Anomaly Detection Against GPS Spoofing Attacks on Connected and Autonomous Vehicles Using Learning From Demonstration

Zhen Yang, Jun Ying, Junjie Shen, Yiheng Feng, Qi Alfred Chen, *Member, IEEE*, Z. Morley Mao, *Fellow, IEEE*, and Henry X. Liu, *Member, IEEE*

***Abstract*— GPS spoofing attacks pose great challenges to connected vehicle (CVs) safety applications and localization of autonomous vehicles (AVs). In this paper, we propose to utilize transportation and vehicle engineering domain knowledge to detect GPS spoofing attacks towards CVs and AVs. A novel detection method using learning from demonstration is developed, which can be implemented in both vehicles and at the transportation infrastructure. A computational-efficient driving model, which can be learned from historical trajectories of the vehicles, is constructed to predict normal driving behaviors. Then a statistical method is developed to measure the dissimilarities between the observed trajectory and the predicted normal trajectory for anomaly detection. We validate the proposed method using two threat models (i.e., attacks targeting the multi-sensor fusion system of AVs and attacks targeting the intersection movement assist application of CVs) on two real-world datasets (i.e., KAIST and Michigan roundabout dataset). Results show that the proposed model is able to detect almost all of the attacks in time with low false positive and false negative rates.**

***Index Terms*— Anomaly detection, GPS spoofing attack, localization, intersection movement assist, connected and autonomous vehicles, learning from demonstration.**

I. Introduction

CONNECTED vehicles (CVs) and autonomous vehicles (AVs) benefit the transportation system from multiple aspects including reducing crashes, improving mobility and sustainability. In both types of vehicles, the localization module, from which the vehicle knows its global and local positions in the driving environment, plays a critical role in information sharing and vehicle navigation. For example, the Basic Safety Messages (BSMs) broadcast by CVs contain vehicle location and motion data for a wide range of applications [1], [2], [3], and AVs utilize the localization results for trajectory planning [4]. Among all sensors that participate in localization, the GPS receiver is the most important one that obtains global positions. Commercial-level GPS receivers can achieve an accuracy of 1 meter, and with dual-frequency GPS units, survey-grade GPS has an accuracy of a few centimeters [5]. Besides GPS, LiDAR locators and Inertial Measurement Units (IMU) are also implemented and tested on AVs [6], [7], [8] for localization purposes. It is critical to ensure that the localization module is accurate, reliable, and highly secure since inaccurate localization results will significantly jeopardize AV trajectory planning and CV safety applications and may cause catastrophic consequences such as crashes.

Unfortunately, existing studies show that vehicle localization module is vulnerable to various types of cyberattacks. Spoofing attack is an emerging issue in modern GPS applications. The GPS spoofing attack generates fabricated GPS signals and interferes with the GPS receivers, which can degrade the performance of the localization system. The fake GPS signal usually has a higher strength to mislead the GPS receiver [9]. The practicality of the GPS spoofing attack has been proved in both research [10], [11] and real-world applications [12], [13]. In addition to the GPS spoofing attack, attacks targeting other sensors can also impact vehicle localization. For example, Petit et al. attacked the LiDAR by injecting false reflected light, and the LiDAR falsely detected a fake wall [14]. Although such LiDAR sensor attacks do not directly target the localization module, misinterpretation of the surrounding environment will also degrade the performance of the LiDAR locator, which is an important input source to the localization module. Usually, Multi-Sensor Fusion (MSF) algorithms are considered as one defense method against sensor attacks since it is highly unlikely that all sensors are compromised at the same time. However, a recent study from Shen et al. managed to construct an MSF attack method, which misleads the sensor fusion algorithms by only spoofing the GPS channel [11].

In general, anomaly detection is applied to defend against GPS spoofing attacks, which can be divided into two categories, node centric detection and data centric detection [15]. In the **node centric detection**, it examines the patterns in the behavior of specific nodes at the protocol level, which usually

Manuscript received 4 April 2022; revised 27 November 2022 and 10 April 2023; accepted 13 April 2023. This work was supported in part by the U.S. National Science Foundation through Grant CNS-1850533, Grant CNS-1930041, Grant CNS-1929771, and Grant CNS-2145493; and in part by the United States Department of Transportation (USDOT) University Transportation Center (UTC) under Grant 69A3552047138. The Associate Editor for this article was L. Wang. *(Corresponding author: Yiheng Feng.)*

Zhen Yang and Henry X. Liu are with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor, MI 48104 USA.

Jun Ying and Yiheng Feng are with the Lyles School of Civil Engineering, Purdue University, West Lafayette, IN 47907 USA (e-mail: feng333@purdue.edu).

Junjie Shen and Qi Alfred Chen are with the Department of Computer Science, University of California at Irvine, Irvine, CA 92697 USA.

Z. Morley Mao is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI 48104 USA.

Digital Object Identifier 10.1109/TITS.2023.3269029

does not consider data semantics. Signatures are adopted to identify if the sender of the messages is malicious. The node centric detection can be further classified as behavioral or trust-based. The behavioral mechanism checks the packet header and metamessage information to detect the anomaly. Common behavioral mechanisms include watchdog [16] and flooding detection [17]. Trust-based mechanisms aggregate the trust of a node and distribute the trust among nodes to filter the malicious nodes. Trust-based mechanisms are usually vulnerable to Sybil attacks.

Different from node centric mechanisms, **data centric detection** mainly focus on data semantics, which can also be categorized into two groups, consistency-based and plausibility-based [15]. The consistency-based method examines the relations between packets to identify the anomaly of the newly received data. A cooperative approach can be adopted to analyze the information from multiple agents to identify conflicting messages [18], [19].

Plausibility-based methods filter out the packets according to the numerical plausibility value contained by the data received, which can be utilized to detect attacks with Sybil nodes. A majority of the plausibility-based detection focus on signal-based method [20], [21], [22]. The main shortage of signal-based methods is generalizability. For example, the method proposed for the anomaly detection of a GPS-only localization system may not be suitable for an MSF localization system. Another plausibility-based detection method is prediction-based. The prediction-based methods focus on predicting the behavior of vehicles and comparing the prediction with the observations. Kalman filter based approach is the most common method in this direction [23], [24], [25], in which the future trajectory of the vehicle is predicted with a Kalman filter. Other than only predicting the positions of vehicles, vehicle dynamics can be also integrated into the prediction-based mechanisms. For example, in [26], vehicle dynamics are considered to predict the bounding boxes of vehicles. The prediction-based methods can be viewed as driving model-based methods that make predictions of the vehicles to detect the anomaly. Such driving model-based methods may not be generalized to different driving scenarios (e.g. highway / urban).

Recent studies applied learning-based approaches to detect GPS spoofing attacks. Dasgupta et al. [27] implemented an LSTM neural network to predict AV's travel distance between two consecutive timestamps. Spoofing attack detection was implemented based on the difference between the perceived location shift and predicted location shift. Jiang et al. [28] applied deep neural networks to estimate vehicle speed and direction sequence and infer vehicle positions. A dynamic Time Wrapping algorithm was applied to measure the similarity between the reported and predicted trajectories. Note that both methods need thousands of vehicle trajectories in the training process, which may take a very long time to collect.

In this paper, a new prediction based method is proposed, which combines model-driven and data-driven approaches for GPS spoofing attack detection, and is proved to have better generalizability in the experiments. The central hypothesis is

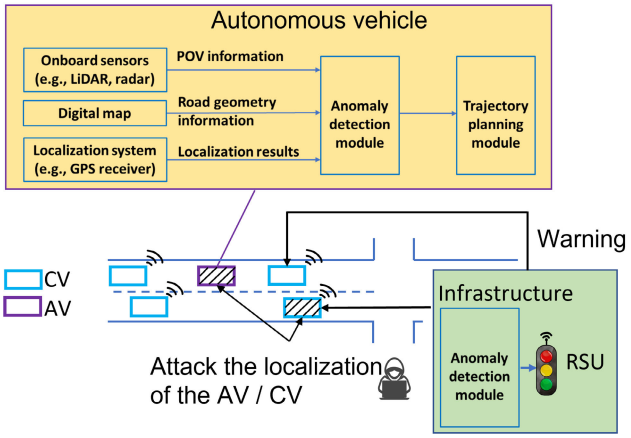


Fig. 1. Concept of abnormal trajectory detection.

that if the data in the GPS signal is compromised, the resultant information sharing from CVs or trajectory planning from AVs will be impacted, which generates abnormal driving behaviors (i.e., abnormal vehicle trajectories). Following this direction, transportation and vehicle domain knowledge is applied with the learning from demonstration framework. This method can be deployed in both vehicles and transportation infrastructure.

Figure 1 illustrates the concept of the proposed anomaly detection method. For the AV deployment, illustrated by the yellow block, the anomaly detection module is located before the trajectory planning module. Three types of information are used as the input to the detection module. First, information of the AV's principle other vehicles (POVs) captured by onboard sensors. The POVs are defined as nearby vehicles that may influence the behaviors of the AV (e.g., a leading vehicle in the same lane). Second, a digital map that contains roadway geometry information. Third, the localization results provided by the localization module. In the anomaly detection module, the normal driving behavior of the AV is represented by a computational-efficient driving model, which can be learned from the historical trajectories of the AV. The normal driving behavior is then compared with the trajectories from the localization module to detect the anomaly.

For infrastructure deployment, a CV environment is assumed. In figure 1, the traffic scenario below the yellow block illustrates the anomaly detection concept at the infrastructure side. CVs broadcast their localization results in the form of BSMs. The infrastructure is equipped with Roadside Unit (RSU) to collect BSMs from the CVs and learns normal driving behaviors. When a CV is under a GPS spoofing attack, it broadcasts BSMs with falsified data elements such as location and speed. The infrastructure compares the learned normal driving behavior and the received CV trajectory to detect the anomaly and send warnings to the victim CV and nearby vehicles. Notice that in this case, we assume that the infrastructure does not have other sensors (e.g., cameras) to cross validate the integrity of the communication messages.

