EASI: Edge-Based Sender Identification on Resource-Constrained Platforms for Automotive Networks

Marcel Kneib
Robert Bosch GmbH
Marcel.Kneib@de.bosch.com

Oleg Schell
Bosch Engineering GmbH
Oleg.Schell@de.bosch.com

Christopher Huth
Robert Bosch GmbH
Christopher.Huth@de.bosch.com

***Abstract*—In vehicles, internal Electronic Control Units (ECUs) are increasingly prone to adversarial exploitation over wireless connections due to ongoing digitalization. Controlling an ECU allows an adversary to send messages to the internal vehicle bus and thereby to control various vehicle functions. Access to the Controller Area Network (CAN), the most widely used bus technology, is especially severe as it controls brakes and steering. However, state of the art receivers are not able to identify the sender of a frame. Retrofitting frame authenticity, e.g. through Message Authentication Codes (MACs), is only possible to a limited extent due to reduced bandwidth, low payload and limited computational resources. To address this problem, observation in analog differences of the CAN signal was proposed to determine the actual sender. Some of the prior approaches exhibit good identification and detection rates, however require high sampling rates and a high computing effort. With EASI we significantly reduce the required resources and at the same time show increased identification rates of 99.98% by having no false positives in a prototype structure and two series production vehicles. In comparison to the most lightweight approach so far, we have reduced the memory footprint and the computational requirements by a factor of 168 and 142, respectively. In addition, we show the feasibility of EASI and thus demonstrate for the first time that voltage-based sender identification is realizable using comprehensive signal characteristics on resource-constrained platforms. Due to the lightweight design, we achieved a classification in under 100 μs with a training time of 2.61 seconds. We also showed the ability to adapt the system to incremental signal changes during operation. Since cost effectiveness is of utmost importance in the automotive industry due to high production volumes, the achieved improvements are significant and necessary to realize sender identification.**

I. Introduction

Vehicles can no longer be considered as closed systems, as they are increasingly equipped with functionality that interacts with the environment [35], [25], [52], [61]. This includes local connections offered directly by the vehicle, such as Bluetooth or WiFi, in order to control various functions or to retrieve information via smartphones. But also retrofittable solutions, e.g. in the form of diagnostic dongles, offer additional interfaces to the vehicle, which can be affected by vulnerabilities [18]. In addition, modern vehicles are increasingly equipped with mobile cellular connections in order to access cloud services and to communicate with other road participants. Besides useful services, these interfaces also offer attackers the possibility to manipulate the vehicle and its ECUs without prior physical access, as shown by several researchers [7], [60], [43], [62]. Once an ECU is compromised, an attacker can remotely control or influence the vehicle respectively individual functions [43], [36], [42], [23]. In particular, it turned out that the lack of security mechanisms for the CAN [53], which is still the most used standard, enables the manipulation of internal vehicle communication [25]. Thus, it is possible for a remote attacker to send forged messages from compromised ECUs, which in turn enables the control of vehicular functions, as demonstrated by Miller and Valasek [43]. Their work on a Jeep Cherokee led to a recall of 1.4 million vehicles. Another demonstration is provided by the work of the Tencent Keen Security Lab [62], [5]. The research team discovered multiple vulnerabilities in various BMW models, including the ability to manipulate ECUs connected via CAN over a wireless connection. This fundamental problem has already been criticized by consumer watchdogs, resulting in a demand for a vehicle kill switch [13] in connected vehicles.

The implementation of cryptographic measures on CAN is only possible with restrictions [66], [39], [20], [34]. This is due to the limited computing power of the ECUs, the small available bandwidth and the short payload per CAN message. For example, MACs are significantly truncated [3] or only used for a limited number of messages. In addition, it is not possible to unambiguously determine the sending ECU using MACs, since they do not provide non-repudiation. However, information about the sender of a message is also relevant for Network-based Intrusion Detection Systems (IDSs) [25], [47], which are expected to be a common security measure around 2023 [52]. These systems analyze the message traffic and can detect attack patterns or deviations from the expected behavior. One way to react on intrusions is, besides to warn the driver [47], [24], to update existing signature-based systems with the observed attack patterns in order to be able to react quickly to zero day exploits. This does not necessarily prevent an attack on a particular vehicle, but immunizes the entire vehicle fleet and prevents potential major damage. The recognition itself can e.g. be carried out in the cloud and appropriate measures can be transmitted to the vehicles over the air. Knowledge about the sender would also improve detection rates and

Network and Distributed Systems Security (NDSS) Symposium 2020
23-26 February 2020, San Diego, CA, USA
ISBN 1-891562-61-4
https://dx.doi.org/10.14722/ndss.2020.24025
www.ndss-symposium.org

accelerate the elimination of vulnerabilities, as the source of the attack can be identified. Should this information be provided by cryptographic measures, digital signatures have to be implemented, whose usage is considerably more expensive.

As an alternative, approaches have been introduced which enable the detection of attacks based on physical characteristics [46], [9], [11], [12], [34]. Characteristics of sent CAN signals are used to generate fingerprints which can be utilized to determine the sender. This enables detecting attacks with a corresponding probability which require the forgery of frames in order to be executed. In this way, Miller and Valasek's attack on the Jeep Cherokee could also have been detected at the point where the researchers accessed the CAN bus via the compromised multimedia system. Unfortunately, existing approaches which use comprehensive signal characteristics require a high sampling rate of up to 2.5 gigasamples per second (GS/s) to generate the fingerprints, which is not provided by standard microcontrollers (MCUs). Besides additional hardware costs for the measurement, this also results in large amounts of data which have to be processed by the system in a limited period of time, e.g. in less than $200\,\mu$s for a standard frame with maximum payload. Should the calculation require too much time, the system has to buffer many messages and also cannot react quickly enough to detected attacks [63].

Some of the aforementioned approaches show good results, but require high demands on the hardware. Even if high speed MCUs are available and their general performance increases, the costs of implementing security functions play a major role [20]. This is especially true for the automotive sector [36], [47], [25], [39], [20], [61], as security features are often difficult to monetize as they are viewed as a fundamental requirement and not as an additional feature [26]. To address this problem, we present EASI (Edge-Based Sender Identification), a novel approach which has significantly lower requirements and thereby increases the cost effectiveness and applicability of the sender identification for automotive networks. Compared to the most lightweight approach [34], we were able to reduce the requirements by two orders of magnitude, i.e. memory utilization by a factor of 168 and the computing effort by 142.

Existing approaches extract the characteristics from the symbols of the entire CAN frame. Our key insight is that there are only minor changes in the characteristics within one frame. Therefore, it is sufficient to generate the fingerprint from a single symbol. This already allows a great reduction in the amount of data, but not in the sampling rate. To achieve this, only individual points of the actual signal are measured which are combined afterwards to a representative symbol. Thus, our proposed approach additionally reduces the required sampling rate, enabling the implementation of fingerprinting technology using low-cost standard hardware. In addition, we demonstrate for the first time that comprehensive signal characteristics can be processed by machine learning algorithms on standard MCU architectures which are comparable to hardware used in actual ECUs. Besides the calculation of the characteristics, this also includes the training of the model, its adjustment due to drifts and the actual classification. As a result, information on the performance requirements of approaches using comprehensive signal characteristics are given for the first time. Based on the methodology of using only a single symbol we tuned the system accordingly, which includes the optimization of the utilized characteristics and a refinement of the system parameters like the necessary update procedure. Besides using the data already used for the evaluation of Scission [34], we extended the evaluation by a deeper analysis of a voltage-aware attacker and a one-week drive involving the utilization of electronic consumers. While keeping the identification rate high, we have improved the detection rate and showed also that even attacks during drive could be detected with an accuracy over 99 %. Besides the practical enablement, especially for the automotive industry, our contributions are:

- Reduction of resource requirements for sender identification using comprehensive signal characteristics.
- Optimization of system parameters to maintain a robust operation over a longer runtime.
- Demonstration of feasibility on a standard low-cost, resource-limited MCU, including model adjustments to changing signal characteristics during runtime.
- Evaluation on a prototype and two production vehicles as well as under changing conditions over one week involving different electronic consumers.

