RESEARCH-ARTICLE

Design of Safer Control for Semi-Autonomous Vehicles

CHUNYU CHEN, University of Michigan, Ann Arbor, Ann Arbor, MI, United States

KANG SHIN, University of Michigan, Ann Arbor, Ann Arbor, MI, United States

Published: 20 January 2026
Online AM: 18 November 2025
Accepted: 28 October 2025
Revised: 27 June 2025
Received: 15 September 2024

Design of Safer Control for Semi-Autonomous Vehicles

CHUN-YU CHEN and KANG G. SHIN, CSE/EECS, University of Michigan, Ann Arbor, Michigan, USA

Component faults, bugs, and malicious attacks can all degrade in, or even prevent Semi-Autonomous Vehicles (SAVs) from, correctly capturing their operation context, which is essential to support critical safety features like emergency braking or wheel-steering. While safety features in contemporary SAVs usually rely on static assignment of control priority, such a design may lead to catastrophic accidents when accompanied with erroneous/compromised control and context estimation. To mitigate the grave consequence of using incorrect data, we propose CADCA, a novel control decision-maker for SAVs, that is designed to operate under sensor/data errors or falsifications as well as malicious/erroneous control inputs with the ultimate goal of resolving conflicting control inputs to ensure safety. Our evaluation of >15,700 test-cases has shown CADCA to achieve a 98% success rate in preventing the execution of incorrect control decisions caused by component failures and/or malicious attacks in the most common (i.e., rear-end) collisions.

CCS Concepts: • **Security and privacy → Intrusion/anomaly detection and malware mitigation**;

Additional Key Words and Phrases: Anomaly detection, Accident Prevention

ACM Reference format:
Chun-Yu Chen and Kang G. Shin. 2026. Design of Safer Control for Semi-Autonomous Vehicles. *ACM Trans. Cyber-Phys. Syst.* 10, 1, Article 2 (January 2026), 25 pages.
https://doi.org/10.1145/3777456

1 Introduction

As advanced driver assistance system and self-driving capabilities are getting integrated into commercial and personal vehicles, we have officially entered the era of **Semi-Autonomous Vehicles (SAVs)**, which can be controlled by either autonomous systems (Lv.2+) or human drivers. In addition to reducing fuel consumption, the main reason for adopting autonomous functions in SAVs is to ensure driving safety because drivers are known to be the reason for up to 95% of car accidents [1]. However, autonomous controls in SAVs still have a long way to go as there can never be a perfect human-in-the-loop system that is free of design flaws or software bugs that can be exploited by attackers (e.g., Jeep Cherokee hack [2]). Besides the attacks that directly send malicious control commands to SAVs, sensor spoofing or data manipulation can also compromise SAVs' safety.

In particular, there are three main causes of SAVs' unsafety:

F1. Sensor failures and/or falsifications that generate incorrect inputs to the autonomous system;

The work reported in this article was supported in part by the Office of Naval Research under Grant No. N00014-22-1-2622.
Authors' Contact Information: Chun-Yu Chen (corresponding author), CSE/EECS, University of Michigan, Ann Arbor, Michigan, USA; e-mail: chunyuc@umich.edu; Kang G. Shin, CSE/EECS, University of Michigan, Ann Arbor, Michigan, USA; e-mail: kgshin@umich.edu.

ACM 2378-9638/2026/1-ART2
https://doi.org/10.1145/3777456

F2. Compromised or imperfect controllers or algorithms that generate dangerous controls even with correct input; and
F3. Malicious (unsafe) control inputs from the human driver.

Since neither autonomous nor manual control is perfect, a conflict or disagreement between the two may arise. To resolve such a conflict, state-of-the-art safety features are usually implemented with *static* assignment of control priority under a specific context (i.e., driving condition). For example, an automatic emergency braking can override the driver's control if an object is detected in front of the vehicle [3]. However, this static assignment of priority can lead to catastrophic accidents when accompanied with incorrect sensor readings.

The two crashes of Boeing 737 MAX [4] are the most iconic example of SAV control conflict from which engineers must learn for the design of SAVs. The original 737 MAX's **Maneuvering Characteristics Augmentation System (MCAS)** was designed to prevent stalls from happening caused by human mistakes and was given priority over the pilot's control. However, when the **Angle-of-Attack (AOA)** sensor used by MCAS malfunctions, pilots need to compete with MCAS for the control of the plane's pitch angle. This static priority assignment eventually caused two fatal crashes when the pilots could not disable MCAS completely in time. Boeing's updated MCAS [5] addresses the above problem by using two (instead of only one) AOA sensors for data integrity verification and modifying MCAS to activate only once (i.e., MCAS alone will never override pilots' control). However, this fix returns the control priority back to the pilot, defeating the original purpose of MCAS (i.e., control automation and preventing human errors) and potentially allowing for malicious/erroneous manual control, e.g., the pilot may intentionally or accidentally crash the airplane, like the Germanwings 9525 incident [6].

Since the static assignment of priority to either human or autonomous control has been shown to fail in safety-critical situations and as more autonomous functions are introduced in SAVs, scalability becomes an issue if the engineers need to hard-code priorities to every potential anomalous situation because the bugs triggered by unforeseen/corner conditions are never completely known and taken care of during implementation. Therefore, we must answer the following safety-critical question:

How can an SAV avoid executing unsafe control (F2 and F3) in the case of attacks and/or failures (F1) that feed incorrect input to the SAV itself?

We propose ***C*ontext-*A*ware *D*etection and resolution of *C*ontrol *A*nomalies (CADCA)** as an effective answer to this critical question. As shown in Figure 1, CADCA is designed for use in SAVs equipped with both autonomous and manual control capabilities. CADCA's architecture is designed for SAVs from low-level (Lv.2) to high-level (Lv.4+) autonomy while stressing the compatibility for lower-level architectures. That is, CADCA does not require any change to the existing autonomous control modules—requiring neither hardware upgrade for additional sets of sensor inputs nor additional control output—since hardware cost is a major concern for car-makers. Specifically, CADCA can be considered as a standalone decision-maker (i.e., separate from the autonomous control system) that can be deployed in the ego SAV and aims at identifying the source(s) of anomalies in order to prevent use of anomalous inputs/controls for its maneuver.

There are two characteristics that differentiate SAVs from typical, stand-alone **Cyber-Physical Systems (CPSs)**:

C1. Behavior consistency does not imply safety: Whether a control decision is safe to execute depends on its operation "context," instead of whether an SAV acts consistently with the control or sensor input (e.g., prior anomaly detection schemes [7–9]). CADCA estimates the operation context in real time based on potentially anomalous data to determine if a control decision is safe to execute.

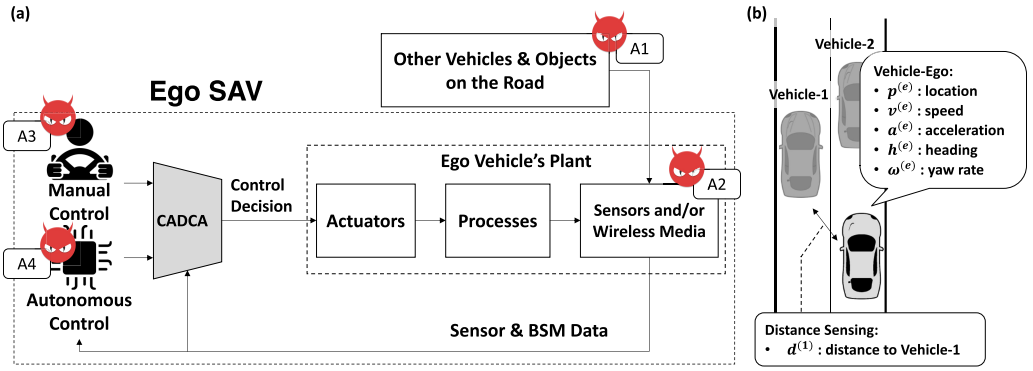


Fig. 1. CADCA's application overview and attack surfaces A1–A4 (more in Section 3). CADCA, Context-Aware Detection and resolution of Control Anomalies.

C2. Limited observability: The autonomous control in an SAV makes decisions based on not only its own measurement/perception but also the data received from other SAVs, for example, via **Basic Safety Messages (BSMs)** [10] or **Vehicle-to-Everything (V2X)** communications [11–16]. Data received from other SAVs do not have the same level of detail as those measured by the SAV itself. Also, the SAV usually does not know other SAVs' initial states and control decisions, which have been assumed by most prior works on estimation-based anomaly detection.

For example, prior studies of robotic vehicles or industrial control systems usually assume that the target system's internal state can be deterministically and uniquely identified by the observed output. However, such an assumption does not hold for SAVs because multiple anomaly scenarios (i.e., faults and attacks) can all lead to the *same* observations due to insufficient input and observations. Also, by treating the detection process as a pure mathematical/optimization problem based on an equation system describing the transitions and causalities between system states and their outputs, they do not account for the feasibility/difficulty of attacks (see Section 2). Furthermore, anomaly detection and risk assessment have commonly been treated as independent/separate problems. Typically, the latter makes implicit assumptions that the input error is bounded, while the former only outputs if there is a data anomaly (i.e., either true or false), missing useful input to the latter.

We, therefore, employ a new design for CADCA that (i) combines anomaly detection with context estimation to eliminate detection uncertainties by identifying the probable anomaly scenario(s) and (ii) restores the thus-identified anomalous data in the corresponding scenario(s). This way, CADCA takes contexts (i.e., the likelihood of the SAV being in certain operation scenarios) into account. As a result, CADCA avoids (1) blindly (thus excessively) consuming computing resources to find all possible (including infeasible) solutions that lead to the same observation, and (2) getting trapped in a single suboptimal/local solution that does not match the real condition.

Specifically, upon acquiring sensor readings of the ego SAV[1] and the (optional) state reports from other entities in the vicinity, CADCA cross-validates the received data to check if any data can be incorrect (meeting F1). If there is any detected inconsistency among the received data, CADCA will restore/correct the anomalous data by constructing one or more *local views*—possible realities that describe the most probable SAV context(s) (e.g., the relative location, speed, and heading of other vehicles). This way, CADCA can capture the conditions that the ego SAV will most likely encounter without restricting itself to only one specific scenario. Finally, CADCA performs risk assessment based on *all* ⟨local view, controls input⟩ combinations, and aggregates their results to select a final

[1]We use *ego* SAV to indicate the SAV that is the focus of the discussion and where CADCA is deployed.

control decision (meeting F2 and F3) based on a safety-first principle while ensuring maximum flexibility of manual control.

This article makes the following contributions:

- A novel control decision-maker, CADCA, that accounts for the difficulty of attack as a context for determining probable operation scenarios, including:
  - An efficient anomaly detection mechanism targeting C1 and C2 (Section 4.1);
  - A mechanism for restoring local views to help the ego SAV comprehend its operation context (Section 4.2);
  - An efficient risk assessment for a given ⟨local view, control input⟩ setting and result aggregation mechanisms under uncertain situations (Sections 4.3); and
- Extensive evaluation of CADCA (with >15,700 test-cases), demonstrating its ability to achieve 98% **Success Rate (SR)** in avoiding use of malicious control inputs and preventing vehicle collisions in the most commonly seen driving scenarios under component failures and/or attacks (Section 6).

