Cyberattacks on Adaptive Cruise Controls and Emergency Braking Systems: Adversary Models, Impact Assessment and Countermeasures

Adriana Berdich and Bogdan Groza

***Abstract*—In the recent years, there has been a lot of focus on designing security for in-vehicle networks and detecting intrusions. Still, no countermeasure is perfect and most of the existing intrusion detection systems have a non-zero false negative rate which implies that adversarial frames may still go undetected on the bus. Unfortunately, answers are largely missing for what will happen with the vehicle in such circumstances, i.e., how is the safety of the vehicle and bystanders affected by adversarial actions that go undetected, while there are little or no answers on the acceptable misclassification rates in real-world deployments. In this work we attempt to provide such answers by pursuing an impact assessment for adversarial actions on the bus assuming low false negative rates. The assessment is based on the effects of such attacks on models for automatic emergency braking and adaptive cruise control systems that are implemented in Simulink, a commonly used tool for designing such systems in the automotive industry. To achieve this, we embed adversarial behaviour into the Simulink model, according to recently reported attacks on in-vehicle CAN buses. This allows us to asses the impact of adversarial actions according to existing safety standards and regulations.**

***Index Terms*—intrusion detection, CAN bus, security, safety**

I. Introduction and motivation

The security vulnerabilities of modern vehicles are now well known [1], [2], [3]. Adversarial actions can be easily coordinated from exposed interfaces that connect to the in-vehicle network, such as the OBD (On-board diagnostics) port, or can be mounted even from remote [4]. The main problem steams from the inadequate security design of in-vehicle buses, such as the Controller Area Network (CAN), which have only reliability mechanisms such as CRC (Cyclic Redundancy Check) codes to handle transmission errors. Despite the current standardization process which calls for the inclusion of cryptographic security [5], these problems will likely persist in the long run due to intrinsic difficulties in adopting security mechanisms. Notably, the payload of standard CAN frames is limited to 64 bits which makes it difficult to embed security elements. Even with security mechanisms in place, there is no such thing as a perfect security mechanism. For this reason, the use of Intrusion Detection Systems, has been vert recently included in automotive standards [6].

As expected, no security countermeasure is perfect and, unfortunately, intrusion detection systems usually have a non-zero false negative rate. That is, the number of undetected intrusions is non-zero and a limited number of adversarial frames may be accepted by the system. In the related work section, we enumerate several recent proposals for designing in-vehicle intrusion detection systems and, the detection rate, i.e., the ratio between the number of correctly identified frames as intrusions and the total number of intrusion frames, is never 100%. Consequently, attacks in which adversarial frames go undetected are realistic. Last but not least, false positives, i.e., legitimate frames which are classified as intrusions, may lead to legitimate frames being dropped. Still, none of the existing works on intrusion detection, endeavors to predict what will happen with the car in such situations when adversarial frames are to be accepted by legitimate ECUs. In this work we pursue an impact assessment that specifically addresses such situations.

Approach and contribution. While security and safety are distinct concerns, the former deals with adversarial interventions and the latter with the protection of traffic participants, it is clear that adversarial interventions may compromise safety. This is in fact acknowledged by the recent-most automotive cybersecurity standard ISO 21434 [7], which also accounts for the impact of security on safety. Arguably, safety is the most important factor considered by the adversarial impact assessment in ISO 21434, the other three factors being the financial, operational, and privacy impact. In this work we analyze the impact of three types of attacks (replay, DoS and fuzzing) on the ACC and AEB systems and their mitigation. Such countermeasures are also required by the UNECE R155 regulations [8] that have to be fulfilled for vehicle homologation in Europe.

We try to address the problem methodologically starting from the model of in-vehicle components used for driving assistance. These technologies gained a lot of momentum since 2014 when the Society of Automotive Engineers (SAE International) defined 6 levels of autonomy as follows: level 0 (no driving automation), level 1 (driver assistance), level 2 (partial driving automation), level 3 (conditional driving automation), level 4 (high driving automation) and level 5 (full driving automation) as specified in the SAE J3016 standard [9]. Vehicles reaching advanced autonomy levels are mandatory equipped with cameras, multiple long-range radars, multiple long-range LiDARs, multiple short-range LiDARs front and rear short and medium-range radars (a good overview of the various sensors for each autonomy level can be found in [10]). In Figure 1 we give an overview of ADAS functionalities with the corresponding sensors, radars and cameras. Many

Adriana Berdich and Bogdan Groza are with the Faculty of Automatics and Computing, Politehnica University of Timisoara, Romania. Email: {adriana.berdich, bogdan.groza}@aut.upt.ro. Corresponding author: Bogdan Groza.

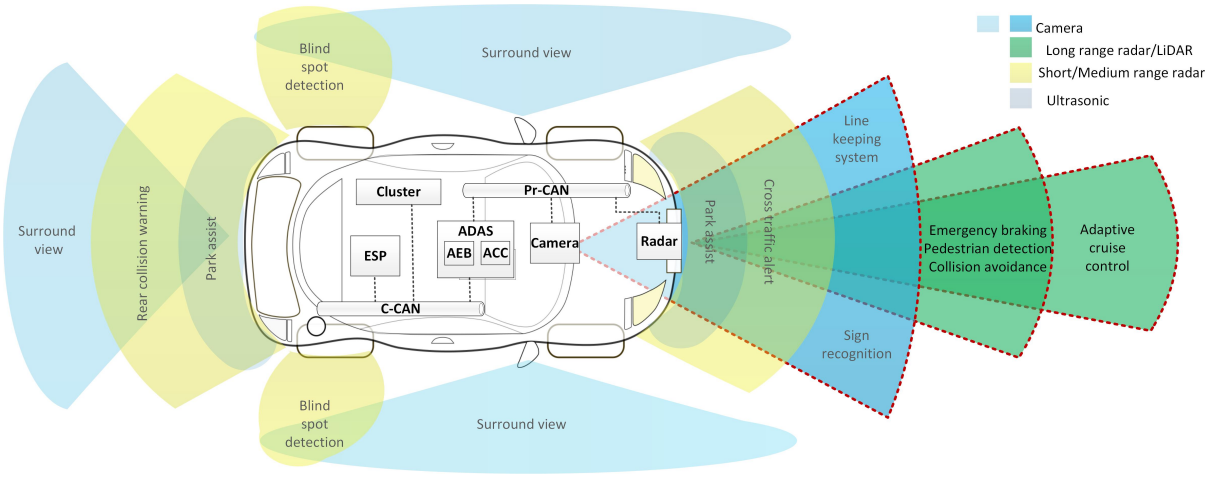


Fig. 1. The ADAS system overview (the dotted red line delimits components that are part of the model in this work)

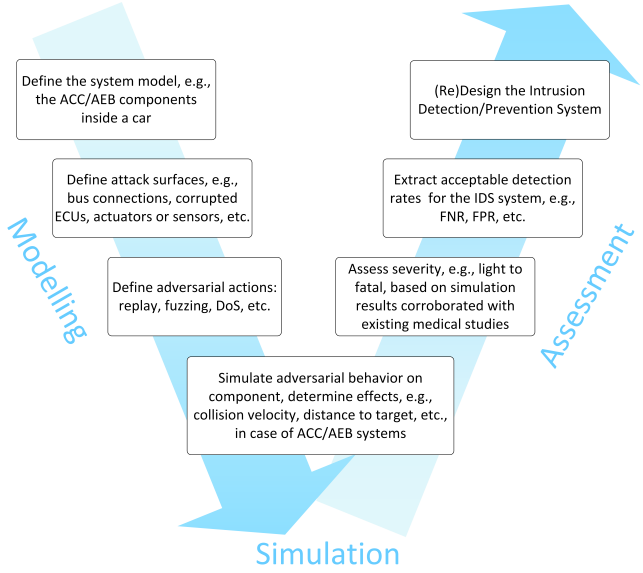


Fig. 2. A V-cycle inspired methodology for assessing the target intrusion detection/prevention rates

other functionalities are continuously developed for vehicles to reach the full autonomy level.

As a case in point, we use our analysis on conceptual attacks over two popular in-vehicle subsystems: the automatic emergency braking system and the adaptive cruise control. These are safety-critical driver assistance technologies and their exploitation may endanger the life of drivers, passengers or of other traffic participants. The Adaptive Cruise Control (ACC) system is designed to automatically adjust the vehicle velocity to a preset speed and maintain a minimum safe distance to the vehicle ahead, without pressing the brake or accelerator pedals. Another autonomous driving technology includes the Autonomous Emergency Braking (AEB), which uses the front long-range radar to warn the driver if a collision is imminent and apply the emergency braking to avoid it.

Figure 2 provides an outline of the procedure suggested and pursued by our work. To assess the impact of security attacks on the aforementioned ADAS systems we use a V-cycle inspired methodology. V-cycles are a commonly used, rigorous development process from the automotive industry, which starts from the system model, determining the attack surfaces and defining the adversary behaviour, then proceeds to the implementation stage which consists in running a model simulation and continues with an assessment stage that uses the simulation output which is corroborated with existing data related to passenger injuries in order to assess the severity of the attack and finally extracts the target detection rates for an IDS, eventually leading to its redesign. The contributions of our work can be summarized as follows:

- we address the security of two safety-critical components, the ACC and AEB systems, within a realistic in-vehicle network topology,
- we add attack surfaces on existing Simulink models for the aforementioned systems by implementing specific adversarial behaviors that are known to affect CAN buses,
- using these Simulink models, we bridge between security incidents and safety concerns by assessing the impacts of adversarial attacks via the CAN bus on safety according to ISO 26262-3,
- we discuss plausibility checks as countermeasures and show how they can improve the attack detection rates and consequently minimize the severity of security incidents on passenger safety.

Having this larger image in mind, the rest of the paper is organized as follows. In Section II we continue with a short discussion on related works. In Section III we give a brief overview of the ACC and AEB modules. The system model, attack surfaces and adversarial actions are discussed in Section IV. In Section V we depict a brief background on safety levels and injury classification based on collision speed. In Section VI we made an impact assessment based on the adversarial model. Section VII holds the conclusion of our work.

II. Related work

In this section we survey the performance of currently proposed in-vehicle intrusion detection systems, pointing out

on the non-zero false negative rates. Afterwards, we discuss some existing security studies dedicated to ADAS systems.

A. Current approaches to deal with intrusions and limitations

Current research works were largely focused on designing intrusion detection systems based on the analysis of data collected inside vehicles or from laboratory setups. The performance of such systems is judged starting from the following four markers: messages that are correctly labelled as attacks (true positives - TP), messages that are correctly labelled as legitimate (true negatives - TN), messages that are incorrectly labelled as attacks (false positives - FP), messages that are incorrectly labelled as legitimate (false negatives - FN). From these, the following two metrics are usually derived: the accuracy $\mathrm{Acc} = (\mathrm{TP}+\mathrm{TN})/(\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN})$, which is the ratio of correct predictions, and the precision $\mathrm{Precision} = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP})$, which is the ratio of legitimate frames that are correctly identified. While a large variety of mechanisms to detect intrusions have been studied, none of these are perfect, all of them having a non-zero false negative rates or, alternatively, an imperfect accuracy or precision. This is outlined in what follows.

