Context-Aware Anomaly Detection Using Vehicle Dynamics

Chun-Yu Chen
University of Michigan
Ann Arbor, MI, USA
chunyuc@umich.edu

Kang G. Shin
University of Michigan
Ann Arbor, MI, USA
kgshin@umich.edu

Soodeh Dadras
Ford Motor Company
Dearborn, MI, USA
sdadras1@ford.com

ABSTRACT

Replacing traditional vehicular components with electronic components brings numerous benefits but also introduces new vulnerabilities. To cope with this double-edged trend, we propose Context-Aware Detection of abnormal vehicle Dynamics (CADD) in general, or abnormal vehicle accelerations in particular. To account for the *limited data availability* common in production vehicles, we propose a new detection mechanism based on *estimated* vehicular contexts, instead of the commonly used "predict-input-then-compare." That is, without relying on the unrealistically assumed availability of detailed measurements for accurate behavior modeling and prediction, CADD utilizes four sets of vehicle data to perform anomaly detection by cross-validating estimations of the underlying driving contexts, including road inclination, tire slippage, and total mass. Our extensive evaluation with >87,000 test-cases has shown CADD to achieve >96% recall and <0.5% false positive rate. Furthermore, CADD can efficiently pinpoint the anomalous group of data with >95% accuracy when the vehicle's behavior deviates 0.07g (0.69 m/s^2) from its normal pattern.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; **Software and application security**.

KEYWORDS

Vehicle anomaly detection; cyber-physical systems

ACM Reference Format:
Chun-Yu Chen, Kang G. Shin, and Soodeh Dadras. 2024. Context-Aware Anomaly Detection Using Vehicle Dynamics. In *The 27th International Symposium on Research in Attacks, Intrusions and Defenses (RAID 2024), September 30-October 02, 2024, Padua, Italy*. ACM, New York, NY, USA, 20 pages. https://doi.org/10.1145/3678890.3678895

1 INTRODUCTION

As we are entering the era of autonomous driving and vehicle electrification, many vehicular components are getting replaced by their electronic counterparts to provide advanced services, such as adaptive cruise control. While embracing the convenience brought by the new technologies, there have been new vulnerabilities brought by the electronic components like software bugs/glitches and cyber attacks to the vehicles. Manipulation, injection, or spoofing of in-vehicle data is shown to cause unexpected vehicle behavior [1, 2] or even seize the control of a vehicle [3]. The consequences of exploiting such vulnerabilities can range from confusing the driver to jeopardizing the safety of driver/passengers.

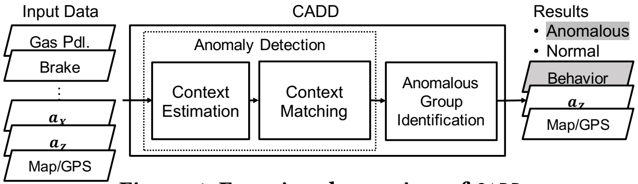


Figure 1: Functional overview of CADD.

To mitigate these risks, researchers have been exploring ways of characterizing the normal vehicle behavior and then using it to detect anomalous behaviors. Their proposed solutions can be categorized according to the detection targets, as the detection of *Message Injection* (MI) or *Data Anomaly* (DA). The former [4–8] — also commonly referred to as *intrusion detection* — focuses on the detection of any (malicious) message injection into the in-vehicle network (IVN). The latter [9–21] deals with the detection of anomalies observed in the *data* level. Specifically, the former focuses on the detection of anomalous MI on the IVN and can achieve high (> 97%) true positive rate (TPR) and low (< 1%) false positive rate (FPR) in exposing the message sent by a malicious/malfunctioning Electronic Control Unit (ECU), but cannot detect/identify the condition in which only the *vehicle* behavior or the data values in messages are changed.

On the other hand, the anomalies in data values are detected by capturing inconsistencies between the data of interest. The correlation (or causality) between in-vehicle data is modeled and then an anomaly will be reported if the expected correlation between the data does not hold. These approaches can detect the conditions in which (i) the anomaly does not originate from the MI behavior, or (ii) only the data value in a message is manipulated without altering other message transmission characteristics. However, most DA solutions have overlooked a common and crucial fact: detailed in-vehicle measurements are usually inaccessible via in-vehicle networks (*e.g.*, wheel torque [19] and brake torque [17]). On the other hand, those solutions that account for limited data availability can at best achieve moderate performance (Section 2). Specifically, there are three major technical challenges to overcome for efficient detection of data anomalies.

C1: *Diverse Operating Contexts.* Production vehicles usually operate in diverse environments with different contexts (*e.g.*, road conditions) and may not be equipped with all the sensors that can detect such contexts directly — since those sensors increase cost and are not required for vehicles' basic operation. However, the vehicle state depends strongly on its operating environment. For example, the level of acceleration resulting from pressing a gas pedal at 50% level when the vehicle is traveling steep uphill, will

RAID 2024, September 30-October 02, 2024, Padua, Italy

ACM ISBN 979-8-4007-0959-3/24/09
https://doi.org/10.1145/3678890.3678895

be much lower than that of traveling downhill or on a flat road. Therefore, it is important for the system to estimate, and account for the operating "context" for accurate detection of anomalies.

C2: *Limited Data Availability.* The availability of in-vehicle data is often limited, and only coarse-grained or specific data can be accessed/surfaced for the following reasons. First, the standardized on-board diagnostic II (OBD-II) messages based on SAE J1979 only cover the data types related to transmission. Second, CAN, the *de facto* standard for in-vehicle communications, has only limited bandwidth, *i.e.* 1Mbps and 5–12Mbps with a high-speed CAN [22] and CAN-FD [23], respectively. Also, CAN is usually divided into subnets with only certain necessary data being sent via a gateway to other subnets. Utilization of detailed measurements that cannot be acquired through standard OBD-II messages or a simple add-on to the vehicle cannot be deployed without a structural change to in-vehicle networks.[1] This limited data availability is the very reason why most, if not all, model-based detection schemes (*e.g.*, [24, 25]) require their models to be re-designed for use in production vehicles (Section 2), i.e., they were tailored to robotic/customized (instead of production) vehicles.

C3: *Training Difficulty.* Due to customization, adjustment, or varying error-tolerance of components in each vehicle, unified model and parameters for vehicles with the same model/make to work consistently are neither existent nor feasible. So, the system must learn/estimate the vehicle's normal behavior pattern based only on partial/incomplete observable information (C1 & C2) for anomaly detection. Also, the system must be able to work correctly even when it encounters a situation never experienced before; training for all possible situations is impossible! The prior work relying on previously observed data pattern (*e.g.*, [26]) cannot work under conditions not in their training scenarios.

In contrast to prior studies that focus on far-future or customized deployment where either 1) limited bandwidth is not a problem or 2) detailed measurements are readily available, we propose Context-aware Anomaly Detection in vehicle Dynamics (CADD), tailored for production vehicles *and* those with limited data availability, that verifies whether the controls from a human/autonomous driver match their resultant vehicle dynamic/acceleration.

Fig. 1 shows the functional overview of CADD, which can be implemented as a module in an ECU/Gateway connected to IVNs. Specifically, CADD takes vehicle data as input, performs anomaly detection followed by anomalous group identification (*i.e.*, the process for identifying potentially anomalous data), and outputs the information of potential anomalous source(s) to the downstream data-consumers. Example applications of CADD include i) an early warning system (*not* for real-time defense) to notify vehicle owners of potential system anomalies including faults and attacks for further inspection, or ii) a tool for detecting data manipulation to cheat on applications like usage-based insurance. CADD has the following salient features:

P1: To facilitate ready-to-use and aftermarket solutions for OEMs and third-parties,CADD does not require any modification to vehicles to facilitate its deployment. In particular, CADD eliminates the need of controller setpoint/output, commonly used/assumed in prior work as input, for the detection of anomalies. That is, neither detailed measurements (*e.g.*, wheel/brake torque) nor vehicle parameters (*e.g.*, gear transition curve) are required for CADD's deployment and the unknown characteristics of normal vehicle behavior considered in CADD can all be obtained during its training.

P2: CADD considers three fundamental contexts *simultaneously* — *road inclination* (RI), *tire slippage* (TS), and *total mass* (TM) of the vehicle including passengers and cargo — that influence a vehicle's longitudinal acceleration behavior regardless of how the vehicle is maneuvered. While these contexts are by no means exhaustive, they are known to be the most influential factors of a vehicle's behavior [27–32].

Due to the lack of sufficient inputs, the common "predict-input-then-compare" approaches (Section 2.3) require modifications to i) their behavior prediction model to be applicable to production vehicles and ii) their detection mechanisms to address prediction uncertainties. Therefore, we propose a detection mechanism based on context estimations. CADD estimates the *(uncertain) context data*, instead of system dynamics, from four sets of data and then determines if there is an anomaly by checking whether the context estimations are consistent with each other. That way, CADD i) eliminates the requirement of knowing the correct context before performing anomaly detection, ii) takes the operation context into account, and iii) does not require a fixed set of trusted sources for its detection (meeting C1 & P2). Context estimation further provides an adjustment of threshold that directly maps to physical properties of the vehicle operation.

Also, the normal operation models constructed by CADD are described based on the common mechanical design of modern vehicles and the laws of physics. Therefore, only the vehicle's (not the driver's) behavior will be captured, enabling CADD to *automatically* extend the thus-constructed models without requiring any actual training under the exact same condition (meeting C3). Furthermore, other than the optional input (*i.e.*, brake), all required data can be directly obtained from standard OBD-II or common CAN messages on IVN (meeting C2 & P1)[2].

This paper makes the following main contributions:

- A new design of anomaly detection based on context estimation, instead of the common "predict-then-compare" of system behavior (Section 4).
- Design of CADD for verifying the relationship between control inputs and vehicle dynamics:
  - Efficient models for capturing vehicle normal behavior and context estimation (Section 5); and
  - Mechanisms for detecting inconsistencies and identifying anomalous sources/groups (Section 6).
- Demonstration of CADD's performance via extensive evaluation. It is shown to achieve >96% recall and <0.5% false-positive rate. CADD efficiently identifies the anomalous group of data with >95% accuracy (Section 7).

2 RELATED WORK AND COMPARISON

2.1 Data Anomaly Detection

Since there already exist two comprehensive surveys [33, 34] of data anomaly detection in *general* control systems, here we only discuss

[1] See Appendix-A for more background information.

[2] Even if some data are not available on IVN, they can also be easily obtained by an external IMU/device (*e.g.*, an OBD-II dongle or smartphone).

the approaches in the *vehicle* domain or those directly related to CADD. Prior related works can be categorized as *fault detection and isolation* (FDI) [9–11] or *anomaly detection* (AD) [12–21, 35, 36]. FDI focuses on the detection and identification/isolation of the faulty source(s) while AD focuses only on the detection of abnormal behavior caused by both component failures and adversarial attacks.

The most common framework used in FDI describes the system of interest using such techniques as Bond Graph [9], and formulates the causal relationships between components as an equations system. Each equation is an observer designed to capture specific aspects of a failure. While treating the failures as unknown variables, the source of a failure can be identified/isolated efficiently by solving the system equations [9–11]. In particular, FDI focuses on modeling the interactions between components while requiring detailed component design and architecture to function as specified.

On the other hand, AD exploits the correlation between data to construct a normal behavior model for detecting anomalies. The authors of [12–15] focused on the detection of a vehcile engine while Xi *et al.* [16] and Cho *et al.* [17] focused on the transmission and brake system, respectively. While these studies focused on a single functional group, Xue *et al.* [37] proposed a detection scheme based on roll, steering and accelerating dynamics, and the authors of [37] explicitly stated that some required input data (*e.g.*, pitch angular speed) are *not* commonly available in modern vehicles. [35] and [36] focused on specific attack (*e.g.*, replay, fuzzing, etc.) types while their detection was designed to capture the features of those attacks by introducing a "watermarked" input requiring ECU modifications. Ganesan *et al.* [18] proposed an anomaly detection system that covers multiple functional groups. They considered pair-wise data correlation and used clustering to determine the context (*i.e.*, traffic pattern) for the detection of abnormal vehicle behavior. Likewise, Guo *et al.*[19] proposed a detection system, called *EVAD*, which is reported to achieve 98.8% TPR and 1% FPR based on pair-wise correlation of detailed in-vehicle measurements. However, some detection pairs in [19] also utilize data usually *unavailable* on IVNs, such as ⟨wheel torque, acceleration⟩ and ⟨brake torque, brake pedal⟩. [20] proposed using neural networks to detect anomalies in vehicle speed and engine rpm/torque. However, it is reported to achieve only modest performance with <80% TPR and >15% FPR in detecting anomalies. Dash *et al.* [21] proposed a detection system, called *PID-Piper*, based on a long short-term memory (LSTM) learning architecture for control behavior monitoring.

2.2 Context Estimation

Since road inclination (RI) and tire slippage (TS) caused by insufficient road friction (RF) are very important for vehicle control systems to enhance system stability and reduce fuel consumption, several approaches have been proposed for their estimation. Mangan *et al.* [27] proposed a method utilizing a vehicle's physical/kinematic model to estimate RI based on the vehicle's speed, acceleration, brake pressure, and engine torque by computing the difference between the expected acceleration on a flat road and the actual measured acceleration. Mahyuddin *et al.* [28] proposed a method that utilizes vehicle speed and driving torque, and adopts filtering and adaptive observer techniques for the estimation. Jauch *et al.* [29] proposed an IMU-based method for RI estimation. Specifically, RI (θ) is estimated based on i) the difference between the measured longitudinal acceleration (a_X) and the derivative of the vehicle's speed (v) relative to the ground: $\theta = \arcsin\left[(a_X - dv/dt)/g\right]$, where t is time and g is gravitational acceleration; and ii) altitude difference Δh (obtained from GPS) divided by the traveled distance $\Delta\ell$: $\theta = \arcsin[\Delta h/\Delta\ell]$. For pure RF estimation, Muller *et al.* [30] proposed a slip-based approach to estimate the maximum friction when the brake is pressed. Similarly to [29], the authors of [31] and [32] utilized measurements of vehicle dynamics together with filtering techniques to estimate RI and RF/TS.

2.3 Comparison of CADD and Prior Work

1) Data Requirement: CADD is designed for the prevalent case of limited data availability to support ready-to-use and aftermarket solutions. While the data commonly available on IVNs are control inputs and dynamic measurements, the lack of final control output, such as wheel torque, will make it ineffective/infeasible to apply prior DA approaches including control/system invariant and residual design [24, 38–40]. Also, due to the nature of their design, the aforementioned approaches focus on mechanisms inherently tied to their required data, and hence there is no easy way to change them without re-designing the entire system.

