RECAP: Protecting Cooperative Adaptive Cruise Control Against Multi-Channel Perception Adversary

Srivalli Boddupalli, *Graduate Student Member, IEEE*, Chung-Wei Lin, *Member, IEEE*, and Sandip Ray, *Senior Member, IEEE*

***Abstract*— Cooperative Adaptive Cruise Control (CACC) is a fundamental connected vehicle application. In CACC, a vehicle coordinates its longitudinal movements to safely and efficiently follow the vehicle in front. The follower vehicle relies on a combination of sensory and communication inputs to identify the position, velocity, and acceleration of the preceding vehicle. Malicious subversion of these inputs can cause catastrophic accidents, string instability, and disruption in the transportation infrastructure. In this paper, we develop a security system, RECAP, to provide real-time resiliency in CACC against adversarial subversion of both sensory and communication inputs. RECAP makes use of a combination of techniques based on kinematics and machine learning to detect anomalous inputs, narrow down the source of subversion, and perform mitigation. We provide extensive simulations to demonstrate the effectiveness of RECAP against a diverse spectrum of attacks under complex, multi-channel adversaries.**

***Index Terms*— Security, sensors, vehicular communication, automated driving, machine learning, anomaly detection.**

I. INTRODUCTION

VEHICULAR systems have seen a rapid transformation in recent years, with an explosive infusion of automated driving features enabled by the integration of a variety of new sensors, actuators, compute elements, communication protocols, and software. Automated driving holds the promise of dramatically improving safety through elimination of human errors, while improving utilization efficiency of road infrastructure and reducing adverse environmental impact. However, this also increases the susceptibility of transportation systems to cyber-attacks. Recent research has shown that it is possible, — even relatively straightforward, — to perform cyber-attacks on virtually any component of a transportation system with potentially catastrophic consequences [10], [22], [25].

A key feature of emergent automated vehicles is the ability to perceive its environment. This is achieved through sensors as well as vehicular communications (V2X). Vehicular communications entail exchange of messages with other vehicles (V2V), various components of transportation infrastructure (V2I), and other devices connected to the Internet (V2IoT). In this paper, we refer to the inputs obtained through sensory and communication channels uniformly as *perception inputs*, and the constituent channels as *perception channels*. Perception channels are crucial to the development of a variety of connected and automated vehicular (CAV) applications such as platooning [37], cooperative route management [12], and many others. However, perception channels also produce a large and highly complex attack surface with possibly catastrophic impact [2], [41]. A crucial feature of cyber-attacks on perception channels is that it is not necessary for an attacker to hack into the hardware or software of the victim component: it is possible to create catastrophic impact simply by providing wrong or misleading sensory or communication inputs. Developing viable resiliency technology against cyber-attacks on perception is crucial for proliferation or even adoption of CAV applications.

In this paper, we consider a simple but fundamental CAV application, Cooperative Adaptive Cruise Control (CACC), and show how to develop resiliency against cyber-attacks on its perception channels. In CACC, a follower vehicle $\mathcal{E}$ (also called the *ego vehicle*) coordinates its longitudinal movements in accordance with the velocity, position, and acceleration of the vehicle $\mathcal{P}$ in front (or the preceding vehicle). In many common CACC implementations, $\mathcal{E}$ obtains the velocity and position of $\mathcal{P}$ through on-board sensors and the acceleration of $\mathcal{P}$ through V2V communications [36], [38]; however, the application itself is oblivious to whether a specific channel uses sensor or communication. Adversarial tampering of perception channels of CACC can disrupt traffic movement and cause catastrophic accidents. Our solution, RECAP (for "Resilient Cooperative Adaptive Cruise Control Against Multi-channel Perception Adversaries") is an in-vehicle real-time anomaly detection system that makes use of kinematics and machine learning (ML) to detect anomalous inputs, narrow

Manuscript received 8 February 2023; revised 7 October 2023, 11 April 2024, and 13 July 2024; accepted 5 August 2024. Date of publication 2 September 2024; date of current version 1 November 2024. This work was supported in part by the National Science Foundation under Grant CNS-1908549, in part by the Ministry of Education (MOE) in Taiwan under Grant NTU-113V2003-2, and in part by the National Science and Technology Council (NSTC) in Taiwan under Grant NSTC-112-2636-E-002-010. The Associate Editor for this article was S. Timotheou. *(Corresponding author: Srivalli Boddupalli.)*

Srivalli Boddupalli was with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611 USA. She is now with Lucid Motors, Newark, CA 94560, USA (e-mail: sv.boddupalli@gmail.com).

Chung-Wei Lin is with the Department of Computer Science and Information Engineering, National Taiwan University, Taipei 10617, Taiwan (e-mail: cwlin@csie.ntu.edu.tw).

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

Digital Object Identifier 10.1109/TITS.2024.3445391

down the source of subversion, and perform mitigation. We demonstrate the effectiveness of RECAP against a diverse spectrum of attacks under complex, multi-channel adversaries.

RECAP is, to our knowledge, the first in-vehicle resiliency architecture developed for CACC to defend against a powerful multi-channel security adversary that can simultaneously corrupt more than one perception channel.[1] Our approach is generally independent of the *mechanism* of attack, and can consequently provide protection against both known and unknown attacks as long as the attack is consistent with the adversary power discussed in Section IV-A. Furthermore, this is achieved in the absence of a fixed source of trusted information used to detect and rectify potential anomalies. Another salient feature of RECAP is its ability to always maintain a a small gap (in the range of 0.55–0.75s in our simulations) even under attack scenarios. Note that in contrast, other related resiliency approaches that fall back to ACC (*i.e.*, adaptive cruise control without cooperation) on detecting attacks and consequently pay the price in terms of a larger time gap (1.2s for the controllers used in our experiments).

Despite significant research on security of CACC (see Section II-C), there has been relatively little work on *real-time resiliency*; most of the related work has been on detecting anomalies or intrusions from analysis of communication or sensor channels after the fact. One previous research that targeted real-time resiliency was by Boddupalli et al. [8] that developed an ML-based strategy for detection and mitigation of cyber-attacks on CACC. They considered attacks that tamper the acceleration input from the preceding vehicle. Many aspects of RECAP are inspired by this work, *e.g.*, the use of ML for detecting unexpected inputs as anomalies, as well as an architecture that includes on-board components for real time detection with trained ML models together with an off-site component for performing computation-intensive training. Nevertheless, RECAP differs from this previous work in a number of aspects particularly because of the expanded attack surface resulting from a significantly more powerful adversary model. The adversary model for RECAP permits subversion of *any* of the perception channels from the preceding vehicle, including possibly multiple channels.[2] This precludes the mitigation approach considered in previous work that made critical use of the assumption that the velocity and position values for the preceding vehicles received through the on-board sensors of the ego vehicle always correspond to the ground truth. In addition to detecting the presence of anomaly in input data as done in previous work, RECAP needs to also accurately *identify* the specific channels containing anomalous data. Developing a resiliency solution under these uncertainties requires a different and novel approach to design and analyze real-time resiliency. Furthermore, the multi-channel adversary considered in this work enables attacks where the source of corruption could arbitrarily change during an on-going attack. While simple model-based solutions may be effective against an adversary with fixed sources of corruption, they cannot provide resiliency under an adversary with dynamically changing anomaly sources. This warrants an ML-based solution that can tackle this challenge by learning the contextual relation among various perception inputs as well as their historical values.

[1]Recently, there has been other related work enabling resiliency in connected vehicle applications in general and CACC in particular. However, the adversaries considered in such research are much weaker. See Section II-C.

[2]We still need to impose some constraints on the adversary to ensure that it is not "all powerful". See the discussions in Section IV.

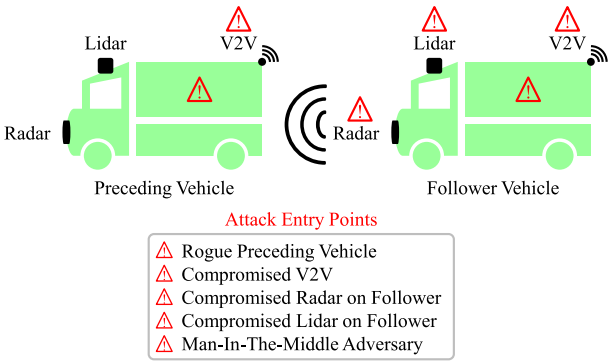


Fig. 1. CAVs engaging in CACC in untrusted environment.

The paper makes the following important contributions. We develop to our knowledge the *first* approach to real-time resiliency for a cooperative connected vehicle application against a multi-channel perception adversary. Multi-channel adversaries with no *a priori* trusted channels brings a number of complex research challenges. We show how to develop a resiliency mechanism that addresses these challenges. Finally. we present a comprehensive evaluation methodology to navigate the attack space. We also provide a roadmap for simulations to validate various design properties of the resiliency system such as recoverability, efficacy against multi-channel corruption and stealthy attacks, etc.

The remainder of the paper is organized as follows. We introduce the basics of CACC application and discuss recent related work in Section II. A brief overview of RECAP's vision is presented in Section III and the unique research challenges associated with real-time resiliency against multi-channel adversary are discussed in Section IV. We dive into the details of RECAP architecture and functionality in Section V. We present a brief summary of our evaluation methodology in Section VI followed by a detailed analysis of our simulation results in Sections VII, VIII, and IX. We conclude in Section X

II. Background and Related Work

A. CACC Basics

CACC is a fundamental vehicle-following application for connected and automated vehicles. It extends the traditional Adaptive Cruise Control (ACC) available in today's vehicles. In CACC, the ego vehicle $\mathcal{E}$ adapts its acceleration $a_{\mathcal{E}}$ to efficiently follow its preceding vehicle $\mathcal{P}$ (Fig. 1) without a human driver's intervention. The goal is to maintain a constant time gap T_{gap}. Most CACC controllers target a constant T_{gap} of approximately 0.55 seconds. To do so, the CACC controller in $\mathcal{E}$ uses three perception channels to detect the state of $\mathcal{P}$ relative to $\mathcal{E}$, *viz.*, $\mathcal{P}$'s velocity $v_{\mathcal{P}}$, $\mathcal{P}$'s (intended)

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 ($4.08s^{-2}$)
τ	Reaction time/controller delay ($0.4s$)
D^{max}	Maximum deceleration constant ($8ms^{-2}$)
G_{min}	Lower bound on space gap ($1.0m$)
T_{gap}	Constant time gap for CACC ($0.55s$ for [4])
g_{safe}	Target safe gap
t_{gap} or TGap	Instantaneous time gap
$a_\mathcal{E}, v_\mathcal{E}, x_\mathcal{E}$	Acceleration, velocity and position of $\mathcal{E}$
$a_\mathcal{P}, v_\mathcal{P}, x_\mathcal{P}$	Acceleration, velocity and position of $\mathcal{P}$ from perception channels
$\hat{a_\mathcal{P}}, \hat{v_\mathcal{P}}, \hat{x_\mathcal{P}}$	Acceleration, velocity and position of $\mathcal{P}$ rectified by RECAP
T_S	CACC stabilization time
T_N	Application engagement time

acceleration $a_\mathcal{P}$, and the gap g between $\mathcal{E}$ and $\mathcal{P}$. CACC controllers operate in two modes depending on the value of g. If $g > g_{\text{safe}}$, it operates in *gap control mode* where it smoothly minimizes the relative velocity with respect to $\mathcal{P}$; if $g \leq g_{\text{safe}}$, it operates in *collision avoidance mode* where $\mathcal{E}$ decelerates at its maximum rated value D_{max}. Here, g_{safe} is called the *safe inter-vehicular distance*, which is a function of the targeted minimum time gap T_{gap}, relative velocities between $\mathcal{E}$, and the rated deceleration value D_{max}.

