V2C: A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems

Citation for the original published paper (version of record):

Rosenstatter, T., Olovsson, T., Almgren, M. (2021)
V2C: A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems
Proceedings of The 16th International Conference on Availability, Reliability and Security (ARES 2021):

(article starts on next page)

V2C: A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems

Thomas Rosenstatter
thomas.rosenstatter@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

Tomas Olovsson
tomas.olovsson@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

Magnus Almgren
magnus.almgren@chalmers.se
Chalmers University of Technology
Gothenburg, Sweden

ABSTRACT

Vehicles have become connected in many ways. They communicate with the cloud and will use Vehicle-to-Everything (V2X) communication to exchange warning messages and perform cooperative actions such as platooning. Vehicles have already been attacked and will become even more attractive targets due to their increasing connectivity, the amount of data they produce and their importance to our society. It is therefore crucial to provide cyber security measures to prevent and limit the impact of attacks.

As it is problematic for a vehicle to reliably assess its own state when it is compromised, we investigate how vehicle trust can be used to identify compromised vehicles and how fleet-wide attacks can be detected at an early stage using cloud data. In our proposed V2C Anomaly Detection framework, peer vehicles assess each other based on their perceived behavior in traffic and V2X-enabled interactions, and upload these assessments to the cloud for analysis. This framework consists of four modules. For each module we define functional demands, interfaces and evaluate solutions proposed in literature allowing manufacturers and fleet owners to choose appropriate techniques. We detail attack scenarios where this type of framework is particularly useful in detecting and identifying potential attacks and failing software and hardware. Furthermore, we describe what basic vehicle data the cloud analysis can be based upon.

CCS CONCEPTS

• **Security and privacy → Intrusion detection systems**; • **Computer systems organization → Embedded systems**.

KEYWORDS

cyber-physical systems, resilience, security, intrusion detection, cloud, vehicular systems, automotive systems

ACM Reference Format:
Thomas Rosenstatter, Tomas Olovsson, and Magnus Almgren. 2021. V2C: A Trust-Based Vehicle to Cloud Anomaly Detection Framework for Automotive Systems. In *The 16th International Conference on Availability, Reliability and Security (ARES 2021), August 17–20, 2021, Vienna, Austria.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3465481.3465750

1 INTRODUCTION

To increase traffic safety and efficiency, Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication is needed to enable the sensing of objects and entities that are out of the vehicle's sight; to allow cooperation between vehicles to increase efficiency such as in platooning [8] and to receive warnings about road conditions, e. g., slippery roads, traffic accidents and road works ahead. In addition, most vehicles also have cellular access to the Internet to provide comfort functions such as remote unlock/lock and to receive essential software updates over the air.

The feasibility of cyber attacks against vehicles that can potentially lead to catastrophic events was demonstrated already in 2011. Checkoway et al. [6] analyzed a modern vehicle for security vulnerabilities and identified several potential attacks requiring physical access, but also attacks possible through short-range wireless connectivity, e. g., Bluetooth, and long-range wireless connectivity such as cellular networks. In 2015 Miller and Valasek [26] demonstrated a remote exploit through the cellular network allowing them to remotely control a vehicle including safety critical functions. A more recent report from 2020 revealed several vulnerabilities allowing remote control of Mercedes-Benz vehicles [39]. Remote attacks pose a significantly higher risk to the safety of road users as they do not require physical access to vehicles and can be performed from anywhere without leaving significant traces.

Security measures have been investigated and proposed for various parts of the automotive system, ranging from intrusion detection for the in-vehicle network (e. g., [27, 29]), secure communication for all networked communications and secure boot for Electronic Control Units, ECUs (e. g., [33]). These methods are important to protect and secure an individual vehicle and its occupants, however, they are not foolproof nor enough to identify wide-spread attacks and anomalies on a fleet level.

Motivation. Analyzing all available data in the cloud when an event from a single vehicle is received is not feasible due to the potentially large number of vehicles in the fleet and the associated computational costs. In general, it is also problematic for a system, i.e., the vehicle such as a passenger car, truck or bus, to reliably assess its own state when it is compromised. Therefore, a framework that allows finding anomalies and intrusions in automotive systems in its entirety is needed. The establishment of *trust* between vehicles interacting via V2V communication is of particular interest as it enables the assessment of a vehicle's behavior, especially in safety-critical situations, not just by the vehicle itself but also by the surrounding vehicles.

Contributions. In this paper, we present *V2C Anomaly Detection*, an overall framework that combines the detection of anomalies by evaluating V2V interactions using *trust scores* with the analysis

ARES 2021, August 17–20, 2021, Vienna, Austria

in the cloud. We divide this framework in modules and for each module, we survey existing solutions, investigate how to implement them, propose modifications when necessary and evaluate them. During the design we put special focus on scalability of the framework and the feasibility to detect attacks and abnormal behavior. We further argue that a framework like *V2C Anomaly Detection* is necessary and complements existing security controls and internal Intrusion Detection Systems (IDSs), as it is based on vehicles independently evaluating each other rather than only evaluating their own internal state.

After giving an overview of the system design and attacks in Section 2, we present related work in Section 3, review existing methods for each module in Section 4, and evaluate suitable methods and, when necessary, suggest modifications in Section 5. Ultimately, we show how to apply and discuss the *V2C Anomaly Detection* framework based on a detailed use case in Section 6 and then conclude the paper in Section 7.

2 OVERVIEW

Our goal is to utilize techniques in both individual vehicles and in the cloud in order to identify compromised vehicles. Security systems inside these compromised or malfunctioning vehicles may not be able to detect the misbehavior themselves and therefore a peer evaluation of independent systems is necessary. In addition, the analysis in the cloud with access to data about each vehicle in a fleet allows an early detection of large-scale attacks.

We first discuss the security threats and attack scenarios that the proposed framework should be able to detect in Section 2.1. An overview of the structure and the modules of the *V2C Anomaly Detection* framework using V2V and cloud data is presented in Section 2.2. Thereafter, an adversary model and assumptions are presented in Section 2.3.

2.1 Attack Scenarios

The following three attack scenarios highlight the need for trust-based anomaly detection in the cloud: (1) when firmware is successfully modified without triggering an alert from other monitoring software in the vehicle; (2) when hardware/software failures occur that cause an incorrect perception or behavior; and (3) when failed updates or faulty software cause functional disturbances. These scenarios and the examples listed in Table 1 emphasize that the *V2C Anomaly Detection* framework focuses not only on detecting intrusions caused by unauthorized entities, it also aims at detecting behavioral changes caused by intentional and unintentional, yet authorized, modifications as well as random faults. These scenarios will be further used in Section 5.1 to evaluate the applicability of proposed trust and reputation models.

Table 1: Examples for the identified scenarios.

Ex. #	Description
Scenario 1 – Unauthorized firmware manipulation	
Ex.1	Manipulation of the firmware such that the owner is able to send traffic congestion warning messages at will to reduce traffic on a desired road segment.
Ex.2	Manipulation of the firmware such that the automated vehicle drives faster than the current speed limit.
Ex.3	Manipulation of the firmware such that an attacker can disrupt traffic by suppressing relaying of messages, spoofing warning messages or flooding the VANET with erroneous messages.
Scenario 2 – HW/SW failures	
Ex.4	A lidar sensor or camera is experiencing a fault and the vehicle is thus not able to perceive its environment properly.
Scenario 3 – Legitimate SW/HW updates	
Ex.5	After a legitimate firmware update, the automated vehicle drives too fast in certain road conditions, e. g., when the road is slippery, since the vehicle perceives the current driving conditions incorrectly.
Ex.6	A defect hardware component is replaced in an authorized workshop and causes compatibility problems resulting in a misbehavior while driving.
Ex.7	The machine learning algorithm for identifying traffic signs has been updated and now causes a misclassification of speed limit signs. The possibility of attacks exploiting machine learning algorithms for the identification of traffic signs has also been demonstrated by Sitawarin et al. [37]. Thus, an update of such systems may even open new attack vectors.