II. Background

A. Controller Area Network

The CAN is a broadcast bus over which the internal ECUs communicate via frames, containing up to 8 bytes of data. The frames do not contain a receiver or sender address, but an identifier which specifies the priority and meaning of the transmitted data. Thus, an ECU can use multiple identifiers exclusively. The identifier is 11 or in extended format 29 bits long and is used by only one ECU in the corresponding bus. Since CAN is a broadcast bus, it is possible that several participants access the bus simultaneously, which would lead to a faulty transmission. This is avoided by the Carrier Sense Multiple Access/Collision Resolution [31], which ensures that the frame with the highest priority prevails the arbitration phase. During the arbitration, the sending ECUs observe the current bit on the bus and compare it with the transmitted bit. If both correspond, the next bit is transmitted, otherwise the transmission is aborted and restarted as soon as the bus is free. The bus consists of two twisted wires, CAN high and low, which are terminated with $120\,\Omega$. When a dominant bit (0) is transmitted, CAN high is pulled to 3.5 V and CAN low to 1.5 V. A recessive bit (1) is represented by 2.5 V on both wires. The final voltage level, known as the differential signal, is then determined by the subtraction of CAN low from CAN high. An advantage of this procedure is that electromagnetic interference affects both lines simultaneously and thus balances out in the differential signal. If five identical bits are transmitted, an additional contrary bit is inserted for synchronization, which is called a stuff bit.

B. Cause of the Signal Characteristics

The characteristics of a signal are determined by the transmitting ECU and the channel to the measuring point [4]. The generated signal during a voltage level change is in theory a square-wave signal. In practice, however, square-wave signals are characterized by rise and fall times, indicating the time required for a signal to reach its target value. Among others, these are influenced by the capacitances and inductances of the circuits and by the power supply of the ECU respectively the transceiver [10]. The primary voltage source used is a 12 V battery for passenger cars and 24 V for trucks. The required operating voltage of 5 V is ensured by voltage regulators, which also stabilize the voltage supply. Due to manufacturing variations and imperfections, electronic components differ slightly [44], leading to variations in the signals. For instance, resistors come with a tolerance of industry typical 5 %. Furthermore, CAN actually requires a third cable for grounding. But in practice, grounding is realized via the vehicle chassis, which can result in different ground voltages between individual devices [64]. Variations are also caused by power reflections of a transmitted signal. This is affected by impedance mismatches and non-linear changes in cable characteristics, including the lengths and terminations of the bus. All together, this may result in overshooting, what can be followed by an oscillation of the signal, known as ringing [45]. Therefore, the topology has a considerable influence on the signal waveform, since impedance mismatches mainly occur at junctions and the devices.

III. Edge-based Sender Identification

A. Security Models

1) System Model: The structure of internal networks depends fundamentally on the manufacturer and model. There are simple vehicles with only one network, but also more complex models whose internal network consist of several individual buses. The buses are used for different functions, such as powertrain, comfort or multimedia, and are interconnected by gateways. In order to prevent the system from being bypassed, we assume that the ECU on which the system is implemented is protected by security mechanisms [52], such as an Hardware Security Module (HSM) [65] and is therefore considered trustworthy. HSMs are already available for the automotive market, e.g. by the Infineon Aurix [30] platform. Especially the gateway is a suitable device for EASI, which connects several buses and thus can react in case of an attack depending on the malicious bus segment. Although the simultaneous monitoring of multiple networks is possible, a single CAN network or segment is considered in the following for simplification. In order to record the signals, the ECU on which the system is implemented has a measuring point on the monitored CAN network. This allows EASI to analyze the CAN message signals and thus to decide if an intrusion is present. Further, the sender identification allows determining the sending ECU if it is known by the system, which will help to accelerate the elimination of the exploited vulnerability. Since the communication flow in vehicles is static, i.e. it is known which ECU is allowed to send which identifiers, the system can determine whether a message is legitimate or not. In addition, we assume that the absence of periodic messages or an increase in their frequency can be detected, as this can be easily realized. Several approaches exist for this purpose [25], [37], [47], [9].

2) Attacker Model: In principle, an attacker with bus access can manipulate the vehicle functions by injecting messages, due to the missing sender authenticity. By flooding, periodic signals can be overwritten, by denial-of-service (DoS) attacks the entire communication can be disturbed and through the bus-off attack [8] ECUs can be disconnected from the bus. These manipulations, which lead to additional or missing messages, can be detected by monitoring the transmission schedule. Therefore, in the following, a more intelligent attacker is considered who aims to use aperiodic messages or intends to take over the sending of periodic messages unnoticed [55]. Since the identity of the sender cannot be determined in CAN, the attacker can impersonate any ECU by using the corresponding message identifier. As a consequence, each connected bus participant is able to start impersonation attacks in order to influence vehicle functions without being noticed. This has far-reaching consequences, especially if this is possible without prior physical access.

This leads to the first attacker model, the *compromised ECU*. Some ECUs have wireless connections, such as WiFi, Bluetooth or cellular, via which they can be attacked and compromised by an attacker. If a vulnerability in a vehicle model exists within an ECU which is accessible via the internet, an attack can be launched on all vehicles of that type. Such an intrusion into the system is possible without prior physical access to the vehicle and remains hidden from the vehicle or its passengers. The consequences are already possible today and have been demonstrated such as influencing safety-critical vehicle functions [43], [62], [5]. Here, the considered compromised ECU is known by EASI in advance. Thus in a previous learning phase, a model was created based on the signal characteristics of the corresponding ECU, which can be used for its identification. Attacks which require prior physical access scale worse, which is why the detection of this attack is the main goal of this work.

However, the assumption that the compromised ECU is known by the system cannot always be guaranteed, as the attack on the Jeep Cherokee shows [42]. This leads to the second attack model, the *unmonitored ECU*. An ECU was used for sending unauthorized CAN messages, which in its original state was only designed as a passive, listening-only device. Due to a vulnerability in the update mechanism, this ECU could be reprogrammed and used to send forged messages. Consequently, EASI does not have a model of the signal characteristics in this attack scenario.

The third model, the *additional ECU*, is present when an attacker connects a simple additional device to the monitored bus in order to send manipulated messages. Among other things, this is used to steal vehicles [29], to deactivate AdBlue systems, to obscure defective airbags or for engine tuning [23], [29]. If an attacker has physical access to a vehicle, additional devices can be connected directly to the bus or with little effort to the on-board diagnostics (OBD)-II port. The OBD-II port is a standardized diagnostic interface located near the dashboard.

