Resilient Cooperative Adaptive Cruise Control for Autonomous Vehicles Using Machine Learning

Srivalli Boddupalli, *Graduate Student Member, IEEE*, Akash Someshwar Rao, and Sandip Ray, *Senior Member, IEEE*

***Abstract*— Cooperative Adaptive Cruise Control (CACC) is a fundamental connected vehicle application that extends Adaptive Cruise Control by exploiting vehicle-to-vehicle (V2V) communication. CACC is a crucial ingredient for numerous autonomous vehicle functionalities including platooning, distributed route management, etc. Unfortunately, malicious V2V communications can subvert CACC, leading to string instability and road accidents. In this paper, we develop a novel resiliency infrastructure, RACCON, for detecting and mitigating V2V attacks on CACC. RACCON uses machine learning to develop an on-board prediction model that captures anomalous vehicular responses and performs mitigation in real time. RACCON-enabled vehicles can exploit the high efficiency of CACC without compromising safety, even under potentially adversarial scenarios. We present extensive experimental evaluation to demonstrate the efficacy of RACCON.**

***Index Terms*— Connected and autonomous vehicles, V2X communication, anomaly detection, security.**

I. Introduction

RECENT years have seen proliferation of electronics and software in automotive systems targeted towards increasing autonomy. Autonomous features hold the promise of dramatically increasing transportation efficiency and road safety by reducing and eventually eliminating human errors [29]. However, an undesired side-effect is the increased vulnerability of Connected and Autonomous Vehicles (CAVs) to cybersecurity threats. Recent research has shown that it is possible, even straightforward, to mount cyber-attacks that compromise a vehicle and control its driving functionality [11], [22], [26], [27]. Increasing dependence of critical vehicular operations on communication with the external world will exacerbate this situation by creating larger attack surfaces. This increases the attacker's ability to compromise the vehicle causing catastrophic impact. Consequently, the proliferation and even adoption of CAVs depends critically on our ability to mitigate such attacks.

Manuscript received December 18, 2020; revised July 1, 2021 and November 21, 2021; accepted December 31, 2021. This work was supported in part by the National Science Foundation under Grant CNS-1908549. The Associate Editor for this article was N. Bekiaris-Liberis. *(Corresponding author: Srivalli Boddupalli.)*

Srivalli Boddupalli and Sandip Ray are with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA (e-mail: bodsrivalli12@ufl.edu; sandip@ece.ufl.edu).

Akash Someshwar Rao was with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA. He is now with Qualcomm Inc., San Diego, CA 92121 USA (e-mail: akash.someshwarr@ufl.edu).

Digital Object Identifier 10.1109/TITS.2022.3144599

An important feature of autonomous vehicles is the ability to interact with other vehicles (V2V), the transportation infrastructure (V2I), and devices connected to the Internet (V2IoT). Vehicular communications, collectively referred to as V2X, form a key constituent of several emergent applications including platooning, cooperative route management, intersection management, cooperative collision detection, etc. Unfortunately, V2X also enables a large class of adversarial opportunities: an adversary can easily create disruption by manipulating communicated messages through mutation, misdirection, or jamming. For example, in platooning, the adversary may cause an accident by simply sending misleading acceleration directive while braking [12].

In this paper, we develop an infrastructure for systematically integrating resiliency against communication attacks on V2V applications. Our focus is a fundamental application of vehicular communications: Cooperative Adaptive Cruise Control (CACC). CACC is an extension of Adaptive Cruise Control (ACC); Adaptive Cruise Control (ACC) uses RADAR/LIDAR measurements to derive relative velocity and headway from the vehicle in front. Additionally, CACC also accounts for the preceding vehicle's (intended) acceleration. The acceleration is communicated through V2V messages, typically as Dedicated Short Range Communication (DSRC) [39] or Cellular Vehicle-to-Everything standard (C-V2X) [41]. CACC is a key component of several connected car applications such as vehicle platooning, cooperative on-ramp merging, etc. Attacks on CACC can disrupt traffic movement, cause catastrophic accidents, and bring down the transportation infrastructure.

Our framework, RACCON (for "Resilient Cooperative Adaptive Cruise Control"), is a real-time anomaly detection and mitigation system for communication attacks on CACC. The key idea is to use machine learning (ML) to develop an on-board prediction model for estimating the response of the following vehicle given normal (benign) patterns of V2V input messages. This enables the detection of anomalies in the vehicle's responses resulting from potentially malicious communications. RACCON involves two cooperative components: (1) an on-board architecture installed in vehicles participating in CACC that enables the follower vehicle (also called *ego vehicle*) to perform real-time anomaly detection and mitigation; and (2) an offline cloud-based infrastructure for construction of prediction models.

The paper makes several important contributions. First, unlike related approaches that focus on *detection* of CACC

TABLE I
GLOSSARY OF NOTATIONS

Term	Definition
$\mathcal{E}$	Ego vehicle
$\mathcal{P}$	Preceding vehicle
K_a	Acceleration constant (0.66)
K_v	Velocity constant ($0.99s^{-1}$)
K_g	Position constant ($4.08s^{-2}$)
D^{max}	Maximum deceleration constant ($8ms^{-2}$)
G_{min}	Lower bound on space gap ($1.0m$)
T^A_{gap}	Constant time headway for ACC ($1.2s$ for [5])
T^C_{gap}	Constant time headway for CACC ($0.55s$ for [5])
g_{safe}	Target safe gap
$a^A_{\mathcal{E}}$	Desired acceleration for ACC
$a^C_{\mathcal{E}}$	Desired acceleration for CACC
$a_{\mathcal{P}}$	Instantaneous Acceleration of $\mathcal{P}$
t_{gap} or THW	Instantaneous time headway
$a^{pred}_{\mathcal{E}}$	Predictor output (anomaly detection)
$a^{est}_{\mathcal{E}}$	Response estimatior output (mitigation)

attacks (see Section II-C), RACCON represents the first framework that also enables *real-time resiliency*. Second, our framework provides high flexibility through attack-agnostic defense against an elaborate set of adversaries in the connected car ecosystem, including man-in-the-middle (MITM) attack, wormhole attack, Sybil, Denial-of-Service (DoS), etc. RACCON is oblivious to the underlying V2X communication technology, *i.e.*, DSRC vs C-V2X.

It also accounts for the natural differences in communication patterns among a variety of driving scenarios, road conditions, etc. Finally, our work represents the most comprehensive experimental evaluation to date on vulnerabilities in CACC, impact of attacks on target vehicles, and the quality of resiliency provided by the security architecture. In addition to showcasing confidence in our approach, we believe the experimental framework will serve as a roadmap for evaluation of resiliency in other CAV applications.

The remainder of the paper is organized as follows. Section II provides relevant background on CACC. We introduce RACCON in Section III and explain its design constraints. Section IV presents details of the RACCON architecture and implementation. A unique contribution of the paper is the extensive evaluation performed to demonstrate the efficacy of RACCON. Sections V through IX explain our experimental results and conclude in Section X.

II. BACKGROUND AND RELATED WORK

We begin with some preliminaries on ACC and CACC to provide the relevant background. The description here (and in the rest of the paper) makes use of several notations for specific parameters. We list the notations in Table I for convenience.

A. ACC and CACC Overview

Adaptive Cruise Control (ACC) enables a vehicle $\mathcal{E}$ to automatically adjust acceleration and closely follow its preceding vehicle $\mathcal{P}$, while maintaining a safe space gap g_{safe}. Most ACC implementations target a *constant time headway*; the goal is to compute $a_{\mathcal{E}}$ such that $\mathcal{E}$ takes at least time T_{gap} to reach the same position as $\mathcal{P}$, where T_{gap} is a design constant. The safe space gap g_{safe} is a function of T_{gap}, the maximum

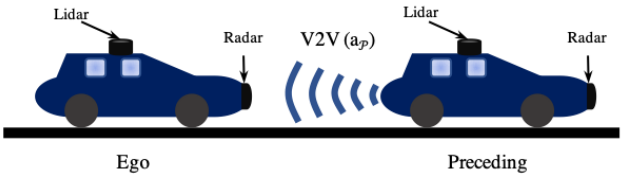


Fig. 1. CACC Overview. Acceleration is provided by V2V. Instantaneous g and $v_{\mathcal{P}}$ are provided by LIDAR or RADAR.

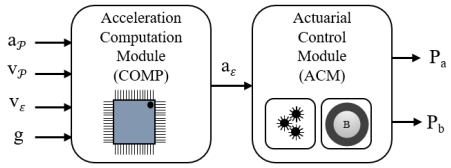


Fig. 2. CACC On-board Architecture. Acceleration Computation Module (COMP) computes desired acceleration. Actuation control module (ACM) computes braking pressure and motor torque.

deceleration capability $D^{\max}_{\mathcal{E}}$ of $\mathcal{E}$, and the velocities $v_{\mathcal{E}}$ and $v_{\mathcal{P}}$. Vehicle $\mathcal{E}$ computes its desired acceleration $a_{\mathcal{E}}$ using (1) the inter-vehicle distance g and velocity $v_{\mathcal{P}}$ of the preceding vehicle $\mathcal{P}$ measured by RADAR/LIDAR; and (2) the velocity $v_{\mathcal{E}}$ and acceleration $a_{\mathcal{E}}$ of $\mathcal{E}$ measured by on-board sensors. *Cooperative Adaptive Cruise Control* (CACC) extends ACC by using the intended acceleration $a_{\mathcal{P}}$ of $\mathcal{P}$ in the computation of $a_{\mathcal{E}}$. Vehicle $\mathcal{P}$ communicates $a_{\mathcal{P}}$ through V2V messages (Fig. 1). Both ACC and CACC operate in two modes. If $g > g_{\text{safe}}$, they operate in *gap control mode*, where $\mathcal{E}$ follows $\mathcal{P}$ as closely as possible; if $g \leq g_{\text{safe}}$, they switch to *collision avoidance mode* and uses maximum deceleration $D^{\max}_{\mathcal{E}}$. The use of the preceding vehicle's acceleration enables CACC to maintain a shorter time headway (THW) than ACC, resulting in a more efficient roadway utilization.

B. CACC Architecture and a Representative Implementation

Fig. 2 shows an on-board CACC architecture.[1] While low-level details vary, all implementations constitute two components: *Acceleration Computation Module (COMP)* computes the desired acceleration $a_{\mathcal{E}}$ of the host vehicle $\mathcal{E}$; *Actuation Control Module (ACM)* manipulates motor output torque or braking pressure to enforce the desired acceleration.

RACCON is oblivious to the underlying CACC implementation. However, for our evaluation we use representative CACC (and ACC) implementations by Amoozadeh *et al.* [5] shown below. The safe space-gap g_{safe} is computed through Equation 1 while Equations 2 and 3 represent the controller operation for computing the desired acceleration under ACC and CACC systems, respectively. K_a, K_v, and K_g are acceleration, velocity,

[1]CACC is often conflated with decentralized connected vehicle platooning. While CACC is a two-vehicle application, platooning, in general, is a multi-vehicle car-following application (and can use various communication paradigms besides CACC) [6], [7], [15], [20], [32]. The scope of this research is confined to V2X security of CACC. Security of general platoon systems is out of scope of this paper. However, recent work [8] shows how to extend some of the ideas discussed here to certain platooning implementations.

and position constants.

$$g_{\text{safe}} = 0.1v_{\mathcal{E}} + \frac{v_{\mathcal{E}}^2}{2D_{\mathcal{E}}^{\max}} - \frac{v_{\mathcal{P}}^2}{2D_{\mathcal{P}}^{\max}} + G_{\min} \quad (1)$$

$$a_{\mathcal{E}}^{A} = -K_a D_{\mathcal{P}}^{\max} + K_v(v_{\mathcal{P}} - v_{\mathcal{E}}) + K_g(g - v_{\mathcal{E}} T_{\text{gap}}^{A} - G_{\min}) \quad (2)$$

$$a_{\mathcal{E}}^{C} = K_a a_{\mathcal{P}} + K_v(v_{\mathcal{P}} - v_{\mathcal{E}}) + K_g(g - v_{\mathcal{E}} T_{\text{gap}}^{C} - G_{\min}) \quad (3)$$

Amoozadeh *et al.* specify $K_a = 0.66$, $K_v = 0.99s^{-1}$, $K_g = 4.08s^{-2}$, $G_{\min} = 1m$, $T_{\text{gap}}^{A} = 1.2s$, and $T_{\text{gap}}^{C} = 0.55s$, as listed in Table I. We use these numbers in the evaluation of RACCON.