Scenario 1 – Unauthorized Firmware manipulation. Someone such as the owner, an employee at a workshop or a remote attacker, performs unauthorized modifications of the firmware in order to (i) suppress relaying of messages in the Vehicular Ad-hoc NETwork, VANET (blackhole attack), (ii) modify or falsify warning messages for events such as traffic accidents, traffic jams and other road conditions (masquerading attack) or (iii) interact with other vehicles during cooperative scenarios, e. g., platooning, in a selfish or malicious way to cause disruption of the traffic flow or even an accident.

Scenario 2 – HW/SW failures. Hardware or software faults occur and cause the vehicle to react improperly to the situation.

Scenario 3 – Legitimate SW/HW updates. The vehicle manufacturer pushes an over-the-air update for one of the computing units, ECUs, in a vehicle or an authorized workshop replaces hardware and causes a degraded or unintended functionality due to the complexity of an automotive system and lack of sufficient testing. Examples of such unintended functionality are inaccurate or wrong sensor readings and changes in the behavior when interacting with other vehicles.

2.2 System Design

The goal of our *V2C Anomaly Detection* framework is to identify anomalous and malicious behavior in a fleet by having vehicles evaluate each other. As a result, we identified four tasks: (i) individually assess vehicle trust which consequently has to indicate anomalous behavior; (ii) combine these trust evaluations; (iii) detect a change in the combined trust evaluations showing that the vehicle's behavior negatively changed; and (iv) analyzing cloud data to identify similar patterns in the fleet. Each task is assigned to a module as shown in Figure 1. *Module 1* evaluates its own and the behavior of other vehicles based on V2V data and the cooperation

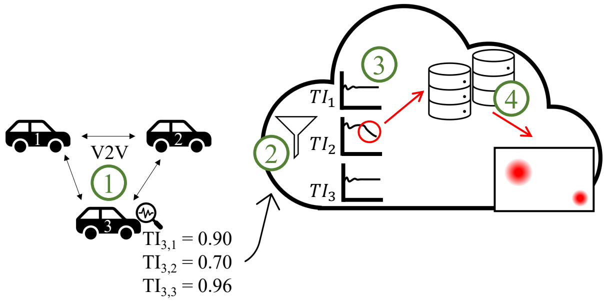

Figure 1: Structure of *V2C Anomaly Detection*.

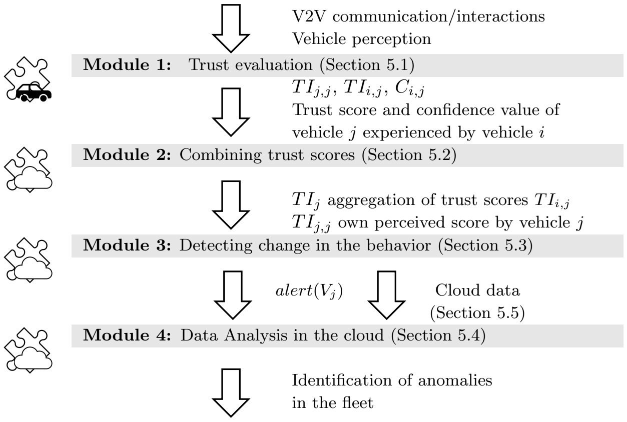
V2V communication/interactions
Vehicle perception
Module 1: Trust evaluation (Section 5.1)
$TI_{j,j}$, $TI_{i,j}$, $C_{i,j}$
Trust score and confidence value of vehicle j experienced by vehicle i
Module 2: Combining trust scores (Section 5.2)
TI_j aggregation of trust scores $TI_{i,j}$
$TI_{j,j}$ own perceived score by vehicle j
Module 3: Detecting change in the behavior (Section 5.3)
$alert(V_j)$
Cloud data (Section 5.5)
Module 4: Data Analysis in the cloud (Section 5.4)
Identification of anomalies in the fleet

Figure 2: Modules of the *V2C Anomaly Detection* framework including their inputs and outputs they produce.

with them. Each vehicle performs this individual peer evaluation which results in a *trust score* and uploads the average of the trust scores for a certain period, e. g., per day, for each evaluated vehicle to the cloud. These trust scores reported by the vehicle fleet are combined to one trust score per vehicle in *module 2*. Combining the trust scores in the cloud has the advantages that no additional computation for combining trust scores is required by the vehicles and that no additional messages need to be exchanged in the VANET. *Module 3* observes the trust scores over time and raises an alert when the trust score of a specific vehicle decreases, indicating that the behavior of the vehicle has negatively changed. *Module 4* then analyzes the available data about the affected vehicle in the cloud to find the cause and allow an early detection in the fleet.

Since this is a framework providing flexibility for selecting and adapting suitable techniques for each module, we provide examples and show how the *V2C Anomaly Detection* framework can be used. Figure 2 shows the required information as input, the output each module produces and links to the respective sections where this is further discussed.

2.3 Adversary Model and Assumptions

Attackers may gain complete control of a vehicle, e. g., through attack scenarios described in [14], allowing them to remotely control it or alter the firmware of important ECUs to perform attacks as mentioned in Section 2.1. It is reasonable to assume that many attacks can be perform stealthy and will only be detected through the behavior of the vehicle.

We assume that the majority of vehicles are honest and report correct trust scores, however, a minority of compromised vehicles may collude. The trust score calculated from the data and behavior involving V2V communication can also be correlated to the data about the vehicles located in the cloud.

3 RELATED WORK

In this section we list related work in regard to this framework, while Section 4 reviews individual methods for each module. The introduction of VANETs and subsequent applications introduce additional security and safety challenges. Cryptographic solutions build the base for secure communication to provide confidentiality, integrity and availability. Nevertheless, dishonest or compromised vehicles need to be considered as well. VANET-specific attacks include spoofing locations of vehicles, sending altered or dropping warning messages and dropping all messages.

Trust and reputation models for evaluating vehicles based on V2V interactions have been proposed in literature [20] to identify vehicles that provide false or incorrect information to others and thus risking the safety of the passengers. Proposed solutions focus on how to evaluate the correctness of reported events, how to verify the correctness of sensor data, how to assess the interaction during cooperative events such as platooning and/or how to distribute this knowledge between the vehicles. The resulting evaluation, a trust or reputation score, is used for decision-making by the automated vehicle. Examples are decisions about whether to trust reported warnings, e. g., road accident ahead, received from a particular vehicle. Guo et al. [17] propose an approach that verifies information received via V2V in form of an IDS whose results are further used for decision-making. Other solutions on collaborative intrusion detection [28, 35] consider only packet headers and parameters, such as packet drop rate and transfer delay, and investigate ways to exchange this information between neighboring vehicles.

To the best of our knowledge there is no similar work to the *V2C Anomaly Detection* framework, which deals with how trust scores can be utilized in a more holistic setting in order to detect large-scale attacks and anomalies by uploading and performing cloud-based analyses.