The most important component in the proposed anomaly detection framework is learning normal driving behaviors. Toward this end, we apply the learning from demonstration

framework, in which an agent can learn expert behaviors with demonstrations (i.e., examples). The demonstrations are state-action pairs collected from a teacher when he/she performs certain tasks. In this work, learning from demonstration is implemented to learn a computational-efficient CV/AV driving model in different driving scenarios. After collecting a sufficient number of historical trajectories as the demonstrations, maximum entropy inverse reinforcement learning is adopted to derive the optimal driving policy (i.e., reward function). The learned driving policy is used to generate a predicted optimal trajectory, which is then compared with the observed trajectory to identify whether the observed trajectory is under attack or not. A statistical method is developed to measure the dissimilarities between the observed trajectory and the predicted optimal trajectory. With appropriate features that capture such dissimilarities, a decision-tree classifier is adopted to differentiate normal trajectories and trajectories under attack.

The proposed detection method is evaluated with two threat models. The first threat model aims at attacking the Multi-Sensor Fusion (MSF) based localization model of an AV. The goal of the attack is to generate lateral deviations to the original trajectory to make the subject AV hit the road curb or drive in the wrong direction. The second threat model aims at attacking the intersection movement assist (IMA) application on CVs. Experiments are conducted on two real-world datasets, KAIST [29] and Michigan roundabout datasets [30]. Experiment results show that the proposed model has a good performance in both offline detection and online detection with low false positive and false negative rates. Further adaptive attack study confirms the robustness of the model in detecting more stealthy attacks with reduced magnitude.

Contributions of this work are three-fold:

1. **We propose an innovative detection framework to detect anomalies in the localization module of CV/AV using learning from demonstration.** The proposed detection framework directly examines the outputs of the localization module, regardless of the mechanism of the localization module and different attack types. Moreover, the learning from demonstration framework requires much fewer data in the training process but still achieves very high accuracy.

2. **The proposed method integrates domain knowledge in detecting cyberattacks.** The methodology leverages transportation and vehicle domain knowledge to learn the driving policy through real-world demonstrations. Compared with other learning based approaches, our proposed framework requires much fewer data in the training stage. To our knowledge, this is the first paper that utilizes learning from demonstration with domain specific knowledge for abnormal trajectory detection.

3. **The proposed detection method has low requirements for implementation.** For AV deployment, the anomaly detection requires onboard sensors and digital map information. Such onboard sensors and digital map information are standard for all the AV configurations [9], [31]. For infrastructure side deployment, no other infrastructure sensors are needed. The required connected vehicle environment has been implemented and tested extensively in the past few years [32], [33].

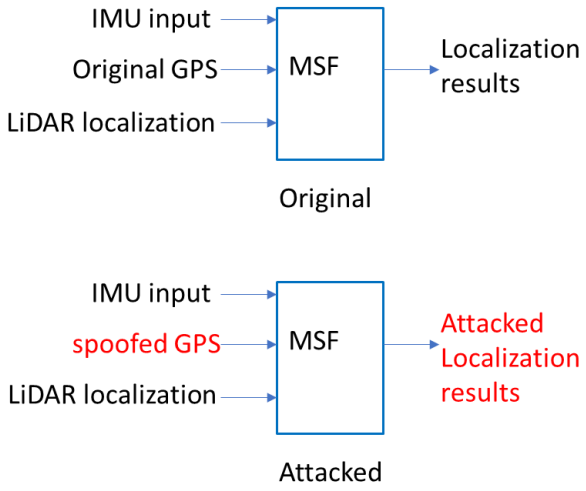


Fig. 2. Threat model on the AV MSF System.

The rest of the paper is organized as follows: we first present the threat models (Section II). In section III, the methodology of the anomaly detection model is introduced. In section IV and V, the proposed model is validated on the AV threat model and CV threat model, respectively. Section VI extends the experiments on the AV threat model to adaptive attacks. Section VII concludes the paper and lays out future research directions.

II. Threat Model

A. Autonomous Vehicle Threat Model

A real-world Multi-Sensor Fusion (MSF) attack conducted on the Baidu Apollo system is applied as the threat model for the AV anomaly detection [11]. In this study, a GPS spoofing attack towards the MSF-based localization system of AVs is designed, shown in Figure 2. At the top, the original MSF algorithm takes the input from GPS, IMU, and LiDAR to generate localization results. In the attack scenario, the GPS channel is spoofed by the FusionRipper algorithm proposed in [11], which can successfully mislead the MSF localization algorithm. The practicality of the GPS spoofing attack is also justified with existing literature in [11].

The FusionRipper algorithm consists of two phases: vulnerability profiling and aggressive spoofing. In the vulnerability profiling phase, the attacker performs a constant GPS spoofing attack and observes the localization results from the MSF system to profile when the vulnerable periods appear (i.e., lateral deviation $\geq$ 0.295 m on urban roads). After the vulnerable period is identified, the aggressive spoofing phase starts in which the attacker performs exponentially aggressive spoofing to quickly induce large lateral deviations. Two attack goals that cause safety hazards are considered, off-road (i.e., hitting road curbs) attack and wrong-way (i.e., driving to the opposite direction of the road) attack, and both attack goals are achieved by large lateral deviations. The off-road attack requires less lateral deviation (0.895m for urban roads) in the localization results to succeed than the wrong-way attack (1.945m for urban roads).

Note that camera-based lane detection [34] is used as the main technology for lane keep in modern cars today. However, such a technology is used only for low-level driving assistance (e.g., Level-2) for local localization (i.e., positioning within the current lane boundaries), so its existence is orthogonal to our AV threat model that is targeted at high-level autonomous driving system (i.e., Level 3 or higher) that requires global localization (i.e., positioning on a city map). Such high-level autonomous driving systems typically use multi-sensor fusion-based global localization (e.g., combining GPS, LiDAR and IMU), but this does not affect the validity of our attack threat model at all since the attack vector we employ is exactly designed to target (and has shown to be effective against) the latest fusion-based localization in high-level autonomous driving using GPS spoofing alone [11].

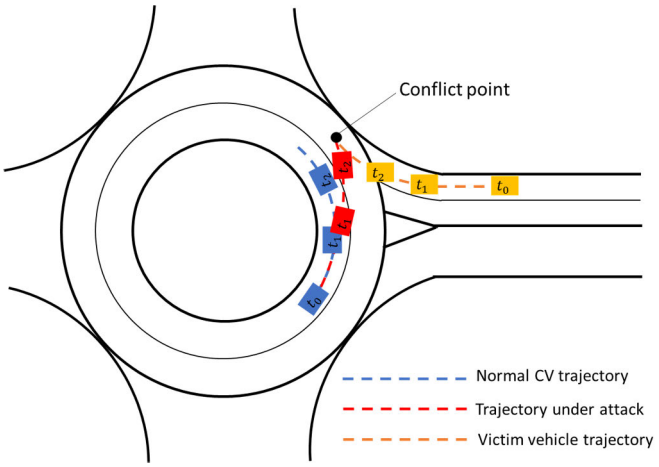


Fig. 3. Threat model on intersection movement assist system.

B. Connected Vehicle Threat Model

In the CV environment, we choose a threat model towards the IMA system, which is an important CV safety application [35], [36]. The IMA application intends to warn the driver of the CV when it is unsafe to enter an unsignalized intersection in case of high collision probability with other vehicles. The application uses data (i.e., BSMs) received from other vehicles to determine if it is unsafe to enter the intersection. In this work, it is assumed that the IMA system only relies on the received CV messages (BSM) to trigger the warning [36]. In our study, we use a roundabout scenario to demonstrate the threat model.

The goal of the attack is to generate falsified BSMs through GPS spoofing to trigger the IMA warning of a CV. Figure 3 illustrates the concept of the attack. It is assumed that BSMs have lane level accuracy, which has been validated in previous studies [37]. In the figure, the blue rectangles denote the normal CV trajectory (i.e., true locations), which is located at the inner lane of the roundabout. The red rectangles represent the BSM trajectory under attack which changes lanes from the inner lane to the outer lane. The yellow rectangles denote the trajectory of the victim CV, located at the entry of the roundabout. A conflict point is defined as the intersection along the trajectories of the victim vehicle and the CV under attack. Given that lane changing behavior is not allowed in the roundabout, if the blue CV is not under attack, its trajectory should not conflict with the victim CV. The IMA warning should not be triggered at the victim CV. When the blue CV is under attack (becomes the red CV), the falsified lane change trajectory will trigger the IMA warning. The values within the rectangles denote timestamps, where at time t_0, the attack starts. At time t_2, the falsified BSM trajectory triggers the IMA warning of the victim (yellow) CV.

To generate feasible attack trajectories, the attack model is formulated as an optimization problem similar as in [38] and [39]. The objective function contains two parts: 1) trigger the IMA warning of the victim CV; and 2) generate a smooth trajectory that is close to the real driving behavior. To achieve the first goal, the arrival time of the victim vehicle to the conflict point needs to be predicted. It is assumed that victim vehicle follows a constant speed to reach the conflict point. Based on the predicted arrival time, the first part of the objective function generates a lane change trajectory to reach the conflict point close to the arrival time of the victim CV to trigger the IMA warning. The second part of the objective function contains driving features such as minimizing acceleration and minimizing heading change rate.