C. Related Work

Over the last decade there have been several high-profile papers showing sophisticated security compromises of automotive systems [11], [22], [27]. With the emergence of CAVs, there has been research on secure cooperative applications such as platooning, intersection management, collision avoidance, lane merge and turn conflict warning, etc. [10], [12], [21]

Several techniques have been proposed to capture malicious vehicular nodes disseminating misinformation in a vehicular ad-hoc network (VANET). Ganesan *et al.* [14] proposed an anomaly detection technique that exploits the natural redundancy and the correlation among the measurements from heterogeneous on-board sensors and the vehicular communication messages received. However, such a redundancy may not be available in emerging CAVs due to the inherent limitations in sensor technologies and the associated cost considerations. Gyawali *et al.* [17] developed a decentralized cooperative misbehavior detection system (MDS) using machine learning to capture falsified position attacks. Each vehicle is equipped with this MDS and broadcasts its detection result to all the vehicles in its vicinity. Based on the aggregate results, a misbehaving vehicle is evicted. Golle *et al.* [16] proposed an attack detection and correction technique using a combination of parsimony assumptions (an attack involving a few malicious nodes is more likely than an attack that requires collusion between a large number of nodes) and on-board sensors to thwart Sybil attacks and the like. Raya *et al.* [34] also propose a misbehavior detection system and an eviction methodology using certificate-based authentication in addition to consolidating the detection results from all the vehicles. Misbehaving vehicles are evicted based on majority voting. However, consensus-based anomaly detection systems relying on cross-validating data from various vehicles are not suitable for CACC where with two participating vehicles. Furthermore, these techniques propose eviction of the misbehaving vehicle as a mitigation measure which is ineffective for real-time autonomous driving application where the victim vehicle must continue making driving decisions even in presence of anomalies.

ML-based anomaly detection has been applied to a number of automotive security problems. Taylor *et al.* [36] present anomaly detection of attacks on CAN bus by monitoring the historical packet timing. Müter *et al.* [28] present an entropy-based anomaly detection for securing in-vehicle networks. Weber *et al.* [42] present an on-line detector based on machine learning for capturing anomalies in automotive CAN communication. Additionally, Vatanparvar *et al.* [40] present a GAN-based detection and recovery approach for automotive control systems against cyber-physical adversaries and demonstrate their solution on battery monitoring system. ML-based techniques have also been applied for detecting certain CACC anomalies as described below.

Since CACC serves as a foundation of several CAV applications, significant attention has been given towards security of CACC. Abdo *et al.* [2] present a survey on application level communication attacks on CACC and their adverse impacts on the target vehicles. Liu *et al.* [25], Parkinson *et al.* [30] and AbdAllah *et al.* [1] discuss the challenges in CACC security and provide research directions. Biron *et al.* [3] and Dutta *et al.* [13] use approaches based on control theory to detect and correct adversarial sensor-based attacks on CACC. Heijden *et al.* [38] propose a misbehavior detection mechanism based on subjective logic, to validate the position information exchanged between vehicles. Nunen *et al.* [39] propose a control-theoretic model-predictive approach to deal with short communication failures and packet dropouts in CACC. Among machine learning approaches, Alotibi *et al.* [4] propose a real-time detection mechanism for platoons, in the context of a compromised leader reporting falsified acceleration values to the following vehicles. Iorio *et al.* [18] propose a misbehavior detection approach for injection attacks on CACC, based on correlation between various vehicular motion parameters. Jagielski *et al.* [19] discuss detection of attacks that compromise communication or manipulate the on-board sensor readings, through physics-based constraints and machine learning. Levi *et al.* [24] present an event-based anomaly detection technique for connected vehicles using Hidden Markov Models. Tiwari *et al.* [37] describe attack features that are undetectable at individual time instances but can be detected from sequential data.

In spite of extensive research, we are not aware of any previous solution addressing detection of the spectrum of attacks explored for RACCON. Control-theoretic approaches require a detailed functional model of the adversarial action. Each attack type (*e.g.*, flooding, jamming, etc.) requires a different detailed adversary model. In contrast, training of ML models in RACCON depends only upon benign V2V communication data. RACCON's attack-agnostic defense is effective against the entire spectrum of V2V adversaries. On the other hand, related ML-based approaches have only been evaluated under a specific subset of attacks, *e.g.*, linear or sinusoidal mutation attacks on acceleration values [4], [19].

III. Introduction to RACCON

A. RACCON Usage Model

The usage model of RACCON envisions it as a vehicular service for connected vehicles. A vehicle can subscribe to the service only if it includes RACCON on-board architecture (see Section IV). We refer to the subscribing vehicle as the *ego*

vehicle,"$\mathcal{E}$"; all our evaluations are done from the perspective of an ego vehicle. When enabled, RACCON collects normal behavior data during $\mathcal{E}$'s operation. Data from all vehicles with RACCON installed is periodically uploaded to a trusted cloud server for progressively refining ML models used by the on-board hardware; $\mathcal{E}$ periodically updates the on-board system by downloading the latest ML models. The communication with cloud is performed when $\mathcal{E}$ is connected to Internet through a trusted network, *e.g.*, when stationary at the owner's residence; on-road connectivity with cloud is not necessary. During driving operation, if CACC is engaged in $\mathcal{E}$, the on-board hardware automatically detects anomalies in V2V communication from the preceding vehicle, and performs mitigation.

B. Design Considerations

A unique feature that distinguishes RACCON from related ML approaches for anomaly detection in CACC is *real-time resiliency*. For our solution to be viable, a number of design constraints must be satisfied.

- *Basic safety:* ML-based solutions can only provide a "high probability" guarantee on prediction accuracy. Consequently, it is critical that the RACCON mitigation generates decisions that are *safe* (under the assumed threat model), *i.e.*, do not increase the risk of accident in response to a detected anomaly.
- *Flexibility:* The solution should work without modification, for the entire adversarial spectrum. Hence, control-theoretic solutions that require detailed customized models of adversarial functionalities are infeasible.
- *Limited Computation and Real-time Requirements:* The security system should operate within the computational constraints of an automotive platform and meet real time requirements of CACC application.
- *Small Data Problem and Machine Learning Attacks:* Any ML-based prediction system requires a significant amount of training data. Significant attack data does not exist in real life, a phenomenon we refer to as the *small data problem*. It is critical for the prediction system to be accurate in the presence of limited anomaly data. Furthermore, the system must be robust against *detector subversion*, *i.e.*, attacks targeted specifically to fool the prediction system (see Section VIII).

RACCON addresses the resource constraints and real-time requirements by separating the training of ML models from on-road prediction. A key observation is that the computation-intensive component of machine learning is training predictor models; once a model is created, detection can be performed within the limited resources of automotive ECUs. Our system includes a cloud-based methodology for training prediction models, while the on-board architecture is responsible for collecting data and performing real-time prediction. We ensure basic safety by introducing a *plausibility checker* which guarantees that RACCON's mitigation cannot compromise safety due to V2V anomalies. To address the small data problem, we observe that while labelled anomalous/malicious data is limited, data on normal behavior is typically plentiful. Consequently, we train prediction algorithms to learn *normal behavior model* (NBM), *i.e.*, the response of $\mathcal{E}$ to normal (benign) pattern of V2V communications rather than anomalous behavior. The on-board anomaly detector then calculates the degree of deviation from NBM as a measure of the anomaly. Finally, for ensuring resiliency under detection subversion attacks, we systematically fine-tune the detection threshold to capture minute anomalies that have a perceptible effect on the safety or efficiency of the target vehicle. As a result, stealthy attacks that indeed subvert the detection system fail to cause any adverse impact on the vehicle.

C. Threat Model

Given our focus on V2V, our threat model assumes that the attacker can tamper with arbitrary V2V messages. This includes mutation, denial of delivery, masquerading as a different vehicular or infrastructure entity, message fabrication, etc. Our framework is oblivious to the source of the attack: it can be a rogue preceding vehicle, a compromised ego vehicle infrastructure component, or an intermediate networking component, *e.g.*, denial of message delivery is possible by compromising the software/hardware component of the ego vehicle or interfering with the communication protocol. We assume that the RACCON on-board system in the ego vehicle, as well as the Actuation components it controls, are not compromised. We also assume that the sensory inputs to the ego vehicle are not corrupted.[2] Note that in addition to malformed/dropped V2V communication messages, the threat model includes well-formed V2V message (*i.e.*, obeying the underlying DSRC/C-V2X protocol) with a payload different from the ground truth. For instance, a rogue vehicle accelerating at $0.5ms^{-2}$ can cause a collision by continuously sending legal V2V messages reporting false acceleration values larger than $0.5ms^{-2}$ for a sustained period of time. Messages with such corrupted payloads cannot be discarded by simple structural/linting checks of compliance with the underlying protocol.

IV. RACCON Implementation

Fig. 3 shows the high-level architecture of RACCON. It includes a cloud component for off-line ML-training and an on-board infrastructure for real-time resiliency. A key insight is that since on-board architecture of most CACC implementations follows the "template" from Fig. 2, it is possible to develop a streamlined resiliency architecture by systematically augmenting the template with resiliency components. RACCON adds three such components: (1) Anomaly Detector; (2) Mitigator; and (3) Data Collector.

A. Anomaly Detector

Anomaly detector checks at each instant t whether the response $a_{\mathcal{E}}(t)$ of the COMP module of CACC deviates from the expected normal behavior; any such deviation is captured

[2]There has been significant research showing how vehicular sensors can be compromised [3], [4], [13], [31]. Nevertheless, since the modalities of compromising sensors and V2V are different, it is reasonable in the context of detecting V2V anomalies to assume that the sensory inputs are trusted.

Algorithm 1 RACCON

1: **procedure** RACCON($a_{\mathcal{P}}^{V2V}, v_{\mathcal{P}}, v_{\mathcal{E}}, gap$)
2: $a_{\mathcal{P}} \leftarrow a_{\mathcal{P}}^{v2v}$
3: **if** V2V communication is lost **then**
4: $no\_comm \leftarrow TRUE$
5: $a_{\mathcal{E}}^{pred} \leftarrow Predictor()$ ***predictor invoked***
6: $a_{\mathcal{E}}^{c} \leftarrow AccelComp(a_{\mathcal{P}}, v_{\mathcal{P}}, v_{\mathcal{E}}, gap)$
7: $anmly\_flag \leftarrow Comparator(a_{\mathcal{E}}^{c}, a_{\mathcal{E}}^{pred})$
8: $a_{\mathcal{E}} \leftarrow Mitigator(anmly\_flag, no\_comm)$
9: $throttle, braking \leftarrow ActuationControl(a_{\mathcal{E}})$
10: $DataCollector()$
11: **return** $throttle, braking$

as an anomaly to be passed on to Mitigator. The detection comprises the following two modules.

1) **Predictor** is a machine learning model that is trained offline. It estimates *predicted acceleration value* $a_{\mathcal{E}}^{pred}(t)$ in real time, taking the same input parameters as COMP. Predictor can capture contextual/conditional anomalies, in addition to point anomalies.
2) **Comparator** computes the deviation between the predicted value $a_{\mathcal{E}}^{pred}(t)$ and $a_{\mathcal{E}}(t)$; if the deviation is beyond a pre-defined threshold, it is detected as an anomaly. The detection threshold is a function of driving conditions and typical velocities of vehicles in a driving environment (See Section VIII).

Remark 1: One can ask why we need an ML model for the Predictor. After all, since the sequence of velocity values of the preceding vehicle is accessible to RACCON and these values are obtained from sensory data which are trusted in our threat model, one can imagine that it is possible to simply use this sequence to compute a projected acceleration to replace the corrupted decision. However, our analysis shows that is not the case. Acceleration values are in fact highly contextual, i.e.*, the same preceding sequence can result in a very different next value depending on a number of environmental factors. For instance, a preceding vehicle acceleration value of* $2ms^{-2}$ *may be within the normal range in a highway setting on a clear day, given its velocity is* $50mph$*. The same acceleration value given this velocity may very well be anomalous in a city environment. Therefore, it is critical to capture the unique context accurately to detect anomalous acceleration values in different driving environments. Furthermore, anomalies caused by minute deviations from dynamic normal reference can have significant impact on safety and efficiency. Purely deterministic predictor based on kinematics rules is not sufficient to capture such minute and contextual anomalies. An ML-based predictor addresses these issues by learning the multi-variate non-linear distribution of the ego vehicle's acceleration as a function of carefully crafted feature set that can accurately capture the context.*