2 Related Work and Formulation

(a) *Risk Assessment and Collision Avoidance.* Researchers in the field of robotics and vehicle systems [17–19] have explored ways of assessing risks to prevent collision. Fraichard and Asama [20] proposed the concept of inevitable collision states for the analysis of navigation and motion planning. Brännström et al. [21] proposed a model-based algorithm to estimate how the vehicle can take actions to avoid collision with an object. Kaempchen et al. [22] proposed how to compute the trigger time for an automatic braking system to avoid at least three types of collision scenarios. As one of the latest risk-assessment schemes, Baek et al. [23] proposed the utilization of sensor fusion and inter-vehicle communications for predicting a vehicle's trajectory. As mentioned earlier, the above systems usually purely focus on risk assessment or avoidance and implicitly assume that their inputs can be trusted or within a certain error bound.

(b) *Reliability Enhancement.* We can, in general, enhance vehicle safety by ensuring the system (*O1*) maximizes its performance/reliability if inputs are correct, or (*O2*) still operates correctly even with anomalous inputs. There exist prior works on enhancing the reliability (*O1*) of vehicle control based on the vehicle's operation context, such as expanding drivers' awareness of the surrounding environment [24, 25], or predicting operation context [26]. Selvaraj et al. [27] provided a formal process of verifying system reliability, while Sha [28] proposed a two-layered approach to utilize a high-performance subsystem for common execution with another high-assurance subsystem to ensure system reliability. These works focus on enhancing the effectiveness/reliability of safety features (*O1*) under the assumption that the data can be trusted like risk assessment/avoidance, while CADCA targets a different but common situation where inputs can be incorrect/manipulated (*O2*). Despite its focus on the different aspects of liability/safety, CADCA can integrate the above-mentioned approaches to ensure driving safety, e.g., by adopting the implementation of [28] to protect CADCA's integrity.

(c) *State Estimation and Anomaly Detection.* Considering the existence of anomalous input (*O2*), a typical approach to detecting anomalies in a single (or ego) system is to formulate the target system with the following equation system [29–31]:

$$\begin{cases} x[k+1] = A(x[k]) + B(u[k]) + \xi_p[k] + \mu[k] \\ z[k] = C(x[k]) + \xi_m[k] + \psi[k], \end{cases} \tag{1}$$

where x is the ground-truth system state, "$[\cdot]$" represents discrete time index, z is the observed output, u is the control input/setpoint, and μ and ψ are the process and measurement noises,

respectively. A, B, and C are transformations used to capture the correlation/causality between the variables. ξ_p and ξ_m are the vectors describing the effect of an attack. Given the constraints of bounded process noise μ and measurement noise ψ (i.e., $\mu \leq \bar{\mu}$ and $\psi \leq \bar{\psi}$ always hold for some $\bar{\mu}$ and $\bar{\psi}$), the attack (i.e., ξ's) can then be identified by solving Equation (1). Note x, z, u, μ, ψ, and ξ's can all be vectors, while A, B, and C have appropriate output dimensions.

Utilizing Equation (1) to have a unique, deterministic solution requires the control input u to be known *a priori* and the CPS to have certain observability properties [31]. However, this requirement cannot be met in CADCA's use-case (C2) that involves multiple entities. That is, the ego SAV does not have u in the first equation of Equation (1) for all non-ego SAVs in practice. Also, prior work (e.g., [31]) usually transforms Equation (1) further to an optimization problem and solves it numerically assuming the existence of a unique solution. This will make the process of finding the solution stop/stuck at the first (and maybe incorrect) solution it identifies without searching for other feasible ones.

There have also been approaches tailored for vehicles under multi-entity scenarios [32–35]. They conduct plausibility/consistency checks on their inputs by cross-validating the received data with measurements and motion prediction. Specifically, Jaeger et al. [36] proposed a data verification scheme based on **Kalman Filter (KF)** to predict vehicle locations which reports the detection of an anomaly if the prediction does not match the received data. Bißmeyer et al. [37] utilize the particle-filter to cross-validate vehicles' location with radar readings to track and verify the motion of neighboring vehicles. Stübing et al. [38] proposed a two-stage data verification scheme for position and velocity based on KF and motion prediction with probabilistic maneuver recognition.

While the above schemes focus on detecting data anomalies, they usually consider the data of interest as a single group and do not identify which data are anomalous, nor do they restore them, thus missing one of the most crucial functions required by safety feature. Also, when there is inconsistent information, multiple scenarios may potentially all lead to the same observation by the ego vehicle as mentioned in Section 1. This is also the very reason why a direct combination of prior anomaly detection (even if it has the capability of restoring anomalous data) and risk assessment cannot work well in practice (Section 6). How to perform risk assessment with multiple control inputs without assuming existence of trusted data (F1), despite its importance, has not yet been fully addressed.

3 Deployment and Attack Models

Deployment Model. The ego SAV perceives/receives up to three sources of data: (i) the measurements of its own state, (ii) the (optional) state report from other SAVs, and (iii) its own measurements/observations of other SAVs in its vicinity. We use the following equation system to capture all the observations/data the ego entity perceives/receives:

$$\begin{cases} z_{(e)} = C_{(e)}(x_{(e)}) + \xi_{m(e)} + \psi_{(e)} \\ z_{(j)} = C_{(j)}(x_{(j)}) + \xi_{m(j)} + \psi_{(j)} \\ z_{(e,j)} = C_{(e,j)}(x_{(e)}, x_{(j)}) + \xi_{m(e,j)} + \psi_{(e,j)}, \end{cases} \tag{2}$$

where x, z, ψ, C, and ξ_m's are the same as in Equation (1), while e and j represent the ego entity and non-ego entity with index j, respectively, and the subscripts (e), (j), and (e,j) correspond to the three data sources mentioned above. We also regard a measurement as anomalous if the measurement noise ψ is greater than a pre-specified error bound $\bar{\psi}$.

The default SAV states (x's) and measurements (z's) considered in CADCA are vehicle location (p), speed (v), acceleration (a), heading (h), and yaw rate (ω), as shown in Figure 1(b). The ego SAV is also capable of making measurements with its own sensors, such as RADAR/LIDAR/cameras, to obtain

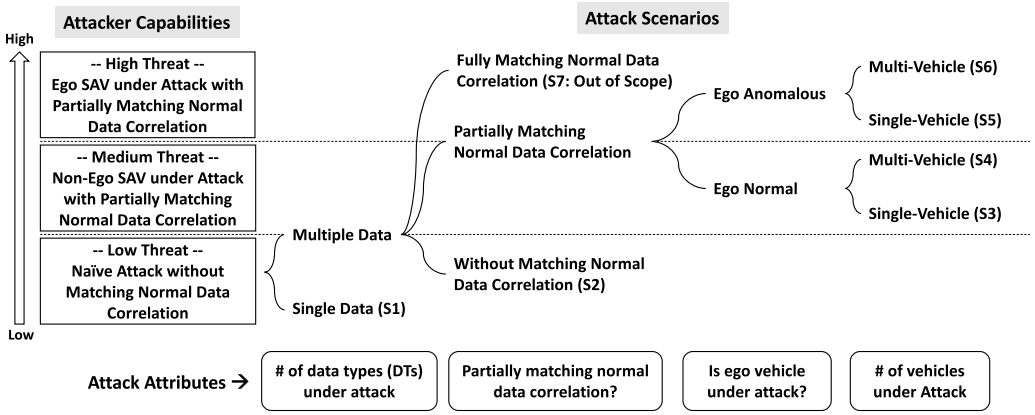


Fig. 2. Attack classification and attributes.

the distances between itself and other SAVs. Since x's, ξ's, and ψ's are all unknown to the ego SAV, Equation (2) does *not* necessarily have a unique solution under attacks. That is, solving Equation (2) to deterministically and uniquely determine the true state (i.e., x) of each SAV may not be possible as there can be multiple solutions that all lead to the same observations as in Equation (2).

To facilitate deployability, the information of non-ego vehicles utilized in CADCA is limited to the vehicles' dynamics measurements—either from the sensors of ego SAV such as RADAR/LIDAR/cameras or BSMs from other SAVs [10]—including location (p), speed (v), acceleration (a), yaw rate (ω), and heading (h). That is, CADCA is designed to run directly on a single vehicle (i.e., the *ego vehicle*) without requiring cooperation (i.e., additional, non-standardized computation, and message exchange for obtaining results) from other vehicles.

Attack Model. Besides malicious/erroneous driver control and erroneous autonomous control algorithms (A3 and A4 in Figure 1), we assume that the adversary has the goal of causing collision to the ego SAV by making the ego SAV incorrectly estimate its operation context (e.g., assuming an incorrect distance/speed w.r.t. nearby vehicles) and further misleading the autonomous control of the ego SAV. The attacker can (i) transmit incorrect data from non-ego vehicles (A1 in Figure 1) and/or (ii) remotely spoof the sensors of the ego vehicle (A2 in Figure 1). To better understand the problem space, we propose an attack-space classification for SAVs, decomposing the entire attack space into three tiers/levels according to the adversary's capabilities (Figure 2). These threat levels are determined by the attack attributes, i.e., classifiers, that are directly linked to the level of difficulty and effort for the adversary to launch attacks. To avoid confusion with the value of a single data sample, we use the term **"data type" (DT)** to denote a specific type of data from a specific vehicle (e.g., Vehicle-1's location and Vehicle-Ego's speed).

The lowest threat level includes S1 and S2, where the (naive) adversary can only manipulate one DT or manipulate multiple DTs but s/he cannot manipulate the DTs in a way that matches the normal data correlation (e.g., the manipulated location and speed do not match each other), where an attack is said to match normal data correlation if the data under the attack show the same correlation/causality in terms of the kinematics of rigid bodies and control-to-dynamics responses as those before the attack (within a predefined error tolerance). In the medium threat level (S3–S4), the adversary can launch an attack or spoof DTs from one or more non-ego vehicles and the anomalous DT from that entity may be consecutively manipulated to *partially* match the normal data correlation. For example, the adversary can manipulate the location and speed in a BSM simultaneously so that the vehicle's moving distance matches the manipulated speed. In the

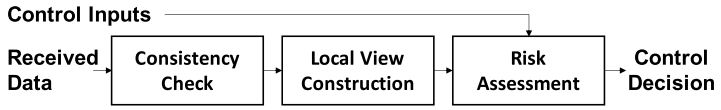


Fig. 3. An overview of CADCA's operation. CADCA, Context-Aware Detection and resolution of Control Anomalies.

highest threat level (S5–S6), the adversaries can expand their target to the ego vehicle's DTs in addition to other non-ego vehicles'.

Since no data can be trusted entirely (i.e., a strong attack model), CADCA is designed mainly for guaranteed defense against the low- and medium-level threats while providing less-than-complete protection against high-level threats. That is, CADCA is *not* designed to operate when the data from multiple entities are *simultaneously* manipulated to achieve a perfect match with their normal correlation/causality (S7). This detection scope should not diminish CADCA's value and practicality. Specifically, unless the adversary has the complete control over non-ego vehicles' BSM transmissions, simultaneously spoofing all of their DTs (other than the location/GPS) to generate specific values without the direct/physical access to all the sensors is proven very difficult, if not impossible. For example, spoofing an MEMS-based accelerometer requires direct feedback from the sensor for phase tuning to generate a specific waveform/value [39]. That is, even though an adversary may have the capability described in the high threat level, these types of attack will not be as scalable as the attacks in the low/medium threat level as the adversary needs to physically compromise multiple vehicles.