The attack starts when the estimated arrival time to the conflict point between the vehicle under attack and the victim vehicle is less than 4s. The planning horizon for the attack trajectory is the same as the estimated arrival time for the victim vehicle to reach the conflict point. A rolling horizon scheme is applied, where the prediction is repeated every 0.4 seconds to minimize the prediction error. The attack trajectory is generated for the whole planning horizon, but only the first 0.4s will be executed. Assuming that the vehicle keeps a constant speed to reach the conflict point, Equation 1 denotes that the attack succeeds when the post encroachment time (PET) to the conflict point between the victim vehicle and the attack trajectory is less than T_g. D_{atk} denotes the distance to the conflict point. v_{atk} denotes the attack vehicle speed and T_g denotes the critical PET that triggers the IMA warning. In this work, $T_g = 2s$, which is consistent with existing studies ref [36].

$$D_{atk}/v_{atk} \leq T_g \tag{1}$$

III. Defense Methodology

This section presents the anomaly detection method to identify the AV/CV attacks introduced in the previous section. There are two major challenges. **Challenge 1: real-time detection.** The operation of AV/CV is highly safety-critical. Therefore, it is vital to detect abnormal or hazardous driving behaviors in time. However, some GPS spoofing attacks can be stealthy (e.g., the first phase of the AV threat model in Section II-A) while some can achieve the attack goal in a few seconds (e.g., the CV threat model in Section II-B), which all pose great challenges in the detection model design. **Challenge 2: validity on different threat models.** Different threat models may cause different abnormal driving behaviors. It would be more meaningful if the anomaly detection method is effective under different types of attacks.

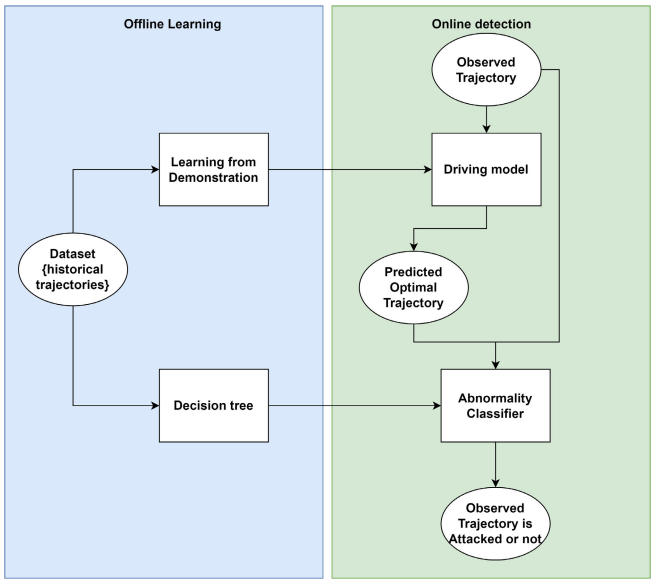


Fig. 4. Anomaly detection framework.

A. Defense Framework

Figure 4 illustrates the anomaly detection framework consisting of two steps. On the left side (i.e., offline learning), learning from demonstration is adopted to learn the driving model via maximum entropy inverse reinforcement learning, using historical trajectories. Besides, a decision tree is trained with both historical trajectories and known attack trajectories by three features (objective ratio, normality score, and trajectory displacement). The trained models are applied in the online detection step as shown on the right side of the figure. When observing a trajectory from the localization module or from the CV, its initial state and environment state are utilized in the learned driving model to generate a predicted optimal trajectory, which is then compared with the observed trajectory in terms of the three features. The results are fed into the trained decision tree classifier, which will finally decide whether the vehicle is under attack or not.

B. Learning From Demonstration

A general trajectory generation problem can be formulated as an optimization problem, shown in Equation 2. The objective of the optimization problem is the utility function of the driving behavior, in which θ is the weight vector associated with different driving utilities. $\mathbf{s}$ is the decision variable of the optimization problem, which denotes the trajectory, a vector of trajectory points s_i. Each trajectory point s_i at time step i can be represented by $(x_i, y_i, v_i, a_i, \psi_i)$, in which x_i and y_i is the longitudinal and lateral coordinate, respectively, and ψ_i is the heading angle of the vehicle, between the longitudinal axis of the vehicle and the longitudinal direction of the road. v_i denotes the speed of the vehicle, and a_i denotes the acceleration. $\mathbf{u}$ represents the initial condition and environment states, which serve as the input parameters for the optimization problem. The initial condition includes the initial position (x_0, y_0), initial speed v_0, initial acceleration a_0, and initial heading angle ψ_0. The environment states include the longitudinal coordinate and the lateral coordinate of the leading vehicle. $f(\mathbf{s}, \mathbf{u})$ is a mapping function that maps the trajectory to a feature vector, which can be different under different maneuvers. The details of the features and vehicle dynamic constraints are introduced next.

$$\begin{aligned} \min_{\mathbf{s}} \quad & \theta^T f(\mathbf{s}, \mathbf{u}) \\ \text{s.t.} \quad & \text{vehicle dynamic constraints} \end{aligned} \tag{2}$$

1) Vehicle Dynamic Constraints: The vehicle dynamic constraints represent the kinematics of vehicle motion, shown in Equations 3,4,5,6, where τ is the step size. Equation 3 reflects the relationship between the longitudinal coordinate change and the heading angle, and similarly, Equation 4 reflects the relationship between the lateral coordinate change and the heading angle. Equation 5 shows the relationship between the heading angle rate ψ and the heading angle. Equation 6 shows the vehicle dynamics between the velocity and the acceleration.

$$x(i+1) = x(i) + v(i)cos(\psi(i) + \psi_r(i))\tau \tag{3}$$

$$y(i+1) = y(i) + v(i)sin(\psi(i) + \psi_r(i))\tau \tag{4}$$

$$\dot{\psi}(i) = \frac{(\psi(i+1) - \psi(i))}{\tau} \tag{5}$$

$$a(i) = \frac{v(i+1) - v(i)}{\tau} \tag{6}$$

2) Feature Vector: The feature vector represents a desired driving policy, which is a linear combination of multiple driving features. In our proposed model, nine features are designed to describe the driving policy including both longitudinal and lateral behaviors. The following provides detailed descriptions of the features, where N represents the total number of data points in a trajectory. In the AV experiment, (1) - (7) are selected as features, and in the CV experiment, (1)(2)(5)(6)(8)(9) are selected as features. The criteria for feature selection come from the differences in the driving scenarios. For example, feature 9 is selected for the CV experiment because we consider a roundabout scenario.

(1) Speed limit: $f_1 = \frac{1}{N}\sum_i (v_i - v^{lim})^2$. This feature measures the difference between the speed at each time step v_i and the speed limit v^{lim}, which models the driving incentive of approaching the desired speed (i.e., speed limit).

(2) Acceleration: $f_2 = \frac{1}{N}\sum_i a_i^2$. It is the summation of the square of the acceleration at each time step, which represents the smoothness of driving behaviors.

(3) Car following: $f_3 = \frac{1}{N}\sum_i \frac{1}{\min(d_i, d_i/v_i)^2}$. d_i is the distance to the leading vehicle at time step i. $min(d_i, \frac{d_i}{v_i})$ chooses the smaller value between the distance and time headway. When the vehicle moves in free flow, the time headway makes an impact. When the vehicle is about to stop, the distance to the leading vehicle makes an impact. This feature models the car-following behavior of the vehicles.

(4) Lateral acceleration: $f_5 = \frac{1}{N}(a_i \sin(\psi_i))^2$. It is the summation of the square of the lateral acceleration at each time step, which measures the smoothness of lateral driving behaviors.

(5) Heading angle: $f_5 = \frac{1}{N}\sum_i \psi_i^2(1 - I^{lanechange})$. ψ_i is the heading angle at time step i. $I^{lanechange}$ is the indicator

of lane change, which is 1 if the heading angle between the longitudinal axis of the vehicle and the longitudinal direction of the road is larger than a threshold.

(6) Heading rate: $f_6 = \frac{1}{N}\sum_i(\dot{\psi}_i)^2$. ψ_i is the heading angle change rate at time step i.

(7) Heading rate: $f_7 = \frac{1}{N-1}\sum_i(\dot{\psi}_{i+1} - \dot{\psi}_i)^2$. This feature measures the change rate of the heading angle rate. Features 5-7 represent the smoothness of the heading angle to measure the smoothness of lateral driving behaviors of the vehicle.

(8) Interaction: $f_8 = \frac{1}{N}\sum_i \frac{1}{(x_i - x_i^{other})^2 + (y_i - y_i^{other})^2}$. This feature measures the interaction between two vehicles in Euclidean distance. Here, $(x_i^{other}, y_i^{other})$ denotes the position of the other interactive vehicle w.r.t. the ego vehicle at time step i.

(9) Curvature: $f_9 = \frac{1}{N}\sum_i(\sqrt{(x_i - x_i^c)^2 + (y_i - y_i^c)^2} - r_i)^2$. When the vehicle is turning, this feature captures the curvature following behavior. (x^c, y^c) is the center of the turning circle, and r denotes the turning radius.

3) Maximum Entropy Inverse Reinforcement Learning: Before solving the optimization problem, the weight vector θ needs to be determined, which balances the driving features in the feature vector. It is usually difficult to specify proper weights, which represent the desired driving policy. In this study, we apply maximum entropy inverse reinforcement learning to determine the weight vector θ. Considering the vehicle trajectory planning as a Markov Decision Process (MDP) with a discounted cost as Equation 7, in which γ is the discounted factor and r is a reward function. If the discounted factor is taken as 1, then the total return is $\theta^T f(s,u)$, for each trajectory s. The goal of inverse reinforcement learning is to find the weight vector θ that maximizes the log-likelihood function $L(\theta)$, shown in Equation 8. D is the demonstration trajectory dataset collected, including m trajectories. $P(s_j|\theta, u_j)$ is the probability of trajectory s_j given parameter θ and the initial condition as well as the environment state of trajectory s_j (i.e., u_j), so when maximizing $L(\theta)$, the likelihood of using weight θ to generate all trajectories in the dataset is maximized. When the policy of the MDP is the maximum entropy policy [40], $P(s_j|\theta, u_j)$ can be written as Equation 9.

$$discountedcost = \sum_{i=0}^{N-1} \gamma^i r(s_i) \approx \theta^T f(s,u) \tag{7}$$

$$L(\theta) = \frac{1}{m}\sum_{s_j \in D} lnP(s_j|\theta, u_j) \tag{8}$$

$$p(s_j|\theta, u_j) = \frac{e^{-\theta^T f(s_j, u_j)}}{\sum_{s_k \in C_j} e^{-\theta^T f(s_k, u_j)}} \tag{9}$$

In this way, the gradient of $L(\theta)$ can be calculated as Equation 10, in which $\tilde{f} = \frac{1}{m}\sum_{s_j \in D} f(s_j, u_j)$denotes the empirical feature vector. Thus, the gradient of $L(\theta)$ is the difference between the expected feature vector with respect to weight θ and the empirical feature vector calculated from the dataset (i.e., observations). Furthermore, the expected feature vector can be approximated by the feature vector of the most likely trajectory.