Algorithm 2 Mitigation

1: **procedure** MITIGATOR($anmly\_flag$, no_comm)
2: **if** ($anmly\_flag$ and no_comm are FALSE) **then**
3: ***operate in normal mode***
4: $a_{\mathcal{E}} \leftarrow a_{\mathcal{E}}^{c}$
5: **else**
6: ***mitigation mode***
7: $sensor\_sampling\_frequency \leftarrow F_{max}$
8: $v_{\mathcal{P}}, gap \leftarrow v_{\mathcal{P}}^{F_{max}}, gap^{F_{max}}$
9: $a_{\mathcal{P}} \leftarrow (v_{\mathcal{P}}(t) - v_{\mathcal{P}}(t-1))/\delta T$
10: $a_{\mathcal{E}}^{c} \leftarrow AccelComp(a_{\mathcal{P}}, v_{\mathcal{P}}, v_{\mathcal{E}}, gap)$
11: $a_{\mathcal{E}}^{est} \leftarrow RespEst(v_{\mathcal{P}}, v_{\mathcal{E}}, gap)$
12: $a_{\mathcal{E}} \leftarrow Plausibility(a_{\mathcal{E}}^{est}, a_{\mathcal{E}}^{c}, v_{\mathcal{P}}, gap, D_{\mathcal{P}}^{max})$
13: **return** $a_{\mathcal{E}}$
14: **procedure** PLAUSIBILITY($a_{\mathcal{E}}^{est}, a_{\mathcal{E}}^{c}, v_{\mathcal{P}}, gap, D_{\mathcal{P}}^{max}$)
15: $t_{gap}^{est}, t_{gap}^{c} \leftarrow GetTGap(a_{\mathcal{E}}^{est}, a_{\mathcal{E}}^{c}, v_{\mathcal{P}}, gap, D_{\mathcal{P}}^{max})$
16: **if** $t_{gap}^{c} > T_{gap}^{c}$ & $t_{gap}^{c} < t_{gap}^{est}$ & $t_{gap}^{c} < T_{gap}^{A}$ **then**
17: $a_{\mathcal{E}} \leftarrow a_{\mathcal{E}}^{c}$ ***corrected CACC output applied***
18: **else if** $t_{gap}^{est} > T_{gap}^{c}$ & $t_{gap}^{est} < T_{gap}^{A}$ **then**
19: $a_{\mathcal{E}} \leftarrow a_{\mathcal{E}}^{est}$ ***Response Estimator output applied***
20: **else**
21: $a_{\mathcal{E}} \leftarrow a_{\mathcal{E}}^{A}$ ***degrade to ACC***
22: **return** $a_{\mathcal{E}}$

B. Mitigator

For each anomaly captured by the detector, Mitigator computes an alternate response overriding the CACC controller response $a_{\mathcal{E}}$, to neutralize any potential adversarial impact. Mitigator comprises the following components.

1) **Response Estimator** is a pre-trained machine learning model analogous to Predictor, that generates an estimated acceleration $a_{\mathcal{E}}^{\text{est}}$. However, unlike Predictor (and indeed, COMP), it uses only trusted sensory inputs, *e.g.*, relative velocity and position of $\mathcal{E}$ and $\mathcal{P}$.
2) **Plausibility Checker** ensures that Response Estimator's output does not compromise the safety of $\mathcal{E}$, even under attack.

Algorithm 2 describes the Mitigator functionality. In the absence of anomaly, sensory inputs are typically sampled at a lower rate F_{normal}. When Mitigator is invoked to handle an anomaly (lines 7 through 10), the sensor sampling frequency is switched to a higher value F_{max} to generate more accurate sensory data. [3]

The anomalous $a_{\mathcal{P}}$ received and $a_{\mathcal{E}}$ computed using that value, are discarded. Instead, $a_{\mathcal{E}}$ is calculated approximately using the rate of change in the velocity of the $\mathcal{P}$ from the previous time step. Lines 14 through 21 describe the

[3]One of the advantages of utilizing V2V communication for perception is to overcome the limitations of sensor systems such as limited accuracy and higher response times. In some emergent vehicles, ranging sensors are being designed to be sampled at flexible rates: in the presence of V2V, sensors sampling can be switched to a lower value F_{normal} which can be significantly lower than the maximum rated value F_{max}. RACCON includes optimizations that can exploit such flexibility if available, thereby reducing incurred computation cost.

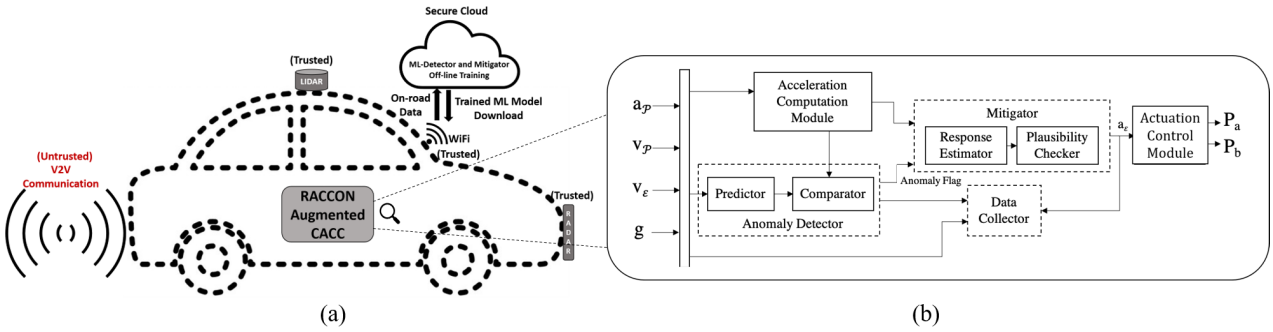


Fig. 3. (a) Vehicle Enabled with RACCON resiliency (b) RACCON 0n-board Architecture - Blocks with dotted-line boundaries introduced for resiliency.

plausibility checker functionality; it accounts for the worst case for safety, *e.g.*, sudden halt of $\mathcal{P}$. The resultant $t$gap is computed for the scenario where $a_{\mathcal{E}}^{\text{est}}$ and corrected $a_{\mathcal{E}}$ were applied. The plausibility checker then determines the optimal choice out of $a_{\mathcal{E}}^{\text{est}}$ and the corrected $a_{\mathcal{E}}$ that is both safe and efficient. If it fails to find such a value, the system falls back to conservative ACC. Consequently, THW never reaches value less than minimum safe threshold Tgap.

C. Data Collector

The Data Collector collects on-road driving data, which is aggregated and periodically communicated to the cloud for improving the ML models (see below). The collected data includes (1) inputs to the CACC controller, *e.g.*, preceding vehicle acceleration, inter-vehicle space headway, and the velocities of the two vehicles; (2) the acceleration value computed by the COMP module of CACC in response to these inputs; and (3) an "anomaly flag" to indicate whether the response is classified as an anomaly by RACCON.

D. Off-Line Cloud Infrastructure

The ML components of RACCON (Predictor and Response Estimator) are trained offline on trusted cloud servers and updated periodically, as new on-road CACC data is made available from the Data Collector modules of different vehicles subscribing to the RACCON service. We assume that these communications cannot be corrupted. This is viable in practice since we do not require real-time communication with the cloud. Data can be transferred from the vehicles when a trusted connection to the cloud is available. RACCON-enabled vehicles securely download the latest instances of trained Predictor and Response Estimators along with a list of anomaly thresholds for different driving environments, prior to CACC engagement in untrusted operating conditions.

V. RACCON Simulation Setup and Evaluation Methodology

A unique aspect of our work is the extensive experimental evaluation of RACCON. In addition to showing the viability of RACCON itself, we believe our experiments provide a roadmap for evaluation of resiliency in other connected vehicular applications as well.

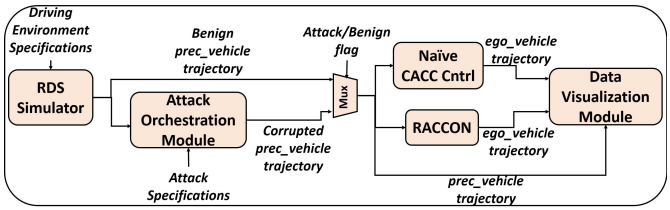


Fig. 4. Data Generation and Attack Orchestration Flow.

A. Simulation Setup and Data Generation

One of our key evaluation goals is to demonstrate the efficacy of RACCON under diverse, realistic driving environments. Programming realistic driving environments as an evaluation testbed is a non-trivial exercise. Consequently, rather than using a traditional simulation environment such as Matlab or Simulink, we use a physical research simulator, we use a physical research simulator RDS1000® [35] as our simulation platform in concert with a software system replicating CACC COMP controller functionality described in Section II-A. Fig 4 represents the data generation and attack orchestration on our customized simulation platform. Physical research simulators enable flexible configuration of various terrains, weather conditions, and environmental parameters, and usually provides pre-configured realistic simulation of lighting, visibility, and road traction attributes. For our experiments, we used 24 driving environments as a cross-product of the following parameters: (i) Road terrain (highway, suburban and urban); (ii) Weather (clear, windy, snowy, rainy); and (iii) Time of day (day, night). The set of parameters (terrain, weather, and time of day) are typically used to analyze (human) driving patterns in the context of safety and congestion analysis [29]. Each of the 24 datasets corresponds to about 15 minutes of driving time and constitutes approximately $90,000$ samples collected at a frequency of 100Hz. The data collected provides the preceding vehicle trajectory; ego vehicle trajectory is computed using the COMP controller from Section II-A. We aggregate data from all environments to create a *global dataset*, which is split 80-20 into training and test data. To enable reproducibility of our experiments, all the data generated from the simulation platform are available publicly through our project website [9].

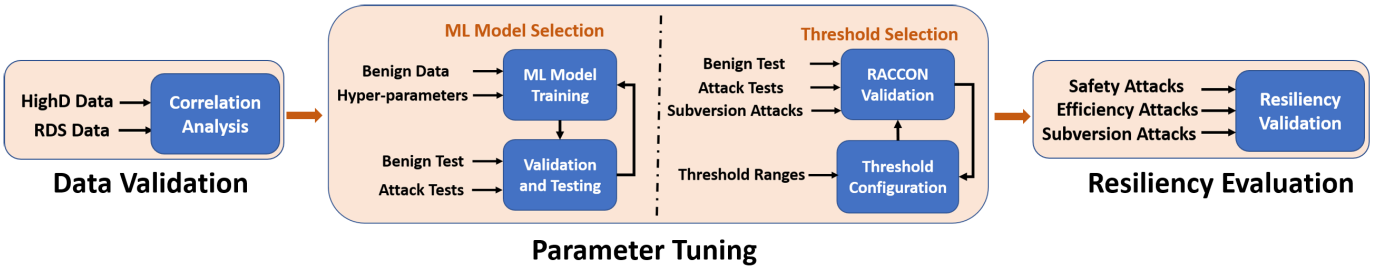


Fig. 5. RACCON Design and Evaluation Flow.

B. Summary of Experiments

Design and evaluation of CAV application resiliency must address a variety of orthogonal facets. Note that within the broad umbrella of ML-based resiliency, the number of architectural parameters available for a security designer to tweak is dauntingly large. This includes the choice of ML model, anomaly threshold, adversary classifications, etc. In addition to evaluating the quality of infrastructure, the methodology must enable systematic estimation of these design parameters. Following is an overview of the experiments performed to design and evaluate RACCON. Fig. 5 presents the three stages of experiments involved: (i) Data Validation, (ii) Parameter tuning, and (iii) Resiliency evaluation. We elaborate on the experiments in Sections VI through Section IX.

1) **Data Validation:** For our conclusions to be meaningful, it is critical that the data we use is realistic. We validate that the vehicular driving patterns reflected in our simulation data conform to real-world patterns from a public dataset. (Section VI)
2) **Parameter Tuning**: Implementing Predictor and Response Estimator functionalities requires selecting and tuning the appropriate ML architecture. We develop a systematic evaluation methodology to address this problem. (Section VII) Additionally, a key factor in the effectiveness of RACCON is the identification of *anomaly threshold*, *i.e.*, the extent of deviation from normal behavior pattern that would be classified as a potential threat. Selecting an appropriate threshold involves balancing the trade-off between maximizing attack detection accuracy and minimizing false alarms.We present a series of experiments to achieve this balance. At the end of this stage, an approximate range for optimal threshold is determined which is further fine-tuned in the next stage. We further fine-tine the threshold to enable robustness against detection subversion attacks (Section VIII).
3) **Resilience Evaluation:** Finally, our evaluation shows the robustness of RACCON against various V2V attacks including collision-causing and efficiency degradation attacks. Various representative instances of known N-day attacks are also orchestrated on RACCON in our analysis (Section IX).

C. Attack Taxonomy and Orchestration Methodology

One critical challenge in evaluating RACCON is to devise an evaluation strategy to comprehensively cover the attack space. Previous works focused on specific attack instances, *e.g.*, Biron *et al.* [3] target jamming and flooding attacks, and Jagielski *et al.* [19] focus on specific mutation attacks. Such evaluation does not provide adequate evidence of resiliency against other potentially unknown attacks.

