GPS-IDS: An Anomaly-based GPS Spoofing Attack Detection Framework for Autonomous Vehicles

Murad Mehrab Abrar, Raian Islam, Shalaka Satam, Sicong Shao, Salim Hariri, *Senior Member, IEEE*, and Pratik Satam

Abstract—Autonomous Vehicles (AVs) heavily rely on sensors and communication networks like Global Positioning System (GPS) to navigate autonomously. Prior research has indicated that networks like GPS are vulnerable to cyber-attacks such as spoofing and jamming, thus posing serious risks like navigation errors and system failures. These threats are expected to intensify with the widespread deployment of AVs, making it crucial to detect and mitigate such attacks. This paper proposes GPS Intrusion Detection System, or GPS-IDS, an Anomaly Behavior Analysis (ABA)-based intrusion detection framework to detect GPS spoofing attacks on AVs. The framework uses a novel physics-based vehicle behavior model where a GPS navigation model is integrated into the conventional dynamic bicycle model for accurate AV behavior representation. Temporal features derived from this behavior model are analyzed using machine learning to detect normal and abnormal navigation behavior. The performance of the GPS-IDS framework is evaluated on the AV-GPS-Dataset — a real-world dataset collected by the team using an AV testbed. The dataset has been publicly released for the global research community. To the best of our knowledge, this dataset is the first of its kind and will serve as a useful resource to address such security challenges.

Index Terms—Autonomous Vehicle, Anomaly Detection, AV-GPS-Dataset, GPS Attacks, Intrusion Detection System, Physics-based Behavior Modeling, Modified Dynamic Bicycle Model.

I. INTRODUCTION

VEHICLES are getting increasingly autonomous and becoming ever more reliant on onboard sensors and communication networks, aiming to make transportation faster, safer, and environmentally sustainable by reducing human intervention in driving tasks [1], [70]. Researchers have shown human error to be the cause of over 90% accidents, resulting in 40 thousand deaths and over 2 million injuries annually in the United States alone [2], [3] — a statistic that will be reduced with the transition to AVs [4]. AVs rely on *Automotive Sensing*, a collection of diverse sensors that enable environmental perception and safe navigation without requiring constant human input [5]. Automotive sensing is broadly categorized into three types: *Self-sensing*, *Surrounding-sensing*, and *Localization* [5]. *Self-sensing* refers to how an autonomous vehicle gathers and interprets information about its own state, including its position, velocity, and acceleration. *Surrounding-sensing* is the ability of a vehicle to perceive its environment, like recognizing traffic signs, understanding weather conditions, or

measuring the state of other vehicles around it. *Localization* determines the local and global positions of the vehicle and helps it to navigate safely to the desired destination. For self-sensing and surrounding sensing, AVs use an array of sensors like Inertial Measurement Units (IMUs), gyroscopes, odometers, Controller Area Network (CAN) bus, cameras, Light Detection and Ranging (LiDARs), etc. [6]. For localization and navigation, they rely on satellite-based navigation systems like Global Navigation Satellite System (GNSS) [7].

GNSSs like GPS (United States), GLONASS (Russian) [35], BeiDou (China) [36], and Galileo (European Union) [37], provide geolocation and time information to a receiver anywhere on or near the Earth. Being the pioneering system, GPS utilizes a 24-satellite constellation to provide its users with highly precise and accurate location and time information for navigation [38]. GPS offers two distinct variants: a secure military-grade GPS that is exclusively accessible to the United States military branches and a civilian GPS that is available for public use [39]. Most autonomous systems, including autonomous vehicles, robots, and drones, rely on the latter variant for navigational purposes. This heavy reliance on civilian GPS raises significant concerns due to the absence of encryption or authentication mechanisms, unlike the military-grade GPS [40]. Researchers have demonstrated that civilian GPS is vulnerable to jamming and spoofing attacks, and commercially available off-the-shelf GPS receivers lack the capability to detect and counteract such attacks [38], [40]–[42]. Various GPS spoofing techniques, including Lift-off-delay [44], Lift-off-aligned [45], Meaconing or Replay [45], Jamming and Spoofing [46], and Trajectory Spoofing [47], pose serious threats to the integrity of GPS. Such spoofing attacks can result in navigational errors, potential vehicle hijacking, or fatal collisions.

Motivated by these challenges, this paper presents GPS-IDS, an Anomaly Behavior Analysis-based Intrusion Detection System that uses a novel physics-based vehicle behavior model — a modification of the conventional dynamic bicycle model, to detect GPS spoofing attacks on AVs. Temporal features extracted from this vehicle behavior model are used to capture the normal behavior of the AV. Afterward, Machine Learning models are employed to detect the modeled normal behavior from abnormal behavior.

The main contributions of this paper are as follows:

- The paper presents the *Autonomous Vehicle Behavior Model*— a modified dynamic bicycle model that integrates an autonomous GPS navigation model. This modification accurately represents the normal navigation behavior of an AV, capturing the lateral dynamics and

states. Temporal features extracted from the AV Behavior Model are used to observe the vehicle's normal behavior. This normal behavior is separated from the attacks using machine learning techniques.

- The paper introduces the "AV-GPS-Dataset" that captures 44 features of GPS-guided navigation of AVs with and without cyber-attacks. In contrast to the existing related datasets collected from simulated environments [8], [10], [11], this dataset is collected from practical field experiments with real-world GPS spoofing attacks performed on an autonomous vehicle testbed.

- The proposed GPS-IDS framework is evaluated on the AV-GPS-Dataset, showing an F1 score of up to 94.4% with up to 13s detection time improvement compared to the Extended Kalman Filter (EKF) detector implemented in the experimental setup.

The rest of the paper is organized as follows: Section II discusses the related work; Section III introduces the GPS-IDS framework and explains the Autonomous Vehicle Behavior Model; Section IV presents the experimental evaluation of the GPS-IDS framework along with the dataset details, and Section V concludes the paper.

II. RELATED WORK

A. Vehicle Models

1) Physics-based Model: A physics-based vehicle model is a mathematical representation that simulates the vehicle's physical behavior using factors like kinematics, dynamics, motion, forces, and environmental interactions. In [15], the authors adopted a physics-based dynamic vehicle model incorporating an EKF estimator to determine the vehicle's longitudinal and lateral velocities and yaw rate. In [30], Kong et al. explored two physics-based vehicle models, namely kinematic and dynamic bicycle models, for model-based controller design in autonomous driving and presented a comparison between them using experimental data. Several research contributions have employed software tools such as Modelica, Robot Operating System (ROS), Gazebo, and Simulink to simulate physics-based vehicle models or their components. Notable applications include racing car modeling [16], vehicle drivability modeling [9], heterogeneous physical system modeling, and simulating AV testbeds [17].

2) Data-centric Model: A data-centric vehicle model uses data-driven techniques and machine learning algorithms to represent the vehicle system. The model analyzes a large volume of data from sensors and sources to determine the vehicle's actions. In [18] the authors proposed a data-driven modeling approach based on Deep Neural Networks (DNNs) to compute and predict the dynamic characteristics of a vehicle. In [19], a data-driven system identification and control method is proposed for autonomous vehicles, where several vehicle dynamic variables are measured and corresponding data is collected to model a physical vehicle. In [20], the authors conducted a comparative analysis between physical and data-driven vehicle models under real-world driving conditions.

B. Intrusion Detection System (IDS)

Intrusion Detection Systems or IDSs are software designed to detect cyber-attacks and can be classified into four types [12]: 1) Signature-based IDS, 2) Anomaly-based IDS, 3) Specification-based IDS, and 4) Hybrid IDS. Signature-based IDSs rely on a pre-learned attack signature database, where each observation matched with an entry in the attack database is considered as an attack. Signature-based IDSs heavily rely on an up-to-date attack signature database and are unable to detect new or modified attacks [13]. An Anomaly-based IDS relies on modeling techniques to capture the system's normal behavior, wherein any observed behavior outside the model-defined norm is classified as malicious. This reliance on normal behavior models allows Anomaly-based IDSs to detect new or modified attacks at the cost of increased modeling complexity [14]. Specification-based IDSs use a set of rules and policies that define the expected behavior of different system components such as sensor nodes or motor commands. Hybrid IDSs are a combination of Signature-based, Anomaly-based, and Specification-based IDSs. The Anomaly-based IDS approach presented in this work detects GPS spoofing attacks on AVs by utilizing an extensive physics-based AV behavior model and real-life vehicular data, aiming to use a more accurate operational baseline and overcome the high error rates present in Anomaly-based IDSs.

C. Cyber-attacks on Vehicles

The increasing deployment of AVs increases the cyber-attack surfaces within the vehicle system that can be potentially exploited. In [56], Hoppe et al. exploited vulnerabilities present in the CAN-bus of a vehicle to attack the windows, lights, and airbags. Similarly, Miller et al. [57] successfully attacked a Jeep Cherokee 2014 by reprogramming a gateway chip in the head unit of the vehicle. The attack enabled the vehicle to send arbitrary CAN messages, allowing access to different critical subsystems like braking and steering [58]. Similar attacks have been demonstrated against Toyota Prius 2010 and Ford Escape 2010 [61], by targeting different vehicular Electronic Control Units (ECU) and head units [60]. Similar to the CAN bus attacks, researchers have successfully targeted navigation systems in AVs like the GPS receiver [21], leading to hijacking attacks.

D. GNSS Security

In this section, we highlight the research aimed at detecting GNSS attacks. GNSS security literature can be classified into two broad categories: (1) GPS signal characteristics-based approach [21], [22], [23], [24], [25], [26], and (2) Machine learning-based approach [65], [69] [66], [10], [67], [68]. The GPS signal characteristics-based approaches rely on signal processing techniques to detect attacks. For instance, in [21], Psiaki and Humphreys proposed a GPS spoofing attack detection scheme based on the direction-of-arrival (DOA) induction. He et al. [23] proposed the use of GPS signal distortion to detect GPS spoofing attacks. For connected and autonomous vehicles, an anomaly detection model is proposed by Yang

et al. [24] to detect GPS spoofing on the localization system using the "Learning From Demonstration" technique. Milaat and Liu [25] proposed a decentralized technique where vehicles exchange GPS code pseudo-range values with neighbors using short-range communications to detect high correlations during spoofed GPS signal arrival. On the other hand, machine learning-based approaches rely on machine learning and data analytics techniques to detect attacks. Researchers have used one-class classifiers [10], Artificial Neural Networks [65], Long Short Term Memory (LSTM) networks [66], [67] to detect GPS attacks. However, the majority of these works adopt a segregated approach focused on sensor node-specific detection techniques and either use a small (and restrictive) GPS dataset, or use an attack dataset collected from simulated environments using tools like Matlab and Gazebo, lacking the representation of the real world.