Algorithm 1 Maximum Entropy Inverse Reinforcement Learning

1: Compute the empirical feature vector over all demonstrations $\tilde{f}_0 = \frac{1}{m}\sum_{s_j \in D} f(s_j, u_j)$. Normalize the feature vector. The normalized feature vector is denoted as $\tilde{f}$
2: Initialize every entry of the weight vector θ.
3: **while** $\frac{1}{m}\sum_{j=1} f(s_j^\theta, u_j) - \tilde{f} > threshold$ **do**
4: **for** For each demonstrated trajectory in the dataset **do**
5: Fix the initial condition and the environment states and optimize the trajectory using equation 2. The optimized trajectories are denoted as $s_1^\theta, \ldots, s_m^\theta$.
6: **end for**
7: The gradient can be calculated as $\nabla_\theta L(\theta) = \frac{1}{m}\sum_{j=1} f(s_j, u_j) - \tilde{f}$.
8: Update the parameter vector: $\theta(k+1) = \theta(k) + \gamma \nabla_\theta L(\theta)$, in which γ is the learning rate.
9: **end while**

$$\begin{aligned}\nabla_\theta L(\theta) &= \frac{1}{m}\sum_{s_j \in D} E_{p(s_j|\theta,u_j)}[f(s_j,u_j)] - \tilde{f} \\ &\approx \frac{1}{m}\sum_{s_j \in D} f(argmin_{s_j}\theta^T f(s_j,u_j)) - \tilde{f}\end{aligned} \tag{10}$$

With the gradient of the log-likelihood function, the pseudo-code of the maximum entropy inverse reinforcement learning algorithm can be summarized in Algorithm 1, given a set of demonstration trajectories $D = s_1, \ldots, s_m$.

C. Anomaly Classifier

To differentiate normal trajectories from abnormal ones, the difference between the observed trajectory and the predicted optimal trajectory should be measured quantitatively by some statistics. In this work, three statistical features are adopted. The first statistical feature is the maximum value of the objective ratio OR_t of all the trajectory points until time step t, calculated by Equation 11. OR_t represents the ratio between the summation of the objective value of the observed trajectory (i.e., $\sum_{\tau=1}^{t} observed\ objective_\tau$) and the summation of the objective value of the predicted optimal trajectory via the learned model (i.e., $\sum_{\tau=1}^{t} optimal\ objective_\tau$) at time step t. It measures how different the observed trajectory is from the optimal trajectory.

$$OR = \max_{1\ldots t} OR_t = \frac{\sum_{\tau=1}^{t} observed\ objective_\tau}{\sum_{\tau=1}^{t} optimal\ objective_\tau} \tag{11}$$

The second statistical feature adopted is the max value of the normality score NS_t of all the observed trajectory points until time step t, calculated by Equation 12. NS_t measures the variation of the objective value of the observed trajectory. $objective_t$ denotes the objective value of the observed trajectory at time step t. $objective\ mean_{1\ldots t}$ is the mean objective value of all the observed time steps until t, $objective\ std_{1\ldots t}$ is the standard deviation of the objective value of all the observed

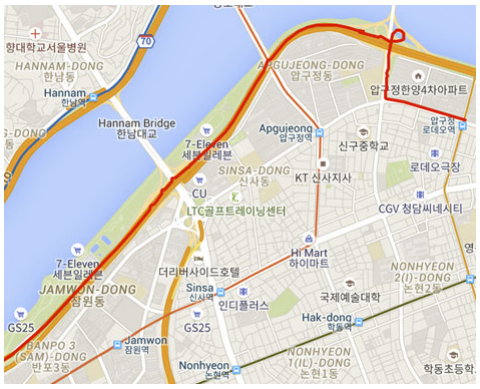


Fig. 5. A sample trajectory in KAIST dataset.

time steps until t.

$$NS = \max_{1...t} NS_t = \frac{objective_t - objective\ mean_{1...t}}{objective\ std_{1...t}} \quad (12)$$

The last statistical feature is the maximum value of the average displacement error ED_t with the prediction horizon of T between the observed trajectory and optimized trajectory at time step t, calculated by Equation 13. The average displacement error at time step t (i.e. ED_t) can be calculated by measuring the average point-wise Euclidean distance between the observed trajectory (x^{obs}, y^{obs}) and the predicted trajectory (x^{pred}, y^{pred}) within the prediction horizon T. This feature captures the difference between the observation and the optimization results in terms of the Euclidean distance in the 2-D space.

$$\begin{aligned} ED &= \max_{1...t} ED_t \\ &= \frac{1}{T}\sum_{i=t}^{t+T}\sqrt{(x_i^{obs} - x_i^{pred})^2 + (y_i^{obs} - y_i^{pred})^2} \end{aligned} \quad (13)$$

With three statistical features defined as the input, a decision tree classifier is applied to differentiate the abnormal trajectories from normal trajectories. More information of the decision tree classifier can be found in [41].

IV. Detection Model Evaluation on AV Threat Model

The AV threat model (i.e., FusionRipper [11]) is implemented on the KAIST Complex Urban dataset [29], which is an AV driving dataset in both urban and highway driving scenarios based on the Apollo system. Figure 5 illustrates a sample trajectory (in red) that consists of both urban and highway driving scenarios in the KAIST dataset. The dataset provides raw data from Lidar, stereo camera, GPS, and IMU. The FusionRipper algorithm takes them as the input and applies the MSF module in Apollo to obtain the compromised localization results. Specifically, lateral deviations are added to the original trajectory data. In this study, the deviated trajectory generated by FusionRipper is considered as the trajectory under attack. Meanwhile, we extract the original AV trajectories in the data as ground truth.

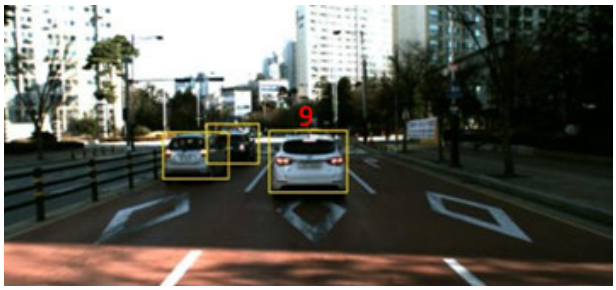


Fig. 6. Vehicle detection and distance measurement result.

A. Data Processing

Before applying the proposed detection method, the original KAIST trajectory data set needs to be processed to calculate two additional data elements (road orientation and car-following distance) that are needed for the proposed learning from demonstration model. The heading angle of the trajectory is the relative angle between the longitudinal axis of the vehicle and the longitudinal direction of the road, but the road orientation is not included in the raw data. To calculate the relative heading angle, vehicle trajectories are allocated to the closest road segment on the OpenStreetMap [42], and then the road orientation is extracted from the waypoints of the OpenStreetMap. In the feature vector, the distance to the leading vehicle is also required to calculate the car-following distance, which is extracted from the raw images from the forward-facing stereo cameras installed on the vehicle, using the YOLO (You only look once) algorithm [43]. The disparity of the detected vehicle between the left and right stereo camera is obtained from the images to calculate the distance to the leading vehicle using triangulation. Interpolation is applied when the front vehicle is missing. Figure 6 presents an example of the detected leading vehicle denoted by the yellow rectangle, and distance measurement in meters denoted by the red number. With the road orientation and distance to the leading vehicle processed, all the features in the learning from demonstration model can be calculated.

B. Experiment Setting

In total, 78 ground truth trajectories are obtained from the KAIST dataset. The ground truth trajectories include both in-lane driving and lane changing cases. We deliberately include the lane changing cases in the training dataset because the FusionRipper attack aims to create abnormal lateral deviations of a trajectory to achieve the off-road or wrong-way attack goal. It is critical to differentiate between a normal lane changing process, which also includes lateral deviations, and the lateral deviations caused by the attack. 51 ground truth trajectories are adopted in IRL to learn the driving model, which is significantly less than the trajectories used in other learning-based detection methods [27], [28]. 88 attacked trajectories from the FusionRipper attack and all ground truth trajectories are utilized in the experiments for detection.

In the experiments, we mainly evaluate the proposed detection method against the off-road attack, which requires less lateral deviation and thus is more difficult for the detection

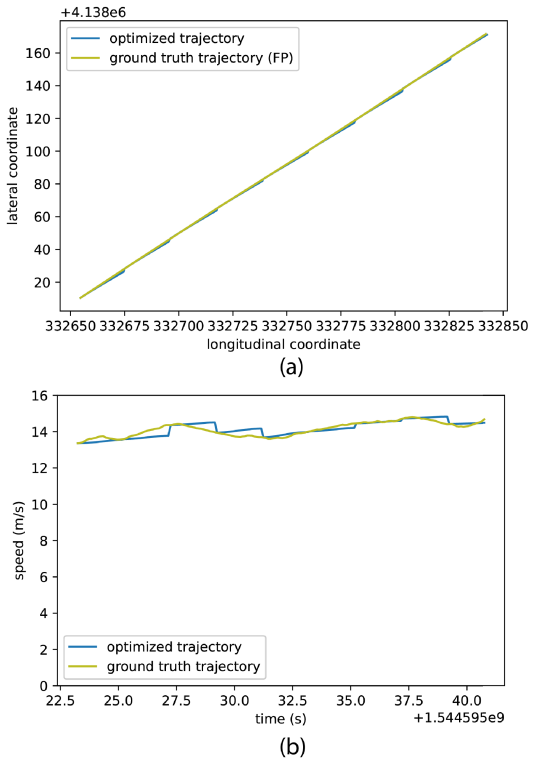


Fig. 7. Comparison between a ground truth trajectory and a predicted optimal trajectory ((a): trajectory profile in 2-D space. (b): speed profile).