For example, [11] reports false negative rates of about 2.8% but this results from an evaluation on simplified traffic that contains only 3 IDs which are simulated based on the Open Car Test-bed and Network Experiments (OCTANE) [12]. In-vehicle networks usually have dozens of IDs and the traffic is much more complex. The work in [13] uses Bloom filters on real-world traffic collected from a high-end vehicle, but this work reports a false negative rate of usually less than 10% but which may occasionally go up to 47% depending on the type of attack. By employing machine learning techniques, the work in [14] reports a false-negative rate in the range 0-49.94%, the larger margin holds in case of low-rate replay attacks which are particularly hard to detect.

By using an in-vehicle braking control system, the work in [15] reports an accuracy rate in the range of 37.32% up to 96.75%. The works in [16], [17] and [18] use entropy in order to detect intrusions, the later reporting an accuracy of 92.3% in case of packet injection (the accuracy increases to 100% in case of DoS attacks, but these attacks cannot be blocked since adversaries can always flood the bus with high-priority IDs). The authors in [19] report more optimistic results with a precision between 98.34% up to 99.76% but the results are on a specific dataset making it harder to compare with the rest of the papers. A more recent work [20] reports an accuracy between 71.55%-97.71% in detecting intrusions. Finally, by using the physical layer, i.e., bus voltage level, the authors in [21] report F-scores of 84.89% up to 98.94%.

For completeness, it is worth adding that in the past decade the industry has also introduced standards for protecting CAN buses. A recent standard from the AUTOSAR (AUTomotive Open System ARchitecture) consortium sets room for the implementation of intrusion detection systems [6]. But the exact mechanism to be implemented is left for the manufacturer to decide and, as outlined previously, none of the existing mechanism is perfect. Other standards, like the AUTOSAR security standard for on-board communication [5], demand authentication for the communication between two ECUs but this requires cryptographic functions that may not be suitable for all in-vehicle controllers. Moreover, the 32 bit security level with an 8 bit freshness parameter demanded by AUTOSAR [5] is very low (due to the limited 64 bit payload of CAN frames) and may leave room for intrusions to remain undetected. Finally, the recently released ISO/SAE 21434 [7] provides guidelines for the evaluation of cyber-security threats on vehicles and asks for threat analysis and risk assessments for which the results in our work may be as well applicable.

B. Security of driver assistance technologies

The work in [22] performs a black-box analysis for LiDAR spoofing attacks and achieves an average success rate of 80% on several models. Also, the authors propose two methods for attacks detection: CARLO (oCclusion-Aware hieRarchy anomaLy detectiOn) which reduces that attack success rate to 5.5% and SVF (sequential view fusion) which reduces it to 2.3%. As we later discuss in our experiments, a 5.5% success rate may have severe consequences.

Remote attacks, e.g., replay, relay, spoofing jamming and blinding attacks on camera and LiDAR sensors are studied in [23]. In [24] a run-time monitor system for the detection and isolation of attacks is implemented on an FPGA and tested over an AEB model. Also, [25] proposes a multilevel monitor for the detection and isolation of attacks injected on the CAN bus and sensors for a Cyber Physical System (CPS). For the implementation and evaluation of the proposed solution, the authors use a Simulink model of the Anti-lock Braking System (ABS) from MathWorks. In [26] a method is proposed for detecting attacks on the CAN bus and sensors in the ABS system. A method for the detection and mitigation of spoofing attacks on radars, which deliver data to the ACC system, is proposed in [27]. The authors also use for the experiments an ACC model from Simulink. They check the radar sensor data integrity using a spatio-temporal challenge-response (STCR) which sends signals in many random directions, then detects and excludes responses from untrustable sources.

In [28] the impact of jamming attacks on the Cooperative ACC system is analyzed. Also, [29] uses a model for a semi-autonomous ACC system to detect attacks on radar or on the wireless communication. In [30] a system is proposed for anomaly detection on Cooperative ACC based on statistical learning and kinematic models. In [31] a system for detection, correction of errors and authentication for ECUs in a steer-by-wire system is evaluated. To avoid low performance of the transmission system during a replay attack on an integrated motor-transmission system, [32] proposes to add a new reset controller and a speed synchronization controller. An anomaly detection method for steering stability control systems is also proposed in [33]. In a more recent work, the authors in [34] have tried to bridge security and real-time demands in accordance with ISO 26262, a standard which is also used by us later to assess the safety level in case of attacks.

A more recently emerged body of works, analyzes the impact of attacks on larger vehicle platoons. Needless to say,

autonomous driving technologies, including AEB and ACC are critical in this respect. The authors in [35] propose a method for anomaly detection (due to attacks or faults) in vehicle platoon networks based on dedicated sensors. Similarly, [36] discusses attack detection over vehicle platoons based on specific sensors. A mechanism for detecting attacks on vehicle platoons is also proposed in [37] where the authors analyze DoS, replay attacks and falsifications of sensor data. Also, [38] discusses DoS attacks on vehicular ad-hoc networks.

III. Brief overview of the cruise-control and autonomous braking subsystems

In this section we describe the ACC (Adaptive Cruise Control) and AEB (Autonomous Emergency Braking) modules that are used in our work to asses the impact of adversarial attacks on the CAN bus.

A. Adaptive Cruise Control (ACC)

The Adaptive Cruise Control (ACC) concept has a history which spans over three decades. Cruise control was introduced by Mitsubishi in 1992 and nowadays it comes either as the basic tempomat which maintains a constant speed or as the more advanced adaptive cruise control which also maintains a safe distance to the car in the front. The majority of cars are equipped with one of these two cruise control features. In Figure 3 (i) we give an overview of the ACC functionality by depicting a car that maintains a safe distance to the car in the front. The ACC module will automatically adjust the ego car speed such that it follows the front vehicle, possibly moving slower than the ego car, keeping a minimum safe distance between the ego car and the front vehicle. During this process, the driver will not press the accelerator or braking pedal.

We use the Simulink model for Adaptive Cruise Control with Sensor Fusion from MathWorks[1] to simulate the vehicle reaction in case of distinct attacks on the CAN bus. For the ACC functionality, we use the test scenario depicted in Figure 3 (ii). In this scenario we consider four vehicles running on a two lane road. The test vehicle (ego vehicle) is the blue car from the left which tries to maintain a fixed safe distance to the red car. Meanwhile the purple car slides left to overtake the yellow car, a point at which the ego car has to reduce its speed and change the lead vehicle for the purple car which now gets in front. Finally, as the purple car passes by the yellow car, it will shift back to the first lane and once again the ego vehicle has to change its lead vehicle for the red car.

B. Autonomous Emergency Braking (AEB)

The AEB module was introduced in 2009 by Volvo. The European New Car Assessment Program (Euro NCAP)[2] introduced AEB tests in 2014 for both low speed and high speed scenarios in the AEB City and AEB Interurban tests. Nowadays, the majority of modern cars are equipped with the AEB feature. The AEB module is part of the FCA system and it is a safety component used to prevent accidents or to significantly reduce the accident injuries by lowering the vehicle speed automatically when an obstacle, e.g., pedestrian, bicycle or sudden braking of the vehicle in front is detected by the front camera and long-range radar. In Figure 3 (iii) we give an overview of the AEB functionality in case when one vehicle approaches the other from rear. When the obstacle is detected by the front long-range radar and front camera, the driver is warned with an acoustic signal and with a visible warning light displayed by the cluster. There are three braking states: a first stage of softer pre-braking (partial braking), a second stage of more intensive pre-braking (partial braking) and then the full braking stage.

Again, we use the Simulink model for Autonomous Emergency Braking with Sensor Fusion from MathWorks[3] to simulate the vehicle reaction in case of distinct attacks on the CAN bus. For the AEB system, MathWorks includes the Car-to-Pedestrian Nearside Child test scenario from Euro NCAP [39] which we will also use in the impact assessment. In Figure 3 (iv) we depict this test scenario. Two vehicles are parked on the side of the road (1 meter from each other), the test vehicle (ego vehicle) is approaching on the road and a child is running from nearside (the view is obstructed) to cross the street. The goal of the AEB system is to trigger the automatic braking such that the car will avoid collision with the child.

C. A suggestive real-world architecture

Nowadays vehicles may contain over 100 ECUs organised in a distributed Electrical/Electronic (E/E) architecture, each functionality having its own ECU [40]. To clarify the real-world architecture of an in-vehicle ADAS system, embedding ACC/AEB functionalities, we now illustrate one specific architecture. For this depiction we have consulted both the Simulink models (discussed in the next section) and the service manual[4] of a recent vehicle equipped with such systems, i.e., a Hyundai Kona, publicly made available by the manufacturer.

Figure 4 gives a brief overview of the network topology. We depict two existing CAN buses: the private CAN bus (Pr-CAN) which is used for data exchange between the camera, radar and ADAS ECUs. Further the ADAS ECU and the ESC ECU are communicating on the Chassis CAN (C-CAN). These two buses, the Pr-CAN and the C-CAN are outlined in the Forward Collision-Avoidance Assist (FCA) system diagram of the Hyundai Kona. There is one difference between our model in Figure 4 and the Kona architecture as this includes a single Radar ECU that embeds the AEB functionalities. To make the drawing compatible with the Simulink models, we separate between camera, radar and the ADAS ECU which carries the specific computational tasks.

IV. System and Adversary model

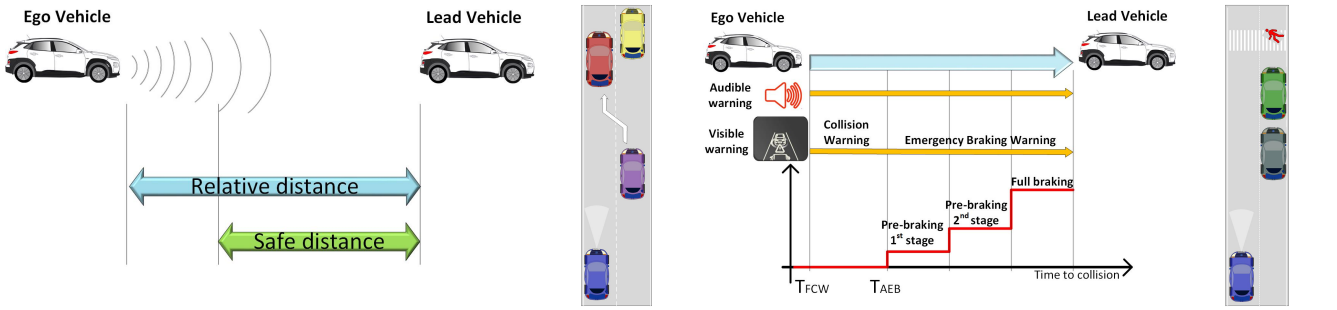
In this section we first discuss the system model for the two components based on the existing Simulink schemes to which we also add attack surfaces, then we present the adversarial behavior that is considered by our work.