This paper overcomes these limitations by introducing an anomaly-based intrusion detection approach and using real-world data. The approach relies on a novel modified dynamic bicycle model with GPS navigation to model the normal behavior of an AV. Temporal features highlighted in this behavior model are extracted from real-world experiments to model the vehicle's normal navigation behavior. Machine learning techniques are then used to classify the normal navigation behavior from the abnormal.

III. GPS INTRUSION DETECTION SYSTEM (GPS-IDS)

GPS-IDS is an Anomaly Behavior Analysis-based Intrusion Detection System designed to identify GPS attacks on AVs by continuously monitoring the behavior of the vehicle and considering any deviation from the expected behavior as anomalous. The framework relies on a physics-based vehicle model, that is a modified dynamic bicycle model, called the *Autonomous Vehicle Behavior Model* to represent the vehicle's normal behavior, as the physics model captures the vehicle'ps state at any instantaneous time t. Features highlighted in the physics model are used in machine learning analysis to detect anomalous behavior of the vehicle. Fig. 1 shows the architecture of the GPS-IDS framework.
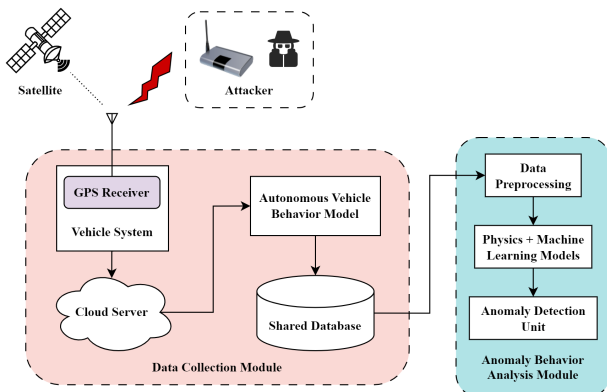

Fig. 1: GPS-IDS Architecture

A. Components of GPS-IDS

The GPS-IDS framework has two modules: the *Data Collection Module*, and the *Anomaly Behavior Analysis Module*.

1) Data Collection Module: This module collects the raw data from the AV system and stores it in a shared database. Raw data includes GPS signals and an ongoing stream of vehicle dynamics data, which are obtained from the onboard state-measuring sensors.

2) Anomaly Behavior Analysis Module: This module reads the raw data from the shared database and performs Anomaly Behavior Analysis to classify the data as normal or abnormal.

B. Anomaly Behavior Analysis (ABA) for GPS-IDS

Satam et al. have presented an IDS for wireless networks based on Anomaly Behavior Analysis [13], [27], which forms the basis of this work. The presented GPS-IDS is defined over a finite set of driving events U. Set U is partitioned into two subsets: $Normal$ events N and $Abnormal$ or $Attack$ events A, such that $N \cup A = U$ and $N \cap A = \emptyset$. To characterize U, a representation map R is used, which maps events in U to patterns in U^R such that $U \xrightarrow{R} U^R$. Likewise, N^R and A^R respectively represent the events in N and A, such that $N \xrightarrow{R} N^R$, $A \xrightarrow{R} A^R$, and $N^R \cup A^R = U^R$. A detector D is defined as $D = (f_{norm}, M)$; where f_{norm} is the normal behavior characterization function expressed as $f_{norm} : U^R \times M \Rightarrow [0, 1]$ and M is the system memory that stores the normal behavior model extracted from the set of normal events N^R. Function f_{norm} specifies the degree of abnormality of a sample $s \in U^R$ by comparing it with M. The higher the value of $f_{norm}(s, M)$, the more abnormal the sample is. If the value of $f_{norm}(s, M)$ exceeds a predefined threshold $\mathbb{T}$, detector D raises an alarm indicating the occurrence of an abnormal or attack event. We can consider D for any sample $s \in U^R$ as:

$$D(s) = \begin{cases} Abnormal & if \quad f_{norm}(s, M) > \mathbb{T} \\ Normal & otherwise \end{cases}$$

Detection takes place when the detector D classifies a sample as abnormal, regardless of whether it is genuinely an anomaly or a regular sample that has been wrongly classified as one. The detection errors are defined over a test set U_t^R which is a subset of U^R, $U_t^R \subseteq U^R$. The detector considers two kinds of errors: *False Positives* and *False Negatives*. A *False Positive* detection occurs when a normal sample $s \in N^R$ is detected as an abnormal event and is defined as $\varepsilon^+ = \{s \in N^R | D(s) = abnormal\}$, while a *False Negative* detection occurs when the detector classifies an abnormal sample $s \in A^R$ as a normal event (undetected anomalies), that is $\varepsilon^- = \{s \in A^R | D(s) = normal\}$. The objective of GPS-IDS is to tune the predefined threshold $\mathbb{T}$ so that the overall error is minimized. Particularly, we prioritize the minimization of *False Negative* errors, as these undetected attacks pose greater risks compared to false alarms in the context of autonomous driving.

C. Autonomous Vehicle Behavior Model

In order to establish a comprehensive behavior model of AVs, it is essential to consider four key components: (1)

Perception and Localization, (2) State Estimation, (3) Motion Planning, and (4) Control [28]. Perception and Localization rely on a combination of internal and external sensors, such as IMUs, Cameras, LiDARs, GPS, etc., to collect information about the surrounding environment. The remaining three components require a mathematical representation of the vehicle that encompasses its dynamics and motion. This study focuses specifically on GPS-guided localization in AVs, aiming to investigate the impact of a spoofing attack on the vehicle's dynamics and motion. Therefore, predicting the dynamic state of the vehicle becomes crucial. To achieve this, a simple 2 Degrees of Freedom (2 DOF) lateral dynamics model is considered, and the bicycle model proposed by Rajamani et al. [29] is employed. Fig. 2 depicts the dynamic bicycle model of a vehicle in a 2-dimensional inertial frame.

1) Dynamic Bicycle Model: The dynamic bicycle model is a simplified representation of a vehicle's dynamics and considers the effects of external forces and yaw moments acting on the vehicle, which results in an accurate calculation of dynamic parameters [29]. In a dynamic bicycle model, the inertial position coordinates and the orientation of the vehicle are defined as follows [30]:

$$v_x = \dot{x} = v\cos(\psi + \beta) \tag{1}$$

$$v_y = \dot{y} = v\sin(\psi + \beta) \tag{2}$$

$$r = \dot{\psi} = \frac{v}{l_r}\sin\beta \tag{3}$$

where x and y are the coordinates of the center of mass of the vehicle in frame (X, Y); v_x and v_y represent the longitudinal and lateral velocities of the vehicle, respectively; ψ is the yaw angle, which is the orientation of the vehicle with respect to the x-axis; r is the yaw rate or the rate of change of the yaw angle; β is the angle of the current velocity of the center of mass with respect to the longitudinal axis of the vehicle; δ is the controlled steering angle of the front wheels; and l_f and l_r are the distances of the front and the rear wheel
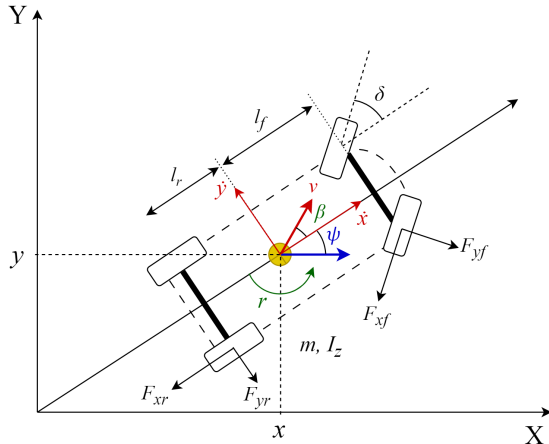

Fig. 2: Dynamic bicycle model of an autonomous vehicle in a 2-dimensional inertial frame

axles from the center of mass, respectively. The differential equations associated with the dynamic bicycle model are:

$$\ddot{x} = \dot{\psi}\dot{y} + a_x \tag{4}$$

$$\ddot{y} = -\dot{\psi}\dot{x} + \frac{2}{m}(F_{yf}\cos\delta + F_{yr}) \tag{5}$$

$$\dot{r} = \ddot{\psi} = \frac{2}{I_z}(l_f F_{yf} - l_r F_{yr}) \tag{6}$$

$$\dot{X} = \dot{x}\cos\psi - \dot{y}\sin\psi \tag{7}$$

$$\dot{Y} = \dot{x}\sin\psi - \dot{y}\cos\psi \tag{8}$$

where $\dot{x}$ and $\dot{y}$ denote the longitudinal and lateral velocities of the vehicle, respectively; a_x is the acceleration of the center of the mass; $\dot{\psi}$ or r is the yaw rate; m and I_z denote the vehicle's mass and yaw inertia, respectively; and F_{yf} and F_{yr} denote the lateral tire forces at the front and rear wheels of the vehicle, respectively. From the dynamic bicycle model, Newton-Euler's equations of motion are defined as follows:

$$\begin{bmatrix} \boldsymbol{F}_x \\ \boldsymbol{F}_y \end{bmatrix} = m\begin{bmatrix} \dot{v}_x - \dot{\psi}v_y \\ \dot{v}_y - \dot{\psi}v_x \end{bmatrix} = \begin{bmatrix} -F_{xf}\cos\delta - F_{yf}\sin\delta - F_{xr} \\ F_{yf}\cos\delta - F_{xf}\sin\delta + F_{yr} \end{bmatrix} \tag{9}$$

$$\boldsymbol{\tau} = I_z\ddot{\psi} = I_z\dot{r} = l_f(F_{yf}\cos\delta - F_{xf}\sin\delta) - l_r F_{yr} \tag{10}$$

These dynamics can be simplified by disregarding the aerodynamic resistance and setting the longitudinal tire forces, F_{xf} and F_{xr}, to zero. The lateral tire forces, F_{yf} and F_{yr} are calculated by using a linear tire model, which simplifies the nonlinear characteristics of tire dynamics by establishing a linear relationship between the tire slip angles and the resulting tire forces. Considering that, F_{yf} and F_{yr} are defined as [31]:

$$F_{yf} = -C_{yf}\alpha_f \quad (11) \qquad F_{yr} = -C_{yr}\alpha_r \quad (12)$$

where C_{yf} and C_{yr} are the cornering stiffness coefficients, and α_f and α_r are the slip angles of the front and rear wheels, respectively. Assuming small slip angles, we obtain [31]:

$$\alpha_f = \frac{v_y + l_f r}{v_x} - \delta \quad (13) \qquad \alpha_r = \frac{v_y - l_r r}{v_x} \quad (14)$$

Finally, a simplified 2 DOF non-linear state space representation of the lateral dynamics of the vehicle can be expressed as follows [62], [63]:

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{bmatrix}\begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{21} \end{bmatrix}\delta \tag{15}$$

where,

$$\mathbf{a}_{11} = \frac{C_{yf} + C_{yr}}{mv_x} \qquad \mathbf{a}_{12} = \frac{l_f C_{yf} - l_r C_{yr}}{mv_x^2}$$

$$\mathbf{a}_{21} = \frac{l_f C_{yf} - l_r C_{yr}}{I_z} \qquad \mathbf{a}_{22} = \frac{l_f^2 C_{yf} - l_r^2 C_{yr}}{I_z v_x}$$

$$\mathbf{b}_{11} = -\frac{C_{yf}}{mv_x} \qquad \mathbf{b}_{21} = -\frac{l_f C_{yf}}{I_z}$$

In the state space model of equation 15, the input is the steering angle δ and the states are the lateral velocity v_y and yaw rate r.