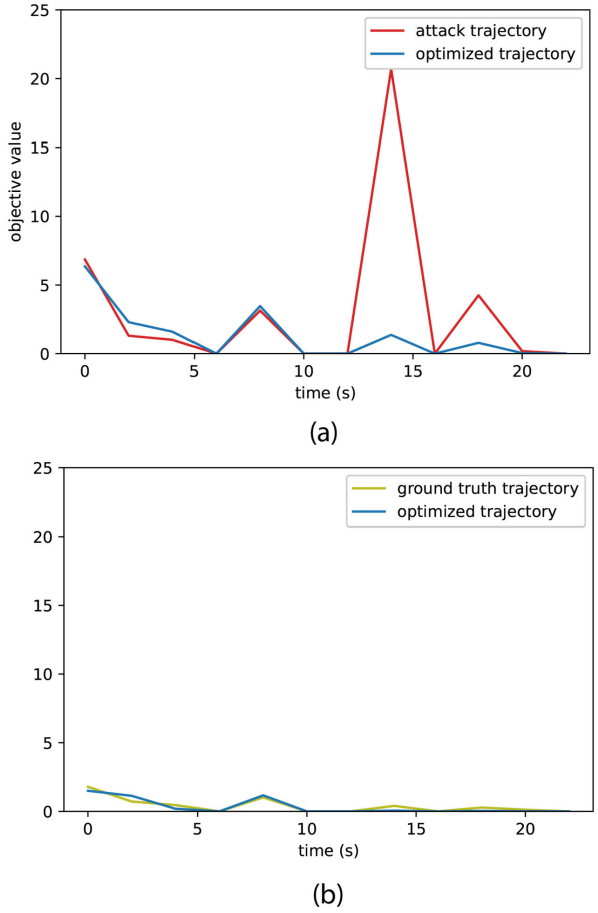


Fig. 8. Objective value comparison of the attacked trajectory and the ground truth trajectory ((a): objective value comparison of an attacked trajectory (b): objective value comparison of a ground truth trajectory).

model. Two types of detection mechanisms, namely offline detection and online detection, are designed and evaluated. In the offline detection, the detection is performed after the full trajectory of the vehicle is observed. In the online detection mode, the anomaly classifier checks the trajectory every 0.5 seconds until classified as abnormal or reaching the end of the attack. The online detection mode is designed to detect the abnormal trajectories in real-time as soon as possible, which is critical to the safety performance of the AVs, but also more challenging.

C. Experiment Results

Figure 7 illustrates the performance of the learning from demonstration model, by comparing a ground truth trajectory (i.e., green curve) with its corresponding predicted optimal trajectory from the learned model (i.e., blue curve). Subfigure (a) shows the position profile and subfigure (b) shows the speed profile. The prediction horizon is 2 seconds, indicating that the learned model takes an accurate position (i.e., the same as the ground truth position profile) and the corresponding speed as the input every 2 seconds. The generated trajectory is very close to the ground truth trajectory. To quantitatively measure the difference between the learned model and the ground truth trajectory, the Average Displacement Error (ADE) between the ground truth trajectory and the predicted optimal trajectory is calculated as 0.76m. The calculation of the ADE is based on the of ED_t in Equation 13. The ADE is less than 1m, which indicates that the predicted optimal trajectory fits the ground truth very well. Comparing with other learning-based approaches such as LSTM [44], the ADE is lower.

Based on the learned model, the value of the objective function of both observed trajectories and predicted optimal trajectories (by solving Equation 2) can be calculated by evaluating $\theta^T f(s,u)$, in which θ is optimized by the maximal entropy inverse reinforcement learning model. Intuitively, if the vehicle is not under attack, the value of the objective function calculated from the observed trajectory should be close to the value calculated from the predicted optimal trajectory. If the vehicle is under attack, then the two values should deviate from each other. Figure 8 illustrates a comparison of the objective values of the attacked trajectory, ground truth trajectory, and predicted optimal trajectory. The red curve is the objective of the attacked trajectory, and the green curve denotes the objective of the ground truth trajectory. The blue curves in both subfigures denote the objective value of the predicted optimal trajectory. In subfigure (a), when the attack is about to succeed (i.e., time $\geq$ 14 s), the objective value of the attacked trajectory deviates from the optimal value significantly. By comparing subfigures (a) and (b), the objective of the ground truth trajectory is much closer to its corresponding optimal objective value than the attacked trajectory.

Figure 9 shows the 3D (subfigure (a)) and 2D (subfigure (b)) scatter plots of the decision tree classification result. The

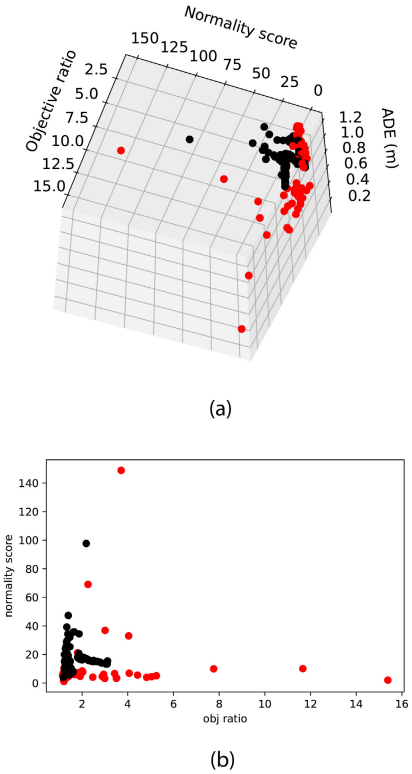


Fig. 9. Scatter plot of the classification problem ((a): scatter plot with three features. (b): scatter plot with two features).

red dots represent the attacked trajectories, and the black dots represent the ground truth trajectories. In the 3D scatter plot, three axes represent the three statistical features introduced in section III-C, which are objective ratio, normality score, and average displacement error. With three features, the normal trajectories in the ground truth can be separated from the attacked trajectories, as shown in the 3D scatter plot. Subfigure (b) shows the distribution of trajectories if the average displacement error feature is removed. The classifier is difficult to differentiate since the normal trajectories and the attacked trajectories are mixed together. The scatter plots illustrate that the choice of these three statistical features is appropriate.

Next, we show the results of the offline and online detection respectively. In the offline detection, false positive (Type I error) indicates that a ground truth trajectory is classified as an attacked trajectory. On the contrary, false negative (Type II error) means that an attacked trajectory is not identified correctly. The false positive rate of offline detection is 8.7% (2/23), and the false negative rate is 3.7% (1/27). Figure 10 illustrates a false positive case and a false negative case. In subfigure (a) and (b), green curves represent the ground truth trajectories in the dataset, and the blue curves denote the optimized trajectories, respectively. Subfigure (a) shows the false positive case in the 2-D space, in which the optimized trajectory does not fit the ground truth very well, compared to a true positive case in subfigure (b) where the optimized trajectories almost overlap with the ground truth trajectory. The reason for the unsatisfying fitting in the false positive case is that the road orientation at this road segment calculated

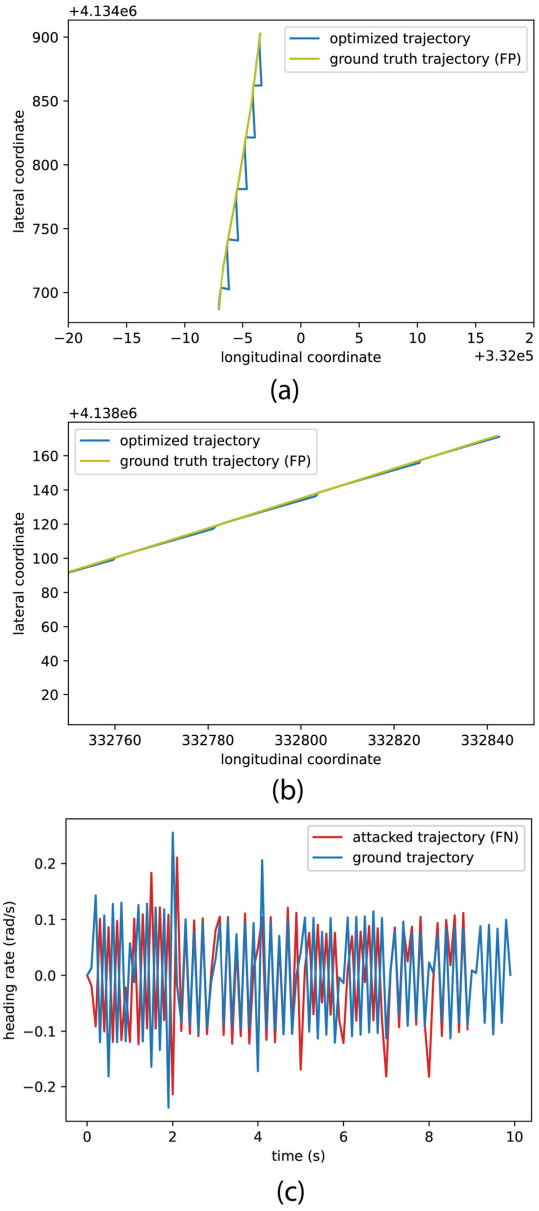


Fig. 10. Misclassification examples of the KAIST experiments ((a): trajectory profile of the FP case. (b): baseline trajectory profile for the FP case. (c) heading rate profile of the FN case).

from the map data is not accurate. The KAIST dataset doesn't contain HD map information, we use the OpenStreetMap in calculating the road orientation. Thus, the optimized trajectory tries to follow the road direction within the prediction horizon and deviates from the ground truth trajectory. Such an issue may be resolved when the AV is equipped with a HD map, which is a standard module. Subfigure (c) shows the heading rate profile of the false negative case that an attacked trajectory is misclassified. The red curve denotes the heading rate profile of the attacked trajectory, and the blue curve denotes the heading rate profile from a ground truth trajectory. The heading rate is a key feature in the learned driving model, and in this case, the heading rate of the attacked trajectory is similar to the heading rate of the ground truth trajectory, which makes the attacked trajectory difficult to be identified.