As CADCA is designed under a strong (and harsh) fault/attack assumption (i.e., no data can be trusted as mentioned previously), this assumption provides the most general deployment scenario without requiring the existence of trusted measurement and/or entities. At a first glance, this assumption may seem to be a drawback of CADCA, but it enables CADCA to have the most powerful deployment scenario without requiring the existence of a trusted entity.

Furthermore, since CADCA does not require any external/cooperative help/computation in generating a decision, it can be deployed in SAVs as an internal module shielded from the SAV plant that may be exposed/connected to an external-facing communication module (e.g., Wi-Fi) as shown in Figure 1. So, CADCA's execution integrity can be protected by digital signatures together with memory integrity verification since its input can be confined to pure sensor readings/BSM/control decisions from other internal components. See Section 7 for more discussion on how to extend CADCA to a relaxed threat model.

4 System Design

CADCA follows the design concept shown in Figure 3. First, it cross-validates the data acquired by the ego SAV itself and those received from other entities to detect if there is any inconsistency indicating the existence of a data anomaly or an attack. Second, upon detection of an anomaly, CADCA will construct local views to establish models of its surrounding environment, each of which describes a probable scenario when different combinations of data are anomalous but can lead to the same observed inconsistency. The above two steps are equivalent to first identifying the attack factor ξ's in Equation (2) to eliminate the uncertainties in the equation system and then restoring the identified anomalous data based on the remaining correct ones under certain attack scenarios. Finally, CADCA conducts risk assessments on all probable ⟨control input, local view⟩ combinations, and then aggregates the assessment results to select a final control decision.

Table 1. This Table Shows Examples of Detection Sets of the Ego Vehicle and Vehicle-1 (Figure 1(b)), Where $d_{(e,1)}$ is the Distance between the Two Vehicles Measured by the Ego Vehicle

DS	Ego Dynamics $z_{(e)}$					Dist.	Vehicle-1 Dynamics $z_{(1)}$				
	p	v	a	ω	h	$d_{(e,1)}$	p	v	a	ω	h
E,1	✓	✓	✓	✓	✓	$DS_{\cdot,1}$:					
E,2	✓	✓				$P_{k+1} = P_k + \int \left(V_k + \int a_k H(\tau) d\tau\right) dt$					
E,3		✓	✓			integrate $\tau = 0 \rightarrow t$ and $t = 0 \rightarrow \Delta t$,					
E,4				✓	✓	$H(\tau)$ = unit vector of $\angle(h_k + \omega\tau)$					
E,5	✓				✓	and P (V) is the vector form of p (v).					
V1,1	Δt: time interval, k: time index						✓	✓	✓	✓	✓
V1,2	$DS_{\cdot,2}$: $v_k = \lvert P_{k+1} - P_k \rvert / \Delta t$						✓	✓			
V1,3	$DS_{\cdot,3}$: $v_{k+1} = v_k + a_k \Delta t$							✓	✓		
V1,4	$DS_{\cdot,4}$: $h_{k+1} = h_k + \omega_k \Delta t$									✓	✓
V1,5	$DS_{\cdot,5}$: h_k = direction of p_k to p_{k+1}						✓				✓
V1,6	✓					✓	✓	$d_{(e,1)} = \lvert P_{(e)} - P_{(1)} \rvert$			

4.1 Consistency Check

CADCA's first function block is the consistency check, determining if there are any inconsistencies among the received/perceived data by cross-validating the correlation between them. While utilizing physical invariants for detecting inconsistencies is not new, CADCA's novelty lies in the consideration of input data in *small and overlapping groups*, instead of a single large group, and the transformation of the result of each group's consistency-check into a corresponding equation in an equation system; solutions of the equation system will directly indicate the potentially anomalous data. We call such data groups ***Detection Sets* (DSs)**.

Table 1 shows an example of DSs when there are only two vehicles (i.e., the ego vehicle and Vehicle-1). Specifically, $DS_{E,1}$–$DS_{E,5}$ ($DS_{V1,1}$–$DS_{V1,5}$) capture the correlation between the dynamics measurements of the ego vehicle (Vehicle-1), and $DS_{V1,6}$ describes the correlation between the distance measurement $d_{(e,1)}$ and the locations of the two vehicles. If there is another vehicle, say Vehicle-2, the ego vehicle can have six more observations ($DS_{V2,1}$–$DS_{V2,6}$) just like $DS_{V1,1}$–$DS_{V1,6}$. The DSs shown here are the default formulations derived from the kinematics of rigid bodies [40].

A DS, DS_l, will have the following general form:

$$\hat{z}'[k'] = \mathscr{F}\left(z[k], z[k+1]\right), \tag{3}$$

where $\mathscr{F}$ is the function describing the correlation between $\hat{z}'[k']$, the prediction of a data $z' \in DS_l$ with timestamp index $k' \in \{k, k+1\}$, and other observations (i.e., $z[k]$ and/or $z[k+1]$, where the latter is optional in the formulation). Note the correlation function $\mathscr{F}$ can be additional observers designed by the engineers, or it can be directly derived from the C's in Equation (2). For example, $DS_{V1,6}$ checks whether the distance between the ego vehicle and Vehicle-1 matches the received location data and has the form of $\hat{d}_{(e,1)} = \mathscr{F}(P_{(e)}, P_{(1)}) = |P_{(1)} - P_{(e)}|$, where "$|\cdot|$" denotes the Euclidean distance, P is vector form of location p, and $\mathscr{F}$ equals $C_{(e,j)}$ when we have $z_{(e,j)} = d_{(e,j)}$ in Equation (2).

Definition 4.1. A DS is properly designed if $z'[k'] = \hat{z}'[k'] = \mathscr{F}\left(z[k], z[k+1]\right)$ is always true when there is no measurement noise or an attack in z' and z.

During runtime, CADCA will use the properly designed correlation function $\mathscr{F}$ in Equation (3) to compute data prediction $\hat{z'}[k']$ and compare it with the received value $z'[k']$. Specifically, if $|\hat{z'}[k'] - z'[k']| > \Gamma_{z'}$ where $\Gamma_{z'}$ is a detection threshold, CADCA will report an anomaly in the DS ($DS_l \rightarrow \boldsymbol{\times}$); otherwise, CADCA will report the DS to have passed the consistency check ($DS_l \rightarrow \checkmark$). $\Gamma_{z'}$'s should be set to values not larger than common error bounds to avoid miss detection for the default values. While small thresholds may cause false positives in the consistency-check phase, it will not influence CADCA much as the anomalous data will be restored in the subsequent phase. CADCA will then use the results from the consistency check to establish an equation system $\mathscr{E}$, and its solution will tell which data may be anomalous.

Each DS, say DS_l, that fails the consistency check will yield an equation:

$$\sum_{\forall z_i \in DS_l} I(z_i) \geq 1, \tag{4}$$

where $I(\cdot)$ is the indicator function describing whether a received data z_i is anomalous (=1) or not (=0). This equation indicates that whenever a DS fails the consistency check, there must be at least one anomalous data within the DS (which may be due to an attack, a fault, or excessive measurement noise) as $\sum_{\forall z_i \in DS_l} I(z_i)$ captures the number of DTs that may be anomalous. However, $DS_l \rightarrow \checkmark$ (i.e., DS_l passes consistency check) does not necessarily mean no anomalous data in DS_l, i.e., it may be a false-negative detection.

Property 4.1. *For every solution to the equation system ($\mathscr{E}$) constructed by Equation (4) under properly designed DSs, there exists at least one corresponding solution for Equation (2).*

Proof. Assume no valid solution for Equation (2). $\Rightarrow$ There always exists some $DS_l \rightarrow \boldsymbol{\times}$ s.t. $z_i[k] = C_i(x[k]), \forall z_i \in DS_l$. However, the definition of properly designed DSs tells us that DS_l will pass the consistency check since all its data do not have measurement noise or attack. This contradicts $DS_l \rightarrow \boldsymbol{\times}$. So, the assumption must be false. □

4.2 Construction of Local Views

(1) *Concept.* A local view, as the name suggests, is what the ego SAV thinks or estimates its current operation context to be (e.g., the status of the ego SAV itself and the surrounding SAVs). In an ideal scenario without component failures or attacks, the ego vehicle should be able to construct a perfect local view that matches the ground truth (with the bounded deviation caused by measurement noises). However, if the ego SAV receives/perceives anomalous data, there may exist some inconsistencies in the data received or locally perceived that may prevent the ego SAV from constructing a correct local view. The basic idea of Local View Construction is to (i) identify potentially anomalous data and (ii) correct the anomalous data, if any, based on the normal data.

To identify the anomalous data that need to be restored, CADCA will solve the equation system ($\mathscr{E}$) to identify the data z_i with $I(z_i) = 1$. Since Equation (2) may not yield a unique solution, there can be multiple solutions that satisfy $\mathscr{E}$ even in a perfect detection scenario (i.e., with neither false positives nor false negatives during the consistency-check phase). Therefore, CADCA constructs multiple local views each of which corresponds to one valid solution to the equation system $\mathscr{E}$ that represents a specific failure/attack scenario. Note that solving the equation system $\mathscr{E}$ is equivalent to the NP-Complete fitting set problem while assuming a perfect consistency-check phase while considering all probable false-negative/positive situations will be equivalent to solving multiple NP-Complete problems. Also, since not all the solutions have the same level of feasibility given CADCA's application context, we have designed six (heuristic) algorithms to solve $\mathscr{E}$ for constructing

Algorithm 1: Solution-Space Algorithm

1 **Function:** Solution_Space ($\mathscr{E}$);
Input : The observer equation system ($\mathscr{E} = \langle Z, Y \rangle$), where $Z = (DS_1, DS_2, \ldots)$ is the definition of each DS in the equation system and $Y = (y_1, y_2, \ldots)$ is the consistency-check result.
Output : A list of anomalous data ($\mathcal{L}$).
2 **Set** $N \leftarrow$ total number of DSs;
3 **Set** $K \leftarrow$ total number of DTs;
4 **Set** $\mathcal{L} \leftarrow \{\}$; `// Set` $\mathcal{L}$ `as an empty list.`
5 **Initialize** $arr1[N][K]$ to FALSE;
6 **Initialize** $arr2[K]$ to FALSE;
7 **for** i **As Integer** = 1 **UpTo** N **do**
8 **if** y_i == "✓" **then**
9 **Set** $arr1[i][j] \leftarrow$ TRUE, $\forall z_j \in DS_i$;
10 **end**
11 **end**
12 **for** j **As Integer** = 1 **UpTo** K **do**
13 **Set** $arr2[j] \leftarrow \bigvee_{i=1}^{N} arr1[i][j]$; `// Column-wise "OR" operation to` $arr1$
14 **if** $arr2[j]$ **equals** FALSE **then**
15 **Set** $\mathcal{L} \leftarrow \mathcal{L} \bigcup \{z_j\}$;
16 **end**
17 **end**
18 **for** i **As Integer** = 1 **UpTo** N **do**
19 **if** $arr2[i][j]$ == TRUE, $\forall z_j \in DS_i$ and y_i == "✓" **then**
20 **Set** $\mathcal{L} \leftarrow \mathcal{L} \bigcup DS_i$;
21 **end**
22 **end**
23 **return** $\mathcal{L}$;

local views (instead of blindly finding all probable solutions for $\mathscr{E}$). Next, we provide an overview of these six algorithms.

