Resiliency in Connected Vehicle Applications: Challenges and Approaches for Security Validation

Srivalli Boddupalli, Richard Owoputi, Chengwei Duan, Tashfique Choudhury, and Sandip Ray
Department of Electrical and Computer Engineering, University of Florida
Gainesville, Florida, USA
bodsrivalli12@ufl.edu,rowoputi@ufl.edu,duan.c@ufl.edu,choudhury.t@ufl.edu,sandip@ece.ufl.edu

ABSTRACT

With the proliferation of connectivity and smart computing in vehicles, a new attack surface has emerged that targets subversion of vehicular applications by compromising sensors and communication. A unique feature of these attacks is that they no longer require intrusion into the hardware and software components of the victim vehicle; rather, it is possible to subvert the application by providing wrong or misleading information. We consider the problem of making vehicular systems resilient against these threats. A promising approach is to adapt resiliency solutions based on anomaly detection through Machine Learning. We discuss challenges in making such an approach viable. In particular, we consider the problem of validating such resiliency architectures, the factors that make the problem challenging, and our approaches to address the challenges.

CCS CONCEPTS

• **Security and privacy** → **Systems security**; • **Computer systems organization** → *Embedded and cyber-physical systems.*

KEYWORDS

Autonomous vehicles, Anomaly detection, Machine learning, Cooperative driving applications

ACM Reference Format:
Srivalli Boddupalli, Richard Owoputi, Chengwei Duan, Tashfique Choudhury, and Sandip Ray. 2022. Resiliency in Connected Vehicle Applications: Challenges and Approaches for Security Validation. In *Proceedings of the Great Lakes Symposium on VLSI 2022 (GLSVLSI '22), June 6–8, 2022, Irvine, CA, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3526241.3530832

1 INTRODUCTION

Modern automotive systems are essentially complex distributed systems that involve coordination of hundreds of Electronic Control Units (ECUs) communicating through a variety of in-vehicle networks and executing several hundred Megabytes of software code. Automotive systems induce two additional constraints that result in significant design complexities beyond traditional distributed systems. First, the systems are cyber-physical: the ECUs coordinate, monitor, and control a variety of sensors and actuators, including LIDAR, camera, radar, light matrix, devices for sensing angular momentum of the wheels, devices for automated brake and steering control, etc. Second, many computational and communication tasks must be accomplished under hard real-time requirements, *e.g.*, a pedestrian detection algorithm must complete a slew of complex activities including capturing sensory data, aggregation, communication, analytics, image processing, security analysis, etc., within the time constraints to enable successful completion of the appropriate actuation response such as warning or automated braking.

One upshot of increasing autonomy is a corresponding increase in the vulnerability of these systems to cyber-attacks. Given that the system involves complex interaction of sensory, actuation, and computational elements, an "innocent" misconfiguration or error in one component may result in a subtle vulnerability which can be exploited in-field with potentially catastrophic consequences. Recent research has shown that it is possible, — even relatively straightforward, — to compromise a vehicle and get control over its driving function [6, 10, 14]. The trend towards increasing autonomy will only exacerbate this situation: the increasing dependence of critical vehicular operations on complex electronics and software will result in an increased attack surface as well as the increasing ability of an attacker to create catastrophic impact from a compromise. ***Consequently, the proliferation or even adoption of autonomous vehicles critically depends on our ability to ensure that they perform securely, in a potentially adversarial environment.***

A critical feature of emergent autonomous vehicles is *connectivity*, *i.e.*, the ability to communicate with other vehicles (V2V), with the infrastructure (V2I), and with other devices connected to the Internet (V2IoT). Vehicular communications, referred to as V2X, have become fundamental for autonomous driving. They enable cooperative information sharing for streamlining traffic movement, improving road safety, and efficiently utilizing traffic and transportation infrastructure. Unfortunately, V2X is also a highly vulnerable feature that can be exploited by an adversary to disrupt traffic movement, cause catastrophic accident, and bring down the transportation infrastructure. A key problem with V2X is that it obviates the need for an adversary to actually hack a vehicle: sending misleading or even malformed V2X communications is often sufficient to disrupt the connected car ecosystem. Unsurprisingly, in a recent survey by the world's second-largest reinsurer Munich Re, 55% of the surveyed corporate risk managers named security of vehicular communications as their top concern for autonomous vehicles [16]. Perhaps even more alarming, 64% of the companies

GLSVLSI '22, June 6–8, 2022, Irvine, CA, USA

ACM ISBN 978-1-4503-9322-5/22/06…$15.00
https://doi.org/10.1145/3526241.3530832

surveyed mentioned that they were completely unprepared to address this threat.