2) Detection Mechanism: Prior invariant/observer-based anomaly detection mainly follows a "predict-input-then-compare" approach that predicts some target data that is *readily available as part of the input data* and then compares the received and the predicted data values. For example, *PID-Piper* [21] predicts the controller input and output (while both controller input and output are inputs to *PID-Piper*) based on the model obtained from LSTM given the controller output and input, respectively. Then, it reports an anomaly if the predicted and the observed controller-inputs/outputs do not match with each other when the cumulative sum (CUSUM) [21, 24] of their differences exceeds a certain threshold. The "predict-input-then-compare" methods require detailed inputs for their prediction models (*e.g.*, wheel torque and/or brake torque [24, 38–40]), and hence they either cannot be directly applied to production vehicles due to insufficient input or can only achieve modest performance if not all the necessary data are available (Section 7). In contrast, CADD has a new detection mechanism based on context estimation, tailored to operate with limited data availability as is commonly the case. That is, instead of performing data prediction *for existing inputs* as in the prior work, CADD divides its inputs into *different data groups* so that anomalies can be detected by utilizing data groups to cross-validate *the same missing information*, i.e., the context that cannot be directly observed.

3) Context-Awareness: There is also no easy way to combine prior detection and context estimation approaches to form a context-aware anomaly detection because the latter usually assumes the availability of detailed vehicle parameters and operation mechanisms for the estimation system (*e.g.*, gear and final drive ratio in [27]). Since prior DA approaches do not consider context estimation in the first place, they do not have any training mechanism to obtain those parameters either.

Besides the availability of necessary data, a detection system must also consider all three major contexts *simultaneously.* However, incorporating all three contexts in a detection system is not

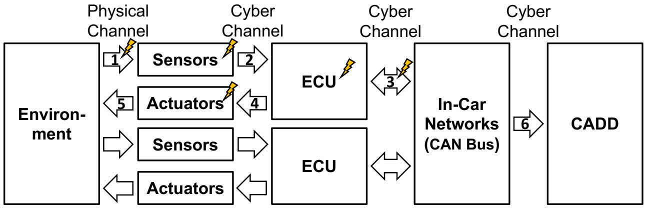


Figure 2: Information flow in CADD, where the arrows represent the interaction channels between different components. While sensors and actuators may have physical interactions (Arrow-1 & 5) with the environment, ECUs and their connected sensors/actuators typically communicate via cyber channels, such as cables or IVNs. The lightning icons are the anomaly and attack interfaces covered by CADD (Section 3.2).

as easy as including different CE approaches in the system. For example, [31] can be used to perform RI estimation and detect tire slippage while [28] focuses on the estimation of vehicle mass. However, since [28] is not designed to operate when a tire slippage occurs, a combination of the two approaches cannot detect anomalies when there is a tire slippage. Thus, prior CE approaches cannot be applied directly to CADD for context estimation. As a result, instead of trying to utilize the models in prior CE approaches, we develop behavior models to account for the effect of driving context to address the aforementioned challenges. The proposed models in CADD can not only capture the vehicle's normal behavior under different contexts but also be used to perform context estimation. That way, all the necessary parameters can be obtained from CADD's training phase, facilitating its deployment.

3 DETECTION SCOPE AND THREAT MODEL

3.1 Detection Scope

We first introduce terminologies to be used throughout the paper. We define the end-to-end (*i.e.*, control-to-dynamics) acceleration behavior of a vehicle as the *vehicle's behavior* (VB). Let *Data Of Interest* (DOI) be the data or measurements that CADD uses in its anomaly detection, including:

- *Major Detection Target (i.e., VB data)*:
  - *Control Input*: gas pedal or throttle position (g_a), brake pedal position or master cylinder pressure (b_r),[3] gear level (g_r), and engine torque (T_q);
  - *Dynamics Measurements*: longitudinal acceleration (a_X), and speed (v);
- *Assistance Data*: vertical acceleration (a_Z) and a sensor for road grade (θ_d) estimation, *e.g.*, GPS (standalone or with maps) or inclinometer.

Specifically, g_a, T_q, g_r and v can be extracted from standard OBD-II messages [41] and the rest are commonly available on IVN in drive-by-wire vehicles [42]; otherwise, a_X, a_Z, and GPS can also be obtained from an external device like a smartphone. As mentioned earlier, unlike prior work, CADD does *not* assume the availability of the final system outputs/setpoints (*e.g.*, wheel and brake torque) since they are usually not available.

CADD is designed to detect VB anomalies in the *data level* (Fig. 2). An *anomaly* (*i.e.*, the detection target) considered in CADD is defined as the occurrence of:

A1. (Physical Space) A change of the vehicle's response to control input that does not come from the change of driving contexts (*i.e.*, RI, TS and TM); or

A2. (Cyber Space) An inconsistency between the VB pattern and the driving context in the *data level*.

For example, if the vehicle changes its acceleration due to an abnormal torque output from the engine, then it will be considered as an anomaly. However, if the low acceleration is caused by driving on an uphill or slippery road, it would not be an anomaly. For A1, longitudinal acceleration (a_X) deviating by more than xg from its normal value is defined as an anomaly, where x is a design parameter and g is the gravitational acceleration. Similarly, input information of road inclination (θ_d) deviating by more than y° from its ground truth is defined as an anomaly in A2, where y is also a design parameter related to x (to be evaluated in Section 7).

3.2 Threat Model

In addition to component failures (the components with lightning icons in Fig. 2) that fundamentally change the vehicle's response to control input (*i.e.*, A1), we assume the adversary has the goal of compromising the estimation/prediction of the vehicle's state to deviate from its set course or waypoints and launches *remote* attacks — those attacks not resulting from the adversary's physical tampering with the vehicle hardware. Some examples of remote attacks are i) spoofing sensors (Arrow 1 in Fig. 2) with electromagnetic or acoustic interference [43] or ii) making ECU transmit incorrect dynamics measurements or assistance data on IVNs (Arrow 3) by exploiting bugs/vulnerabilities of ECUs via wireless connection [3].

CADD is designed to detect anomalies when not all status measurements (*i.e.*, dynamic measurements or assistance data defined in Section 3.1) are compromised without knowing which data is correct. Note that the above setting is a common requirement for *all* data-driven approaches without modifying existing data transmission. That is, there must be at least one degree of freedom (DoF) in the target system that the attacker cannot fully control (*i.e.*, cannot precisely set the value of the manipulated data), where DoF is defined as the number of data that cannot be determined by another (set of) data and can only be obtained as control inputs or measurements. Finally, CADD is also capable of identifying the anomalous source in the presence of a component fault or a naïve attack, where the anomaly source is limited to either VB or one of the assistance data.

Specifically, this threat model assumes the adversary launches remote attacks and represents a practical real-world condition (*i.e.*, not able to fully control every DoF) based on the following facts. First, while spoofing GPS is shown to be plausible from a distant location, spoofing an accelerometer to generate a specific waveform (*i.e.*, not random values) requires a learning process for phase tuning based on the feedback from the accelerometer itself [43]. That is, targeted manipulation of an accelerometer is proven difficult without direct physical access to the vehicle's internal components. Second, while prior work [3] only showcases the possibility of data

[3] b_r is treated as optional since it is not included in the standard OBD-II message.

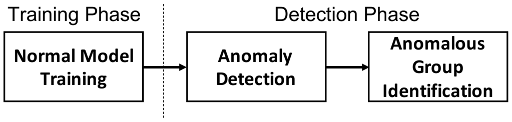


Figure 3: Basic operation of CADD.

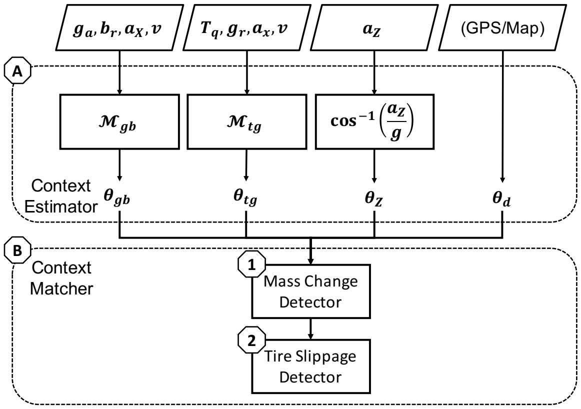


Figure 4: System structure of CADD's detection phase. The last data group can be GPS (only) or sensors with road grade.

injection/manipulation through *a single ECU* attached to the infotainment system, using a single ECU to mimic data transmission of multiple ECUs can be prevented with MI approaches [6–8]. Third, since most ECUs have limited communication capability, attackers must have physical access to those internal ECUs to manipulate their core operation. Nevertheless, we still assume a stronger attacker than the aforementioned attack examples to be able to manipulate all data but one status measurement.

4 SYSTEM OVERVIEW

CADD's basic operation consists of two phases: *training* and *detection* (Fig. 3). During training, CADD captures a vehicle's basic operation of regular driving while there is no specific constraint on how the vehicle should be maneuvered as long as 1) there is no (excessive) tire slippage, and 2) the training phase includes different speed ranges that the vehicle will normally experience. Ideally, this training needs only once for each car unless some physical modification or maintenance is made to the car. The user can also configure CADD to perform training periodically to account for component aging/wear-out. However, it is the user's choice whether to do so since a positive detection reported by CADD can also be the indication that the vehicle requires a routine maintenance. The normal VB can be described by two models, $\mathscr{M}_{gb}$ and $\mathscr{M}_{tg}$. The former captures the relationship between longitudinal acceleration and the drivers' control input (Section 5.2) while the latter captures the relationship between longitudinal acceleration and the engine torque (Section 5.3).

Fig. 4 shows the system structure for the detection phase — context estimation and matching. CADD performs the anomaly detection by estimating the features of driving contexts (*i.e.*, RI, TS, and TM) and comparing them to check if they are consistent with each other. What makes CADD special is that it utilizes those contexts that cannot be directly observed as part of the detection mechanisms, eliminating their uncertainty in modeling and even transforming them into useful information for detection. Specifically, CADD first assumes that all VB changes are caused by the road inclination (RI). So, CADD estimates RI from four perspectives independently: (i) the difference between measured (longitudinal) acceleration (a_X) and expected acceleration based on gas/brake pedal position input ($\hat{a}_{X,gb}$); (ii) the difference between a_X and expected acceleration based on engine torque and gear level ($\hat{a}_{X,tg}$); (iii) the difference between measured vertical acceleration (a_Z) and gravity (g); and (iv) RI information provided by sensors, such as GPS/map. If these four estimations of RI match each other, CADD will conclude that the vehicle behaves normally. Otherwise, CADD will check further if the change of the vehicle's behavior is caused by the other two contexts (*i.e.*, TS and TM) or the combinations of the three contexts by cross-validating the data.

If the anomalous behavior is determined not caused by any of the contextual changes, CADD will report the detection of an anomaly (along with the identified anomalous data/component group). The choice of data utilized in CADD is based on their functional roles in the speed control and their accessibility. We chose gas and brake pedal positions as they are the direct inputs from the drivers.

5 NORMAL BEHAVIOR MODEL

5.1 Fundamentals of Vehicle Dynamics

In general, there are six types of forces that can directly influence a vehicle's acceleration (Fig. 5) [27]:

- Drive force (F_E): force generated from engine torque (T_q);
- Brake force (F_B): force generated from braking;
- Aerodynamic drag (F_D): force caused by air resistance;
- Rolling drag (F_R): force required for tires to roll passively;
- Normal force (F_N): supportive force perpendicular to the contact surface; and
- Gravitational force (F_G): force caused by gravity.

Therefore, the vehicle acceleration **a** can be described by:

$$m\mathbf{a} = \mathbf{F_E} + \mathbf{F_B} + \mathbf{F_R} + \mathbf{F_D} + \mathbf{F_N} + \mathbf{F_G} + \mathbf{F_O}, \tag{1}$$

where m is the vehicle's total mass, F_O is the aggregated effect from other minor factors, and the terms in bold fonts represent vectors. If we look at the vehicle's longitudinal direction (presented by subscript X) and plug it in the detailed expression of each force based on the available DOI of CADD, then, when $|\mathbf{F_E} + \mathbf{F_B}| \leq \mu F_N$, Eq. (1) can be rewritten as:

$$\begin{aligned} m_{(\Sigma)} a_X = {} & F_E - F_B - F_R - F_D - F_{G,X} + F_{O,X} \\ = {} & \frac{T_q i_g i_f}{R_{(\Sigma)}} S_{E(\Sigma)} - b_r k_{b(\Sigma)}(v) S_{B(\Sigma)} - f_{(\Sigma)} m_{(\Sigma)} g \cos\theta \\ & - 0.5\, \sigma_{(\Sigma)} c_{(\Sigma)} A_{(\Sigma)} v^2 - m_{(\Sigma)} g \sin\theta + \psi_{(\Sigma)}, \end{aligned} \tag{2}$$

where μ is the coefficient of friction between the tires and the road surface, i_g is the gear ratio, i_f is the final drive ratio, and R is the tire radius. k_b is the braking coefficient (including the effect of slip ratio, friction coefficient, etc.), and S_E (S_B) is the adjustment on F_E (F_B) due to vehicle steering. f is the coefficient of the vehicle's rolling resistance. σ is the air density, c is the vehicle's drag coefficient, and A is the vehicle's cross-section area. Finally, ψ is the aggregated effect of other minor factors, and the terms with (Σ) indicate that

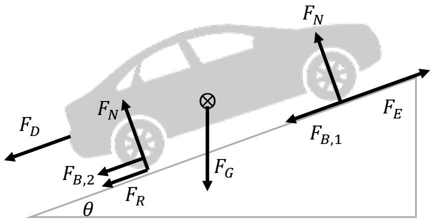


Figure 5: Forces related to vehicle acceleration.

their values can be influenced by contexts not available to CADD and may also include effects from steering as S_E does.

Next, we introduce how CADD models VB without all the detailed parameters in Eq. (2) while accounting for the reality of limited data availability. Note that the unobservable adjustments/uncertainties caused by (Σ) will be treated as (model) noise on top of their regular patterns during run-time. However, they can be incorporated if they become standardized data. To facilitate a more concise presentation, we omit (Σ) in the following discussions.