As mentioned, the main goal of the proposed system is to detect remote attacks by determining the sender of received messages. Therefore, we additionally consider a *voltage-aware attacker* who is aware of the existence of the proposed system. In order to bypass the system, an attacker can specifically try to influence the signals of the compromised ECU in such a way that it resembles the signal of the ECU to be faked. Influencing the signal directly, i.e. the shape of the rising edges, is not possible by remote, since these are defined by the structure of the present CAN and the electrical components of the ECU. However, we enable the attacker to manipulate the voltage level by draining the battery and heating up or

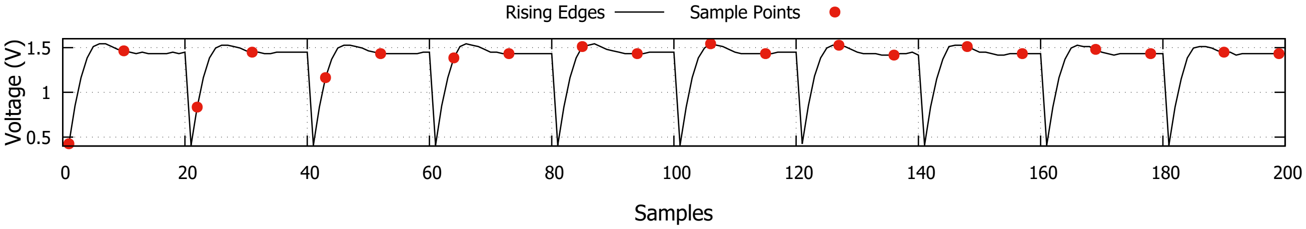

Fig. 1: Sampling points of the rising edges of a frame.

cooling down the compromised ECU. While discharging the battery leads to a decrease in the voltage level of all ECUs, the change in temperature of the compromised ECU causes the corresponding voltage level to rise or fall. In addition, we allow the attacker to analyze the voltage level of all ECUs connected to the bus. This is a very strong capability, for which it would be necessary, for instance, to have the analog-digital-converter (ADC) of an ECU directly connected to the bus. Usually, ECUs are only connected to the bus over the CAN transceiver, which does not provide any information about the signal characteristics. Although the attacker could analyze the characteristics, our measurements have shown that the actual measuring position has a large influence on the received characteristics. This makes it considerably more difficult for an attacker to make statements about the signals actually recorded by EASI. However, for our evaluation we neglect this circumstance, which means that we allow the attacker to receive the same information as our IDS.

B. Phase 1: Signal Gathering

The first step is to record the differential signal of the actual CAN frame. Since the amount of data to be processed per frame has a large influence on the required computing power, a major goal is to reduce the required amount of sampling points. Thus, the system only considers the bits which contain the most important characteristics with regard to sender identification. This applies to those bits which contain a rising edge, i.e. dominant bits which are preceded by a recessive bit [34]. A further reduction is achieved by not recording the entire frame or all corresponding bits, but only a single rising edge. This is not considered to be a major disadvantage, as we observed that the relevant characteristics only change very slightly within one frame, as illustrated in Figure 1. Smaller deviations within a frame are due to noise and are not particularly relevant for the identification.

In order to obtain extensive characteristics from a CAN signal, a certain number of samples per bit is necessary, as otherwise too much information is lost. For illustration, Figure 2 shows the signal curve of a single symbol with different sampling rates. Proposed fingerprinting approaches use rates between 20 MS/s [34] and 2.5 GS/s [11], whereby identification rates up to 99.85 % [34] are achieved.

However, existing MCU architectures, such as the Infineon TriCore [30], NXP MPC [49] or STM32 [59], are often only equipped with ADCs with lower sampling rates. Therefore, an additional ADC is necessary even for a scanning with relatively low rates of 20 MS/s, which leads to increased costs. For instance, a reduction of the sampling rate by a factor of 10 is necessary to implement sender identification on a standard MCU providing 2 MS/s. This is achieved by not recording the edge with a high sampling rate, but several rising edges with a lower sampling rate. The sampling times are shifted accordingly so that a representing bit can be composed of the measurements after the sampling phase. This procedure is called Random Interleaved Sampling (RIS) [50] and is a common technique of Digital Storage Oscilloscopes (DSOs) to achieve high sampling rates for repetitive signals. For example, an absolute sampling rate of 20 MS/s with an actual rate of 2 MS/s can be achieved by using 10 rising edges. The procedure is illustrated in Figure 1, in which the sampling points of 10 rising edges are marked. Figure 3 shows the resulting rising edge, a complete edge and an average edge calculated from all 10 rising edges. As seen in the figure, the different recordings do not show much difference or are comparable to the differences that also occur between signals from the same ECU.

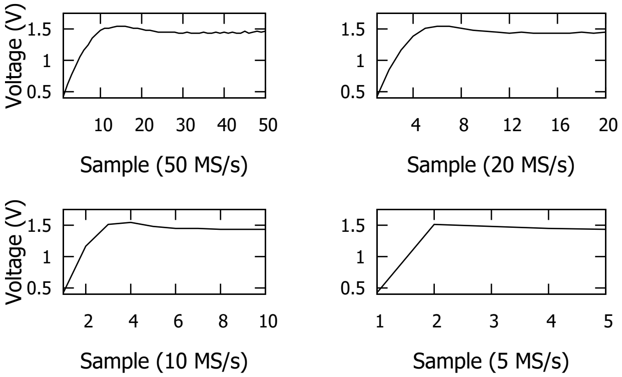

Fig. 2: Rising edge recorded with different sampling rates.

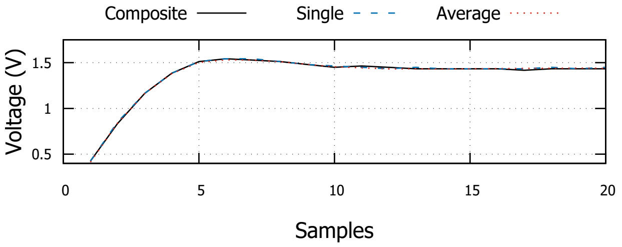

Fig. 3: Rising edge recorded in different modes.

TABLE I: Features extracted from the rising edge for classification while x contains the measured voltages, N is the cardinality of x, y_m and y_f are the magnitude coefficients and frequencies, respectively, and M is the number of elements of y_m and y_f.

Rank	Feature	Description	Type	IG Prototype	IG Fiat	IG Porsche	IG General
1	Ratio Max Plateau	$\frac{Maximum}{Plateau}$	Descriptive	3.3	2.6	2.6	8.5
2	Skewness	$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x(i)-\mu}{\sigma}\right)^3$	Time	3.1	2.4	2.8	8.3
3	Plateau	$\frac{N}{4}\sum_{i=\frac{3}{4}N}^{N} x(i)$	Descriptive	3.1	2.3	2.7	8.1
4	Kurtosis	$\frac{1}{N}\sum_{i=1}^{N}\left(\frac{x(i)-\mu}{\sigma}\right)^4$	Time	3.1	2.5	2.5	8.1
5	Overshoot height	$Maximum - Plateau$	Descriptive	2.9	2.5	2.6	8
6	Irregularity	$\frac{\sum_{j=1}^{M-1}(y_m(j)-y_m(j+1))^2}{\sum_{j=1}^{M-1} y_m(j)^2}$	Frequency	3.3	1.9	2.6	7.8
7	Centroid	$\frac{\sum_{j=1}^{M} y_f(j)*y_m(j)}{\sum_{j=1}^{M} y_m(j)}$	Frequency	3.2	1.8	2.7	7.7
8	Flatness	$\sum_{j=1}^{M} y_m(j) * \frac{\sqrt[M]{\prod_{k=1}^{M} y_m(k)}}{\sum_{k=1}^{M} y_m(k)}$	Frequency	3.1	2	2.5	7.6
9	Mean	$\mu = \frac{1}{N}\sum_{i=1}^{N} x(i)$	Time	3.2	1.7	2.6	7.5
10	Variance	$\sigma^2 = \frac{1}{N}\sum_{i=1}^{N}(x(i)-\mu)^2$	Time	2.6	2.3	2.6	7.5
11	Power	$\frac{1}{N}\sum_{i=1}^{N} x(i)^2$	Time	3.1	1.5	2.7	7.3
12	Maximum	$max(x(i))_{i=1...N}$	Descriptive	3	1.9	2.3	7.2

C. Phase 2: Characteristic Derivation

After the signal is recorded and a representing edge is calculated, the system extracts various features from it. If no representative bit is used, a single or average edge can of course also be used at this point. Appropriate statistical features from time and frequency domain have already been investigated in previous work [11], [4], [34]. New in this case is that additionally *signal descriptive features* were examined. In this category those characteristics were categorized that contain specific knowledge about the signal course of rising edges. Especially the course of the stabilization is essential for the distinction of the signals. From all candidates, the 12 most important features were selected. For this purpose, the Information Gain (IG) algorithm from the Weka 3 Toolkit [56] was used. IG is a method for calculating how much information a feature provides about the class, which also allows to rank the considered features. In order to prevent features being selected which only fit the current situation or a specific setup, three different setups were considered. From the validation set of a prototype and two series production vehicles a common feature set was derived, which is shown in Table I. The feature vector extracted represents the fingerprint of the signal and the associated ECU.