B. Representative CACC Implementation

RECAP can be used to build resiliency on top of any CACC controller. Nevertheless, for concreteness we will use the representative CACC implementation by Amoozadeh et al. [4]. Equations 1 and 2 represent the controller functionality. Due to the vehicular dynamics and the underlying response time, the desired acceleration computed in Equation 2 cannot be applied immediately. This actuation delay is modeled as a first order time lag. Ultimately, the acceleration applied to the vehicle is computed as represented in Equation 3 accounting for the difference between the desired acceleration and the previous acceleration of the vehicle. In Equation 3, Δt refers to the interval for updating the vehicle's acceleration. It is set to be 0.01s indicating that the underlying controller is operated at a frequency of 100Hz. The current acceleration to be applied to $\mathcal{E}$ and its previous acceleration are represented by $a_\mathcal{E}(t)$ and $a_\mathcal{E}(t-1)$ respectively. Table I shows the glossary of parameters used in this paper and the controller constants for the representative CACC model.

Remark 1: There are several CACC controllers that account for engine dynamics and realistic traffic simulations, e.g., *Zhang et al. [43], Milanés and Shladover [24], Ploeg et al. [32], Xiao et al. [40], Bu et al. [9], etc. Our choice is governed by the goal to showcase the independence of the resiliency system to the specific low-level details of the controller implementation. Note that an explicit objective of* RECAP *is that it can be installed on top of* any *CACC controller. However, demonstrating this flexibility in our experiments requires using a controller with a well-defined, high-level interface that can serve as an abstraction of the underlying CACC system and can be easily integrated into a software-level simulation framework. This enables use of software simulation to efficiently demonstrate resiliency of* RECAP *over the spectrum of attack scenarios considered in this paper. Furthermore, note that the controller we use is widely adopted in related work on CACC security research [5], [6], [16], [31].*

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

$$a_{des} = K_a a_\mathcal{P} + K_v(v_\mathcal{P} - v_\mathcal{E}) + K_g\left(g - v_\mathcal{E}T_{gap} - G_{min}\right) \tag{2}$$

$$a_\mathcal{E}(t) = \frac{a_{des} - a_\mathcal{E}(t-1)}{\tau}\Delta t + a_\mathcal{E}(t-1) \tag{3}$$

C. Related Work

There has been significant research recently on cyber-security of CACC. In this section, we summarize the research in different categories of security solutions for CACC and highlight their limitations. Additionally, several excellent surveys are available that provide a comprehensive treatment of attacks on CAVs engaging in cooperative driving applications [14], [29], [30], [36].

1) Offline Detection Techniques: Biroon et al. [7] propose a diagnostic scheme based on a partial differential equation observer model to detect an adversary injecting ghost vehicles into the platoon. Keijzer and Ferrari [20] propose a sliding-mode observer (SMO) approach for detecting attacks on CACC-based platoon compromising V2V and local sensors simultaneously. Jagielski et al. [16] present a discussion on detection of communication and sensor attacks compromising CACC. Alotibi and Abdelhakim [3] propose a detection technique to capture falsified leader communication in a platoon. Mokari et al. [26] propose a detection and estimation approach against Denial of Service (DoS) attacks on platoons. Sajjad et al. [34] propose an adversarial-aware CACC control scheme that utilizes only local sensor information obviating the need for inter-vehicle communication. *This category of techniques target detection of anomalous samples in historic data after application engagement and therefore, ineffective for real-time resiliency.*

2) Consensus-Based Approaches: Lu et al. [23] present attack-resilient sensor fusion approach for platoons by utilizing the spatial information provided by other participating CAVs thereby achieving more accurate estimation. Yang and Lv [42] propose adding redundant sensor systems to obtain robust estimates of the physical parameters susceptible to security attacks. Petrillo et al. [31] present a secure adaptive CACC controller capable of evicting malicious CAVs in the platoon and mitigating the effects of network-induced perturbations. Kamel et al. [19] propose a simulation framework for misbehavior detection in vehicular platoons through plausibility and consistency checks of the reported messages. Garlichs et al. [13] propose a trust model to identify bad actors and prevent safety hazards in dynamically formed platoon systems.

Obviously, approaches relying on majority voting cannot apply to 2-vehicle CACC. These are also ineffective in the absence of a fixed trusted source of information.

3) Mitigation Techniques: Khanapuri et al. [21] propose a detection and mitigation approach for defending platoon systems against attacks on V2V communication. They utilize local sensor information to mitigate V2V attacks. Wolf et al. [39] propose various general mitigation strategies against data injection attacks and perception channel faults. According to one of the detection-oriented approaches suggested here, the CACC controller gradually degrades to ACC in response to the suspiciousness index of the received inputs. Iorio et al. [15] present a correlation-based anomaly detection technique for capturing injection attacks on CACC. Jin et al. [17] develop an adaptive CACC controller that can defend against sensor and actuator attacks. Sun et al. [35] present a detection and mitigation approach for V2V attacks on CACC based on deep-learning classifiers. Kalogiannis et al. [18], propose a misbehavior detection and mitigation technique for platoons based on Guassian Mixture Model, under a collaborative adversary with an internal and external attack agent acting together to subvert the platoon system. Mousavinejad et al. [27], propose a distributed detection and recovery method based on state prediction and state estimation to identify malicious platoon messages. *Each of these approaches generally targets a specific attack category (*e.g.*, DoS or injection). It is difficult to integrate individual resiliency solutions into vehicular electronics to address each category of attack instances. Furthermore, such solutions cannot provide resiliency against attacks that are discovered after deployment.* Previous work by Boddupalli et al. [8] addresses some of these limitations. They develop an ML-based real-time resiliency architecture for 2-vehicle CACC to capture and mitigate anomalies in the untrusted V2V channel reporting preceding vehicle acceleration values. A unique feature of this work is that it provides assured resiliency against attacks on V2X *irrespective of the attack mechanism*; consequently, the protection also extends to unknown attacks. This work is most closely related to our work in this paper and provides an initial roadmap for developing resiliency architecture for CACC that we extend and consolidate with RECAP. However, the adversary considered in this previous work could only corrupt the acceleration channel; all other channels (*e.g.*, preceding velocity and gap between the ego and preceding vehicles) were assumed to report ground truth. *Their mitigation critically depends on this fact to compute an alternative response in case an anomaly is detected. Consequently, such an approach is precluded under the multi-channel adversary considered for* RECAP *where no channel can be assumed trusted.*

Remark 2: CACC controllers have been developed to gracefully handle the inherent imperfections in communication and senor systems [1], [11], [28]. These controllers are designed to tolerate a specific model of random noise and/or packet loss. However, intentional corruption of V2X and sensor channels through malicious interference cannot be captured under these models. As a result, noise-tolerant controllers cannot automatically be considered resilient against security adversaries.

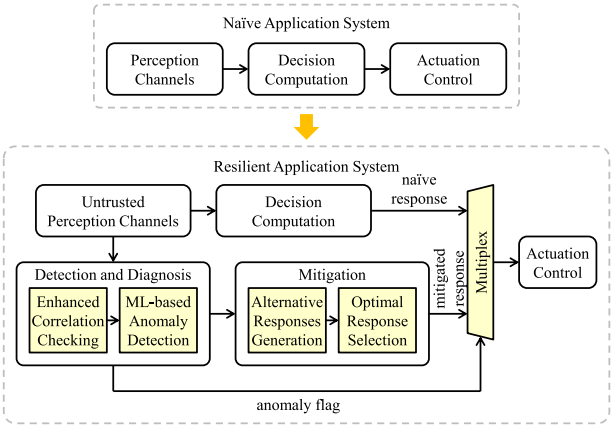


Fig. 2. RECAP-Augmented CACC System.

III. RECAP OVERVIEW

RECAP performs real-time detection and mitigation of perception attacks on a vehicle $\mathcal{E}$ engaging in CACC in an untrusted driving environment. The in-vehicle architecture of RECAP is shown in Fig. 2. It comprises two components: a *Detection and Diagnosis* subsystem and and a *Mitigation* subsystem. The Detection and Diagnosis subsystem monitors the instantaneous state of $\mathcal{P}$, *i.e.*, *position, velocity, and acceleration,* received through untrusted perception channels. If one or more parameter values are determined to be anomalous, they are corrected and an alternate optimal decision is computed by Mitigation system; in the absence of an anomaly, RECAP allows the decision computed by the naive CACC controller to take effect. RECAP adopts a combination of ML techniques and kinematics correlation checking for anomaly detection, precise source identification, as well as mitigation. An important feature of RECAP is its ability to reliably rectify its past errors achieving robustness against sophisticated and stealthy attacks. Furthermore, RECAP is designed to remain agnostic to the underlying perception technology. Therefore, it can provide resiliency to vehicles using either sensors, V2V, or a combination of both for perception.

ML-based anomaly detection involves two steps: (i) offline training of ML regression model to learn the normal behavior of a vehicle engaging in CACC as a function of perception parameters, and (ii) using this trained ML model in real-time as a reference generator for detecting anomalies in untrusted perception inputs. The idea is to train the ML model to learn the normal behavior of a vehicle under benign circumstances; in field, deviation of the behavior beyond a pre-defined threshold would be treated as an anomaly, triggering an alternate, mitigatory response. The input features of the ML model are selected carefully to capture the context sufficiently to enable effective anomaly detection. It is configured with optimal detection thresholds to achieve accurate anomaly detection of untrusted perception inputs and identification of anomaly sources. Appropriate ML architecture that enables effective anomaly detection with minimal computation and power foot-print is identified through a careful selection process.[3] We explain the ML model parameters and inputs in Section V.

[3]Boddupalli et al. [8] provide a detailed discussion of architecture selection and threshold tuning. RECAP follows a similar approach.

Trained and configured RECAP resiliency components are downloaded to the subscribing CAVs through secure connection before commute. During CACC engagement the ML model acts as a reference generator for estimating the normal response. The reference generator output at each step is compared to the response of the naive CACC application controller in real-time. Any deviation between the two beyond a pre-defined threshold is captured as an anomaly and mitigated. RECAP provides real-time resiliency without requiring persistent connection to the cloud. Note that the training and configuration of RECAP does not require any prior knowledge of attack data.

IV. Research Challenges With Multi-Channel Adversary and RECAP Approach

Developing real-time resiliency against multi-channel perception adversaries poses several unique research challenges. These are primarily attributed to the complexity of the powerful adversary and the absence of a fixed source of trusted information, as described below.

A. Adversary Model

A crucial aspect of resiliency design is defining the adversary capabilities. An adversary should be (i) *practical*, meaning it should be possible to realize the adversarial action (and constraints upon it) in practice, (ii) *non-trivial*, meaning the adversary capabilities are not too weak that there is a trivial solution to mitigate any subversion, (iii) *impactful*, meaning attacks if not mitigated have serious consequences to the application being subverted, and and (iv) *viable*, meaning the adversary is not "all-powerful" such that it becomes impossible in principle to defend against a subversion. Under a complex multi-channel adversary, it is quite challenging to construct a threat model balancing the conflicting goals of practicality, triviality, impact, and viability. For instance, an adversary that can collusively corrupt[4] all the perception channels of the victim vehicle does not satisfy the viability condition. On the other hand, an adversary that is constrained to only corrupt a single pre-defined channel (*e.g.*, position of the preceding vehicle) does not satisfy non-triviality since it is possible to mitigate it by simply ignoring information from the untrusted channel and reconstructing it from the values reported by other channels using kinematics.