The Solution-Space algorithm (Section 4.2.2) is designed to capture the search space of anomalous DTs and provide the other five algorithms with a list of potential anomalous DTs as a starting point for identifying the real anomalous DTs. The Greedy (Section 4.2.3) and Anomalous-Individual algorithms (Section 4.2.4) are designed to capture solutions for $\mathscr{E}$ that have the minimum number of anomalous DTs and vehicles, respectively. The Trust-Ego (Section 4.2.5), Anomalous-Ego-Only (Section 4.2.6), and Anomalous-Ego-and-Others (Section 4.2.7) algorithms are designed to capture the special scenarios related to the normality of the ego vehicle (i.e., S3–S6). While the six algorithms do not have an obvious mapping associated with the six attack scenarios, they do collectively cover S1–S6 defined in Figure 2, which will be introduced later.

(2) *Solution-Space Algorithm* (Algorithm 1) is designed to capture the search space of anomalous DT(s). First, it temporarily assumes no false negatives in the consistency check and considers a DT to be normal as long as it is included in a DS that passed the consistency check (Lines 7–11 in Algorithm 1). Second, those data in the DSs that fail the consistency check (denoted by *anomalous DSs*) but not ruled out by the first step will be considered potentially anomalous (Lines 12–17). Third, the Solution-Space algorithm will perform a sanity check (Lines 18–22) to see if all anomalous DSs contain at least one anomalous data. If there is an anomalous DS that does not satisfy the above condition, then the Solution-Space algorithm will conclude the existence of a false negative in the

Algorithm 2: Greedy Algorithm

```
1  Function: Greedy(𝒞, ℒ);
   Input  : The observer equation system (𝒞) and ℒ = Solution_Space(𝒞).
   Output : A list of anomalous DTs (ℒ_𝒢).
2  Set ℒ_𝒢 ← {};
3  Set ℒ_𝒟𝒮 ← {};
4  for all z_j ∈ ℒ do
5  |   ℒ_𝒟𝒮 ← ℒ_𝒟𝒮 ⋃{i}, ∀DS_i ∋ z_j;
6  end
7  while ℒ_𝒟𝒮 ! = {} do
8  |   z_t ← the DT z_j ∈ ℒ that covers the most number of DSs in ℒ_𝒟𝒮;
9  |   ℒ ← ℒ \ {z_t};
10 |   ℒ_𝒟𝒮 ← ℒ_𝒟𝒮 \ {i}, ∀DS_i ∋ z_t;
11 |   ℒ_𝒢 ← ℒ_𝒢 ⋃{z_t};
12 end
13 return ℒ_𝒢;
```

consistency check and consider all the data in that anomalous DS to be potentially anomalous. We call this step as *normality inversion*.

Property 4.2. *(Correctness) $\mathcal{L}$ is a solution space of the equation system $\mathscr{E}$ (i.e., there must exist a list of anomalous data $\mathcal{L}_S \subseteq \mathcal{L}$ that is a solution to $\mathscr{E}$), where $\mathscr{E}$ and $\mathcal{L}$ are, respectively, the input and output defined in Algorithm 1.*

Proof. Assume $\mathcal{L}$ is not the solution space of $\mathscr{E}$, indicating at least one equation in $\mathscr{E}$ cannot be covered by $\mathcal{L}$. However, the sanity check (Lines 18–22) of the Solution-Space algorithm ensures every equation (or anomalous DS) in $\mathscr{E}$ will be covered by $\mathcal{L}$. Therefore, $\mathcal{L}$ must not be the output of the Solution-Space algorithm. □

(3) *Greedy Algorithm* (Algorithm 2) is designed to find a solution to $\mathscr{E}$ that requires as few anomalous DTs as possible (i.e., $\arg\min_{I(z_i)} \sum I(z_i)$ given Equation (4) as the constraint) in a greedy way. Specifically, it will identify the anomalous DTs starting from the list obtained from the Solution-Space algorithm (i.e., $\mathcal{L}$) and pick the anomalous DT z_t that covers the most number of anomalous DSs. A DS_i is said to be *anomalous* if $\exists\ z_t$ s.t. $z_t \in (\mathcal{L} \cap DS_i)$. The Greedy algorithm will then remove all the DSs in $\cup_{\forall DS_i \ni z_t} i$ from $\mathcal{L}_{\mathcal{DS}}$ and remove z_t from $\mathcal{L}$. Next, it will continue to pick the DT in $\mathcal{L}$ that covers the most remaining anomalous DSs and repeat the process until all the anomalous DSs are covered.

(4) *Anomalous-Individual Algorithm* (Algorithm 3) is designed to capture a solution to $\mathscr{E}$ that requires the attacker to compromise the least number of vehicles. Unlike the Greedy algorithm that directly selects the data covering the most number of anomalous DSs, the Anomalous-Individual algorithm will first identify the vehicle that covers the most number of anomalous DSs and follows the same concept of the Greedy algorithm to identify the anomalous data associated with that particular vehicle first. Next, the Anomalous-Individual algorithm will target the vehicle that covers the most number of anomalous DSs, excluding those already covered, and repeat the process.

(5) *Trust-Ego Algorithm* (Algorithm 4) is designed to capture the situation in which no anomalous data originated from the ego vehicle. The algorithm also starts from the results obtained from the Solution-Space algorithm, but it will only try to construct a list ($\mathcal{L}_{TE}$) of anomalous DTs from

Algorithm 3: Anomalous-Individual Algorithm

Function: Anomalous_Individual($\mathscr{E}$, $\mathcal{L}$);
Input : The observed equation system ($\mathscr{E}$) and $\mathcal{L}$ = Solution_Space($\mathscr{E}$).
Output : A list of anomalous DTs ($\mathcal{L}_{AI}$).
Set $\mathcal{L}_V \leftarrow \{V_0, V_1, ..., V_M\}$; `// A list of all SAVs, where` V_0
Set $\mathcal{L}_{AI} \leftarrow \{\}$; `// is the ego SAV and` M `is the number of`
Set $\mathcal{L}_{\mathcal{DS}} \leftarrow \{\}$; `// non-ego SAVs.`
for all $z_j \in \mathcal{L}$ **do**
  $\mathcal{L}_{\mathcal{DS}} \leftarrow \mathcal{L}_{\mathcal{DS}} \bigcup \{i\}, \forall DS_i \ni z_j$;
end
while $\mathcal{L}_{\mathcal{DS}}$! = {} **do**
  $V_t \leftarrow$ the vehicle $V_j \in \mathcal{L}_V$ that covers the most DSs in $\mathcal{L}_{\mathcal{DS}}$;
  while $\exists\, z_j \in \mathcal{L}$ and z_j belongs to V_t **do**
    $z_t \leftarrow$ the DT $z_j \in \mathcal{L}$ that covers the most DSs in $\mathcal{L}_{\mathcal{DS}}$ and z_j belongs to V_t;
    $\mathcal{L} \leftarrow \mathcal{L} \setminus \{z_t\}$;
    $\mathcal{L}_{\mathcal{DS}} \leftarrow \mathcal{L}_{\mathcal{DS}} \setminus \{i\}, \forall DS_i \ni z_t$;
    $\mathcal{L}_{AI} \leftarrow \mathcal{L}_{AI} \bigcup \{z_t\}$;
    if z_j does not belong to V_t, $\forall z_j \in \mathcal{L}$ **then**
      break;
    end
  end
  $\mathcal{L}_V \leftarrow \mathcal{L}_V \setminus \{V_t\}$;
end
return $\mathcal{L}_{AI}$;

Algorithm 4: Trust-Ego Algorithm

Function: Trust_Ego($\mathscr{E}$, $\mathcal{L}$);
Input : The observer equation system ($\mathscr{E}$) and $\mathcal{L}$ = Solution_Space($\mathscr{E}$).
Output : A list of anomalous DTs ($\mathcal{L}_{TE}$).
Set $\mathcal{L}_{TE} \leftarrow \{\}$;
Set $\mathcal{L}_{\mathcal{DS}} \leftarrow \{\}$;
for all $z_j \in \mathcal{L}$ **do**
  $\mathcal{L}_{\mathcal{DS}} \leftarrow \mathcal{L}_{\mathcal{DS}} \bigcup \{i\}, \forall DS_i \ni z_j$;
end
while $\mathcal{L}_{\mathcal{DS}}$! = {} **do**
  $z_t \leftarrow$ the DT $z_j \in \mathcal{L}$ that covers the most DSs in $\mathcal{L}_{\mathcal{DS}}$ and z_j does not belong to ego vehicle;
  $\mathcal{L} \leftarrow \mathcal{L} \setminus \{z_t\}$;
  $\mathcal{L}_{\mathcal{DS}} \leftarrow \mathcal{L}_{\mathcal{DS}} \setminus \{i\}, \forall DS_i \ni z_t$;
  $\mathcal{L}_{TE} \leftarrow \mathcal{L}_{TE} \bigcup \{z_t\}$;
  if $\mathcal{L}_{\mathcal{DS}}$! = {} and z_j belongs to ego vehicle, $\forall z_j \in \mathcal{L}$ **then**
    return NULL;
  end
end
return $\mathcal{L}_{TE}$;

non-ego vehicles' DTs by following the same procedure as the one in the Greedy algorithm. Note that the Trust-Ego algorithm may not find a valid solution to $\mathscr{E}$.

(6) *Anomalous-Ego-Only Algorithm* is designed to capture the condition where only the ego vehicle is anomalous. Opposite to the Trust-Ego algorithm, the Anomalous-Ego-Only algorithm tries to find the solution that satisfies the observed equation system assuming that the anomalous data can only originate from the ego vehicle. Like the Trust-Ego algorithm, the Anomalous-Ego-Only algorithm may not have a valid solution.

(7) *Anomalous-Ego-and-Others **(AEnO)** Algorithm* is designed to capture the situation in which the ego vehicle and at least one non-ego SAV are anomalous. AEnO follows the same initial procedure as Anomalous-Ego-Only that focuses on the ego SAV's DTs first and then moves on to examine the non-ego vehicles' DTs.

(8) *Restoration of Anomalous Data.* The final step in Local View Construction is to restore (or estimate) data identified as anomalous and construct local views for each algorithm's output. Specifically, CADCA will directly utilize the data correlation captured in the DSs for restoring anomalous data if enough correct data are observed. Otherwise, CADCA will use the last (set of) data determined to be correct to build the local view.

4.3 Risk Assessment

Concept. The process of risk assessment will be triggered whenever an anomaly is detected or CADCA receives more than one control input, and the result of risk assessment will be updated whenever the ego vehicle receives new data. Its goal is to identify whether there exists any safety risk (e.g., possible collision) associated with the control inputs and helps the SAV choose which of the control inputs to accept and execute. CADCA inherently accounts for operation in an uncertain situation even when there are multiple probable operation contexts that can lead to the same observed sensor inputs. While CADCA generates a local view for each of the algorithm's result, it also performs risk assessment for every ⟨local view, control input⟩ pair.

CADCA then determines which control input is safe to execute based on the following (safety-first) rules:

—*Rule-1*: If there is no safety concern in any of the combinations, execute the manual control.
—*Rule-2*: If only one control input has a safety concern (in any of the probable local views), execute the other safe one.
—*Rule-3*: If both control inputs have safety concerns, select the control input with a longer Time-to-Collision (TTC) (i.e., less unsafe) under the most likely local view (see below).