5.2 $\mathscr{M}_{gb}\{g_a, b_r, v, a_X\}$

1) Model Formulation: $\mathscr{M}_{gb}$ captures the relation between a_X and drivers' control inputs (*i.e.*, g_a and b_r). We first combine F_D, F_R, and $F_{G,X}$ together as:

$$F_R + F_D + F_{G,X} = (fmg\cos\theta + \frac{\sigma c A v^2}{2}) + mg\sin\theta \qquad (3)$$

$$\approx \underbrace{fmg + 0.5\sigma c A v^2}_{\mathscr{T}_{R,D}(v)} + \underbrace{mg\sin\theta}_{\mathscr{T}_G(\theta).} \qquad (4)$$

We can make the approximation in Eq. (4) because $\tan\theta \leq 0.07$ according to the US government's guideline for road construction [44]. We then use this function form to present $F_E = \mathscr{T}_T(T_q, g_r)$ and $F_B = \mathscr{T}_B(b_r, v)$. Since the engine torque (T_q) is controlled by the throttle while gear ratio is determined by the gear level (g_r) which is correlated with the vehicle speed (v) and throttle position (g_a), F_E can be described by a function of g_a and v. Finally, we can obtain the model formulation of $\mathscr{M}_{gb}$ by plugging Eq. (4) into Eq. (2):

$$a_X = \frac{1}{m}[\mathscr{T}_T(T_q, g_r) - \mathscr{T}_B(b_r, v) - \mathscr{T}_{R,D}(v) - \mathscr{T}_G(\theta)] \qquad (5)$$

$$= \underbrace{\mathscr{H}_T(T_q, g_r)}_{\mathscr{H}_E(g_a,v)} + \mathscr{H}_B(b_r, v) + \mathscr{H}_{R,D}(v) + \mathscr{H}_G(\theta) \qquad (6)$$

where $\mathscr{H}_i$ is the aggregated effect of $\mathscr{T}_i$ on a_X.

Note the formulation in Eq. (5) provides us the detection condition when this physical modeling is used. First, other than the control input and vehicle state, θ and m have direct impact on vehicle acceleration. Second, $\mathscr{T}_T(T_q, g_r)$ and $\mathscr{T}_B(b_r, v)$ are valid only when $|\mathbf{F_E} + \mathbf{F_B}| \leq \mu F_N$ as mentioned in Section 5.1. The above two characteristics are the reasons why CADD is designed to use RI, TS, and TM for anomaly detection.

2) Training: To simplify the model training process, we divide the $\mathscr{M}_{gb}$ model into two submodels — $\mathscr{M}_u$ and $\mathscr{M}_d$, where the former focuses on modeling the vehicle's behavior when only gas pedal is applied and the latter focuses on any other situation. Since $\mathscr{H}_G$ is not dependent on any vehicle state, we can compute its effect on a_X by training on any road with a known slope. Also, since modern automatic transmission is usually controlled based on vehicle speed (v) and throttle position (g_a) [16], the aggregated effect of $\mathscr{H}_T(T_q, g_r) + \mathscr{H}_{R,D}(v) = \mathscr{H}_E(g_a, v) + \mathscr{H}_{R,D}(v) = \Gamma(g_a, v)$ can be obtained by only considering the training data when the brake is not pressed. So, we can now express the simplified model $\mathscr{M}_u$ as:

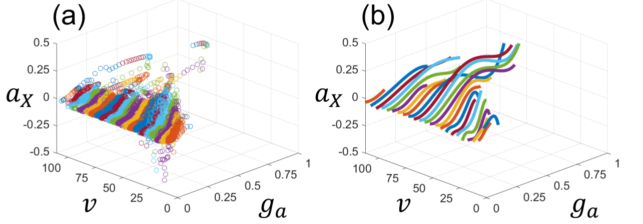


Figure 6: (a) The acceleration characteristics partitioned by v. (b) $\mathscr{M}_u$'s final model example.

$$a_X = \Gamma(g_a, v) + \mathscr{H}_G(\theta), \text{ if } b_r = 0. \qquad (7)$$

Training $\mathscr{M}_u$ is to identify Γ given the training data $\{g_a[t], v[t], a_X[t]\}$. Since the acceleration characteristics vary with vehicles, we do not use a fixed model form to approximate the function Γ. Instead, we model the VB based on the vehicle's speed. That is, we assign a function Γ_k to describe the correlation between g_a and a_X in each speed interval $v \in [v_k, v_{k+1})$, where $v_k = \delta_v \times (k-1)$ and δ_v is the size of each group (Fig. 6a). This way, we can model Γ without committing to a specific model form and achieve the flexibility of this approach. We set δ_v = 5km/h in our implementation.

Since any function can be approximated by its Taylor series expansion form, we use a polynomial approximation:

$$\Gamma_k(g_a[t]) = c_{k,0} + c_{k,1}g_a[t] + c_{k,2}g_a^2[t] + \cdots, \qquad (8)$$

where coefficients $[c_{k,0}, c_{k,1}, c_{k,2}, \cdots]^T = \mathbf{c_k}$ are determined by minimizing the loss function:

$$L_k = \Sigma_{t_0}^{t_n}(a_X[t] - \mathbf{c_k}^T\mathbf{G_a[t]})^2 + \lambda|\mathbf{c_k}|^2, \qquad (9)$$

where $v[t] \in [v_k, v_{k+1})$, $\mathbf{G_a}[t] = [1, g_a[t], g_a^2[t], \cdots]^T$, and λ is a hyperparameter parameter for preventing overfitting. Fig. 6b shows the final model form, where the lines are Γ_k in the speed interval k.

We use another training process to capture model $\mathscr{M}_d$ (*i.e.*, VB when the brake is applied):

$$a_X = \mathscr{H}_{R,D}(v) + \mathscr{H}_E(g_a, v) + \mathscr{H}_B(b_r, v) + \mathscr{H}_G(\theta) \qquad (10)$$

$$= \Gamma(g_a, v) + \mathscr{H}_B(b_r, v) + \mathscr{H}_G(\theta) \qquad (11)$$

Since $\mathscr{H}_G(\theta) = -g\sin\theta$ can be compensated by performing the training process on a road with known inclination, the main idea is to determine the parameter values under different speed conditions by approximating Eq. (11) similar to $\mathscr{M}_u$:

$$a'_X = b_{X,o,0}(v) + \sum_{i=1}^{N_g} b_{X,g,i}(v)g_a^i + \sum_{j=1}^{N_b} b_{X,b,j}(v)b_r^j \qquad (12)$$

where $a'_X = a_X - \mathscr{H}_G(\theta)$, and N_g (N_b) are the orders of Taylor expansion forms of Γ ($\mathscr{H}_B$). Theoretically, we can perform a similar procedure of $\mathscr{M}_u$ to obtain the parameters. However, because the duration of time the brake is pressed is usually only a small portion of time during driving, the amount of total training data needed to construct a usable model is huge according to our preliminary experimentation. Therefore, we treat the parameters (b_X's) as functions of v to make up for the missing scenarios in the training data instead of their direct training for every v interval as in $\mathscr{M}_u$.

CADD partitions the data into small segments by a fixed time window size (T_w). For each segment, the next step is to perform the ridge regression [45] to estimate $b_{X,o,0}$, $\{b_{X,g,n}\}$, and $\{b_{X,b,n}\}$ in this segment:

$$\hat{\mathbf{b}}_{\mathbf{k}} = \text{argmin}_{\mathbf{b}_{\mathbf{k}}} (a'_{X,k} - \mathbf{b}_{\mathbf{k}}{}^T \omega_{\mathbf{k}})^2 + \lambda|\omega_{\mathbf{k}}|^2, \tag{13}$$

where k is the index of the segment, $\mathbf{b}_{\mathbf{k}} = [b_{(X,o,0,k)}, b_{(X,g,1,k)}, \ldots, b_{(X,g,N_g,k)}, b_{(X,b,1,k)}, \ldots, b_{(X,b,N_b,k)}]^T$, $\omega_{\mathbf{k}} = [1, g_a(t_{k,1}), \ldots, g_a(t_{k,N_T}), b_r(t_{k,1}), \ldots, b_r(t_{k,N_T})]^T$, N_T is the total number of data items in the segment, $a'_{X,k} = a_{X,k} - \mathscr{H}_G(\theta_k)$, and λ is the regularization term. Finally, CADD performs regression again with $\langle v_k, \hat{\mathbf{b}}_{\mathbf{k}} \rangle$ to identify the mapping from speed to the values of parameters.

Note that our implementation uses $\mathscr{M}_u$ to model the VB when only the gas pedal is pressed, and uses $\mathscr{M}_d$ for other situations. This design is based on the fact that the formulation of $\mathscr{M}_u$ can capture more detailed acceleration behavior resulting from the pressing of gas pedal than $\mathscr{M}_d$.

One of the most important characteristics of approximation with the Taylor series expansion for $\mathscr{M}_u$ and $\mathscr{M}_d$ is that Eqs. (9) and (13) can be directly solved by ridge regression [45]. Also, according to our testing, a 3rd-order approximation with $\lambda = 1$ can already achieve significantly more accurate RI estimation (only -0.08° median error and 1.09° absolute average error) than GPS ($\geq 14°$ as will be shown in Section 7.1). While the detection capability of CADD is bounded by the largest contributor of the estimation error, CADD does not make any noticeable performance gain by increasing the order of approximation beyond 3.

5.3 $\mathscr{M}_{tg}\{T_q, g_r, v, a_X\}$

1) Model Formulation: $\mathscr{M}_{tg}$ captures the relation between engine output and a_X. When the brake is not pressed, the formulation can be directly obtained from Eq. (2) as:

$$a_X = \underbrace{\mathscr{H}_T(T_q, g_r) + \mathscr{H}_{D,R}(v)}_{\mathscr{H}_{T,g_r}(T_q, v)} + \mathscr{H}_G(\theta) \tag{14}$$

where $\mathscr{H}_{T,g_r}$ is the aggregated acceleration component contributed by engine torque at gear level g_r.

2) Training: Training $\mathscr{M}_{tg}$ is equivalent to identifying $\mathscr{H}_{T,g_r}$ for each gear level. We can further express $\mathscr{H}_{T,g_r}$ as:

$$\mathscr{H}_{T,g_r}(T_q, v) = h_{g_r,0} + h_{g_r,1} T_q + h_{g_r,2} v + h_{g_r,3} v^2. \tag{15}$$

Similarly to the training of $\mathscr{M}_{gb}$, we can utilize the ridge regression to obtain $\mathbf{h}_{\mathbf{g_r}} = [h_{g_r,0}, h_{g_r,1}, h_{g_r,2}, h_{g_r,3}]^T$ by minimizing the following loss function:

$$L_j = \Sigma_{g_r[t]=j} (a_X[t] - \mathbf{h}_{\mathbf{j}}{}^T \mathbf{Y}[t])^2 + \lambda|\mathbf{h}_{\mathbf{j}}|^2, \tag{16}$$

where j is the target gear level and $\mathbf{Y} = [1, T_q, v, v^2]^T$.

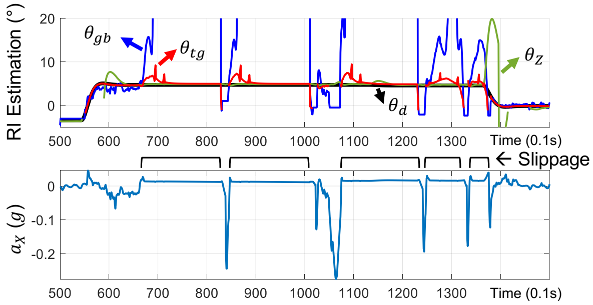


Figure 7: Example of tire slippage.

CADD's training complexity is $O(N^2)$, where N is number of data points with ≈40KB/minute (10Hz sampling rate).

5.4 RI Estimation based on $\mathscr{M}_{gb}$ and $\mathscr{M}_{tg}$

Since we can separate the acceleration contribution of RI (*i.e.*, $\mathscr{H}_G(\theta)$) from other DOI in the model formulation ($\mathscr{M}_{gb}$ and $\mathscr{M}_{tg}$), all CADD has to do for RI estimation is to compute the expected acceleration ($\hat{a}_X$) based on DOI while assuming $\theta = 0$, and take an arc sine of the difference between $\hat{a}_X$ and the received a_X divided by gravity:

$$\hat{\theta} = \sin^{-1}[(\hat{a}_X - a_X)/g]. \tag{17}$$

This estimation is valid only when TM remains the same as the training phase, and there is no tire slippage. If TM changes, say from m to m', the expected acceleration will need to be calibrated by a factor of m'/m to obtain the correct slope estimation (Eq. (2)). If there is a tire slippage, the total force can be generated from F_E and F_B will be capped at μF_N. Therefore, pressing more gas or brake pedal will not generate higher acceleration or deceleration and this will lead to a larger/smaller RI estimation magnitude than the actual value, depending on which control input is applied. CADD utilizes this characteristic in its anomaly detection to identify whether a TM change or tire slippage occurs by comparing the RI estimations from VB models with that from vertical acceleration (*i.e.*, $\theta_Z = \cos^{-1}(a_Z/g)$) and RI directly obtained from GPS/map or other sensor (θ_d). Our preliminary evaluation shows that $\mathscr{M}_{gb}$ and $\mathscr{M}_{tg}$ are able to capture the vehicle behavior accurately and produce RI estimation with merely 0.1° median error.

6 CONTEXT-AWARE ANOMALY DETECTION

6.1 Detection Procedure

As shown in Fig. 4, CADD first estimates RI utilizing the VB models while assuming the vehicle is traveling without tire slippage and the mass in the vehicle remains the same as the training phase. If the estimated RIs do not match each other, CADD further looks into the data to see if the inconsistency is the result of other contexts or their combinations. Specifically, CADD first estimates the RI from four perspectives (Section 5.4): i) θ_{gb} from $\mathscr{M}_{gb}$, ii) θ_{tg} from $\mathscr{M}_{tg}$, iii) θ_Z from a_Z, and iv) θ_d from GPS/map (Block A in Fig. 4). If the estimations match each other (*i.e.*, maximum and minimum differences of the estimations are smaller than a threshold η_{RI}), CADD will report the pass of verification and keep monitoring the DOI. Otherwise, CADD looks further into whether it is the change of TM that leads to the inconsistencies of slope estimations (Block B in Fig. 4). Note that η_{RI} is a design parameter that can be set by developers according to their preference. We will discuss more on η_{RI} in Section 7.

As discussed in Section 5.4, θ_{gb} and θ_{tg} will be different from θ_Z and θ_d if TM changes. However, θ_{gb} and θ_{tg} will match each other since they are both estimated from the difference between $\hat{a}_X$ and a_X. Therefore, CADD utilizes this characteristic to check whether $\{\theta_{gb}, \theta_{tg}\}$ and $\{\theta_Z, \theta_d\}$ form two separate groups to determine if the mismatch of RI estimation is caused by TM change. Since a mass change other than fuel consumption[4] is not likely to occur during a trip (*i.e.*, from the vehicle starts moving to the vehicle stops moving), CADD filters out the false-positives caused by the mass change due to passengers and cargo by checking whether the ratio $(a_X - g\sin\theta_{(d \text{ or } Z)})/(a_X - g\sin\theta_{(gb \text{ or } tg)})$, which is the mass ratio of the mass at the time of training to the current mass derived from Eq. (17), remains at a constant level during a trip. Note this filtering mechanism is an optional function since mass change can also be an indication of system tampering.