RECAP is designed to address a general class of adversaries satisfying the following constraints.

1) Any attack instance is assumed to corrupt up to two of the three perception channels from the preceding vehicle, *i.e.*, at most two of the following: preceding vehicle position, velocity, or acceleration. Note that a consequence of this constraint is that one channel is corruption-free throughout an attack instance. However, the channel(s) being targeted in an attack are not fixed *a priori* and can be different for different attack instances.
2) Attacks are assumed to target stable CACC engagement but not during the initial stabilization period at the start of the engagement, *i.e.*, if a CACC engagement takes a time period T_S to stabilize, there is assumed to be no attack during that time. Furthermore, two different attack instances are assumed to be non-overlapping and distant in time from each other, *i.e.*, there is assumed to be a time interval of at least T_m between the end of one attack instance and the start of a subsequent one (if any).
3) An attack instance is assumed to be upper-bounded by a time interval T_C, *i.e.*, if an attack is initiated at time t and causes corruption in channel $\mathcal{L}$ then the attack must be over and $\mathcal{L}$ must start reporting ground truth on or before time $(t + T_C)$.

We now explain how these constraints satisfy the requirements of practicality, non-triviality, impact, and viability. For viability, note that if the adversary were not limited to corrupt at most two out of three channels then it is possible to create attacks that are not viable, *e.g.*, collusive attacks on all three channels as discussed above. Without the restriction of non-overlapping attack instances, it is possible to craft scenarios with two overlapping attacks that can collectively cause collusive corruption of all three channels. Without Constraint 3, an attacker can introduce an infinitesimally small anomaly to a victim channel for an indefinite amount of time: this would enable the attack to remain undetected for a sustained initial period and ultimately disrupting any resiliency mechanism entirely. Note that these restrictions are only intended to eliminate non-viable corner-case scenarios, and do not unreasonably restrict natural adversarial actions. For instance, each perception channel in practice involves different sensory or communication inputs. Therefore, corrupting each channel would involve a different attack technology (*e.g.*, jamming the sensor, introducing fabricated messages into the V2V communication, etc.). It is reasonable to assume that the same adversary cannot simultaneously compromise all three channels within the same attack instance.

Remark 3: Note that the framework does permit collusive attacks, but only precludes attacks that enable collusive corruption of all three preceding vehicle channels at the same time, such that the laws of kinematics are maintained. It is easy to show that no *framework can provide resiliency against an adversary that can collusively corrupt all three channels, as follows. Suppose the ground truth is represented by the velocity, position, and acceleration values* $\langle v, a, x\rangle$ *but the adversary corrupts these values to report* $\langle v', a', x'\rangle$*. Suppose a resiliency mechanism* $\mathcal{R}$ *can identify this triple as malicious. Observe that since the kinematics laws hold within the tuple* $\langle v', a', x'\rangle$*, the tuple can also arise under a benign situation. This implies that* $\mathcal{R}$ *would then classify such benign tuples as malicious. A consequence of the restriction is that* RECAP *generally applies to adversaries that corrupt the perception data input to a (follower) vehicle, not attackers that hack into the vehicle to modify the messages going internally through its bus.*

Remark 4: At cursory read, the upper-bound on duration of each attack instance may appear restrictive. However,

[4]We say that two channels are collusively corrupted if they report values different from ground truth but in such a way that all laws of kinematics relating the values are obeyed.

no attack in practice continues indefinitely and RECAP *incorporates this insight by providing the upper-bound on any continuous attack. One view of T_C is as a programmable design parameter. The security designer has the choice of selecting a value of T_C targeted for a specific domain, and tuning the resiliency system to work for that value. Furthermore, changing the value of T_C (perhaps in response to data collected in field) would simply entail retraining and re-tuning the ML model for the new value.*

B. Anomaly Source Identification

A key challenge with a multi-channel adversary as considered in this paper is that there is no channel that is known *a priori* to be trusted. An attacker can dynamically switch the channel targeted for corruption during the course of the attack (as permitted by the adversary model). In the absence of a known trusted channel, it is challenging to determine which channel is actually corrupted (and hence find an appropriate mitigation for corruption) even when an anomaly is detected. In particular, simple correlation-checking approaches cannot be directly used to detect a source of anomaly. For instance, suppose the adversary collusively corrupts the velocity $v_{\mathcal{P}}$ and acceleration $a_{\mathcal{P}}$ of the preceding vehicle $\mathcal{P}$ in such a way that the kinematic equations connecting these variables are satisfied; an anomaly detection technique would then find that the values of $v_{\mathcal{P}}$ and $a_{\mathcal{P}}$ are mutually consistent but inconsistent with the reported value of the gap g. However, there is no way to determine if the corruption is in $v_{\mathcal{P}}$ and $a_{\mathcal{P}}$ (collusively) or in g. Simple model-based techniques cannot resolve this ambiguity and can be clearly misled by collusive attacks. We provide empirical evidence of how a simple correlation checking system (or a kinematics model) fails under representative collusive attacks in Section IX. RECAP addresses this challenge by adopting a context-aware ML-based detection approach, explained in Section V. This enables it to detect intelligently crafted anomalies in the perception channels that appear normal when each channel is analyzed independently but are indeed corruptions added by an attacker. Careful selection of ML input parameters and effective offline training allow for learning the context sufficiently, achieving highly accurate detection and anomaly source identification under a multi-channel adversary.

C. Safe and Efficient Mitigation

Real-time actions of CAVs not only affect their own safety but can have a direct impact on the safety and efficiency of the neighboring traffic and the transportation infrastructure. Under multi-channel adversaries, it is non-trivial to simultaneously guarantee safety and efficiency during an attack. Furthermore, since anomalies in different perception channels manifest differently, a single strategy of mitigation is generally not effective against corruption in different channels. RECAP's mitigation in response to a detected anomaly consequently involves computing several alternate decisions and vetting each of them for safety first. Out of the safe alternatives, RECAP picks the decision that results in smoothest transition from previous state. This guarantees RECAP's response is always close to the ideal behavior in terms of safety and efficiency. Additionally, this also ensures that the consequences of erroneous judgment are not too drastic and can be safely mended.

D. Recoverability

Ideally, a resiliency system should compute the same decisions under attack as a naive application controller would under benign conditions. Our goal in building a resiliency is to incur only a tolerable deviation from this ideal behavior. This requires the resiliency system to be capable of distinguishing between normal and anomalous inputs it receives with a high degree of accuracy. However, even when it makes mistakes, the resultant behavior must not deviate so much that subsequently normal behavior begins to appear anomalous and vice versa. Recoverability is the property that ensures that the resiliency system can always remain within tolerable accuracy of ideal behavior *even after it commits an error*. Note that inherent imperfections of probabilistic decision-making components in RECAP may result in false alarms or missed anomalies, leading to non-recoverability of the overall system. Even when an anomaly is detected correctly, improper source identification also impacts recoverability since it leads to erroneous rectification of inputs or sub-optimal alternate responses applied to the vehicle. Consequently, RECAP is equipped with a number of checks to identify and neutralize errors in detection or correction in the previous time step, ensuring recoverability of the system. We provide more insights on recoverability and RECAP's approach to address it in Section V

V. RECAP ARCHITECTURE

In this section, we describe the various in-vehicle components of RECAP and discuss their functionality in detail. The operation of RECAP-augmented follower vehicle $\mathcal{E}$ engaging in CACC is shown in Algorithm 1. We make the following assumptions about the underlying CACC application. These assumptions are validated over the simulation data used for the implementation and evaluation analysis of RECAP.

Assumption 1: RECAP *assumes that the follower vehicle $\mathcal{E}$ receives information from all its perception channels and computes its decisions at discrete instances. The interval Δt between two such sampling instants (also referred to as* sampling interval*) is assumed to be small enough that the acceleration $a_{\mathcal{P}}$ of the preceding vehicle can be treated as constant during the interval. Consequently, the mutual relationship between the $a_{\mathcal{P}}$, $v_{\mathcal{P}}$, and $x_{\mathcal{P}}$ can be represented using simplified kinematics represented in Equations 4 and 5*

$$v_{\mathcal{P}}(t+1) = v_{\mathcal{P}}(t) + a_{\mathcal{P}}(t)\Delta t; \tag{4}$$

$$x_{\mathcal{P}}(t+1) = x_{\mathcal{P}}(t) + \left[v_{\mathcal{P}}(t)\Delta t + \frac{a_{\mathcal{P}}(t)}{2}\Delta t^2 \right]. \tag{5}$$

Assumption 2: We assume that the magnitude of change in acceleration Δa between any two consecutive time instances t and $t+1$ to be approximately same under stable CACC in benign conditions. This is generally true of CACC engagements in practice, to ensure the movement of the vehicles under stable CACC engagement to be smooth, (or minimally jerky).

Algorithm 1 ReCAP Functionality

1: **while** $\mathcal{E}$ engaged in stable CACC **do**
2: $a_{\mathcal{P}}(t), v_{\mathcal{P}}(t), x_{\mathcal{P}}(t) \leftarrow ReadPerception()$
3: $a_{\mathcal{E}}(t)^{naive} \leftarrow AcclCmp(a_{\mathcal{P}}(t), v_{\mathcal{P}}(t), x_{\mathcal{P}}(t), v_{\mathcal{E}}(t), x_{\mathcal{E}}(t))$
4: ********* **Detection and Diagnosis** *********
5: $anmFlgs \leftarrow EnhCorrChk(v_{\mathcal{P}}(t), x_{\mathcal{P}}(t), a_{\mathcal{P}}(t-1))$
6: $anmFlg^{v}, anmFlg^{x}, anmFlg^{aPrev} \leftarrow anmFlgs$
7: **if** $anmFlg^{v}$ True **then**
8: $\hat{v_{\mathcal{P}}}(t) \leftarrow CorrectVel()$
9: **if** $anmFlg^{x}$ True **then**
10: $\hat{x_{\mathcal{P}}}(t) \leftarrow CorrectPos()$
11: **if** $anmFlg^{aPrev}$ True **then**
12: $a_{\mathcal{P}}(\hat{t-1}) \leftarrow CorrectPrevAcc()$
13: $anmFlg^{a} \leftarrow MLAnmDet(a_{\mathcal{P}}(\hat{t-1}), \hat{v_{\mathcal{P}}}(t), \hat{x_{\mathcal{P}}}(t))$
14: **if** $anmFlg^{a}$ True **then**
15: $\hat{a_{\mathcal{P}}}(t)) \leftarrow CorrectAcc()$
16: ********* **Mitigation** *********
17: **if** $all\ anmFlgs$ False **then**
18: $a_{\mathcal{E}}(t) \leftarrow a_{\mathcal{E}}(t)^{naive}$
19: $\hat{a_{\mathcal{P}}}(t), \hat{v_{\mathcal{P}}}(t), \hat{x_{\mathcal{P}}}(t) \leftarrow a_{\mathcal{P}}(t), v_{\mathcal{P}}(t), x_{\mathcal{P}}(t)$
20: **else**
21: $a_{\mathcal{E}}(t) \leftarrow Mitigation(\hat{a_{\mathcal{P}}}(t), \hat{v_{\mathcal{P}}}(t), \hat{x_{\mathcal{P}}}(t))$
22: $throttle, braking \leftarrow ActuationControl(a_{\mathcal{E}}(t))$
23: $AppendCorruptHstry(anmFlgs)$
24: $DataCollection(\hat{a_{\mathcal{P}}}(t), \hat{v_{\mathcal{P}}}(t), \hat{x_{\mathcal{P}}}(t), a_{\mathcal{E}}(t), anmFlgs)$