Specifically, CADCA's local view construction is equivalent to identifying the potential states of the ego SAV in the state space and CADCA's risk assessment is equivalent to determining which control input can potentially lead the ego SAV to the unstable/unsafe state. CADCA will try to avoid any potential safety risk if possible; otherwise, it will try to delay the occurrence of a safety-critical event as much as possible according to the most likely condition.

CADCA's design is not restricted to any specific metrics or types of risks as long as the risk can be clearly defined. As vehicle collision is the most obvious safety risk, here we use collision avoidance as a concrete example to illustrate CADCA's risk assessment process. In this example, the risk of collision could be defined by the time left for a driver to react to an urgent condition, and CADCA may perform risk assessments of potential vehicle collision for each control input by checking whether TTC $\leq$ a threshold $T_C = \max(T_S, T_U + T_R)$, where T_S(=4.5 s) is determined by the (medium) driver's reaction time while ensuring a smooth transition of control (e.g., no sudden braking all the time) [41], T_U is the minimum time required for the ego vehicle to avoid the collision through speed control, and T_R(=2.6 s) is the driver's reaction time in an urgent situation [42].

Estimation of Time to Avoid Collision (T_U). T_U is computed based on (i) the relative speed $v'_{(\eta)} = v_{(\eta)} - v_{(e)}$ between the ego and the target vehicle η, (ii) target vehicle η's acceleration $a_{(\eta)}$,

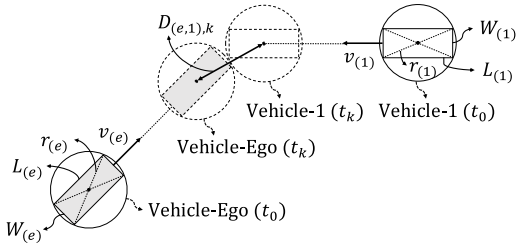


Fig. 4. An example of computing time-to-collision, where $D_{(e,1),k}$ is the distance between the centers of the two vehicle circles at time t_k.

(iii) the ego vehicle's maximum acceleration $a^+_{(e)}$ and deceleration $a^-_{(e)}$ capability, and (iv) the relative location $p'_{(\eta)}$ of η with respect to ego vehicle's travelling direction:

$$T_U = \begin{cases} |v'_{(\eta)}|/(|a^-_{(e)}| + a_{(\eta)}), \text{ if } v'_{(\eta)}, p'_{(\eta)} < 0; \\ |v'_{(\eta)}|/(|a^+_{(e)}| - a_{(\eta)}), \text{ if } v'_{(\eta)}, p'_{(\eta)} > 0; \\ 0, \text{ otherwise.} \end{cases} \tag{5}$$

Note that Equation (5) does not consider every possible combination of $\langle v'_{(\eta)}, a_{(\eta)}, p'_{(\eta)} \rangle$ because computation of T_U is required only when there is a possibility of collision.

Estimation of TTC. To reduce the computation workload and account for the possibility that the ego vehicle may not have the accurate information of (other) vehicles' physical dimensions (i.e., lengths and widths), CADCA treats each vehicle as circles as in [23], instead of rectangles, in a 2D plane and the diameters the circles are required to cover the vehicle body. Figure 4 shows an example where the circle of the vehicles can just cover the vehicle body, i.e., $(2r_{(e)})^2 = L^2_{(e)} + W^2_{(e)}$ and $(2r_{(1)})^2 = L^2_{(1)} + W^2_{(1)}$. Note that the vehicles' dimensions can be obtained from BSM Part-I or estimated from the ego vehicle's LIDAR or camera sensor readings.

CADCA will then predict the future trajectory of vehicles using the following equations:

$$p^{(X)}_{k+1} = p^{(X)}_k + \frac{v_k}{\omega_k}[\sin(h'_k) - \sin(h_k)] + \frac{a_k \Delta t}{\omega_k}\sin(h'_k) - \frac{a_k}{\omega^2_k}[\cos(h_k) - \cos(h'_k)], \tag{6}$$

$$p^{(Y)}_{k+1} = p^{(Y)}_k + \frac{v_k}{\omega_k}[\cos(h_k) - \cos(h'_k)] - \frac{a_k \Delta t}{\omega_k}\cos(h'_k) - \frac{a}{\omega^2_k}[\sin(h_k) - \sin(h'_k)], \tag{7}$$

$$v_{k+1} = v_k + a_k \Delta t, \tag{8}$$

$$h_{k+1} = h'_k = h_k + \omega_k \Delta t, \tag{9}$$

where subscripts k and $k+1$ are the timestamp indices of data, and $p^{(X)}$ and $p^{(Y)}$ represent the vehicle's X (east–west) coordinate and Y (north–south) coordinate, respectively. Finally, CADCA can determine TTC by checking the timing when the distance between the ego vehicle and another vehicle is less than the sum of their circles' radii (e.g., $D_{(e,1),k} < r_{(e)} + r_{(1)}$ in Figure 4). Note that CADCA updates TTC estimation upon receipt of new data.

Most Likely Local View. CADCA determines the local view that is most likely to capture the actual situation based on a record of whether a vehicle has been determined to be potentially anomalous before. Table 2 shows an example of this history record. Specifically, each row of the record shows if a vehicle is determined to be potentially anomalous (normal) at some time t_k, and the entry will be marked with 1 (0) under the t_k column (with frequency of 10 Hz, same as BSM). CADCA will consider a non-ego vehicle to be potentially anomalous if any algorithm except for the Solution-Space

Table 2. An Example of Vehicle Anomaly History, Where $\text{WAC}_{k,i}$ Is the WAC of the Vehicle i at Time t_k

Vehicle	t_1	t_2	t_3	t_4	t_5	t_6	t_7	...	$\text{WAC}_{k,i}$
Ego	0	0	0	1	1	1	1	...	0.5
Vehicle-1	-	-	-	0	1	1	0	...	0.1
Vehicle-2	1	1	1	1	1	1	1	...	1.0
...					...				

algorithm determines the vehicle to be anomalous. In contrast, the ego vehicle will be determined to be anomalous if the Trust-Ego algorithm cannot find a solution and will be determined to be normal if AEnO algorithm cannot find a solution. Otherwise, its integrity will be determined by the Greedy algorithm. The rationale behind treating the ego vehicle differently than others is: (i) the observed equation system is centered around the ego vehicle and (ii) Trust-Ego, Anomalous-Ego-Only and AEnO algorithms are designed to target special cases of the ego vehicle.

To identify the most likely scenario, CADCA summarizes the history of each vehicle and uses a scalar value to represent the likelihood of each vehicle being anomalous. Since this value is computed based on a weighted sum of anomaly counts, we call it ***Weighted Anomaly Count* (WAC)**:

$$\text{WAC} = \min(W_A + W_R, 1) \in [0, 1],$$

where $W_A \in [0, 1]$ is computed over all the records of indicating a vehicle's overall credibility and W_R is the additional factor/weight used to account for the recency of anomaly detection. Specifically, $W_A \in [0, 1]$ is the ratio of the number of anomalies (N_A) to the total number of records (N_T). For example, since the ego vehicle in Table 2 has seven records, four of which are marked to be anomalous (=1), W_A of the ego vehicle will be $4/7 \approx 0.57$. To avoid unstable W_A when a vehicle just enters the ego vehicle's communication range, we design CADCA to compute W_A normally if there are $\geq N_{min}$ records; otherwise, CADCA will fill in the "missing" records with 1 (i.e., assuming the vehicle cannot be trusted initially):

$$W_A = \begin{cases} (N_A + N_{min} - N_T)/N_{min}, \text{ if } N_T < N_{min}; \\ N_A/N_T, \text{ if } N_T \geq N_{min}. \end{cases}$$

W_R controls the influence of most recent N_R records:

$$W_R = (1 - \alpha^{N_{A,R}})/(1 - \alpha^{N_R}),$$

where $N_{A,R}$ is the number of anomaly reports within the latest N_R records and α is the parameter designed to adjust how fast W_R should increase if an anomaly report is received. This design ensures that W_R will (i) increase while more and more "1" records are received (ii) gradually (but not drastically) decrease while the vehicle is determined to become normal again before N_R number of "0" records are received.

The last step of identifying the most likely local view is to find a solution obtained from the algorithms that has the best match with the WACs (i.e., the observed history). Specifically, we can present the WACs as a vector $\vec{W}_k = (\text{WAC}_{k,e}, \text{WAC}_{k,V1}, \ldots)$, where $\text{WAC}_{k,i}$ is the WAC of vehicle i at time t_k, and this vector represents a snapshot of the vehicles' normality perceived by the ego vehicle. Similarly, we can present the solutions/results from the algorithms in a vector form $\vec{\phi}_{k,\mathscr{A}} = (b_{(e)}, b_{(1)}, b_{(2)}, \ldots)$, where $b_{(j)}$ is the normality of vehicle j determined by algorithm $\mathscr{A}$ and $b_{(j)} = 1$ (0) indicates the vehicle is anomalous (normal). For example, if the Greedy algorithm determines that the ego vehicle and Vehicle-2 are anomalous and Vehicle-1 is normal, its vector

form will be $\vec{\phi}_{k,\mathscr{E}} = (1, 0, 1)$. CADCA then utilizes the common normalized inner product to compute the similarity (S) between the vehicles' normality perceived by the ego vehicle and the result of each algorithm:

$$S_{k,\mathscr{A}} = (\vec{W}_k \cdot \vec{\phi}_{k,\mathscr{A}})/|\vec{\phi}_{k,\mathscr{A}}|. \tag{10}$$

CADCA then selects the view with the largest S as the most likely local view L_M and recommends the control that has the largest TTC under L_M if both of the controls are determined to be unsafe.

5 Analytical Properties of CADCA

We now discuss the security properties of CADCA while mapping them to different attack scenarios (S1–S6) in Figure 2.

Property 5.1. *(S1–S6: Detection Guarantee) If not all of the data are simultaneously manipulated to match the data correlation,* CADCA *is guaranteed to detect the anomaly.*

Proof. Due to the design of overlapping DSs as shown in Table 1, the DSs in CADCA *cannot* be partitioned into two groups, say G_A and G_B, such that:

$$\left(\bigcup_{\forall DS_j \in G_A} DS_j\right) \bigcap \left(\bigcup_{\forall DS_j \in G_B} DS_j\right) = \phi \text{ (a null set)}, \tag{11}$$

meaning that there will always be DTs overlapping across two DS groups. That is, as long as not all the data are manipulated simultaneously, there will be at least one DS containing both correct and anomalous data since there is no way to partition the data into a normal and an anomalous groups based on the DSs. □

Property 5.2. *(Low-Threat S1: Data Identification Guarantee) Greedy algorithm is guaranteed to identify the anomalous data x_a under a single-DT anomaly if no two DTs are covered by the same DSs.*

Proof. (Part-I: The first identified data must be x_a.) Assume that the Greedy algorithm identifies another normal data x_n ($\neq x_a$) first. Since the algorithm chooses x_n first (i.e., x_n must cover the same number as, or more anomalous DSs than, x_a) and no two data are covered by the exact same DSs, there must exist some anomalous DS_k, such that $x_n \in DS_k$ but $x_a \notin DS_k$. However, this contradicts the fact that x_a is the only anomalous data since any anomalous DS must contain at least one anomalous data.