[1]https://nl.mathworks.com/help/driving/ug/adaptive-cruise-control-with-sensor-fusion.html

[2]www.euroncap.com/en/vehicle-safety/safety-campaigns/2013-aeb-tests/

[3]https://nl.mathworks.com/help/driving/ug/autonomous-emergency-braking-with-sensor-fusion.html

[4]www.hkona.com/components_and_components_location-1136.html



(i) ACC functionality (ii) ACC scenario (iii) AEB functionality (iv) AEB scenario

Fig. 3. Overview of the ACC functionality (i), test scenario (iii), AEB functionality (iii) and test scenario (iv)

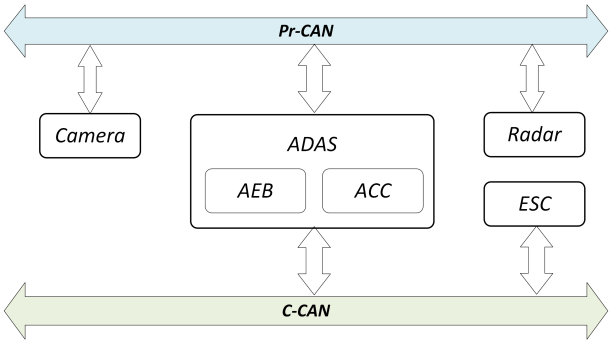


Fig. 4. Suggestive network topology for an ADAS ECU

A. System model and attack surfaces

The Matlab model of vehicle dynamics that we employ is based on the commonly used equations, used by various papers such as the already mentioned work from [33], which can be synthesized as follows:

$$\begin{cases} \ddot{x}(t) = \dot{y}(t)r(t) + \frac{F_{xf}(t)+F_{xl}(t)+Fxext(t)}{m} \\ \ddot{y}(t) = -\dot{x}(t)r(t) + \frac{F_{yf}(t)+F_{yl}(t)+Fyext(t)}{m} \\ \dot{r}(t) = \frac{aF_{yf}(t)+bF_{yr}(t)+M_{zext}(t)}{I_{zz}} \end{cases}$$

Here $\dot{x}, \ddot{x}$ are the longitudinal velocity and acceleration, $\dot{y}, \ddot{y}$ are the lateral velocity and acceleration, r is the angular velocity, m is the vehicle's mass, M_{zext} is the external moment related to the z-axis, I_{zz} is the vehicle body moment of inertia on the z-axis, F_{xf}, F_{xr} are the longitudinal forces applied to the front and rear wheels, F_{yf}, F_{yr} are the lateral forces applied to the front and rear wheels, F_{xext}, F_{yext} are the external forces applied to the vehicle along the x and y-axes, a is the distance of the front and rear wheels and b is the distance from the projection point of the vehicle center of gravity to the axle plane.

In what follows, we focus on the description of the ADAS subsystems that we analyse.

1) The ACC module and its attack surfaces: In Figure 5 we depict the schematic of the Simulink model used for the ACC module and the attack surfaces on this subsystem. As we already mentioned in the previous section, this model contains two main subsystems: i) the *ACC with sensor fusion* which processes the data from sensors and computes the acceleration requests on the vehicle to achieve the target speed and ii) the *vehicle and environment* which contains the model of the ego car and that of the environment.

The ACC with sensor fusion subsystem is also split in two subsystems which we consider as separate ECUs communicating over the Pr-CAN. The first, is the *tracking and sensors fusion* subsystem embodied by the Camera/Radar ECU. In this subsystem, incoming data from the camera and radar are processed to compute the relative velocity and distance. The relative velocity is the difference between the velocity of the lead vehicle and the velocity of the ego car, similarly, the relative distance is the distance between the ego car and the lead vehicle. The relative velocity and distances are consumed by the adaptive cruise control module, which embodies the ACC/ADAS ECU. The relative velocity and distance are received by the ACC ECU via the Pr-CAN bus. The ACC ECU computes the required acceleration to achieve the target vehicle speed, maintaining a safe distance to the lead vehicle based on the relative distance and velocity received via the Pr-CAN and based on the longitudinal velocity of the ego car received via the C-CAN. In the Simulink model the vehicle speed controller is implemented in two ways by using two types of controllers: a classical controller and a MPC (Model Predictive Control) controller. In our simulation we use the classical controller. The controller checks if the relative distance is lower than the safe distance, then requests to reduce the acceleration in order to maintain a safe distance, otherwise, if the relative distance is greater than the safe distance, the target is to achieve the preset velocity and further maintain the safe distance. The safe distance between the ego car and the lead car is computed as $D_{safe} = D_{default} + T_{gap} \times V_{ego}$, where $D_{default}$ is the default spacing, T_{gap} is time gap between the ego car and the lead vehicle and the V_{ego} is the longitudinal velocity of the ego car.

The vehicle and environment subsystem is split into four components: the vehicle dynamics model, the driver steering model, the actors and sensors simulation and the curvature. The vehicle dynamics subsystem and the driver steering model represents the vehicle, including the ESC ECU. The *vehicle dynamics* subsystem models the vehicle dynamic based on a bicycle model. This subsystem uses the acceleration received from the ACC ECU and the steering angle to compute the longitudinal velocity which is send back to the ACC ECU. The communication is done via the C-CAN bus. In the driver steering model, the steering of the ego vehicle is computed

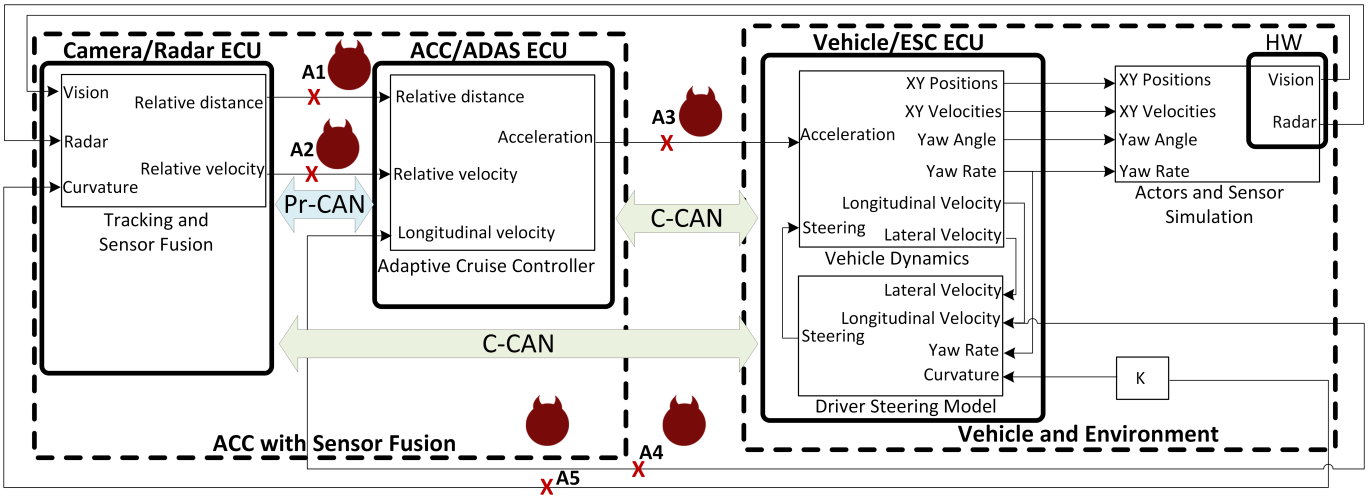


Fig. 5. ACC model and its attack surfaces

based on the lateral velocity, longitudinal velocity, yaw rate and curvature. The *actors and sensor simulator* subsystem contains the driving scenario and acquires the data from the sensors. The K subsystem computes the curvature.

Attack surfaces on vehicles may range from entry points on the bus (such as the commonly exploited OBD port or in-vehicle infotainment units) up to corrupted in-vehicle ECUs (which may be the result of flashing with a malicious software or of a supply chain attack). In this work we focus on CAN bus adversaries. That is, we define the following 5 adversaries on the ACC model A1 to A5 which are acting by injecting corrupted data on the bus. Two of them A1 and A2 are on the Pr-CAN reporting false data on the relative distance and velocity and the other three A3, A4 and A5 are on the C-CAN reporting false data on acceleration, longitudinal velocity and curvature. Vision and radar related hardware may also be corrupted, but these modules are not communicating over the CAN bus and we omit them in our analysis which targets CAN bus intrusions alone. Whether the adversarial frames are caused by malicious nodes plugged on the bus or due to corrupted units is of no concern for the current work. In all cases however we assume that the attack frame has a small probability of occurrence, e.g., $p = 0.1$. This may be due to an existing IDS which filters malicious frames (this is the main assumption of our work) but it may also be the case that a corrupted component reports false values at a low rate in order to remain inconspicuous. Clearly, a malicious unit that continuously injects false data in the system at a high rate may be easily detected during system testing and replaced.

2) The AEB module and its attack surfaces: In Figure 6 we depict the AEB model and the attack surfaces on it. The AEB model is structured similarly to the ACC model, i.e., it contains two subsystems: i) the *AEB with sensor fusion* subsystem which processes the data from sensors and computes the deceleration requests if a potential collision is detected and ii) the *vehicle and environment* which contains the model of the ego car and that of the environment.

The AEB with sensor fusion subsystem is now split into four subsystems which we group under two ECUs. The *tracking and sensors fusion* subsystem which contains the Camera/Radar ECU as in the case of the ACC model. Then, another ECU is the AEB/ADAS ECU which contains the other three subsystems: the AEB controller, the speed controller and the accelerator robot. In the AEB controller the FCA and AEB algorithms are implemented. The deceleration and the AEB status (which indicates if the AEB is active) are computed based on the relative distance and velocity received from the Camera/Radar ECU via the Pr-CAN and based on the longitudinal velocity of the ego car received via C-CAN from Vehicle/ESC ECU. The acceleration of the ego vehicle is computed in the speed controller subsystem based on the longitudinal velocity using a proportional integral (PI) controller. Finally, in the *accelerator robot* subsystem the throttle is computed based on the AEB status and acceleration. If the AEB is active the throttle is set to zero, otherwise the throttle is equal with the computed acceleration.

The *vehicle dynamics* subsystem is similar to the same subsystem from the ACC model, except for the fact that instead of the acceleration in the ACC model, here we use the deceleration and the throttle signals received via the C-CAN from the AEB/ADAS ECU.

Based on a similar reasoning to the ACC model, we define 6 adversaries for the AEB. Adversaries A1 to A6 are acting by injecting corrupted data on the bus, two of them A1 and A2 are on the Pr-CAN reporting false data on the relative distance and velocity and the other four A3, A4, A5 and A6 are on the C-CAN reporting false data on the deceleration, throttle, longitudinal velocity and curvature. For the same reason as in the case of the ACC model, we are not interested in the corruption of the hardwired Vision and Radar modules.