4 EXISTING METHODS

The following sections survey existing methods relevant for each module. In Section 5 we discuss why and how these methods can be applied and motivations when modifications are necessary.

4.1 Trust and Reputation Models

The main task for module 1 is to evaluate vehicle behavior based on V2V data received and ways to assess cooperative tasks performed together. We use a recent survey on trust solutions for VANETs, namely Hussain et al. [20], as a base and further perform a search on *Google Scholar* to find additional trust and reputation-based models. We briefly present a diverse set of solutions to give an overview of methods found in literature, but refer to Hussain et al. [20] for a more complete overview of solutions dealing with trust or reputation in VANETs.

The reputation model presented by Engoulou et al. [11] verifies the vehicle id, width and length of the vehicle, driving direction, location, speed, acceleration, transmission rate and message frequency. Each of these parameters is either correct (1) or false (0) and the resulting sum is the proposed reputation score. This model lacks the evaluation of the other vehicles' cooperative behavior, for example, if the vehicles provide correct warning messages and how they behave during cooperative scenarios such as platooning.

Soleymani et al. [38] make use of fuzzy logic to model trust. Three modules, such as an experience module which evaluates the interactions, a plausibility module which verifies the location of the sender, and an accuracy level module, result in one of the three levels: *high*, *medium* or *low*. A fuzzy inference engine further defines the combinations in which the trust level is *acceptable* and *not-acceptable*. The approach of using fuzzy logic is interesting, however, for our proposed anomaly detection framework we require a more detailed representation of trust/reputation.

A trust system focusing on modeling the trust in the surrounding vehicles as well as the ego vehicle is presented by Rosenstatter and Englund [32]. The authors identify trust evaluation criteria for the own (ego) vehicle, all surrounding vehicles including more specific criteria for the vehicle in front, as this vehicle can be additionally verified with the vehicle's front sensors. Unlike other proposed solutions, this system has been evaluated with data from a real environment consisting of several Vehicle-to-Everything (V2X)-enabled vehicles. In addition, the presented trust model is flexible for adaptations, which is also shown by the authors' detailed discussions of factors that can be considered for evaluating the behavior of a vehicle.

Bißmeyer et al. [4] propose a trust score based on plausibility checks, e. g., reported location, using particle filters. An aging factor is introduced to define ratio of the impact of a new trust score compared to the past trust scores. The authors also propose to apply an additional particle filter to the ego vehicle to verify the trust in the own system as the ego vehicle acts as the reference for evaluating the other vehicles. An extension of the proposed model would be necessary to cover also an evaluation of the behavior of other vehicles during cooperative scenarios such as platooning.

In Section 5.1 we return to trust models and give more details on the demands on the trust models and how they can be applied in the context of the *V2C Anomaly Detection* framework.

4.2 Combining Trust Scores

The combination of knowledge, i. e., the trust scores $TI_{i,j}$, reported by individual vehicles can be carried out in various ways by module 2. Overall, there are two distinct concepts for combining trust scores, (i) calculating the mean, e. g., weighted, arithmetic or geometric, and (ii) using Dempster-Shafer theory (DST) of evidence [34], more specifically, Dempster's rule of combination.

Arithmetic and Geometric Mean. The trust scores reported by each vehicle need to be combined in such a way that it compensates for a minority of dishonest vehicles. The arithmetic mean is generally used for data with no significant outliers whereas the geometric mean is used when the difference between the data points is logarithmic.

The arithmetic mean, for instance, is used by Engoulou et al. [11] to calculate the indirect trust, an aggregation of the trust scores of vehicle j received from other vehicles. Furthermore, the authors use a weighted average for combining the local trust, calculated by the own system for vehicle j and the indirect trust. Rosenstatter and Englund [32] also use the weighted average for the calculation of the combined trust score reflecting the current situation.

Halabi and Zulkernine [18] propose a cooperative game model for preventing malicious vehicles from entering a vehicle coalition. The authors argue for using the product of the trust scores ($TI_{i,j}$) when computing the trust in a vehicle group to increase the impact of low trust scores. A model using the geometric mean is also presented by Kerrache et al. [22]. The described trust model evaluates only binary events and combines the received trust ratings using geometric mean, however, the authors do not provide any reasoning for choosing the geometric mean over the arithmetic mean.

Dempster-Shafer Theory. DST is used in situations which require the combination of evidence reported by several (unreliable) observers. Chen and Venkataramanan [7] present how DST can be applied in the context of intrusion detection in VANETs. The authors argue that Bayesian inference is less suited for such a use as it requires the complete knowledge of prior probabilities, which often need to be estimated in practice. DST, however, handles the lack of a complete probabilistic model by introducing belief and plausibility instead of probabilities. Chen and Venkataramanan highlight the difference with the following example: Node A is trustworthy with a probability of 0.8 respectively untrustworthy with probability 0.2 and reports that node S is trustworthy. In the event that A is indeed trustworthy, the claim that S is trustworthy is accurate, but A being not trustworthy, does not automatically imply that A is inaccurate – it says that the claim of A has 0.8 degrees of belief for S being trustworthy and 0 (not 0.2) degrees of belief that S is untrustworthy [7].

Combining trust or reputation scores reported by several vehicles using DST has been proposed in several models. For instance, Zhang et al. [40] use the computed reputation value per vehicle as degree of belief for either trusting or distrusting a vehicle. A unit in the cloud, a so-called trust authority, combines these reports according to Dempster's rule of combination.

A similar approach as proposed by Chen and Venkataramanan, and Zhang et al. can potentially be applied to the trust score. The trust score could be used as evidence for vehicle j being either trustworthy or untrustworthy. In this example the number of elements is limited to three, trustworthy, untrustworthy and uncertain (either trustworthy or untrustworthy), where each element is associated with a belief mass value.

4.3 Detecting Change in Behavior

The combined trust score about vehicle j, i. e., TI_j, is continuously monitored in the cloud and a suitable technique is needed in module 3 to detect changes in TI_j. The most basic detection technique would be to trigger an alert when a certain threshold, for example TI_j below 0.4, is reached. Naturally, this approach does not provide the flexibility of finding more subtle changes in a vehicles' behavior. Observing the mean and raising an alert as soon as the mean deviates from a given threshold on the other hand would be very slow and may not detect slow but steady changes.

Aminikhanghahi and Cook [2] provide a survey of change detection techniques for time series data including machine learning algorithms. They categorize existing work in unsupervised and supervised methods that are further split in more detailed categories. For this framework we require an unsupervised method that does not require training with labeled data. Hence, we investigate the use of two unsupervised methods, namely CUSUM and one Bayesian detection approach.

Cumulative Sum (CUSUM) was originally proposed by Page [30] and has been adapted over the years to, for instance, support online detection [2, 3]. This method is able to detect small and steady changes, as the cumulative sum of the deviations from a target value is calculated [3]. Granjon [16] describes the CUSUM algorithm, discusses practical considerations, such as choosing the detection threshold, and refers to other proposed variations of CUSUM.

Bayesian approaches for change point detection have, compared to CUSUM, the advantage of being faster; they require fewer samples for detecting change and have a lower computational cost [2]. In Section 5.3 we give more details and compare an implementation of CUSUM and Bayesian change detection.