(Part-II: Greedy algorithm must terminate after identifying x_a.) If the Greedy algorithm does not terminate after selecting x_a, then there are other DSs that fail the consistency check but do not contain x_a, contradicting the fact that x_a is the only anomalous data. Thus, the assumption must be false. □

Property 5.3. *(Low-Threat S1–S2: Entity Identification Guarantee) Solution-Space algorithm can identify the anomalous entity under a naive attack.*

Proof. Since no false-negatives can occur under a naive attack, no anomalous data will be ruled out by the algorithm. Therefore, the anomalous entity must be identified. □

Property 5.4. *(Low-Threat S1–S2: Ruling-Out Condition) If the normality inversion (Line 20 of Algorithm 1) is activated in the Solution-Space algorithm, then there must be a false-negative in the consistency check,* i.e., CADCA *is dealing with an attack of medium or high-level threat.*

Proof. Assume that no false negative occurs in the consistency check, but the Solution-Space algorithm activates the normality inversion for an anomalous DS (DS_k). Then, there must exist an x_a such that $I(x_a) = 1$ and $x_a \in DS_k$, but it is not included in the final result of the Solution-Space algorithm, and hence must be ruled out by some $DS_j \rightarrow \checkmark$. However, this contradicts the fact that DS_j must fail the consistency check under a naive attack if $x_a \in DS_j$. Thus, there must be a false-negative detection. □

Property 5.5. *(Mid-Threat S3–S4: Entity Identification Guarantee) Trust-ego algorithm will identify any non-ego, anomalous vehicle as long as not all the data of that entity are manipulated to match the normal data correlation.*

Proof. Suppose the output ($\mathscr{L}_{TE}$) of the Trust-Ego algorithm does not contain any data from an anomalous entity (E_t). Since the Trust-Ego algorithm will not determine any data from the ego vehicle to be potentially anomalous and there is no data of E_t determined to be anomalous in $\mathscr{L}_{TE}$, all DSs associated with E_t must pass the consistency check, contradicting the fact that not all data of that entity are manipulated to match the normal data correlation. Thus, $\mathscr{L}_{TE}$ must contain at least one data from E_t. □

Property 5.6. *(Mid-Threat S3–S4: Ruling-Out Condition) If the Trust-Ego algorithm cannot find a solution, the ego vehicle must be anomalous. (By algorithm design.)*

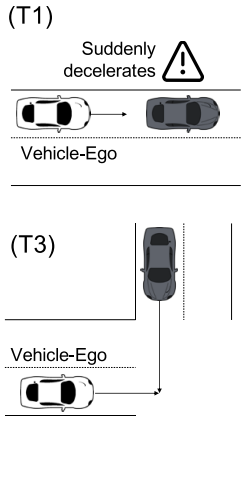
Property 5.7. *(High-Threat S5: Ruling-Out Condition and Entity Identification Guarantee) If no solution can be found by the Anomalous-Ego-Only algorithm, there must exist an anomalous data from a non-ego vehicle. (By algorithm design.)*

Property 5.8. *(High-Threat S5–S6: Entity Identification Guarantee) Anomalous-Ego-and-Others algorithm is guaranteed to identify the anomalous ego entity and any anomalous entity (E_t) if not all E_t's status data (i.e., the data excluding the distance measurement) are manipulated to match the normal data correlation.*

Proof. Since not all E_t's status data are manipulated to match the data correlation, there must exist some DS_k associated only with E_t and $DS_k \rightarrow \boldsymbol{\times}$. Therefore, the AEnO algorithm must identify at least one anomalous data $x_a \in DS_k$ from E_t to cover DS_k. □

Property 5.9. *(S1–S6: Entity Identification Guarantee) Anomalous-Individual algorithm is guaranteed to include the anomalous entities as long as not all the data/measurement of the anomalous source are manipulated coordinately. (Proof similar to Property 5.8.*

Computational Complexity. All algorithms have an $O(NK)$ computational complexity, where K and N are the numbers of data types per vehicle and DSs, respectively. See Section 6.5 for CADCA's end-to-end computation time analysis.



Low Threat	1743
T1	1404
T2	144
T3	51
T4	144
Mid Threat	2785
T1	2340
T2	240
T3	90
T4	120
High Threat	11255
T1	6425
T2	2940
T3	855
T4	1035
Total	15783

Fig. 5. CADCA's basic testing scenarios and the number of cases tested. We omitted different potential maneuvers of the ego SAV in the figures. CADCA, Context-Aware Detection and resolution of Control Anomalies; SAV, Semi-Autonomous Vehicle.

6 End-to-End Evaluation

6.1 Experimental Settings

Since CADCA is designed to operate under safety-critical conditions with sensor failures or malicious attacks,[2] we use Simulink with the automated driving toolbox [43] to evaluate CADCA's performance. The ego vehicle is equipped with a front camera and radar while its autonomous system controls both the steering and acceleration of the ego vehicle. Vehicles periodically broadcast BSMs to inform nearby vehicles of their location, speed, acceleration, heading, and yaw rate. To account for noisy sensor measurements in the real world, the ego vehicle's radar is assumed to have a maximum detection range of 174 m, range resolution of 2.5 m, 90% detection probability, and false alarm rate of 10^{-4}% and the camera generates 480×640 image frames every 0.1 s while its object recognition algorithm has a 90% detection rate [23]. These settings will be altered to account for the conditions where the sensors are anomalous, or under the influence of an inclement weather.

The testing scenarios (T1–T4) are designed to cover 98.56% of (in-transport) vehicle crashes in 2021 according to [44], including rear-end (41.8% coverage), sideswipe (19.62% coverage), angle (33.19% coverage), and head-on (3.90% coverage) collisions. See Figure 5 for the statistics and illustration of the test-cases:

T1: (Rear-end) The most crucial scenario for collision avoidance: a front vehicle suddenly brakes;
T2: (Rear-end and sideswipe) An unsafe lane change;
T3: (Angle) Potential side-collision; and
T4: (Sideswipe, angle, and head-on) At an intersection, the ego SAV may experience different safety-critical encounters with vehicles from other directions.

In each scenario, there are 1–5 non-ego vehicles in the vicinity (i.e., within two lanes) of the ego vehicle. We evaluate the conditions in which both autonomous and manual controls can lead to a collision. Specifically, each test case has a unique combination of ⟨initial vehicle location, vehicle maneuver, control input magnitude, control timing, attack/anomaly timing, attack magnitude⟩.

[2]It is too dangerous to conduct experiments on a production vehicle.

Also, the control input to vehicle acceleration and deceleration is limited to up to $2\,m/s^2$, which is below the maximum $2.24\,m/s^2$ acceleration and the maximum $3.36\,m/s^2$ deceleration observed in [45] for petrol vehicles. In each test case, at least one control input will lead to a collision. The attack will start right after a test case begins and last until a collision occurs or until the test case ends. We have also implemented a baseline approach "RA-BSM" based on [23], which is, to the best of our knowledge, the most recent risk assessment that accounts for data (un)availability and also utilizes both vehicle state estimation with inter-vehicle communications.

CADCA's performance is evaluated using two metrics:

— SR: The probability of successfully disallowing a control input to prevent collision.
— ***Incorrect Blockage Rate* (IBR)**: The probability of unnecessarily disallowing a control, which should have been allowed.

6.2 Low Threat

We first evaluate CADCA's performance when there is only one anomalous DT (the low-level threat in Figure 2) on top of (natural) measurement noises embedded in the sensor data. We start with the attack that tries to misguide the ego SAV's path planning by making the ego SAV believe a non-ego vehicle is traveling at an incorrect speed with the value deviation $\Delta v = \pm 5\text{–}15\,m/s$ (or 18–54 kph). We will henceforth use "VSC" to denote the *v*ehicle that will cause a *s*afety-*c*ritical situation and, unless specified otherwise, VSC will also be the victim of an attack (i.e., VSC's data perceived by the ego vehicle can be incorrect or manipulated).

Under each of T1–T4 scenarios, we design test-cases where the ego SAV's autonomous system may generate both safe and unsafe controls (due to imperfect control or sensors' blind spots) and further inject manual controls that may either cause or avoid collision. Note that the speed manipulations are designed to make the ego vehicle believe that it is in a safe condition while it is actually not. In T1 and T2 (T3 and T4), VSC's speed perceived by the ego vehicle will be larger (smaller) than the ground truth, making the ego vehicle believe that it has more time to react to VSC's action.

Figure 6(A) shows the SRs and IBRs of CADCA and RA-BSM. One can observe that CADCA is able to achieve ≥92.86% SR (T2) in disallowing unsafe controls and the level of data manipulation does not have any significant impact on CADCA's performance. CADCA is further shown to achieve >98% SR for T1, T3 and T4. On the other hand, RA-BSM can only achieve 55.15% SR and its performance further deteriorates when the perceived data deviates more from the ground truth, indicating that it cannot properly perform risk assessment when there is a data anomaly. The SR of RA-BSM decreases as attack magnitude increases because that SR is not entirely equivalent to detection rate. That is, while a system can detect the attack as the deviation increases, it may still have an incorrect estimation/recovery of the ground-truth condition. These results showcase CADCA's advantage over typical sensor-fusion techniques—the incorrect inputs may bias the state estimation if the input does not deviate significantly from the ground truth to make the filters exclude the tampered data. Furthermore, CADCA is able to achieve 0% IBR in all the test-cases, indicating that CADCA will not unnecessarily disallow any of the safe controls. That is, CADCA will allow drivers to manually control the vehicle if the control input is determined to be safe as if there were no additional "control-filtering."

6.3 Medium Threat

Let us consider the condition in which the attacker can manipulate the VSC's (entire) status report/BSM to prevent the ego SAV from performing accurate risk assessment. The data in manipulated VSC's BSMs will perfectly match the normal data correlation, leaving no trace of tampering.

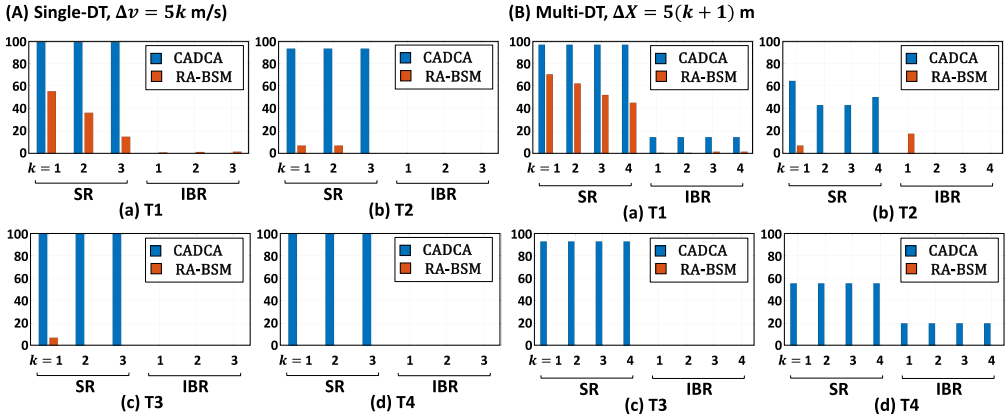


Fig. 6. Performance (%) comparison of CADCA (left/blue bars) and RA-BSM (right/red bars) under (A) single-DT and (B) multi-DT anomalies targeting VSC's speed and location, respectively. Missing bars indicate 0%. CADCA, Context-Aware Detection and resolution of Control Anomalies.