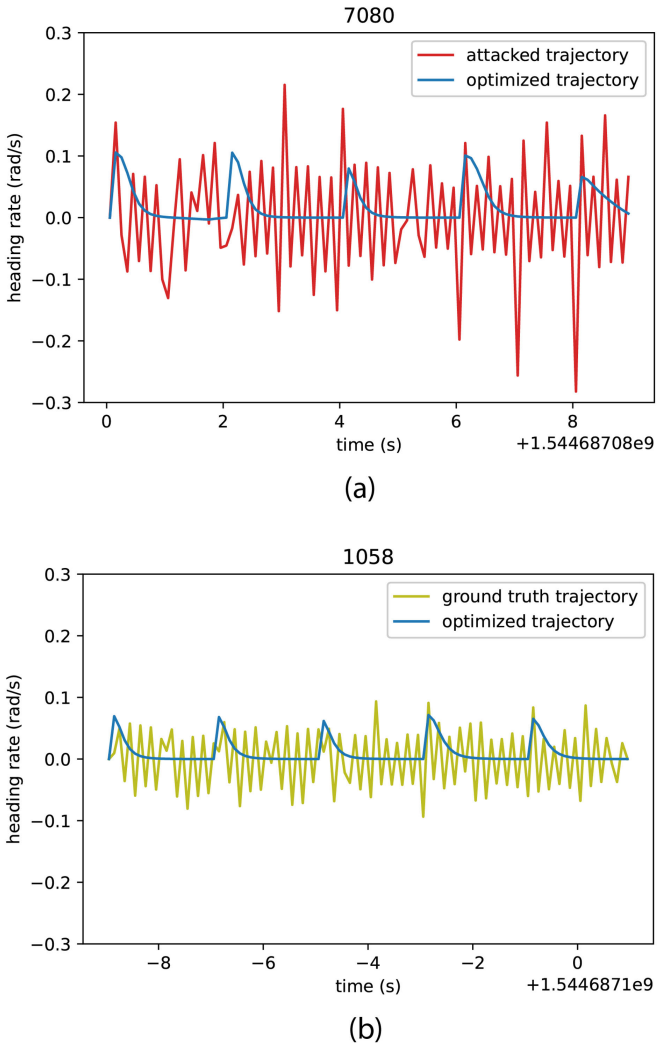


Fig. 11. Heading rate profiles of the attacked trajectory and the ground truth trajectory ((a): attacked trajectory. (b): ground truth trajectory).

Figure 11 further shows the heading rate profile comparison, which reveals the reason why the attacked trajectories can be differentiated from the ground truth trajectories. The red curve denotes the heading rate profile of the attacked trajectory, and the green curve denotes the heading rate profile of the ground truth trajectory. The blue curves in both subfigures denote the heading rate profiles of the corresponding predicted optimal trajectories. Notice that the heading rate profiles of the predicted optimal trajectories fluctuate every 2 seconds since the prediction horizon is 2 seconds. In subfigure (a), the heading rate profile of the attacked trajectory has larger fluctuations compared to the predicted optimal trajectory. The ground truth trajectory, on the contrary, has smaller values and small fluctuations. Such differences are reflected in the objective function value, which is one feature in the classification model.

In the online detection, the anomaly classifier checks the trajectory every 0.5 seconds. The performance of the online detection is shown in Table I. The false positive rate is 8.7% (2/23), and the false negative rate is 3.7% (1/27), which is the same as the offline detection results. For online detection,

TABLE I
PERFORMANCE OF ONLINE DETECTION

FP	FN	Mean attack success time (s)	Mean detection time (s)	Mean time to attack success (s)
2/23	1/23	28.7	12.7	16.0

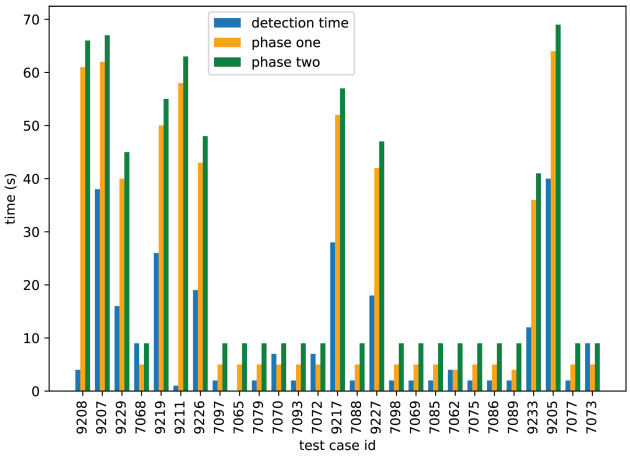


Fig. 12. Detection time in online anomaly detection for KAIST experiments.

it is important to identify the attacked trajectory as early as possible, but at least before the attack succeeds. Therefore, we further calculate the mean detection time to compare it with the mean attack success time. The mean success time of the off-road attack is 28.7 s, and the mean detection time is 12.7 s. The time to attack success is defined as the time duration from the success time of the detection to the success time of the attack, which measures how early the attack can be detected before attack success. In the online detection, the attacked trajectories can be identified 16.0 s before the attack success time on average.

Figure 12 further illustrates the detection time (i.e., blue bars) in the online detection, compared to the duration of attack phases one and two. The green bars denote the duration of phase one (i.e., vulnerability profiling), and the red bars denote the attack success time, which is the end of phase two (i.e., aggressive spoofing). In general, except for the false negative case 7109, the detection time of all other test cases is not longer than the attack success time, which indicates that the anomaly classifier can successfully detect the spoofing attack before it achieves the attack goal. For most of the test cases, the detection time is even less than the duration of the vulnerability profiling phase, which leaves sufficient time for applying further mitigation strategies.

V. DETECTION MODEL EVALUATION ON CV THREAT MODEL

To validate the CV threat model, experiments are conducted using a dataset collected from a two-lane roundabout in Ann Arbor, Michigan [30]. Infrastructure sensors (i.e. cameras and radars) are installed at the four corners of the roundabout, to detect all vehicles approaching and entering the roundabout. The trajectory dataset is collected at a 2.5 Hz frequency in a

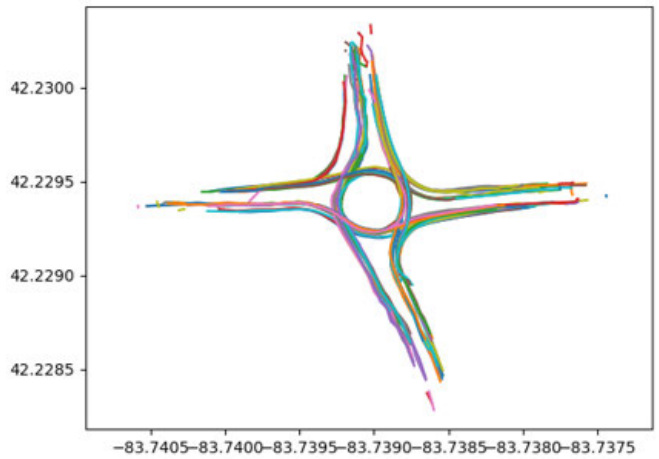


Fig. 13. Trajectory overview at the two-lane roundabout.

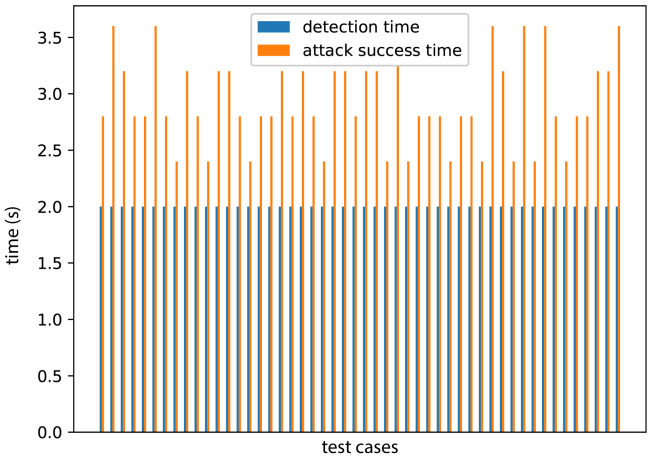


Fig. 14. Detection time in online anomaly detection for roundabout experiments.

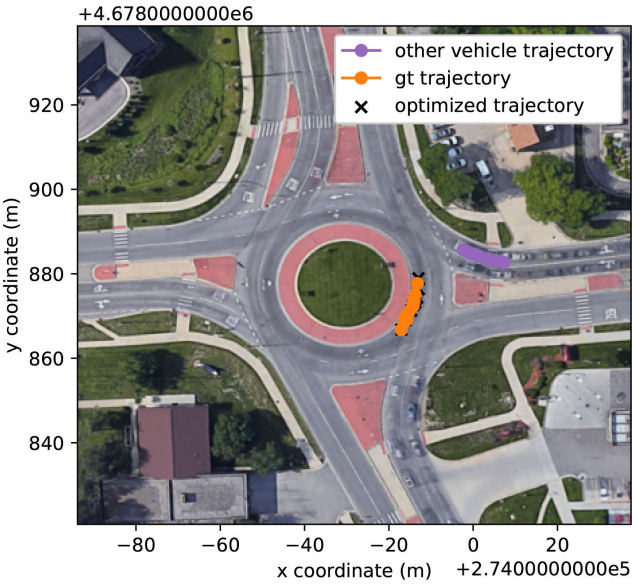


Fig. 15. Misclassification examples of the roundabout experiments (a FP case).