If RI estimations fail to match TM change, CADD further determines if there are signatures indicating tire slippage (Block B2 in Fig. 4). Specifically, CADD checks if: i) $|\theta_Z - \theta_d| < \eta_{RI}$; ii) θ_{gb}, θ_{tg} >(<) mean(θ_Z, θ_d) when gas (brake) is applied; and iii) a_X remains at the same level. These signatures are determined based on the fact that tire slippage occurs when the friction between the tires and road surface cannot provide the force required to perform intended acceleration. Therefore, the decrease of acceleration will make θ_{gb}, θ_{tg} to overestimate (underestimate) RI when gas (brake) pedal is pressed. Meanwhile, $|a_X|$ will be capped at μF_N. Fig. 7 is an example of showing the signatures of tire slippage.

6.2 Identification of Anomalous DOI

Upon detection of an anomaly, CADD further determines whether the anomaly is caused by the DOI related to VB itself or that used to estimate the context. The output of this identification can be one of the following four possibilities:

S1. a_Z is anomalous;
S2. The direct RI measurement (*i.e.*, θ_d) is anomalous;
S3. Vehicle behavior (VB) is anomalous; or
S4. All DOIs are *potentially* anomalous.

CADD identifies the source or the group of sources caused an anomaly by finding the outlier(s) in RI estimations. That is, CADD will determine whether a set of DOIs is anomalous if the RI estimation based on them deviates from other RI estimations by clustering. Specifically, CADD sorts the four estimations based on their values and computes the differences between the sorted values. CADD then finds the largest difference between the sorted estimations and divides the group into two subgroups. If there is one group (group A) that only contains one estimation and the largest difference between the estimations in the other group (group B) is less than η_{RI}, CADD will report that the DOI used to provide the estimation of group A is the potentially anomalous source. For example, the sorted estimations are 1,2,3 and 6°. The differences between the estimations are 1, 1, and 3. Since the largest difference is 3, CADD will divide the estimations into two groups, {1,2,3} and {6}. Since the maximum difference of the first group is 3-1=2≤ η_{RI}=2 (*i.e.*, the estimations in this group match each other), CADD will determine the DOI used to estimate 6° as anomalous. CADD also determines if the two subgroups are $\{\theta_{gb}, \theta_{tg}\}$ and $\{\theta_Z, \theta_d\}$, and if the differences in each group are less than η_{RI}. If yes, CADD will conclude that VB is anomalous (S3). If none of the aforementioned conditions can be found (*i.e.*, RI estimations are not consistent with each other at all), CADD will determine that all DOIs are potentially anomalous (S4).

7 EVALUATION

7.1 Experimental Setup

We utilize both i) the data from real-world driving and ii) that generated from CarSim [46] to perform two sets of evaluation. While the first set of evaluation (E_1) assesses CADD's performance when it is directly applied to basic/lower-end commodity vehicles with limited sensor support (*i.e.*, only with GPS and accelerometer), the second set (E_2) shows CADD's real potential for advanced/higher-end vehicles with access to more sensors and/or information (*e.g.*, GPS/map or inclinometer).

1) Testing Scenarios (E_1). The first set of evaluations (E_1) utilize 13 real-world driving traces we collected by OpenXC VI [47], a commercial off-the-shelf OBD-II dongle. All data are retrieved directly by using the OBD-II dongle without support from any additional sensors, which represents a basic third-party deployment scenario. One of these traces is used as training data and the rest as testing data. The training trace consists of 5.1km driving data in an urban area and the testing data include hilly/bumpy roads in freeway, urban, and downtown areas with up to 3.7km (3.8min) traveling distance (time) per trace. The vehicles are driven by 3 different drivers and the roads have up to 5° slope.[5] As mentioned, the vehicle data are directly read off from the CAN bus without any additional information provided from other devices or cloud. Therefore, the road slope (θ_d) required by CADD is computed based on the GPS elevation measurements. Naturally, the detection granularity/threshold is bounded by the resolution of input data. Note the elevation measurements from GPS only have coarse-grained accuracy [48]. In the collected traces, the maximum resolution is 3.05m (10ft) and the update rate is ≈1Hz, which can cause up to 14° error to θ_d if the vehicle is traveling at ≥45km/h on a flat road. Due to safety concerns, we did not manually induce tire slippage for E_1. Instead, we evaluate the scenarios with tire slippage in E_2.

2) Testing Scenarios (E_2). E_2 evaluation utilizes data from E_1 with CarSim simulation. We chose CarSim because it can simulate realistic vehicle behavior and road condition (*e.g.*, auto-generated pot holes or cracked/bumpy roads), and it is used by 7 major OEMs for design testing [46]. It also allows us to adjust RI and RF without relying on the weather or physical location of testing and, the most important of all, we can test dangerous scenarios (*e.g.*, experiencing excessive tire slippage) without jeopardizing driver safety. We use the driving profiles collected from E_1 to create realistic driving behaviors and the vehicle are automatically controlled by the route/speed setpoints based on CarSim's sensor simulation. See Appendix-C.1 for test case examples.

[4] Negligible during a short trip. See Appendix-A for more on this.

[5] In addition to the driver, three traces have no passenger and others have 1 passenger with a total of five ⟨driver, passenger⟩ combinations.

Thresholds →		E_1: $\eta_{RI} = 7°, 11°, 14°$	E_2: $\eta_{RI} = 1°, 2°, 3°$
Data	Unit	Behavior/Data Deviation	
Long. Acc.	$0.01g$	5, 15, 20, 25, 30	5, 6, 7, 8, 9
GPS Slope	°	8, 10, 12, 14, 16	2, 2.25, 2.5, 2.75, 3
Vert. Acc.	$0.01g$	1, 2, 5, 10, 50	1, 2, 3, 4, 5

Table 1: Testing scenarios.

Data	Unit	Magnitude of Manipulation or Deviation								
Δa_X	0.01g	5	6	7	8	9	15	20	25	30
Δv	0.01kph	18	21	25	28	32	53	71	88	106
ΔX	10^{-4}m	25	29	34	39	44	74	98	123	147

Table 2: This table shows the maximum resulting effects (per data sample taken at a 10Hz rate) of data-manipulation attacks, where Δa_X is the resulting effect magnitude, Δv and ΔX are the resulting (maximum) speed and location deviation/errors perceived by the vehicle.

We created 10 testing elevation templates. Each template has 10 segments with various RI ranging from -5 to 5° slope (≈7% grades, which is the maximum (urban) RI suggested in the road design guideline specified in Table 8-1 of [44]). The length of a segment projected on the horizontal plane is 250 to 1000m. For each template, we further set the friction coefficient μ to 0.2–0.5, and 0.9, thus creating a total of 50 RI–RF combinations as testing contexts. Together with the driving profile captured in E_1, we use CarSim for vehicle simulation. The average length of testing data is 7.5km and the traveling time is 7–8min per trace. The training data consists of 28.25km traveling distance and 41min traveling time on a separate training map (flat road with $\mu = 0.9$).

3) Noise Injection. To account for the worst-case scenario in which the vehicle has poor sensing quality or is traveling on bumpy roads (for E_2), we tested the scenarios where there can be excessive measurement (or environmental) noise by injecting AWGN noise with 63% mean and 8% median error to the collected traces (more details in Appendix-C.1).

4) Attacks. Since CADD neither combines consecutive results during detection nor relies on specific attack patterns for its detection, it is the deviation from the normal vehicle behavior (or data value) that matters to CADD's detection, not how the data (*e.g.*, by which interface) are manipulated/attacked in time domain. We consider the cases where the vehicle behavior/acceleration (a_X) and assistance data (*i.e.*, θ_d and a_Z) can all be manipulated, and see if CADD is still able to tell whether the anomaly is caused by the VB related DOI or the incorrect input from other sensors.

We assume a *strong* attacker who can launch stealthy or small enough attacks such that i) their manipulation levels are within the common permissible error and ii) they can evade the detection of prior approaches (*i.e.*, with >7s detection latency in E_1 when Δa_X = 0.3g or >1.7s in E_2 when Δa_X = 0.09g, where Δ indicates the manipulation level). Table 1 lists the manipulation scenarios and thresholds (η_{RI}) we tested for E_1 and E_2. We introduce attacks to the test-cases by shifting the data to deviate from their ground truths according to Table 1 in addition to measurement noises, *i.e.* perceived value x' = ground truth x_{gt} + noise ϕ + attack effect magnitude x_a. Therefore, the vehicle will have an incorrect knowledge of its current state after an attack is launched. Note x_a is the resulting effect of an attack, which covers any attack types that will change VB and are not only limited to spoofing a_X measurement. Table 2 further shows the attack's effects on the vehicle's perception of speed and location under different levels of manipulation tested in this section. Specifically, even if we look at the maximum a_X manipulation (0.3g), the attack will only generate a 0.0147m location deviation and a 1.06 km/h speed deviation every 0.1s while pure GPS localization is known to have meter-level error (4.9m [49]) and the European law (ECE-R39) only requires the speedometer to report with $0.1v_{GT}$ + 4 km/h error, where v_{GT} is the ground truth of the vehicle speed.

For each scenario, we generate 50 (100) test cases for E_1 (E_2) where the data manipulation starts randomly and lasts for <1min (1–5min), which is equivalent to an average of 63.9% (12.9–64.5%) traveling time. Each test-case has a different combination of i) attack target, ii) attack magnitude, iii) attack start time, iv) attack duration, v) noises, vi) road grade & condition, and vii) ground truth behavior.

5) Detection Threshold. Detection threshold (η_{RI}) also defines the minimum behavior change CADD is able to detect ($\Delta a_{X,min}$): $\Delta a_{X,min} \approx g \times \eta_{RI}$. For example, $\Delta a_{X,min}$'s of η_{RI} = 7°, 11°, and 14° are 0.12g, 0.19g, and 0.24g, respectively. However, setting a small threshold is not always a good choice since it may induce more false positives due to measurement noises. Specifically, CADD should set η_{RI} to a value higher than the average difference/error between actual road slope and the input slope data during the training phase.

6) Baseline Comparison. For baseline comparisons, we also implemented EVAD [19] and PID-Piper [21], because, to the best of our knowledge, EVAD is reported to yield the best performance and covers the most similar data types as CADD while PID-Piper is the state-of-the-art CI based on machine learning and can be adapted to cover the same detection scope of CADD. For a fair comparison, they have access to the same data types as CADD does, but they cannot access those detailed data that are not commonly available on in-vehicle networks.

While EVAD [19] has already covered similar data as CADD, we implemented the correlation pairs in EVAD that cover the available data listed in Section 3.1. For PID-Piper [21], its Feed-Forward Controller (FFC) and Feed-back Controller (FBC) are both implemented based on long short term memory (LSTM) machine learning as in [21]. Specifically, the prediction output of FFC is the vehicle's acceleration (a_X) while other available data listed in Section 3.1 are treated as its input. Similarly, the prediction outputs of FBC are the values of gas (g_a) and brake (b_r) pedals while other available data are treated as FBC's input.

7) Evaluation Metrics. We use the following metrics for evaluating the *detection* performance of CADD.

- *Detection Rate* (*True Positive Rate*): the probability that the anomaly is successfully detected: TPR = TP / (TP + FN), where T (F) is true (false) and P (N) is positive (negative).
- *False Positive Rate*: the probability that CADD identifies a normal behavior as an anomaly: FPR = FP / (FP + TN).
- *Detection Latency/Delay* (DL): the time between the anomaly occurs and the time CADD receives the first data sample that triggers a positive detection. This metric tells us how long it takes for CADD to detect an anomaly.

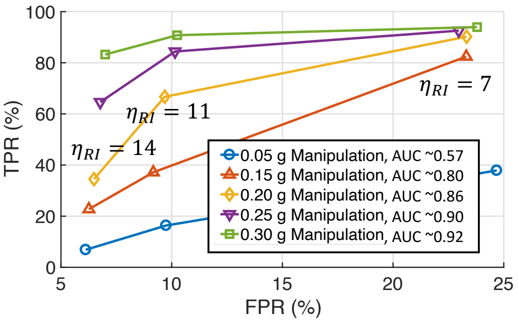


Figure 8: ROC curves with a_X manipulation in E_1, where AUC represents the area under the ROC curve.

Metric	Δa_Z (g)				
	0.01	0.02	0.05	0.10	0.50
TPR (%)	10.39	27.47	72.82	74.70	74.22
FPR (%)	6.20	6.10	6.49	6.68	6.98
DL (ms)	16,204	56,66	2,495	2,311	2,759
Metric	$\Delta\theta_d$				
	8°	10°	12°	14°	16°
TPR (%)	8.18	12.94	32.04	54.87	70.40
FPR (%)	6.28	6.56	6.12	6.08	6.17
DL (ms)	17,780	14,948	9,418	7,148	6,530

Table 3: Performance when a_Z and θ_d is manipulated in E_1.

Target	Acc_{id}	Acc_{in}	Acc_{id}	Acc_{in}	Acc_{id}	Acc_{in}
a_X	$\Delta a_X = 0.05$g		$\Delta a_X = 0.15$g		$\Delta a_X = 0.20$g	
	49.33	49.33	90.67	90.67	91.67	91.67
a_Z	$\Delta a_Z = 0.01$g		$\Delta a_Z = 0.02$g		$\Delta a_Z = 0.05$g	
	64.83	64.83	94.00	96.17	96.67	97.67
θ_d	$\Delta\theta_d = 12°$		$\Delta\theta_d = 14°$		$\Delta\theta_d = 16°$	
	30.83	30.83	66.33	66.33	76.80	76.80

Table 4: Identification performance in E_1 (η_{RI} = 14°).

Note that we use *a single data sample* as a unit to compute TPR and FPR, instead of using the entire trip/attack duration. Also, since basic CADD *does not* combine consecutive detection results for its anomaly detection, TPR also captures the probability of detecting an anomaly with a single data sample. Thus, TPR and FPR of CADD are independent of the attack duration/length.

For CADD's identification performance, we use:

- *Accuracy of Identification* (Acc_{id}): the probability of identifying the exact group of anomalous DOIs.
- *Accuracy of Inclusion* (Acc_{in}): the probability of classifying the anomalous DOI group as potentially anomalous.

7.2 Performance in Traditional Vehicles (E_1)

1) Detecting VB Change. Let us consider CADD's performance when it is directly applied to modern vehicles. We use "Δ" in front of a DOI to indicate the amount of manipulation/deviation applied to that DOI/VB. While using η_{RI} = 14° and ≤1min data manipulation as our main evaluation setting, Fig. 8 also shows the receiver operating characteristic (ROC) when η_{RI} is set to 7° and 11°. One can observe that TPRs are always higher than 60% if the manipulation is greater than the minimum detectable value derived by the detection threshold. Even though 60% might seem to be underwhelming, it is actually the result of using each sampling time as one unit to compute TPR. Since CADD generates a detection result every detection cycle, the definition of TPR presented here is equivalent to the ratio of the number of cycles CADD correctly detects an anomaly to the total number of cycles that a data manipulation is active. If we directly examine whether a single session of data manipulation can be detected before the session ends, there will be 0 false negatives, meaning that each and every manipulation can be captured by CADD.