In this paper, we consider the problem of introducing real-time resiliency in connected autonomous vehicular (CAV) applications. We consider adversaries that can target the "perception" channels of a vehicle: we deliberately leave the mechanism of perception undefined, allowing it to be through either sensory input or V2X communication. An adversarial attack is a perturbation of some (subset of) the perception channels involved in the application, such that the input received would be different from actual. For instance, in Cooperative Adaptive Cruise Control (CACC), a vehicle $\mathcal{E}$ receives the velocity, relative position, and acceleration of its preceding vehicle $\mathcal{P}$; during an attack, the values received by $\mathcal{E}$ would be assumed to be different from ground truth. The focus of *real-time resiliency* is to augment the application functionality so that $\mathcal{E}$ can perform safely and efficiently, even during attack. The focus of the paper is to identify and enumerate challenges in developing such real-time resiliency in CAV applications. In particular, previous work has shown how to use anomaly detection based on Machine Learning (ML) in addressing the problem [4]; however, the use of ML brings in new challenges in design and validation. We briefly summarize our approach in addressing these challenges, taking two illustrative CAV applications: CACC and multi-vehicle platooning.

2 LIMITATIONS OF HARDWARE SECURITY TECHNIQUES AND ROLE OF ML-BASED ANOMALY DETECTION

The last decade saw significant advancement in hardware security research, from design to validation. It is worth considering why these solutions cannot be directly adapted to CAV resiliency. Common design approaches to address communication vulnerabilities between untrusted entities or through untrusted communication channels entail the use of either (1) some authentication or attestation paradigm to preclude masquerade, sybil, or misdirection attacks; or (2) cryptographic techniques to ensure non-observability of communications by malicious man-in-the-middle attackers. Note that these techniques tend to be highly computation-intensive. These techniques may be appropriate for many automotive functionalities that do not involve real-time decisions. However, they are not applicable if they fall in the critical path of real-time actuation and decision making. This, in fact, is the case when defending against an adversary that sends maliciously modified inputs influencing the actuation of the victim vehicle. This is particularly crucial considering the limited computation resources available in a vehicular system.

Boddupalli *et al.* [4, 5] developed an approach called ReDeM that addresses these limitations. The approach is an ML-based anomaly detection, that extends the application controller with specific on-board components to provide resiliency. Fig. 1 shows the ReDeM on-board architecture. The key idea is to include a trained ML model that classifies inputs coming from the untrusted perception channels as either benign or malicious. The detection component determines the presence of anomalies in the perception inputs and

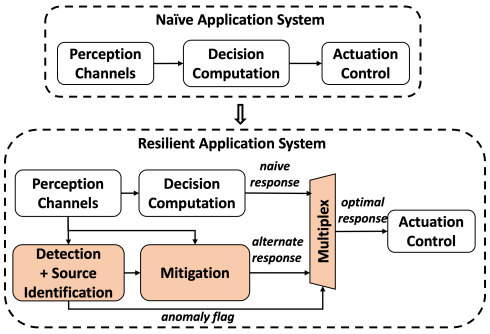

Figure 1: ReDeM-augmented CAV Application System

raises a flag if detected to initiate mitigation. The Mitigation component is responsible for neutralizing the adverse effects of anomalous perception inputs. It also computes an alternative decision that overrides the corrupted response from the naive application controller. The Anomaly Detector of ReDeM is founded upon an ML model trained to learn normal behavior of the Decision Computation Module of the targeted application; a substantial deviation of the actual value from the predicted one is flagged as an anomaly. ReDeM addresses the computational constraints noted above by separating the computationally expensive training of ML from the in-field prediction. In particular, ReDeM includes a cloud-based infrastructure for training prediction models, while the on-board architecture collects data and performs real-time prediction. The trained model is periodically downloaded on-board. Furthermore, real-world data is collected and periodically transferred to a cloud server to retrain the machine learning components.