4.4 Data Analysis in the Cloud

First, the data needs to be analyzed in order to design and apply appropriate anomaly detection techniques. Knowledge Discovery in Databases (KDD) [13] is a process for gathering new knowledge from data. This process consists of several steps starting from understanding the area of application to interpreting and evaluating patterns found in the data, thus gaining new knowledge. Data mining is one step in this process and supports the developer in identifying new patterns in the data by applying specific algorithms [5, 13].

Hemdan and Manjaiah [19] provide an overview of the principles of digital forensics, intrusion detection and suitable types of data science methods in the context of Internet of Things. The authors list prediction, classification, clustering and relation rule techniques as appropriate techniques for intrusion detection. Torres et al. [25] review machine learning techniques in the context of cyber security. They provide more detail and discussions on relevant techniques, such as self-organizing maps (SOM) and random forest. Kang [21] focuses on anomaly detection techniques for monitoring a product's health, however, the overview of machine learning techniques and discussions about the advantages and disadvantages of each category of machine learning techniques is still relevant for identifying techniques for module 4.

5 V2C ANOMALY DETECTION FRAMEWORK

We present a detailed description of each module and discuss the applicability of methods identified in Section 4 in the following subsections. Moreover, we propose how the selected methods can be adapted when necessary.

5.1 Trust Evaluation

In **module 1**, a trust score, often a value between 0 and 1, for each vehicle is computed every time a vehicle interacts or receives V2V messages, e. g., Cooperative Awareness Messages (CAM) [12]. The trust score needs to reflect the cooperativeness when performing cooperative actions, such as platooning, lane change and crossing an intersection. In addition, vehicles should also be evaluated based on the accuracy of the information they provide, e. g., speed, geographical position, driving intentions as well as correctness of warning messages. Such an evaluation of each vehicle's behavior and accuracy is typically performed using a trust or reputation model (see Section 4.1). The trust model should also be applied on the own vehicle's system, the ego vehicle.

Below, we discuss how trust models for calculating the trust scores of the ego vehicle and surrounding vehicles can be applied for the purpose of the *V2C Anomaly Detection* framework. For instance, vehicle i should calculate its own trust score $TI_{i,i}$ and the trust score for vehicles it interacted with, i. e., $TI_{i,j}$. These trust scores are updated over a specified period and maintained in the vehicle's own database along with a confidence value indicating the level of confidence for each specific trust score. After the specified period, the computed trust scores and corresponding confidence values are uploaded to the cloud for further analysis.

Trust Model. The trust models in [4] and [32] both include an evaluation of the vehicle's own performance with respect to trust. Compared to Bißmeyer et al. [4], Rosenstatter and Englund [32] also provide a detailed discussion about the relevant behavior specific to the vehicle in front, surrounding vehicles it interacts with and the ego vehicle.

The trust score presented in [32] is considered by the ego vehicle for decision-making and is comprised of four trust scores with range $[0, 1]$ that are combined using a weighted average: (1) ego vehicle; (2) vehicle in front; (3) interacting vehicles; (4) environment. The authors differentiate between the vehicle in front and other vehicles it interacts with as the vehicle in front is most important from a safety perspective and since it can be easily evaluated with its own sensors like the front radar. Moreover, the authors have shown in real scenarios with sensor noises how such a trust score can represent the current situation respectively trust in the surrounding other vehicles.

In our *V2C Anomaly Detection* framework, the trust scores of the vehicles should be evaluated based on all available knowledge of the ego vehicle, similar to [32], and split into two distinct and one combined trust score:

$TI_{i,j}$	Trust score of vehicle j computed by vehicle i.
$TI_{i,i}/TI_{j,j}$	Own perceived trust score of vehicle i/j.
TI_j	Combined trust score of vehicle j (see Section 5.2).

Anomaly Identification. The selected trust model needs to be able to identify the attacks and anomaly types defined in the scenarios in Section 2.1. To further clarify situations where trust scores and the consequent analysis in the cloud are extremely useful, we will give one practical example for each scenario in Table 2.

Update Frequency. The trust scores should ideally be calculated and updated upon receipt of new messages from relevant vehicles within the VANET. However, due to limited computational resources or the high number of vehicles identified as relevant, it

Table 2: Examples of how a trust model can identify attacks and anomalies.

Anomaly/Attack	Identification
Unauthorized Firmware Manipulation	
The attacker, such as the owner, upgrades the ECU firmware to let the automated vehicle (vehicle j) drive faster than the speed limit.	Surrounding vehicles are able to observe the speed of the affected vehicle using their own sensors, e. g., radar, and therefore reduce $TI_{i,j}$.
HW/SW failures	
The camera provides noisy images and therefore the vehicle's lane keeping assistant is misbehaving and causes the vehicle to bounce within the lane.	Surrounding vehicles observe the behavior (lateral movement within the lane) and therefore reduce $TI_{i,j}$. This behavior can be also detected by the vehicle itself.
Legitimate SW/HW update	
A legitimate update causes the vehicle to misinterpret the current road conditions, e. g., icy roads, and drives too fast.	Surrounding vehicles observe the higher speed and therefore reduce $TI_{i,j}$.
Cooperative Driving	
To disrupt traffic an attacker performs an unauthorized firmware modification so that the vehicle prevents other vehicles from performing a cooperative merge into one lane (i. e., does not allow vehicles to merge in front).	This behavior is mainly identified by vehicles that experience the denial of entering the lane in front of vehicle j, but also by other surrounding vehicles.

might be necessary to reduce the computation of trust scores to a lower frequency. Relevant vehicles are defined as those in direct proximity of the ego vehicle, but also other vehicles it interacts with, such as vehicles farther away that send a warning message, vehicles participating in a platoon or vehicles cooperating to efficiently pass an intersection.

The trust score needs to be updated as the vehicles are periodically re-evaluated. This can be achieved by computing the moving average respectively the weighted moving average when new trust evaluations for a vehicle are computed in order to maintain a single trust score $TI_{i,j}$ per vehicle per evaluation period, e. g., per day.

Confidence $C_{i,j}$. Due to the fact that interactions with a specific vehicle can last from a few seconds to several hours when considering platooning, we propose to include a confidence value or counter ($C_{i,j}$) to each $TI_{i,j}$ to indicate the confidence in the calculated trust score per vehicle. This way, individual vehicles can also use this confidence value to decide whether they should keep (and store) the trust score and report it to the cloud in cases when the maximum capacity of the allocated memory for the trust scores is reached.

Kerrache et al. [22] include a factor that considers the number of verified legal/correct actions within their calculation of the direct trust score (corresponds to $TI_{i,j}$). We suggest to upload a confidence value, such as the total number of interactions, together with the trust scores in order to indicate the quality of the calculated trust score $TI_{i,j}$.

Role of $TI_{i,i}$. Continuously evaluating a vehicle's own reliability and performance is important as the own system acts as the reference (its own sensors and knowledge about the environment) when evaluating other vehicles. When a vehicle, for instance, experiences a sudden drop in its own trust score while interacting with other vehicles or driving autonomously it can automatically enable logging of the most recent events. Logs can be uploaded to the cloud along with the trust scores of the other vehicles it interacted with recently. Including evidence of such a situation is crucial for further investigations, when, for instance, other vehicles also report a lower trust score for this vehicle.

Role of $TI_{i,j}$. The assessment of the other vehicles can be split in two different categories: (i) verification of reported sensor information, e. g., position, speed, acceleration, lane; and (ii) behavior, e. g., correctly reporting warning messages, interaction/cooperation with other vehicles, speed according to laws and road conditions.