B. Adversarial actions and impact on safety

As stated in the introduction, an increased number of attacks and vulnerabilities have been reported for modern cars [1], [2], [3]. Regardless of the attack entry point, in order to cause problems at the control system level, the adversary has to

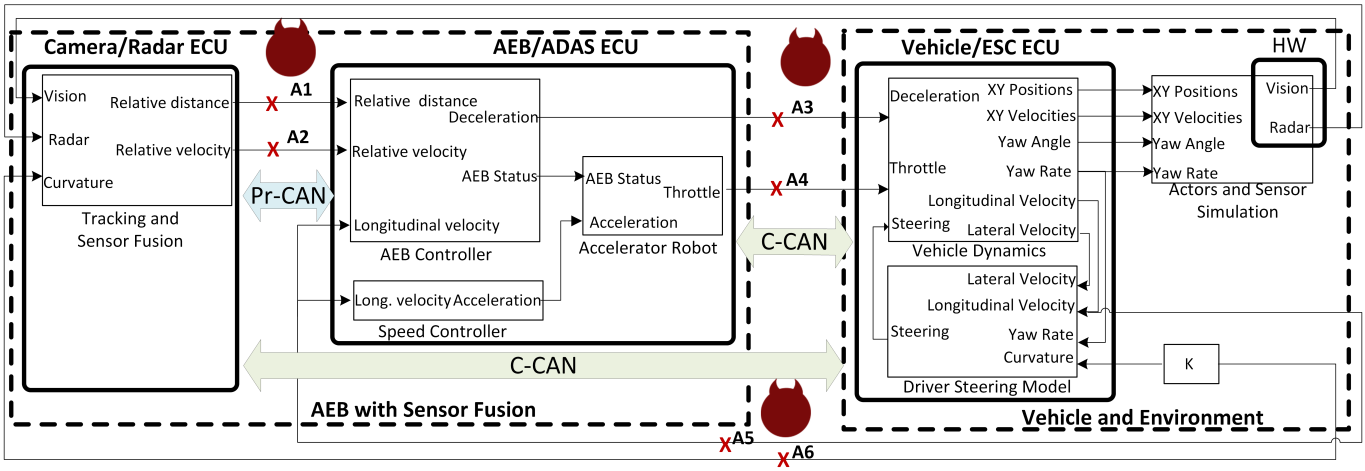


Fig. 6. AEB model and its attack surfaces

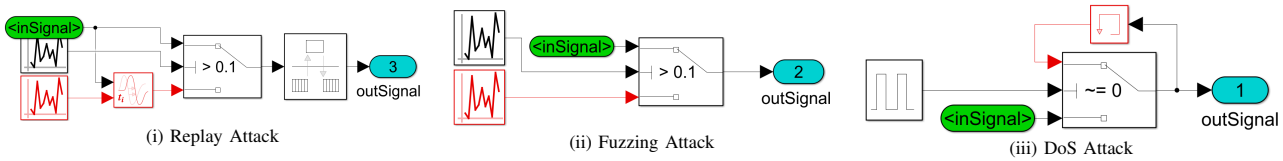


Fig. 7. CAN bus attacks modeled in Simulink: Replay Attack, Denial of Service (DoS) Attack and Fuzzing Attack

TABLE I
ATTACKED SIGNALS RANGES

Signal	Accel.	Decel.	Relative Distance	Relative Velocity	Long. Velocity
Range	-3...3	-12.8...12.7	0...200	0...255	0...255
Unit	m/s^2	m/s^2	m	km/h	km/h

gain access to the CAN bus. Therefore, in our analysis we consider three types of attacks on CAN buses that have been commonly evaluated by research works dedicated to intrusion detection for in-vehicle buses: replay, Denial of Service (DoS) and fuzzing attacks. These are modeled by us as Simulink components in order to embed them in the car control system model. The Simulink attack blocks are summarized in Figure 7, we discuss each of them in what follows. To cope with the possibility that an intrusion detection system is in place and not all adversarial frames are to be received, each attack is assigned a fixed success probability.

Replay Attacks. Under this type of attack CAN frames are re-transmitted with a random delay. To model the attack probability we generate a random signal in the 0-1 range with 0.1 resolution and if the signal is higher than a fixed threshold, i.e., the attack probability, the legitimate signal is received, otherwise the CAN bus is attacked and a previous signal, delayed by a random value between 0.1s and 1s, is received.

Fuzzing Attacks. Under this type of attack, random values are injected on the CAN bus. We use the same procedure to simulate the attack probability. If the signal is greater than the threshold, the legitimate signal is received, otherwise the CAN bus is under attack and a random value is received. The range of the random value complies with range of the legitimate input of the model, except the range of relative velocity where we consider only the positive range, i.e., 0...255km/h. In Table I we depict the ranges used for the fuzzing attack.

DoS Attacks. Under this type of attack the CAN bus can no longer transmit data and the signals are freeze. This type of attack is specifically difficult to stop since an adversary can always write high priority frames on the bus and the only solution is to decouple parts of the network as recently proposed in [41]. Distinct to the previous attacks, the DoS is a continuous attack, i.e., no data is received during the attack. To model this, we use a rectangular signal, if the signal is 0 the legitimate signal is received, otherwise the CAN bus is under attack and the previously recorded signal on the bus is maintained. The attack probability can be interpreted as the percentage of a second that the attack is active. Concretely, for a DoS signal with attack probability $p = 0.5$, the DoS will be active $\Delta = 500ms$, for a DoS signal with $p = 0.2$, the DoS will be active $\Delta = 200ms$ and for a DoS signal with $p = 0.1$, the DoS will be active $\Delta = 100ms$.

Threat modelling can be also employed for a better understanding of the causes and impact of the previous adversarial actions. ISO 21434 [7] mentions several approaches, e.g., EVITA (E-safety Vehicle Intrusion Protected Applications), TVRA (Threat, Vulnerability, Risk Analysis), PASTA (Process for Attack Simulation and Threat Analysis), and STRIDE (spoofing, tampering, repudiation, information disclosure, denial of service, elevation of privilege), which are used in several recent works [42], [43]. Due to space constraints and since our work is focused strictly on the impact of the attacks and defining countermeasures for two car components (ACC and AEB), such an analysis would be out of scope. It is worth mentioning however that STRIDE considers six types of threats: spoofing, tampering, repudiation, information

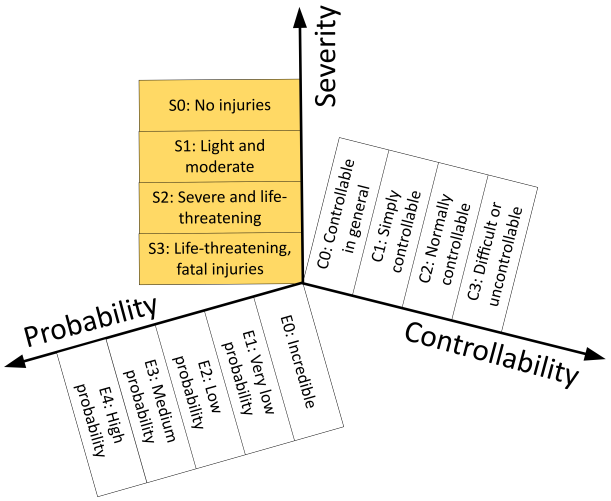


Fig. 8. A 3D view over ISO 26262-3 levels

disclosure, denial of service and elevation of privilege. With respect to the CAN bus communication, all these attacks should ultimately manifest in the form of the malicious actions considered by our adversary model.

V. Brief background on safety levels

In this section we discuss safety level according to standards, i.e., ISO 26262-3, and provide a brief analysis on injury classification based on existing literature data.

A. Safety levels according to ISO 26262-3:2018

The ISO 26262 standard is an adaptation of the IEC 61508 standard for functional safety of road vehicles in the automotive industry. The ISO 26262 standard contains all functional safety steps from the V-cycle development process which includes: concept, requirements specification, design, implementation, integration, software verification, system verification and validation, production and maintenance. In our work we use the ISO 26262-3 [44] which defines several ASIL (Automotive Safety Integrity Levels) levels for in-vehicle subsystems. Basically, four ASIL levels from A to D are defined, where A is the minimum and D the maximum level. Another level exists which is QM (quality management) and is attributed to non-safety relevant components.

The ASIL level is determined by asserting the following three classes in case of a failure: its severity, its probability of exposure and its controllability. In this work our goal is to classify the hazardous events produced by attacks on the CAN bus according with these three classes used for ASIL determination. The *classes of severity (S)* are defined from S0 to S3 with the following allocation: S0 - no injuries, S1 - light and moderate injuries, S2 - severe and life-threatening injuries (survival probable) and S3 - life-threatening injuries were survival is uncertain or fatal injuries. As an additional detail in ISO 26262-3 [44], the severity class is determined using the Abbreviated Injury Scale (AIS) which contains seven classes with the following interpretation: AIS 0 - no injuries, AIS 1 - light injuries, AIS 2 - moderate injuries, AIS 3 - severe but no life-threatening injuries, AIS 4 - severe injuries (life-threatening, survival probable), AIS 5 - critical injuries (life-threatening, survival uncertain), AIS 6 - extremely critical or fatal injuries. In this more detailed context, the four severity classes S0-S3 are further allocated as follows: S0 - AIS 0 and less than 10% probability of AIS 1-6 or no safety related damage, S1 - more than 10% probability of AIS 1-6 but not S2 or S3, S2 - more than 10 % probability of AIS 3-6 and not S3 and S3 - more than 10 % probability of AIS 5-6.

We also note that ISO 26262-3 [17] quantifies, in addition to the severity, the probability of exposure as well as the controllability of the vehicle in case of a failure. The *classes of probability of exposure (E)* are defined from E0 to E4 with the following allocation: E0 - incredible, E1 - very low probability, E2 - low probability, E3 - medium probability and E4 - high probability, e.g., every driving cycle. Since our work is focused on attacks and we assume the adversary is present on the bus and able to act, it makes no sense for our context to address these classes of exposure as we consider this class to be E4, i.e., high probability (we know the adversary is there and has some fixed success rate). The *classes of controllability (C)* are defined from C0 to C3 with the following allocation: C0 - controllable in general, C1 - simply controllable, C2 - normally controllable and C3 - difficult to control or uncontrollable, e.g., less than 90% of the drivers are able to take control over the fault. In our case, i.e., for the ACC and AEB modules, the driver can always override them and thus the controllability class will be C0. However, we consider that an adversary can still inflict damage since the driver may confidently rely on the system without knowing that it has been compromised.

B. Injury classification based on collision speed

In order to obtain the correct severity level in case of an attack, we need to establish a relation between vehicle speed at the time of collision and the injury level. A white-paper from the World Health Organization (WHO) [45] depicts the probability of death of the pedestrian as a sigmoid-like function centered at 50% probability of death for a car running at 50 km/hour. Above 60km/h the probability of death increases between 90-100%.

Two datasets, which include all years old groups, are analysed in [46] and the results show that at 50km/h there is a 75% likelihood of light injuries, 21.9% of severe injuries and 3.1% of fatal injuries. The work in [47] surveys several papers from 1997-2005 and outlines very mixed results. For example, pedestrian fatality ranges from 7% up to 85% at 50 km/h depending on the study/year. In principle, older papers seem to favour a higher risk of fatality, while some of the recent papers (but not all) favour a decreased risk of fatality. Finally, the work in [48] seems to be the closest to our needs. Basically, for adults 15-59 years old, an impact speed below 37km/h has a higher probability of light injuries, between 37-67km/h it results in serious injuries and above 67km/h the most likely outcome is fatal. The authors in [49] shows results based on a dataset from the police and an On the Spot (OTS) study. The cumulative probability of severity is depicted as a sigmoid-like function centered at 50% cumulative probability

of severity. An impact speed below 30km/h has a higher cumulative probability of light injuries, between 30-47km/h it results in serious injuries and above 47km/h the most likely outcome is fatal. Also, [49] analyses the case of side impacts with another vehicles. We consider this case to be the closest to our scenario from the ACC model, where the impact is from the rear or side of the car. Again, the cumulative probability of severity is depicted as a sigmoid-like function centered at 50% cumulative probability of severity. An impact speed below 24km/h has a higher cumulative probability of light injuries, between 24-40km/h it results in serious injuries and above 40km/h the most likely outcome is fatal.