2) Perception/ Localization: The GPS enables the vehicle to determine its position and localize to the destination. Since this paper focuses on presenting a GPS intrusion detection system, only GPS-based localization is considered, and camera or vision sensor-based perception is not taken into account. To ensure safe GPS-guided localization for autonomous vehicles, it is necessary to continuously monitor various parameters such as GPS latitude and longitude, signal quality, GPS Dilution of Precision (DOP), and the number of satellites the vehicle is locked with. The normal behavior of the GPS signal for vehicular localization can be modeled by constantly monitoring the following parameters:

$$s_{gps}(t) = (lat, lon, dop, sat_{lock}, sat_{count})^t \qquad (16)$$

where $s_{gps}(t)$ is the incoming GPS signal from satellites at time t; lat and lon are the GPS latitude and longitude at t, respectively; dop denotes the Dilution of Precision (DOP) or the quality of the incoming GPS signal at t; and sat_{lock} and sat_{count} respectively denote the number of satellites the vehicle is locked with and the number of available satellites at t. It is noteworthy that the current GPS-guided localization behavior model only considers positional parameters and signal strength, excluding the analysis of physical layer parameters of GPS signal due to existing research on such GPS attack detectors, as outlined in section II-D.

3) State Estimation: One of the well-established solutions for estimating the state of nonlinear systems is the Extended Kalman Filter or EKF. It integrates measurements from multiple sensors with a system model to estimate the state of the system with improved accuracy [53]. In the context of a GPS-guided autonomous vehicle, the EKF utilizes multiple sensor measurements in conjunction with the dynamic vehicle model to make accurate predictions about the vehicle's state, including its position, orientation, velocity, and other relevant parameters. This estimation procedure takes into account the nonlinear relationship between the measurements and the position, while also considering measurement noise. As highlighted in the Related Work section, EKF state estimator is capable of detecting sensor anomalies and attacks against autonomous vehicles by monitoring the measurement deviations of multi-sensor readings [48], [55]. A similar EKF dynamic state estimator has been employed in our vehicle model and experimental testbed. In later sections, it will be shown that the GPS-IDS approach is able to detect a GPS spoofing attack faster than the employed EKF state estimation-based detection approach, which is supported by experimental validation.

For ease of description, we can rewrite equation 15 in the following forms:

$$\left. \begin{array}{l} \dot{\mathbf{x}}(t) = f_{state}(\mathbf{x}(t), u(t)) \\ \dot{\mathbf{x}} = \mathbf{a}x + \mathbf{b}u \end{array} \right\} \qquad (17)$$

where vehicle state $\mathbf{x} = [v_y \ r]^t$, control input $u = [\delta]^t$, and f_{state} is the nonlinear function reproducing equation 15. According to the dynamic bicycle model, the normal behavior of the vehicle's lateral dynamics is characterized by the state parameters, v_y and r, and the steering angle δ. In conjunction with the lateral dynamics model, the EKF allows us to fuse and compare multiple state-measuring sensor readings to estimate the vehicle's state with improved accuracy. To incorporate an EKF-based estimation, the vehicle can be described as a discrete time-varying system in the following forms:

$$\mathbf{x}_{k+1} = f_{cd}(\mathbf{x}_k, u_k, W_k) \qquad\qquad \mathbf{y}_{k+1} = g_{cd}(\mathbf{x}_k, E_k)$$

where f_{cd} is the prediction equation; $\mathbf{x}_k$ and u_k are the state and the input at the kth time, respectively; W_k is the prediction noise; g_{cd} is the observation equation; and E_k denotes the observation noise. The Jacobian matrix of the nonlinear prediction and observation equation are defined as:

$$F = \left. \frac{\partial f_{cd}}{\partial \mathbf{x}} \right|_{\hat{x}_{k-1}, u_k} \qquad\qquad G = \left. \frac{\partial g_{cd}}{\partial \mathbf{x}} \right|_{\hat{x}_k}$$

Thus, the nonlinear system can be transformed into a linear system with the following equations:

$$\mathbf{x}_{k+1} = F\mathbf{x}_k + W_k \qquad\qquad \mathbf{y}_{k+1} = G\mathbf{x}_k + E_k$$

The update process is as follows:

$$\mathbf{P}_{k+1|k} = F_k \mathbf{P}_{k|k} F_k^T + \mathbf{Q}$$

$$\mathbf{S}_{k+1|k} = G_k \mathbf{P}_{k+1|k} G_k^T + \mathbf{R}$$

$$\mathbf{K}_{k+1|k} = \frac{\mathbf{P}_{k+1|k} G_k^T}{\mathbf{S}_{k+1|k}}$$

$$\mathbf{P}_{k+1|k+1} = (\mathbf{I} - \mathbf{K}_{k+1|k} G_k) \mathbf{P}_{k+1|k}$$

where $\mathbf{Q}$ is the covariance matrix of the prediction noise, $\mathbf{R}$ is the covariance matrix of the observation noise, and $\mathbf{I}$ is an identity matrix. Thus:

$$\mathbf{x}_{k+1|k} = F\hat{\mathbf{x}}_{k|k} + W_k \qquad\qquad \mathbf{y}_{k+1|k} = G\hat{\mathbf{x}}_{k|k} + E_k$$

The final EKF estimated value can be obtained by:

$$\hat{\mathbf{x}}_{k+1|k+1} = x_{k+1|k} + \mathbf{K}_{k+1|k}(\mathbf{y}_{k+1} - \mathbf{y}_{k+1|k})$$

4) Motion Planning: The vehicle plans a safe and efficient path to follow from its current position (x, y) to the target destination (x_t, y_t). It uses GPS to determine its current position and generates a continuous sequence of target yaw angles, $\psi_t(t)$ and cross-track errors, $e(t)$ to guide itself to the desired path. The cross-track error refers to the perpendicular distance of the vehicle from the current position to the desired path. Algorithm 1 outlines the target yaw angle calculation process from GPS coordinates [32].

Algorithm 1 Target Yaw Calculation from GPS Coordinates

1: **Function** $TargetYawFromGPS$ (current_latitude, current_longitude, target_latitude, target_longitude)
2: **while** ($TargetYawFromGPS$ = True) **do**
3:    Δ longitude $\leftarrow$ target_longitude $-$ current_longitude
4:    $p = \sin$ (Δ longitude) $\times \cos$ (target_latitude)
5:    $q = \cos$ (current_latitude) $\times \sin$ (target_latitude) $- \sin$ (current_latitude) $\times \cos$ (target_latitude) $\times \cos$ (Δ longitude)
6:    $\psi_t = arctan(p, q)$ \qquad [ψ_t = target yaw angle]
7:    **return** ψ_t
8: **end while**

From the calculated target yaw, $\psi_t(t)$ and current yaw, $\psi(t)$, the motion planning algorithm can calculate the instantaneous cross-track error, $e(t)$ using the following relationship:

$$e(t) \propto \sin\left(\psi_t(t) - \psi(t)\right) \quad (18)$$

From 18, we can say that when the vehicle is traveling along the desired path, $\psi(t) = \psi_t(t)$, and $e(t) = 0$. When the vehicle deviates from the desired path, $\psi(t)$ differs from $\psi_t(t)$ and the $e(t)$ increases accordingly. In general, an increase in the difference between the target yaw angle and the current yaw angle leads to a corresponding amplification in the cross-track error.

5) Controller: The vehicle requires a control algorithm to mitigate the cross-track error and navigate safely. There are different types of controllers depending on vehicle-specific models and computational resources, such as Model Predictive Controller (MPC), Fuzzy Logic Controller, Proportional-Integral-Differential (PID) Controller, etc. In this study, we focus on the PID controller as it is one of the most widely used control algorithms and easy to implement on testbeds.

The PID controller is typically used for controlling the *Position* (e.g. latitude and longitude) and *Attitude* (e.g. yaw/heading angle) of an AV. The controller continuously adjusts the steering and acceleration control signals to minimize the cross-track error. According to the proportional (P), integral (I), and derivative (D) terms of $e(t)$, the control signal $u_c(t)$ is obtained, which is formulated as follows:

$$u_c(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (19)$$

where K_p, K_i, and K_d are the Proportional, Integral, and Differential gain coefficients, respectively. The term $e(t)$ denotes the present cross-track error, $\int_0^t e(t)dt$ denotes the accumulated cross-track error over time, represented as the definite integral of $e(t)$ with respect to time, and $\frac{de(t)}{dt}$ is the change in cross-track error with respect to time, represented as the derivative of $e(t)$ with respect to time. The final control signal $u_f(t)$ will be comprised of an acceleration component and a steering angle component, which can be expressed as:

$$u_f(t) = u_{acc}(t) + u_{str}(t) = K_{acc}a(t) + K_{str}\delta(t) \quad (20)$$

where $u_{acc}(t)$ and $u_{str}(t)$ are the acceleration and steering control signal components, respectively. K_{acc} is the proportional gain for acceleration, K_{str} is the proportional gain for steering, and $a(t)$ and $\delta(t)$ are instantaneous acceleration and steering angle, respectively. Fig. 3 illustrates the PID cascade control architecture used to control the position and attitude of the AV model.

Derived from the discussed software components, the architecture of the Autonomous Vehicle Behavior Model is depicted in Fig. 4.

D. Attacker Model

We focus on the GPS spoofing attack that interferes with the target AV by injecting a malicious GPS signal into the
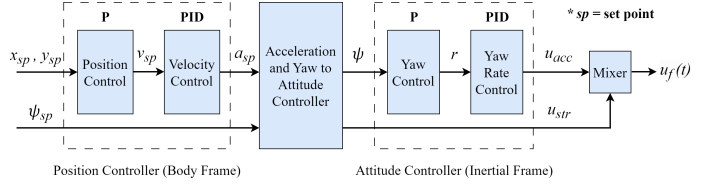

Fig. 3: PID Cascade Control Architecture

vehicle's GPS receiver unit. The attacker is modeled as a malicious entity capable of generating, recording, and transmitting fake GPS signals with false coordinates corresponding to any location of his choice wirelessly using a spoofer device. The attacker can be an internal entity who is inside the victim vehicle and very close to the GPS receiver, or an external entity who is moving along with the victim vehicle with a directional antenna pointed toward it and staying within a specified range. It is assumed that the attacker only knows the current location of the victim vehicle and has no knowledge about (1) the low-level control algorithm settings; (2) control commands from the autonomous navigation system of the vehicle. In this study, it is shown that even though the attacker has no prior knowledge about the vehicle's control system, a GPS spoofing attack is capable of compromising the closed-loop control system and the measured states of the vehicle.