Algorithm 2 Enhanced Correlation Checking

1: $a_{\mathcal{P}}(t-1)^{v} \leftarrow ComputeKinEst(v_{\mathcal{P}}(t), v_{\mathcal{P}}(t-1))$
2: $a_{\mathcal{P}}(t-1)^{x}_{est} \leftarrow ComputeKinEst(x_{\mathcal{P}}(t), x_{\mathcal{P}}(t-1))$
3: $corrFlg^{va} \leftarrow CompareEsts(a_{\mathcal{P}}(t-1)^{v}_{est}, a_{\mathcal{P}}(t-1))$
4: $corrFlg^{xa} \leftarrow CompareEsts(a_{\mathcal{P}}(t-1)^{x}_{est}, a_{\mathcal{P}}(t-1))$
5: $corrFlg^{vx} \leftarrow CompareEsts(a_{\mathcal{P}}(t-1)^{v}_{est}, a_{\mathcal{P}}(t-1)^{x}_{est})$
6: **if** all $corrFlgs$ True **then**
7: $a_{\mathcal{P}}(t-1), v_{\mathcal{P}}(t), x_{\mathcal{P}}(t)$ ***normal***
8: $anmFlgs \leftarrow$ False
9: **else**
10: $corruptHstry \leftarrow ReadCorruptHstry()$
11: **if** $corruptHstry$ *shows 2 untrusted channels* **then**
12: ***(Trusted channel identified)***
13: $anmFlgs \leftarrow ResolveAmbiguity(trstChnl)$
14: **else**
15: ***(Checking Data Property)***
16: $\Delta a \leftarrow |a_{\mathcal{P}}(t-2) - a_{\mathcal{P}}(t-3)|$
17: $\Delta a^{v} \leftarrow |a_{\mathcal{P}}(t-1)^{v}_{est} - a_{\mathcal{P}}(t-2)|$
18: $\Delta a^{x} \leftarrow |a_{\mathcal{P}}(t-1)^{x}_{est} - a_{\mathcal{P}}(t-2)|$
19: $\Delta a^{a} \leftarrow |a_{\mathcal{P}}(t-1) - a_{\mathcal{P}}(t-2)|$
20: $trstChnl \leftarrow MinDif(\Delta a, \{\Delta a^{v}, \Delta a^{x}, \Delta a^{a}\})$
21: $anmFlgs \leftarrow ResolveAmbiguity(trstChnl)$
22: **return** $anmFlgs$

A. Detection and Diagnosis

Detection and Diagnosis is responsible for determining whether one or more perception inputs is anomalous and rectifying the corrupted input(s). RECAP adopts two distinct techniques to determine the presence of anomalies in different perception channels. It performs: (i) enhanced correlation checking for detection and correction of anomalies in velocity and position channels and (ii) ML-based detection and correction of anomalies in acceleration channel.

1) Enhanced Correlation Checking: This step in RECAP resiliency is represented in Algorithm 1, line 5 and expanded further in Algorithm 2. Anomalies in $v_{\mathcal{P}}(t)$ and $x_{\mathcal{P}}(t)$ are detected by checking the correlation between them and the previous acceleration $a_{\mathcal{P}}(t-1)$. However, this correlation check can accurately determine the presence of anomalies in $v_{\mathcal{P}}(t)$ and $x_{\mathcal{P}}(t)$ only if $a_{\mathcal{P}}(t-1)$ is error-free and represents ground truth. Due to the inherent limitations in machine learning systems, anomalies in acceleration channel may sometimes go undetected or inaccurately corrected resulting in an erroneous $a_{\mathcal{P}}(t-1)$. Additionally, a stealthy adversary can carefully corrupt two out of three channels simultaneously in such a way that the correlation between them is preserved. By construction, a simple correlation check will not be able to detect such attacks (discussed further in Section IX). Therefore, RECAP adopts an "*Enhanced Correlation Checking*" technique equipped to rectify past errors and accurately determine the presence of anomalies in velocity and position channels. It is a 3-step approach that involves: (a) correlation checking, (b) corruption history analysis, and (c) normal-data property checking.

a) Correlation checking: As explained before, under normal conditions, $a_{\mathcal{P}}(t-1)$, $v_{\mathcal{P}}(t)$, and $x_{\mathcal{P}}(t)$ obey Equations 4 and 5. We compute acceleration estimates by re-arranging the subject of these equations as shown in Equations 6 and 7. If the computed estimates are close to each other and $a_{\mathcal{P}}(t-1)$, the correlation check passes.

$$a_{\mathcal{P}}(t-1)^{v}_{est} = \frac{\{v_{\mathcal{P}}(t) - v_{\mathcal{P}}(t-1)\}}{\Delta t}; \tag{6}$$

$$a_{\mathcal{P}}(t-1)^{x}_{est} = \frac{2}{\Delta t^{2}}[\{x_{\mathcal{P}}(t) - x_{\mathcal{P}}(t-1)\} - v_{\mathcal{P}}(t-1)\Delta t]. \tag{7}$$

If the deviation between any corresponding pair of estimates is beyond a predefined threshold, the correlation check fails indicating the presence of corruption. This can be caused due to one of the following: (i) the presence of an anomaly in one or both of the current inputs $v_{\mathcal{P}}(t)$ and $x_{\mathcal{P}}(t)$, (ii) undetected anomaly in $a_{\mathcal{P}}(t-1)$ from the previous cycle, or (iii) inaccurate correction of $a_{\mathcal{P}}(t-1)$ in the previous cycle. The following two steps, 1.*b* and 1.*c*, are used to identify the root-cause from among these possibilities.

b) Corruption history analysis: The recorded corruption history is used by the resiliency system to determine which channels were detected to be untrusted in the past. This helps resolve the ambiguity and identify the root cause of

TABLE II
ML-MODEL ARCHITECTURE AND TRAINING HYPERPARAMTERS

Architecture Hyper-parameters	
Hidden Layers	1
Hidden Units	15
Activation	ReLU*
Training Hyper-parameters	
Training Epochs	20
Feature Scaling	Minmax
Learning Algorithm	SGD†

*ReLU: Rectified Linear Unit
†SGD: Stochastic Gradient Descent

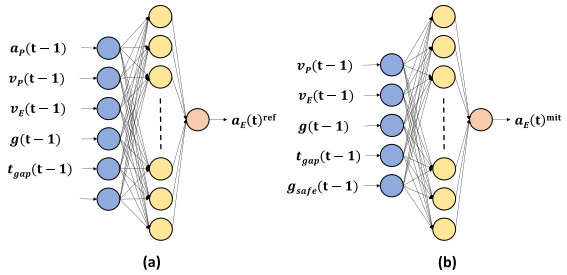


Fig. 3. ML-model architecture: (a) Reference Generator and (b) Mitigator.

inconsistency in a few scenarios. For instance, if the corruption history indicates that anomalies were detected in velocity and position channels in the past, the correlation check failure can be attributed to potential anomalies in one or both of these channels again in the current cycle. Since the adversary model guarantees at least one channel that is corruption-free all through out the application engagement, in this scenario, the anomaly cannot be in $(a_{\mathcal{P}}(t-1))$.

c) Data property checking: If the corruption history fails to irrefutably indicate the presence of past anomalies in two out of three channels, RECAP cannot automatically resolve the ambiguity. In such scenarios, the estimates are checked to see if they obey the data-driven property exhibited in perception channels under normal conditions. While several such data properties can be learned, RECAP uses the property observed in time series data of $a_{\mathcal{P}}$ as stated in Assumption 2. The estimate that closely obeys the property is considered to be trusted and the ambiguity in anomaly source(s) is resolved accordingly.

After discovering the sources of anomaly (if any) through enhanced correlation checking, any previous errors carried forward in $a_{\mathcal{P}}(t-1)$ are rectified. The values of $v_{\mathcal{P}}(t)$, and $x_{\mathcal{P}}(t)$ are corrected using the value of rectified $a_{\mathcal{P}}(t-1)$. This is represented by Algorithm 1, lines 8, 10, and 12.

2) ML-Based Anomaly Detection: Once anomalies in velocity and position are captured and rectified, acceleration channel is scrutinized for possible anomalies. A pre-trained machine learning model referred to as *Reference Generator* computes a normal reference estimate $a_{\mathcal{E}}(t)^{ref}$ which is compared against the acceleration computed by the naive controller $a_{\mathcal{E}}(t)$. The naive controller (discussed in Section II) takes all three untrusted parameters of the preceding vehicle as inputs. Any discrepancy in the decision it computed could be attributed to the anomalies in one or more of these channels. However, since $v_{\mathcal{P}}(t)$ and $x_{\mathcal{P}}(t)$ are corrected during Enhanced Correlation Checking, a deviation from the reference beyond the detection threshold indicates an anomaly in acceleration. Reference Generator takes vetted inputs from the previous time steps to compute the normal reference estimate (see Table II and Fig 3(a)). If an anomaly is detected in $a_{\mathcal{P}}(t)$, it is rectified using the data-driven property of acceleration time-series represented in the "Assumption 2".

B. Mitigation

Mitigation is triggered if anomalies are detected in one or more current perception inputs, or in case of an erroneous $a_{\mathcal{P}}(t-1)$ due to mis-detection/correction in the previous cycle. RECAP mitigation involves the following two steps.

1) Optimal Response Computation: To determine the optimal response for vehicle $\mathcal{E}$, a list of alternatives are computed as follows: (i) by re-invoking the naive CACC acceleration computation module with corrected perception inputs, (ii) by using a trained ML-model that generates an alternate response, and (iii) by invoking the conservative ACC acceleration computation module. A (hypothetical) worst case safety scenario is considered where vehicle $\mathcal{P}$ decelerates at the maximum rated value. Among the alternate responses, acceleration values that are deemed safe in that worst case are shortlisted. Consequently, these alternatives are safe under any possible acceleration of $\mathcal{P}$. In order to preserve the behavior of CACC and to limit error accumulation in case of a potential mis-prediction, mitigating action is taken in small steps. Finally, the alternative that is closest to the previous $a_{\mathcal{E}}(t-1)$ among the safe candidates is determined to be the optimal response and applied to vehicle $\mathcal{E}$.