24/7 manner. Figure 13 illustrates the trajectory overview at the two-lane roundabout

In this experiment, 841 ground truth trajectories from the dataset are extracted. 809 attack trajectories are generated with the attack model illustrated in Section II-B. Overall, there are 1650 trajectories in the experiment. 182 ground truth trajectories are utilized in learning the driving model with IRL, which is still much fewer compared with other learning-based detection algorithms [27], [28]. 70% of the trajectories in the dataset are used as the training data, and the rest 30% are used as the testing data. Compared to the threat model for AV, the CV threat model is much more aggressive with a much shorter attack duration, which greatly increases the difficulty for both offline and online detection. Besides, the frequency of the ground truth data is only 2.5 Hz, which also makes the defense more challenging. Similar as in the AV case, in offline detection, the detection is performed after the full trajectory of the vehicle is observed. In online detection, after a sufficient number of trajectory points are observed from a vehicle (e.g., 5 data points), the online detection is conducted at every time step.

In offline detection, the false positive rate is 0.004% (1 out of 252), and the false negative rate is 2.0% (5 out of 243). Overall, the anomaly detection model shows a very good performance in offline detection. In online detection, the anomaly classifier checks the trajectory every 0.4 seconds until classified as abnormal or reaching the end of the attack. Overall, the false positive rate is 0.008% (2 out of 252), and the false negative rate is 0% (0 out of 243). Notice that in the online detection, all the attack trajectories can be identified by the proposed anomaly detector. Figure 14 shows the relationship between the detection time (blue bar) and the attack success time (orange bar) in the first 50 cases. All the cases can be detected before the attack success time, and the average detection time is 2.0 seconds after the attack starts. The average attack success time is 2.8 seconds, and the average time to attack success is 0.8 seconds. Thus, even with very short attack duration, the proposed anomaly detection still manages to identify the falsified trajectories before the attack succeeds.

Figure 15 shows a misclassification example of a false positive (FP) case in the online detection. The purple trajectory denotes the trajectory of the vehicle at the entrance link. The orange trajectory denotes a ground truth trajectory traversing the roundabout, and the black trajectory with crosses denotes the predicted optimal trajectory w.r.t. the orange trajectory. The x-axis and y-axis represent the local coordinate system in meters. In 3.6 seconds, the ground truth trajectory drives 12.7 m in total, and the average speed of the vehicle in the roundabout is only 3.5 m/s. Such low speed is very rare at this roundabout, which makes the proposed anomaly detector consider it as an abnormal trajectory. In fact, identifying this uncommon behavior in the ground truth trajectory also enriches the usage of the proposed anomaly detection framework. Other than detecting cyber attacks, this method may also be utilized to identify abnormal driving behaviors.

VI. Detection on Adaptive Attack

To further evaluate the capability of the proposed anomaly detection model, an adaptive attack scenario is designed and

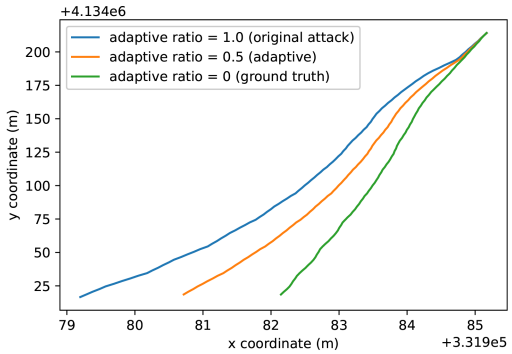


Fig. 16. Adaptive attack example on the KAIST dataset.

TABLE II
DETECTION PERFORMANCE ON THE ADAPTIVE ATTACK

Adaptive ratio	False positive	False negative
0.8	2/23	2/27
0.5	2/23	2/27
0.2	2/23	10/27

implemented. In the AV threat model, the key idea is to add lateral deviations to the original trajectory, which either causes the subject AV hit the roadside, or leads to the emergency behavior of the victim vehicle. To make the attack more stealthy and difficult to be detected, the adaptive attack reduces the magnitude of the lateral deviations added to the ground truth trajectory, and we use the adaptive ratio to represent the significance of the magnitude reduction, ranging from 0 to 1. Figure 16 shows an example of the adaptive attack implemented on the KAIST dataset. The green curve denotes the ground truth trajectory that is not attacked (i.e., adaptive ratio = 0). The blue curve denotes the original attack trajectory that is evaluated in section IV (i.e., adaptive ratio = 1). The orange curve denotes the trajectory under adaptive attack, with the adaptive ratio of 0.5. Notice that the lateral deviations of the orange trajectory are half of the lateral deviations of the original attack trajectory, w.r.t. the ground truth trajectory. In this way, as the adaptive ratio decreases, the adaptive attack trajectories become more and more similar to the ground truth trajectory. The adaptive attack trajectories with a very small adaptive ratio (e.g., 0.1) can be very close to the ground truth trajectories, which are very difficult to identify.

The adaptive attack is implemented on the KAIST dataset with the adaptive ratios of 0.8, 0.5, and 0.2. The experiment setting is the same as in section IV, and the driving models and the decision tree classifier are also the same. Table II shows the anomaly detection performance on the adaptive attack. When the adaptive ratio is 0.8, the performance of the anomaly detection degrades a little, with a false negative rate of 2/27. Nonetheless, the detection results are still satisfying, and most of the attacked trajectories can be identified correctly. When the adaptive ratio is 0.5, the performance of the anomaly detection is the same as the performance with the adaptive ratio of 0.8. When the adaptive ratio is 0.2, the false negative rate become 10/27. However, in this case, the adaptive attack trajectories are very close to the ground truth trajectories. In the urban scenario, the success criterion of the off-road attack is 0.895 m [11]. With the adaptive ratio of 0.2, the final lateral deviation w.r.t. the ground truth trajectory is only 0.179 m. In such cases, the subject AV is still driving within the original lane. Although the proposed model fails to detect some attack trajectories, the consequence is not hazardous.

VII. DISCUSSION AND CONCLUSION

In this paper, an anomaly detection model using learning from demonstration is proposed to detect GPS spoofing attacks towards the localization system of the CV/AV. Maximum entropy inverse reinforcement learning is applied to learn the normal driving model. The learned driving model is then utilized to generate optimal vehicle trajectories which are compared with the observed vehicle trajectories using a decision tree classifier to determine whether the observed trajectories are under attack. The proposed detection method is evaluated in two realistic GPS spoofing attacks on AV and CV, respectively.

In both AV and CV experiments, the proposed anomaly detection method can identify most of the abnormal trajectories before the attacks succeed. Such experiment results validate the generality of the proposed model. Notice that although in this paper, the anomaly detection model is only validated by GPS spoofing attack experiments, we do not utilize any specific feature of GPS signals in the model. In other words, the proposed model has the potential to detect a variety of sensor attacks towards the localization system as well. The reason is that the key concept of the proposed model is to compare normal versus abnormal driving behaviors. Thus, it is not sensitive to the input types or states of the localization system. As long as the driving behaviors are affected by certain cyber attacks, the proposed method can be applied to detect anomaly.

One limitation of the proposed method is that it can be only applied to detect known attacks, because the decision tree classifier requires attack trajectories as training data. To extend the proposed method to be more generic for detecting unknown attacks, we will explore one-class classification methods [45] where only the ground truth trajectories are needed for training a single classifier. Another limitation of this paper is that it mainly focuses on the detection of GPS spoofing attacks without proposing defense solutions. In future work, we will investigate corresponding mitigation strategies. For example, when a trajectory is identified as abnormal, its autonomous driving functions can be temporarily suspended. In the online detection, a warning can be sent to the trajectory planning module of the AV to choose safe maneuvers (e.g., stop) or directly ask the driver to take over. In a CV environment, the certificate of the vehicle could be revoked so that the messages sent from this particular CV will be discarded by other vehicles.

ACKNOWLEDGMENT

The views presented in this paper are those of the authors alone.

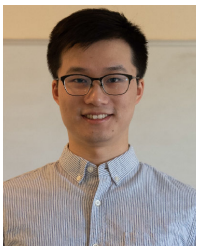
REFERENCES

[1] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected vehicles: Solutions and challenges," *IEEE Internet Things J.*, vol. 1, no. 4, pp. 289–299, Aug. 2014.