1) Mathematical model of the spoofed GPS signal: To model a spoofed GPS signal, the attacker must replicate the Radio Frequency carrier wave, the Pseudorandom noise code (PRN), and the data bits of each GPS signal that he or she intends to spoof. As stated by Psiaki and Humphreys, [33] a typical GPS signal $y(t)$ takes the following form:

$$y(t) = \mathbf{Re}\left\{\sum_{i=1}^N A_i D_i[t - \tau_i(t)]C_i[t - \tau_i(t)]e^{j\varphi}\right\}$$

$$\varphi = \omega_c t - \phi_i(t)$$

where N is the number of individual signals transmitted by each GPS satellite. A_i, $D_i(t)$, and $C_i(t)$ correspond to the carrier amplitude, data bit stream, and spreading code (often a Binary Phase Shift Keying, BPSK-PRN code or a Binary Offset Carrier, BOC-PRN code) of the ith signal, respectively. $\tau_i(t)$ is the ith signal's code phase, ω_c is the nominal carrier frequency, and $\phi_i(t)$ is the ith beat carrier phase. The attacker sends a set of spoofed signals $y_{spf}(t)$ that are similar to as follows:

$$y_{spf}(t) = \mathbf{Re}\left\{\sum_{i=1}^{N_{spf}} A_{i_{spf}}\hat{D}_i[t - \tau_{i_{spf}}(t)]C_i[t - \tau_{i_{spf}}(t)]e^{j\hat{\varphi}}\right\}$$

$$\hat{\varphi} = \omega_c t - \phi_{i_{spf}}(t)$$

where N_{spf} is the number of spoofed signals (typically $N_{spf} = N$). Each spoofed signal must have the same spreading code $C_i(t)$ as the corresponding true signal in order to deceive the receiver, and usually, it broadcasts its best estimate of the same data bit stream $\hat{D}_i(t)$. The spoofed amplitudes, code phases, and carrier phases are, respectively, $A_{i_{spf}}$, $\tau_{i_{spf}}(t)$, and $\phi_{i_{spf}}(t)$ for $i = 1, ..., N_{spf}$. These quantities are likely to differ from their true counterparts for reasons that are specific to the type of attack that is being mounted. During a spoofing
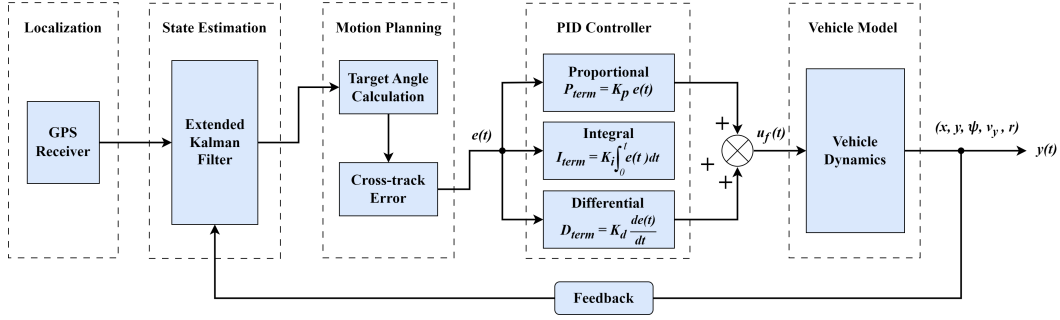
Fig. 4: Autonomous Vehicle Behavior Model

attack, the total signal received, $y_{total}(t)$ at the victim vehicle's GPS receiver antenna is:

$$y_{total}(t) = y(t) + y_{spf}(t) + \nu(t) \tag{21}$$

where $\nu(t)$ is the noise component, which can either be naturally generated or be contributed by the attacker.

2) Mathematical model of the attack impacts: If the attack is successful, the GPS receiver of the vehicle will receive incorrect location information, and the current latitude and longitude will be replaced by spoofed latitude and longitude. The function $TargetYawFromGPS()$ in Algorithm 1 takes the spoofed latitude and longitude as inputs and calculates the spoofed yaw angle, $\psi_{t_{spf}}$. This means that the proportionality of equation 18 calculates a spoofed cross-track error, $e_{spf}(t)$. The PID controller takes $e_{spf}(t)$ as input and equation 19 and 20 become:

$$u_{spf}(t) = \begin{cases} K_p e_{spf}(t) + K_i \int_0^t e_{spf}(t)dt + K_d \frac{de_{spf}(t)}{dt} \\ K_{acc} a_{spf}(t) + K_{str}\delta_{spf}(t) \end{cases} \tag{22}$$

where $u_{spf}(t)$, $a_{spf}(t)$, and $\delta_{spf}(t)$ represent the spoofed control signal, spoofed acceleration, and spoofed steering angle, respectively. $u_{spf}(t)$ goes to the vehicle system as actuator command, which results in $\delta_{spf}(t)$ and the vehicle loses its control at a time instance t. The state space representation of equation 15 thus becomes:

$$\begin{bmatrix} \dot{v}_{y_{spf}} \\ \dot{r}_{spf} \end{bmatrix} = \begin{bmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} \mathbf{b}_{11} \\ \mathbf{b}_{21} \end{bmatrix} \delta_{spf} \tag{23}$$

where $[\dot{v}_{y_{spf}} \ \dot{r}_{spf}]$ is the spoofed lateral dynamic state of the vehicle. In this state, the vehicle will start to deviate from its normal lateral dynamic behavior and move toward the spoofed GPS location. Based on the discussion, the flow of the GPS spoofing attack is summarized in Algorithm 2.

E. Problem Statement

In autonomous driving applications, GPS spoofing attacks involve the deliberate manipulation of GPS signals to deceive the vehicle's onboard navigation system, which can cause the vehicle to deviate from its intended path, leading to accidents or even hijacking. To ensure the safety of passengers and pedestrians, attacks must be detected in a timely manner. Based on the above discussions, the GPS intrusion detection problem investigated in this paper can be stated as follows:

Algorithm 2 GPS Spoofing Attack

Input: Spoofed signal y_{spf}, Duration of attack t_{attack}, Bias angle caused by attack ϑ, Current state $\mathbf{x}$, Spoofed state $\mathbf{x}_{spf}$, current_latitude, current_longitude spoofed_latitude, spoofed_longitude
Output: Attack flag $\mathbf{A}$, State update $\mathbf{x}_{new}$

1: Attacker sends y_{spf}
2: **if** receiver antenna receives $y_{total} = y + y_{spf} + \nu$ **then**
3:    $\mathbf{A}$ = True
4: **else**
5:    $\mathbf{A}$ = False
6:    **while** (t_{attack} = True) **do**
7:      Compute $\psi_{t_{spf}}$ using $TargetYawFromGPS$
8:      (current_longitude, current_latitude,
9:      spoofed_longitude, spoofed_latitude)   [algorithm 1]
10:     $\psi_{t_{spf}} \leftarrow \psi_t + \vartheta$ [update ψ_t with bias ϑ]
11:     $e_{spf} \Rightarrow \boldsymbol{f}(\psi_{t_{spf}} - \psi_t)$ [using equation 18]
12:     $u_{spf} \Rightarrow e_{spf}$ [using equation 22]
13:     $\delta_{spf} \Rightarrow u_{spf}$
14:     Update $\mathbf{x}$ [using equation 17]
15:    **end while**
16: **end if**

Let us consider an AV modeled by Fig. 4, that has a state space representation described by equation 15 under normal conditions and equation 23 under the influence of a GPS spoofing attack. Assuming the vehicle is under normal operating conditions at the initial time, we have to design an anomaly-based intrusion detection strategy to detect GPS anomalies and, in turn, detect the GPS attacks.

IV. EXPERIMENTAL EVALUATION

To validate the proposed GPS-IDS framework and demonstrate its effectiveness, field experiments have been conducted using an AV robotic testbed. The testbed has been designed and developed by following the Autonomous Vehicle Behavior Model presented in Fig. 4. This testbed has been utilized to perform GPS spoofing attack experiments and collect relevant data. The collected datasets have been used to conduct a series of experiments and validate the GPS-IDS framework.

A. Autonomous Vehicle Testbed (AVT)

The Autonomous Vehicle Testbed or AVT is a custom-built autonomous rover that can navigate through a predefined path

using GPS guidance. It utilizes an array of state-measuring sensors, including accelerometer, magnetometer, barometer, gyroscope, and digital compass for accurate state estimation. In accordance with the Autonomous Vehicle Behavior Model, the AVT employs a PID controller and a multi-sensor fusion-based EKF failsafe. The EKF failsafe is triggered when the EKF variances associated with any two state-measuring sensor readings exceed a predefined EKF threshold value for 1 second. The AVT uses Ardupilot [64], an open-source software and hardware platform designed for building custom unmanned ground and aerial robotic vehicles. An Ardupilot-based autopilot generates the velocity and steering angle commands from the GPS and feedback from the state-measuring sensors of the AVT. These commands are then passed to the PID controller for execution. To program the AVT and define a path to follow using GPS, a Ground Control Station software supported by Ardupilot was utilized. The Ground Control Station computer receives all vehicular data via the telemetry modules in real time and stores them in the local memory of the computer. These vehicular data are referred to as "Dataflash logs", and collected at Ardupilot's default data update rate of 1 Hz.

The hardware architecture of the AVT is divided into two parts: the Rover Unit and the Attacker Unit. The Rover Unit consists of a $1/10^{th}$ scale Radio Controlled truck chassis, a Pixhawk flight controller, a Neo M8N GPS receiver with a built-in compass, telemetry modules, Radio Transmitter-Receiver modules, and other related components. The Attacker Unit consists of a Raspberry Pi-4 model B to generate the spoofed GPS baseband signal data stream. A HackRF One Software Defined Radio (SDR) converts this data stream to Radio Frequency and transmits the spoofed signal with an antenna. The hardware architecture of the AVT is presented in Fig. 5.

B. GPS Spoofing Attack on the AVT

To perform the GPS spoofing attack on the AVT system, a real-life spoofing attack scenario was generated based on the *Attacker Model* outlined in Section III-D of this paper. The attack experiment utilized a single spoofed signal, mathematically denoted as $N_{spf} = 1$. Consistent with the standard L1 GPS signal, the spoofed signal employed a carrier wave frequency of 1575.42 MHz. The navigation message contained information specific to a single spoofed location, which was modulated onto the carrier wave using the BPSK modulation

Fig. 5: Hardware Architecture of the AVT

technique. As outlined in Algorithm 2, the spoofing attack affected the AVT control system and compromised its normal state.