Evaluating not only the trust when interacting with other vehicles is important from a safety perspective as vehicles will rely on sensor information received from other vehicles during platooning and other cooperative scenarios.

Reporting Trust Scores. Each vehicle maintains its own database of trust scores. The database maintains a trust score per vehicle per defined period, e. g., per day, comprising of 4 columns: (1) vehicle ID, (2) date, (3) trust score ($TI_{i,i}$ or $TI_{i,j}$), and (4) confidence value ($C_{i,j}$).

This data is uploaded to the cloud periodically, e. g., on a daily basis. By including $C_{i,j}$, it is possible to further filter trust scores and consider whether this specific trust score should be considered in the aggregated trust score per vehicle or not.

Recommendation. Adapting the selected trust model (e. g., [32]) to detect the attack examples shown in Tables 1 and 2 with a significant change in the trust score is essential for this framework. In Table 2 we showed how it is possible to identify such attacks. Moreover, we propose and discuss the types of trust scores, and how often they, including a confidence value, should be uploaded to the cloud.

5.2 Combining Trust Scores

Module 2 of the *V2C Anomaly Detection* framework is located in the cloud and receives trust scores ($TI_{i,j}$) from other vehicles about vehicle j periodically. Section 4.2 provides an overview of approaches to combine this knowledge, namely aggregation using mean and the use of Dempster's rule of combination.

Comparing the uses of arithmetic and geometric mean and considering the use case for this work as well as the range of the trust score, $[0, 1]$, we see the arithmetic mean to be more applicable since the expected data does not vary logarithmically.

Table 3: Comparison of the mean and DST with a varying ratio of honest ($TI_{i,j} = 0.9$) and dishonest ($TI_{i,j} = 0.2$) vehicles.

Reported Trust Scores $TI_{i,j}$		Arithmetic Mean	Geometric Mean	DST*	
0.9	0.2			$m(t)$	$m(d)$
100%	0%	0.90	0.90	1.00	0.00
80%	20%	0.76	0.67	1.00	0.00
60%	40%	0.62	0.49	1.00	0.00
50%	50%	0.55	0.42	0.97	0.03
40%	60%	0.48	0.37	0.39	0.61
20%	80%	0.34	0.27	0.00	1.00
0%	100%	0.20	0.20	0.00	1.00

*Dempster-Shafer theory: degree of belief; trust $m(t)$ and distrust $m(d)$

The strength of DST and Dempster's rule of combination is to combine knowledge provided by different observers without the need of having complete knowledge of the prior probabilities. The methods discussed in Section 4.2 highlight when DST is applicable in the context of VANETs: *When modeling and combining the uncertainty about whether to trust or not trust a specific node or information received, e. g., a warning message.*

Use Case. Consider a case when vehicles report contradicting trust scores about vehicle j. These vehicles can either be honest and report a high trust score (0.9), or dishonest or incorrect and report a low trust score (0.2). Table 3 shows the results with varying ratios of honest and dishonest vehicles. Comparing arithmetic and geometric mean shows that the geometric mean is more pessimistic and that a minority of dishonest or disagreeing vehicles have a higher impact on the combined trust score. As the *V2C Anomaly Detection* framework aims at identifying anomalies by observing the aggregated trust score reported by a majority of honest vehicles, it is desirable to use the arithmetic mean.

For the DST-based approach, we adapt Chen and Venkataramanan [7] and define three elements: Trust, Distrust, and Uncertainty. If a vehicle trusts vehicle j with probability of α it results in the following basic belief masses: $m(t) = \alpha$; $m(d) = 0$; $m(u) = 1-\alpha$; If a vehicle distrusts vehicle j it results in: $m(t) = 0$; $m(d) = \alpha$; $m(u) = 1 - \alpha$. In addition, we define that a $TI_{i,j}$ below 0.5 equals to distrust with a probability of $1 - TI_{i,j}$, e. g., $TI_{i,j} = 0.2$ in Table 3 indicates a distrust of $m(d) = 0.8$ and uncertainty of $m(u) = 0.2$. The results when applying a DST approach similar to the one we described shows that DST does not provide a sufficient level of detail. For instance, when the belief is high in both cases, honest $m(t) = 0.9$ and dishonest $m(d) = 0.8$ vehicle, the uncertainty is still smaller than 10^{-2}.

Recommendation. Considering the behavior of these three different approaches we suggest applying the arithmetic mean for trust scores with a sufficiently high confidence value. For specific cases where the majority of vehicles report a low confidence value $C_{i,j}$, it can be tested whether a weighted average yields a better result when the change detection (module 3) is applied.

5.3 Detecting Change in Behavior

The observation of the historic development of the trust score is essential for this framework as an alert from **module 3** triggers further investigations. Section 5.1 defines the factors and means for computing a trust score reflecting other vehicles' and the own vehicle's behavior to identify anomalies in the scenarios described in Section 2.1. The combined trust score TI_j described in Section 5.2 is a one-dimensional time series with a new data point added each evaluation period, e. g., each day.

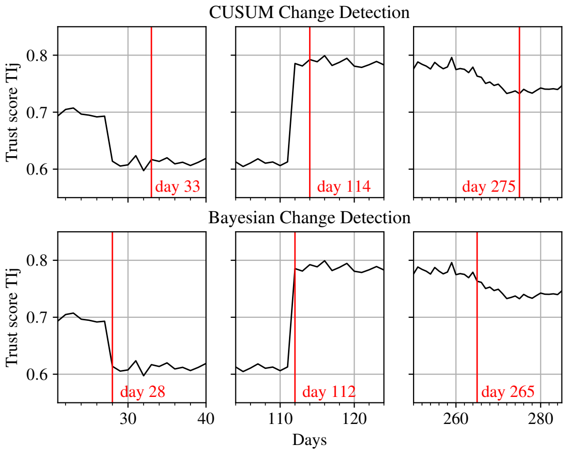

Figure 3: Comparison of detection techniques applied to the trust score TI_j when 3 different types of changes occur.

A variety of change point detection techniques are applicable for this kind of data (see Section 4.3). Figure 3 shows a simulation of the aggregated trust score TI_j of vehicle j reported daily using the arithmetic mean over a period of one year with three different types of changes injected: (i) an immediate drop of 0.1 on day 28; (ii) an immediate increase of 0.2 on day 112; and (iii) a slow decrease of TI_j where each day one more out of the 9 vehicles reports a decrease of the trust score of 0.05 starting from day 262.

We used the CUSUM detector implementation from Duarte [10] and modified it according to [3, p.40] describing the two-sided CUSUM algorithm. The Bayesian online change detection implementation from Kulick [23] following Adams and MacKay [1] was chosen as a second candidate.

Figure 3 shows that both detection techniques were able to detect the changes of the trust score. For these specific injected changes CUSUM was 5 days respectively 2 days slower in detecting the changes (i) and (ii). Change (iii) is already detected by the Bayesian detection after 3 out of the 9 vehicles reported a $TI_{i,j}$ decreased by 0.05 whereas the CUSUM detected the change once TI_j stabilized.

Recommendation. The comparison of both techniques shows that the Bayesian change detection [23] outperforms the two-sided CUSUM detection [3, p.40]. This behavior is also confirmed by Aminikhanghahi and Cook [2], as they state that Bayesian approaches require less samples for detecting change and have a lower computational cost.