In what follows, we choose to classify the severity in case of pedestrians for the AEB model based on the scales from [48] and [49] due to their similarity in injury classification to the AIS stages from ISO 26262-3. In their interpretation, the injuries are classified: light, severe and fatal. In this work we link these three classes at the AIS stages used in ISO 26262-3 for the severity determination as follows: light - AIS 1 (light injuries) and AIS 2 (moderate injuries), severe - AIS 3 (severe but no life-threatening injuries) and AIS 4 (severe injuries, life-threatening, survival probable), and fatal - AIS 5 (critical injuries) and AIS 6 (extremely critical or fatal injuries). In case of car collisions for the ACC model, we will classify the car driver injuries based on the scale from [49] which, as stated, is similar to ISO 26262-3.

VI. Impact assessment for the adversary actions

In this section we assess the impact of the adversarial behavior on the vehicle and traffic participants based on existing standards and regulations.

A. Simulations results regarding collision velocity

We start by determining collision velocity in case of each attack surface and adversary behaviour. Tables II and III gather the results in case of the 5/6 attack surfaces and 3 types of adversary actions. Note that since the curvature is constant, a replay or DoS attack will not change the previously reported value, a reason for which these attacks are omitted and only fuzzing attacks are considered for the curvature. We use *no coll.* as placeholder to denote that no collision takes place. In case of the ACC model the car continues at the predefined speed while for the AEB model the car will successfully stop, i.e., the speed is 0. Also, we use *n/a* as placeholder in case that the experiment is not applicable in the specific context, e.g., this happens for the DoS attack which cannot be implemented at an attack time of $50ms$, i.e., $p = 0.05$ since the model has a simulation step of $100ms$ (which makes it feasible to implement DoS only at multiples of $100ms$). For each attack we simulate the impact at 4 attack probabilities: $p = 0.5$, $p = 0.2$, $p = 0.1$ and $p = 0.05$ which give good coverage for the results reported so far on the success rate of intrusion detection systems.

1) Experimental result for the simulations on the ACC model: For the ACC model in case of an attack with low probability, i.e., $p = 0.1$ and $p = 0.05$, no collision occurs while when increasing the attack probability to $p = 0.2$ or $p = 0.5$ a collision occurs for fuzzing attacks. The ego car velocity at collision is between 64.8km/h and 93.6km/h for $p = 0.2$ and between 88.05km/h and 112.21km/h in case of $p = 0.5$. Apparently, none of the attacks on the longitudinal velocity produces a collision at the attack probabilities we define. Also, in the ACC model the replay and DoS attacks did not produce a collision for the defined probabilities. For replays, the explanation is rather obvious since by replaying existing messages the car will generally behave in the way intended by the previous command with delays in taking the new command. For the case of DoS attacks, the duration of $500ms$ appears to be too low to cause problems in this scenario. By further experiments we determined that a DoS of $3.5s$ will be needed to cause a collision in case of the ACC system with the velocities in our model. In cases when no collision occurs, the distance to the target decreases with an increase in the attack probability. This is more obvious for the replay and DoS attacks while for fuzzing attack it may not always be the case due to the insertion of random values.

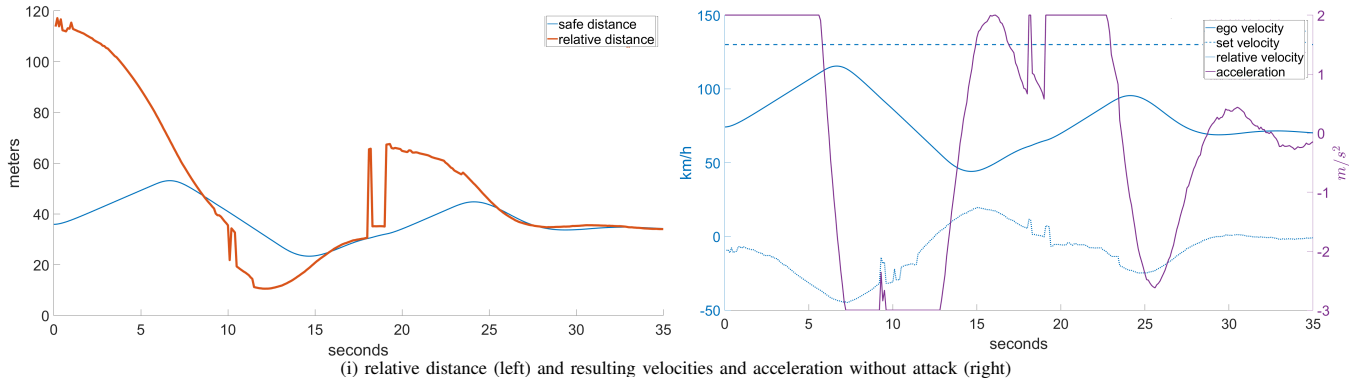
Figure 9 illustrates the effect of fuzzing attacks on the relative distance with attack probability $p = 0.2$. Figure 9 (i) depicts the signals without the attack while (ii) shows the same signals under the fuzzing attack on the relative distance. The left side of Figure 9 (ii) shows the attacked relative distance signal and it can be seen that slightly before the 13-th second the collision occurs as the relative distance decreases to zero. In the right side of Figure 9 (ii) we show the impact of the attack on the ACC systems, i.e., the acceleration is drifting and the ego velocity is also changed.

2) Experimental result for the simulations on the AEB model: For the AEB model, collisions occur even with replay and short term DoS attacks. Also, collisions occur even at the lowest attack probability on deceleration, i.e., $p = 0.05$. The vehicle speed at the time of collision increases with the attack probability. For example, a fuzzing attack on deceleration with $p = 0.05$ results in a collision velocity of 29.95km/h, while with $p = 0.1$ and $p = 0.2$ the collision velocity is 43.88km/h. Finally, with $p = 0.5$ the velocity at the time of collision is 55.40km/h. The attacks on the relative velocity and distance also resulted in collisions while the attacks on the throttle, longitudinal velocity and on the curvature did not cause such a collision at the predefined attack probabilities. Also, we observed that the fuzzing attack produces a collision at high speed, i.e., 29.95km/h at $p = 0.05$ up to 55.40km/h at $p = 0.5$, followed by the replay attack which produces a collision with a speed between 30.45km/h and 43.44km/h at $p = 0.5$. Finally, the DoS attack produces a collision with a vehicle speed between 15.69km/h and 17.34km/h at $p = 0.5$. For both the ACC and AEB modules, the distance to the target was computed as the difference between the relative distance (in case of an attack on the relative distance we consider the relative distance before attack) and the length of the car, i.e., $d_t = d_r - 3.7$ where d_t is the distance to the target, d_r is the relative distance and 3.7m is the car length.

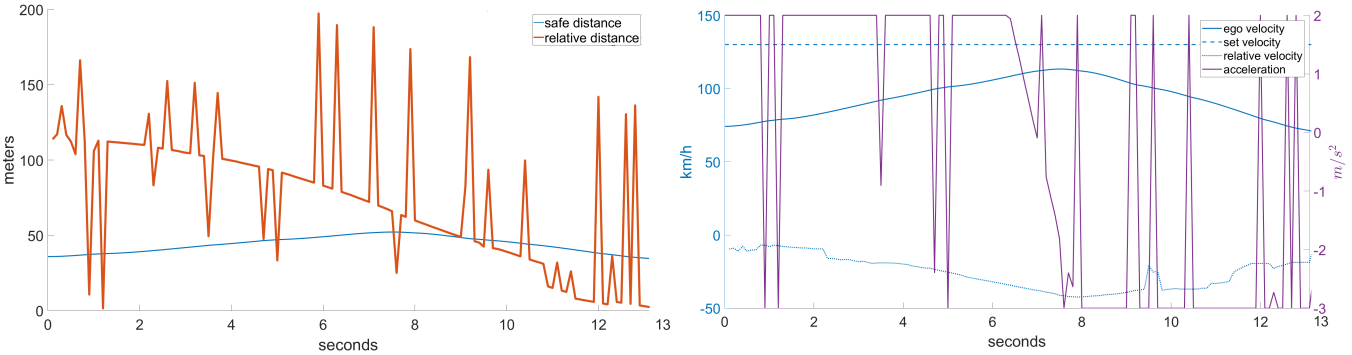
Figure 10 shows the effects of a fuzzing attack on the deceleration signal with attack probability $p = 0.1$. Figure 10 (i) depicts signals without the attack and Figure 10 (ii) shows the signals following a fuzzing attack on the deceleration.

TABLE II
ACC RESULTS: COLLISION VELOCITY AND DISTANCE TO TARGET AT VARIOUS ATTACK SUCCESS RATES

Signal	Attack	$p = 0.5$		$p = 0.2$		$p = 0.1$		$p = 0.05$	
		Ego velocity at collision[km/h]	Distance to target[m]	Ego velocity at collision[km/h]	Distance to target[m]	Ego velocity at collision[km/h]	Distance to target[m]	Ego velocity at collision[km/h]	Distance to target[m]
A1 (relative distance)	Fuzzing	100.8	0	72	0	no coll.	5.82	no coll.	5.82
	Replay	no coll.	3.03	no coll.	5.02	no coll.	5.89	no coll.	5.91
	DoS	no coll.	4.08	no coll.	5.72	no coll.	5.93	n/a	n/a
A2 (relative velocity)	Fuzzing	108	0	93.6	3.54	no coll.	3.54	no coll.	5.40
	Replay	no coll.	6.17	no coll.	5.68	no coll.	6.65	no coll.	6.65
	DoS	no coll.	5.58	no coll.	5.97	no coll.	6.62	n/a	n/a
A3 (Acceleration)	Fuzzing	88.05	0	64.8	0	no coll.	6.55	no coll.	6.00
	Replay	no coll.	2.65	no coll.	4.92	no coll.	5.66	no coll.	6.57
	DoS	no coll.	3.64	no coll.	5.52	no coll.	5.89	n/a	n/a
A4 (Long. velocity)	Fuzzing	no coll.	24.15	no coll.	5.36	no coll.	2.36	no coll.	5.81
	Replay	no coll.	5.66	no coll.	6.01	no coll.	6.05	no coll.	6.85
	DoS	no coll.	5.78	no coll.	6.04	no coll.	6.06	n/a	n/a
A5 (Curvature)	Fuzzing	112.21	0	73.8	0	no coll.	10.21	no coll.	11.88



(i) relative distance (left) and resulting velocities and acceleration without attack (right)



(ii) fuzzing attack on relative distance (left) and resulting velocities and acceleration under a fuzzing attack (right)

Fig. 9. ACC signals without adversarial intervention (i) and under a fuzzing attack on the relative distance (ii)