The approach above has shown promise, in particular addressing the computational resource constraints in automotive on-board electronics vis-a-vis the computational needs for real-time resiliency. It also addresses the so-called *small data problem* that often represents an Achilles heel in adapting ML solutions to security. The small data problem refers to the fact that while normal behavior model is generally plentiful in the real world, the amount of real-world attack data is limited, which makes it difficult to train an ML classifier to discriminate between normal and attack data. By formulating the CAV resiliency in terms of anomaly detection rather than a problem classifying between normal and attack scenarios, ReDeM avoids the thorny problem created by lack of attack data: ReDeM models need to only be trained to learn normal behavior while anomaly is characterized simply as a deviation from the predicted normal.

Nevertheless, the use of ML-based anomaly detection does induce new challenges. A key problem is in developing effective approaches for validation. Validation is obviously crucial for a CAV resiliency system, since it targets highly safety-critical applications. However, the unique nature of ML-based resiliency makes it highly challenging to achieve. In the rest of the paper, we discuss various facets of the validation problem and ReDeM's approach to addressing them. The goal is not to justify the ReDeM solutions. Instead, our focus is to point to the spectrum of validation issues involved in ML-based resiliency in connected vehicle applications: we describe the ReDeM approaches simply to point to directions that we have found viable.

3 THE PROBLEM OF DATA

The efficacy of *any* ML-based prediction system depends upon the availability of high-quality data. This is true of ReDeM, since its key work horse is ML-based anomaly detection. So a critical question to ask is: how do we get copious high-quality data necessary to make the ML-based predictions viable? We addressed this question above in the context of achieving ML-based training *after deployment*: we argued that normal driving behavior data is plentiful in real world and this enables the on-board system to simply collect such data under various conditions when the vehicle is on road. However, this does not address the *validation* issue: how can we have enough real-world data *before* connected vehicle applications have been widely deployed in field, so that we can evaluate and tune the resiliency infrastructure? At this point, we obviously cannot depend on availability and collection of real-world driving data for connected vehicle applications, since widespread deployment of CAV applications has not happened yet. Indeed, we face a "chicken and egg" problem: until we can develop reliable resiliency solutions we cannot expect widespread deployment, and until we have widespread deployment we do not have the accurate real-world data necessary to validate the resiliency solution.

One way to address this problem is to look for sources of driving behavior data. There are two sources, described below, each imperfect.

Datasets. The first source includes some publicly available traffic datasets. Some examples include KDD99 dataset [15], DSRC vehicle communication dataset [8], Audi A2A2 autonomous driving dataset [9], AMUSE [11], the Ford Autonomous driving dataset [1]. A key challenge with these datasets is that they are collected for a variety of different purposes and may not be directly suitable. For example, our work would typically require the driving behavior of individual vehicles under a variety of environments. On the other hand, many of the datasets include behavior of a number of vehicles each collected for a very short duration.

Simulators. The second source is through various automotive simulators. There is a plethora of simulators, ranging from desktop simulators targeting specific functionality to physical driving simulators. Three well-known desktop simulators are SUMO [13], CARLA [7], and VENTOS [2]. SUMO is an open source, microscopic, multi-modal traffic simulator. Each vehicle has its own route, and moves individually through the network. CARLA targets development, training, and validation of autonomous driving systems. In addition to open-source code and protocols, CARLA provides open digital assets (urban layouts, buildings, vehicles). The simulation platform supports flexible specification of sensor suites, environmental conditions, full control of all static and dynamic actors, maps generation, etc. VENTOS is an integrated C++ simulator for studying vehicular traffic flows, collaborative driving, and interactions between vehicles and infrastructure through DSRC-enabled wireless communication capability. It makes use of SUMO for vehicular traffic mobility models together with OMNET++ [17] for wireless communication among the different nodes. In addition to desktop simulators, there are also physical automotive simulators. They are generally custom-built, and enable detailed simulation and analysis of vehicular trajectory data in diverse driving environments. The

Figure 2: RDS-1000 Simulator Used for ReDeM Data Collection

focus of these simulators have traditionally been to study human behavior while driving. They provide a physical interface for a human driver to control the vehicle in simulation. Simulators can be used to capture logs of a variety of synthetic driving data. However, a problem with this data is that there is no *a priori* reason to believe that these logs do indeed capture real-life scenarios. Consequently, training and inference based on such data could be spurious.