5.4 Data Analysis in the Cloud

An alert triggers **module 4** when a change in the trust score of individual vehicles in module 3 is detected. The trust scores ($TI_{i,j}$) reported by the affected vehicle will be temporarily excluded in the computation of TI_j in module 2 to avoid including possibly wrong or forged trust scores. The cloud data is utilized by module 4 to investigate the cause of the anomaly and help to identify whether more vehicles in the fleet are affected. This cloud analysis can be split in two parts: (i) manual investigations; and (ii) automated anomaly & intrusion detection.

Manual Investigations. Initial manual investigations are necessary in order to identify and apply appropriate techniques for anomaly detection. Thus, it is important to follow a structured approach such as the Knowledge Discovery in Databases (KDD) process [13] which defines the steps for analyzing data. This process includes data mining as one step for applying statistics and machine learning techniques on the pre-processed data with the aim to discover new patterns and gain knowledge. As the type and the amount of data differs between the different entities, it is needed to follow such a process to being able to automate this analysis.

Anomaly and Intrusion Detection. Anomalies can be detected by deploying IDSs in the cloud. IDSs can be designed to detect misuse (knowledge-based [9]), i. e., signatures of known attacks, or to detect a change in behavior or specification (anomaly-based [9]).

A majority of anomalies and intrusions can be detected with specification-based techniques, such as specifications about the defined protocol handshakes, network protocol specifics and conformity to the application protocol. Other parameters that can be validated with such specifications are the location and the IP address from which the requests were sent and the exact time a workshop claimed to have performed an update.

Anomaly detection techniques based on artificial intelligence need to be adjusted for the dataset, thus it is important to follow a process such as KDD. In Section 4.4 we refer to publications that identify and categorize relevant machine learning techniques for anomaly detection. An example of a relevant solution is a method based on Isolation Forest [24] presented by Siddiqui et al. [36]. The introduced anomaly detection system produces not only an anomaly score, but also generates explanations for detected anomalies for the analysts and additionally includes a feedback loop to increase the detection performance. Another technique to consider are self-organizing maps (SOM). Qu et al. [31], for instance, explore SOM-based clustering techniques used for intrusion detection.

Recommendation. Considering the fact that vehicle manufacturers may have to monitor a fleet of more than a million vehicles and that many detection techniques have a high time complexity (see [5]), it may not be feasible to perform an on-line analysis on the entire data available in the cloud. Therefore, a scalable solution is required. The *V2C Anomaly Detection* framework reduces the computational costs by only triggering a search for anomalies in the cloud when an anomaly of the trust score, reported by several independent vehicles, is detected. Moreover, we suggest the following two steps:

(1) Upon receiving an alert from module 3 the cloud data relevant to vehicle j needs to be analyzed in order to find more information about the anomaly and its cause.
(2) Once patterns of anomaly or intrusion in the cloud data specific to the alert about vehicle j have been identified, the entire vehicle fleet or a subset, such as vehicles within a specific region, can be analyzed.

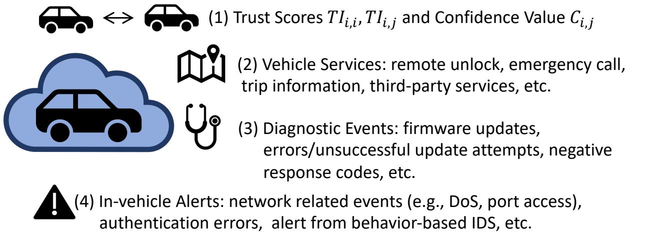
↔ (1) Trust Scores $TI_{i,i}, TI_{i,j}$ and Confidence Value $C_{i,j}$
(2) Vehicle Services: remote unlock, emergency call, trip information, third-party services, etc.
(3) Diagnostic Events: firmware updates, errors/unsuccessful update attempts, negative response codes, etc.
(4) In-vehicle Alerts: network related events (e.g., DoS, port access), authentication errors, alert from behavior-based IDS, etc.

Figure 4: Data accessible in the cloud

5.5 Vehicle Data in the Cloud

In this section we explore the types and variety of data available in the cloud. The type of data strongly depends on the actor performing the analysis and their terms of agreement with their customers. The actor is not limited to the vehicle manufacturer, but it may also be the owner of a vehicle fleet, e. g., logistics company, or an authority. We therefore identify and describe the type of data that should be considered for the analysis in module 4.

Data uploaded to the cloud can be split in four categories: (1) trust scores; (2) application data from services using the server back end; (3) diagnostic events, e. g., update events, negative response codes (NRCs) according to ISO 14229 [15]; (4) in-vehicle IDS alerts. Figure 4 provides an overview and examples of these categories.

Vehicle Services. Many vehicle services, such as remote unlock, emergency call and smartphone application, require communication to the server back end. The data of these services can contain status updates from the vehicle, for instance, location, traffic and road conditions, and trip statistics. The information shared depends on the services the customer subscribes to as well as the user agreement for using private sensitive data for security analysis. Metadata from the communication between the vehicle and the server, and data from the application protocol and possible deviations from it can be additional evidence of an anomaly or intrusion.

Events and service information performed by an authorized workshop are also uploaded to the cloud in order to keep track of the modifications and repairs performed on the individual vehicles. Such data includes information about the workshop, time and location as well as which vehicle unit was upgraded, defect and replaced, and firmware versions of each vehicle unit.

In addition, third-party service providers may choose to share the identity or more information about anomalously behaving vehicles.

Diagnostic Events. Vehicles not only need to report that firmware updates have been performed, they also need to provide information about when and how, e. g., over-the-air or manually at the workshop, the new firmware was downloaded. Furthermore, OEMs need to know whether the firmware update was successful and if it took more than one attempt to upgrade. This information can be used, for instance, to identify whether vehicles were subject to a possible intrusion attempt. ISO 14229 [15], *Road Vehicles – Unified Diagnostic Services (UDS)*, is widely used and also incorporated in the AUTOSAR system architecture.[1] From a security perspective UDS events are of high interest as these services are used to modify firmware and for short-term controls, e. g., triggering ECU restarts.

[1] https://www.autosar.org

Appendix A.1 of ISO 14229 [15, Part 1] lists the defined Negative Response Codes (NRCs). An example when such NRCs are sent is an event that the limit of failed authentication attempts has been reached or the attempt of sending specific UDS commands while the vehicle was driving faster than the specified limit.

In-vehicle IDS Alerts. In-vehicle IDSs are essential for detecting and subsequently preventing a large number of attacks. This need is also reflected in the *ENISA Good Practices for Security of Smart Cars* [14]. The detail of information when IDS alerts are received depends on the type of IDS. Thus, the information from alerts can range from detailed information about which attack has been recognized, to only an alert indicating that behavior outside the modeled normal behavior has been detected.

6 DISCUSSION

To highlight the benefits of the *V2C Anomaly Detection* framework, we will discuss a use case similar to change (iii) in Section 5.3 to detail the tasks and discuss the challenges for each module. The introduced change causes the combined trust score TI_j to continuously drop as more vehicles report a lower trust score for vehicle j. Possible causes for this change could be either a malicious actor who performed an illegal modification of the firmware (scenario 1 in Section 2.1) or a legitimate firmware update (scenario 3).

(1) Trust Evaluation. Each day more vehicles observe an undesired behavior of the automated vehicle j, such as speeding or driving faster than what the current road condition allows. Therefore, the vehicles lower the individually perceived trust score $TI_{i,j}$ according to the trust model and report it together with their own trust score $TI_{i,i}$ and $C_{i,j}$ to the vehicle manufacturers' cloud.