C. Data Collection

The datasets were collected in two distinct locations separated by approximately 4 miles: The University of Arizona campus, and the Alvernon Park— both located in Tucson, Arizona, USA. During data collection, the vehicle was consistently operated in autonomous mode guided by GPS. As a safety precaution, a wireless remote controller was employed for manual control of the vehicle to mitigate potential hazards.

To collect normal data, a circular path consisting of 7 waypoints including the home location was mapped and uploaded to the AVT from the Ground Control Station. Each round of normal data collection was considered completed if the vehicle followed all 7 waypoints and returned to its home location. This way, 180 rounds of normal data were collected while no attack was imposed. To collect attack data, the AVT was operated in autonomous mode for 10 seconds without imposing any attack; afterward, the attack was launched. During one round of attack data collection, the spoofing attack was performed on the vehicle for 300 seconds. This way, 65 rounds of attack data were collected. By extracting the dataflash logs after each autonomous operation, all vehicular data were stored, labeled, and added to the dataset. To distinguish between the two categories of data, the normal and attack data were labeled by 0's and 1's, respectively.

D. Autonomous Vehicle GPS Dataset (AV-GPS-Dataset)

The AV-GPS-Dataset is composed of three subsets, each with varied entries, namely AV-GPS-Dataset 1, AV-GPS-Dataset 2, and AV-GPS-Dataset 3. Each subset contains two classes of data: Normal and Attack data, and they are extracted as Comma Separated Value (CSV) files from the Ground Control Station computer. The AV-GPS-Dataset mainly consists of temporal features of the AVT. The features and labels are consistent across all datasets, ensuring uniformity in the categorization of the data.

AV-GPS-Dataset 1 is the largest subset containing 62,042 entries, out of which, 46,787 are labeled as normal vehicular data (approximately 75%), and 15,255 are labeled as attack data (approximately 25%). This set contains experimental data from two different locations at the University of Arizona Campus. Both locations were chosen so that the AVT could have a clear reception of the GPS signal and no obstacle could block the GPS reception. Here, we collected the normal data and attack data in separate sessions, and afterward, merged the two types of data to form the dataset. The dataset comprises three distinct GPS spoofing scenarios- **Scenario I:** GPS spoofing when the AVT is following a straight line, **Scenario II:** GPS spoofing when the AVT is making turns, and **Scenario III:** GPS spoofing when the AVT is stationary.

AV-GPS-Dataset 2 is the second subset and contains 6,890 entries, with 5,184 labeled as normal (approximately 75%) and 1,706 labeled as attack data (approximately 25%). This dataset was collected from outside the University of Arizona Campus
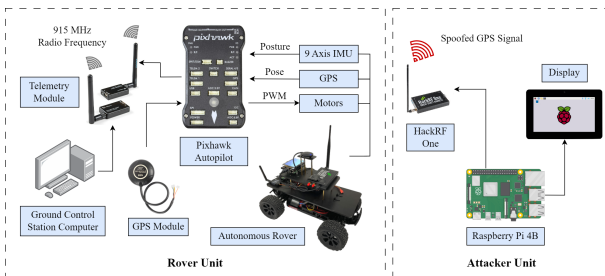
(Alvernon Park), incorporating a different location and driving environment. This location had more trees to obstruct GPS reception partially, making the autonomous operation more challenging for the AVT. This subset was also collected in separate sessions, with normal and attack data merged to form the dataset. It comprises two spoofing scenarios, **Scenario I** and **Scenario III**.

AV-GPS-Dataset 3 is the smallest subset collected from the eastern part of the University of Arizona Campus. It contains only 636 entries, with 241 labeled as normal (approximately 38%) and 395 labeled as attack data (approximately 62%). This subset was collected in a single session to capture the transition between the normal and attack state of the AVT. The complete AV-GPS-Dataset is publicly available at [34], with a comprehensive explanation of the dataset features.
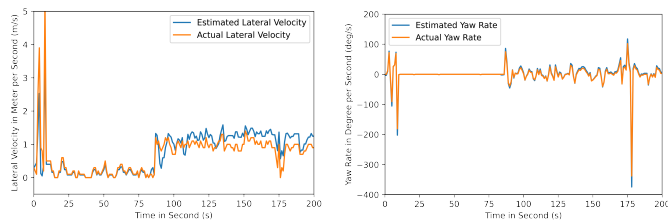
E. Experimental Analysis

In this section, we present the experiments that were used to analyze the AV-GPS-Datasets and evaluate the performance of the GPS-IDS approach.

1) Experiment 1: State Estimation of the AVT using the Dynamic Bicycle Model: This experiment establishes a connection between the Autonomous Vehicle Behavior Model and the actual AVT. In this experiment, we estimated the states of the AVT using the dynamic bicycle model presented in section III-C1. The estimated results are then compared with the actual vehicle states obtained from 46,787 instances of normal data from AV-GPS-Dataset 1. Since a small-scale testbed is used to represent the real vehicle, it is assumed that the distances of the front and the rear wheel axle from the center of mass are equal. It is also assumed that the cornering stiffness coefficients for the front and rear wheels are equal to 1. Table I presents the physical parameters of the AVT that were used to identify the system matrix "**a**" and the input matrix "**b**". Substituting these values in Equation 15, the final dynamic state equation obtained is as follows:

$$\begin{bmatrix} \dot{v}_y \\ \dot{r} \end{bmatrix} = \begin{bmatrix} 0.8 & 0.0 \\ 0.0 & 1.1169 \end{bmatrix} \begin{bmatrix} v_y \\ r \end{bmatrix} + \begin{bmatrix} -0.4 \\ -2.5384 \end{bmatrix} \delta \quad (24)$$

The estimated results of the lateral dynamics of the AVT compared with the actual values are shown in Fig. 6. From Fig. 6, it can be observed that the estimated states using the dynamic bicycle model follow a similar distribution as the actual values of the yaw rates and lateral velocities. Therefore,

TABLE I: Physical Parameters of the Autonomous Vehicle Testbed (AVT)

AVT Parameters	Values
Mass, m	2.5 kg
Length, l	0.56 m
Width, w	0.32 m
Yaw moment of inertia, I_z	0.0867 $kg - m^2$
Distance between the front wheel axle and Center of Mass, l_f	0.22 m
Distance between the rear wheel axle and Center of Mass, l_r	0.22 m
Nominal velocity, v	1 m/s

the derived dynamic bicycle model can represent the lateral dynamics of the AVT well. This points out the necessity of adapting the dynamic bicycle model to capture the behavior of the autonomous vehicle and collect data on the parameters identified by the vehicle behavior model. Since the presented vehicle model is capable of estimating the distributions of the next states of the AVT correctly, the state space representation accurately models the testbed. The differences in these values can be accepted based on the assumption that the field experiments added some noise that is not considered in the state equation.

2) Experiment 2: Attack Impact on Pose Measurements and Controller Input/ Output: This experiment presents a comprehensive analysis of the behavior of the AVT under normal operating conditions and under the influence of GPS spoofing attack. We initially depicted the AVT's normal behavior in terms of pose measurements and controller input/ output by plotting one round of normal data, as shown in Fig. 7. Fig. 7a and Fig. 7b illustrate the AVT's position and orientation, while Fig. 7c and Fig. 7d illustrate the cross-track error and velocity, respectively, under normal operating conditions. From Fig. 7, it is observed that the AVT follows a circular trajectory with a velocity ranging from 0 m/s to 2.5 m/s. During its path traversal, it effectively corrects any cross-track errors and adjusts its yaw angles accordingly.

To facilitate a clear comparison of the attack impact, we then plotted one round of normal data with the corresponding attack data pertaining to the pose measurements and controller input/ output of the AVT, as presented in Fig. 8. In Fig. 8, it is apparent that the AVT's position is significantly affected by the spoofed GPS signals. It deviates from its regular circular trajectory and moves toward a different direction along the X coordinate (Fig. 8a) at an irregular velocity spiking to as high as 80 m/s, and then decelerating to 0 m/s (Fig. 8d). In Fig. 8c, we can observe an abnormal rise in cross-track errors exceeding 740 meters, then dropping to around 0 meters after 35 seconds and continuing to be the same. The AVT was unable to correct this error and adjust its heading, causing the yaw angle to remain unchanged (Fig. 8b). During this time, the AVT had lost all the satellite locks and remained stationary due to the variance in EKF estimates. Nonetheless, on the Ground Control Station map, the AVT was observed virtually progressing toward the location indicated by the spoofed signals, which is shown in Fig. 9.

3) Experiment 3: Performance Analysis of the Machine Learning Models with Different Training Sets: This experi-
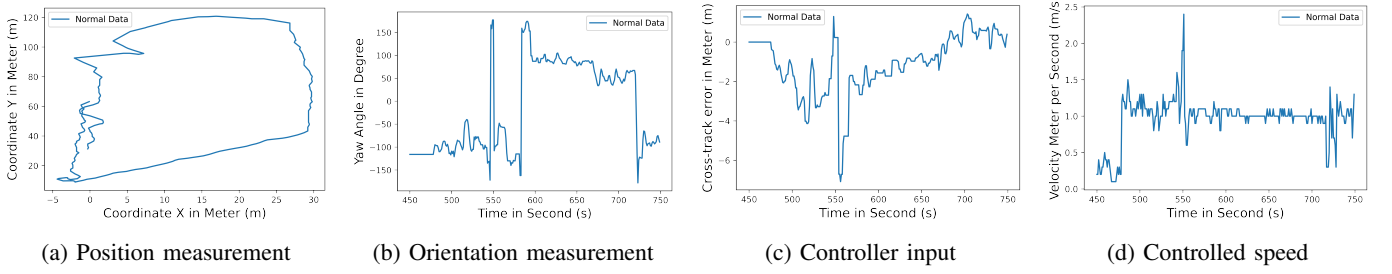

(a) Estimated Lateral Velocity vs Actual Velocity

(b) Estimated Yaw Rate vs Actual Yaw Rate

Fig. 6: Estimated Dynamics vs Actual Dynamics

(a) Position measurement   (b) Orientation measurement   (c) Controller input   (d) Controlled speed

Fig. 7: AVT pose measurements and controller input/ output without imposing GPS spoofing attack