2) Shadow State Computation: In case of a mis-detection in previous cycle, RECAP tries to neutralize the incorrect action $a_{\mathcal{E}}(t-1)$ taken by vehicle $\mathcal{E}$ in the previous cycle. This is achieved by computing a "shadow response". First, $a_{\mathcal{E}}(t-1)$ decision is recomputed using the corrected value of $a_{\mathcal{P}}(t-1)$. The resultant change in state is computed with the new $a_{\mathcal{E}}(t-1)$. This is considered the shadow state of vehicle $\mathcal{E}$ at time t if it hadn't made an inaccurate decision at time $t-1$. An intermediate decision is computed with this hypothetical state of $\mathcal{E}$ receiving the corrected perception inputs. A resultant next state is computed using the intermediate decision. This is ideally the state vehicle $\mathcal{E}$ reaches in time $t+1$ if it hadn't made any incorrect decisions at time t. Finally, $a_{\mathcal{E}}(t)_{shadow}$ is computed as an action required to reach this ideal state from the current state of vehicle $\mathcal{E}$. The value $a_{\mathcal{E}}(t)_{shadow}$ is added to the list of alternate decisions computed and is applied to vehicle $\mathcal{E}$ if it is determined to be the optimal choice in terms of safety and efficiency.

a) Preserving recoverability: Enhanced Correlation checking is designed to preserve the recoverability of RECAP. The idea is to equip the resiliency system with the ability to rectify potential errors in a cycle in a timely manner without letting them propagate indefinitely into the subsequent cycles of operations. Since the primary source of errors are

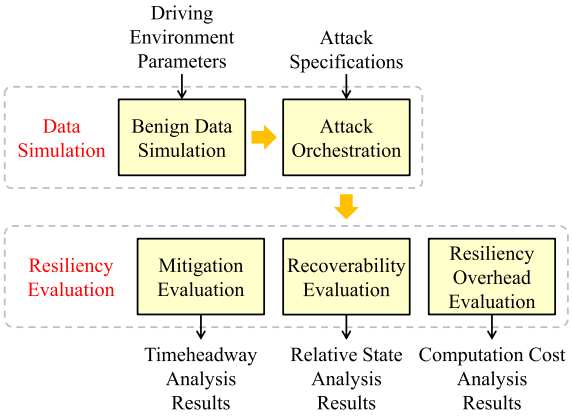


Fig. 4. Data Simulation and Evaluation Flow.

the imperfect ML components in RECAP (used for anomaly detection of acceleration channel and mitigation), Enhanced Correlation Checking technique and shadow state computation involves detecting and correcting past errors (in $a_{\mathcal{P}}(t-1)$) at each time step. This ensures that the imperfect ML-predictions (*i.e.*, false positives or false negatives in anomaly detection, and sub-optimal corrections) are rectified within the next time instance. While it is impossible to *eliminate* ML errors altogether, this effectively *controls* the error propagation thereby preserving the property of recoverability of RECAP resiliency system.

VI. Simulation Results

A. Simulation Setup

Fig. 4 shows the evaluation flow of RECAP. We generate realistic driving data to represent the trajectory of the preceding vehicle using a state-of-the-art physical automotive simulator RDS1000® [33]. This simulator enables flexible configuration of various terrains, weather conditions, and environmental parameters, and includes a repository of pre-configured scenarios developed by engineering experts to reflect real-world environment conditions, lighting, visibility, and road traction attributes. For our simulations, 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 test subjects were selected from students in the university and were instructed to drive the simulated vehicle manually as natural to them for the corresponding driving environment and traffic conditions, following the traffic rules and speed limits. 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. the following vehicle trajectory is computed through a software model of the CACC controller discussed in Section II-A. For performing the simulations presented here, we selected the setting corresponding to: {highway-clear weather-daytime} which is split 60-20-20 into training, validation, and test data. We further discuss the details of simulated data under benign and attack scenarios in Section VII and Table III.

Remark 5: In generating the driving behavior on the simulated vehicle, care is taken to ensure that the behavior is "typical" for each driving condition, i.e.*, neither too aggressive nor too conservative. This characteristic is reflected in the ML model learned by the training process, and deviations flagged as anomalies are computed based on that learning. Like other ML-based systems, it is crucial for effectiveness of* RECAP *in practice that the dataset used for training reflect the driving behavior anticipated to be encountered on road under benign conditions for the different environments targeted for training.*

B. Summary of Experiments

We analyzed the viability of RECAP in the context of the challenges explained in Section IV. We briefly describe the organization of the simulation results here followed by numerical/quantitative evidence for our conclusions in the remainder of this manuscript.

1) ***Attack orchestration and impact visualization:*** Attack impact visualization helps the security architect to gain a holistic perspective on the resiliency problem. In Section VII, we explain the attack orchestration method, a comprehensive attack taxonomy, and also present various examples of representative perception attacks on CACC. We show their impact on the target vehicle in terms of the resultant time-gap between the vehicles.
2) ***Resiliency evaluation:*** In Section VIII-A, we show the efficacy of RECAP resiliency in terms of resultant safety and efficiency achieved under various representative attacks. We show that RECAP ensures recoverability under different attack scenarios in Section VIII-B. We show the computation cost associated with RECAP under various attacks in Section VIII-C.
3) ***Stealthy attack analysis:*** In Section IX, we present analysis on a special class of "stealthy attacks" where two perception channels are simultaneously corrupted in such a way that they remain mutually consistent with each other even under attack. We show that RECAP can effectively mitigate them.

VII. Attack Orchestration and Impact Visualization

We characterize each attack using the following "attack specifications":

1) *Corrupt Channels* specify the perception channels that are corrupted during the course of an attack. According to the adversary assumptions, this can arbitrarily change during the course of an attack such that at most two channels are simultaneously corrupt at a given instance of time and at least one perception channel remains unmodified during the entire course of CACC engagement.
2) *Attack Frequency* specifies the instances of time or the pattern at which corruption occurs. Particularly, we define 3 categories of attacks based on attack frequency: (i) continuous attacks, (ii) cluster attacks,

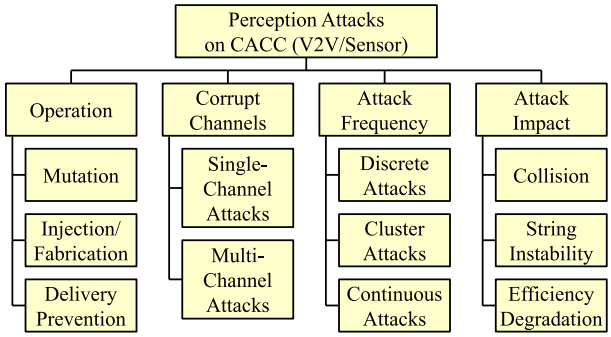


Fig. 5. Taxonomy of Perception Attacks on CACC.

and (iii) discrete attacks. Under a continuous attack, the adversary corrupts the channel(s) for a sustained duration of time. Under a cluster attack, the adversary corrupts the target channel(s) in disjoint intervals of time where the adversary reports corrupted messages for a specific duration followed by reporting the ground truth and so on. Under a discrete attack, the adversary corrupts perception parameters at distinct individual time instances separated by intervals of ground truth.

3) *Corruption Bias* specifies the deviation between the fake perception inputs received and the ground truth. While the magnitude of the bias influences how severe the attack impact is, the sign of the bias influences what kind of adverse impact an attack has on the victim vehicle. Generally, a positive bias leads to the occurrence (or increased risk) of collision while a negative bias leads to efficiency degradation. Consequently, a fluctuating positive-negative bias can cause string instability and in severe cases also cause a collision.
4) *Bias Type* can be (i) constant, (ii) linear, or (iii) sinusoidal. Under a constant bias, attack the deviation between the fake perception readings and the ground truth remains constant at a specified value. Under a linear attack, the bias linearly varies with time according to the specified slope. Under a sinusoidal attack, the bias varies with time according to the sinusoidal amplitude and frequency specified. These biases account for typical corruption types considered in related work [3], [16].

Attack Taxonomy: In order to systematically navigate the attack space under this adversary, we use a taxonomy showed in Fig. 5.[5] This approach of attack categorization obviates the need to specify the *mechanism* and rather characterizes the attacks based on the *manifestation* of an attack. Note that irrespective of mechanism an attack manifests in one of three categories: mutation, fabrication, or delivery prevention.

Our simulation platform includes a flexible attack generation module that takes the attack specifications and simulates attack instances. We orchestrated a total of 72 representative attacks grouped into various categories to cover different combinations of attack specifications described above. Table III

[5]The taxonomy presented here is inspired by by Boddupalli et al. [8]. However, the previous work only accounts for attacks under a single untrusted perception channel (V2V communication). In this paper, we extend that taxonomy to be applicable to multi-channel adversaries.

TABLE III
ATTACK CATEGORY DESCRIPTION

Attack Category	Frequency	Bias Type	Impact
1	Continuous	Constant	Safety Degradation
2	Continuous	Linear	
3	Continuous	Sinusoidal	
4	Cluster	Constant	
5	Cluster	Linear	
6	Cluster	Sinusoidal	
7	Continuous	Constant	Efficiency Degradation
8	Continuous	Linear	
9	Continuous	Sinusoidal	
10	Cluster	Constant	
11	Cluster	Linear	
12	Cluster	Sinusoidal	

Untrusted Parameter	Bias		
	Constant	Linear	Sinusoidal
$a_{\mathcal{P}}^{fake}$ (ms^{-2})	$a_{\mathcal{P}}^{true} \pm 0.2$	$a_{\mathcal{P}}^{true} \pm 0.05t$	$a_{\mathcal{P}}^{true} \pm 0.2sin(0.5t)$
$v_{\mathcal{P}}^{fake}$ (ms^{-1})	$v_{\mathcal{P}}^{true} \pm 2.5$	$v_{\mathcal{P}}^{true} \pm 0.2t$	$v_{\mathcal{P}}^{true} \pm 2.5sin(0.5t)$
$x_{\mathcal{P}}^{fake}$ (m)	$x_{\mathcal{P}}^{true} \pm 5$	$x_{\mathcal{P}}^{true} \pm 0.5t$	$x_{\mathcal{P}}^{true} \pm 5sin(0.5t)$

provides the details of the different attack categories we orchestrated. Each attack category comprises 6 individual attack instances covering all combinations of single (acc, vel, and pos attacks) and 2-corrupt channel attacks (acc-vel, acc-pos, and vel-pos attacks). Attack categories 1–6 include safety degradation attacks while categories 7–12 include efficiency degradation attacks. We carefully selected the magnitude of biases added to each perception channel during attacks to be small but significant such that the fake readings fall well within the range of practical values (constrained by vehicle dynamics) for each perception channel. This is to ensure that we orchestrate non-trivial, hard-to-detect attacks that can compromise safety and efficiency nonetheless. Note that detection of evidently high deviations between the fake and actual values (*e.g.*, $a_{fake} = 50ms^{-2}$ or $v_{fake} = 200ms^{-1}$) can be easily solved with a simple threshold comparison. CACC engagement starts at $t = 0$ and ends at $t = 3000$ for a total duration of 30s (constituting 3000 data samples).

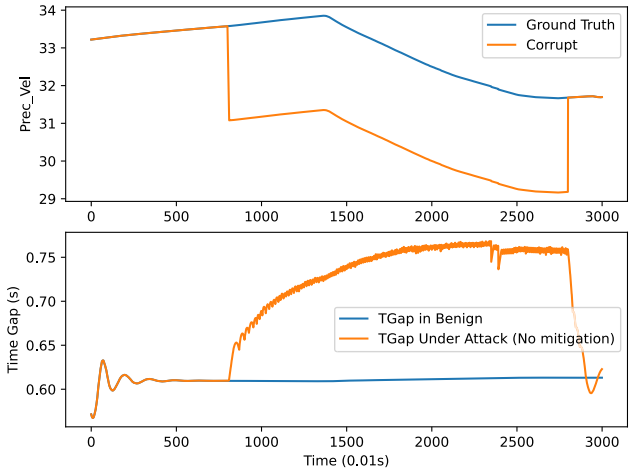
Continuous attacks start at $t = 800$ and end at $t = 2800$ while cluster attacks have 5 arbitrarily spaced attack pulses each with a duration of 200 samples separated by a finite duration of unmodified perception values.[6] We show the impact of a few individual attack instances in Fig. 6. The metric we use to quantify the impact on safety or efficiency is the time gap (TGap) between the two vehicles.