Note the data manipulation considered here can be viewed as the final stage of Drift-with-Devil attacks proposed in [46], which is one of the strongest attacks against the approaches utilizing state estimation. CADCA will face cases when it cannot deterministically identify anomalous data.

Figure 6(B) shows the performance of CADCA, where the manual controls are injected when VSC's location drifts away from its actual location by $\Delta X = 10$–25 m. CADCA is shown to achieve >92% SR in T1 and T3 while achieving 42–64% SR in T2 and T4. The lower SRs in T2 and T4 are caused by the fact that it is harder for CADCA to predict and reconstruct the maneuvering behavior of vehicles since the manipulated BSMs do not contain any useful information that reveals the actual VSC's maneuvering behavior. When CADCA's IBR is compared with those under single-DT manipulation, the IBRs in Figure 6(B) are slightly larger due to CADCA's safety-first design to always execute the safer control decision under an uncertain situation. Note that even though CADCA can only achieve moderate SRs in T2 and T4, it still has a >40% (absolute) SR advantage over RA-BSM.

6.4 High Threat

Next, we evaluate CADCA's performance under the high-level threat in which data of multiple entities (including the ego SAV) can be anomalous in addition to the attack considered in Section 6.3. That is, the attacker may now tamper with VSC's location data ($\Delta X = 5$–25 m) along with other DTs just like in Section 6.3 and *simultaneously* spoof the sensors/algorithms of the ego vehicle. To simulate this extreme condition, we purposely adjust the object detection rates of RADAR/camera to 0.9, 0.5 and 0.1 and assume there can be measurement errors ($\Delta d = 6$–10 m) for the ego vehicle's distance sensing. Note that the degradation of sensing quality can also be the result of severe weather condition. CADCA is shown to achieve 90.43–98.33% SR and ≤2.44 IBR in T1 while RA-BSM can only achieve ≤87.83 SR and ≤3.83 IBR (Table 3). We use the same set of ground-truth vehicle behaviors within a ⟨scenario, detection rate⟩ case to capture the effects of anomalous distance sensing on CADCA's performance.

Specifically, it is worth noting that CADCA does not experience any noticeable performance degradation even if Δd increases from 6 to 10, indicating CADCA's effective resolution of control conflicts irrespective of how much error is embedded in the data once the anomaly is detected. RA-BSM, however, will yield a much worse performance with larger data deviations. This observation further showcases CADCA's advantage in that its identification of anomalous data does not directly

Table 3. CADCA and RA-BSM's Performance (%) under Both Sensor/Algorithm Failure and Data Manipulation

Δd	T1				T2				T3				T4			
	CADCA SR	R-B SR	CADCA IBR	R-B IBR	CADCA SR	R-B SR	CADCA IBR	R-B IBR	CADCA SR	R-B SR	CADCA IBR	R-B IBR	CADCA SR	R-B SR	CADCA IBR	R-B IBR
Object Detection Rate = 0.1																
6	90.43	87.83	0.00	0.85	88.89	44.44	1.12	0.00	100	0.00	10.53	0.00	100	100	1.56	1.56
7	90.43	78.26	0.00	3.83	88.89	27.78	1.12	0.00	100	0.00	10.53	0.00	100	60.00	1.56	1.56
8	90.43	76.52	0.00	2.98	88.89	22.22	0.56	0.00	100	0.00	10.53	0.00	100	0.00	0.00	0.00
9	90.43	66.09	0.00	2.55	72.22	22.22	0.56	0.00	100	0.00	10.53	0.00	100	0.00	0.00	0.00
10	90.43	60.00	0.00	6.38	94.44	22.22	1.12	0.00	100	0.00	10.53	0.00	100	0.00	0.00	0.00
Object Detection Rate = 0.5																
6	98.25	68.42	2.44	0.73	50.00	50.00	0.00	0.00	95.12	0.00	0.00	0.00	100	100	1.56	1.56
7	98.25	57.89	2.44	0.73	50.00	33.33	0.00	0.00	95.12	0.00	0.00	0.00	100	60.00	1.56	1.56
8	98.25	47.37	2.44	0.73	66.67	27.78	0.00	0.00	95.12	0.00	0.00	0.00	100	0.00	0.00	0.00
9	98.25	35.09	2.44	0.73	66.67	22.22	0.56	0.56	95.12	0.00	0.00	0.00	100	0.00	0.00	0.00
10	98.25	31.58	2.44	0.73	66.67	22.22	0.56	0.56	95.12	0.00	0.00	0.00	100	0.00	0.00	0.00
Object Detection Rate = 0.9																
6	98.33	68.33	1.96	0.49	50.00	50.00	0.00	0.00	97.50	0.00	0.00	0.00	100	100	1.56	1.56
7	98.33	53.33	1.96	0.25	55.56	33.33	0.00	0.00	97.50	0.00	0.00	0.00	100	60.00	1.56	1.56
8	98.33	41.67	1.96	0.49	66.67	27.78	0.00	0.00	97.50	0.00	0.00	0.00	100	0.00	0.00	0.00
9	98.33	31.67	1.96	0.74	66.67	22.22	0.56	0.56	97.50	0.00	0.00	0.00	100	0.00	0.00	0.00
10	98.33	28.33	1.96	0.98	66.67	22.22	0.00	0.00	97.50	0.00	0.00	0.00	100	0.00	0.00	0.00

rely on data causality. A similar pattern can also be observed in T2–T4. As expected, CADCA shows a slightly worse performance in T2 due to the unpredictability of VSC's behavior. However, it can still achieve up to 72.22(= 94.44 − 22.22)% absolute SR increase over RA-BSM.

6.5 Execution Time Analysis

We evaluate CADCA's execution time on a 2016 MacBook Pro with 2.6 GHz Quad-Core Intel Core i7 CPU, which is relatively old hardware. Specifically, CADCA is implemented in Matlab with single-thread execution (only on CPU). We measure the execution time when there are 2–50 vehicles (including the ego vehicle). Figure 7 shows the average execution time that increases linearly with the number of vehicles. This result matches our analysis in Section 5—CADCA's algorithms have an $O(NK) \propto O(N)$ computation complexity when K, the number of data types, is fixed and N, the number of DSs, is proportional to the number of vehicles. Furthermore, even if CADCA operates with 50 vehicles, it only requires a 33.3 ms execution time on average—only 0.7% of T_S (i.e., the medium time to achieve a smooth control transition) and only 1.2% of T_R (i.e., the driver's reaction time to an urgent situation)—thus reacting to safety-critical events much faster than a typical driver. Even under the high-threat scenario of requiring more resources to perform data recovery and risk assessment, the execution time has a <8.3% increase (3.8% on average).

7 Discussion

7.1 Vulnerability of CADCA to a Super Adversary

There are two ways an attacker can harm CADCA's performance. First, to evade CADCA's detection, an adversary can skillfully craft an attack that tampers with the data to fully match the correct data correlation (S7 in Figure 2). This requires the adversary to be able to simultaneously manipulate multiple data with fine-grained control as mentioned in Section 3. Second, to defeat the collision avoidance (that CADCA may still detect), an adversary can attack when non-ego vehicles change their maneuvering drastically to keep CADCA from predicting their behavior under tampered inputs. This requires the adversary to be able to predict how non-ego vehicles would maneuver or have direct

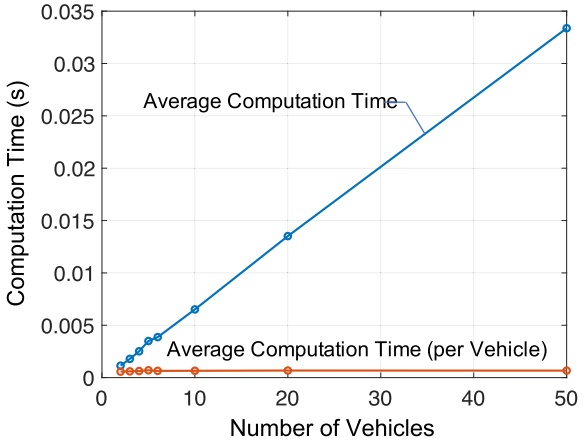


Fig. 7. The execution time of CADCA, where "per vehicle" represents the total computation time divided by the number of vehicles within the ego SAV's perception range. The execution time is computed based on the average of 10,000 executions for scenarios with the same number of vehicles. CADCA, Context-Aware Detection and resolution of Control Anomalies.

control over non-ego vehicles. Both exploits require the adversary to have super (unrealistic/non-scalable) capabilities to manipulate data at specific instants or control non-ego vehicles.

7.2 Extension of CADCA

7.2.1 General Expansion. CADCA can be adjusted to support a more relaxed threat model than the one specified in Section 3. Specifically, the attack magnitude (ξ) of data m in Equation (2) can be set to 0 if m is trusted (only measurement errors can occur and within a certain known bound ψ). All the rest of CADCA's operations will still be the same, except that m will no longer be considered as an anomaly source.

If the assumption of no trusted data holds, the challenge of enabling CADCA to handle attacks with coordinated multiple entities is the lack of objective evidence (based on the law of physics) that can definitively determine the received data to be anomalous. As multi-entity coordination attacks can control all degrees-of-freedom in Equation (2) to evade detection mechanisms, new detection dimensions must be introduced to detect such a strong adversary.

For example, this new detection dimension can be the characteristics of vehicle capability, such as the limit of acceleration that the target vehicle can achieve. By doing so, CADCA can report that an anomaly has occurred if the received data indicate that a vehicle exceeds this limit. However, this new dimension can also be the adversary's manipulation target, as this context information also needs to be either obtained from the external entities or estimated by the ego vehicle itself.

The vehicle's behavior pattern can be another detection dimension. By comparing the observed vehicle movement or maneuver with some pre-existing behavior models, CADCA can detect and report an anomaly if the current observed vehicle behavior deviates from the models. The challenge of this approach is the source of the behavior models. In general, the models can be derived from either external entities' reports/messages or the ego vehicle's observation. While a strong adversary can also falsify the former, the latter requires long-term observation, which could be difficult to achieve in a dynamic driving scenario.

Although the introduction of the above two additional dimensions will not eliminate the possibility of multi-entity coordinated attacks, adding more observations and constraints will make attacks harder to launch, creating more and higher barriers to advanced attacks.

7.2.2 Cooperative Scenario. While SAE J2945/8 defines how vehicles broadcast perception information of roads/objects nearby, CADCA can extend it further to support the case when vehicles can share their observations with one another. Specifically, Table 1 can be expanded to include the measurements/DSes from the non-ego to other vehicles (similar to Row V1,6) while the algorithms in CADCA already support this cooperative scenario as they are designed to operate on expandable DSes, instead of specific equations.

7.2.3 Absence of V2X Capabilities. The assumption of V2X capabilities is not a hard requirement of CADCA. Specifically, as mentioned in Section 3, the state report from other entities is optional. CADCA can either entirely remove the state report from Equation (2) during its operation, or the ego vehicle can use its own measurements (based on radar, LIDAR, camera, etc.) to estimate the state of external entities. Note that if the measurements are made by the ego vehicle itself, the corresponding detection thresholds for anomalies should also be adjusted based on the error bound of its own sensors.

8 Conclusions

To reduce/eliminate the danger of static assignment of control priority in safety-critical situations due to potential anomalies (i.e., failures and attacks) and malicious/erroneous control input, we have proposed CADCA, a decision-maker for SAVs, that can perform risk assessment and resolve control conflicts when any of the received/perceived data is anomalous. Specifically, CADCA selects a control input that is safe to execute under uncertain situations. Our extensive evaluation has shown CADCA to achieve a 98% success rate in avoiding use of unsafe control inputs (T1) and have a $\geq$0.4 more success rate than the latest representative prior work for most commonly seen scenarios (T2).