We address this problem by developing a comprehensive taxonomy of V2V attacks on CACC (Fig. 6) that is used to systematically navigate the attack space. The taxonomy is inspired by threat modeling approaches in hardware and system security [33], but adapted for V2V adversaries. The idea is to represent a V2V attack through three features, *viz.*, stealth, operation, and impact. This feature combination forms a holistic characterization of any attack under the RACCON adversary model. In particular, since the adversary is confined to V2V communications, the only choices for the adversary are to (1) mutate an existing message, (2) fabricate a new message, and (3) prevent the delivery of a message. Correspondingly, since the message payload constitutes the preceding vehicle's acceleration, the impact of an attack can be to (1) increase the probability of collision (by reporting a lower than actual acceleration value), (2) reduce efficiency through an increased headway (by reporting a higher than actual acceleration value), or (3) creating instability (*e.g.*, through random mutation of the actual value). We refer to deviations by a positive bias as *collision attacks* and deviations by a negative bias as *efficiency degradation attacks*. Note that the taxonomy is oblivious to the *mechanics* of the attack (*e.g.*, man-in-the-middle, rogue vehicle, hardware-software modules of the ego vehicle, etc.), but only considers the effect on V2V messages. For instance, delivery prevention operation accounts for jamming, flooding, channel subversion, etc., each of which can be carried out through a variety of ways. Table II shows how the taxonomy accounts for different well-known attacks. The focus on *attack characteristics* rather than the mechanics enables a comprehensive classification of V2V attacks.

We used the taxonomy above to develop a systematic attack orchestration framework. Attacks are represented as 3-tuples, representing the three features identified in the taxonomy. Delivery prevention attacks are realized through intermittent or absent communication. Mutation and fabrication attacks are realized through fake acceleration messages that deviate from ground truth. We consider four different ways for generating fake accelerations:

$$a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} \pm b \tag{4}$$

$$a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} \pm bt \tag{5}$$

TABLE II

REPRESENTATIVE N-DAY ATTACK INSTANCES. ALL RELEVANT COMBINATIONS OF THE OPERATION, FREQUENCY AND IMPACT FEATURES FOR EACH ATTACK MECHANISM INDICATED BY "✓"

Attack Mechanism	Attack Origin		Operation			Frequency			Impact		
	Preceding Vehicle	MITM	Mutation	Fabrication	Delivery Prevention	Discrete	Cluster	Continuous	Collision	Efficiency degradation	String Instability
Message falsification	✓	✓	✓			✓	✓	✓	✓	✓	✓
DoS (Jamming)		✓			✓	✓	✓			✓	✓
DoS (Flooding)	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
Masquerade		✓	✓	✓		✓	✓	✓	✓	✓	✓
Replay		✓		✓		✓	✓		✓	✓	✓
Misdirection		✓			✓	✓	✓			✓	✓

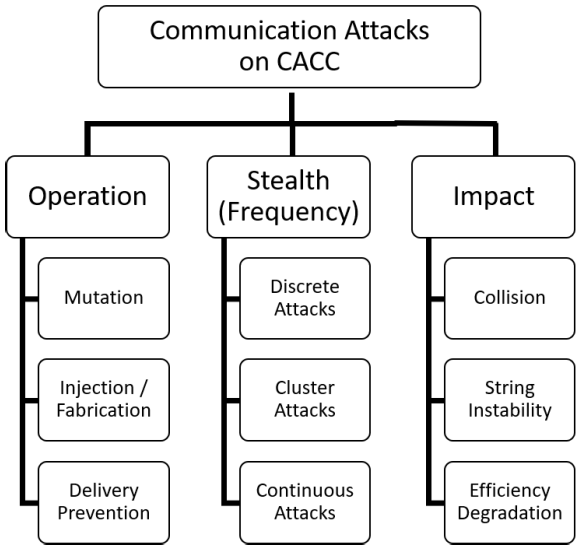


Fig. 6. Taxonomy of Communication Attacks on CACC.

$$a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} \pm bsin(ft) \quad (6)$$

$$a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} \pm random \quad (7)$$

Equation (4) represents a constant bias added to the ground truth. Equations (5) and (6) represent linear and sinusoidal time-varying biases, respectively. Given a specific combination of attack features (*e.g.*, discrete mutation attack with collision as targeted impact), the framework permits attack impact simulation. We use THW (t_{gap}) as a natural measure to quantify the risk of collision or the extent of efficiency degradation. An erratic change in t_{gap} can also potentially indicate string instability in the traffic.

VI. DATA VALIDATION

A key challenge with using simulator data is to ensure that it is realistic. Unfortunately, there is no available repository of sufficient real-world driving data across different driving scenarios. Indeed, the lack of available real-world data is the reason why we rely on simulated data in the first place. To address this problem, we observe that while sustained data over a period of time is unavailable, there are datasets that provide short-duration driving patterns. These snippets can then be used to corroborate data obtained from the simulator under similar driving conditions.

We carried out this experiment with HighD dataset [23] that provides trajectory data corresponding to real vehicles

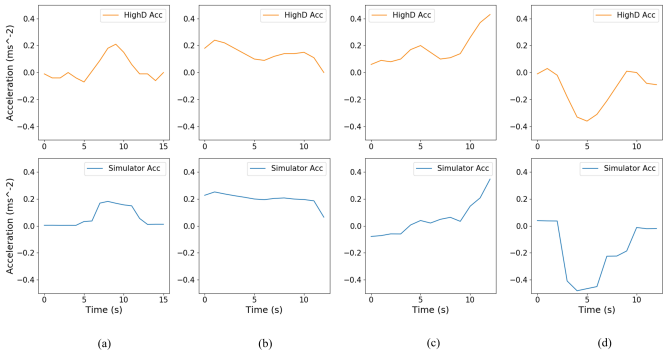


Fig. 7. Correlation Between Simulated Data and HighD. Each plot indicates correlation between the acceleration trajectory of an arbitrary vehicle in HighD and the simulated vehicle.

driving in German highways. The length of individual vehicle trajectories is approximately 15 seconds. We compare acceleration patterns of similar length trajectories collected from the simulator. Fig. 7 shows sample comparisons for four vehicles from HighD data. The results clearly indicate that the acceleration patterns from the simulator correlate closely with HighD data.

VII. ML MODEL SELECTION

Viability of RACCON critically depends on the presumption that the ML components Predictor and Response Estimator can accurately capture anomalous communication and mitigate the adverse effects We can formulate the ML regression problem for these components in two ways: (i) stateless prediction and (ii) time-series prediction. Cumulatively, these result in a prohibitively large space ML architecture choices. It is important to navigate this space systematically and converge to an optimal architecture. The ML model must address two orthogonal requirements: (1) avoid false alarms for benign messages and (2) accurately classify malicious messages as anomalous. Furthermore, it must be possible to perform real-time prediction under the computation and storage constraints of automotive systems. Finally, since driving patterns vary according to driving conditions, we must determine whether each driving environment requires a customized ML model.

A. Identifying ML Architecture

Since detecting malicious activity essentially involves identifying anomalous behavior, it is imperative that the model

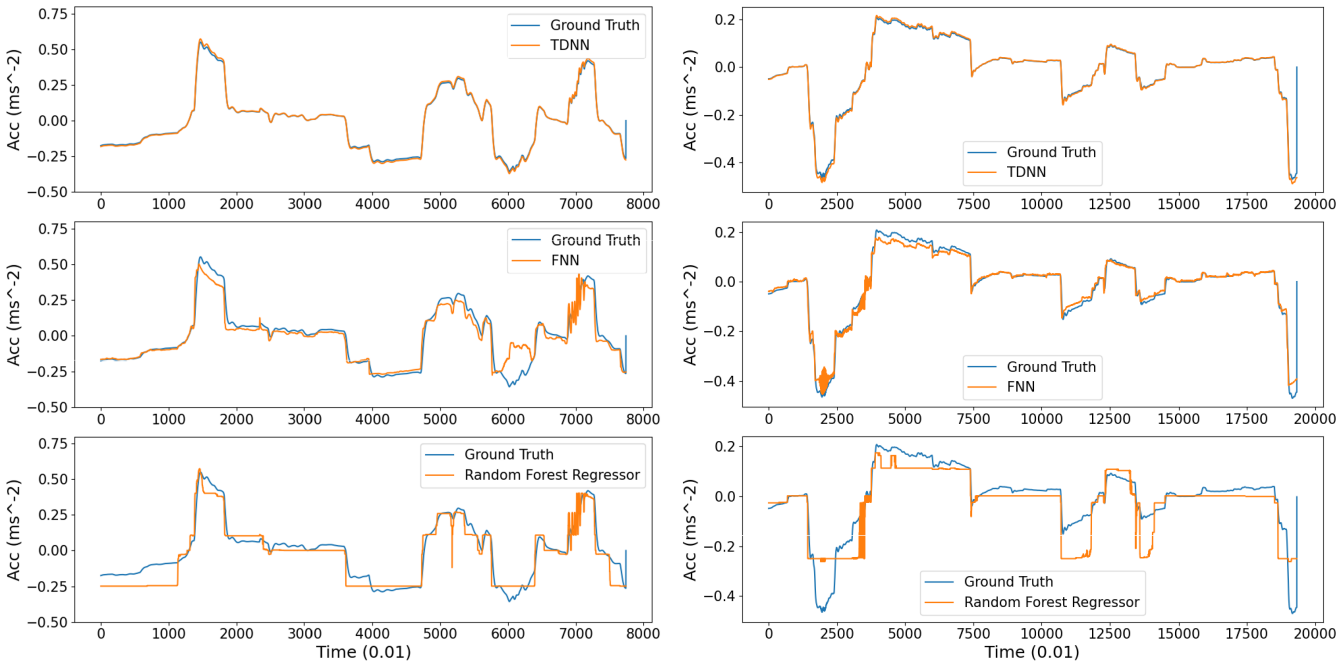


Fig. 8. Prediction of TDNN, FNN and Random Forest in Benign Environments. (a) Highway-Day-Windy. (b) City-Night-Snow.

TABLE III
ARCHITECTURE AND TRAINING HYPER-PARAMETERS FOR ML MODELS

Model	Architecture Hyper-parameters		Training Hyper-parameters	
FNN	No of Hidden Layers	1	No of Training Epochs	20
	No of Hidden Units	15	Feature Scaling	Minmax
	Hidden Layer Activation	ReLU*	Learning Algorithm	SGD*
TDNN	Window Length	10	No of Training Epochs	20
	No of Hidden Units	15	Feature Scaling	Minmax
	Hidden Layer Activation	ReLU*	Learning Algorithm	SGD*
SVM	Kernel	rbf	Feature Scaling	Minmax
	Regularization	100	Epsilon	0.1
RF	No of trees	100	Minimum Sample Split	2
	Maximum Depth	10	Split Criteria	Gini Index
LSTM	No of LSTM Layers	1	No of Training Epochs	20
	LSTM Units	50	Feature Scaling	Minmax
	Activation function	tanh	Learning Algorithm	Adam

*SGD: Stochastic Gradient Descent
*ReLU: Rectified Linear Activation

learns NBM (*i.e.*, estimating the normal behavior of CACC controller) accurately for effective performance in adversarial settings. Furthermore, *efficiency* of a resiliency solution depends primarily on the prediction accuracy under benign scenarios, since most of the messages encountered by vehicles in field are likely benign. Our methodology entails the following steps to determine the appropriate ML architecture from a potentially large candidate set.

1) Find a set of candidate architectures that can satisfy automotive resource constraints.
2) Discard candidates that do not provide acceptable prediction accuracy under benign conditions.
3) Of the remaining candidates, select the architecture with highest accuracy under malicious conditions.

In our evaluations, our candidate set included five architectures: Random Forest Regressor (RF), Support Vector Machine (SVM), and Feed-forward Neural Network (FNN) are examined for stateless prediction; Univariate Time Delayed Neural Network (TDNN) and Multivariate Long Short-Term Memory (LSTM) network are examined for time-series prediction. Architectures more sophisticated than LSTM were estimated to be too complex, given the constraints of automotive systems. The architectural details of each ML model considered are presented in Table III. For these candidates, we apply a two-step triage process based on prediction accuracy in benign environment. In the first step, we compute the Mean Absolute Error (MAE) in prediction, under six different driving environments, for each ML architecture. This provides a "coarse" evaluation of accuracy and facilitates identification of a small subset of candidates (Table IV). Clearly RF, TDNN, and FNN show much better accuracy than SVM and LSTM. In the next step, we examine them more closely to identify any local "kinks". Fig. 8 plots the accuracy of Predictor in two different environments. Note that RF is ineffective in capturing minute variations in acceleration (indicated by several flat lines in prediction). This behavior can be attributed to the fact that the RF regressor ignores minute variations in the data as noise. Since tracking such variations is critical for accurate anomaly detection, RF is discarded as a viable candidate.