VIII. RESILIENCY EVALUATION

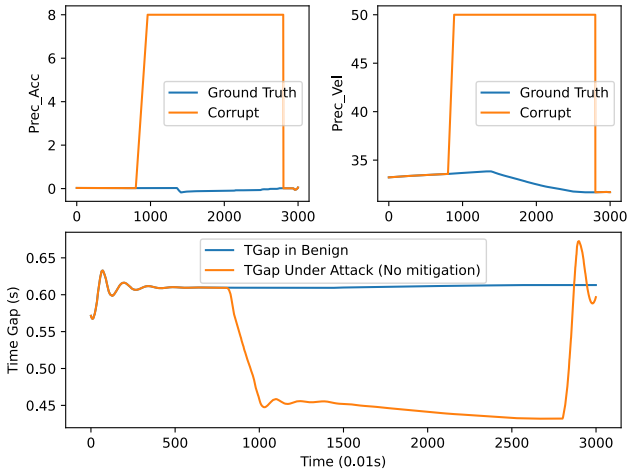
A. Mitigation Efficacy Analysis

RECAP resiliency is evaluated under various representative attacks in terms of the resultant safety and efficiency. We select time gap (TGap) as a metric to indicate safety and efficiency. Under ideal conditions, the CACC system considered in this paper achieves a stable time gap of around 0.55s. However,

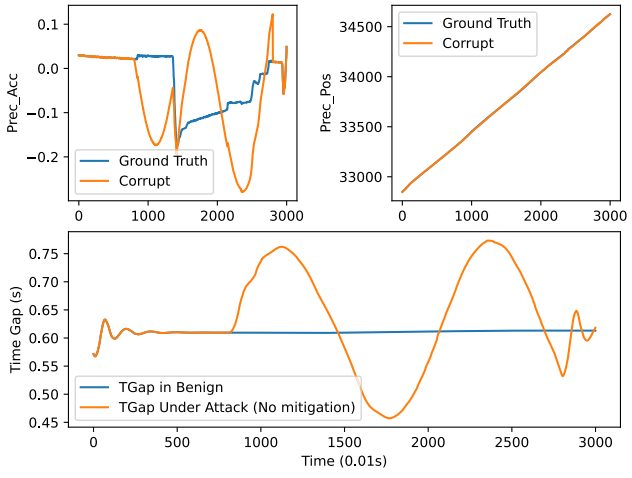
[6]We do not include discrete attacks analysis here in the interest of space and since the impact of discrete attacks is relatively minimal in terms of safety or efficiency. We also exclude delivery prevention attacks from the scope of this work as they have been analyzed by Boddupalli et al. [8] in previous work.



(a)

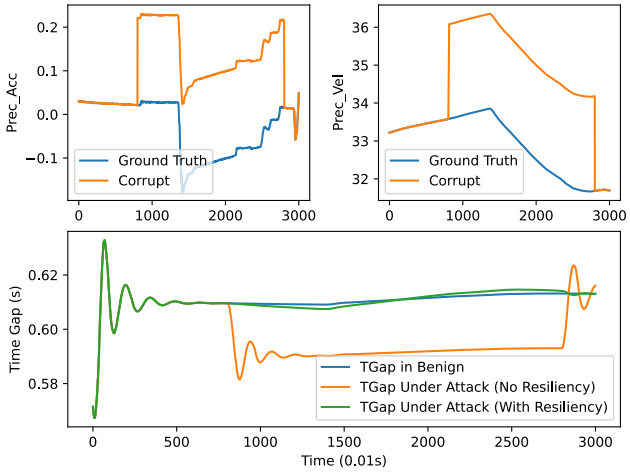


(b)

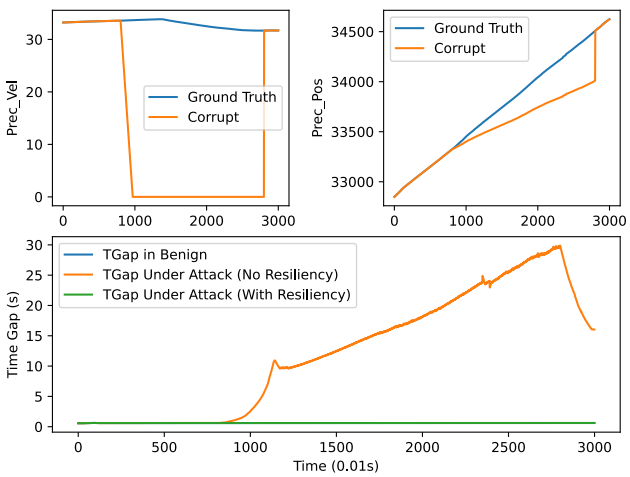


(c)

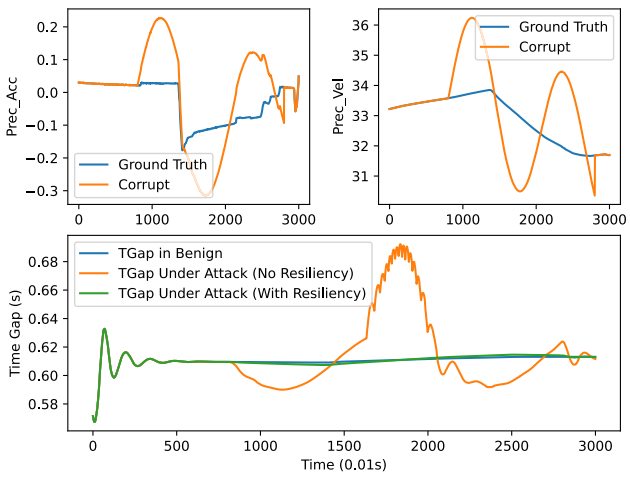
Fig. 6. Attack Impact Visualization: (a) Single-corrupt Channel (Vel) constant bias continuous attack, (b) Two-corrupt channel (Acc-Vel) linear bias cluster attack, and (c) Two-corrupt channel (Acc-Pos) sinusoidal bias continuous attack.



(a)



(b)



(c)

Fig. 7. RECAP Resiliency Visualization: (a) Two-corrupt channel (Acc-Vel) constant bias continuous Attack, and (b) Two-corrupt channel (Vel-Pos) linear bias continuous attack (c) Two-corrupt channel (Acc-Vel) sinusoidal bias continuous attack.

in practice due to the enforced speed limits, initial conditions and vehicle dynamics, the stable time gap may be around 0.6–0.65s under benign conditions. The resiliency goal of RECAP is to maintain a TGap that is well within this range under benign as well as attack conditions.

Fig. 7 shows representative single and multiple corrupt perception input attacks causing safety or efficiency degradation.

TABLE IV
RESILIENCY EVALUATION UNDER DIFFERENT ATTACK CATEGORIES

Attack Category	Safety Degradation Attacks									
	TGap< 0.55s		TGap: {0.55–0.75s}		TGap>0.75s		Min TGap		Max TGap	
	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP
Continuous Constant Bias Attacks	44.96%	0%	55.03%	100%	0%	0%	$0.52s$	$0.60s$	$0.64s$	$0.61s$
Continuous Linear Bias Attacks	79.94%	0%	20.05%	100%	0%	0%	$-0.83s$	$0.60s$	$0.39s$	$0.61s$
Continuous Sinusoidal Bias Attacks	15.64%	0%	76.4%	100%	7.95%	0%	$0.52s$	$0.60s$	$0.72s$	$0.61s$
Cluster Constant Bias Attacks	20.61%	0%	79.38%	100%	0%	0%	$0.49s$	$0.60s$	$0.64s$	$0.61s$
Cluster Linear Bias Attacks	31.62%	0%	67.91%	100%	0.46%	0%	$0.33s$	$0.60s$	$0.69s$	$0.61s$
Cluster Sinusoidal Bias Attacks	7.4%%	0%	90.96%	100%	1.63%	0%	$0.53s$	$0.60s$	$0.67s$	$0.61s$
Attack Category	Efficiency Degradation Attacks									
	TGap< 0.55s		TGap: {0.55–0.75s}		TGap>0.75s		Min TGap		Max TGap	
	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP	Naive CACC	RECAP
Continuous Constant Bias Attacks	0%	0%	30.17%	100%	69.82%	0%	$0.60s$	$0.60s$	$0.76s$	$0.61s$
Continuous Linear Bias Attacks	0%	0%	16.66%	100%	83.33%	0%	$6.62s$	$0.60s$	$16.27s$	$0.61s$
Continuous Sinusoidal Bias Attacks	15.32%	0%	75.64%	100%	9.03%	0%	$0.52s$	$0.60s$	$0.72s$	$0.61s$
Cluster Constant Bias Attacks	0%	0%	72.34%	100%	27.65%	0%	$0.60s$	$0.60s$	$0.86s$	$0.61s$
Cluster Linear Bias Attacks	0%	0%	31.03%	100%	68.96%	0%	$2.60s$	$0.60s$	$6.42s$	$0.61s$
Cluster Sinusoidal Bias Attacks	9.56%	0%	86.13%	100%	4.3%	0%	$0.52s$	$0.60s$	$0.70s$	$0.61s$

We also show the mean TGap distribution in Table IV for each attack category. RECAP-augmented CACC always maintains TGap values within the ideal range (closely following the TGap achieved under benign conditions), while naive controller under attack shows unsafe and inefficient TGap values.

Remark 6: Sudden deviations in the reported acceleration, position, or velocity are trivial anomalies and could be captured by any simple control systems. However, an intelligent adversary such as the one considered in this work can subvert such mechanisms by initially launching a coordinated stealthy attack followed by an impactful attack. We further discuss stealthy attacks in detail with the help of simulation results in Section IX. We show the ineffectiveness of simple correlation checking systems in the face of coordinated attacks that cannot be detected by the controller, gradually corrupting its perception of ground truth. The adversary then proceeds to launch a more impactful attack on the ego vehicle which can no longer identify the discrepancies in the reported values. RECAP *is robust against attacks that are not only impactful but also collusive and stealthy. This enables holistic defense against the entire spectrum of attacks under the adversary model.*

B. Recoverability Analysis

As explained in Section IV, the goal of RECAP is to ensure the behavior of the target CAV under attack is as close to that of a naive CAV under benign conditions. Recoverability of RECAP is, therefore, analyzed from the deviation between the state variable (instantaneous acceleration, velocity and position) progression of RECAP augmented $\mathcal{E}$ under attack and the naive $\mathcal{E}$ under benign conditions over the duration of CACC engagement. For each orchestrated attack, we recorded the state deviation at every individual time instance and computed the mean over the attack duration (3000 instances or 30s). As a result, we tabulated a total 72 different mean error readings each for acceleration, velocity and position. The distribution of mean error in acceleration, velocity and position is shown under each attack category in Figs. 8, 9, and 10.

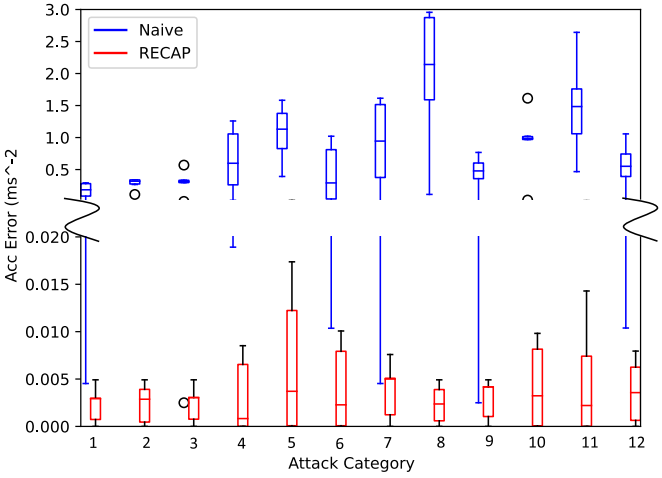


Fig. 8. Mean acceleration error incurred by RECAP and naive controller under different attack categories.