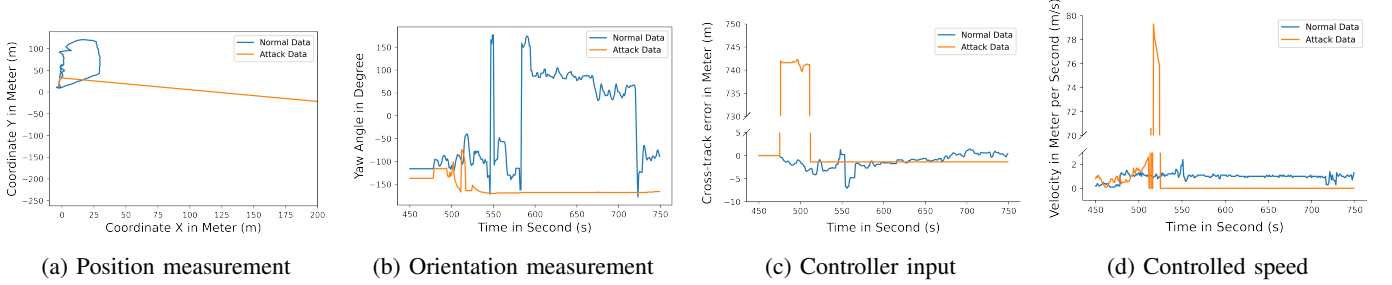(a) Position measurement   (b) Orientation measurement   (c) Controller input   (d) Controlled speed

Fig. 8: AVT pose measurements and controller input/ output under the influence of GPS spoofing attack. Axis breaks are utilized to shrink down large segments and enhance readability
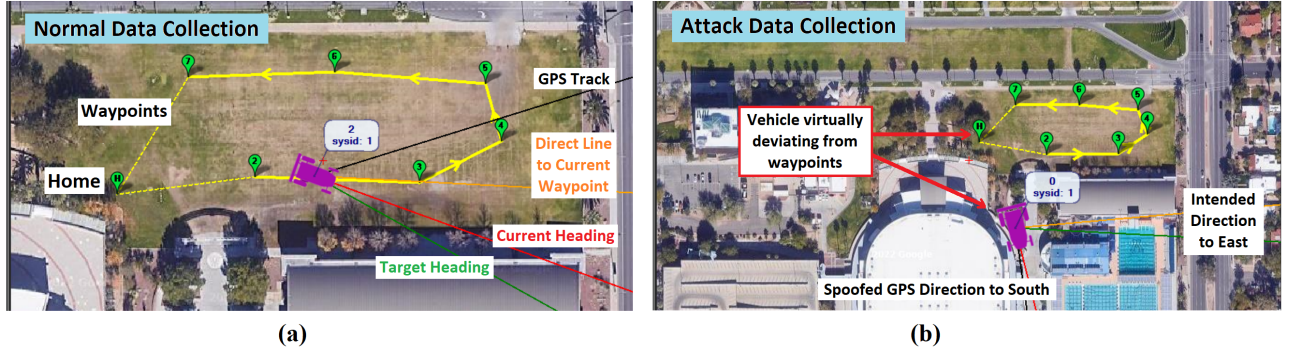

Fig. 9: Screenshot of the data collection procedure from the Ground Control Station computer. Fig. (a) AVT following given waypoints during normal data collection, Fig. (b) AVT virtually moving to spoofed GPS location during attack data collection

ment evaluated the GPS-IDS approach on the AV-GPS-Dataset using different machine learning models. To benchmark various machine learning techniques, including ensemble methods, neural networks, and tree-based algorithms, seven models were chosen for detection: Random Forest (RF), XGBoost (XGB), Support Vector Machine Classifier (SVC), Multi-Layer Perceptron (MLP), AdaBoost, Gradient Boosting (GB), and Decision Tree (DT).

Features associated with equations 15 (Vehicle Model block), 16 (Localization block), 17 (State Estimation block), 18 (Motion Planning block), and 20 (PID Controller block) from the Autonomous Vehicle Behavior Model (Fig. 4) are extracted and utilized in the machine learning models. To maintain the proportion of samples for each class consistently across both training and testing phases, stratified 5-fold sampling was employed to partition the data into training and testing sets. The training approach is based on supervised learning, where 4 folds (80%) were utilized for training, and

the remaining fold (20%) was utilized for testing. Three cases were considered to train the models, **Case I:** Train on 80% of AV-GPS-Dataset 1, and test on 20% of AV-GPS-Dataset 1, AV-GPS-Dataset 2 and AV-GPS-Dataset 3; **Case II:** Train on 80% of AV-GPS-Dataset 2, and test on AV-GPS-Dataset 1, 20% of AV-GPS-Dataset 2 and AV-GPS-Dataset 3; **Case III:** Train on 80% of AV-GPS-Dataset 3, and test on AV-GPS-Dataset 1, AV-GPS-Dataset 2 and 20% of AV-GPS-Dataset 3. For all three cases, the performance of the machine learning models is measured in terms of Accuracy, Precision, Recall, and F1-score. For each model, a combination of hyperparameters was utilized and fine-tuned to find the optimal performance. The performance of machine learning algorithms on different datasets is presented in Table II. The results indicate that the performance varies across different datasets as the training set changes. It can be observed that Case I exhibits high accuracy and F1 scores for most models, while the F1 scores drop for most models in Case III. This drop can be attributed to the

TABLE II: Performance of the Machine Learning Classification Models on AV-GPS-Datasets

	AV-GPS-Datasets	Metrics	RF	XGB	SVC	MLP	Adaboost	GB	DT
Case I Trained with 80% of Dataset 1	Test on 20% of Dataset 1	Accuracy	0.972	0.965	0.972	0.974	0.867	0.970	0.933
		Precision	0.994	0.969	0.985	0.979	0.671	0.962	0.810
		Recall	0.899	0.893	0.903	0.919	0.935	0.919	0.965
		F1 Score	0.944	0.929	0.942	0.948	0.782	0.939	0.881
	Test on Dataset 2	Accuracy	0.975	0.979	0.999	0.995	0.250	0.842	0.861
		Precision	0.912	0.933	0.999	0.984	0.248	0.610	0.640
		Recall	0.997	0.986	0.997	0.996	1	1	0.997
		F1 Score	0.953	0.959	0.998	0.990	0.397	0.758	0.780
	Test on Dataset 3	Accuracy	0.965	0.945	0.963	0.948	0.636	0.874	0.902
		Precision	0.997	1	0.997	0.967	0.636	0.839	0.883
		Recall	0.948	0.913	0.945	0.950	1	0.992	0.975
		F1 Score	0.972	0.954	0.970	0.958	0.778	0.909	0.927
	Average F1 Score in Case I		0.956	0.947	**0.970**	0.965	0.652	0.869	0.862
Case II Trained with 80% of Dataset 2	Test on Dataset 1	Accuracy	0.891	0.890	0.760	0.964	0.891	0.982	0.973
		Precision	0.992	0.994	0.980	0.996	0.992	0.978	0.997
		Recall	0.576	0.570	0.058	0.861	0.577	0.952	0.899
		F1 Score	0.729	0.725	0.109	0.924	0.730	0.965	0.946
	Test on 20% of Dataset 2	Accuracy	0.998	0.998	0.998	0.996	0.974	0.970	0.922
		Precision	0.998	0.998	0.994	0.998	0.998	0.961	0.926
		Recall	0.996	0.996	0.995	0.985	0.897	0.919	0.753
		F1 Score	0.997	0.997	0.997	0.992	0.945	0.939	0.830
	Test on Dataset 3	Accuracy	0.973	0.971	0.963	0.971	0.973	0.874	0.965
		Precision	1	1	0.997	0.997	0.997	0.839	1
		Recall	0.958	0.955	0.945	0.958	0.960	0.992	0.945
		F1 Score	0.978	0.977	0.970	0.977	0.978	0.909	0.972
	Average F1 Score in Case II		0.902	0.900	0.692	**0.964**	0.884	0.938	0.916
Case III Trained with 80% of Dataset 3	Test on Dataset 1	Accuracy	0.891	0.891	0.760	0.903	0.739	0.766	0.888
		Precision	0.983	0.982	0.939	0.904	0.487	0.536	0.982
		Recall	0.581	0.582	0.061	0.693	0.546	0.586	0.570
		F1 Score	0.730	0.731	0.115	0.784	0.515	0.560	0.721
	Test on Dataset 2	Accuracy	0.998	0.998	0.807	0.990	0.953	0.973	0.995
		Precision	0.995	0.993	0.979	0.961	0.842	0.903	0.985
		Recall	1	1	0.228	1	1	1	0.998
		F1 Score	0.997	0.996	0.369	0.980	0.914	0.949	0.991
	Test on 20% of Dataset 3	Accuracy	0.962	0.959	0.945	0.940	0.952	0.951	0.971
		Precision	0.984	0.965	0.967	0.946	0.963	0.953	1
		Recall	0.955	0.970	0.945	0.960	0.963	0.970	0.955
		F1 Score	0.969	0.968	0.956	0.953	0.963	0.962	0.977
	Average F1 Score in Case III		0.899	0.898	0.480	**0.905**	0.797	0.823	0.896
	Average F1 Score of all three cases		0.919	0.915	0.714	**0.944**	0.777	0.876	0.891

smaller size of the training set in Case III, which contains a limited number of entries representing only the transition of the attack. In this case, the test sets are significantly larger than the training set (AV-GPS-Dataset 1 is over 110 times larger, and AV-GPS-Dataset 2 is over 12 times larger), which declines the model performances. Overall, MLP consistently performs with F1 scores above 90% in all three cases, achieving the highest average F1 score of 94.4%. Following closely, RF and XGB demonstrated similar and second-best performances, achieving average F1 scores of 91.9% and 91.5%, respectively, in the three considered cases. The DT classifier secured the third-best performance, attaining an average F1 score of 89.1% across the three cases.