Remark 2: One can ask about the completeness of the candidate set itself. Our choice is governed by the desire to choose representatives including traditional ML models as well as deep learning models, with the over-arching requirement that the models must be light-weight to facilitate deployment on automotive on-board platforms. Our results essentially suggest that a traditional ML model may not be sufficient while a simple deep learning model is adequate. However, our goal is not to advocate a specific ML model. Rather, for a specific underlying CACC controller and given a set of candidate

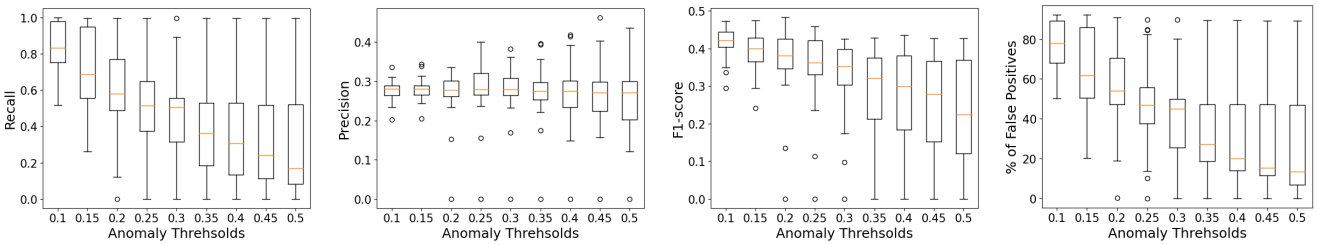


Fig. 9. Distribution Box-plots of Detection metrics vs Anomaly Threshold for 24 Driving Environments. Plots (a) through (c) show the distribution of recall, precision and f1-score under a sinusoidal attack. Plot (d) shows the distribution of false positive percentage in benign conditions.

TABLE IV

MEAN ABSOLUTE ERROR IN THE PREDICTION OF EGO VEHICLE ACCELERATION UNDER SIX DIFFERENT TEST DRIVING ENVIRONMENTS

Test Environment	ML Model				
	RF	FNN	LSTM	TDNN	SVM
HW-Day-Windy	0.040	0.021	0.155	0.007	0.440
HW-Day-Snow	0.149	0.177	1.640	0.027	1.057
HW-Night-Clear	0.101	0.116	0.985	0.021	0.787
SU-Night-Snow	0.075	0.089	1.166	0.010	0.510
SU-Night-Windy	0.199	0.310	1.130	0.035	0.364
City-Night-Windy	0.062	0.073	0.201	0.010	0.987

models, we propose going through the steps described to systematically identify the optimal architecture.

FNN and TDNN are further examined under simulated attacks to determine anomaly detection and mitigation efficacy. In each attack, malicious acceleration values are generated by adding a bias (constant or sinusoidal) to the ground truth. Clearly, FNN performs significantly better than TDNN in mitigating attacks, as indicated by the resultant THW values in Table V. *Based on these results, FNN is determined as the appropriate ML architecture for the RACCON detection system.*. [4]

VIII. THRESHOLD SELECTION

RACCON resiliency depends on the choice of the anomaly threshold: a threshold higher than optimal may lead to reduced detection accuracy, while a lower threshold may lead to increased false alarms in detection. High degree of false alarms results in inefficient invocation of RACCON's Plausibility checker. Although plausibility checking computation is lightweight, the cumulative overhead can become significant since on-road vehicles operate mostly under benign conditions. An optimal threshold would enable safety as well as efficiency under adversarial scenarios while incurring minimal performance overhead in benign conditions. Our threshold estimation methodology works in three stages:

1) Identify an acceptable threshold range for adversarial scenarios.
2) Compute an approximate threshold value within the range by accounting for performance overhead under benign conditions.
3) Fine-tune the value to optimize for detection subversion attacks.

A. Computing Acceptable Threshold Range

We use three detection metrics: *recall*, *precision*, and *f1-score*, to estimate the quality of resiliency under attacks. A high precision value reflects smaller percentage of false alarms while a high recall reflects smaller percentage of undetected anomalies. A high f1-score (computed as the harmonic mean of recall and precision) indicates a combination of high precision as well as recall. We prioritize recall over precision since it is important to capture any anomaly that can possibly cause an undesired impact. Fig. 9(a), (b), and (c) show the distribution box-plots of the three detection metrics over all 24 environments. The evaluation is carried out under a clustered sinusoidal attack corrupting about 25% of the V2V messages. This attack is representative since it includes characteristics of both discrete and continuous attacks, and incorporates both positive and negative biases within the same attack instance. Note that recall degrades as the anomaly threshold increases from 0.1 to 0.5. The best recall values (close to 1) are observed for thresholds in the range 0.1-0.2; however, the corresponding precision values are only 0.25-0.35, indicating higher number of false alarms. Consequently, f1-scores reach an optimal value ($\sim$ 0.4) for smaller values of the threshold (0.05-0.25) but decrease as the threshold increases.

Remark 3: Observe from Fig. 9 that the f1-score boxes are not tightly packed around the mean, implying that the optimal anomaly threshold (based on f1-score) can vary across environments. Consequently, RACCON supports on-the-fly adjustment of threshold based on the current environment, using parameters from maps (e.g., *location, terrain, etc.), ambient weather, and clocks.*

B. Performance Overhead in Benign Conditions

Fig. 9(d) illustrates the distribution of false positives under benign conditions for thresholds ranging 0.1-0.5. Since larger thresholds result in low recall (see above), values larger than 0.5 are disregarded. As with f1-score, thresholds in the range 0.05-0.25 have a high variance, indicating fluctuation with changing driving environment. The optimal anomaly threshold is selected by balancing the trade-off between better coverage

[4]We believe the better performance of FNN over TDNN is due to the stateless design of the CACC controller. The stateless FNN model captures the context well and approximates the controller behavior while time-dependant regression models learn spurious temporal dependencies making them ineffective in detecting anomalous inputs.

TABLE V

RESULTANT THW FOR TDNN AND FNN PREDICTORS UNDER FOUR DIFFERENT ATTACKS

Time Headway	Cluster Attack (Bias:1.5)			Cluster Attack (Bias:-0.8)			Continuous Attack (Bias:0.1)			Continuous Attack (Bias:sin(0.05t))		
	FNN	TDNN	Naive CACC	FNN	TDNN	Naive CACC	FNN	TDNN	Naive CACC	FNN	TDNN	Naive CACC
THW $<$ 0.55s	0%	30.85%	80.64%	0%	0%	0%	0%	63.22%	63.22%	0%	20.47%	21.14%
THW: $\{0.55 - 0.75s\}$	100%	65.81%	19.36%	100%	55.89%	34.24%	100%	36.78%	36.78%	100%	77.97%	78.86%
THW $>$0.75s	0%	3.34%	0%	0%	44.11%	65.76%	0%	0%	0%	0%	1.56%	0%

TABLE VI

ANOMALY THRESHOLD AND SUBVERSION DETECTABILITY UNDER ATTACKS OF VARYING STEALTH FACTOR

Anomaly Threshold	False Positives Benign Condition	Subversion Detectability Index					
		Continuous (Tolerable bias: 0.04)		Cluster (Tolerable bias: 0.1)		Discrete (Tolerable bias: 5.0)	
		Min. constant bias	Min. sinusoidal bias	Min. constant bias	Min. sinusoidal bias	Min. constant bias	Min. sinusoidal bias
0.25	0%	0.35	0.25sinft	0.4	0.35sinft	0.5	3sinft
0.2	2.96%	0.3	0.2sinft	0.3	0.3sinft	0.35	1sinft
0.18	10.74%	0.3	0.2sinft	0.3	0.3sinft	0.35	0.35sinft
0.15	11.91%	0.01	0.01sinft	0.03	0.02sinft	0.25	0.25sinft
0.13	21.2%	0	0	0.0001	0.0001sinft	0.01	0.01sinft
0.12	58.1%	0	0	0	0	0	0

TABLE VII

RESILIENCY EVALUATION UNDER COLLISION ATTACKS

	Spurious communication: Linear function of ground truth								
	Continuous Attack (linear bias= 0.3t)			Cluster Attack (constant bias= +0.8)			Discrete Attack (constant bias= +2.0)		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	84.54%	0%	0%	73.83%	0%	0%	0%
THW: $\{0.55 - 0.75s\}$	100%	54.01%	15.46%	100%	51.13%	26.17%	100%	55.28%	100%
THW $>$0.75s	0%	45.99%	0%	0%	48.86%	0%	0%	44.72%	0%
Collision	No	No	Yes	No	No	Yes	No	No	No
	Spurious Communication: Sinusoidal function of ground truth								
	Continuous Attack (bias= 0.5sin(0.02t))			Cluster Attack (bias= 0.8sin(0.03t))			Cluster Attack (bias= sin(0.05t))		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	33.03%	0%	0%	12.60%	0%	0%	3.81%
THW: $\{0.55 - 0.75s\}$	100%	54.64%	66.97%	100%	54.81%	87.40%	100%	53.94%	96.19%
THW $>$0.75s	0%	45.36%	0%	0%	45.19%	0%	0%	46.06%	0%
Collision	No	No	Yes	No	No	No	No	No	No

TABLE VIII

RESILIENCY EVALUATION UNDER EFFICIENCY DEGRADATION ATTACKS

	Spurious communication: Linear function of ground truth								
	Continuous (linear bias= -0.3t)			Cluster (constant bias= -0.8)			Discrete (constant bias= -2.0)		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	0%	0%	0%	0%	0%	0%	0%
THW: $\{0.55 - 0.75s\}$	100%	55.42%	21.55%	100%	55.25%	18.85%	100%	54.83%	100%
THW $>$0.75s	0%	44.58%	78.45%	0%	44.75%	81.15%	0%	45.17%	0%
Maximum THW	0.65s	1.56s	1.79s	0.65s	1.55s	1.54s	0.65s	1.54s	0.70s
	Spurious Communication: Sinusoidal function of ground truth								
	Continuous Attack (bias= -0.5sin(0.02t))			Cluster Attack (bias= -0.8sin(0.03t))			Cluster Attack (bias= -sin(0.05t))		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	0%	0%	0%	0%	0%	0%	0%
THW: $\{0.55 - 0.75s\}$	100%	54.67%	79.97%	100%	54.14%	94.01%	100%	54.49%	98.35%
THW $>$0.75s	0%	45.33%	20.03%	0%	45.86%	5.99%	0%	45.51%	1.65%
Maximum THW	0.65s	1.56s	0.83s	0.65s	1.55s	0.79s	0.65s	1.54s	0.75s

under attack conditions and minimal overhead in benign conditions.

As an example, we obtain the optimal threshold for the environment *Highway-Day-Windy* as follows. First, we determine the ballpark range 0.1-0.25 that gives the best f1-score (recall close to 1 and precision close to 0.4). We eliminate thresholds less than 0.1 to keep the false positives below 30%, refining the range to 0.13-0.25. This is fine-tuned after evaluation under detection subversion to obtain the optimal choice 0.15 (Section VIII).

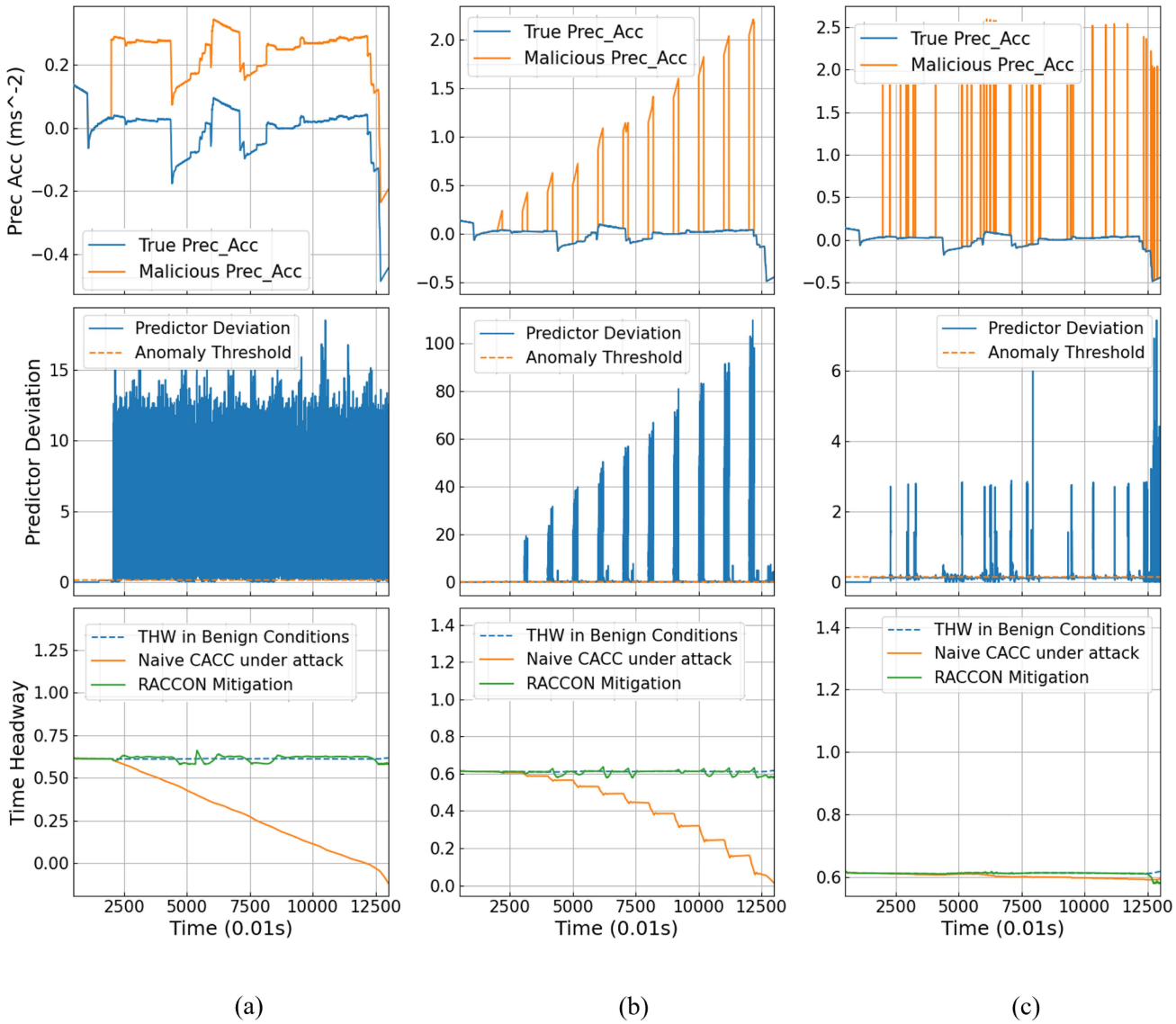


Fig. 10. RACCON Resiliency under Sample Collision Attacks. (a) Continuous Attack constant bias +0.25. (b) Cluster Attack with linear bias +0.1t. (c) Discrete Attack with constant bias +2.5.