One can observe that the error distribution boxes for naïve controller (in blue) and ReCAP (in red) cannot be displayed over the same scale due to the large differences in the errors incurred for both controllers. For instance, while the naïve controller typically experiences position errors in the range of 10m to 100m under various attack categories, ReCAP experiences less than 0.03m across the entire spectrum. We show this drastic difference in the robustness of the controllers by displaying these error box plots on a split scale where the range in the lower half is in 10^{-2} units while the upper half is in $10–10^2$ units.

In case of the naive CAV, the mean error distribution boxes are centered around 1-1.5ms^{-2} for acceleration, around 2ms^{-1} for velocity, and around 8-10m for position channels. The highest magnitude of errors are reported under efficiency degradation and safety degradation linear attacks (Attack Categories: 8, 11, 2, and 5) where the maximum mean errors reach as high as 3ms^{-2}, 10ms^{-1}, and 200m in acceleration, velocity, and position respectively. On the other hand, the

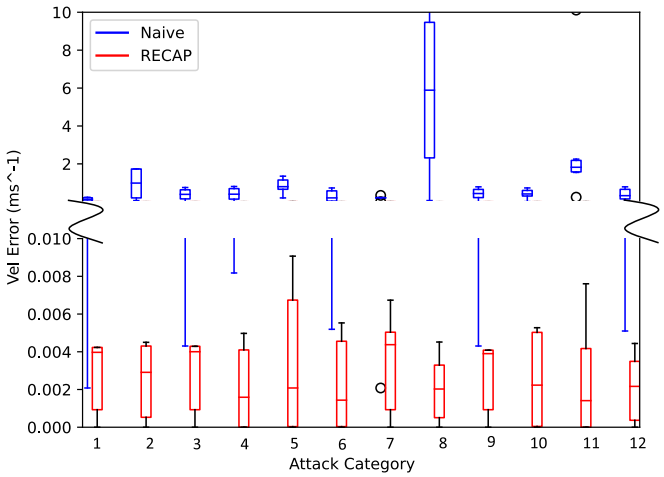


Fig. 9. Mean velocity error incurred by RECAP and naive controller under different attack categories.

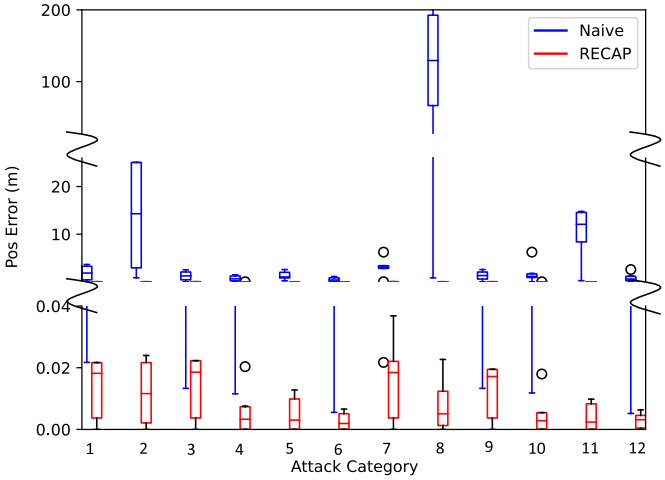


Fig. 10. Mean position error incurred by RECAP and naive controller under different attack categories.

errors incurred by RECAP are several orders smaller in comparison to the naive CAV under every attack category for all three state variables. Distribution of mean error in acceleration for RECAP is around $0.005ms^{-2}$ with a maximum error reaching only as high as $0.0175ms^{-2}$ over the entire attack spectrum. Similarly, the mean error in velocity and position variables is also around $0.005ms^{-1}$ and $0.02m$ respectively. Thus RECAP is robust against the entire adversarial spectrum ensuring recoverability and effective resiliency under all the representative perception attacks.

Remark 7: While the results in Table IV may seem perfect and unrealistic, RECAP *is a viable and practical solution. Note that* RECAP *is not designed to strictly result in a TGap of* 0.55*s at all times. Such an assumption would not be practically realizable. Instead, the goal of* RECAP *is to guarantee a TGap within an acceptable range of values that is still considered safe and efficient in comparison to an ACC system. The results in the table indicate that this design goal is met by* RECAP *under all attack scenarios. This is also indicated by the small but non-zero error distributions of* RECAP *with respect to each perception parameter as shown in Fig. 8, 9, and 10.*

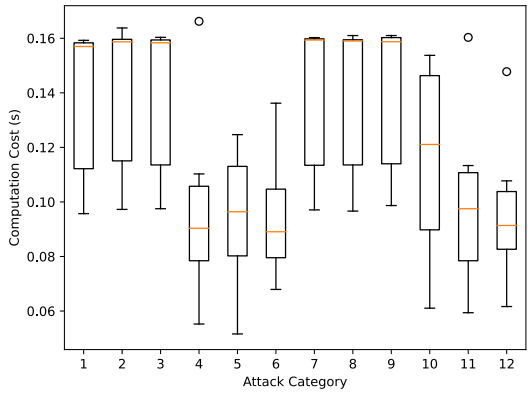


Fig. 11. Distribution of Computation Cost (per time step) under Various Attack Categories.

C. Resiliency Overhead Analysis

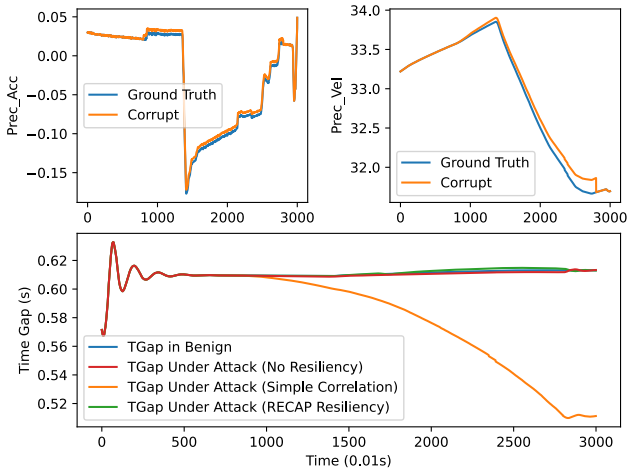
RECAP-augmented CACC system incurs additional computation overhead due to the presence of various on-board resiliency components. Furthermore, the ML-based detection system suffers from a small number of false positives, false negatives, and inaccurate source identification, which lead to additional error correction and mitigation costs. We show the distribution of the computation overhead under each attack category in Fig. 11. The simulations are performed on a computer with 8^{th} Generation i7-8500U processor and a memory of 16GB. These specifications could be considered comparable to the on-board computational resources available in today's automotive ECUs. The cost is well within the decision making interval, making RECAP a viable system to adopt.[7]

IX. Stealthy Attack Analysis

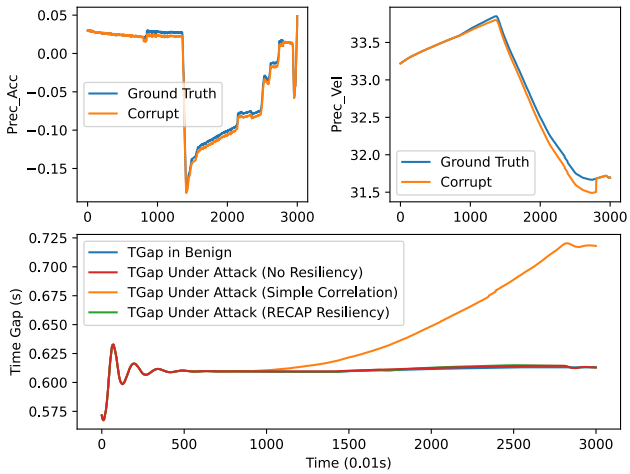
The adversary considered in this work accounts for a special category of stealthy multi-channel corruption attacks, *i.e.*, attacks that are orchestrated to remain undetected or misdiagnosed by the underlying detection system. To achieve a high degree of stealth, the magnitude of corruption is restricted to very small values. As a result, stealthy attacks generally do not show a significant impact on safety or efficiency of the naive victim CAV. However, such attacks can affect the recoverability of in-vehicle resiliency system making it dysfunctional and consequently compromising the safety and efficiency of the vehicle.

Under these attacks, the adversary simultaneously corrupts two channels: a primary perception channel and a secondary channel. A small offset is added to the primary channel and correspondingly, the secondary channel is corrupted such that the kinematics relation between them remains satisfied. Fig. 12(a) shows a collusive attack where the acceleration channel (primary channel) is corrupted by adding a small constant bias of $0.005ms^{-2}$. The velocity channel is corrupted in conjunction to the corrupted acceleration channel following the kinematics relationship represented in Equation 4. Correlation check based on majority vote can be easily bypassed by such

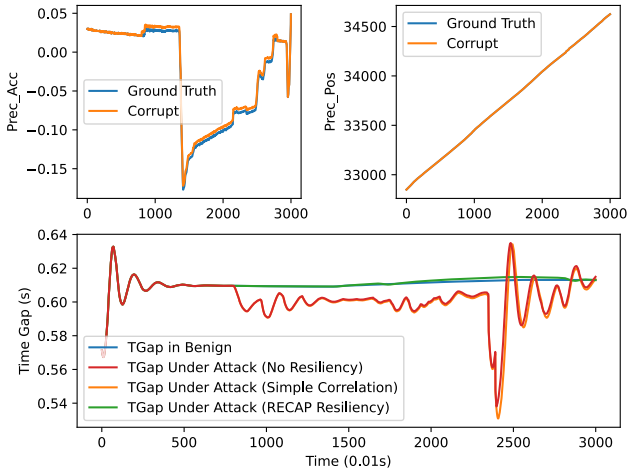
[7]Note that the current implementation of RECAP is a prototype. The computation costs for a deployed architecture will likely be significantly lower.



(a)



(b)



(c)

Fig. 12. RECAP Resiliency under 2-Corrupt channel collusive attacks: (a) Acc-Vel attack ($a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} + 0.005$; $v_{\mathcal{P}}^{fake}$: collusive corruption), (b) Acc-Vel attack ($a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} - 0.005$; $v_{\mathcal{P}}^{fake}$: collusive corruption), and (c) Acc-Pos attack ($a_{\mathcal{P}}^{fake} = a_{\mathcal{P}}^{true} + 0.005$; $x_{\mathcal{P}}^{fake}$: collusive corruption).

an attack since two out of three channels that are systematically corrupt report false information. Such a detection system would incorrectly identify the one unmodified channel to be corrupt and rectify it according to the other two corrupted channels. In this attack example, such a detection would incorrectly identify the position channel as corrupted. Since it no longer can distinguish between normal and anomalous inputs, this will prevent the victim vehicle $\mathcal{E}$ from having an accurate perception of the true state of vehicle $\mathcal{P}$ as the attack proceeds. Ultimately, it can lead to collisions, dangerously low time-gap between vehicles or severe degradation in efficiency.