[2] D. G. Yang et al., "Intelligent and connected vehicles: Current status and future perspectives," *Sci. China-Technol. Sci.*, vol. 61, no. 10, pp. 1446–1471, Oct. 2018.
[3] Z. Yang, Y. Feng, X. Gong, D. Zhao, and J. Sun, "Eco-trajectory planning with consideration of queue along congested corridor for hybrid electric vehicles," *Transp. Res. Record, J. Transp. Res. Board*, vol. 2673, no. 9, pp. 277–286, Sep. 2019.
[4] T. G. R. Reid et al., "Localization requirements for autonomous vehicles," 2019, *arXiv:1906.01061*.
[5] S. Campbell et al., "Sensor technology in autonomous vehicles: A review," in *Proc. 29th Irish Signals Syst. Conf. (ISSC)*, 2018, pp. 1–4.
[6] Y. Gao, S. Liu, M. Atia, and A. Noureldin, "INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm," *Sensors*, vol. 15, no. 9, pp. 23286–23302, Sep. 2015.
[7] A. Soloviev, "Tight coupling of GPS, laser scanner, and inertial measurements for navigation in urban environments," in *Proc. IEEE/ION Position, Location Navigat. Symp.*, Jun. 2008, pp. 511–525.
[8] J. Kelly and G. S. Sukhatme, "Visual-inertial sensor fusion: Localization, mapping and sensor-to-sensor self-calibration," *Int. J. Robot. Res.*, vol. 30, no. 1, pp. 56–79, 2011.
[9] X. Sun, F. R. Yu, and P. Zhang, "A survey on cyber-security of connected and autonomous vehicles (CAVs)," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 6240–6259, Jul. 2021.
[10] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Capkun, "On the requirements for successful GPS spoofing attacks," in *Proc. 18th ACM Conf. Comput. Commun. Secur.*, Oct. 2011, pp. 75–86.
[11] 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 *Proc. 29th USENIX Secur. Symp. (USENIX Security)*, 2020, pp. 931–948.
[12] T. E. Humphreys et al., "Assessing the spoofing threat: Development of a portable GPS civilian spoofer," in *Proc. 21st Int. Tech. Meeting Satell. Division Inst. Navigat. (ION GNSS)*, 2008, pp. 2314–2325.
[13] 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-from-regulus-cyber-shows/
[14] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," *Black Hat Eur.*, vol. 11, p. 995, Nov. 2015.
[15] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl, "Survey on misbehavior detection in cooperative intelligent transportation systems," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 1, pp. 779–811, 4th Quart., 2018.
[16] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the usefulness of watchdogs for intrusion detection in VANETs," in *Proc. IEEE Int. Conf. Commun. Workshops*, May 2010, pp. 1–5.
[17] A. Hamieh, J. Ben-Othman, and L. Mokdad, "Detection of radio interference attacks in VANET," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Nov. 2009, pp. 1–5.
[18] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan, and M. Rajarajan, "Host-based intrusion detection for VANETs: A statistical approach to rogue node detection," *IEEE Trans. Veh. Technol.*, vol. 65, no. 8, pp. 6703–6714, Aug. 2016.
[19] J. Grover, M. S. Gaur, V. Laxmi, and N. K. Prajapati, "A Sybil attack detection approach using neighboring vehicles in VANET," in *Proc. 4th Int. Conf. Secur. Inf. Netw.*, 2011, pp. 151–158.
[20] A. Kalantari and E. G. Larsson, "Statistical test for GNSS spoofing attack detection by using multiple receivers on a rigid body," *EURASIP J. Adv. Signal Process.*, vol. 2020, no. 1, pp. 1–16, Dec. 2020.
[21] E. Schmidt, N. Gatsis, and D. Akopian, "A GPS spoofing detection and classification correlator-based technique using the LASSO," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 6, pp. 4224–4237, Dec. 2020.
[22] R. Matsumura, T. Sugawara, and K. Sakiyama, "A secure LiDAR with AES-based side-channel fingerprinting," in *Proc. 6th Int. Symp. Comput. Netw. Workshops (CANDARW)*, Nov. 2018, pp. 479–482.
[23] H. Stubing, A. Jaeger, C. Schmidt, and S. A. Huss, "Verifying mobility data under privacy considerations in car-to-X communication," in *Proc. 17th ITS World Congr. ITS Jpn. ITS AmericaERTICO*, 2010.
[24] H. Stübing, J. Firl, and S. A. Huss, "A two-stage verification process for car-to-X mobility data based on path prediction and probabilistic maneuver recognition," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2011, pp. 17–24.
[25] A. Jaeger, N. Bissmeyer, H. Stubing, and S. Huss, "A novel framework for efficient mobility data verification in vehicular ad-hoc networks," *Int. J. Intell. Transp. Syst. Res.*, vol. 10, pp. 11–21, Jan. 2012.
[26] C. Yavvari, Z. Duric, and D. Wijesekera, "Vehicular dynamics based plausibility checking," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.
[27] S. Dasgupta, M. Rahman, M. Islam, and M. Chowdhury, "A sensor fusion-based GNSS spoofing attack detection framework for autonomous vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 12, pp. 23559–23572, Dec. 2022.
[28] P. Jiang, H. Wu, and C. Xin, "DeepPOSE: Detecting GPS spoofing attack via deep recurrent neural network," *Digital Communications and Networks*, vol. 8, no. 5, pp. 791–803, 2021.
[29] J. Jeong, Y. Cho, Y.-S. Shin, H. Roh, and A. Kim, "Complex urban dataset with multi-level sensors from highly diverse urban environments," *Int. J. Robot. Res.*, vol. 38, no. 6, pp. 642–657, May 2019.
[30] R. Zhang, Z. Zou, S. Shen, and H. X. Liu, "Design, implementation, and evaluation of a roadside cooperative perception system," in *Proc. 101st Transp. Res. Board (TRB) Annu. Meeting*, 2022, pp. 273–284.
[31] C. Badue et al., "Self-driving cars: A survey," *Exp. Syst. Appl.*, vol. 165, Mar. 2021, Art. no. 113816.
[32] D. Bezzina and J. Sayer, "Safety pilot model deployment: Test conductor team report," Nat. Highway Traffic Safety Admin., Washington, DC, USA, Tech. Rep. DOT HS 812 171, 2014.
[33] D. Gopalakrishna et al., "Connected vehicle pilot deployment program phase 1, concept of operations (ConOps), ICF/WYDOT [phase 2 update]," U.S. Dept. Transp., Intell. Transp., Washington, DC, USA, Tech. Rep. FHWA-JPO-16-287, 2020.
[34] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: A survey," *Mach. Vis. Appl.*, vol. 25, no. 3, pp. 727–745, 2014.
[35] M. Maile, Q. Chen, G. Brown, and L. Delgrossi, "Intersection collision avoidance: From driver alerts to vehicle control," in *Proc. IEEE 81st Veh. Technol. Conf. (VTC Spring)*, May 2015, pp. 1–5.
[36] H.-S. Seo, D.-G. Noh, C.-J. Lee, and S.-S. Lee, "Design and implementation of intersection movement assistant applications using V2V communications," in *Proc. 5th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, 2013, pp. 49–50.
[37] Y. Feng, "Intelligent traffic control in a connected vehicle environment," Ph.D. dissertation, Dept. Syst. Ind. Eng., Univ. Arizona, Tucson, AZ, USA, 2015.
[38] S. E. Huang, Y. Feng, and H. X. Liu, "A data-driven method for falsified vehicle trajectory identification by anomaly detection," *Transp. Res. C, Emerg. Technol.*, vol. 128, Jul. 2021, Art. no. 103196.
[39] J. Ying and Y. Feng, "Full vehicle trajectory planning model for urban traffic control based on imitation learning," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2676, no. 7, pp. 186–198, Jul. 2022.
[40] B. D. Ziebart, A. L. Maas, J. A. Bagnell, and A. K. Dey, "Maximum entropy inverse reinforcement learning," in *Proc. AAAI*, Chicago, IL, USA, 2008, pp. 1433–1438.
[41] D. Landgrebe, "A survey of decision tree classifier methodology," *IEEE Trans. Syst., Man Cybern.*, vol. 21, no. 3, pp. 660–674, May 1991.
[42] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.
[43] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[44] S. H. Park, B. Kim, C. M. Kang, C. C. Chung, and J. W. Choi, "Sequence-to-sequence prediction of vehicle trajectory via lstm encoder–decoder architecture," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2018, pp. 1672–1678.
[45] P. Perera, P. Oza, and V. M. Patel, "One-class classification: A survey," 2021, *arXiv:2101.03064*.

Zhen Yang received the B.S. degree in automotive engineering from Tsinghua University, China, in 2017, and the M.S. degree in computer science and engineering from the University of Michigan, where he is currently pursuing the Ph.D. degree with the Department of Civil and Environmental Engineering. His research interests include the cooperative automation for the trajectory planning of the autonomous vehicles, vehicle trajectory prediction in complex urban scenarios, cyber security of the autonomous vehicles, integrated control of autonomous vehicle, and traffic signals.

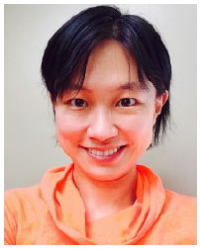
Jun Ying received the master's degree from the Department of Civil Engineering, University of Michigan, in 2020. She is currently pursuing the Ph.D. degree with the Lyles School of Civil Engineering, Purdue University. Her current research interests include cooperative driving automation and transportation system cybersecurity.



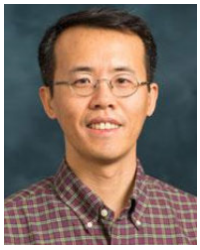
Junjie Shen received the B.E. degree from Hangzhou Dianzi University and the M.S. degree from North Carolina State University. He is currently pursuing the Ph.D. degree with the Department of Computer Science, University of California at Irvine. His current research interests include autonomous driving security with a focus on localization and perception security.

Yiheng Feng received the B.S. and M.E. degrees from the Department of Control Science and Engineering, Zhejiang University, Hangzhou, China, in 2005 and 2007, respectively, and the Ph.D. degree in systems and industrial engineering from The University of Arizona in 2015. He is currently an Assistant Professor with the Lyles School of Civil Engineering, Purdue University. His research interests include traffic signal systems control and security, and CAV testing and evaluation.

Qi Alfred Chen (Member, IEEE) received the Ph.D. degree from the University of Michigan in 2018. He is currently an Assistant Professor with the Department of Computer Science, University of California at Irvine. His research interests include software and AI security, systems security, network security, and security problems at the AI and software stacks in autonomous CPS and the IoT systems, such as autonomous driving and intelligent transportation. He was a recipient of the NSF CAREER Award and the ProQuest Distinguished Dissertation Award at the University of Michigan.



Z. Morley Mao (Fellow, IEEE) received the B.S., M.S., and Ph.D. degrees from the University of California at Berkeley, Berkeley, CA, USA. She is currently a Professor with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor, MI, USA. She was a recipient of the NSF CAREER Award, the Sloan Fellowship, and the IBM Faculty Partnership Award. She has been named as the Morris Wellman Faculty Development Professor.



Henry X. Liu (Member, IEEE) received the bachelor's degree in automotive engineering from Tsinghua University, China, in 1993, and the Ph.D. degree in civil and environment engineering from the University of Wisconsin–Madison in 2000. He is currently a Professor with the Department of Civil and Environmental Engineering and the Director of the Mcity, University of Michigan, Ann Arbor. He is also a Research Professor with the University of Michigan Transportation Research Institute and the Director of the Center for Connected and Automated Transportation (USDOT Region 5 University Transportation Center). His research interests include interface of transportation engineering, automotive engineering, artificial intelligence, traffic flow monitoring, modeling, control, and testing and evaluation of connected and automated vehicles. From August 2017 to August 2019, he served as the DiDi Fellow and the Chief Scientist of smart transportation with DiDi Global Inc., one of the leading mobility service providers in the world. He is the Managing Editor of *Journal of Intelligent Transportation Systems*.