS-free sensor measurements. The overall performance of the proposed solution is highly dependent on the magnitude of the attack that is modeled as a bias, with better detection results for higher GPS attack bias values. In addition, the introduction of the sliding window is instrumental in achieving higher detection accuracy. Finally, it is shown that the Euclidean and Bhattacharyya distances attain comparable attack detection performance, indicating that they are both applicable for enhancing the security of CAVs against GPS spoofing attacks. As part of our future work, we plan to compare our technique against existing solutions, and adjust the detection threshold while the CAV is moving to adapt to varying conditions, e.g., the uncertainty in the GPS readings is higher in cluttered urban settings compared to open-sky rural areas.

References

[1] R. K. Jaiswal and C. Jaidhar, "A performance evaluation of location prediction position-based routing using real gps traces for vanet," *Wireless Personal Communications*, vol. 102, no. 1, pp. 275–292, 2018.

[2] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future directions," *Proceedings of the IEEE*, vol. 108, no. 2, pp. 357–372, 2019.

[3] R. Ferreira, J. Gaspar, P. Sebastião, and N. Souto, "Effective gps jamming techniques for uavs using low-cost sdr platforms," *Wireless Personal Communications*, vol. 115, no. 4, pp. 2705–2727, 2020.

[4] S. Hussein, A. Krings, and A. Azadmanesh, "Vanet clock synchronization for resilient dsrc safety applications," in *2017 Resilience Week (RWS)*. IEEE, 2017, pp. 57–63.

[5] T. Infrastructure, "Vulnerability assessment of the transportation infrastructure relying on the global positioning system," Technical Report, Center, John A. Volpe National Transportation Systems, Tech. Rep., 2001.

[6] A. Neri, C. Stallo, A. Coluccia, V. Palma, P. Salvatori, A. Vennarini, O. Pozzobon, G. Gamba, S. Fantinato, M. Barbuto *et al.*, "An anti-jamming and anti-spoofing digital beamforming platform for the gnss-based ertms train control system," in *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 2017, pp. 3538–3556.

[7] R. Xu, M. Ding, Y. Qi, S. Yue, and J. Liu, "Performance analysis of gnss/ins loosely coupled integration systems under spoofing attacks," *Sensors*, vol. 18, no. 12, p. 4108, 2018.

[8] E. Ranyal and K. Jain, "Unmanned aerial vehicle's vulnerability to gps spoofing a review," *Journal of the Indian Society of Remote Sensing*, pp. 1–7, 2020.

[9] Y.-C. Liu, G. Bianchin, and F. Pasqualetti, "Secure trajectory planning against undetectable spoofing attacks," *Automatica*, vol. 112, p. 108655, 2020.

[10] L. He, H. Li, and M. Lu, "A fundamental architecture of anti-spoofing gnss receiver," in *China Satellite Navigation Conference*. Springer, 2017, pp. 899–909.

[11] M. A. Hossain, I. Elshafiey, and A. Al-Sanie, "Cooperative vehicle positioning with multi-sensor data fusion and vehicular communications," *Wireless Networks*, vol. 25, no. 3, pp. 1403–1413, 2019.

[12] C. Vitale, N. Piperigkos, C. Laoudias, G. Ellinas, J. Casademont, J. Escrig, A. Kloukiniotis, A. S. Lalos, K. Moustakas, R. D. Rodriguez *et al.*, "Caramel: results on a secure architecture for connected and autonomous vehicles detecting gps spoofing attacks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2021, no. 1, pp. 1–28, 2021.

[13] F. L. Lobo, D. C. Grael, H. A. d. Oliveira, L. A. Villas, A. Almehmadi, and K. El-Khatib, "A distance-based data fusion technique for minimizing gps positioning error in vehicular ad hoc networks," in *Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks*, 2019, pp. 101–108.

[14] K.-C. Kwon and D.-S. Shim, "Performance analysis of direct gps spoofing detection method with ahrs/accelerometer," *Sensors*, vol. 20, no. 4, p. 954, 2020.

[15] J. Shen, J. Y. Won, Z. Chen, and Q. A. Chen, "Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under {GPS} spoofing," in *29th {USENIX} Security Symposium ({USENIX} Security 20)*, 2020, pp. 931–948.

[16] J. Friedt, W. Feng, D. Rabus, and G. Goavec-Merou, "Real time gnss spoofing detection and cancellation on embedded systems using software defined radio," *EuCAP, Düsseldorf, Germany*, 2021.

[17] F. A. Milaat and H. Liu, "Decentralized detection of gps spoofing in vehicular ad hoc networks," *IEEE Communications Letters*, vol. 22, no. 6, pp. 1256–1259, 2018.

[18] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "Carla: An open urban driving simulator," in *Conference on robot learning*. PMLR, 2017, pp. 1–16.

[19] C. Laoudias, A. Moreira, S. Kim, S. Lee, L. Wirola, and C. Fischione, "A survey of enabling technologies for network localization, tracking, and navigation," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 4, pp. 3607–3644, 2018.

[20] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization," *IEEE Transactions on Control Systems Technology*, vol. 15, no. 6, pp. 1080–1088, 2007.