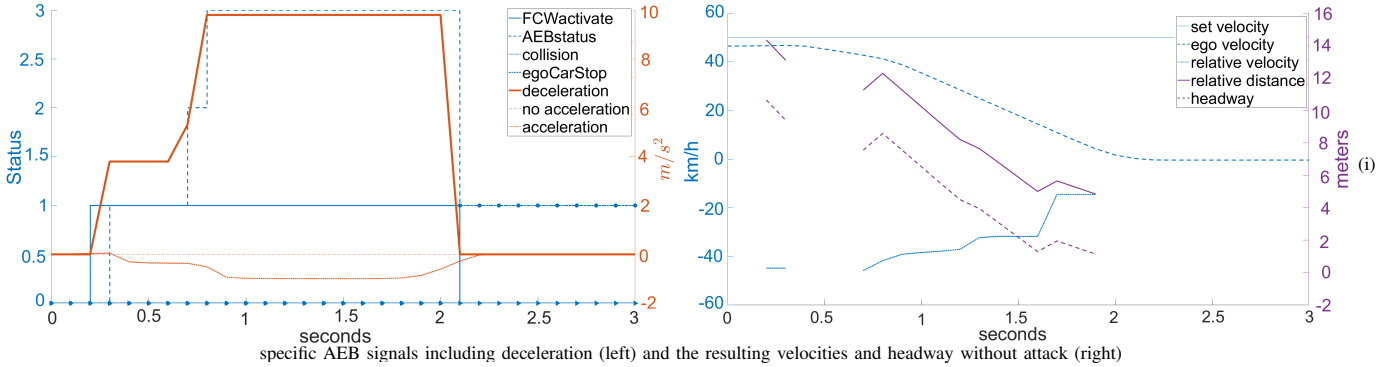
In the left side of Figure 10 (ii) we show the attacked deceleration signal where it can be seen that the impact occurs at 1.5 seconds and the collision signal is set to 1, while the egoCarStop signals is still set to zero. In the right side of Figure 10 (ii) we show the relative distance which gets to 3.7m at collision time (3.7m is the length of the car) and further progresses to 2m.

3) Integration with real-world network simulators: The Simulink models can be interfaced with network simulators like the CANoe environment[5] which can reproduce traffic from a real-world bus. As a proof-of-concept we have also linked a CAN bus simulation with the Simulink components by using the Matlab Integration Package. This allowed us to send and receive frames from a CAN bus on top of which a more complex network architecture can be implemented. Having the Simulink model interfaced with CANoe, a Replay Block was added on the PrCAN which is able to reproduce a CAN Trace extracted from a real car. For this we used data collected in a real world Hyundai i20, a small hatchback produced between 2014-2020. With this trace running in the background, a small bus overhead of about 35% was added. The increased busload had no influence on the results, suggesting that the attacks will have similar effects even on more complex buses. A more in-depth exploration of more complex networks may be subject of future work for us.

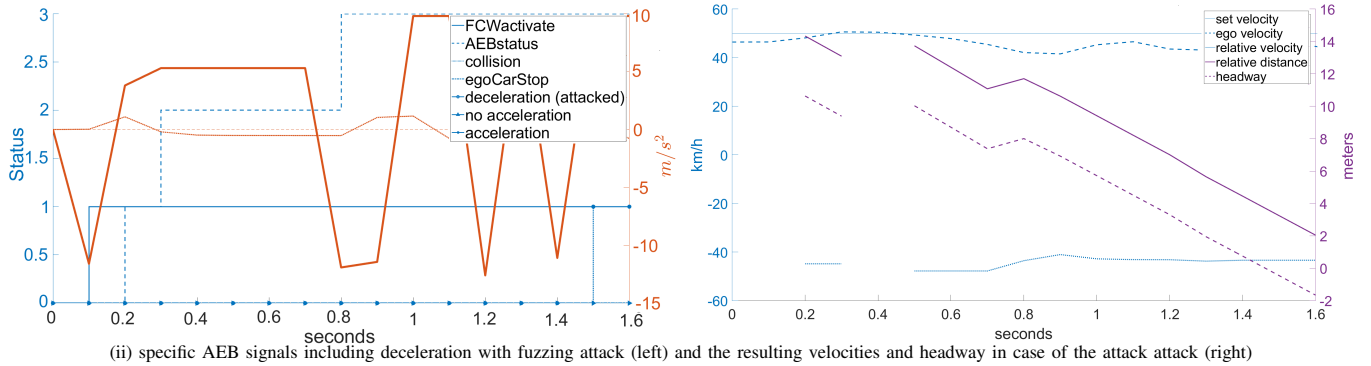
[5]https://www.vector.com/int/en/products/products-a-z/software/canoe/

TABLE III
AEB RESULTS: COLLISION VELOCITY AND DISTANCE TO TARGET AT VARIOUS ATTACK SUCCESS RATES

Signal	Attack	$p = 0.5$		$p = 0.2$		$p = 0.1$		$p = 0.05$	
		Collision velocity[km/h]	Distance to target[m]	Collision velocity[km/h]	Distance to target[m]	Collision velocity[km/h]	Distance to target[m]	Collision velocity[km/h]	Distance to target[m]
A1 (relative distance)	Fuzzing	45.61	0	45.43	0	34.16	0	no coll.	1.14
	Replay	43.44	0	no coll.	1.14	no coll.	1.14	no coll.	1.14
	DoS	no coll.	2.57	no coll.	1.3	no coll.	1.3	n/a	n/a
A2 (relative velocity)	Fuzzing	49.03	0	43.56	0	37.87	0	no coll.	1.14
	Replay	30.45	0	no coll.	1.14	no coll.	1.14	no coll.	1.14
	DoS	17.34	0	no coll.	1.14	no coll.	1.14	n/a	n/a
A3 (Deceleration)	Fuzzing	55.40	0	43.88	0	43.88	0	29.95	0
	Replay	no coll.	0.36	no coll.	0.72	no coll.	0.72	no coll.	0.72
	DoS	15.69	0	no coll.	0.34	no coll.	1.14	n/a	n/a
A4 (Throttle)	Fuzzing	no coll.	2	no coll.	1.77	no coll.	1.77	no coll.	1.73
	Replay	no coll.	0.09	no coll.	1	no coll.	1	no coll.	0.98
	DoS	no coll.	1.15	no coll.	1.14	no coll.	1.14	n/a	n/a
A5 (Long. velocity)	Fuzzing	no coll.	0.46	no coll.	0.15	no coll.	0.15	no coll.	0.86
	Replay	no coll.	0.94	no coll.	1.13	no coll.	1.13	no coll.	1.13
	DoS	no coll.	1.14	no coll.	1.14	no coll.	1.14	n/a	n/a
A6 (Curvature)	Fuzzing	no coll.	0.34	no coll.	0.34	no coll.	0.34	no coll.	1



specific AEB signals including deceleration (left) and the resulting velocities and headway without attack (right)



(ii) specific AEB signals including deceleration with fuzzing attack (left) and the resulting velocities and headway in case of the attack attack (right)

Fig. 10. AEB signals without adversarial intervention (i) and under a fuzzing attack on deceleration (ii)

B. Hazard analysis and risk assessment

1) ACC results interpretation based on existing literature data: We now perform a hazard analysis and risk assessment for the ACC results. In Table IV we classify the severity of impact on the passengers in case of the ACC model based on the injury risk curves from [49]. To determine the impact on injuries, we consider a worst case in which the front vehicle is stationary and recall the reasoning in [49] which uses the change in velocity to quantify the severity injuries. In Table IV we classify the severity of impact on the passengers considering that the lead vehicle is stationary, i.e., $v_{collision} = \frac{v_{ego}}{2}$, where $v_{collision}$ is the collision velocity, v_{ego} is the ego car velocity. In this scenario, a fuzzing attack with attack probabilities $p = 0.2$ and $p = 0.5$ on the ACC system leads to a cumulative probability of fatal/severe injuries in case the attack is on the relative distance, relative velocity, acceleration or curvature. Based on the ISO 26262-3 specification, we conclude that a fuzzing attack with probability $p = 0.2$ and $p = 0.5$ can be assigned an AIS 4 to 6. Therefore, it shall not be acceptable for an IDS to have a false-positive rate larger than 20% when performing in an ACC system.

2) AEB results interpretation based on existing literature data: In Table V we classify the severity of the impact in case of the AEB model based on the injury risk curves from both

[49] and [48]. There are differences between the interpretations of the two works, i.e., [49] quantifies based on the cumulative probability of severity while [48] uses the probability alone (non-cumulative). Thus, there may be slight differences in the interpretations outlined by us in Table V. A fuzzing attack with probability $p = 0.05$ on the AEB system leads to leads to 50% cumulative probability of severe injuries and 70% probability of light injuries in case the attack is on the deceleration. This injuries can be assigned to AIS 1 to 4. The same fuzzing attack when the probability increases to $p = 0.1$ leads to 60% cumulative probability of severe injuries and 60% probability of light injuries for the pedestrian in case of an attack on the relative distance. The cumulative probability becomes 64% for severe injuries or alternatively 50% for light injuries for the pedestrian in case of an attack on the relative velocity. In case of an attack on the deceleration, there is a 77% cumulative probability of severe injuries and a 60% probability of severe injuries for the pedestrian. This allows an assigned to AIS 1 to 4. By further increasing the probability of the fuzzing attack to $p = 0.2$ the results are similar to the previous case for $p = 0.1$ except for the attacks on the relative distance and velocity which now have a severe effect in both interpretations. Fuzzing attacks and replay attacks with probabilities $p = 0.5$ lead to a similar impact as in case of the probability $p = 0.2$ for the attack on the relative distance. Also, for the relative distance the replay attack has the same impact as the fuzzing attack. A fuzzing attack on the relative velocity leads to a 51% cumulative probability of fatal injuries and a 80% probability of severe injuries for the pedestrian. For replay attacks on the relative velocity, the outcome is a 50% cumulative probability of severe injuries and a 95% probability of light injuries for the pedestrian. The DoS attack on the relative velocity leads to a 20% cumulative probability of light injuries and a 90% probability of light injuries for the pedestrian. Fuzzing attack on the deceleration leads to a 63% cumulative probability of fatal injuries and a 80% probability of severe injuries for the pedestrian. Finally, the DoS attack on the relative velocity leads to a 17% cumulative probability of light injuries and a 95% probability of light injuries for the pedestrian.

Due to obvious safety concerns and experimentation costs, it is not possible for us to further validate such attacks on the AEB or ACC systems inside a real car. This is also the case with related works addressing intrusions on the CAN bus which generally use data collected from a car that is augmented with attacks in an off-line manner [14], [20], while only a recently emerged body of works considered attacks on Simulink models [27], [25]. Since the models that we use for vehicle dynamics and the AEB/ACC systems are industry standard, being employed in various works [24], [28], [29], and we considered an architecture from a real-world Hyundai coupe, the results from our work should be as close as possible to the real-world behavior in case when such attacks occur. Of course, no simulation is perfect. But, needless to say, this is precisely the role of a simulation, to provide a first line of evidence for how the system will respond without risking physical damage, while the validity of the results should be within reasonable bounds as long as the models are correct.