D. Phase 3: Sender Identification

Identifying a sender on the basis of a feature vector, i.e. the actual fingerprint, is a classification problem. For these kind of problems a variety of algorithms exist, which are able to determine to which class a new observation belongs. In the selection of suitable algorithms with regard to the field of the application presented here, the data properties must be considered. The frames from which the fingerprints are extracted, and which are also used to create the model, are transmitted periodically and are therefore available incrementally. Keeping the resource-limited hardware in mind, this results in the necessity for a fast calculation, *Classification Speed*, as otherwise too many fingerprints have to be buffered. To achieve a low *Memory Footprint*, it is also necessary to select an algorithm which allows to establish the model from an incremental stream of training examples. Thus, and as changes in the characteristics are expected to happen, the *Model Adjustment* is another important criteria. A further point is the *Overall Complexity* of the algorithm, which also includes the number of freely selectable parameters. The assessment of the criteria regarding the considered machine learning algorithms is shown in Table II. Based on these estimations, the focus is primarily set on Logistic Regression (LR) since the algorithm has already proven to be suitable for sender identification [34] and Naive Bayes (NB), due to its low complexity. Further, we consider Support Vector Machines (SVMs) in the evaluation as it allows a non-linear separation using the radial basis function kernel.

TABLE II: Assessment of the machine learning algorithms.

	Classification Speed	Memory Footprint	Model Adjustment	Overall Complexity
LR	+	○	+	○
Naive Bayes	○	+	+	+
SVM	○	-	○	○
Decision Tree	+	-	-	+
Neural Network	-	○	-	-

Before it is possible to estimate the sender of transmitted frames based on their fingerprints during the operation of the IDS, the according relationship between ECUs and characteristics has to be learned supervised. Therefore, the system generates multiple fingerprints from several transmitted frames for each ECU, whereby the mapping between the frames and the ECUs is done using the included identifier. This is possible as each identifier is only used by one ECU and the communication is static. Thereby, it is particularly important to prevent an attacker from influencing the training data, known as poisoning attacks [27], [33]. When the system is initially trained, where also a key for later model adjustments is exchanged between the IDS and each ECU, it must be ensured that no bus manipulations are present. This can be done, for example, in an authorized workshop or, for new vehicles, during the production. After the training, the system calculates the average probabilities $avgP^k_{init}$ of each classifier k using a validation set, a separate data set for the unbiased evaluation of the generated model.

E. Phase 4: Intrusion Decision

For the analysis of the probabilities of the fingerprints and thus for the recognition of attacks, we use dynamic thresholds, a development of the approach introduced in [34]. In the following we will discuss these detection methods for the attacks described in Section III-A.

1) Compromised ECU: Normally, the ECU with the highest probability would be selected as the source of the received frame. If this ECU would not be allowed to use the present identifier, an attack would be assumed. However, we use an upper threshold $t^k_{max} = \alpha * avgP^k_{init}$ for each classifier k. Only if the probability of an ECU exceeds t^k_{max} and this device is not allowed to use the present identifier, the message is marked as malicious. This has the advantage that if the classification of a fingerprint is not clearly possible, e.g. due to a electromagnetic interference, the occurrence of a false alarm is less likely. Since it is assumed that the amount of trustworthy messages compared to malicious is significantly higher over the entire deployment time of the system, the reduction of false positives, i.e. wrong alarms, is of special interest. As an illustration, already a low false positive rate of 0.2 % would lead to a wrong alarm every 166 ms on a common CAN bus which transfers 3000 frames per second. At the same time, a slightly lower detection rate of malicious messages is to be expected. However, many attacks require the transmission of more than one message, which increases the general attack detection probability. In order to increase robustness against outliers, e.g. triggered by electromagnetic interferences, and to reduce the required computational effort, the system uses a further threshold $t^k_{min} = (1 - \alpha) * avgP^k_{init}$. Since it is known in advance which ECU is authorized to use an identifier, only the sender's probability for the corresponding ECU is initially calculated. Only if this probability undercuts t^k_{min}, the message is marked as suspicious and the probabilities of the remaining ECUs are calculated. If one of these probabilities exceeds the upper threshold t^k_{max}, the message is finally marked as malicious. In order to determine the threshold parameters, a statistical analysis of validation sets for the vehicle is used. Either a single value can be defined for all vehicles as presented in this work, or an individual one, fitted to the actual bus architecture, to satisfy the needs of the car manufacturer.

2) Unmonitored ECU: Here, three cases must be considered. If the signal of the unmonitored ECU is very similar to an ECU known to the system, which is not allowed to use the current identifier, the attack is detected as in the previous section. Should the signal from the unmonitored ECU be equal to the signal of the authorized ECU, the attack cannot be detected. Signals which are not similar to any of the known signals will lead to an increase in messages that have been classified as suspicious but not malicious. So, the system monitors the number of suspicious messages to detect this type of attack. Each message marked as suspicious causes an increase of a counter per ECU, which is decremented if a trustworthy message is present. If one of these counters exceeds a threshold, an alarm is triggered.

3) Additional Device: In addition to remote attacks, the system is also able to detect the connection of additional devices. The system is limited to the detection of simple devices, since an attacker with physical access, appropriate knowledge and sufficient resources has basically unlimited possibilities to manipulate a vehicle and thus bypass the system. However, this is associated with higher effort. The detection is possible as the topology changes when an additional device is connected to the monitored bus, which leads to an abrupt change in the signals of all monitored ECUs. This leads to a reduction in the identification probabilities of the existing ECUs in the moment of the modification. Thus, the number of suspicious frames will increase. If the sum of the suspicious counters exceeds a threshold value, an alarm is triggered.

4) Voltage-aware Attacker: Influencing the voltage levels of all ECUs by an attacker leads to a reduction of all identification probabilities. These are continuously monitored by EASI, which leads to a permanent adaptation of the models, whereby the sender identification is retained. However, if an attacker can abruptly and significantly influence the voltage level of all ECUs, so that a sufficiently high identification is no longer possible, a complete learning phase becomes necessary. In principle, the system's performance decreases during this learning phase. However, a persistent and abrupt change can be assumed as unlikely, since a standard ECU is not capable of such a rapid discharge of the battery.

If it is possible for an attacker to influence the voltage level of the compromised ECU, the voltage level can generally rise or fall. However, since such changes are not abruptly possible, the system is still able to continuously adapt the model. In principle, an attacker has the ability to approximate the signal of the ECU to be faked, but cannot achieve an exact adjustment, since the general shape of the signal remains unchanged. Even if the signals are similar, the success of such an attack is unlikely, since an attacker has no information about the characteristics of both the compromised and the ECU to be faked. In addition, the signal is defined by the actual topology, while the system is able to continuously adapt the model, we consider the system to be able to recognize such an attacker.

F. Model Adjustments

Since changes in the signal characteristics are expected, e.g. due to aging of components or corrosion, it is necessary to adjust the trained model according to these concept drifts [19]. A distinction must be made between an incremental drift, where the changes are occurring over a longer period of time, and an abrupt drift, where the changes are occurring spontaneously. In order to detect these changes in the signal characteristics and to react accordingly, the system monitors its performance by calculating the average classification probability $avgP_{op}^{k}$ of each classifier k in a sliding window. Observing the averages offers the opportunity to detect a deterioration of the system performance, which in turn can be utilized as an indication for a required model update. A big difference to the initial training is that after the deployment a non-manipulated system can no longer be assumed, which means that the update process must expect potential manipulation.