ReDeM addresses this by using simulator log from a physical simulator, RDS1000® (https://www.faac.com/realtime-technologies/products/rds-1000-single-seat-simulator) for data collection but using real-world datasets to vet the collected data. Fig. 2 shows the simulator used for data collection. The key observation is that physical simulators provide realistic simulation of driving environment (*e.g.*, terrain, weather, etc.). Consequently, the driving behavior of a human driver operating on the simulator would likely mimic how the driver would operate an actual vehicle in practice. Consequently, data from such operation could be used as proxy for "normal driving behavior" under the given condition. By sweeping through a range of environmental conditions, we can collate a comprehensive dataset of normal behavior. The collated data may be skewed by the driving idiosyncrasies of the human operating the simulator. We address this by matching the collated driving data with data from available datasets. Note 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 [12] that provides trajectory data corresponding to real vehicles driving in German highways. The length of individual vehicle trajectories is approximately 15 seconds. Our experiments showed that the driving patterns from the simulator correlate closely with HighD data, justifying our use of the simulator data for subsequent evaluation.

4 VALIDATING RESILIENCY AGAINST ZERO-DAY ATTACKS

A key requirement for resiliency is that it provides protection against a spectrum of attacks including those not are necessarily known at deployment. Attacks evolve over the lifetime of a vehicle. A solution that protects against a very specific mechanism

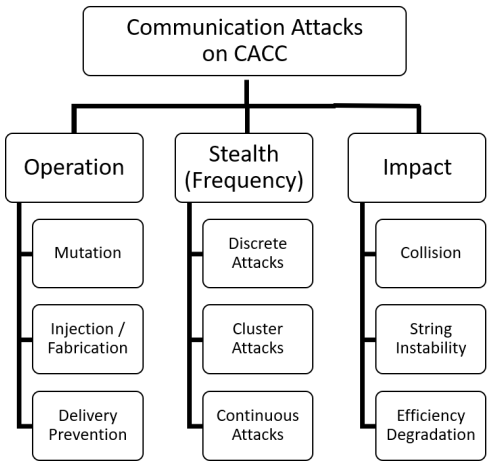

Figure 3: A V2X Attack Taxonomy. The figure is taken from a previous paper [4] co-authored by some of the authors of this paper.

of attack would become ineffective as the attacker finds a new different attack mechanism, *e.g.*, a solution preventing an attacker from performing denial-of-service (DoS) attacks through jamming vehicular communication, would be useless once the attacker finds a different DoS attack that does not require jamming. On the other hand, it is clearly infeasible to determine in advance all possible mechanisms through which an attacker can compromise communication or sensory perception. Indeed, new attacks can become feasible only after technology (and hence sophistication of attack) advances during the in-field life of the vehicle in ways not necessarily anticipated at deployment. This results in a conundrum for security validation: how can we ensure that the resiliency system is indeed effective, not only against known attacks but against a spectrum of attacks that are unknown at deployment time?

ReDeM addresses this problem by noting that it is possible to develop a resiliency system that accounts for attacks based on its manifestation features, stealth, and impact rather than detailed mechanism. Furthermore, it is possible to comprehensively classify the spectrum of attacks in this manner simply from the threat model. For instance, consider a CACC application where a vehicle follows its preceding vehicle by maintaining a specific time headway. If 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). Going through this argument enables us to create a taxonomy of V2X attacks as shown in Fig. 3. Note that if our validation covers attack space defined by the taxonomy then the above argument suggests that we indeed comprehensively cover the space of all attacks defined by the threat model, including unknown attacks.

Note that the taxonomy, while crucial, is only the first step in identifying and defining attacks to be accounted for in resiliancy evaluation. Evaluating a resiliency solution additionally requires comprehending the the *impact* of these attacks. After all, it is not critical that the resiliency solution can effectively detect or mitigate an attack that has no significant in the first place. On the other hand, mitigating an attack with the potential for catastrophic accident is critical even if the attack is rare. Note that the impact depends not only on the magnitude of the bias (deviation from normal) but also the frequency: an attack with a small bias, but performed for a long duration, can cause a significant impact on the victim vehicle. An interesting observation from ReDeM is that the taxonomy can help determine the impactful attacks. ReDeM achieves this by systematically exploring the attack space defined by the taxonomy and identifying subspaces that represent attacks with high impact. As example, Figs. 4, 6, 5, and 7 show the results of impact analysis for continuous mutation attacks, discrete mutation attacks, and random mutation and delivery prevention attacks on CACC. It is clear from the analysis that continuous mutation attacks are significantly more impactful than discrete mutation attacks and random attacks.