4) Experiment 4: Tuning of the Detection Threshold $\mathbb{T}$: In this experiment, we conducted an analysis to optimize the predefined detection threshold $\mathbb{T}$ of the GPS-IDS framework by evaluating the probability scores of normal data and attack data from AV-GPS-Dataset 3. The machine learning model that performed the best in Experiment 3, namely MLP, was employed for this purpose. The resulting data were plotted to determine a suitable detection margin, as illustrated in Fig. 10. Fig. 10 depicts the probability score distributions for normal and attack data obtained by applying Case I and Case II

TABLE III: False Positive (FP) and False Negative (FN) Rates for Different Detection Margins

Detection Margin	Normal Data Misclassified	Attack Data Misclassified	FP Rate	FN Rate
0.4 - 0.5	**1**	**1**	**0.0021**	**0.0012**
0.3 - 0.5	7	1	0.0146	0.0012
0.4 - 0.6	1	7	0.0021	0.0087
0.3 - 0.6	7	7	0.0146	0.0087

training methodologies on the same graph. By comparing the distributions, it becomes evident that the two kinds of data can be effectively distinguished with a significant margin indicated by the dotted bars. Table III shows the False Positive and False Negative Rates as we expand the margin. Based on the findings presented in Table III, we can see that selecting a detection threshold $\mathbb{T}$ within a range of **0.4 to 0.5** produces optimal outcomes. This range leads to the misclassification of only 1 instance of normal data and 1 instance of attack data, resulting in a False Positive Rate of 0.0021 or 0.21% and a False Negative Rate of 0.0012 or 0.12%. Thus, it can be deduced that opting for a detection threshold $\mathbb{T}$ in the range of 0.4 to 0.5 offers the lowest incidence of false detection and misclassification.
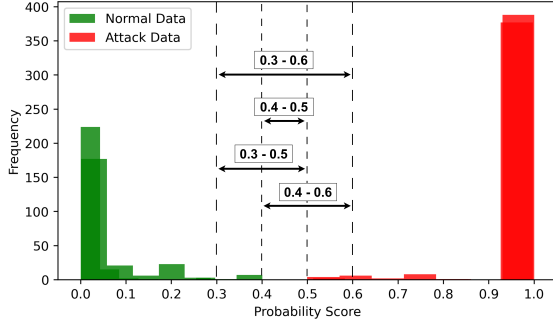
Fig. 10: Comparison of probability score distribution for normal and attack data
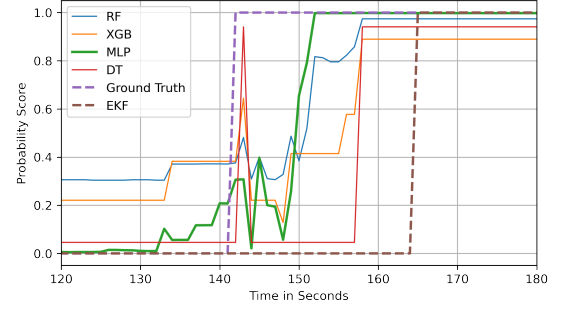
TABLE IV: Results of the Time Series Analysis

Attack Detection Time (in seconds)					
	MLP	XGB	RF	DT	EKF
Case I	10	16	16	16	23
Case II	10	10	10	16	23

5) Experiment 5: Time Series Analysis of the Machine Learning Models: In this experiment, we evaluated the effectiveness of the 4 best-performing models from Experiment 3 (MLP, RF, XGB, and DT) in detecting an attack with respect to time on AV-GPS-Dataset 3. The training phase involved training the models using Case I and Case II methodologies, followed by testing on AV-GPS-Dataset 3. The outcomes of the time series analysis are represented in Fig. 11. As depicted in Fig. 11, the attack was initiated in the 142th second and was subsequently detected by the AVT's EKF algorithm in the 165th second (23 seconds delay). In both training cases, all the models succeeded in detecting the attack before the EKF. In Fig. 11a, MLP was capable of detecting the attack at the 152nd second, followed by RF, XGB, and DT at the 158th second. Similarly, in Fig. 11b, the MLP, XGB, and RF identified the attack at the 152nd second, while the DT detected it at the 158th second. Table IV summarizes the attack detection times for each classifier, revealing that MLP achieved the fastest detection time of 10 seconds in both training methodologies.
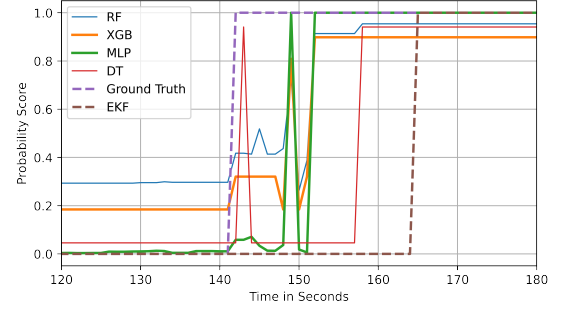
While a 10-second detection delay is considerably high in AV applications, it is important to highlight that the attack requires approximately 4-7 seconds to jam the reception of the authentic GPS signal and be effective on the AV. There are scopes for further improving the detection delay by fusing multiple detection techniques and increasing granularity.

V. CONCLUSION AND FUTURE WORK

This paper presents a novel anomaly behavior analysis-based GPS Intrusion Detection framework called the GPS-IDS that detects abnormal GPS navigation in AVs. The approach uses physics-based vehicle modeling to represent the behavior of the vehicle. A modified dynamic bicycle model is utilized to capture the normal behavior of the vehicle and machine learning techniques are used to detect the anomalous behavior. Moreover, a novel dataset family called the AV-GPS-Dataset with over 69,000 instances of real-world autonomous vehicle normal data and GPS spoofing attack data, is introduced

(a) Using Case I training methodology

(b) Using Case II training methodology

Fig. 11: Time series analysis on AV-GPS-Dataset 3

in this paper. The performance of the proposed GPS-IDS approach is evaluated using the AV-GPS-Dataset, and the experimental results affirm its high detection rate of GPS spoofing attacks. Additionally, it is validated that the proposed approach exhibits faster detection times in comparison to the EKF-based detection algorithm. In contrast to the existing segregated intrusion detection techniques that concentrate only on individual sensor data, our approach considers the overall behavior of the system, providing a more comprehensive approach to detect GPS attacks. We argue that using GPS-IDS in conjunction with sensor-specific intrusion detection systems can provide a powerful defense mechanism against GPS spoofing attacks in autonomous vehicles.

Moving forward, there are several promising research avenues to extend the contributions of this work. The GPS-IDS algorithm can be refined by incorporating additional operational parameters in the Autonomous Vehicle Behavior Model, such as integrating accurate modeling of multiple sensors and more robust control algorithms. Further research can be done to explore the effectiveness of integrating GPS-IDS with existing sensor-specific detection systems to enhance the overall security of autonomous vehicles. Additionally, the feasibility of real-world deployment scenarios, including cloud environment computing, onboard deployment on resource-constrained edge devices, or hybrid computation strategies that balance cloud and onboard processing can be investigated to validate the scalability and practicality of the GPS-IDS approach. Finally, the development of a digital twin for the autonomous vehicle behavior model emerges as a promising direction, providing a virtual environment for comprehensive testing and simulation.

REFERENCES

[1] SAE International. "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." SAE International 4970.724 (2018): 1-5.

[2] Singh, Santokh. "Critical reasons for crashes investigated in the national motor vehicle crash causation survey." No. DOT HS 812 115. 2015.

[3] National Highway Traffic Safety Administration. "Early Estimates of Motor Vehicle Traffic Fatalities And Fatality Rate by Sub-Categories in 2022" (2023): 1-9.

[4] Petrović, Dorde and Mijailović, Radomir and Pešić, Dalibor. "Traffic accidents with autonomous vehicles: type of collisions, maneuvers and errors of conventional vehicles' drivers." Transportation Research Procedia 45 (2020): 161-168.

[5] Van Brummelen, Jessica, et al. "Autonomous vehicle perception: The technology of today and tomorrow." Transportation Research Part C: Emerging Technologies 89 (2018): 384-406.

[6] Ma, Yifang, et al. "Artificial intelligence applications in the development of autonomous vehicles: A survey." IEEE/CAA Journal of Automatica Sinica 7.2 (2020): 315-329.

[7] Maurer, Markus, et al. Autonomous driving: technical, legal and social aspects. Springer Nature, 2016.

[8] Rajbahadur, Gopi Krishnan, et al. "A survey of anomaly detection for connected vehicle cybersecurity and safety." 2018 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2018.

[9] Griffin, John, John Batteh, and Johan Andreasson. "Modeling Vehicle Drivability with Modelica and the Vehicle Dynamics Library." Proceedings of the 9th International MODELICA Conference; September 3-5; 2012; Munich; Germany. No. 076. Linköping University Electronic Press, 2012.

[10] Whelan, Jason, et al. "Novelty-based intrusion detection of sensor attacks on unmanned aerial vehicles." Proceedings of the 16th ACM symposium on QoS and security for wireless and mobile networks. 2020.

[11] Whelan, Jason, et al. February 26, 2020, "UAV Attack Dataset", IEEE Dataport, doi: https://dx.doi.org/10.21227/00dg-0d12.

[12] Zarpelão, Bruno Bogaz, et al. "A survey of intrusion detection in Internet of Things." Journal of Network and Computer Applications 84 (2017): 25-37.

[13] Satam, Pratik, and Salim Hariri. "WIDS: An anomaly-based intrusion detection system for Wi-Fi (IEEE 802.11) protocol." IEEE Transactions on Network and Service Management 18.1 (2020): 1077-1091.

[14] Pacheco, Jesus, and Salim Hariri. "Anomaly behavior analysis for IoT sensors." Transactions on Emerging Telecommunications Technologies 29.4 (2018): e3188.

[15] Katriniok, Alexander, and Dirk Abel. "Adaptive EKF-based vehicle state estimation with online assessment of local observability." IEEE Transactions on Control Systems Technology 24.4 (2015): 1368-1381.

[16] Andreasson, Johan, Andreas Möller, and Martin Otter. "Modeling of a Racing Car with Modelica's Multi-Body Library." Workshop Proceedings. 2000.

[17] Bhadani, Rahul Kumar, Jonathan Sprinkle, and Matthew Bunting. "The cat vehicle testbed: A simulator with hardware in the loop for autonomous vehicle applications." arXiv preprint arXiv:1804.04347 (2018).

[18] Pan, Yongjun, et al. "Data-driven vehicle modeling of longitudinal dynamics based on a multibody model and deep neural networks." Measurement 180 (2021): 109541.

[19] Fényes, Dániel, Németh, Balázs and Gáspár, Péter. "A novel data-driven modeling and control design method for autonomous vehicles." Energies 14.2 (2021): 517.

[20] James, Sebastian S., Sean R. Anderson, and Mauro Da Lio. "Longitudinal vehicle dynamics: A comparison of physical and data-driven models under large-scale real-world driving conditions." Ieee Access 8 (2020): 73714-73729.

[21] Psiaki, Mark L., Todd E. Humphreys, and Brian Stauffer. "Attackers can spoof navigation signals without our knowledge. Here's how to fight back GPS lies." IEEE Spectrum 53.8 (2016): 26-53.

[22] Liang, Chen, et al. "Detection of GPS spoofing attack on unmanned aerial vehicle system." Machine Learning for Cyber Security: Second International Conference, ML4CS 2019, Xi'an, China, September 19-21, 2019, Proceedings 2. Springer International Publishing, 2019.

[23] He, Liang, et al. "Civilian unmanned aerial vehicle vulnerability to gps spoofing attacks." 2014 Seventh International Symposium on Computational Intelligence and Design. Vol. 2. IEEE, 2014.

[24] Yang, Zhen, et al. "Anomaly Detection Against GPS Spoofing Attacks on Connected and Autonomous Vehicles Using Learning From Demonstration." IEEE Transactions on Intelligent Transportation Systems (2023).