When a drift of a signal is recognized by the system, i.e. if the average classification probability of a classifier decreases for corresponding frames by more than 5 % or increases for frames send from other ECUs by more than 10 % compared to the reference probabilities $avgP_{init}^{k}$, the update phase is triggered. Within the adjustment phase, the IDS composes an update batch with already classified fingerprints from all ECUs and thus does not require additional computing capacity. The batch consist of those fingerprints, which can still be assigned to exactly one ECU with a probability greater than t_{max}^{k}, which is then used to adjust the corresponding model.

While this method deals with incremental drifts, it is also necessary to handle abrupt changes. This is especially relevant when it is not possible to generate an update batch due to fingerprints that cannot be classified with a probability greater than t_{max}^{k}. In this situation it is necessary to use the keys, which are exchanged during the initial training phase between the system and the observed ECUs. These can be used to generate authenticated and thus securely labeled data, which are necessary for a model update. Another case where abrupt changes may occur is when the vehicle is in a workshop for repair or maintenance. This may require a complete retraining of the system, which must be triggered e.g. by a secure diagnostic access [2]. Necessary cryptographic procedures are provided by the AUTOSAR module Secure Onboard Communication (SecOC) [3]. The use of cryptography for system retraining has basically the same problem of bandwidth limitation, but is not used continuously and without hard real-time requirements, which is why calculations are realistic even for resource-constrained ECUs. If, for example, 16 additional frames are sent once for six control units, the bus utilization, which is normally loaded by 65 % and corresponds to approximately 3000 frames/s at 500 kb/s, increases by 2.08 % for one second. The continuous use of MACs with a length of 24 bits and additional 8 bits for freshness values halves the available payload and hence leads to transmitting twice the amount of frames. This in turn results in a load of 130 %, which is obviously not possible without structural changes. Depending on the platform, it would be also possible to use further countermeasures [33], [16], [28] against poisoning attacks. However, they are not exactly tailored for mini-batch training on platforms with limited resources, as considered in this work. This means that not all data can be held during the training phase and must therefore be processed in pieces.

IV. Evaluation

In the following sections, the presented system is evaluated with regard to the basic sender identification and the intrusion detection. We use data from a prototype assembly, a Fiat 500 and a Porsche Panamera S E-Hybrid, which was also used in [34], thus enabling a direct comparison of both approaches. During the measurements for the initial evaluation, the vehicles were switched on, but stationary, while no electrical consumers have been actuated.

The prototype consists of five Arduino Unos, each equipped with two CAN shields and supplied from the same power source, a wall socket. The shields are identical in construction and use an MCP2515 [40] controller and an MCP2551 [41] transceiver. For the assembly original cables were used, while the bus was terminated with 120 Ω and the stubs with 2400 Ω. Higher resistors at the ends of the stubs are used to minimize reflections, as the bus topologies in many vehicles are not implemented exactly according to the standards in order to reduce costs. From this structure a total of 48128 frames with random payload were recorded.

The Fiat 500 has six internal ECUs, each using up to seven identifiers. In addition, two Raspberry Pis were connected, each equipped with a CAN Shield, in order to increase the number of ECUs. The first Raspberry Pi, referred to as ECU 6, was connected to the OBD-II port together with the DSO. The second Raspberry Pi, referred to as ECU 7, was connected directly to the bus in the trunk of the vehicle. Altogether 35129 frames were recorded from the Fiat 500, while its engine was switched off.

The second vehicle, the Porsche Panamera, has several separate CAN buses, whereby the powertrain domain was used for the evaluation. The considered segment has six internal control units and in order to increase the number as well, two additional Raspberry Pis were connected to the bus. These were connected together with the DSO directly to the bus near the armrest, since the OBD-II port has no direct connection to the observed bus. In this manner 9543 frames were recorded from the Porsche while its engine was switched on and off. Thus, in comparison to [34], the set additionally contains frames of the switched-on vehicle, which exhibit electromagnetic interference due to the hybrid system.

All data sets were divided into a training, validation and test set. The first 200 recorded frames from each available ECU were used to train the models, 10 % of the succeeding data for the validation and the remaining frames for the test set. The signals were recorded with a PicoScope 5204 at a sampling rate of 500 MS/s and a resolution of 8 bit. For processing, the signals were first converted to a differential signal, from which a representative bit was extracted. In order to achieve realistic results, the conversion and sampling times of constrained ADCs and comparators were considered when creating the representative bit. Initially, the first rising edge was detected by a voltage rise above 0.2 V, which represents the beginning of a frame. For the implementation, a comparator can be used here. Henceforth, the system was on hold for the time of the arbitration phase. Afterwards, two samples from each of the following ten rising edges are recorded. Due to the presumed sample rate of 2 MS/s, the recording was made at a distance of 500 ns. After the detection of an edge, the time of sampling was shifted by 25 ns per measurement already taken. After the sample phase, which works without complex synchronization, a representative bit was created from the 20 samples.

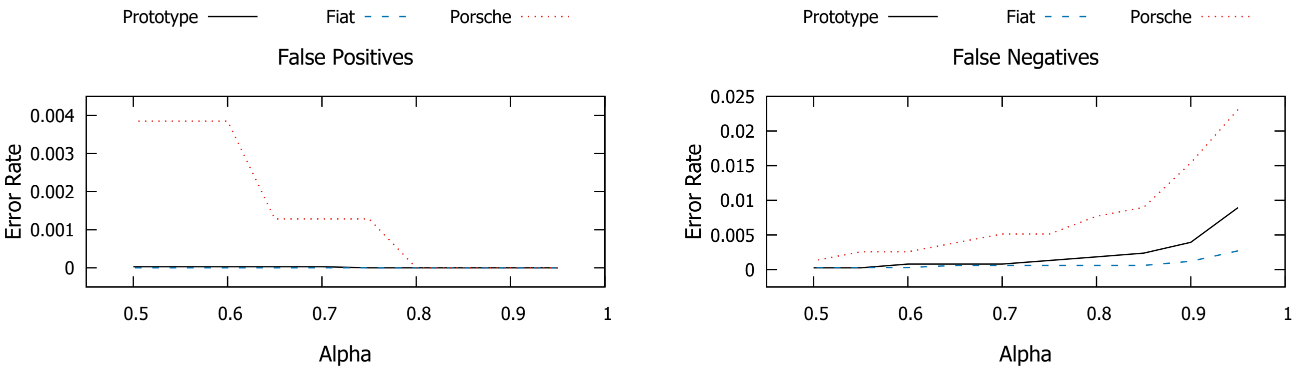

Fig. 4: Error rates for all setups with varying thresholds.

A. *Sender Identification*

First, the accuracy of the introduced sender identification is evaluated. For this, the approach was implemented in Python, using the Scikit Machine Learning and SciPy library. For Logistic Regression, the Newton-CG method was used for minimization and the abort condition was set to 1^{-6}. For the SVM, the Radial basis function was used as kernel and the existing parameters were retained. No additional parameters were necessary for the Naive Bayes classifier. The average, as well as the minimum identification rates of the ECUs of the three considered setups are shown in Table III. It can be seen that, with the exception of the Porsche, there are no significant differences in the identification rates between the various machine learning algorithms. The lowest identification rate of the Porsche is explained by the two additionally connected Raspberry Pis, which are identical in construction and are located approximately at the same position on the bus. Evaluating the Porsche with only one Raspberry Pi connected, results in an identification rate of 100%.

TABLE III: Identification rates.