Opening such a peer evaluation between vehicles to a group of vehicle manufacturers would naturally lead to more independent evaluations by other vehicles encountered on the roads and thus more data. However, the trust models need to be similar in terms of evaluation criteria in order to avoid biases of different implementations. Another challenge is the identification of vehicles when pseudonym certificates are used to hide the true identities of vehicles in the VANET. In this case the corresponding entity, i. e., certificate authority, needs to be involved in order to correlate the trust score to the correct vehicles.

(2) Combining Trust Scores. Module 2 combines the trust scores $TI_{i,j}$ with a sufficiently high confidence value $C_{i,j}$ in form of TI_j. In the beginning only a few vehicles may experience a change in the behavior as example 5 in scenario 3, for instance, is only perceived when the road is slippery or icy, e. g., in the mornings and evenings, however, over time more vehicles report a lowered $TI_{i,j}$ causing the overall trust score to drop more significantly. The trust score perceived by the vehicle itself, i. e., $TI_{j,j}$, may stay stable as the vehicle itself perceives the environment incorrectly due to the firmware update. For this reason, we suggest to monitor $TI_{j,j}$ separate from TI_j.

(3) Detecting Change in Behavior. With the TI_j degrading each day, module 3 detects the change and raises an alert once a certain threshold is reached. Sections 4.3 and 5.3 show that there are many suitable algorithms for detecting change in one-dimensional time series data. In this work we explored two techniques in more detail, namely CUSUM and Bayesian online detection. The comparison in Figure 3 shows that the Bayesian online detection is faster compared to CUSUM as it detects the change already while the change is progressing. This characteristic is also described in the comparison of both techniques [2].

(4) Cloud Analysis. As soon as the observation of the combined trust score per vehicle TI_j triggers an alert raised by module 3, the anomaly detection defined in module 4 gets activated. In the first step, the anomaly detection performs an analysis on data related to vehicle j. In this context, a specification-based IDS may detect an unusually high number of unsuccessful firmware update attempts, possibly caused by an attacker, followed by a successful upgrade within a specified period. Another analysis can inspect whether the vehicle was recently in a workshop or if the firmware was upgraded. Based on these anomalies, the system can further search for vehicles with similar patterns in the database.

In addition to specification and rule-based intrusion detection mechanisms, machine learning techniques, such as classification and clustering techniques are appropriate candidates for this type of data. Nevertheless, it is necessary to follow an approach such as KDD to explore the data, find new patterns and consequently adapt existing detection techniques.

Evaluation. Representing undesired and malicious behavior in form of trust scores is essential for detecting anomalies using this framework. Existing trust models [4, 32] show how data plausibility checks and misbehavior detection can be used as a base for making decisions in automated vehicles. In Section 5.1, we describe the requirements for such trust models and specifics on how to use them including detailed examples highlighting what kinds of anomalies a trust model needs to be able to identify to significantly reduce the trust score. Moreover, we evaluated the modules on their own with initial experiments and discussions considering also the practicality of each module. A sophisticated prototype covering individual vehicles including the cloud data they produce is planned in future work.

7 CONCLUSION

In this paper, we present the *V2C Anomaly Detection* framework, which is a novel framework combining the assessment of Vehicle-to-Vehicle communication and the perceived quality of cooperative interactions between vehicles resulting in a *trust score*, with vehicle data in the cloud. The peer evaluation of vehicle behavior allows identification of local anomalies and attacks even when important security controls such as in-vehicle IDSs fail to detect them, for example due to an attacker exploiting a vulnerability, an insider or a firmware upgrade causing unintended behavior. Furthermore, the analysis of cloud data makes it possible to detect and identify patterns of anomalies and intrusions on a wider scale such as on a fleet level. Ultimately, the advantage of the *V2C Anomaly Detection* framework lies in the fact that it is designed to reduce the computational costs in the cloud by triggering a cloud analysis once the combined trust evaluation performed by independent vehicles shows a significant change, i. e., a changed behavior resulting in a decline of the trust score.

We have provided scenarios focusing on persistent threats to explain the requirements for each module of the *V2C Anomaly Detection* framework in terms of functionality, inputs and outputs.

We also provide an initial identification and detailed discussions to aid in choosing or adapting techniques for each module so that a vehicle manufacturer, fleet owner or other actor in the cloud is able to select and adapt relevant techniques depending on the available data.

ACKNOWLEDGMENTS

This research was supported by the Vinnova-funded projects “CyReV (phase 1 & 2)” and “KIDSAM” and by the Swedish Civil Contingencies Agency (MSB) through the projects RICS2 and RIOT.

REFERENCES

[1] Ryan P. Adams and David. J. C. MacKay. 2007. Bayesian Online Changepoint Detection. arXiv:0710.3742 [stat.ML]

[2] Samaneh Aminikhanghahi and Diane J. Cook. 2017. A survey of methods for time series change point detection. *Knowledge and Information Systems* 51, 2 (2017), 339–367. https://doi.org/10.1007/s10115-016-0987-z

[3] Michèle Basseville and Igor V. Nikiforov. 1993. *Detection of abrupt changes: theory and application.* Vol. 104. Prentice Hall, Englewood Cliffs, NJ.

[4] Norbert Bißmeyer, Sebastian Mauthofer, Kpatcha M. Bayarou, and Frank Kargl. 2012. Assessment of node trustworthiness in VANETs using data plausibility checks with particle filters. In *Vehicular Networking Conference (VNC)*. IEEE, Seoul, South Korea, 78–85. https://doi.org/10.1109/VNC.2012.6407448

[5] Anna L. Buczak and Erhan Guven. 2016. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys Tutorials* 18, 2 (2016), 1153–1176. https://doi.org/10.1109/COMST.2015.2494502

[6] Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shacham, et al. 2011. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *USENIX Security Symposium*. USENIX, San Francisco, CA, 77–92.

[7] Thomas M. Chen and Varadharajan Venkataramanan. 2005. Dempster-Shafer theory for intrusion detection in ad hoc networks. *IEEE Internet Computing* 9, 6 (2005), 35–41. https://doi.org/10.1109/MIC.2005.123

[8] Arturo Davila and Mario Nombela. 2012. Platooning - Safe and Eco-Friendly Mobility. *SAE Technical Paper 2012-01-0488*, Article 2012-01-0488 (2012), 5 pages.

[9] Hervé Debar, Marc Dacier, and Andreas Wespi. 1999. Towards a taxonomy of intrusion-detection systems. *Computer Networks* 31, 8 (1999), 805 – 822. https://doi.org/10.1016/S1389-1286(98)00017-6

[10] Marcos Duarte. 2020. detecta: A Python module to detect events in data. https://github.com/demotu/detecta. visited on 2020-11-12.

[11] Richard G. Engoulou, Martine Bellaiche, Talal Halabi, and Samuel Pierre. 2019. A Decentralized Reputation Management System for Securing the Internet of Vehicles. In *International Conference on Computing, Networking and Communications (ICNC)*. IEEE, Honolulu, HI, 900–904. https://doi.org/10.1109/ICCNC.2019.8685551

[12] ETSI. 2014. *Basic Set of Applications; Part 2: Specification of Cooperative Awareness Basic Service.* Intelligent Transport Systems (ITS) – Vehicular Communications EN 302 637-2 V1.3.2. ETSI.