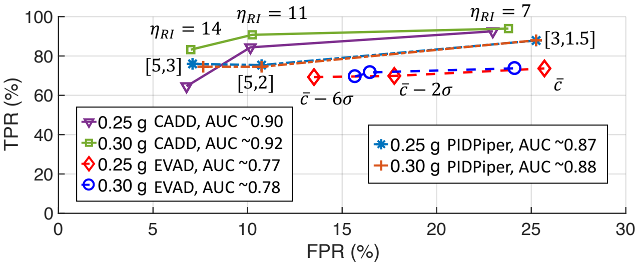


Figure 9: ROC comparison between CADD and prior studies in E_1, where the labels are the threshold settings of EVAD and PID-Piper, $\bar{c}$ (σ) is the mean (standard deviation) of correlation coefficients of data pairs utilized in EVAD, and the tuple ($[x, y]$) indicates the number of standard deviations to report (x) and cancel (y) an anomaly alarm in PID-Piper.

As mentioned in Section 7.1, since there can be ≥14° error when GPS is used to provide θ_d information, we should set η_{RI} to be ≥14°. With this setting, CADD achieved very low (~6%) FPR while achieving >80% (>60%) TPR in detecting 0.3g (0.25g) manipulations. We will later discuss how to further reduce false-positives.

2) Detecting Anomalous Assistance Data. Table 3 shows the results when a_Z and θ_d are under attacks with η_{RI} = 14°. These conditions can also simulate the situation when GPS or accelerometer is spoofed by a malicious party. The TPR remains stable once Δa_Z reaches 0.05g, indicating CADD's capability of capturing data manipulation consistently after this level. As mentioned before, since we only have coarse-grained θ_d input in both temporal and magnitude perspectives and the choice of threshold η_{RI} is bounded by the error level of this input, CADD seems to yield lower TPR. However, from the identification results in Table 4, CADD is shown to be able to achieve an excellent identification rate even if the data manipulation does not reach the theoretical detection level thanks to the low FPR. Specifically, CADD is able to achieve 90.67 and 94% rates of anomaly source identification (Acc_{id}) when Δa_X = 0.15g and Δa_Z = 0.02g, respectively.[6]

3) Baseline Comparison. We now compare the performances of CADD, EVAD, and PID-Piper (Fig. 9). CADD can achieve similar or up to 21.5% higher (absolute) TPR with only half of EVAD's FPR. The lower TPR and higher FPR of EVAD are the results of EVAD's inability to account for the influence of context when detailed measurements are not available. The results presented here also showcase a major drawback of correlation-coefficient-based approaches: *they usually do not have any efficient threshold selection*

[6] Modern sedans can achieve >0.5g acceleration (0-60 mph in 3.2s) [50].

Scenario	Δa_X: Amount of Manipulation to a_X				
	0.05g	0.15g	0.20g	0.25g	0.30g
CADD (7° ≈ 0.12g)	2,576	901	417	376	361
EVAD ($\bar{c}$)	12,943	10,027	11,001	10,422	10,217
PIDPiper ([3,1.5])	1,054	1,104	1,044	1,102	1,155
CADD (11° ≈ 0.19g)	6,282	4,272	1,652	977	559
EVAD ($\bar{c} - 2\sigma$)	18,664	11,829	11,626	12,457	11,581
PIDPiper ([5,2])	7,242	7,225	7,219	7,207	7,193
CADD (14° ≈ 0.24g)	18,076	6,573	5,571	2,171	1,628
EVAD ($\bar{c} - 6\sigma$)	19,068	13,503	12,118	12,613	12,730
PIDPiper ([5,3])	7,254	7,212	7,213	7,208	7,194

Table 5: Detection delay (ms) in E_1. Note that some test cases do not exceed the detection threshold (*i.e.*, η_{RI} or $\Delta a_{X,min}$).

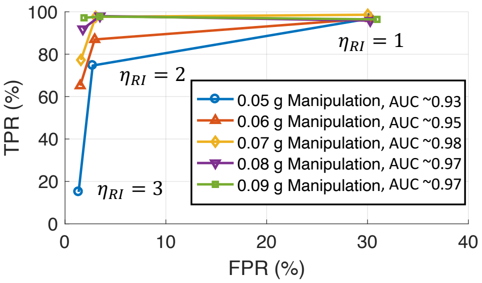


Figure 10: ROC curves without tire slippage in E_2.

mechanism since correlation coefficients are not directly linked to any cyber-physical properties of the vehicle system. That is, we cannot adjust EVAD further to achieve a higher TPR or a lower FPR while maintaining meaningful detection results (*i.e.*, TPR > FPR). On the other hand, PID-Piper is shown to achieve a similar TPR as CADD (with a 5–10% difference depending on the settings). We would like to stress again that CADD *does not* rely on consecutive results for its detection, and hence its TPRs are equivalent to the probability that CADD detects an attack based on one single data sample regardless of how attacks/manipulations are launched in time domain. However, because both EVAD and PID-Piper *do* rely on consecutive observations for their detection, their TPRs presented here are the best-case performance when there is a persistent attack lasting for a certain period of time (*i.e.*, larger than their DLs). A further look at the DLs in Table 5 reveals that they require more than 7s on average to detect an anomaly while CADD incurs ≤20% of PID-Piper's/EVAD's DL when considering the settings to achieve low FPR and $\Delta a_X \geq 0.25$g. These results indicate that a short-lived attack (<7s) will have a high probability to evade prior detections and showcase CADD's superiority to both.

7.3 Performance with θ_d Support (E_2)

We now explore CADD's performance when more accurate inclination estimation θ_d is available (with <2° error) based on CarSim. As mentioned in Section 3.1 and Fig. 4, θ_d does not have to be obtained from the elevation data of GPS (as in E_1). That is, it can also come/computed from other available sensor data, such as i) the combination of GPS and Map, ii) an inclinometer, or iii) LIDAR/camera as long as the inclination is not computed based on a subset of other data groups in Fig. 4. For example, Wang *et al.* [51] showed that Google Earth can achieve 1.32m mean absolute error in the US while Khalid *et al.* [52] showed that Google Earth can achieve as low as 0.51m mean error in some specific regions (*e.g.*, Dabaa City, Egypt), which can be translated to 2.7 and 1.05° error when the vehicle is traveling at 100km/h and 1Hz update rate, respectively. Commercial inclinometers are claimed to achieve 0.1–2° error [53–55] and slope estimation based on vision sensors (*e.g.*, camera) is shown to achieve 0.2° error [56]. Even though the above technologies may not be available to every car and everywhere on earth currently, they indicate that a more accurate θ_d (than E_1) can be obtained for CADD to realize its true potential in future.

η_{RI}	Δa_X: Amount of Manipulation to a_X				
	0.05g	0.06g	0.07g	0.08g	0.09g
1°	524	396	409	414	325
2°	1905	1034	802	745	620
3°	11597	2642	1908	1175	957

Table 6: Detection delay (ms) without tire slippage in E_2.

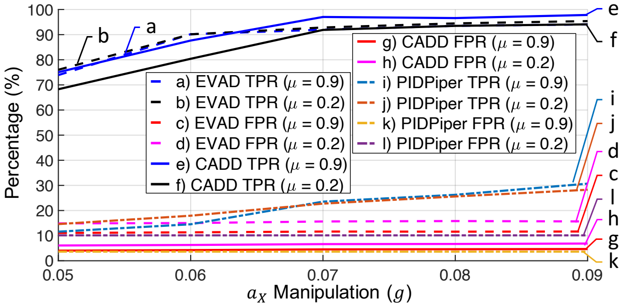


Figure 11: Comparison between CADD and prior work in E_2.

1) Without Tire Slippage. In E_2, we use η_{RI} = 2° as our target evaluation scenario based on the noise level observed in the training data. Fig. 10 shows ROC curves when η_{RI} is set to 1 – 3° (marked on each curve) and the curves indicate different levels of behavior deviation. CADD is able to achieve a high TPR (75%) even if the deviation (Δa_X) is only 0.05g and its FPR is only 4% (η_{RI} = 2°). Note that we injected a large amount of noise (*i.e.*, 13% of data have larger than 50% noise in our test cases), which will inevitably generate false positive detection. CADD can efficiently detect any behavior deviation greater than 0.07g with 97% TPR while the mean DL is less than 1s. Developers can plot the ROC curve (from the training data) as a guideline for setting η_{RI}. For example, if we assume Fig. 10 were the ROC obtained from training phase and a developer wishes to detect any behavior deviating from its norm by 0.06g with >60% TPR and <2% FPR, then s/he should set η_{RI} to 3°.

2) In the Presence of Tire Slippage. Let us consider CADD's detection performance in the presence of tire slippage (TS), simulating the vehicle traveling under different weather conditions that can cause different levels of road friction. Table 7 shows CADD's performance when μ is set to 0.2–0.5 and 0.9, where 0.9 is the road condition we used in Section 7.3.1. The average time of TS (in percentage of the entire trip) is also listed in Table 7. We can observe that CADD's performance is almost identical to the condition without TS when there is no excessive TS during the trip (*i.e.*, μ = 0.3–0.5). Even when there is up to 34% of TS time (μ = 0.2), CADD is able to achieve ~92% of TPR and only ~7% of FPR when the Δa_X =

Metric	Δa_X: Amount of Manipulation to a_X				
	0.05g	0.06g	0.07g	0.08g	0.09g
	μ = 0.9 (No Tire Slippage)				
TPR (%)	75.05	87.65	97.03	96.59	97.83
FPR (%)	4.02	4.24	4.53	4.60	4.67
DL (ms)	1,905	1,034	802	745	620
	μ = 0.5 (< 1% Slippage Time)				
TPR (%)	74.90	87.63	96.98	96.36	96.12
FPR (%)	4.07	4.31	4.48	4.75	4.72
DL (ms)	1,680	1,124	895	746	771
	μ = 0.4 (< 1% Slippage Time)				
TPR (%)	75.02	87.50	97.94	97.77	96.96
FPR (%)	4.09	4.34	4.61	4.61	4.78
DL (ms)	1,663	1,168	799	764	613
	μ = 0.3 (< 1% Slippage Time)				
TPR (%)	75.11	87.48	97.07	96.97	96.68
FPR (%)	4.28	4.47	4.70	4.79	5.06
DL (ms)	1,608	1,032	863	712	786
	μ = 0.2 (5 ~ 34% Slippage Time)				
TPR (%)	68.27	80.39	91.88	93.49	94.09
FPR (%)	6.08	6.29	6.63	6.69	6.85
DL (ms)	2,027	1,663	1,315	1,226	998

Table 7: Detection performance in E_2 (a_X anomalous).

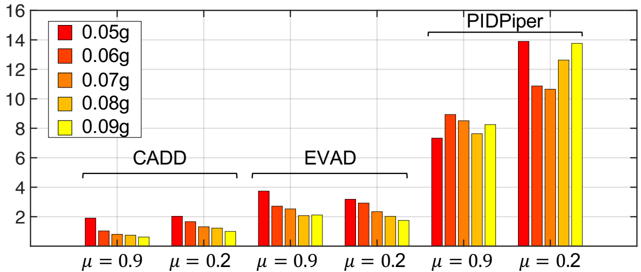


Figure 12: Detection latency (s) of CADD and prior work when Δa_X = 0.05~0.09g in E_2. Accurate values in Appendix-C.

0.07g. That is, CADD is able to significantly reduce the occurrence of false-positive detections (from potentially up to 34%) to only 7%. The slightly lower TPR than the scenario without TS (μ = 0.9) is due to TS's cancellation of the effect of an a_X change. When a_Z or θ_d is anomalous (Table 8)[7], CADD is able to achieve almost identical TPRs and FPRs with or without TS. That is, CADD is able to distinguish TS from the actual anomaly and output the correct results. Since CADD does not combine consecutive results for its detection, the performance of traveling on a road with changing μ can be approximated by considering small road segments with different μ.

Next, we compare CADD's FPR performance with EVAD by adjusting EVAD's threshold settings to have the same level of TPR as CADD (Fig. 11). Similar to the results shown in E_1, CADD's FPR is only 34.5–38.5% of that of EVAD regardless of (non)occurrences of tire slippage. For PID-Piper, it can only achieve <30.7% TPR when having a similar FPR (~4%) as CADD (Fig. 11). If we adjust PID-Piper's settings to achieve similar TPR (>90% when $\Delta a_X \geq$ 0.07g) as CADD, its FPR increases to >90%. Even for given special training data with 10 different RI conditions, PID-Piper still suffers a 66.95% FPR. This result further shows a major drawback of a ML-based approach without physical modeling: its detection will be ineffective when operating in a constantly changing environment because of the training difficulty and the lack of necessary data access. On the other hand, since CADD's and EVAD's detection is based on the vehicle's physical model and data correlation, there will be no significant performance degradation under such a condition. Furthermore, CADD is shown to achieve <51% of EVAD's/PID-Piper's DL thanks to CADD's model-based detection mechanism (Fig. 12).

	Δa_Z (0.01g)					$\Delta \theta_d$ (°)				
	1	2	3	4	5	2	2.25	2.5	2.75	3
	μ = 0.9 (No Tire Slippage)									
TPR	87.8	88.6	91.2	92.7	92.7	78.7	95.5	98.0	98.6	99.1
FPR	4.5	4.7	4.7	4.8	4.7	3.6	3.5	3.6	3.7	3.6
DL	7.38	7.54	7.21	7.41	7.44	0.79	0.26	0.07	0.08	0.02
	μ = 0.2 (5 ~ 34% Slippage Time)									
TPR	88.3	89.1	91.1	93.1	93.3	79.1	96.8	98.4	98.9	99.0
FPR	7.0	7.0	7.1	7.1	7.2	5.7	5.7	5.8	5.8	5.9
DL	7.58	7.16	7.32	7.71	7.47	0.57	0.15	0.09	0.07	0.06

Table 8: Detection results of E_2 (a_Z or θ_d anomalous), where TPR, FPR, and DL are presented in %, %, and s, respectively.

Ano	AD	μ = 0.9					μ = 0.2				
Dev. Lvl.		1	2	3	4	5	1	2	3	4	5
a_X	5	100	100	100	100	100	97.5	99.5	99.5	100	100
	2.5	94.4	97.0	97.7	99.3	99.1	89.1	95.2	96.8	97.1	99.5
	1	90.8	96.5	98.8	98.5	99.0	97.5	98.0	99.5	100	99.5
a_Z	5	100	100	100	100	100	98.9	100	100	100	100
	2.5	100	100	100	100	100	91.8	92.2	93.2	93.0	90.0
	1	97.5	95.8	98.5	94.8	96.0	62.7	62.5	67.5	68.2	62.0
θ_d	5	100	100	100	100	99.9	95.0	99.2	99.4	98.9	98.2
	2.5	93.4	96.5	94.5	93.1	90.7	75.7	85.8	85.4	82.5	81.3
	1	81.2	85.5	84.0	80.2	79.5	55.4	63.9	67.0	63.5	62.7

Table 9: CADD's source identification results (Acc_{id}) in E_2. See Section 7.3.3 for detailed description.