	Prototype	**Fiat**	**Porsche**	**Average**
LR Avg	99.99	100	99.86	99.98
LR Min	99.95	100	99.41	99.92
SVM Avg	100	99.98	99.81	99.98
SVM Min	100	99.83	98.87	99.84
NB Avg	100	100	97.64	99.79
NB Min	100	100	87.15	98.88

B. *Detecting Compromised ECUs*

An identification rate of on average 99.98 % results in a false positive rate of 0.02 %, which means that every 5000 frames a false alarm occurs. The goal of the proposed intrusion detection is to reduce this rate, which is evaluated in this section. Since the amount of recognized malicious frames is another important criterion, attacks were simulated to investigate the false negative rate. For this purpose, 10 % of the signals from the test sets were selected for each ECU and handled as attacks. The targets of the attacks were continuously changed so that each ECU was counterfeited by every other ECU, which was achieved by changing the identifiers of the frames into identifiers used by other ECUs. This ensures that not only attacks from particularly well distinguishable devices are considered. However, before it is possible to perform the evaluation, it is necessary to configure the value α for the calculation of the node-dependent upper and lower threshold. This parameter was determined using the validation sets of the three setups. Therefore, we calculated the False Positive (FP) and False Negative (FN) rates, which are shown in Figure 4 for different α values. As $\alpha = 0.8$ is a good trade-off between the FP and FN rates, we selected this value for the following evaluation. The resulting confusion matrices for the different setups and classifiers are shown in Table IV. It can be seen how many of the original and the faked signals are recognized correctly by the system. The rates show that the threshold approach has reduced the FP to zero when Logistic Regression is used. However, as mentioned in Section III-E1, the detection rate of attacks decreases at the same time by 0.32 %. Accordingly, the chance for the Fiat setup is 0.06 % to miss a single forged frame, 0.000036 % to miss a second and to miss a third one the chance is already at 0.0000000216 %. This allows EASI to detect all attacks with the given probabilities that require sending at least one forged frame. Furthermore, the rates of the Porsche show that the robustness of the system increases, since the threshold approach compensates the minimum identification rates, when Naive Bayes is used.

C. *Detecting Unmonitored ECUs*

This section evaluates the detection of attacks via unmonitored ECUs. Since an already existing frequency analysis is assumed, i.e. missing and additional messages are detected, it is necessary that the ECU to be counterfeited is first deactivated and then its messages are taken over by the unmonitored ECU [8], as otherwise, the impersonated ECU can easily recognize forged frames [14]. We assume that the attacker has these capabilities and that an intrusion remains unnoticed by the vehicle. As a result, the unmonitored ECU must continuously send CAN frames to not get detected due to missing frames, which in turn can be analyzed by EASI. For evaluation, the Fiat 500 data set is used by training the system without ECU 7 and using it to send forged frames which are normally send from ECU 6. These ECUs were chosen as they are identical in construction and use the same power supply, i.e. they have similar characteristics. Only the position in the bus differs. Depending on the frequency of suspicious

TABLE IV: Confusion matrices of the IDS.

	Attack	Predicted		Suspicious
		0	1	Frames
Logistic Regression				
Prototype	0	100	0	0.01
	1	0.19	99.81	0.16
Fiat 500	0	100	0	0
	1	0.06	99.94	0.03
Porsche Panamera	0	100	0	0.03
	1	0.77	99.23	0.64
Support Vector Machines				
Prototype	0	100	0	0
	1	0	100	0
Fiat 500	0	100	0	0.03
	1	0.21	99.79	0.18
Porsche Panamera	0	99.99	0.01	0
	1	0.51	99.49	0.26
Naive Bayes				
Prototype	0	100	0	0
	1	0	100	0
Fiat 500	0	100	0	0
	1	0	100	0
Porsche Panamera	0	99.31	0.69	0
	1	2.31	97.69	1.93

frames and the number of frames after which such an attack should be detected, the parameters for the detection must be determined. For the Fiat 500, we identified a suspicious frame rate of about 0.06, which means that every 1666 frames such a frame erroneously occurs. Therefore, we increment the counter by 1 for each suspicious frame and decrement it by 0.006 for a normal frame. In addition, it is assumed that an alarm should be triggered after 10 fake frames, which is why the alarm threshold is set to 10. Figure 5 shows the course of the counter for suspicious frames. It can be seen, that it increases rapidly when the sender changes from ECU 6 to ECU 7 after frame 851. As configured, an alarm is triggered after 10 forged frames sent from the unmonitored ECU.

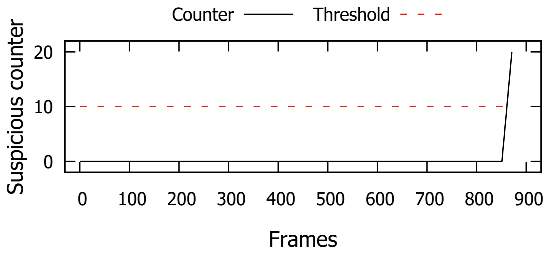

Fig. 5: Suspicious counter of ECU 6 from the Fiat 500 during unmonitored ECU attack.

D. Detecting Additional ECUs

For detection evaluation of simple additional ECUs, the data set of the Fiat 500 was used. Initially, the system was trained with signals from an unmodified bus, i.e. without ECU 6 and ECU 7 being connected to the bus. Both data sets were recorded one after another under the same conditions. Only the DSO was connected via the OBD-II port. After 500 signals were processed by the system, the data set was changed to signals of the changed bus. In order to detect this attack, the counters of the suspicious frames of all ECUs are considered. If the sum of the counters exceeds the set threshold value, an alarm is triggered. The threshold for the detection of additional ECUs is $\frac{\#ECU}{2}$ times the threshold for the detection of unknown ECUs. The course of the summed counter is shown in Figure 6. The attack was detected after 86 frames have been processed by the system. The difference to the detection of attacks via unmonitored ECUs is that by changing the bus topology the detection rates of all ECUs decrease, i.e. the number of suspicious frames increases at several ECUs. This makes the detection of this manipulation independent of forged messages.

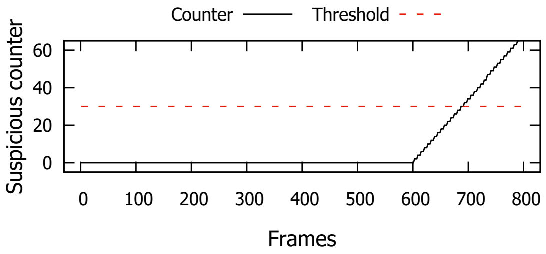

Fig. 6: Summed suspicious counter of all ECUs from the Fiat 500 during additional ECU attack.

E. Detecting Voltage-aware Attacker

For the simulation of this attack we have scaled the existing signals of an attacker in such a way that its voltage level is similar to the signal to be forged. The exact adaptation to an existing signal is not possible in real conditions without physical access, since an attacker has no information about the characteristics of the ECU which has been taken over or is to be forged. Even if the characteristics of an ECU could be measured exactly at the point of bus access, they are not identical with the characteristics measured by EASI. This is due to the different channels between the observed ECU and the two measurement points. This can be seen in Figure 7, where the same rising and falling edge of one frame is displayed, recorded at different bus positions.