5 VALIDATION CHALLENGE FOR PROBABILISTIC PREDICTION SYSTEMS

No ML system is accurate in 100% of cases. This creates a vexing evaluation problem: how to ensure that the vehicle can perform reliably under unpredictability of ML? Note that this cannot be solved by simply showing high value of an accuracy metric (*e.g.*, precision, recall, or f1-score). We must additionally ensure that the functionality is preserved even in those (hopefully rare) scenarios in which the ML component does a misprediction.

To understand this issue, suppose the adversary model for CACC enables compromise of multiple perception channels of the victim vehicle, *e.g.*, communication channels providing velocity, position, and acceleration of its preceding vehicle. Even if the resiliency system detects an anomaly, it may not always correctly identify its source. For instance, if the adversary manipulates velocity and position channels, the resiliency system may only observe a discrepancy between the position and velocity values and the values of acceleration and conclude that the compromised channel is acceleration. A viable resiliency solution should obviously enable safe and efficient operation even if the mistake happens. However, the system must additionally enable proper functionality after the attack is completed. Since the resiliency system "thought" that the acceleration values deviate from ground truth while in reality the velocity and position values are compromised, the victim vehicle's perception of all three values would be different from reality at conclusion of the attack. Consequently, the victim vehicle could predict benign (ground truth) values of all three channels as anomalous. The resiliency system should ensure that the victim vehicle can recover from this situation and can continue engage in CACC.

The above issue must be accounted for during validation. We must show that, either (1) no matter what attack is instigated, the victim vehicle's perception is always within tolerable limits of reality; or (2) if the perception of the victim vehicle deviates significantly