We orchestrate such stealthy attacks and show that unlike simple correlation checking methods, the robust detection approach adopted by RECAP is capable of detecting and mitigating them. This is achieved with the help of data-driven components incorporated in RECAP detection (enhanced correlation checking and ML-based anomaly detection) as explained in Section V. We compare the performance of RECAP with a kinematics correlation checker based on majority voting. The resultant TGap achieved under different stealthy attack scenarios is plotted in Fig. 12. Under all the attacks, RECAP-augmented CAV successfully mitigated the collusive attacks and preserved its recoverability. The impact of Attacks (a) and (b) shown in the figure is insignificant on the naive CAV. However, under Attack (a), there is a dip in the resultant TGap for the CAV with simple correlation checker falling below the safety limit of $0.55s$. Similarly under Attack (b), the resultant TGap indicates efficiency degradation for the simple correlation checker. Under Attack (c), both naive and simple correlation checker result in negative deviation in TGap. However, RECAP consistently mitigates all the attacks successfully and ensures a high degree of safety and efficiency simultaneously.

X. Conclusion and Future Work

We have presented what we believe is the first comprehensive real-time resiliency framework for CACC against multi-channel adversaries with no *a priori* trusted input. Our framework RECAP uses machine learning to predict the ego vehicle's responses for capturing anomalies in real time, identify the source of the anomaly, and perform mitigation. We discussed the challenges in designing resiliency against multi-channel adversaries and our approaches to resolve these challenges. We also developed a comprehensive methodology for resiliency evaluation in connected vehicle applications and showed the viability and effectiveness of RECAP.

In future work, we will explore applications and extensions of RECAP for other cooperative connected vehicle applications. A key application is multi-vehicle platooning where the ego vehicle receives inputs from other vehicles based on an information flow topology in addition to the preceding vehicle.

References

[1] Z. A. Biron, S. Dey, and P. Pisu, "Sensor fault diagnosis of connected vehicles under imperfect communication network," in *Proc. Dyn. Syst. Control Conf.*, vol. 50695, 2016, Art. no. V001T16A003.

[2] M. S. Al-Kahtani, "Survey on security attacks in vehicular ad hoc networks (VANETs)," in *Proc. 6th Int. Conf. Signal Process. Commun. Syst.*, Dec. 2012, pp. 1–9.

[3] 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.

[4] 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.

[5] M. Amoozadeh et al., "Security vulnerabilities of connected vehicle streams and their impact on cooperative driving," *IEEE Commun. Mag.*, vol. 53, no. 6, pp. 126–132, Jun. 2015.

[6] R. Arvin, A. J. Khattak, M. Kamrani, and J. Rio-Torres, "Safety evaluation of connected and automated vehicles in mixed traffic with conventional vehicles at intersections," *J. Intell. Transp. Syst.*, vol. 25, no. 2, pp. 170–187, Mar. 2021.

[7] R. A. Biroon, Z. A. Biron, and P. Pisu, "False data injection attack in a platoon of CACC: Real-time detection and isolation with a PDE approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 7, pp. 8692–8703, Jul. 2022.

[8] S. Boddupalli, A. S. Rao, and S. Ray, "Resilient cooperative adaptive cruise control for autonomous vehicles using machine learning," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15655–15672, Sep. 2022.

[9] F. Bu, H.-S. Tan, and J. Huang, "Design and field testing of a cooperative adaptive cruise control system," in *Proc. Amer. Control Conf.*, Jun. 2010, pp. 4616–4621.

[10] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in *Proc. USENIX*, vol. 4, San Francisco, CA, USA, 2011, pp. 1–16.

[11] V. S. Dolk, J. Ploeg, and W. P. M. H. Heemels, "Event-triggered control for string-stable vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 12, pp. 3486–3500, Dec. 2017.

[12] L. Du, L. Han, and X. Li, "Distributed coordinated in-vehicle online routing under mixed strategy congestion game," *Transp. Res. B, Methodol.*, vol. 67, pp. 235–252, 2014.

[13] K. Garlichs, A. Willecke, M. Wegner, and L. C. Wolf, "TriP: Misbehavior detection for dynamic platoons using trust," in *Proc. IEEE Intell. Transp. Syst. Conf. (ITSC)*, Oct. 2019, pp. 455–460.

[14] A. Ghosal et al., "Truck platoon security: State-of-the-art and road ahead," *Comput. Netw.*, vol. 185, Feb. 2021, Art. no. 107658.

[15] 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.

[16] 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.

[17] X. Jin, W. M. Haddad, Z.-P. Jiang, and K. G. Vamvoudakis, "Adaptive control for mitigating sensor and actuator attacks in connected autonomous vehicle platoons," in *Proc. IEEE Conf. Decis. Control (CDC)*, Dec. 2018, pp. 2810–2815.

[18] K. Kalogiannis, M. Khodaei, W. M. N. M. Bayaa, and P. Papadimitratos, "Attack impact and misbehavior detection in vehicular platoons," in *Proc. 15th ACM Conf. Secur. Privacy Wireless Mobile Netw.*, May 2022, pp. 45–59.

[19] J. Kamel, M. R. Ansari, J. Petit, A. Kaiser, I. B. Jemaa, and P. Urien, "Simulation framework for misbehavior detection in vehicular networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6631–6643, Jun. 2020.

[20] T. Keijzer and R. M. G. Ferrari, "Detection of network and sensor cyber-attacks in platoons of cooperative autonomous vehicles: A sliding-mode observer approach," in *Proc. Eur. Control Conf. (ECC)*, Jun. 2021, pp. 515–520.

[21] E. Khanapuri, T. Chintalapati, R. Sharma, and R. Gerdes, "Learning based longitudinal vehicle platooning threat detection, identification and mitigation," *IEEE Trans. Intell. Vehicles*, vol. 8, no. 1, pp. 290–300, Jan. 2023.

[22] K. Koscher et al., "Experimental security analysis of a modern automobile," in *Proc. IEEE Symp. Secur. Privacy*, May 2010, pp. 447–462.

[23] P. Lu, L. Zhang, B. B. Park, and L. Feng, "Attack-resilient sensor fusion for cooperative adaptive cruise control," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Nov. 2018, pp. 3955–3960.

[24] V. Milanés and S. E. Shladover, "Modeling cooperative and autonomous adaptive cruise control dynamic responses using experimental data," *Transp. Res. C, Emerg. Technol.*, vol. 48, pp. 285–300, Nov. 2014.

[25] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, pp. 1–91, Aug. 2015.

[26] H. Mokari, E. Firouzmand, I. Sharifi, and A. Doustmohammadi, "DoS attack detection and resilient control in platoon of smart vehicles," in *Proc. 9th RSI Int. Conf. Robot. Mechatronics (ICRoM)*, Nov. 2021, pp. 144–150.

[27] E. Mousavinejad, F. Yang, Q.-L. Han, X. Ge, and L. Vlacic, "Distributed cyber attacks detection and recovery mechanism for vehicle platooning," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 9, pp. 3821–3834, Sep. 2020.

[28] S. Öncü, J. Ploeg, N. van de Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Network-aware analysis of string stability," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 4, pp. 1527–1537, Aug. 2014.

[29] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[30] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and LiDAR," *Black Hat Eur.*, vol. 11, p. 995, Nov. 2015.

[31] A. Petrillo, A. Pescapé, and S. Santini, "A secure adaptive control for cooperative driving of autonomous connected vehicles in the presence of heterogeneous communication delays and cyberattacks," *IEEE Trans. Cybern.*, vol. 51, no. 3, pp. 1134–1149, Mar. 2021.

[32] J. Ploeg, E. Semsar-Kazerooni, G. Lijster, N. van de Wouw, and H. Nijmeijer, "Graceful degradation of cooperative adaptive cruise control," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 1, pp. 488–497, Feb. 2015.

[33] Realtime-Technologies. *Physical Automotive Simulator*. Accessed: Jan. 11, 2022. [Online]. Available: https://www.faac.com/realtime-technologies/products/rds-1000-single-seat-simulator

[34] I. Sajjad, D. D. Dunn, R. Sharma, and R. Gerdes, "Attack mitigation in adversarial platooning using detection-based sliding mode control," in *Proc. 1st ACM Workshop Cyber-Phys. Syst.-Secur. PrivaCy*, Oct. 2015, pp. 43–53.

[35] G. Sun, T. Alpcan, B. I. P. Rubinstein, and S. Camtepe, "Strategic mitigation against wireless attacks on autonomous platoons," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*. Cham, Switzerland: Springer, 2021, pp. 69–84.

[36] R. van der Heijden, T. Lukaseder, and F. Kargl, "Analyzing attacks on cooperative adaptive cruise control (CACC)," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Nov. 2017, pp. 45–52.

[37] V. Vukadinovic et al., "3GPP C-V2X and IEEE 802.11p for vehicle-to-vehicle communications in highway platooning scenarios," *Ad Hoc Netw.*, vol. 74, pp. 17–29, May 2018.

[38] A. J. Watkins. *Cadillac's [CTS] Sedans Can Now 'Talk' to Each Other, Which May Make Driving Way Less Deadly*. Accessed: Jan. 11, 2022. [Online]. Available: https://www.theverge.com/2017/3/9/14869110/cadillac-cts-sedan-v2v-communication-dsrc-gm

[39] M. Wolf et al., "Securing CACC: Strategies for mitigating data injection attacks," in *Proc. IEEE Veh. Netw. Conf. (VNC)*, Dec. 2020, pp. 1–7.

[40] L. Xiao, M. Wang, and B. van Arem, "Realistic car-following models for microscopic simulation of adaptive and cooperative adaptive cruise control vehicles," *Transp. Res. Rec.*, vol. 2623, no. 1, pp. 1–9, Jan. 2017.

[41] C. Yan, W. Xu, and J. Liu, "Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle," *Def Con*, vol. 24, no. 8, p. 109, 2016.

[42] T. Yang and C. Lv, "A secure sensor fusion framework for connected and automated vehicles under sensor attacks," *IEEE Internet Things J.*, vol. 9, no. 22, pp. 22357–22365, Nov. 2022.

[43] Y. Zhang, Y. Bai, M. Wang, and J. Hu, "Cooperative adaptive cruise control with robustness against communication delay: An approach in the space domain," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5496–5507, Sep. 2021.

Srivalli Boddupalli (Graduate Student Member, IEEE) received the M.S. and Ph.D. degrees from the University of Florida, Gainesville, FL, USA. She is currently a Senior Data Scientist with Lucid Motors, working in the domain of cyber-security of vehicular systems. She is developing security architectures using machine learning techniques for connected vehicle applications. Her research interests include automotive security and intelligent transportation systems.

Chung-Wei Lin (Member, IEEE) received the Ph.D. degree in electrical engineering and computer sciences from the University of California at Berkeley. He was a Researcher with the Toyota InfoTechnology Center, USA, from 2015 to 2018. He is currently an Associate Professor with the Department of Computer Science and Information Engineering, National Taiwan University (NTU). His research interests include cyber-physical systems, connected and autonomous vehicles, security, system design methodology, and model-based design.

Sandip Ray (Senior Member, IEEE) received the Ph.D. degree from The University of Texas at Austin. He is currently a Professor with 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 with NXP Semiconductors and prior to that, he was a Research Scientist with the Intel Strategic CAD Laboratories. His current research interests include correct, dependable, secure, and trustworthy computing through the cooperation of specification, synthesis, architecture, and validation technologies. He is the author of three books and over 100 publications in international journals and conferences. He has served as a technical program committee member for over 50 international conferences, the Program Chair for ACL2 2009, FMCAD 2013, and IFIP IoT 2019, the Guest Editor for IEEE DESIGN & TEST, IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS, and ACM TODAES, and an Associate Editor for Springer HaSS and IEEE TRANSACTIONS ON MULTI-SCALE COMPUTING SYSTEMS.