[13] Usama Fayyad, Gregory Piatetsky-Shapiro, and Padhraic Smyth. 1996. From Data Mining to Knowledge Discovery in Databases. *AI Magazine* 17, 3 (Mar. 1996), 37. https://doi.org/10.1609/aimag.v17i3.1230

[14] The European Union Agency for Network and Information Security (ENISA). 2019. *Good Practices for Security of Smart Cars.* Technical Report. ENISA.

[15] International Organization for Standardization (ISO). 2020. *ISO 14229:2020 Road vehicles – Unified diagnostic services (UDS).* Standard. ISO.

[16] Pierre Granjon. 2013. *The CuSum algorithm-a small review.* Technical Report hal-00914697. GIPSA-lab.

[17] Pinyao Guo, Hunmin Kim, Le Guan, Minghui Zhu, and Peng Liu. 2018. VCIDS: Collaborative Intrusion Detection of Sensor and Actuator Attacks on Connected Vehicles. In *Security and Privacy in Communication Networks*, X. Lin, A. Ghorbani, K. Ren, S. Zhu, and A. Zhang (Eds.). Springer International Publishing, Cham, 377–396.

[18] Talal Halabi and Mohammad Zulkernine. 2019. Trust-Based Cooperative Game Model for Secure Collaboration in the Internet of Vehicles. In *International Conference on Communications (ICC)*. IEEE, Shanghai, China, 1–6. https://doi.org/10.1109/ICC.2019.8762069

[19] Ezz E. Hemdan and D. H. Manjaiah. 2018. *Cybercrimes Investigation and Intrusion Detection in Internet of Things Based on Data Science Methods.* Springer International Publishing, Cham, 39–62. https://doi.org/10.1007/978-3-319-70688-7_2

[20] Rasheed Hussain, Jooyoung Lee, and Sherali Zeadally. 2020. Trust in VANET: A Survey of Current Solutions and Future Research Opportunities. *IEEE Transactions on Intelligent Transportation Systems* (2020), 1–19.

[21] Myeongsu Kang. 2019. *Machine Learning: Anomaly Detection.* Wiley-IEEE Press, Hoboken, NJ, Chapter 6, 131–162. https://doi.org/10.1002/9781119515326.ch6

[22] Chaker A. Kerrache, Carlos T. Calafate, Nasreddine Lagraa, Juan-Carlos Cano, and Pietro Manzoni. 2016. Hierarchical adaptive trust establishment solution for vehicular networks. In *27th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, Valencia, Spain, 1–6. https://doi.org/10.1109/PIMRC.2016.7794617

[23] Johannes Kulick. 2020. Bayesian Changepoint Detection – Python Implementation. https://github.com/hildensia/bayesian_changepoint_detection visited on 2020-11-12.

[24] Fei T. Liu, Kai M. Ting, and Zhi-Hua Zhou. 2008. Isolation Forest. In *Eighth International Conference on Data Mining*. IEEE, Pisa, Italy, 413–422. https://doi.org/10.1109/ICDM.2008.17

[25] Javier Martínez Torres, Carla Iglesias Comesaña, and Paulino J. García-Nieto. 2019. Review: machine learning techniques applied to cybersecurity. *International Journal of Machine Learning and Cybernetics* 10, 10 (01 Oct 2019), 2823–2836. https://doi.org/10.1007/s13042-018-00906-1

[26] Charlie Miller and Chris Valasek. 2015. Remote exploitation of an unaltered passenger vehicle. *Black Hat USA* (2015), 91.

[27] Michael Müter, André Groll, and Felix C. Freiling. 2010. A structured approach to anomaly detection for in-vehicle networks. In *Sixth International Conference on Information Assurance and Security*. IEEE, Atlanta, GA, 92–98. https://doi.org/10.1109/ISIAS.2010.5604050

[28] Tarak Nandy, Rafidah M. Noor, Mohd Yamani Idna Bin Idris, and Sananda Bhattacharyya. 2020. T-BCIDS: Trust-Based Collaborative Intrusion Detection System for VANET. In *2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA)*. Durgapur, India, 1–5. https://doi.org/10.1109/NCETSTEA48365.2020.9119934

[29] Nasser Nowdehi, Wissam Aoudi, Magnus Almgren, and Tomas Olovsson. 2019. CASAD: CAN-Aware Stealthy-Attack Detection for In-Vehicle Networks. (2019). arXiv:1909.08407 [cs.CR]

[30] Ewan S Page. 1954. Continuous inspection schemes. *Biometrika* 41, 1/2 (1954), 100–115.

[31] Xiaofei Qu, Lin Yang, Kai Guo, Linru Ma, Meng Sun, et al. 2019. A survey on the Development of Self-Organizing Maps for Unsupervised Intrusion Detection. *Mobile Networks and Applications* (02 Oct 2019). https://doi.org/10.1007/s11036-019-01353-0

[32] Thomas Rosenstatter and Cristofer Englund. 2018. Modelling the Level of Trust in a Cooperative Automated Vehicle Control System. *IEEE Transactions on Intelligent Transportation Systems* 19, 4 (2018), 1237–1247. https://doi.org/10.1109/TITS.2017.2749962

[33] Steffen Sanwald, Liron Kaneti, Marc Stöttinger, and Martin Böhner. 2020. Secure Boot Revisited: Challenges for Secure Implementations in the Automotive Domain. *SAE Int. J. Transp. Cyber. & Privacy* 2, 2 (aug 2020), 69–81. https://doi.org/10.4271/11-02-02-0008

[34] Glenn Shafer. 1992. Dempster-shafer theory. *Encyclopedia of artificial intelligence* 1 (1992), 330–331.

[35] Erfan A. Shams, Ahmet Rizaner, and Ali H. Ulusoy. 2018. Trust aware support vector machine intrusion detection and prevention system in vehicular ad hoc networks. *Computers & Security* 78 (2018), 245–254. https://doi.org/10.1016/j.cose.2018.06.008

[36] Md A. Siddiqui, Jack W. Stokes, Christian Seifert, Evan Argyle, Robert McCann, et al. 2019. Detecting Cyber Attacks Using Anomaly Detection with Explanations and Expert Feedback. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, Brighton, United Kingdom, 2872–2876. https://doi.org/10.1109/ICASSP.2019.8683212

[37] Chawin Sitawarin, Arjun N. Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. 2018. DARTS: Deceiving Autonomous Cars with Toxic Signs. arXiv:1802.06430 [cs.CR]

[38] Seyed A. Soleymani, Abdul H. Abdullah, Mahdi Zareei, Mohammad H. Anisi, Cesar Vargas-Rosales, et al. 2017. A Secure Trust Model Based on Fuzzy Logic in Vehicular Ad Hoc Networks With Fog Computing. *IEEE Access* 5 (2017), 15619–15629. https://doi.org/10.1109/ACCESS.2017.2733225

[39] Minrui Yan, Jiahao Li, and Guy Harpak. 2020. Security Research Report on Mercedes-Benz Cars. *Black Hat USA* (2020), 38. https://www.blackhat.com/us-20/briefings/schedule/index.html#security-research-on-mercedes-benz-from-hardware-to-car-control-20746

[40] Chunhua Zhang, Kangqiang Chen, Xin Zeng, and Xiaoping Xue. 2018. Misbehavior Detection Based on Support Vector Machine and Dempster-Shafer Theory of Evidence in VANETs. *IEEE Access* 6 (2018), 59860–59870. https://doi.org/10.1109/ACCESS.2018.2875678