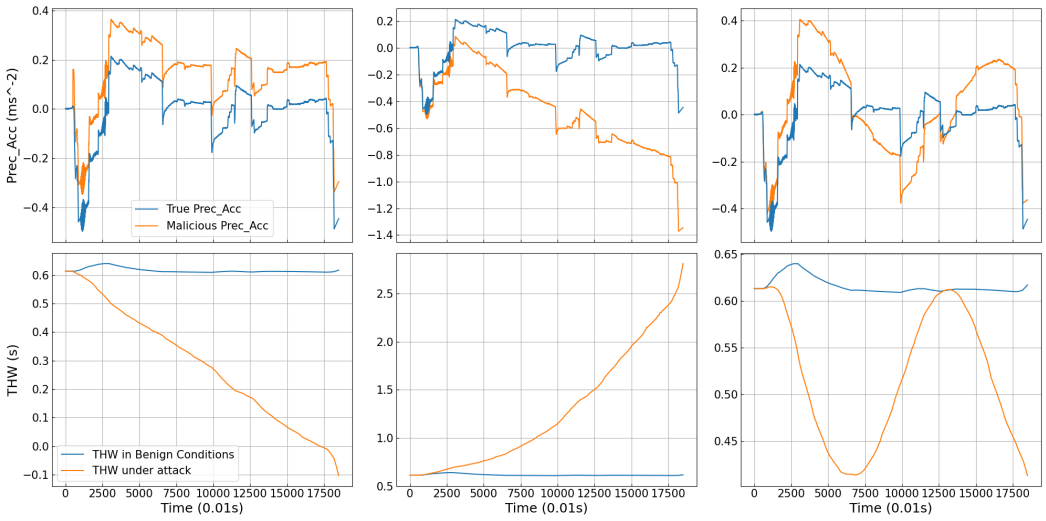

Figure 4: Continuous Mutation Attacks on CACC

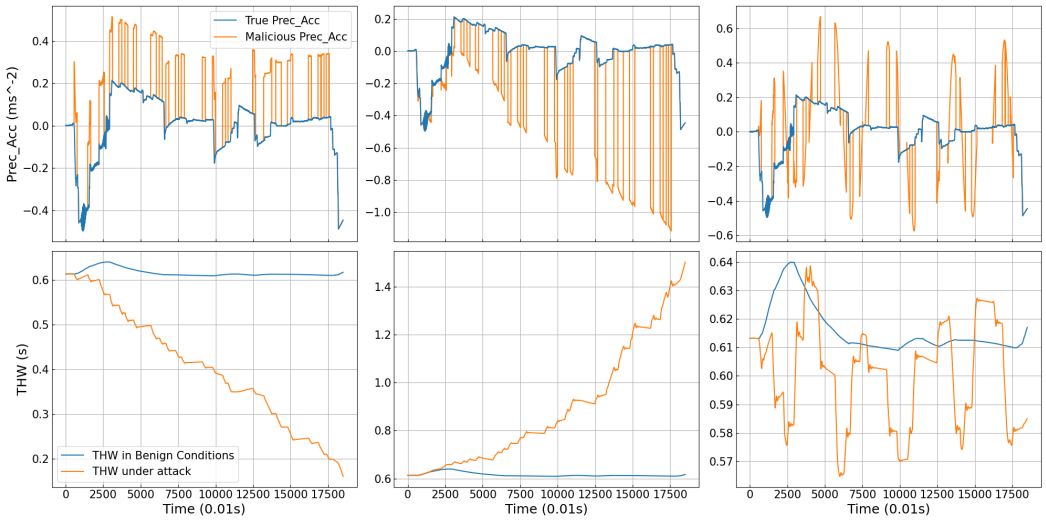

Figure 5: Cluster Mutation Attacks

from reality then its response still ensures safe and efficient operation, and after the attack is over it eventually returns to a state in which benign inputs are treated as benign.

6 CONCLUSION AND FUTURE WORK

In this paper, we considered the problem of real-time resiliency in connected autonomous vehicle applications. With increasing electrification and connectivity in vehicles, such applications will come with a large, complex, and vulnerable attack surface. We discussed some of the new challenges in validation of such applications, and our initial efforts to address these challenges through ReDeM.

The ReDeM framework is work in progress. In future work, we will consider more applications in ReDeM and potentially identify new challenges (and solutions) to validation. One target is multi-vehicle platooning. Our recent resilient platooning research [3] is based on a simplified implementation, but nevertheless brought interesting challenges. We will explore realistic platoon implementations and evaluate the efficacy of ReDeM validation strategies.

Acknowledgements: This research has been supported by the National Science Foundation under Grant No. CNS-1908549.

REFERENCES

[1] S. Agarwal et al. 2020. Ford multi-AV seasonal dataset. *The International Journal of Robotics Research* 39, 12 (2020), 1367–1376.
[2] M. Amoozadeh et al. 2019. VENTOS: Vehicular network open simulator with hardware-in-the-loop support. *Procedia Computer Science* 151 (2019), 61–68.
[3] S. Boddupalli, A. Hegde, and S. Ray. 2021. RePLACe: Real-time Security Assurance in Vehicular Platoons Against V2V Attacks. In *IEEE Intelligent Transportation System Conference.*