C. Threshold Fine-Tuning: Detector Subversion

The fact that RACCON is an ML-based framework can make it vulnerable to adversaries subverting the learning and prediction systems themselves. Such adversaries can create anomalous data that is nevertheless accepted as normal by the detector, thereby bypassing any mitigation against the attack. We call these attacks *detector subversion*.

Obviously, a very low selection of anomaly threshold can ensure high robustness against detector subversion. However, recall from Section VIII that a low anomaly threshold can result in high false alarms. Consequently, we fine-tune the threshold value within the ballpark range obtained from Section VIII, balancing the trade-off. We use the following parameters in our analysis.

- *Tolerable Bias:* This is the maximum bias added to the ground truth, beyond which there is a perceptible impact on the target vehicle's safety or efficiency.
- *Subversion Detectability Index:* This is the minimum bias added to ground truth, that can be successfully captured by the detection system.
- *False Positives in Benign Conditions:* This is the percentage of normal communication messages, incorrectly tagged as anomalies by RACCON in benign operating conditions.

The goal is to determine the optimal anomaly threshold which enables the detection of every attack beyond the tolerable bias, while keeping the number of false positives small.

Table VI presents results for threshold choices for a representative driving environment, *Highway-Day-Windy*. We determined the approximate optimal threshold range for this environment to be 0.12-0.25 previously. To fine-tune for resiliency under detector subversion, we determine the tolerable bias for attacks of varying stealth factor; note that it is much smaller for a continuous attack (0.04) than a discrete attack (5.0). For optimal threshold, the subversion detectability index should

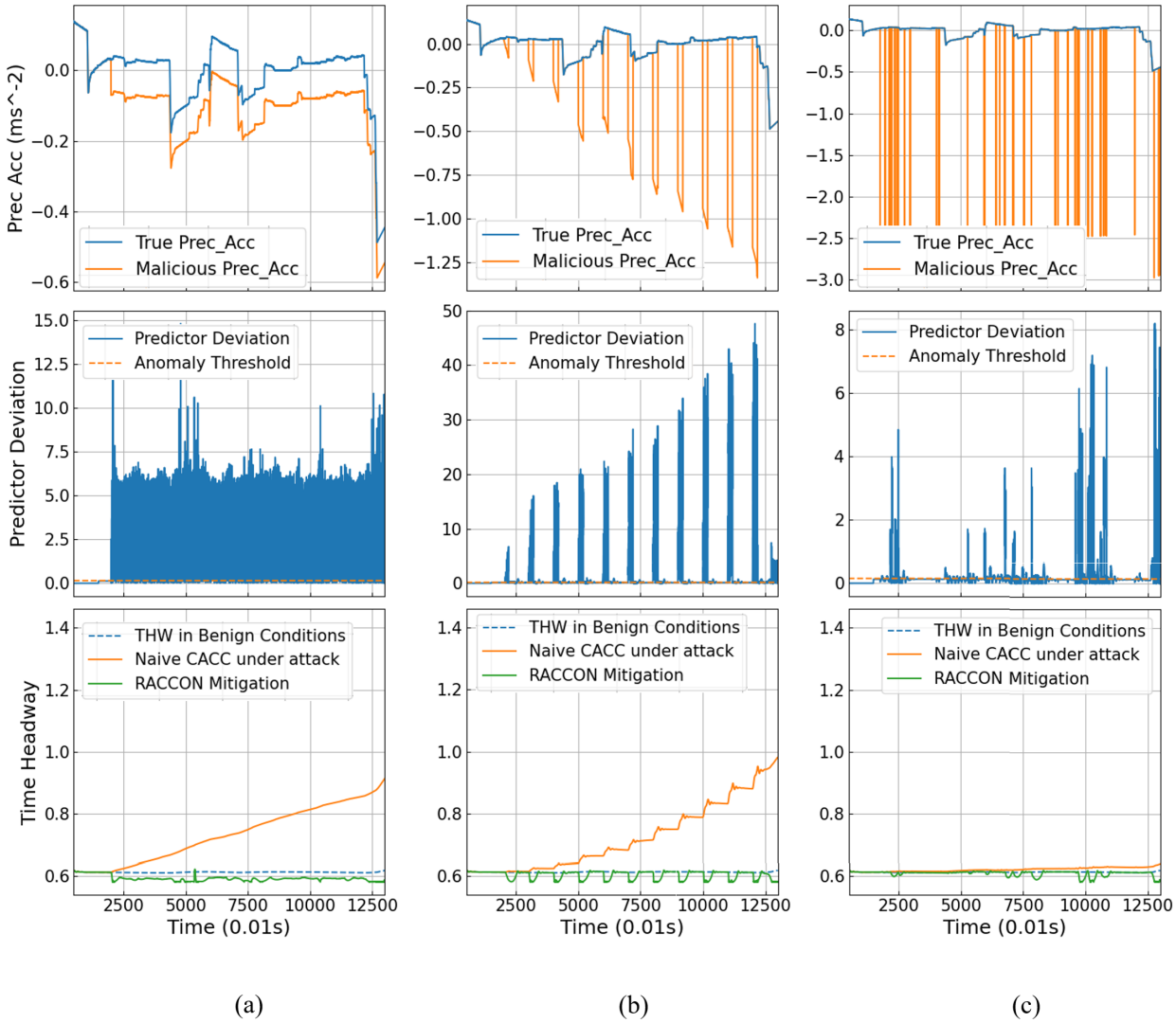


Fig. 11. RACCON Resiliency under Sample Efficiency Degradation Attacks. (a) Continuous Attack (constant bias -0.1). (b) Cluster Attack (linear bias -0.06t). (c) Discrete Attack (constant bias -2.5).

be less than the tolerable bias for each class of attack. The highlighted row shows the optimal choice of the anomaly threshold (0.15), since it has the minimum fraction of false positives out of all the choices providing acceptable subversion detectability.

IX. RACCON Resiliency Evaluation

We performed extensive evaluation of RACCON resiliency using our flexible attack orchestration framework. Note that related work on detecting V2V compromises (see Section II-C) does not include real-time mitigation; the only implied mitigation entails degrading to ACC (conservative controller action relying only on the trusted sensor systems). To provide a fair evaluation of RACCON, we compare it with (1) Naive CACC with no resiliency; and (2) CACC that degrades to ACC as mitigation. One way to view this evaluation is as a comparison between two extremes for safety-compromising attacks: the naive CACC controller is efficient but at the cost of safety, while degradation to ACC provides safety guarantee but at a significant efficiency cost (since ACC headway is much larger than CACC). The goal of RACCON is to enable optimal efficiency while guaranteeing safety, by maintaining THW in the range 0.55-0.75s.

A. Collision and Efficiency Degradation Attacks

Tables VII and VIII show the numerical results for evaluation under six representative collision and efficiency attack scenarios. Figs. 10, 11 and Fig. 12 provide visual representation of RACCON mitigation. We showcase attacks that are impactful yet hard to detect due to small biases or infrequent malicious activities. In each table, we present a comparison between RACCON, mitigation degrading to ACC, and naive CACC with no resiliency. Tabular entries indicate the amount of time (as percentages of total driving time) during which the vehicle experiences THW values falling within a certain range. Based on these results we make the following observations.

- **Collision Attacks:** RACCON successfully mitigates the collision attacks, maintaining THW within the optimal range of 0.55-0.75s at all times. CACC without any resilience results in unsafe headway of less than 0.55s, and eventually, collision in some cases. Degrading to

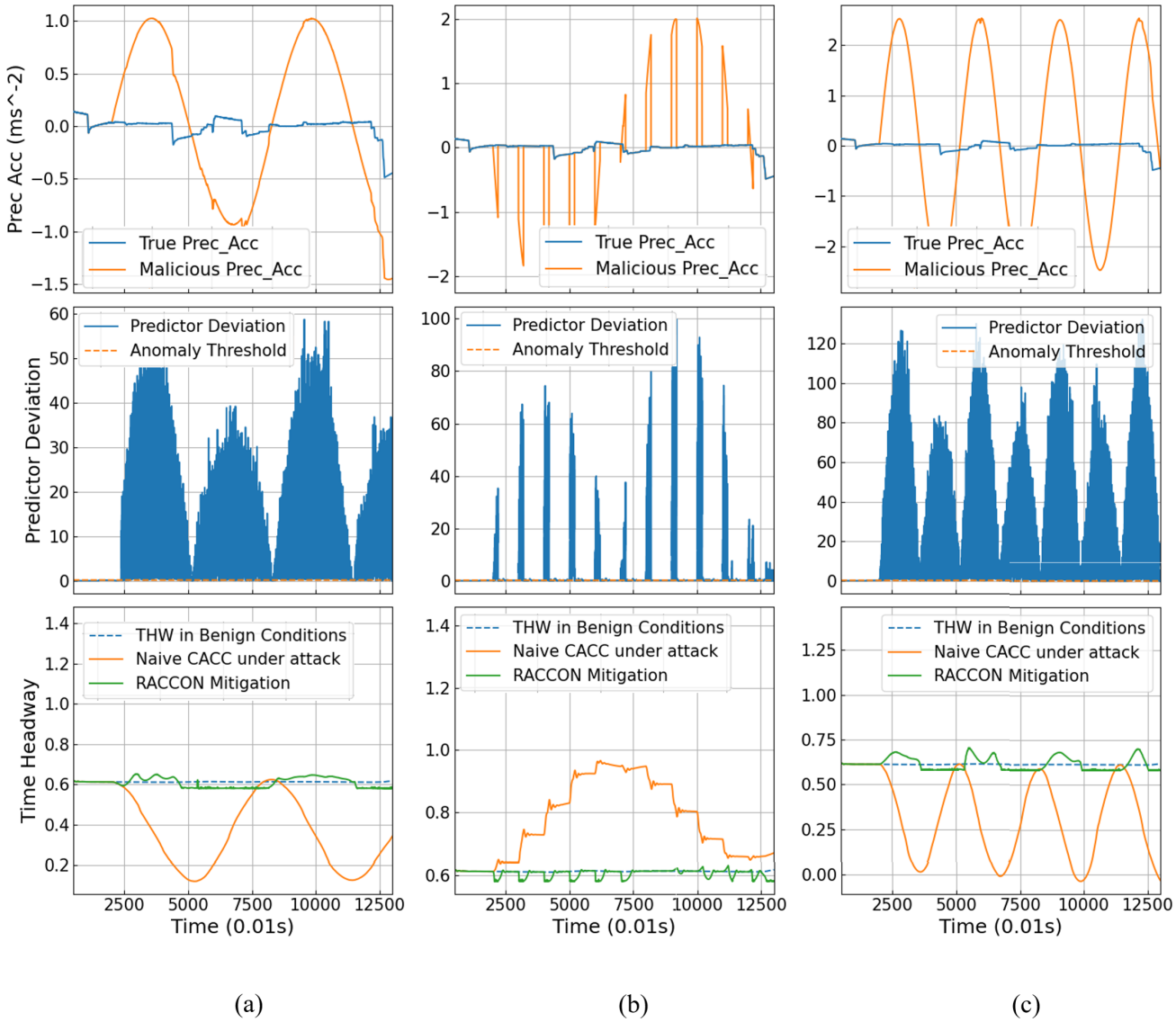


Fig. 12. RACCON Resiliency under Sample Sinusoidal Attacks. (a) Continuous Attack (bias sin(0.1t)). (b) Cluster Attack (bias -2sin(0.3t)). (c) Continuous Attack (bias 2.5sin(0.2t)).