It can also be assumed that an ECU has no possibility of making such a fine adjustment. The attacker simulated here is therefore more powerful than it can be assumed in reality. ECU 6 and ECU 7 of the Fiat 500 were used to evaluate the detection of this attack, as they are identical in construction and use the same power supply, which makes them more difficult to distinguish. 300 signals from ECU 7 have been adapted so that their voltage levels match the voltage level of ECU 6. The identifier has also been adapted accordingly to ECU 6. The system was trained with the original data and

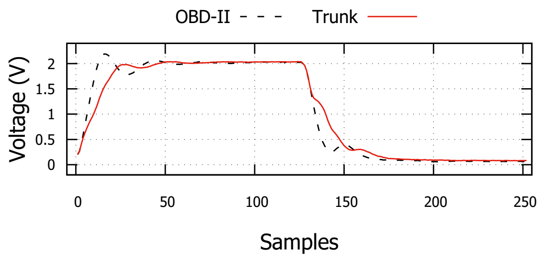

Fig. 7: Exemplary parallel measuring difference.

has subsequently processed the 300 manipulated signals. The change has reduced the average probability of ECU 7 by more than 3 %. However, the distinction between ECU 6 and ECU 7 remains, so that 95 % of the forged and adapted messages could be correctly detected as an attack. All the undetected forged messages were marked as suspicious, which indicates that this attack should also be considered when determining the thresholds. The simulation shows that adjusting the voltage level is not enough to circumvent the system dependably. Apart from that, the monitoring was able to detect the drop of the probabilities while maintaining the functionality of the system. This makes it possible to adapt to a change at an early stage.

F. Varying Conditions

For the evaluation of the performance of the approach under changing conditions, additional analyses were performed. Three datasets of the Fiat in its original state, i.e. without having the Raspberry Pis connected, were used, which were recorded via the OBD-II port of the vehicle. The first data set was recorded at an ambient temperature of approximately 25°C (77°F) while the engine was switched off and cold. The first 200 frames per ECU of the set were used for the initial training and the remaining 5685 frames of the data set were classified completely correct. After the start of the vehicle the second data set was recorded, which contains data of a trip of approximately 30 minutes at an ambient temperature of over 32°C (89.6°F). After the vehicle was completely heated up, it was parked in an underground garage at approximately 23°C (73.4°F). All of the 6672 frames were correctly classified using the already trained classifiers. The recording of the third data set was started after a cooling phase of three hours while the vehicle was switched off. After a short time the vehicle was started and driven for another 20 minutes at an ambient temperature of approximately 36°C (96.8°F). The 4863 frames recorded were fully classified except for one, but due to the proposed threshold approach a false positive was prevented. During the whole process no re-training of the models was necessary and an identification rate of 99.99 % was achieved.

G. Manipulation of the Power Supply

We also performed a fourth measurement with 6885 frames while the engine of the vehicles was switched off but with a battery pack connected in order to analyze the behavior of the system when manipulating the power supply. This data set was used in a second evaluation as training data set. Subsequently, the remaining frames and the three data records from Section IV-F were processed by the system. Although differences in the voltage level of over 20 mV and signal deviations are visible, the robust selection of diverse features ensures that tampering had no effect. Here, only a single frame showed irregularities, but due to the threshold approach this does not lead to a false alarm.

H. One Week Drive

For the analysis of environmental factors as well as the effect of electrical consumers in the vehicle on the signal characteristics, we performed another series of measurements with the Fiat. Data from a total of nine half-hour trips were recorded over a period of one week. The vehicle was in the original state and the DSO was connected to the bus via the OBD-II port. The measurements were carried out in winter with rain, wetness and drought at ambient temperatures between -2°C (28.4°F) and 10°C (50°F). During the journey, different consumers were used, such as lights, turn signals, windscreen wipers, heating and also the start-stop automatic of the vehicle. Altogether more than 65,000 frames were recorded.

Again, the first 200 frames per ECU of the first trip were used to train the model while 10 % of the test frames were handled as attacks. During the analysis we noticed another striking feature, the distance between the overshoot and the lowest point of the subsequent oscillation. We replaced this feature with the variance in order to keep the amount of features. With this adjustment we achieved a identification rate of 99.98 % instead of 99.96 % and a false positive rate of 0 % instead of 0.01 % for the unadjusted feature set. With a detection of 99.6 % of the malicious frames we could also show that the attack detection still works while driving. Again, no update was necessary during operation. Overall, the measurements show that a model can maintain a high identification rate even under changing conditions over a longer period of time.

V. Microcontroller Implementation

In order to estimate the performance requirements of the approach on limited hardware, the fingerprint generation, classification, model training and update mechanism was implemented on an STM32 NUCLEO-F446RE [59], a 14 $ development board. The MCU used runs at a clock frequency of 180 MHz, offers 512 kB flash and 128 kB SRAM, provides a floating point unit (FPU) and also a digital signal processor (DSP). This platform is quite comparable with MCUs used in today's vehicles, which include the STM SPC58 [58] with up to 3x180 MHz and the Infineon TriCore TC3x [30] with up to 6x300 MHz. These automotive MCUs also offer an HSM to realize the requested security measures, like Secure Boot [52].

A. Signal Gathering

To achieve a realistic comparison we have used the signals of the Fiat 500 as already done in Section IV. The representative rising edges, each consisting of 20 samples, were first determined on the PC and then transmitted via UART together with the associated identifier to the evaluation platform. For the sampling on the actual device, a fast-compare channel or automotive capable high-speed comparator can be used to control the internal ADC according to the description in Section IV. The SPC58 and the TC3x series also offer Generic Timer Modules (GTMs) [57], mainly used for powertrain tasks [38], which are free programmable and designed to process repetitive task. Such a module can be utilized to control the internal ADC and thus to generate a representative edge without using the actual MCU. Thus, the MCU must be used for feature calculation and classification only after the edge has been captured.

B. Characteristic Derivation

The features listed in Table I are first calculated from each transmitted edge using the CMSIS DSP Software Library [1]. The resulting features are stored temporarily as floating point numbers for further processing. This already offers optimization potential by switching the approach to integer arithmetic. Calculating the features requires 16,730 cycles, which corresponds to a duration of 92.94 μs. This also includes the normalization of the features with 2,283 cycles, which improves Logistic Regression.

C. Model Generation

Before classification is possible, the model has to be trained. Here we use Logistic Regression with L2 regularization. In order to achieve a low memory utilization, a mini-batch approach was implemented. Instead of using 200 frames per ECU at once, a new mini-batch with 8 frames per ECU is used after each iteration. Altogether the training consists of 25 iterations, whereby in each iteration 20 minimization steps using the conjugate gradient method are performed. For each ECU 2,348,400 cycles are required per iteration, which corresponds to 13,046 μs. Accordingly, the learning time requires 2.61 seconds for the given training set of 200 frames per ECU.

D. Sender Identification and Intrusion Decision

After the training phase, the classifiers can be used for sender identification. For each classifier, 663 cycles or 3.68 μs are needed for the calculation of the probabilities. In the optimum case, i.e. with a high probability of detection, a total duration of 96.62 μs per signal was achieved. Making the decision about an intrusion requires with 468 cycles further 2.6 μs. If all probabilities have to be calculated, the total duration was 124.98 μs. A CAN frame with 8 bytes of payload occupies the bus for minimum 222 μs with an automotive typical bandwidth of 500 kB/s [32]. Therefore, the presented approach is capable to process the signals sufficiently fast in order to identify the sender on a 100 % loaded bus, as illustrated in Figure 8. However, in practice the load is much lower to ensure safety requirements.

Regarding the performance on the system, an identification rate of 99.94 % and a false negative rate of 0.03 % was achieved for the Fiat 500 data set from Section IV-A. As in the previous evaluations, no false positives occurred.

E. Naive Bayes

We have also implemented and analyzed Naive Bayes on the considered platform. As expected, the training of the model is less computationally expensive, which is reflected in an approximately 20 times faster training time. The classification achieves similar good results, whereby the estimation of the origin of a frame takes more than three times as long compared to Logistic Regression. In addition, with the NB classifier the calculation cannot be carried out only for a single ECU, which leads to the fact, that in contrast to LR, it require almost the entire time.