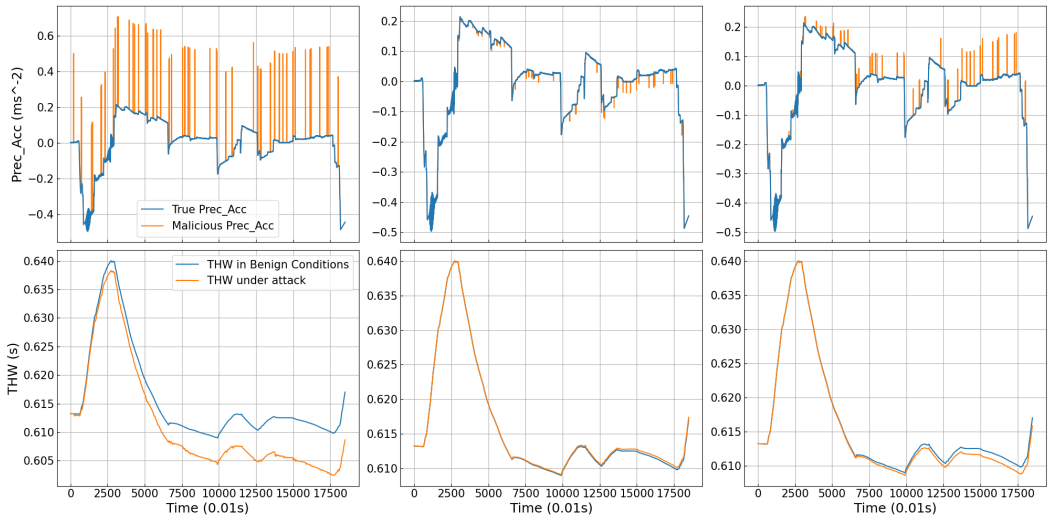

Figure 6: Discrete Mutation Attacks on CACC

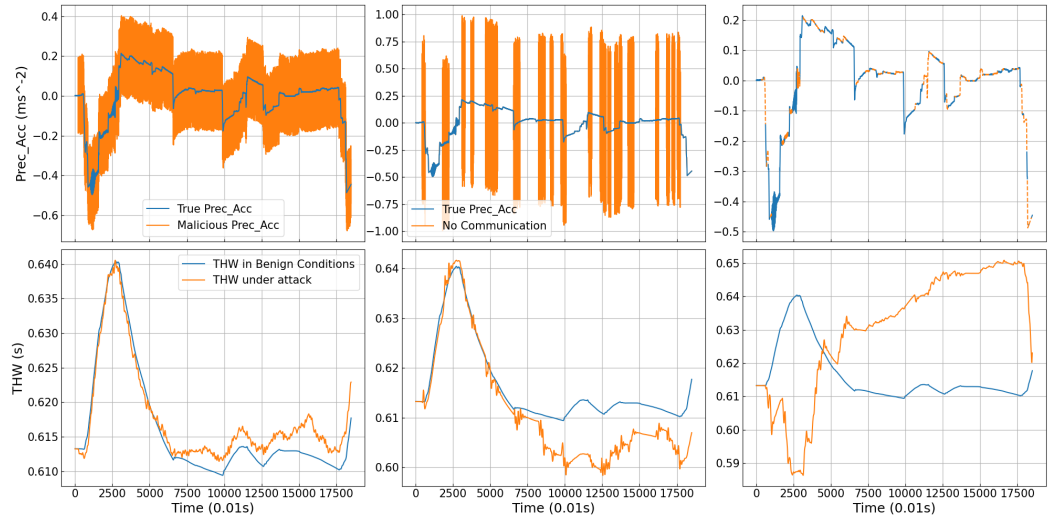

Figure 7: Random Mutation and Delivery Prevention Attacks on CACC

[4] S. Boddupalli, A. S. Rao, and S. Ray. 2022. Resilient Cooperative Adaptive Cruise Control for Autonomous Vehicles Using Machine Learning. *IEEE Transactions on Intelligent Transportation Systems* (2022), 1–18. https://doi.org/10.1109/TITS.2022.3144599

[5] S. Boddupalli and S. Ray. 2019. REDEM: Real-Time Detection and Mitigation of Communication Attacks in Connected Autonomous Vehicle Applications. In *IFIPIoT*.

[6] S. Checkoway et al. 2011. Comprehensive experimental analyses of automotive attack surfaces. In *20th USENIX Security Symposium (USENIX Security 11)*.

[7] A. Dosovitskiy et al. 2017. CARLA: An open urban driving simulator. In *Conference on robot learning*. PMLR, 1–16.

[8] D. Dua and C. Graff. 2017. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml

[9] J. Geyer et al. 2020. A2D2: Audi Autonomous Driving Dataset. *ArXiv* abs/2004.06320 (2020).

[10] K. Koscher et al. 2010. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*. 447–462. https://doi.org/10.1109/SP.2010.34

[11] P. Koschorrek et al. 2013. A multi-sensor traffic scene dataset with omnidirectional video. In *Ground Truth - What is a good dataset? CVPR Workshop 2013*.

[12] R. Krajewski, J. Bock, L. Kloeker, and L. Eckstein. 2018. The highD Dataset: A Drone Dataset of Naturalistic Vehicle Trajectories on German Highways for Validation of Highly Automated Driving Systems. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2118–2125. https://doi.org/10.1109/ITSC.2018.8569552

[13] P. Lopez et al. 2018. Microscopic Traffic Simulation using SUMO. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. 2575–2582. https://doi.org/10.1109/ITSC.2018.8569938

[14] C. Miller and C. Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* 2015, S 91 (2015).

[15] Atilla Özgür and Hamit Erdem. 2016. A review of KDD99 dataset usage in intrusion detection and machine learning between 2010 and 2015. *PeerJ Prepr.* 4 (2016), e1954.

[16] R. Toews. 2016. The biggest threat facing connected autonomous vehicles is cybersecurity. https://techcrunch.com/2016/08/25/the-biggest-threat-facing-connected-autonomous-vehicles-is-cybersecurity/

[17] A. Varga and R. Hornig. 2008. An overview of the OMNeT++ simulation environment. In *Proceedings of the 1st international conference on Simulation tools and techniques for communications, networks and systems& workshops*. 1–10.