3) Identifying an Anomalous Group. We now look at CADD's end-to-end performance in not only detecting the occurrence of an anomaly but also pinpointing the anomalous group. First, we consider the condition in which a_X is anomalous as in Section 7.3.1. We then test the conditions in which a_Z and θ_d are anomalous, simulating the scenario that the sensor data used for context estimations are faulty. The purpose of this evaluation is to see whether CADD can correctly identify the group of anomalous DOIs. In this evaluation, we set CADD to output an identification result when each trip ends. This identification result is determined by taking the majority of anomalous groups captured in the identification process (*i.e.*, S1–S4 in Section 6.2) within a single trip. Table 9 summarizes the results of identifying an anomalous group. Since the results of μ = 0.3–0.9 are, in general, identical, we only present the results of μ = 0.9 and 0.2 here. The "Ano" column is the anomalous DOI, and the "AD" column is the anomalous duration in minutes. We tested the cases with 1, 2.5, or 5min AD, which translate to 12.9, 32.3, and 64.5% anomalous duration ratio (ADR), respectively. The second row (Dev. Lvl.) of the table indicates the level of deviation (k) from the ground truth. The actual deviation is given by Δa_X = [0.04 + 0.01k]g, Δa_Z = [0.01k]g, and $\Delta \theta_d$ = [1.75 + 0.25k]°. While $|Acc_{in} - Acc_{id}| < 0.1$ holds for all the scenarios we have tested, indicating the rare occasion of CADD's inability to determine an anomalous DOI, only Acc_{id}'s are shown here.

[7] We omit the results when μ= 0.3–0.5 in these two tables because they are almost identical to μ=0.9 as in Table 7.

When μ = 0.9 (*i.e.*, without TS) and a_X or a_Z is anomalous, CADD is able to identify the exact source of anomaly with ≥94.4% Acc_{id} if $ADR \geq 32.3\%$. Even if $ADR = 12.9\%$, CADD can still achieve ≥90.8% Acc_{id}. These results show CADD to be able to identify the anomalous source even when the anomaly lasts only for 1 min ($ADR = 12.9\%$). Compared to the condition without TS, CADD has lower Acc_{id}, especially in view of the performance when a_Z or θ_d is anomalous. The lower Acc_{id} is the result of CADD's tendency to determine the source of anomaly as a_X when TS occurs. That is, CADD won't be able to distinguish whether the anomaly is caused by anomalous a_X or the other two DOIs, and it will determine the source to be a_X because $\Delta\theta_{gb}$ and $\Delta\theta_{tg}$ caused by TS will be larger than $\Delta\theta_d$ in our testing scenarios, where $\Delta\theta_X$ is the deviation of estimation θ_X from the ground truth. Nevertheless, CADD is still able to achieve ≥75.7% Acc_{id} when $ADR \geq 32.3\%$ and $\Delta\theta_d = 2°$.

4) Discussion. While most reported false-positives are caused by transient noise, CADD's FPR can be reduced further by a false-positive filtering mechanism (FPFM) even though CADD can already achieve low FPR under excessive noisy conditions. Specifically, since any effective attack must last for a certain period of time to pass through a low pass filter (*i.e.*, a common practice of data/signal processing), CADD can choose to report an anomaly detected only when there are α consecutive positive detections, where α is a design parameter to balance between the FPR and the detection delay. CADD is able to achieve merely 0.5% (1.0%) FPR and 96% (96.8%) TPR when α = 50 (30) even with excessive noise injected (See Appendix-B for more analysis).

8 DISCUSSION

8.1 Environmental Influences and Latency

CADD's detection performance is bounded by the quality of its sensor measurement. More specifically, since CADD compares the RI estimations from different data groups, the detection capability will be bounded by data groups that generate the largest noise (*e.g.*, GPS in E_1). This characteristic also indicates that the surrounding environment can influence the CADD's performance. For example, when operating at a location with poor GPS accuracy, the data group that utilizes GPS may be more unreliable than other data groups. CADD may report a data anomaly for GPS even in the absence of an actual attack. However, this detection can also be useful to let the consumer of CADD's results know that GPS is not reliable at that moment and should not be used for making any critical decision.

Detection latency can also be affected by the data quality. For example, in Table 3, it takes CADD ≈ 17.8s to detect a data anomaly when $\Delta\theta_d = 8°$ while the normal operation can generate a 14° error. That is, any attack that stays below the normal error range can take longer for CADD to detect. However, the vehicle control should be designed to tolerate the error below its normal level of error in the first place. Therefore, having a longer detection time will not diminish the value of CADD in such a scenario. CADD in that case acts as a data observer to alert users on potential minor data anomalies.

8.2 Resiliency against Full-Scale Attacks

As mentioned in Section 3, CADD cannot detect a full-scale data manipulation (*i.e.*, all data are under the attacker's control), which is a common characteristic of all approaches without the data of final control output/setpoint (or a trusted data source). However, when map support is also available to the vehicle, a real-time full-scale attack will be very difficult to evade CADD's detection for the following reasons. To generate a full-scale attack in real time, it requires the attacker to change the road inclination context θ_d utilized by CADD to match other data after manipulation. With map support, θ_d will not be directly computed based on the elevation measurements embedded in the GPS data on IVN. Instead, CADD will use the geo-coordinates to look up θ_d on the map, meaning that the attacker will need to find a series of geo-coordinates with exactly same θ_d's that match the manipulated data (even if such a series of geo-coordinates exists). The above observations indicate that launching a full-scale attack requires careful planning and accurate control over the data values, limiting the applicable scenario and scalability of such an attack.

On the other hand, to launch a replay attack or a pre-computed attack, unless *every* data in the beginning of the recorded trace used by the attacker exactly matches *every* data of CADD when the attack is launched, there will be value gaps between before and after the replay attack is launched, thus leaving an obvious "footprint" of data manipulation.

8.3 Steep Slope

While the approximation in Eq. (4) assumed that the road grade or $\tan(\theta)$ is usually less than 7%, what would be the impact of this assumption if the vehicle runs on a steeper road of 40% grade like in San Francisco? This steeper road grade will result in the rolling drag F_R to be overestimated by $7\% \approx 1 - \cos(\tan^{-1} 0.4)$ or up to 0.5° estimation error for RI with the rolling resistance coefficient $f = 0.13$ [27]. It also shows that CADD's performance (*i.e.*, minimum detectable difference) is bounded by this estimation error $\sin^{-1}[f(1-\cos\theta)]$ in the case of steep roads. While the error of this extreme example is already much smaller than the 14° RI estimation error from GPS (E_1), improving our model to account for steep roads is part of our future inquiry.

9 CONCLUSIONS

We have presented CADD, a practical vehicle anomaly detection system that accounts for limited data availability. CADD utilizes 4 combinations of data for detection of context information and determines whether an anomaly has occurred by comparing the context estimations. Our extensive evaluation (87,000+ test cases, including trace- and simulation-based evaluations) has shown CADD to achieve high (>96%) TPR and low FPR (<0.5%) even in the presence of excessive measurement noise. CADD can also identify the anomaly source with accuracy of >95% even if the anomalous duration is only 32.3% of the observation time (Δa_X = 0.07g). Even when compared to the prior work with the best reported performance, CADD (without applying any false positive filtering mechanism) is shown to achieve not only less than 40% of its FPR but also less than 51% required time to report anomalies.

ACKNOWLEDGEMENT

The work reported in this paper was supported in part by the US Office of Naval Research under Grant No. N00014-22-1-2622.

REFERENCES

[1] Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, Tadayoshi Kohno, Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, and Stefan Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, 2010.

[2] Ishtiaq Rouf, Rob Miller, Hossen Mustafa, Travis Taylor, Sangho Oh, Wenyuan Xu, Marco Gruteser, Wade Trappe, and Ivan Seskar. Security and privacy vulnerabilities of in-car wireless networks: a tire pressure monitoring system case study. *Proceedings of the 19th USENIX conference on Security*, 2010.

[3] Charlie Miller and Chris Valasek. Remote Exploitation of an Unaltered Passenger Vehicle. In *Black Hat USA*, 2015.

[4] Michael Muter, Andre Groll, and Felix C. Freiling. A structured approach to anomaly detection for in-vehicle networks. In *2010 Sixth International Conference on Information Assurance and Security*, Aug 2010.

[5] Michael Muter and Naim Asaj. Entropy-based anomaly detection for in-vehicle networks. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, Jun 2011.

[6] Kyong-Tak Cho and Kang G. Shin. Fingerprinting Electronic Control Units for Vehicle Intrusion Detection. *USENIX Security Symposium*, 2016.

[7] Omar Y. Al-Jarrah, Carsten Maple, Mehrdad Dianati, David Oxtoby, and Alex Mouzakitis. Intrusion Detection Systems for Intra-Vehicle Networks: A Review. In *IEEE Access*, volume 7, 2019.

[8] Wufei Wu, Renfa Li, Guoqi Xie, Jiyao An, Yang Bai, Jia Zhou, and Keqin Li. A survey of intrusion detection for in-vehicle networks. *IEEE Transactions on Intelligent Transportation Systems*, 21(3), Mar 2020.

[9] Ming Yu and Danwei Wang. Model-based health monitoring for a vehicle steering system with multiple faults of unknown types. *IEEE Transactions on Industrial Electronics*, 61(7), 2014.

[10] Z. Gao, S. X. Ding, and Y. Ma. Robust fault estimation approach and its application in vehicle lateral dynamic systems. *Optimal Control Applications and Methods*, 28(3), May 2007.

[11] Shai A. Arogeti, Danwei Wang, Chang Boon Low, and Ming Yu. Fault detection isolation and estimation in a vehicle steering system. *IEEE Transactions on Industrial Electronics*, 59(12), 2012.

[12] J. Gerler, M. Costin, Xiaowen Fang, Z. Kowalczuk, M. Kunwer, and R. Monajemy. Model based diagnosis for automotive engines-algorithm development and testing on a production vehicle. *IEEE Transactions on Control Systems Technology*, 3(1), Mar 1995.

[13] Qiao Sun. Sensor fusion for vehicle health monitoring and degradation detection. In *Proceedings of the Fifth International Conference on Information Fusion. FUSION 2002.*, volume 2, 2002.

[14] Andrzej Puchalski. A technique for the vibration signal analysis in vehicle diagnostics. *Mechanical Systems and Signal Processing*, May 2015.

[15] S.E. Muldoon, M. Kowalczyk, and J. Shen. Vehicle fault diagnostics using a sensor fusion approach. In *Proceedings of IEEE Sensors*, volume 2, 2002.

[16] Lu Xi, Xu Xiangyang, and Liu Yanfang. Simulation of Gear-shift Algorithm for Automatic Transmission Based on MATLAB. In *2009 WRI World Congress on Software Engineering*, 2009.

[17] Kyong-Tak Cho, Kang G. Shin, and Taejoon Park. CPS approach to checking norm operation of a brake-by-wire system. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, 2015.

[18] Arun Ganesan, Jayanthi Rao, and Kang Shin. Exploiting Consistency Among Heterogeneous Sensors for Vehicle Anomaly Detection. In *SAE*, 2017.

[19] Fei Guo, Zichang Wang, Suguo Du, Huaxin Li, Haojin Zhu, Qingqi Pei, Zhenfu Cao, and Jianhong Zhao. Detecting Vehicle Anomaly in the Edge via Sensor Consistency and Frequency Characteristic. In *IEEE Transactions on Vehicular Technology*, volume 68, Jun 2019.

[20] Armin Wasicek and Andre Weimerskirch. Recognizing Manipulated Electronic Control Units. In *SAE*, Apr 2015.

[21] Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman. PID-Piper: Recovering Robotic Vehicles from Physical Attacks. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, 2021.

[22] High Speed CAN - ISO 11898-2:2016. https://www.iso.org/standard/67244.html. Accessed: 2020-09-13.

[23] CAN-FD - ISO 11898-1:2015. https://www.iso.org/standard/63648.html. Accessed: 2020-09-13.

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

[25] Raul Quinonez, , Jairo Giraldo, Luis Salazar, Santa Cruz, and Erick Bauman. SAVIOR: Securing Autonomous Vehicles with Robust Physical Invariants. In *29th USENIX Security Symposium*, 2020.

[26] Harsha Kumara, Kalutarage M Omar Al-Kadri, Madeline Cheah, Garikayi Madzudzo, M Omar Kalutarage, Madeline Al-Kadri, and Garikayi Cheah. Context-aware Anomaly Detector for Monitoring Cyber Attacks on Automotive CAN Bus. 19, 2019.

[27] Stephen Mangan and Jihong Wang. Development of a Novel Sensorless Longitudinal Road Gradient Estimation Method Based on Vehicle CAN Bus Data. *IEEE/ASME Transactions on Mechatronics*, 12(3), 2007.

[28] Muhammad Nasiruddin Mahyuddin, Jing Na, Guido Herrmann, Xuemei Ren, and Phil Barber. Adaptive Observer-Based Parameter Estimation With Application to Road Gradient and Vehicle Mass Estimation. *IEEE Transactions on Industrial Electronics*, 61(6), Jun 2014.

[29] Jens Jauch, Johannes Masino, Tim Staiger, and Frank Gauterin. Road Grade Estimation With Vehicle-Based Inertial Measurement Unit and Orientation Filter. *IEEE Sensors Journal*, 18(2), Jan 2018.

[30] Steffen Muller, Michael Uchanski, and Karl Hedrick. Estimation of the Maximum Tire-Road Friction Coefficient. *Journal of Dynamic Systems, Measurement, and Control*, 125(4), Jan 2004.

[31] Matthijs Klomp, Yunlong Gao, and Fredrik Bruzelius. Longitudinal velocity and road slope estimation in hybrid electric vehicles employing early detection of excessive wheel slip. *Vehicle System Dynamics*, 52, May 2014.

[32] Havard Fjaer Grip, Lars Imsland, Tor A. Johansen, Jens C. Kalkkuhl, and Avshalom Suissa. Estimation of road inclination and bank angle in automotive vehicles. In *2009 American Control Conference*. IEEE, 2009.

[33] Zhiwei Gao, Carlo Cecati, and Steven X. Ding. 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.

[34] Danfeng (Daphne) Yao, Xiaokui Shu, Long Cheng, and Salvatore J. Stolfo. Anomaly Detection as a Service: Challenges, Advances, and Opportunities. *Synthesis Lectures on Information Security, Privacy, and Trust*, 9(3), Oct 2017.

[35] Yilin Mo and Bruno Sinopoli. Secure control against replay attacks. *2009 47th Annual Allerton Conference on Communication, Control, and Computing, Allerton 2009*, 2009.