TABLE IV
INTERPRETATION OF ACC RESULTS BASED ON EXISTING LITERATURE DATA CONSIDERING THAT THE LEAD VEHICLE IS STATIONARY

Signal	Attack	$p = 0.5$ Severity based on [49]	$p = 0.2$ Severity based on [49]
A1 (relative distance)	Fuzzing	fatal 85%	severe 85%
A2 (relative velocity)	Fuzzing	fatal 87%	fatal 82%
A3 (Acceleration)	Fuzzing	fatal 80%	severe 80%
A5 (Curvature)	Fuzzing	fatal 100%	severe 86%

C. Proposed countermeasures

Our methodology from Section 1 outlines that the scope of the impact assessment is to determine the target rates for the accuracy of an intrusion detection system on specific functionalities/components. For completeness however, we now also discuss several ways to mitigate the effects of the aforementioned attacks. While fault confinement and diagnosis methods exists at the ECU level, these do not respond to the previously analyzed attacks.

One obvious way to decrease the adversary success rate is by increasing the number of legitimate packets. This can be done by increasing the cycle time of specific messages in case that intrusions are detected or by adding additional sources (ECUs or sensors) to increase redundancy in case of corrupted units. However, this may also increase the bus-load and lead to unreasonable costs in bandwidth. For this reason, in what follows we discuss the introduction *plausibility checks* that can be done at the component level, i.e., by the software routine that implements the specific functionality. The main role of the plausibility checks is to filter intrusions, reducing the attack probability, sometimes even completely eliminating the attack, e.g., when the probability of attack is low as in some of our experiments. We consider that *plausibility checks* can be efficiently deployed inside the module/component where the signal is consumed because there is much more control and understanding of the functionality at design time. The consumed signal can be easily checked and in case of conspicuous errors, it can be more easily corrected by using default/neutral values, e.g., old values, zeros, the maximum value, the minimum value, or even another signal that is calculated internally or comes from other source. These checks can be also performed by the IDS, however, if the IDS is done at the CAN communication layer, specific know-how about the functionality that consumes the signal may be missing.

Since only fuzzing attacks had a great impact on the ACC and AEB systems, e.g., causing hazards with a high severity even if the attack probability was low $p = 0.05 - 0.2$, we illustrate such corrections on this type of attack alone. Consequently, we have implemented in Simulink a plausibility check for the signals in order to mitigate hazards. For the ACC model we implement plausibility checks for acceleration, relative distance, velocity and curvature. In case of the AEB model we implement plausibility checks for deceleration, relative distance and velocity. A mandatory validation condition for all signals, is the gradient check, i.e., verify that the absolute value of the difference between two consecutive samples is greater than a predefined threshold. If the gradient is lower

TABLE V
INTERPRETATION OF AEB RESULTS BASED ON EXISTING LITERATURE DATA

Signal	Attack	$p = 0.5$ Severity base on [49]	$p = 0.5$ Severity base on [48]	$p = 0.2$ Severity base on [49]	$p = 0.2$ Severity base on [48]	$p = 0.1$ Severity base on [49]	$p = 0.1$ Severity base on [48]	$p = 0.05$ Severity base on [49]	$p = 0.05$ Severity base on [48]
A1 (relative distance)	Fuzzing	severe 80%	severe 70%	severe 80%	severe 70%	severe 60%	light 60%	no coll.	no coll.
	Replay	severe 80%	severe 70%	no coll.	no coll.	no coll.	no coll.	no coll.	no coll.
A2 (relative velocity)	Fuzzing	fatal 51%	severe 80%	severe 77%	severe 60%	severe 64%	light 50%	no coll.	no coll.
	Replay	severe 50%	light 95%	no coll.	no coll.	no coll.	no coll.	no coll.	no coll.
	DoS	light 20%	light 90%	no coll.	no coll.	no coll.	no coll.	n/a	n/a
A3 (Deceleration)	Fuzzing	fatal 63%	severe 80%	severe 77%	severe 60%	severe 77%	severe 60%	severe 50%	light 70%
	DoS	light 17%	light 95%	no coll.	no coll.	no coll.	no coll.	n/a	n/a

TABLE VI
ACC RESULTS: INJURY REDUCTION IN CASE OF FUZZING ATTACKS USING THE PLAUSIBILITY CHECK OF THE RECEIVED SIGNALS

Signal	$p = 0.5$ Collision velocity[km/h]	$p = 0.5$ Distance to target[m]	$p = 0.2$ Collision velocity[km/h]	$p = 0.2$ Distance to target[m]
A1 (rel. distance)	no coll. reduction: fatal → none	9.68	no coll. reduction: severe → none	8.09
A2 (rel. velocity)	no coll. reduction: fatal → none	17.63	no coll. reduction: fatal → none	7.85
A3 (accel.)	no coll. reduction: fatal → none	9.68	no coll. reduction: severe → none	7.35
A5 (curvature)	no coll. reduction: fatal → none	10.56	no coll. reduction: severe → none	10.56

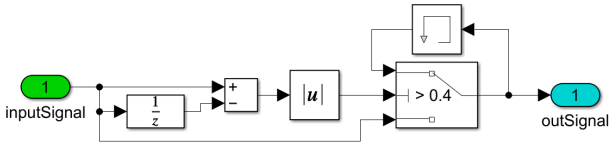


Fig. 11. Plausibility checks on acceleration, deceleration and curvature

than the threshold, we consider that the current value of the signal is legitimate. Otherwise, we consider that the current value of the signal may be corrupted and in this case we freeze the signal. The improvements on the severity level are highlighted in Tables VI and VII where we show the injury reduction after the plausibility checks are implemented. We detail all these with respect to each signal in what follows.

1) Plausibility checks on the acceleration: The plausibility check for the acceleration is outlined in Figure 11. In case of the ACC model, our simulation results shows that checking the gradient alone with a 0.4 threshold is sufficient to avoid the collisions in case of the fuzzing attacks with probabilities $p = 0.5$ and $p = 0.2$. The minimum distance to the target is 9.68m in case of $p = 0.5$ and 7.35m in case of $p = 0.2$. It means a reduction from fatal/severe injuries to none.

2) Plausibility checks on the brake signal: For the brake signal identical checks to the case of acceleration are sufficient since the brake signal for the AEB has a similar role to the acceleration signal for the ACC. Consequently, the Simulink design for plausibility checks on deceleration is identical to the previous one and we omit it in the figure. By using a gradient check with a 0.4 threshold, the collisions are avoided in case of the fuzzing attacks for all the previous attack probabilities: $p = 0.5$, $p = 0.2$, $p = 0.1$ and $p = 0.05$. This means a reduction

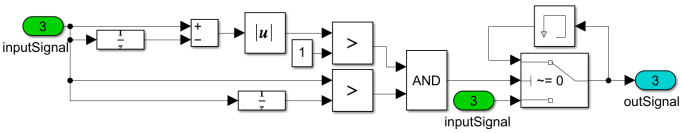


Fig. 12. Plausibility checks on relative distance and velocity for ACC

from fatal/severe injuries to none or from severe/light to none.

3) Plausibility checks on the relative distance and velocity for ACC model: We depict the Simulink design for relative distance and velocity in Figure 12. The relative distance and velocity are the main inputs for the ACC and the AEB systems which will trigger the brake or deceleration when these signals are decreasing, i.e., this means that the ego vehicle is approaching an obstacle. Because of this, gradient checks alone are no longer sufficient to filter the attacks. We modify the plausibility checks as follows. If the current value of the signal is greater than the previous value and the gradient larger than a predefined value, i.e., 1 in our case, then it may be that the current value of the signal is corrupted and in this case we use its old value. Otherwise, we consider that the current value of the signal is the legitimate one. In case of the ACC model, for the relative distance and velocity signals, our simulations show that these plausibility checks can avoid the collisions in case of the fuzzing attacks with probabilities $p = 0.5$ and $p = 0.2$. The minimum distance to the target is 9.68m for the attack on the relative distance and 17.63m for the attack on the relative velocity, in case when $p = 0.5$ and 8.09 m for the attack on the relative distance and 7.85m for the attack on the relative velocity in case of $p = 0.2$. It means a reduction from fatal/severe injuries to none.

4) Plausibility checks on the relative distance and velocity for AEB model: Since in case of the AEB model the relative distance and velocity can be discontinuous, i.e., the obstacle may go undetected for a short period of time, we add a supplementary condition that checks if the signal is finite. Then for the plausibility check of the relative distance and velocity from the AEB model we use 3 conditions: i) check if the signal is finite, ii) gradient check with a 1 threshold iii) check that the current value of the signal is larger than the previous value. We depict the Simulink design for relative distance and velocity in Figure 13. For the AEB relative distance and velocity signals, our simulation shows that these checks will avoid the collisions in case of the fuzzing attacks with low probabilities, i.e., $p = 0.1$ and $p = 0.05$. For an

TABLE VII
AEB RESULTS: INJURY REDUCTION IN CASE OF FUZZING ATTACKS USING THE PLAUSIBILITY CHECK OF THE RECEIVED SIGNALS

Signal	$p = 0.5$ Collision velocity[km/h]	$p = 0.5$ Distance to target[m]	$p = 0.2$ Collision velocity[km/h]	$p = 0.2$ Distance to target[m]	$p = 0.1$ Collision velocity[km/h]	$p = 0.1$ Distance to target[m]	$p = 0.05$ Collision velocity[km/h]	$p = 0.05$ Distance to target[m]
A1 (relative distance)	44.31	0	30.56	0	no coll.	0.85	no coll.	1.14
	no injury reduction		reduction: severe → severe/light		reduction: severe/light → none		reduction: n/a	
A2 (relative velocity)	49.03	0	no coll.	0.85	no coll.	0.85	no coll.	1.14
	no injury reduction		reduction: severe → none		reduction: severe/light → none		reduction: n/a	
A3 (Deceleration)	no coll.	0.22	no coll.	0.22	no coll.	0.22	no coll.	0.47
	reduction: fatal/severe → none		reduction: severe → none		reduction: severe → none		reduction: severe/light → none	

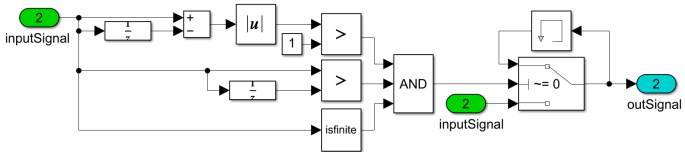


Fig. 13. Plausibility checks on relative distance and velocity for AEB

attack probability of $p = 0.2$, the collision is avoided only for the attack on the relative velocity, while for the attack on the relative distance the collision occurs but the velocity is reduced from 45.43km/h to 30.56km/h which means a reduction from severe to severe/light injuries. For the attack probability $p = 0.5$, no significant improvement was obtained.

5) Plausibility checks on the curvature for ACC model: For the curvature, identical checks to the case of acceleration and brake are sufficient since the curvature, in our model is a constant value. Consequently, the Simulink design for plausibility checks on curvature is identical to the plausibility checks for acceleration and brake and we omit it in the figure. By using a gradient check with a 0 threshold, the collisions are avoided in case of the fuzzing attacks for the $p = 0.5$ and $p = 0.2$ attack probabilities. This means a reduction from fatal injuries to none.