References

[1] National Highway Traffic Safety Administration, US Department of Transportation. 2015. Traffic Safety Facts. Technical Report. US Department of Transportation.

[2] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. In *Black Hat USA 2015*, 1–93. Retrieved from https://www.ioactive.com/wp-content/uploads/pdfs/IOActive_Remote_Car_Hacking.pdf

[3] GMC Division of General Motors. Explore GMC Safety and Driver Assistance Technology. Retrieved February 16, 2022 from https://www.gmc.com/safety-features

[4] Dominic Gates and Mike Baker. The inside story of mcas: How boeing's 737 max system gained power and lost safeguards. Retrieved January 15, 2020 from https://www.seattletimes.com/seattle-news/times-watchdog/the-inside-story-of-mcas-how-boeings-737-max-system-gained-power-and-lost-safeguards/

[5] Boeing 737 MAX Updates—737 MAX Return To Service Updates and Information. Retrieved September 9, 2021 from https://www.boeing.com/737-max-updates/

[6] Laura Smith-Spark. Report: Germanwings crash co-pilot tested 100-foot descent setting. Retrieved February 6, 2023 from https://www.cnn.com/2015/05/06/europe/france-germanwings-crash-report/index.html

[7] Hongjun Choi, Wen Chuan Lee, Yousra Aafer, Fan Fei, Zhan Tu, Xiangyu Zhang, Dongyan Xu, and Xinyan Deng. 2018. Detecting attacks against robotic vehicles: A control invariant approach. In *Proceedings of the ACM Conference on Computer and Communications Security*. ACM, New York, NY, 801–816.

[8] Raul Quinonez, Jairo Giraldo, Luis Salazar, Santa Cruz, and Erick Bauman. 2020. SAVIOR: Securing autonomous vehicles with robust physical invariants. In *29th USENIX Security Symposium*, 895–912.

[9] Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman. 2021. PID-Piper: Recovering robotic vehicles from physical attacks. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, Taipei, Taiwan, 26–38.

[10] SAE International. J2735D: Dedicated Short Range Communications (DSRC) Message Set Dictionary—SAE International. Retrieved February 1, 2020 from https://www.sae.org/standards/j2735_201603-dedicated-short-range-communications-dsrc-message-set-dictionary

[11] United States Department of Transportation. Intelligent Transportation Systems—ITS in Use Today. Retrieved February 1, 2020 from https://www.its.dot.gov/resources/fastfacts.htm

[12] United States Department of Transportation. AERIS Operational Scenarios. Retrieved April 5, 2021 from https://www.its.dot.gov/research_archives/aeris/pdf/AERIS_Operational_Scenarios011014.pdf

[13] United States Department of Transportation. Intelligent Transportation Systems—Connected Vehicle Pilot Deployment Program. Retrieved April 24, 2021 from https://www.its.dot.gov/pilots/
[14] Ford Media Center. Ford Accelerates Connectivity Strategy in China; Targets Production of First C-V2X-Equipped Vehicle in 2021. Retrieved January 5, 2021 from https://media.ford.com/content/fordmedia/fap/cn/en/news/2019/03/26/Ford_Accelerates_Connectivity_Strategy_in_China_And_Targets_Production_of_First_C-V2X-Equipped_Vehicle_in_2021.html
[15] Buick Debuts V2X Technology and Launches Refreshed GL6 MPV in China. Retrieved January 5, 2021 from https://media.gm.com/media/cn/en/gm/news.detail.html/content/Pages/news/cn/en/2020/Nov/1120-Buick.html
[16] 5G NR-based C-V2X (Qualcomm). Retrieved January 5, 2021 from https://www.qualcomm.com/invention/5g/cellular-v2x
[17] Stéphanie Lefèvre, Dizan Vasquez, and Christian Laugier. 2014. A survey on motion prediction and risk assessment for intelligent vehicles. Retrieved from https://dx.doi.org/10.1186/s40648-014-0001-z
[18] Adam Berthelot, Andreas Tamke, Thao Dang, and Gabi Breuel. 2011. Handling uncertainties in criticality assessment. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 571–576.
[19] Andreas Lawitzky, Daniel Althoff, Christoph F. Passenberg, Georg Tanzmeister, Dirk Wollherr, and Martin Buss. 2013. Interactive scene prediction for automotive applications. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, 1028–1033.
[20] Thierry Fraichard and Hajime Asama. 2003. Inevitable collision states—A step towards safer robots? In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)*, 2–9.
[21] Mattias Brännström, Erik Coelingh, and Jonas Sjöberg. 2010. Model-based threat assessment for avoiding arbitrary vehicle collisions. *IEEE Transactions on Intelligent Transportation Systems* 11, 3 (Sep. 2010), 658–669.
[22] Nico Kaempchen, Bruno Schiele, and Klaus Dietmayer. 2009. Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios. *IEEE Transactions on Intelligent Transportation Systems* 10, 4 (Dec. 2009), 678–687.
[23] Minjin Baek, Donggi Jeong, Dongho Choi, and Sangsun Lee. 2020. Vehicle trajectory prediction and collision warning via fusion of multisensors and wireless vehicular communications. *Sensors* 20, 1 (2020), 288. DOI: https://doi.org/10.3390/s20010288
[24] A. Nourbakhshrezaei, M. Jadidi, M. R. Delavar, and B. Moshiri. 2024. A novel context-aware system to improve driver's field of view in urban traffic networks. *Journal of Intelligent Transportation Systems* 28, 3 (2024), 297–312.
[25] Erfan Pakdamanian, Erzhen Hu, Shili Sheng, Sarit Kraus, Seongkook Heo, and Lu Feng. 2022. Enjoy the ride consciously with cawa: Context-aware advisory warnings for automated driving. *Main Proceedings of the 14th International ACM Conference on Automotive User Interfaces and Interactive Vehicular Applications (Automotive UI '22)*, 75–85
[26] Guang Li Huang, Arkady Zaslavsky, Seng W. Loke, Amin Abkenar, Alexey Medvedev, and Alireza Hassani. 2023. Context-aware machine learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems* 24, 1 (2023), 17–36.
[27] Yuvaraj Selvaraj, Wolfgang Ahrendt, and Martin Fabian. 2023. Formal development of safe automated driving using differential dynamic logic. *IEEE Transactions on Intelligent Vehicles* 8, 1 (2023), 988–1000.
[28] Lui Sha. 2001. Using simplicity to control complexity. *IEEE Software* 18, 7 (2001), 20–28.
[29] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. 2015. A survey of fault diagnosis and fault-tolerant techniques-part I: Fault diagnosis with model-based and signal-based approaches. *IEEE Transactions on Industrial Electronics* 62, 6 (Jun. 2015), 3757–3767.
[30] Danfeng (Daphne) Yao, Xiaokui Shu, Long Cheng, and Salvatore J. Stolfo. 2017. Anomaly detection as a service: Challenges, advances, and opportunities. *Synthesis Lectures on Information Security, Privacy, and Trust* 9, 3 (Oct. 2017), 1–173.
[31] Yasser Shoukry, Pierluigi Nuzzo, Alberto Puggelli, Alberto L. Sangiovanni-Vincentelli, Sanjit A. Seshia, and Paulo Tabuada. Secure state estimation for cyber-physical systems under sensor attacks: A satisfiability modulo theory approach. *IEEE Transactions on Automatic Control* 62, 10 (Oct. 2017), 4917–4932.
[32] Rens Wouter van der Heijden, Stefan Dietzel, Tim Leinmuller, and Frank Kargl. 2019. Survey on misbehavior detection in cooperative intelligent transportation systems. *IEEE Communications Surveys and Tutorials* 21, 1 (2019), 779–811.
[33] Fatih Sakiz and Sevil Sen. 2017. A survey of attacks and detection mechanisms on intelligent transportation systems: VANETs and IoV. *Ad Hoc Networks* 61 (Jun. 2017), 33–50. DOI: https://doi.org/10.1016/j.adhoc.2017.03.006
[34] Gopi Krishnan Rajbahadur, Andrew J. Malton, Andrew Walenstein, and Ahmed E. Hassan. 2018. A survey of anomaly detection for connected vehicle cybersecurity and safety. In *Proceedings of the IEEE Intelligent Vehicles Symposium*. Institute of Electrical and Electronics Engineers Inc., 421–426.
[35] Safa Boumiza and Rafik Braham. 2018. Intrusion threats and security solutions for autonomous vehicle networks. In *Proceedings of the IEEE/ACS International Conference on Computer Systems and Applications (AICCSA '18)*. IEEE Computer Society, 120–127.

[36] Attila Jaeger, Norbert Bißmeyer, Hagen Stübing, and Sorin A. Huss. 2012. A novel framework for efficient mobility data verification in vehicular ad-hoc networks. *International Journal of Intelligent Transportation Systems Research* 10, 1 (Jan. 2012), 11–21.

[37] Norbert Bißmeyer, Sebastian Mauthofer, Kpatcha M. Bayarou, and Frank Kargl. 2012. Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters. In *IEEE Vehicular Networking Conference (VNC '12)*, 78–85.

[38] Hagen Stübing, Jonas Firl, and Sorin A. Huss. 2011. A two-stage verification process for car-to-x mobility data based on path prediction and probabilistic maneuver recognition. In *2011 IEEE Vehicular Networking Conference (VNC '11)*. IEEE, 17–24.

[39] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. 2017. WALNUT: Waging doubt on the integrity of MEMS accelerometers with acoustic injection attacks. In *Proceedings of the 2nd IEEE European Symposium on Security and Privacy (EuroS and P '17)*. Institute of Electrical and Electronics Engineers Inc., 3–18.

[40] D. Halliday, R. Resnick, and J. Walker. 2013. *Fundamentals of Physics*. Fundamentals of Physics. Wiley.

[41] John L. Campbell, James L. Brown, Justin S. Graving, Monica G. Lichty, Thomas Sanquist, Diane N. Williams, and Justin F. Morgan. 2016. Human Factors Design Guidance For Driver-Vehicle Interfaces. Technical Report. U.S. Department of Transportation, National Highway Traffic Safety Administration.

[42] Beshr Sultan and Mike Mcdonald. 2003. Assessing The Safety Benefit of Automatic Collision Avoidance Systems (During Emergency Braking Situations). Technical Report, University of Southampton.

[43] MathWorks. Automated Driving Toolbox. Retrieved September 18, 2021 from https://www.mathworks.com/products/automated-driving.html/

[44] NHTSA. 2023. Section "Collision With Motor Vehicle in Transport" in Table 29 of Traffic Safety Facts 2021—A Compilation of Motor Vehicle Traffic Crash Data. Retrieved February 24, 2024 from https://crashstats.nhtsa.dot.gov/Api/Public/Publication/813527

[45] P. S. Bokare and A. K. Maurya. 2017. Acceleration-deceleration behaviour of various vehicle types. *Transportation Research Procedia* 25 (2017), 4733–4749.

[46] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. 2020. Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under GPS spoofing (extended version). arXiv:2006.10318. Retrieved from https://arxiv.org/abs/2006.10318

Received 15 September 2024; revised 27 June 2025; accepted 28 October 2025