TABLE IX
RESILIENCY EVALUATION UNDER RANDOM MUTATION AND DELIVERY PREVENTION ATTACKS

	Random Mutation Attacks								
	Continuous (random bias=-2.0,2.0)			Cluster (random bias=-2.0,2.0)			Discrete (random bias=-2.0,2.0)		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	0%	0%	0%	0%	0%	0%	0%
THW: $\{0.55 - 0.75s\}$	100%	54.07%	100%	100%	55.69%	100%	100%	55.20%	100%
THW $>$0.75s	0%	45.93%	0%	0%	44.31%	0%	0%	44.80%	0%
Max THW	0.65	1.54	0.73	0.65	1.55	0.65	0.65	1.54	0.65
	Delivery Prevention Attacks								
	Intermittent (frequency= 0.2Hz, duration=1.5s)			Intermittent (frequency= 0.1Hz, duration=2s)			Intermittent (frequency= 0.2Hz, duration=5s)		
	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC	RACCON	Degrade ACC	Naive CACC
THW $<$ 0.55s	0%	0%	0%	0%	0%	0%	0%	0%	3.29%
THW: $\{0.55 - 0.75s\}$	100%	54.86%	100%	100%	54.88%	100%	100%	54.92%	96.71%
THW $>$0.75s	0%	45.14%	0%	0%	45.12%	0%	0%	45.08%	0%
Max THW	0.65	1.54	0.65	0.65	1.54	0.65	0.65	1.54	0.66

ACC prevents collisions, but THW is above 0.75s for over 40% of the attack duration.

- **Efficiency Degradation Attacks:** With RACCON, the maximum THW is around 0.65s. Without resilience,

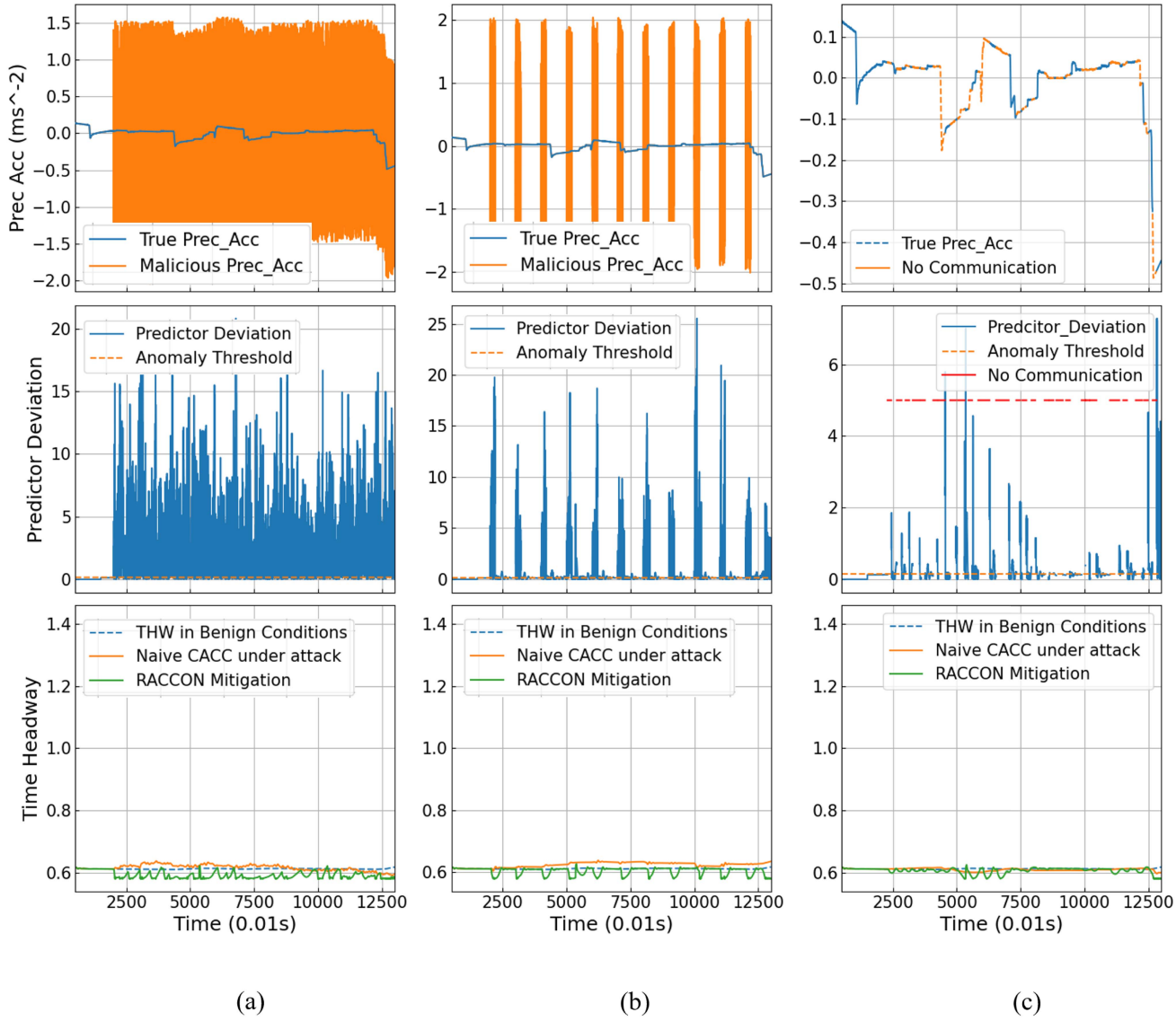


Fig. 13. RACCON Resiliency under Random Mutation and Delivery Prevention Attacks: Comparison between RACCON and naive CACC with no resiliency, in terms of resultant THW; (a) Continuous Attack (random bias -1.5, 1.5); (b) Cluster Attack (random bias -2.0, 2.0); (c) Intermittent communication.

THW reaches 1.8s. Degrading to ACC also results in THW as high as 1.5s.

Remark 4: In addition to the evaluations above, it would have been compelling to provide a direct comparison of RACCON with related work. Unfortunately, as mentioned in Section II-C, we are aware of no other research that targets real-time resiliency of CACC against V2V attacks. In particular, related approaches [4], [19] to anomaly detection do not include mitigation approaches, and implicitly assume degradation to ACC as a potential mitigation. Nevertheless, we thought it illustrative to compare RACCON with anomaly detection algorithms of Jagielski et al. *[19] and Alotibi* et al. *[4]. To achieve this, we orchestrated* 9 *different sinusoidal attacks as described in their work. Fig 12 and Tables VII and VIII show that RACCON resiliency effectively mitigates both collision-causing and efficiency degrading sinusoidal attacks by maintaining safe and efficient time headway during the entirety of each attack instance. Note that the implicit mitigation of degrading to ACC, on the other hand, would result in a significant loss in efficiency by increasing THW to the ACC values (*e.g.*, 1.2s from 0.55s).*

B. Random Communication and Delivery Prevention Attacks

We also studied effects of random message mutation and delivery prevention (Table IX and Fig. 13). These attacks have much less impact than Collision and Efficiency Degradation attacks. A critical aspect of resiliency evaluation is to ensure it does not incur high mitigation overhead. Both RACCON and naive CACC maintain t_{gap} within the ideal range at all times; however, degrading to ACC incurs significant efficiency loss.

C. N-Day Attacks

Attacks orchestrated in Sections IX-A and IX-B systematically cover the taxonomy discussed in Section V-C. Since our taxonomy comprehensively represents the whole V2V attack spectrum, it is established from our evaluation results that

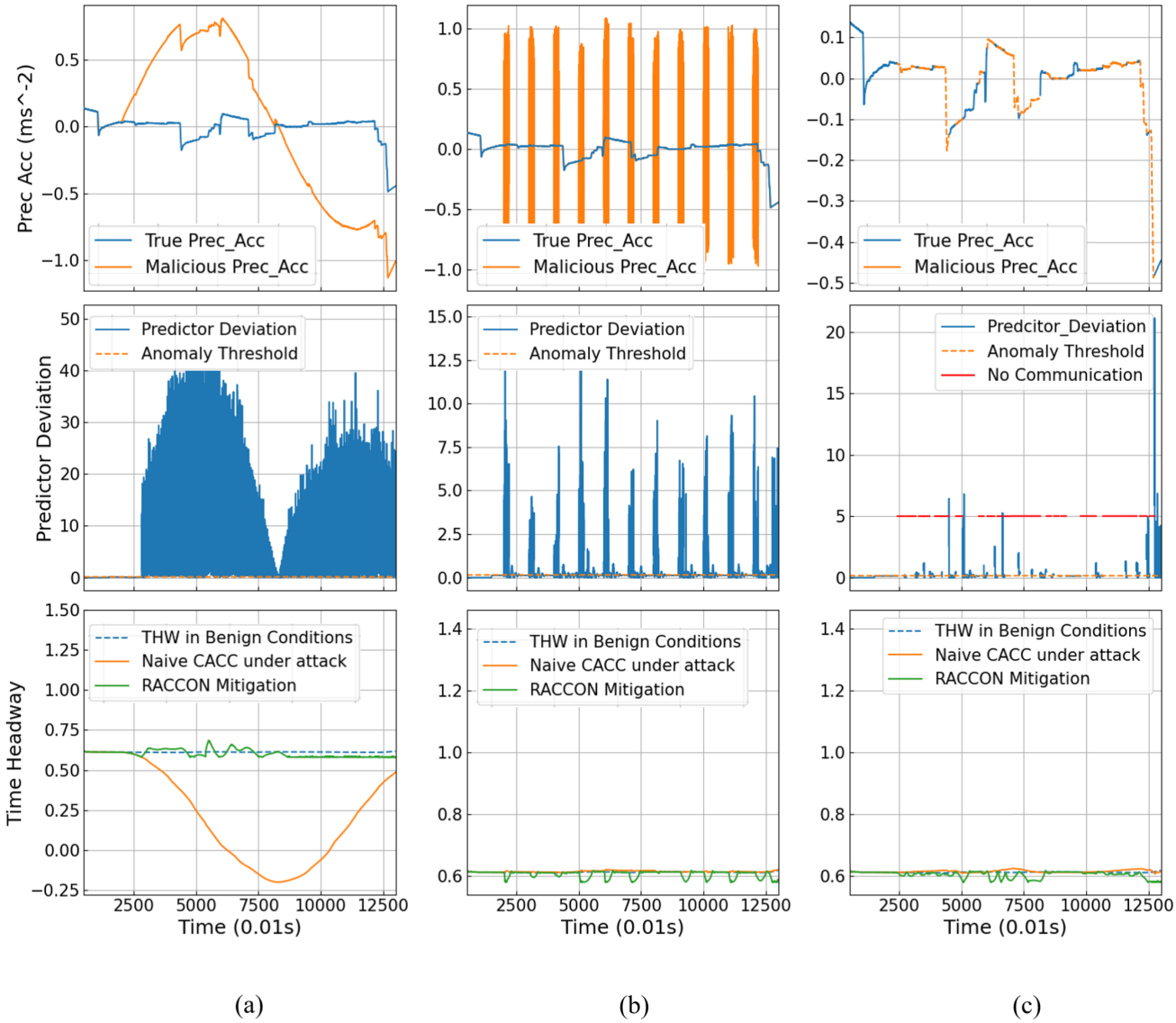


Fig. 14. RACCON Resiliency under Representative N-day Attacks. (a) MITM Attack with continuous sinusoidal bias. (b) Flooding Attack with cluster random bias. (c) DoS Attack with intermittent communication.

RACCON is robust against any arbitrary V2V attack under the threat model, including both known (N-day) and unknown (0-day) attacks. Nevertheless, it is illustrative to directly evaluate RACCON against some known attacks. In this section, we consider three well-known attacks, *e.g.*, Man-in-the-Middle (MITM), Denial-of-Service (DoS) through Jamming, and DoS through Flooding.

- **MITM Attack:** We instantiate an MITM adversary that mutates the preceding vehicle acceleration values by adding a continuous sinusoidal bias, using the function $0.8 \sin 0.05t$.
- **DoS through Jamming:** We implement a DoS attack in which the adversary jams the communication channel, preventing delivery of (legitimate) V2V messages. The channel is jammed for 2 seconds once every 20 seconds.
- **DoS through Flooding:**. The adversary floods the communication channel with fabricated packets that interfere with delivery of legitimate communication. We add fabricated packets in bursts, once every 10 seconds, for a duration of 2 seconds.

Fig. 14 illustrates RACCON mitigation efficacy under these attacks. It maintains t_{gap} close to ideal at all times, while CACC without resiliency results in t_{gap} of less than 0.55s for MITM. Mitigation based on fallback to ACC results in significant efficiency degradation for the jamming attack.