[36] Dario Stabili, Raffaele Romagnoli, Mirco Marchetti, Bruno Sinopoli, and Michele Colajanni. A multidisciplinary detection system for cyber attacks on powertrain cyber physical systems. *Future Generation Computer Systems*, 2023.

[37] Lei Xue, Yangyang Liu, Tianqi Li, Kaifa Zhao, Jianfeng Li, Le Yu, Xiapu Luo, Yajin Zhou, and Guofei Gu. Said: State-aware defense against injection attacks on in-vehicle network. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 1921–1938. USENIX Association, 2022.

[38] Cheng Feng, Venkata Reddy Palleti, Aditya Mathur, and Deeph Chana. A Systematic Framework to Generate Invariants for Anomaly Detection in Industrial Control Systems. In *Network and Distributed Systems Security (NDSS) Symposium*, 2019.

[39] Wissam Aoudi, Mikel Iturbe, and Magnus Almgren. Truth will out: Departure-based process-level detection of stealthy attacks on control systems. In *Proceedings of the ACM Conference on Computer and Communications Security*, Oct 2018.

[40] David I. Urbina, Jairo Giraldo, Alvaro A. Cardenas, Nils Ole Tippenhauer, Junia Valente, Mustafa Faisal, Justin Ruths, Richard Candell, and Henrik Sandberg. Limiting the impact of stealthy attacks on Industrial Control Systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, Oct 2016.

[41] SAE J1979A: E/E Diagnostic Test Modes - SAE International. https://www.sae.org/standards/content/j1979_201702/. Accessed: 2020-07-24.

[42] GitHub - commaai/opendbc: democratize access to car decoder rings. https://github.com/commaai/opendbc. Accessed: 2020-07-24.

[43] Timothy Trippel, Ofir Weisse, Wenyuan Xu, Peter Honeyman, and Kevin Fu. WALNUT: Waging Doubt on the Integrity of MEMS Accelerometers with Acoustic Injection Attacks. In *2nd IEEE European Symposium on Security and Privacy*, Jun 2017.

[44] American Association of State Highway and Transportation Officials. *Table 8-1 of A policy on geometric design of highways and streets.* 2018.

[45] Christopher M. Bishop. *Pattern recognition and machine learning.* Springer, 2006.

[46] Mechanical Simulation. Carsim. https://www.carsim.com/products/carsim/index.php. Accessed: 2019-10-04.

[47] OpenXC. OpenXC Vehicle Interface Reference Design. http://vi.openxcplatform.com/. Accessed: 2020-11-23.

[48] U.S. Geological Survey. U.S. Geological Survey - Global Positioning Application and Practice. https://water.usgs.gov/osw/gps/. Accessed: 2020-01-15.

[49] Official U.S. government information about the Global Positioning System (GPS) and related topics. https://www.gps.gov/systems/gps/performance/accuracy/. Accessed: 2021-07-31.

[50] Tesla. Tesla Model 3 . https://www.tesla.com/model3. Accessed: 2020-09-14.

[51] Yinsong Wang, Yajie Zou, Kristian Henrickson, Yinhai Wang, Jinjun Tang, and Byung-Jung Park. Google Earth elevation data extraction and accuracy assessment for transportation applications. *PLoS ONE*, 12(4), Apr 2017.

[52] Dr Khalid and L A El-Ashmawy. Investigation of the Accuracy of Google Earth Elevation Data. *Artificial Satellites*, 51(3), 2016.

[53] TE Connectivity. Using Sensor Fusion to Improve the Performance of Tilt Sensors White Paper. https://www.te.com/usa-en/industries/sensor-solutions/insights/tilt-sensors-white-paper.html. Accessed: 2021-07-30.

[54] Tilt-Tech CC. Tiltmeters and Inclinometers. https://tilt-tech.co.za/T-TNEW/products/tilt-meters-inclinometers/. Accessed: 2021-07-30.

[55] Engineering360. Inclinometer. https://www.globalspec.com/industrial-directory/inclinometer. Accessed: 2021-07-31.

[56] Eser Ustunel and Engin Masazade. Vision-based road slope estimation methods using road lines or local features from instant images. *IET Intelligent Transport Systems*, 13(10), Oct 2019.

Appendix A DISCUSSION

A.1 Deployment, Application, and Limitation

CADD is designed with easy deployment in mind and all the required data can be obtained entirely from the vehicle itself, or from the combination of the vehicle and an external device equipped with GPS (or inclinometer) and IMU (*e.g.*, a smartphone). For a car maker, there will be no problem retrieving all the DOI of CADD from IVN most of the time and, therefore, CADD's implementation does not need any extra sensors to be installed nor major modification to existing ECU/IVN architecture. The above characteristic is important because while CAN, with limited bandwidth and priority-based protocol, is still the de facto standard in-vehicle network (at least in foreseeable future), introducing new data on CAN may require multiple ECU modifications for corresponding priority assignment of CAN messages. Since not all ECUs are connected to the same CAN (sub)network, it may also require an architectural change to the vehicle to access the detailed data from different (sub)networks.

While implemented as a third-party solution (*e.g.*, a dongle plugged in the OBD-II port), the data required by CADD may not be accessible from the standard OBD-II messages (*e.g.*, acceleration). Therefore, an extra IMU may be required in this implementation. Since most of the OBD-II dongles provided by insurance companies already have built-in IMUs, they will not incur additional hardware cost to add CADD in their dongle.

As mentioned in Section 1, the major application of CADD is an early warning system to notify vehicle owners/managers of potential system anomalies (including both faults and attacks) to enhance driving safety. That is, CADD's current design under the assumption of limited data availability is to identify any early sign of anomaly and warn/advise the users to perform further inspection in order to prevent a more severe, safety-critical situation in the near future. This type of safety enhancement does not necessarily have to be made in real time and CADD may not directly interact with the controllers for changing their control decisions.

The other potential application of CADD is to detect modifications to driving traces for an application (*e.g.*, usage-based insurance). Even if the IMU (*i.e.*, a_X readings) of an external device could be easier to spoof (than the in-vehicle IMU), a_X can also be derived from vehicle speed (which is usually computed from the vehicle's wheel speed and the slip ratio in the vehicle internally) and the detection system can cross-validate the two readings to check if any of them is compromised. That is, the combination of in-vehicle data and the (external) IMU readings will further raise the level of difficulty to deceive the application.

A.2 Consideration of Other Context (Σ)

A.2.1 Steering. As part of our future work, we discuss how to improve our model formulation if more data types become available to CADD in the future. First, we look at the adjustments that can be influenced by vehicle steering (*i.e.*, $S_{E(\Sigma)}$ and $S_{B(\Sigma)}$ in Eq. (2)). Specifically, since the force generated by wheel torque will now contribute to both longitudinal and lateral acceleration when the vehicle has lateral movement, $S_{E(\Sigma)}$ and $S_{B(\Sigma)}$ will be dependent on the steering angle (*i.e.*, the angle ϕ_T between the tire direction and the vehicle's body direction) which is determined by the *steering wheel* angle (ϕ_s) and its corresponding steering ratio $k(\phi_s)$[8] :

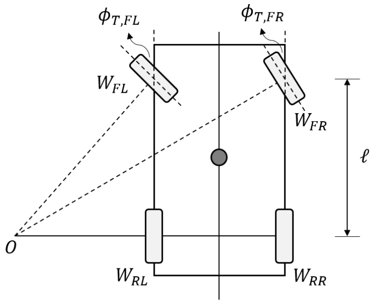


Figure 13: Example of steering angles (Ackermann steering geometry).

$$\phi_T = \phi_s / k(\phi_s). \tag{18}$$

$S_{E(\Sigma)}$ ($S_{B(\Sigma)}$) can then be presented as a function of ϕ_s as $S_{E(\Sigma)}(\phi_s)$ ($S_{B(\Sigma)}(\phi_s)$). The reason for utilizing ϕ_s instead of ϕ_T is: i) the former is more commonly available on IVN in drive-by-wire vehicles and ii) modern vehicles usually have different ϕ_T's between tires to maintain a consistent center of turning circle (O) as shown in Fig. 13. Note that we preserve (Σ) in the presentation since there are still other factors that can influence S_E, such as sideslip caused by the deformation of the tires. For a generic steering design, CADD can utilize the training data when the vehicle is on a straight road to obtain $\mathscr{M}_{gb}$ and $\mathscr{M}_{tg}$ without the adjustment of $S_{E(\Sigma)}$ and $S_{B(\Sigma)}$, and then performs training with data when $\phi_s \neq 0$ to obtain $S_{E(\Sigma)}$ and $S_{B(\Sigma)}$. However, to account for a more sophisticated steering assistance system, CADD can utilize a similar training procedure of $\mathscr{M}_{gb}$ as described in Section 5.2 that partitions the training data according to both v and ϕ_s (instead of just v) to capture the steering adjustment in different driving scenarios.

A.2.2 Fuel Consumption and Mass Change. While fuel level is *not* a common data that can be easily retrieved from IVN, CADD by default treats the mass change due to fuel consumption as a model noise. This assumption is grounded on the fact that modern (medium size) vehicles usually weigh more than 3000lbs (1360.7kg)[9] and the average fuel consumption is 30.8mpg (miles per gallon)[10]. That is, there will be only 0.2% mass change for every 30.8 miles (49.57km) of travel. However, if fuel-level information is available, it can be accounted for by Mass Change Detector.

Appendix B FALSE-POSITIVE FILTERING MECHANISM

As mentioned in Section 7, CADD can employ a false-positive filtering mechanism (FPFM) to further enhance its detection performance. Since the false-positive detection is usually caused by transient environmental noise (*e.g.*, travelling through a pothole), CADD can choose to report the anomaly only if it lasts α detection cycles, instead of reporting it immediately upon each detection. The reason for using FPFM to enhance detection performance in practical

[8] Luyang Liu, Hongyu Li, Jian Liu, Cagdas Karatas, Yan Wang, Marco Gruteser, Yingying Chen, and Richard P. Martin. 2017. BigRoad: Scaling Road Data Acquisition for Dependable Self-Driving. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys '17).

[9] How does NHTSA categorize vehicles? (FAQ-06). https://www.nhtsa.gov/ratings.

[10] Highlights of the Automotive Trends Report. https://www.epa.gov/ automotive-trends/highlights-automotive-trends-report.

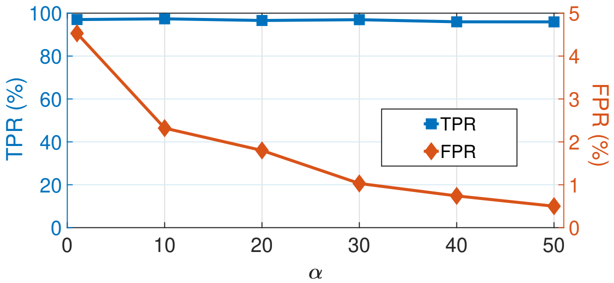


Figure 14: CADD's performance after applying FPFM.

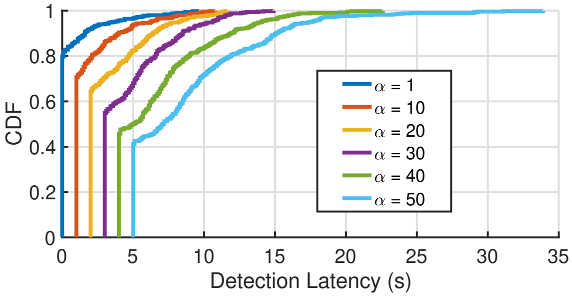
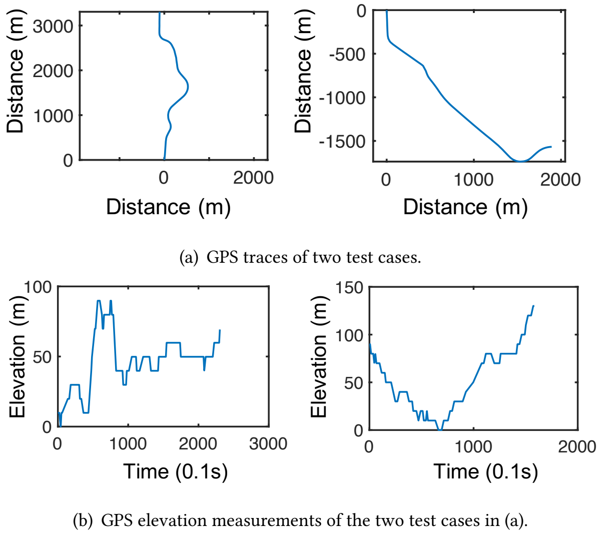


Figure 15: CDFs of CADD's detection latency with FPFM.

scenarios is the fact that the data covered by CADD are continuous data/measurements, not sporadic/discrete commands, which should (and will) go through a low pass filter before being consumed by any other subsystems. So, any "successful" attack must last for a certain amount of time. Fig. 14 shows CADD's detection performance when $\Delta a_X = 0.07g$ under different α values. We can observe that FPFM can effectively reduce FPR (from 4.5% to 0.5%) when $\alpha = 1 \rightarrow 50$ detection cycles. Furthermore, this makes no significant impact on overall TPR (from 97.0% to 96.0%), proving its effectiveness in enhancing CADD's performance. Since FPR is independent of the manipulation/attack level, there is no need to adjust α according to the target detection level. However, a larger α incurs a larger detection delay while providing a better FPR performance. Specifically, Fig. 15 showcases CADD's detection latency with FPFM applied. The results have shown CADD to successfully report an anomaly right after the observation window of FPFM in 50–80% (>40%) test cases even when α = 1–40 (α = 50), indicating CADD can still capture anomalies immediately most of the time (α = 1–40) right after their occurrences (but just chooses to report them at a later time).

If CADD is implemented as an alarm system, detection delay will not be a major concern, and the application developers should provide the flexibility to users to adjust how aggressive FPFM should be according to their preference. On the other hand, if CADD is to be used for on-line defense that directly integrates into the vehicle's control system, the value of α should also depend on the system requirement/characteristics, including the maximum tolerable detection delay and the maximum tolerable attack duration, which are known to the component suppliers or car makers. In other words, even if the attacker tries to evade CADD's detection when FPFM is applied by shortening its attack duration, the system should be able to handle the absence of correct data during that period of time.



(a) GPS traces of two test cases.

(b) GPS elevation measurements of the two test cases in (a).

Figure 16: Examples of test-case traces.

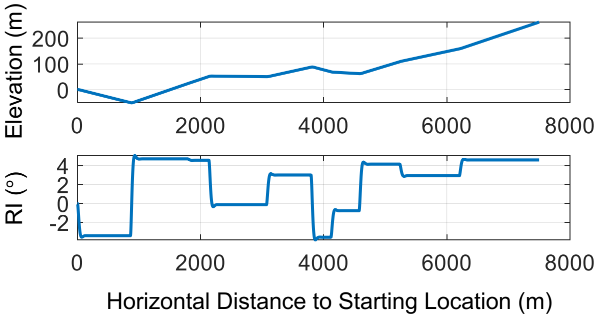


Figure 17: Example of elevation template utilized in E_2.