In general, the plausibility checks from the model set room for a reduction of injuries, e.g., even from fatal/severe down to none. However, if the attack probability was too high, e.g., $p = 0.5$ and the attack was performed on the relative distance and/or velocity, the impact could not be reduced which suggests that a $p = 0.5$ in detecting intrusions is not workable for AEB systems (reductions to severe/light or none do happen for $p = 0.2$). These plausibility checks should not influence the signals if the threshold is correctly chosen and the adversary is not present on the bus, i.e., no attacks are detected and the signals are correct. If the threshold is incorrect, delays of one recurrence may occur, i.e., 100ms in our case, but these delays do not significantly impact the system.

VII. CONCLUSION

Model-based design is commonly used in the automotive industry to save time and costs as it allows simple code reuse and generation as well as real-time simulations [50]. This approach is also recommended for safety components [51] and, more recently, for ADAS components [52]. Our work suggests a V-cycle inspired methodology based on such models that may be useful when designing intrusion detection systems for in-vehicle networks. In the light of this methodology we showed how the impact of cyber-attacks can be assessed on two safety-critical, wide spread car components: the ACC and AEB. Notably, cyber-attacks can inflict from light to severe or even fatal injuries. Our results show that false negatives of 5% may lead to severe injuries in case of fuzzing attacks, while for replays 40% false negatives will have similar consequences. In case of DoS attacks, a duration longer than $500ms$ will also cause a collision which leads at least to light injuries. This suggests a 97.5% success rate to be acceptable for detecting fuzzing attacks and an acceptable communication loss below $500ms$. Nonetheless we also discuss countermeasures which for our models can significantly reduce the risks of injuries. Our work is the first to discuss the relation between accuracy levels for intrusion detection and their impact on safety and we hope that more works will follow in this direction.

REFERENCES

[1] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham *et al.*, "Experimental security analysis of a modern automobile," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 447–462.

[2] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX Security Symposium*. San Francisco, 2011.

[3] C. Miller and C. Valasek, "A survey of remote automotive attack surfaces," *Black Hat USA*, pp. 1—-90, 2014.

[4] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[5] AUTOSAR, *Specification of secure onboard communication*, 2017.

[6] ——, *Specification of Intrusion Detection System Protocol*, 2020.

[7] *ISO/SAE 21434: Road vehicles—Cybersecurity engineering*, 1st ed., ISO/SAE, 2020.

[8] UNECE, *Addendum 154 – UN Regulation No. 155: Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system*. United Nations, 2021.

[9] *SAE J3016 Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*. SAE International, 2016.

[10] B. Ondrej, D. Johannes, and S. Jan Paul, "Automotive software and electronics 2030," McKinsey & Company, Atlanta, GA, USA, 2019.

[11] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," *PloS one*, vol. 11, no. 6, p. e0155781, 2016.

[12] P. Borazjani, C. Everett, and D. McCoy, "Octane: An extensible open source car security testbed," in *Proceedings of the Embedded Security in Cars Conference*, 2014.

[13] B. Groza and P.-S. Murvay, "Efficient intrusion detection with bloom filtering in controller area networks," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 4, pp. 1037–1051, 2018.

[14] C. Jichici, B. Groza, and P.-S. Murvay, "Integrating adversary models and intrusion detection systems for in-vehicle networks in canoe," in *Intl. Conf. on Info. Tech. and Com. Security*. Springer, 2019, pp. 241–256.

[15] Y. Yang, L. Wang, Z. Li, P. Shen, X. Guan, and W. Xia, "Anomaly detection for controller area network in braking control system with dynamic ensemble selection," *IEEE Access*, vol. 7, pp. 95 418–95 429, 2019.

[16] M. Marchetti, D. Stabili, A. Guido, and M. Colajanni, "Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms," in *Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*. IEEE, 2016, pp. 1–6.

[17] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *Intelligent Veh Symposium*. IEEE, 2011, pp. 1110–1115.

[18] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45 233–45 245, 2018.

[19] C. Wang, Z. Zhao, L. Gong, L. Zhu, Z. Liu, and X. Cheng, "A distributed anomaly detection system for in-vehicle network using htm," *IEEE Access*, vol. 6, pp. 9091–9098, 2018.

[20] H. M. Song and H. K. Kim, "Self-supervised anomaly detection for in-vehicle network using noised pseudo normal data," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1098–1108, 2021.

[21] W. Choi, K. Joo, H. J. Jo, M. C. Park, and D. H. Lee, "Voltageids: Low-level communication characteristics for automotive intrusion detection system," *IEEE Tran. on Information Forensics and Security*, 2018.

[22] J. Sun, Y. Cao, Q. A. Chen, and Z. M. Mao, "Towards robust lidar-based perception in autonomous driving: General black-box adversarial sensor attack and countermeasures," in *29th USENIX Security Symposium*, 2020, pp. 877–894.

[23] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[24] S. Gautham, A. Rajagopala, A. V. Jayakumar, C. Deloglos, E. Karincic, and C. Elks, "Heterogeneous runtime verification of safety critical cyber physical systems," *arXiv preprint arXiv:2009.09533*, 2020.

[25] S. Gautham, A. V. Jayakumar, and C. Elks, "Multilevel runtime security and safety monitoring for cyber physical systems using model-based engineering," in *International Conference on Computer Safety, Reliability, and Security*. Springer, 2020, pp. 193–204.

[26] L. Kang and H. Shen, "Attack detection and mitigation for sensor and can bus attacks in vehicle anti-lock braking systems," in *2020 29th International Conference on Computer Communications and Networks (ICCCN)*. IEEE, 2020, pp. 1–9.

[27] P. Kapoor, A. Vora, and K.-D. Kang, "Detecting and mitigating spoofing attack against an automotive radar," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–6.

[28] A. Alipour-Fanid, M. Dabaghchian, and K. Zeng, "Impact of jamming attacks on vehicular cooperative adaptive cruise control systems," *IEEE trans. on vehicular technology*, vol. 69, no. 11, pp. 12 679–12 693, 2020.

[29] W. Jeon, Z. Xie, A. Zemouche, and R. Rajamani, "Simultaneous cyber-attack detection and radar sensor health monitoring in connected acc vehicles," *IEEE Sensors Journal*, 2020.

[30] F. Alotibi and M. Abdelhakim, "Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model," *IEEE Trans. on Intelligent Transp. Systems*, 2020.

[31] A. Munir and F. Koushanfar, "Design and analysis of secure and dependable automotive cps: A steer-by-wire case study," *IEEE Transactions on Dependable and Secure Computing*, 2018.

[32] X. Xu, X. Li, P. Dong, Y. Liu, and H. Zhang, "Robust reset speed synchronization control for an integrated motor-transmission powertrain system of a connected vehicle under a replay attack," *IEEE Transactions on Vehicular Technology*, 2020.

[33] L. Guo, B. Yang, and J. Ye, "Enhanced cyber-physical security of steering stability control system for four-wheel independent drive electric vehicles," in *2020 IEEE Transportation Electrification Conference & Expo (ITEC)*. IEEE, 2020, pp. 1240–1245.

[34] V. K. Kukkala, S. Pasricha, and T. Bradley, "Sedan: Security-aware design of time-critical automotive networks," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 8, pp. 9017–9030, 2020.

[35] M. Pirani, S. Baldi, and K. H. Johansson, "Impact of network topology on the resilience of vehicle platoons," *IEEE Transactions on Intelligent Transportation Systems*, 2022.

[36] M. H. Basiri, M. Pirani, N. L. Azad, and S. Fischmeister, "Security of vehicle platooning: A game-theoretic approach," *IEEE Access*, vol. 7, pp. 185 565–185 579, 2019.

[37] E. Mousavinejad, F. Yang, Q.-L. Han, X. Ge, and L. Vlacic, "Distributed cyber attacks detection and recovery mechanism for vehicle platooning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3821–3834, 2019.

[38] S. Xiao, X. Ge, Q.-L. Han, and Y. Zhang, "Secure distributed adaptive platooning control of automated vehicles over vehicular ad-hoc networks under denial-of-service attacks," *IEEE Trans. on Cybernetics*, 2021.

[39] "Test protocol – aeb vru systems, version 3.0.3," in *Vulnerable Road User (VRU) Protection*. Euro NCAP, June 2020.

[40] V. Bandur, G. Selim, V. Pantelic, and M. Lawford, "Making the case for centralized automotive e/e architectures," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 2, pp. 1230–1245, 2021.

[41] B. Groza, L. Popa, P.-S. Murvay, Y. Elovici, and A. Shabtai, "CANARY-a reactive defense mechanism for controller area networks based on active relays," in *30th USENIX Security Symposium*, 2021.

[42] A.-M. Jamil, L. Ben Othmane, and A. Valani, "Threat modeling of cyber-physical systems in practice," in *International Conference on Risks and Security of Internet and Systems*. Springer, 2022, pp. 3–19.

[43] A. Chattopadhyay, K.-Y. Lam, and Y. Tavva, "Autonomous vehicle: Security by design," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, pp. 7015–7029, 2020.

[44] *ISO 26262-3:2018 Road vehicles — Functional safety — Part 3: Concept phase*, 2nd ed., 2018.

[45] M. Peden *et al.*, *World report on road traffic injury prevention*. World Health Organization, 2004.

[46] H. R. Kröyer, "Is 30 km/ha 'safe'speed? injury severity of pedestrians struck by a vehicle and the relation to travel speed and age," *IATSS research*, vol. 39, no. 1, pp. 42–50, 2015.

[47] D. McKibbin, *The role of speed in the frequency and severity of Road Traffic Collisions*. Northern Ireland Assembly, 2013, vol. NIAR 171-14.

[48] G. A. Davis, "Relating severity of pedestrian injury to impact speed in vehicle-pedestrian crashes: Simple threshold model," *Transportation research record*, vol. 1773, no. 1, pp. 108–113, 2001.

[49] D. Richards, *Relationship between speed and risk of fatal injury: pedestrians and car occupants*. Department for Transport, UK, 2010.

[50] *Why Adopt Model-Based Design?l*, MathWorks, 2021.

[51] I. Fey, J. Müller, and M. Conrad, "Model-based design for safety-related applications," SAE Technical Paper, Tech. Rep., 2008.

[52] "Siemens broadens mbse to engineer beyond individual autonomous vehicles," CIMdata, Tech. Rep., 2021.

Adriana Berdich received her Engineer and Ph.D title from Politehnica University of Timisoara (UPT) in 2017 and 2024. She has a 9-year background in the automotive industry as a Function Developer with main focus on torque structure and vehicle motion functions, through all phases of the V-model. Currently she is Technical Project Manager for several Selective Catalytic Reduction (SCR) projects. She was also a research student in the PRESENCE project (2019-2020) focusing on environment-based device association inside cars.

Bogdan Groza is Professor at Politehnica University of Timisoara (UPT). He received his Dipl.Ing. and Ph.D. degree from UPT in 2004 and 2008 respectively. In 2016 he successfully defended his habilitation thesis having as core subject the design of cryptographic security for automotive embedded networks.Besides regular participation in national and international research projects in information security, he lead the CSEAMAN (2015-2017) and PRESENCE (2018-2020) projects, two national research programs dedicated to in-vehicle security.