[25] Milaat, Fahad Ali, and Hang Liu. "Decentralized detection of GPS spoofing in vehicular ad hoc networks." IEEE Communications Letters 22.6 (2018): 1256-1259.

[26] Oligeri, Gabriele, et al. "GPS spoofing detection via crowd-sourced information for connected vehicles." Computer Networks 216 (2022): 109230.

[27] Alipour, Hamid, et al. "Wireless anomaly detection based on IEEE 802.11 behavior analysis." IEEE transactions on information forensics and security 10.10 (2015): 2158-2170.

[28] Pendleton, Scott Drew, et al. "Perception, planning, control, and coordination for autonomous vehicles." Machines 5.1 (2017): 6.

[29] Rajamani, Rajesh. Vehicle dynamics and control. Springer Science & Business Media, 2011.

[30] Kong, Jason, et al. "Kinematic and dynamic vehicle models for autonomous driving control design." 2015 IEEE intelligent vehicles symposium (IV). IEEE, 2015.

[31] Pepy, Romain, Alain Lambert, and Hugues Mounier. "Path planning using a dynamic vehicle model." 2006 2nd International Conference on Information & Communication Technologies. Vol. 1. IEEE, 2006.

[32] Mehrab Abrar, Murad, Raian Islam, and Md Abid Hasan Shanto. "An autonomous delivery robot to prevent the spread of coronavirus in product delivery system." 2020 11th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON). IEEE, 2020.

[33] Psiaki, Mark L., and Todd E. Humphreys. "GNSS spoofing and detection." Proceedings of the IEEE 104.6 (2016): 1258-1270.

[34] AV-GPS-Dataset: Autonomous Vehicle Global Positioning System Dataset. [Online]. Available: https://github.com/mehrab-abrar/AV-GPS-Dataset/

[35] Urlichich, Yuri, et al. "GLONASS modernization." Proceedings of the 24th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2011). 2011.

[36] Han, Chunhao, Yuanxi Yang, and Zhiwu Cai. "BeiDou navigation satellite system and its time scales." Metrologia 48.4 (2011): S213.

[37] Benedicto, J., et al. "GALILEO: Satellite system design." European Space Agency. Int. Business, 2000.

[38] Tippenhauer, Nils Ole, et al. "On the requirements for successful GPS spoofing attacks." Proceedings of the 18th ACM conference on Computer and communications security. 2011.

[39] Wang, Shenqing, et al. "Intelligent detection algorithm against uavs' gps spoofing attack." 2020 IEEE 26th International Conference on Parallel and Distributed Systems (ICPADS). IEEE, 2020.

[40] Humphreys, Todd E., et al. "Assessing the spoofing threat: Development of a portable GPS civilian spoofer." Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008). 2008.

[41] Shen, Junjie, et al. "Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under GPS spoofing." Proceedings of the 29th USENIX Conference on Security Symposium. 2020.

[42] Inside GNSS. Tesla Model S and Model 3 Prove Vulnerable to GPS Spoofing Attacks, Research from Regulus Cyber Shows. Accessed: Apr. 20, 2023. [Online]. Available: https://insidegnss.com/tesla-models-and-model-3-prove-vulnerable-to-gps-spoofing-attacks-research-fromregulus-cyber-shows/

[43] Haddad, Ranwa, et al. "GPS modernization and beyond." 2020 IEEE/ION Position, Location and Navigation Symposium (PLANS). IEEE, 2020.

[44] Neri, Alessandro, et al. "An anti-jamming and anti-spoofing digital beamforming platform for the GNSS-based ERTMS train control system." Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017). 2017.

[45] Xu, Rui, et al. "Performance analysis of GNSS/INS loosely coupled integration systems under spoofing attacks." Sensors 18.12 (2018): 4108.

[46] Ranyal, Eshta, and Kamal Jain. "Unmanned aerial vehicle's vulnerability to gps spoofing a review." Journal of the Indian Society of Remote Sensing 49 (2021): 585-591.

[47] Liu, Yin-Chen, Gianluca Bianchin, and Fabio Pasqualetti. "Secure trajectory planning against undetectable spoofing attacks." Automatica 112 (2020): 108655.

[48] Zhang, Dong, et al. "Cyber-attack detection for autonomous driving using vehicle dynamic state estimation." Automotive Innovation 4 (2021): 262-273.

[49] Cui, Jin, and Biao Zhang. "VeRA: A simplified security risk analysis method for autonomous vehicles." IEEE Transactions on Vehicular Technology 69.10 (2020): 10494-10505.

[50] Feng, Shuo, and Simon Haykin. "Cognitive risk control for anti-jamming V2V communications in autonomous vehicle networks." IEEE Transactions on Vehicular Technology 68.10 (2019): 9920-9934.

[51] Nguyen, Van-Linh, Po-Ching Lin, and Ren-Hung Hwang. "Enhancing misbehavior detection in 5G vehicle-to-vehicle communications." IEEE Transactions on Vehicular Technology 69.9 (2020): 9417-9430.

[52] Nowdehi, Nasser, et al. "CASAD: CAN-aware stealthy-attack detection for in-vehicle networks." arXiv preprint arXiv:1909.08407 (2019).

[53] Reina, Giulio, and Arcangelo Messina. "Vehicle dynamics estimation via augmented Extended Kalman Filtering." Measurement 133 (2019): 383-395.

[54] Xu, Wenyuan, et al. "Analyzing and enhancing the security of ultrasonic sensors for autonomous vehicles." IEEE Internet of Things Journal 5.6 (2018): 5015-5029.

[55] Liu, Qipeng, et al. "Secure pose estimation for autonomous vehicles under cyber attacks." 2019 IEEE Intelligent Vehicles Symposium (IV). IEEE, 2019.

[56] Hoppe, Tobias, Stefan Kiltz, and Jana Dittmann. "Security threats to automotive CAN networks–practical examples and selected short-term countermeasures." Computer Safety, Reliability, and Security: 27th International Conference, SAFECOMP 2008 Newcastle upon Tyne, UK, September 22-25, 2008 Proceedings 27. Springer Berlin Heidelberg, 2008.

[57] Miller, Charlie, and Chris Valasek. "A survey of remote automotive attack surfaces." black hat USA 2014 (2014): 94.

[58] Miller, Charlie. "Lessons learned from hacking a car." IEEE Design & Test 36.6 (2019): 7-9.

[59] Koscher, Karl, et al. "Experimental security analysis of a modern automobile." 2010 IEEE symposium on security and privacy. IEEE, 2010.

[60] Checkoway, Stephen, et al. "Comprehensive experimental analyses of automotive attack surfaces." USENIX security symposium. Vol. 4. No. 447-462. 2011.

[61] Miller, Charlie, and Chris Valasek. "Remote exploitation of an unaltered passenger vehicle." Black Hat USA 2015.S 91 (2015): 1-91.

[62] Leung, King Tin, et al. "Road vehicle state estimation using low-cost GPS/INS." Mechanical Systems and Signal Processing 25.6 (2011): 1988-2004.

[63] Egerstedt, M., Xiaoming Hu, and A. Stotsky. "Control of a car-like robot using a virtual vehicle approach." Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No. 98CH36171). Vol. 2. IEEE, 1998.

[64] Ardupilot Official Website. [Online]. Accessed: June 2022. Available: https://ardupilot.org/

[65] Manesh, Mohsen Riahi, et al. "Detection of GPS spoofing attacks on unmanned aerial systems." 2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC). IEEE, 2019..

[66] Bae, Gimin, and Inwhee Joe. "UAV anomaly detection with distributed artificial intelligence based on LSTM-AE and AE." Advanced Multimedia and Ubiquitous Engineering: MUE/FutureTech 2019 13. Springer Singapore, 2020.

[67] Agyapong, Richmond Asiedu, et al. "Efficient detection of GPS spoofing attacks on unmanned aerial vehicles using deep learning." 2021 IEEE Symposium Series on Computational Intelligence (SSCI). IEEE, 2021.

[68] Panice, G., et al. "A SVM-based detection approach for GPS spoofing attacks to UAV." 2017 23rd International Conference on Automation and Computing (ICAC). IEEE, 2017.

[69] Shafique, Arslan, Abid Mehmood, and Mourad Elhadef. "Detecting signal spoofing attack in uavs using machine learning models." IEEE access 9 (2021): 93803-93815.

[70] Satam, Shalaka. Autonomous Vehicle Security Framework (AVSF). Diss. The University of Arizona, 2022.

Raian Islam is a Master's student in the Department of Electrical and Computer Engineering at the University of Arizona, Tucson, AZ, USA. She received her B.Sc. degree in Electrical and Electronic Engineering from Ahsanullah University of Science and Technology, Dhaka, Bangladesh, in 2019. Her current research interests include data analytics, photovoltaics, and machine learning.

Shalaka Satam received her B.E. degree in Electronics and Telecommunications Engineering from the University of Mumbai in 2015. She received the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Arizona in 2017 and 2022, respectively. Her research focuses on cybersecurity, autonomous vehicles, and Internet of Things.

Sicong Shao is an assistant professor of the School of Electrical Engineering and Computer Science at the University of North Dakota (UND). Before joining UND, he was a research assistant professor in the Department of Electrical and Computer Engineering (ECE) at the University of Arizona where he also received his Ph.D. in ECE. His research bridges the areas of cybersecurity, artificial intelligence (AI), machine learning (ML), and software engineering with a research theme that has focused on applying AI/ML to overcome security challenges in system and software security, network security, digital forensics, and social-cybersecurity.

Salim Hariri (Senior Member, IEEE) received an M.Sc. degree from Ohio State University in 1982, and a Ph.D. degree in Computer Engineering from the University of Southern California in 1986. He is a Professor in the Department of Electrical and Computer Engineering, the University of Arizona, and the Director of the NSF Center for Cloud and Autonomic Computing (NSF-CAC). His research focuses on autonomic computing, cybersecurity, cyber resilience, and cloud security.

Pratik Satam is an Assistant Professor in the Department of Systems and Industrial Engineering at the University of Arizona, Tucson, AZ, USA. He is also part of the new Software Engineering program in College of Engineering at the University of Arizona. He received his B.E. degree in Electronics and Telecommunication Engineering from the University of Mumbai, in 2013, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Arizona in 2015 and 2019, respectively. From 2019 to 2022, he has been a Research Assistant Professor at the Department of Electrical and Computer Engineering, the University of Arizona. His current research interests include autonomic computing, cyber security, cyber resilience, secure critical infrastructures, and cloud security. He is an Associate Editor for the scientific journal Cluster Computing.

Murad Mehrab Abrar is currently a Master's student in the Department of Electrical and Computer Engineering, the University of Arizona, Tucson, AZ, USA. He completed his B.Sc. in Electrical and Electronic Engineering from Ahsanullah University of Science and Technology, Dhaka, Bangladesh in 2019. His research interest focuses on robotics, autonomous vehicles, and machine learning.