Appendix C SUPPLEMENTARY EVALUATIONS

C.1 Test Case Examples

Figs. 16(a) and 16(b) show examples of driving traces used in our evaluation. These figures also show that GPS only has coarse-grained values in both temporal and magnitude perspectives. Note that we *do* include the scenarios of varying unobservable contexts in our evaluation, such as the vehicle travels on a curvy road, performs lane changing and turns with steering applied. While Fig. 17 shows an example elevation template created for E_2, Figs. 19 and 20 show the example CDFs of vehicle acceleration in E_1 and E_2, respectively.

C.2 Additional Evaluation on Detection Latency

CADD is shown to require only one anomalous data sample to detect an anomaly in >70% test cases as shown in Fig. 21.

C.3 Performance Analysis of Prior Work

C.3.1 Supplementary Evaluation Analysis. Table 10 presents the performance values of EVAD and PID-Piper shown in Figs. 11 and 12 (Section 7.3), including the performance when PID-Piper is adjusted

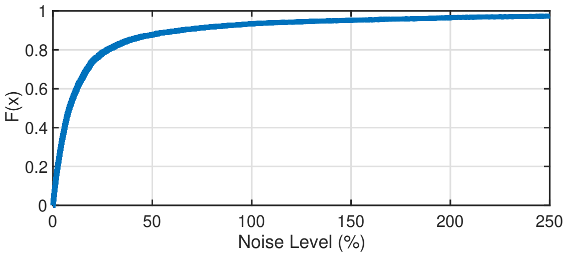


Figure 18: CDF of noise injected in E_2.

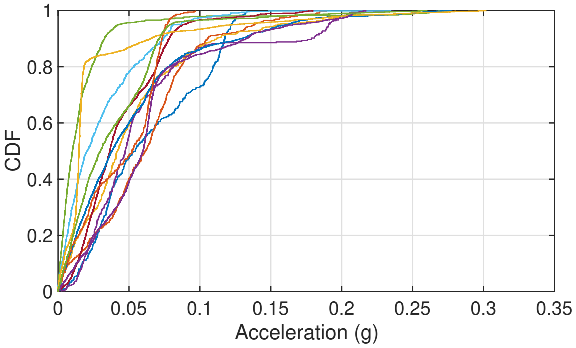


Figure 19: CDFs of acceleration measurements in each trace of E_1.

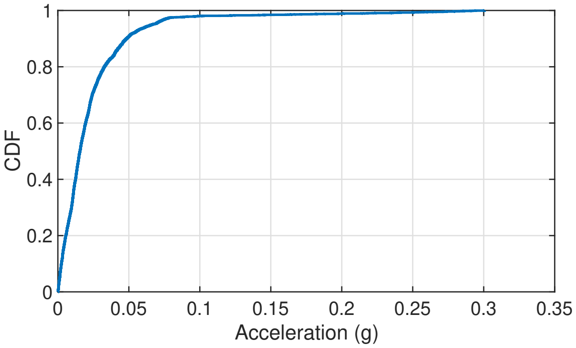


Figure 20: An example CDF of the acceleration profile used in E_2.

to match the TPR of CADD and EVAD. As one can observe, PID-Piper can only achieve a moderate (<30.7%) TPR while maintaining a reasonably low FPR. However, if we decrease its detection threshold to raise the TPR in order to match CADD's TPR performance, there will be a significant tradeoff between TPR and FPR to the point where it can no longer produce meaningful detection results (*i.e.*, TPR ≈ FPR). This is due to PID-Piper's design that relies purely on machine learning without taking the data/physical correlation of vehicle component into account. Due to the difficulty of including every possible scenario in training data (C3 of Section 1) and the lack of detailed data/measurements (C1 and C2 of Section 1), PID-Piper will be forced to use a very loose threshold for its detection to maintain a reasonable FPR, becoming unable to efficiently detect anomalies.

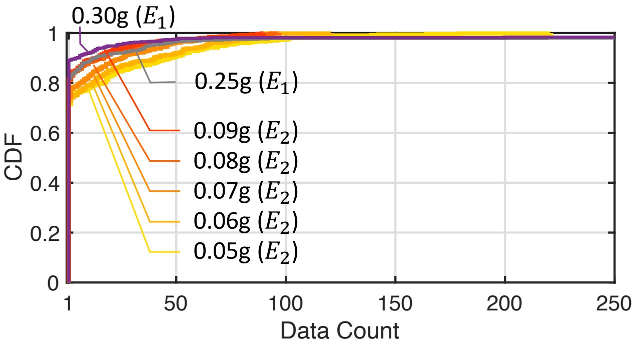
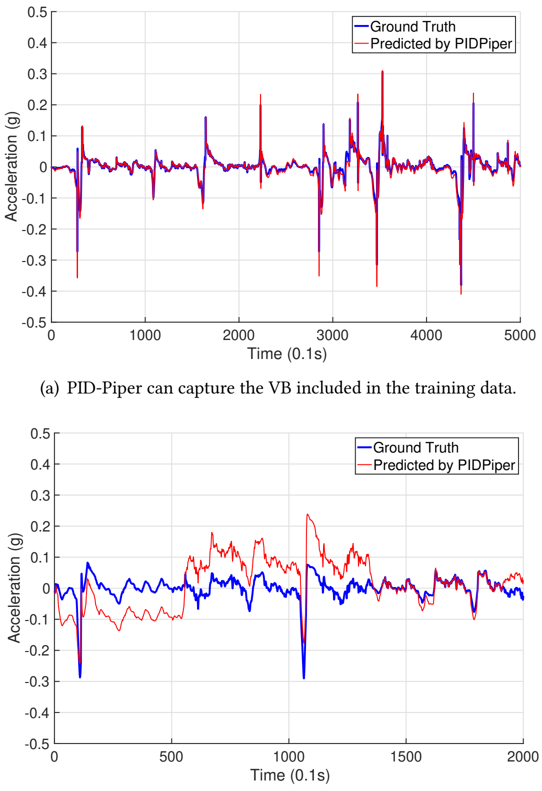


Figure 21: CDF of the required data count for CADD to detect an anomaly (as an alternative way to present detection latency) when Δa_X is anomalous.

Metric	Δa_X: Amount of Manipulation to a_X				
	0.05g	0.06g	0.07g	0.08g	0.09g
EVAD: $\mu = 0.9$ (No Tire Slippage)					
TPR (%)	73.90	90.16	91.98	93.50	94.06
FPR (%)	11.45				
DL (ms)	3739	2717	2529	2075	2116
EVAD: $\mu = 0.2$ (5 ∼ 34% Slippage Time)					
TPR (%)	76.02	90.13	92.84	94.48	95.39
FPR (%)	15.40				
DL (ms)	3182	2926	2345	2022	1745
PIDPiper: $\mu = 0.9$ (No Tire Slippage)					
TPR (%)	11.58	14.52	23.54	26.25	30.70
FPR (%)	3.63				
DL (ms)	7333	8937	8511	7632	8250
PIDPiper: $\mu = 0.2$ (5 ∼ 34% Slippage Time)					
TPR (%)	14.60	17.94	22.74	25.59	28.26
FPR (%)	10.12				
DL (ms)	13900	10872	10653	12630	13764
PIDPiper (Match TPR): $\mu = 0.9$ (No Tire Slippage)					
TPR (%)	91.71	93.15	93.45	91.03	91.64
FPR (%)	91.63				
DL (ms)	39	8	15	19	11
PIDPiper (Match TPR): $\mu = 0.2$ (5 ∼ 34% Slippage Time)					
TPR (%)	93.87	94.22	95.19	91.69	92.14
FPR (%)	91.70				
DL (ms)	29	15	30	8	42

Table 10: Performance of EVAD and PID-Piper in E_2 (a_X anomalous).

The above statement can be confirmed by comparing the training prediction results with the testing prediction results of PID-Piper's models (Fig. 22). Specifically, Fig. 22(a) shows the fact that PID-Piper's models can indeed capture the (normal) vehicle behavior/acceleration included in the training data. However, Fig. 22(b)



(a) PID-Piper can capture the VB included in the training data.

(b) PID-Piper cannot accurately predict the sensor readings under scenarios it has not encountered before.

Figure 22: Training and testing of PID-Piper.

shows that there will be prediction error/offset caused by different driving contexts (*e.g.*, RI and RF) that cannot be fully modeled by the standard training process of PID-Piper. Note that CADD, EVAD, and PID-Piper all use the same set of training data during the evaluation.

We further tested the condition where PID-Piper has access to a special set of training data that includes 10 driving segments, where the vehicle is travelling on a 10 different slope, and see if this special treatment enables PID-Piper to overcome the training difficulty. Table 11 shows PID-Piper's performance after this training process. While PID-Piper's performance can indeed be improved by having more training data, it still incurs a 66.95% FPR when its thresholds are adjusted to match the TPR performance of CADD and EVAD. In summary, the training difficulty is still a major obstacle (to a pure ML-based approach) that cannot be easily overcome.

C.4 Resilience against Different Attacks

Table 12 shows CADD's detection and identification performance when Δa_X is smaller than the detection threshold (η_{RI} = 2 or $\Delta a_{X,min}$ = 0.035g). As expected, CADD can only achieve moderate TPR since the data manipulation is below the detection threshold. However, CADD is shown to achieve ≥71.8% Acc_{id} even if ADR is only 12.9%. These results indicate that even with data manipulation equal to only 28.6% of detection threshold, CADD can still capture/pinpoint the data in question with good accuracy. However, any manipulation under the detection threshold should not be considered as a valid attack since, in a practical deployment scenario, the detection threshold should also account for the error tolerance of the vehicle system.

Metric	Δa_X: Amount of Manipulation to a_X				
	0.05g	0.06g	0.07g	0.08g	0.09g
PIDPiper: μ = 0.9 (No Tire Slippage)					
TPR (%)	5.26	4.98	5.39	6.88	9
FPR (%)	4.24				
DL (ms)	7431	7858	5874	3703	4141
PIDPiper: μ = 0.2 (5 ∼ 34% Slippage Time)					
TPR (%)	12.70	12.56	13.18	14.43	16.76
FPR (%)	13.31				
DL (ms)	4876	3639	1561	1730	1484
PIDPiper (Match TPR): μ = 0.9 (No Tire Slippage)					
TPR (%)	75.58	81.76	88.30	93.75	96.16
FPR (%)	66.95				
DL (ms)	67	51	65	57	90
PIDPiper (Match TPR): μ = 0.2 (5 ∼ 34% Slippage Time)					
TPR (%)	76.36	81.99	88.51	93.77	96.43
FPR (%)	66.87				
DL (ms)	69	40	86	85	91

Table 11: Detection performance of PID-Piper in E_2 (a_X anomalous), given special training data.

ADR	Metric	μ = 0.9, FPR = 4.41%			μ = 0.2, FPR = 6.51%		
Manipulation		0.01g	0.02g	0.03g	0.01g	0.02g	0.03g
64.5	TPR	4.27	8.62	20.57	7.58	10.05	20.65
	Acc_{id}	71	92	100	90	96	100
32.3	TPR	4.27	8.43	19.93	7.14	10	20.29
	Acc_{id}	70.6	81.6	99.4	90	93.8	99.8
12.9	TPR	4.29	7.77	18.93	7.05	9.83	19.87
	Acc_{id}	71.8	76.6	95	90	91.2	96

Table 12: This table shows the detection and identification performance (%) of CADD when Δa_X is smaller than the detection threshold ($\Delta a_{X,min}$ = 0.035g).

Scenario (μ)	0.05g	0.06g	0.07g	0.08g	0.09g
CADD (0.9)	870.0	238.0	8.8	7.4	8.4
CADD (0.2)	849.8	212.4	16.6	25.4	70.0
EVAD (0.9)	58707.8	56852.2	57956.2	56511.0	57471.6
EVAD (0.2)	44644.6	46727.8	46164.4	41862.8	46385.4
PID-Piper (0.9)	13547.8	16276.6	16913.2	14846.4	16472.0
PID-Piper (0.2)	13771.8	17128.4	16763.8	15688.4	16159.4

Table 13: This table shows the detection latency (ms) of CADD and prior work under (coordinated) multi-data attacks.

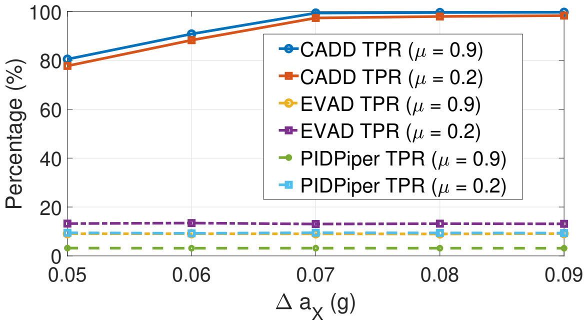


Figure 23: Comparison of TPRs between CADD and prior work under (coordinated) multi-data manipulation attacks. Since the FPRs will not be influenced by the attacks, the FPR comparison is the same as Fig. 11.

We also tested the condition where all dynamics measurements and the assistance data, except for θ_d, are manipulated simultaneously. Note that the evaluation settings here are basically the same as in E_2, except that all aforementioned data are manipulated simultaneously to match the normal (physical) data correlation (*e.g.*, vehicle acceleration now perfectly matches DOIs other than θ_d while ignoring measurement noise) and the vehicle behavior tested in the test cases are indeed captured from the normal vehicle behavior but in a different driving context (*i.e.*, RI). As shown in Fig. 23, CADD's TPRs are almost identical to those shown in Fig. 11. EVAD and PID-Piper, however, can only achieve <20% TPR because the former does not have any special design to consider driving context and the latter cannot efficiently account for θ_d to detect the anomaly while other data match their normal correlation. As a result, they also have significantly higher detection delays than CADD (Table 13).

We now consider the condition in which the adversary tries to evade CADD's detection by disguising the attack as a mass change to the vehicle. Note any vehicle behavior change caused by total mass (including passengers and cargoes) change must be consistent throughout the entire trip as described in Section 6.1. Therefore, in order for an attack to disguise as a mass change, the manipulated acceleration must be maintained at a constant ratio, compared to the ground-truth acceleration, throughout the entire trip. This will further lead to the requirement of simultaneously manipulating vehicle speed and GPS to evade a simple consistency check between those sensor readings. However, the road grade information from a map (looked up by using GPS readings) cannot be controlled by the attacker and, therefore, the inconsistency between road grade and vehicle acceleration can be detected by CADD

Also, for an attack to disguise as a tire slippage, the adversary needs to further manipulate both road grade estimation (*e.g.*, GPS or GPS and map) and vertical acceleration readings simultaneously to match proper tire slippage features w.r.t. to the road grade estimation obtained from gas/brake and engine torque, *i.e.* leading to requiring manipulating all the status measurements considered in CADD.