X. Conclusion and Future Work

We have presented what we believe is the first comprehensive resiliency framework for CACC against V2V attacks. Our work uses machine learning to predict the ego vehicle's responses, and capture communication anomalies in real-time, based on deviation between the predicted and actual responses. We also developed a robust real-time mitigation technique that can effectively nullify the adverse effects of anomalous

communication. A unique feature of this mitigation is to guarantee safety while preserving efficiency. Unlike systems that degrade to ACC in response to an anomaly, our solution enables the target vehicle to safely engage in CACC even under attack. We have also developed one of the most comprehensive experimental frameworks for resiliency evaluation, based on a taxonomy of adversaries capturing the entirety of the V2V attack spectrum. Our experiments clearly demonstrate the viability of RACCON as a means for providing resiliency in CACC under V2V attacks.

A unique aspect of RACCON is *real-time resiliency*, contrasting with related works that target offline detection. This requirement has guided several components of RACCON's design and evaluation. First, while all related ML-based anomaly detection approaches focus on identifying discrepancies in controller *inputs*, RACCON is designed to monitor the controller's *response*. This permits RACCON to correct the erroneous response appropriately and minimize the impact of anomalous (and potentially malicious) inputs on the ego vehicle. Second, the need for resiliency requires us to determine the severity of the attack: an attack is impactful and needs mitigation if it results in the ego vehicle performing an unsafe or inefficient action. This requirement has also led to the understanding of the trade-offs between stealth and impact, *e.g.*, clustered and continuous attacks are more impactful (and less stealthy) than discrete attacks. Third, real-time requirements together with resource constraints of automotive platform force consideration of trade-off between accuracy and computation efficiency of ML models. Finally, the trade-off between robustness and performance has guided our methodology for anomaly threshold computation.

We should note however that RACCON is not a panacea. In particular, there are inherent challenges in adopting any ML-based solution. Careful data collection and processing required for ML-model training is resource-consuming and data availability is limited. We have demonstrated how to address these challenges through: (i) customized simulation platform based on physical automotive simulator for realistic data collection and (ii) effective evaluation mechanisms based on a comprehensive attack taxonomy derived from the adversary model.

In future work, we will explore extension of RACCON to other connected car applications. We will also augment RACCON with existing techniques for additionally detecting sensor attacks, resulting in more robust CACC.

ACKNOWLEDGMENT

The authors would like to thank Sattanaathan Thayumanan for his contributions to an earlier version of the paper, and the anonymous reviewers for their valuable feedback and suggestions.

REFERENCES

[1] E. G. Abdallah, M. Zulkernine, Y. X. Gu, and C. Liem, "Towards defending connected vehicles against attacks," in *Proc. 5th Eur. Conf. Eng. Computer-Based Syst.*, Aug. 2017, pp. 1–9.

[2] A. Abdo, S. M. B. Malek, Z. Qian, Q. Zhu, M. Barth, and N. Abu-Ghazaleh, "Application level attacks on connected vehicle protocols," in *Proc. 22nd Int. Symp. Res. Attacks, Intrusions Defenses (RAID)*, 2019, pp. 459–471.

[3] Z. A. Biron, S. Dey, and P. Pisu, "Real-time detection and estimation of denial of service attack in connected vehicle systems," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 12, pp. 3902–3983, Feb. 2018.

[4] F. Alotibi and M. Abdelhakim, "Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3468–3478, Jun. 2021.

[5] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal, "Platoon management with cooperative adaptive cruise control enabled by VANET," *Veh. Commun.*, vol. 2, no. 2, pp. 110–123, Apr. 2015.

[6] C. Bergenhem *et al.*, "Vehicle-to-vehicle communication for a platooning system," *Proc. Social Behav. Sci.*, vol. 48, pp. 1222–1233, Apr. 2012.

[7] C. Bergenhem, S. Shladover, E. Coelingh, C. Englund, and S. Tsugawa, "Overview of platooning systems," in *Proc. 19th ITS World Congr.*, Vienna, Austria, Oct. 2012, pp. 1–8.

[8] S. Boddupalli, A. Hegde, and S. Ray, "Replace: Real-time security assurance in vehicular platoons against V2 V attacks," in *Proc. IEEE Int. Intell. Transp. Syst. Conf. (ITSC)*, Sep. 2021, pp. 1179–1185.

[9] S. Boddupalli, A. S. Rao, and S. Ray. *RACCON: Resilient Cooperative Adaptive Cruise Control*. Accessed: Jan. 22, 2022. [Online]. Available: http://sandip.ece.ufl.edu/projects/automotives/raccon.html

[10] A. Buzachis, A. Celesti, A. Galletta, M. Fazio, and M. Villari, "A secure and dependable multi-agent autonomous intersection management (MA-AIM) system leveraging blockchain facilities," in *Proc. IEEE/ACM Int. Conf. Utility Cloud Comput. Companion (UCC Companion)*, Dec. 2018, pp. 226–231.

[11] S. Checkoway *et al.*, "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX*, vol. 4. San Francisco, CA, USA, 2011, p. 2021.

[12] S. Dadras. *Cybersecurity for Autonomous Vehicle Platooning*. Accessed: Jan. 22, 2022. [Online]. Available: https://digitalcommons.usu.edu/cgi/viewcontent.cgi?article=1559&context=researchweek

[13] R. G. Dutta, F. Yu, T. Zhang, Y. Hu, and Y. Jin, "Security for safety: A path toward building trusted autonomous vehicles," in *Proc. Int. Conf. Comput.-Aided Design*, Nov. 2018, pp. 1–6.

[14] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," in *Proc. SAE Tech. Paper Ser.*, Mar. 2017, pp. 1–9.

[15] G. Giordano, M. Segata, F. Blanchini, and R. L. Cigno, "The joint network/control design of platooning algorithms can enforce guaranteed safety constraints," *Ad Hoc Netw.*, vol. 94, Nov. 2019, Art. no. 101962.

[16] P. Golle, D. Greene, and J. Staddon, "Detecting and correcting malicious data in VANETs," in *Proc. 1st ACM Workshop Veh. Ad Hoc Netw. (VANET)*, 2004, pp. 29–37.

[17] S. Gyawali and Y. Qian, "Misbehavior detection using machine learning in vehicular communication networks," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[18] M. Iorio, F. Risso, R. Sisto, A. Buttiglieri, and M. Reineri, "Detecting injection attacks on cooperative adaptive cruise control," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2019, pp. 1–8.

[19] M. Jagielski, N. Jones, C.-W. Lin, C. Nita-Rotaru, and S. Shiraishi, "Threat detection for collaborative adaptive cruise control in connected cars," in *Proc. 11th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, Jun. 2018, pp. 184–189.

[20] P. Kavathekar and Y. Chen, "Vehicle platooning: A brief survey and categorization," in *Proc. ASME/IEEE Int. Conf. Mech. Embedded Syst. Appl., A B*, vol. 3, Jan. 2011, pp. 829–845.

[21] Y. Kim, I. Kim, and C. Y. Shim, "A taxonomy for DOS attacks in VANET," in *Proc. 14th Int. Symp. Commun. Inf. Technol. (ISCIT)*, Sep. 2014, pp. 26–27.

[22] K. Koscher *et al.*, "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy*, Aug. 2010, pp. 1–16.

[23] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein, "The highD dataset: A drone dataset of naturalistic vehicle trajectories on German highways for validation of highly automated driving systems," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 2118–2125.

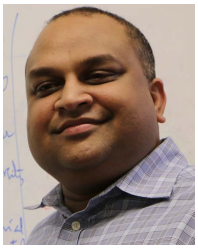
[24] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected cars cyber security," 2017, *arXiv:1711.01939*.

[25] J. Liu and J. Liu, "Intelligent and connected vehicles: Current situation, future directions, and challenges," *IEEE Commun. Standards Mag.*, vol. 2, no. 3, pp. 59–65, Sep. 2018.

[26] A. Lopez, A. V. Malawade, M. A. Al Faruque, S. Boddupalli, and S. Ray, "Security of emergent automotive systems: A tutorial introduction and perspectives on practice," *IEEE Des. Test.*, vol. 36, no. 6, pp. 10–38, Dec. 2019.
[27] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, p. 91, Aug. 2015.
[28] M. Muter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 1110–1115.
[29] National Highway Traffic Safety Association. *Road Accidents in USA*. Accessed: Jan. 22, 2022. [Online]. Available: https://www.recalls.gov/nhtsa.html
[30] S. Parkinson, P. Ward, K. Wilson, and J. Miller, "Cyber threats facing autonomous and connected vehicles: Future challenges," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 11, pp. 2898–2915, Nov. 2017.
[31] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," in *Proc. Black Hat Eur.*, vol. 11, 2015, p. 2015.
[32] R. Rajamani, H.-S. Tan, B. K. Law, and W.-B. Zhang, "Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons," *IEEE Trans. Control Syst. Technol.*, vol. 8, no. 4, pp. 695–708, Jul. 2000.
[33] S. Ray, E. Peeters, M. M. Tehranipoor, and S. Bhunia, "System-on-chip platform security assurance: Architecture and validation," *Proc. IEEE*, vol. 106, no. 1, pp. 21–37, Jan. 2018.
[34] M. Raya, P. Papadimitratos, I. Aad, D. Jungels, and J. P. Hubaux, "Eviction of misbehaving and faulty nodes in vehicular networks," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 8, pp. 1557–1568, Oct. 2007.
[35] Realtime-Technologies. *Physical Automotive Simulator*. Accessed: Jan. 22, 2022. [Online]. Available: https://www.faac.com/realtime-technologies/products/rds-1000-single-seat-simulator/
[36] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive CAN bus," in *Proc. World Congr. Ind. Control Syst. Secur. (WCICSS)*, Dec. 2015, pp. 45–49.
[37] A. Tiwari *et al.*, "Safety envelope for security," in *Proc. 3rd Int. Conf. High Confidence Netw. Syst.*, Apr. 2014, pp. 85–94.
[38] R. W. van der Heijden, A. Al-Momani, F. Kargl, and O. M. F. Abu-Sharkh, "Enhanced position verification for VANETs using subjective logic," in *Proc. IEEE 84th Veh. Technol. Conf. (VTC-Fall)*, Sep. 2016, pp. 1–7.
[39] E. van Nunen, J. Verhaegh, E. Silvas, E. Semsar-Kazerooni, and N. van de Wouw, "Robust model predictive cooperative adaptive cruise control subject to V2 V impairments," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 1–8.
[40] K. Vatanparvar and M. A. Al Faruque, "Self-secured control with anomaly detection and recovery in automotive cyber-physical systems," in *Proc. Design, Autom. Test Eur. Conf. Exhib. (DATE)*, Mar. 2019, pp. 788–793.
[41] V. Vukadinovic *et al.*, "3GPP C-V2X and IEEE 802.11 P for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Netw.*, vol. 74, pp. 17–29, May 2018.
[42] M. Weber, S. Klug, E. Sax, and B. Zimmer, "Embedded hybrid anomaly detection for automotive can communication," in *Proc. 9th Eur. Congr. Embedded Real Time Softw. Syst. (ERTS)*, Jan. 2018, pp. 1–11.

Srivalli Boddupalli (Graduate Student Member, IEEE) received the B.Tech. degree from the Chaitanya Bharathi Institute of Technology, Hyderabad, India, in 2016, and the M.S. degree from the University of Florida, Gainesville, FL, USA, in 2018, where she is currently pursuing the Ph.D. degree with the Department of Electrical and Computer Engineering. She is currently developing security architectures using machine learning techniques for connected vehicle applications. Her research interests include automotive security and intelligent transportation systems.

Akash Someshwar Rao received the B.E. degree in electronics and communication engineering from Visvesvaraya Technological University, India, and the master's degree from the University of Florida, in Spring 2020. Then, he joined Qualcomm Inc. He worked as a Software Engineer at Robert Bosch, Bengaluru, for two years. His research interests include microprocessor security, automotive safety and security, computer architecture, firmware, and machine learning.



Sandip Ray (Senior Member, IEEE) received the Ph.D. degree from The University of Texas at Austin. He is currently a Professor at the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA, where he holds an Endowed IoT Term Professorship. Before joining the University of Florida, he was a Senior Principal Engineer at NXP Semiconductors, and prior to that, he was a Research Scientist with Intel Strategic CAD Laboratories. He is the author of three books and over 100 publications in international journals and conferences. His current research targets correct, dependable, secure, and trustworthy computing through cooperation of specification, synthesis, architecture, and validation technologies. He has served as a Technical Program Committee Member for over 50 international conferences, a Program Chair for ACL2 2009, FMCAD 2013, and IFIP IoT 2019, a Guest Editor for IEEE DESIGN AND TEST, IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS (TMSCS), and *ACM Transactions on Design Automation of Electronic Systems* (TODAES), and an Associate Editor for *Journal of Hardware and Systems Security* (HaSS) (Springer) and IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS (TMSCS).