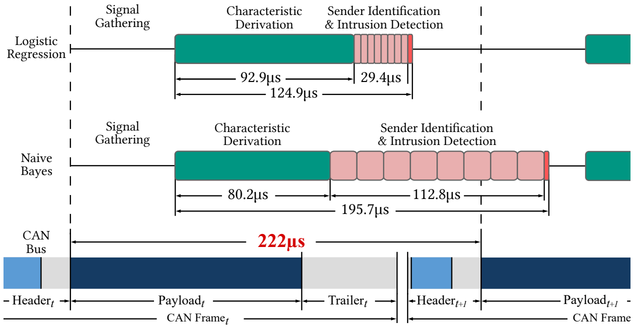

Fig. 8: Worst case timings on the resource-constrained platform.

F. Model Adjustment

Since it is of interest whether the presented update mechanism is able to maintain a consistently high identification rate even with very strong changes, the update mechanism was evaluated with LR and NB. However, since almost no real signal changes occurred during the measurements, the evaluation was performed with simulated changes. After the analysis of 1000 unchanged frames, the voltages of 4000 randomly selected frames were scaled incrementally to 80 %, followed by 3000 frames, which were scaled to 110 % of their original value. Fig. 9 shows the identification rates for LR and NB with and without the update mechanism. Obviously, even with such strong changes of up to 30 % after frame 5000 in a very short period of time, the LR is able to maintain a high identification rate with the proposed update procedure.

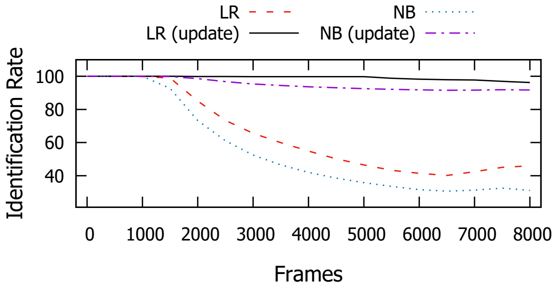

Fig. 9: Comparison of model updates for the Fiat 500 set.

Besides the evaluation of incremental drifts, it is also of interest how to handle the model adaptation to abrupt changes. Therefore it was examined whether it is more efficient to adapt the existing model or to create it from scratch, by comparing the number of required iterations, i.e. mini-batches. For comparison with the existing model, it was first trained with unchanged data. In the following, randomly selected frames were scaled by a factor of 0.8 to simulate the abrupt drift. Afterwards, the two approaches were analyzed for both machine learning algorithms. While it is an advantage for the LR to adjust the existing model, it is more efficient for NB to train a new model, as it takes longer for the changes to be fully adopted. In contrast to the complete training with 25 iterations, only 4 iterations are necessary for the adjustment with LR, which takes less than one second on the system used here. In such a situation, the secure labeling of the data for which we exchanged keys in the initial training phase must be guaranteed. Summarizing, it was shown that the sender identification can be maintained during incremental drifts with a high detection rate, and that the model can be adapted to abrupt drifts.

VI. Related Work

Cryptographic measures, such as MACs, are the most suitable methods to ensure authenticity. Since AUTOSAR 4.2.1, cryptographic algorithms are provided by the SecOC module [3]. In order to prevent collisions, the general recommendation for the minimum length of the MAC tag is 64 bit [15]. However, this also corresponds to the maximum payload of a CAN message, which is why tags are strongly truncated. But even with an exemplary length of 24 bits plus 8 bits as freshness counter, the available bandwidth is reduced by more than 50 %, as regular synchronization of the freshness counter is necessary as well. Another factor is key management, which includes not only key generation and distribution but also its secure storing. This requires HSMs, which can provide additional acceleration for the calculation of cryptographic algorithms [65]. However, even for ECUs which include such a hardware extension, the calculation or verification of the tags is not trivial, as further overhead is added by the communication with the HSM. For example, for ECUs that process many frames, such as gateways, it is difficult to process all tags accordingly. But although a MAC can be implemented without problems, they do not offer non-repudiation. As a result, ECUs capable of verifying frames are also able to forge them. Non-repudiation is provided by digital signatures, but their requirements are considerably greater and therefore not suitable for CAN communication.

Murvay and Groza [46] have shown for the first time that the differences of the CAN signal characteristics can be used for sender identification and remain unchanged over several months on a prototype setup. Based on these results, Choi *et al.* [11] optimized the approach by using machine learning methods for classification. Using a neural network and a sample rate of 2.5 GS/s a classification rate of 96.48 % was achieved. The approach also requires to embed a fixed 18-bit value for all ECUs in the extended identifier field to extract the characteristics from the corresponding signal. This allows a classification independent of the transmitted data, but at the same time reduces the available bandwidth. In addition, the extended identifier can no longer be used for its original purpose. The best results are achieved by using a fixed value which, apart from additional stuff bits, consists exclusively of dominant bits. Therefore, characteristics contained in the rising or falling edge only slightly influence the classification. Thus, the voltage level is the most important and main characteristic. With VoltageIDS [12] Choi *et al.* have presented a further development of their approach operating without the extended identifier field. The approach was evaluated in two vehicles, with identification rates ranging from 90.01 % to 99.61 % being achieved while driving. In addition, a lower sample rate of 250 MS/s was analyzed, resulting in a detection rate of up to 93.54 % on real vehicles.

The IDS extension Voltage-based attacker identification (Viden) [10] works only on the basis of the voltage level, whose goal is to identify the attacking ECU after an intrusion has been detected by a high-level IDS. The system generates a model based on the average voltage level of dominant bits of the ECUs, collected from multiple frames. Although the basic concept has low resource requirements, a 200-tree Random Forest is used to verify the decisions, which negates the performance advantage. The verification phase is necessary for the detection of voltage-aware attackers and to distinguish near-equivalent voltage profiles, for which we see an increased probability due to the use of a single signal characteristic. In addition, there are no details about the life cycle of the classifier, whose training respectively actualization we consider to be very complex, as with Decision Trees. Beyond that, Viden uses the two signals, high and low, separately instead of the difference signal, making it more sensitive to interferences.

Scission [34] uses comprehensive signal characteristics for the identification. By analyzing the individual symbols of a received frame separately, an identification rate of 99.85 % could be achieved, while all false positives were prevented during the evaluation in two production vehicles. Even if Scission has a lower computing requirement compared to the work of Choi *et al.* [11], there is still a high hardware demand. This is mainly caused by the resulting data rate due to the methodology as well as the sampling rate of 20 MS/s. Scission can therefore only be partially implemented on an automotive platform, which must also provide an external ADC.

Simple [17] creates an average symbol like VoltageIDS [12] and therefore also utilizes comprehensive signal characteristics for sender identification. The difference to previous methods lies in the fact that the average symbol is used as a direct input for the identification via a distance metric. While this provides runtime benefits, we have observed that using machine learning is an advantage regarding the robustness as well as the identification and detection rates. This is also shown by the results, as despite a sampling rate of 50 MS/s and the use of a less complex bus architecture, the equal error rate of 0.89 is higher than the rates we achieved with EASI.

With TACAN [67] an approach to use covert channel-based transmitter authentication by exploiting physical characteristics of communication was introduced. Among others, the inter-arrival times are specifically adapted to transmit information for the authentication of ECUs. While this option does not have a negative impact on bandwidth compared to the use of MACs, key management and the additional resource demand for resources for the calculations remain. Especially with the adoption of CAN with flexible data rate (CAN-FD), the issue of limited bandwidth for the transmission of MAC tags loses importance, as the maximum payload rises to 64 bytes.

Besides using sender identification methods it is also possible to use package-inspection in order to detect malicious behavior, like done by EVAD [21] or CASAD [48]. Rare cases and combinations, which have to be considered for training, make these systems prone to wrong decisions in operation [63]. Depending on the actual requirements there is either an additional cloud connection or a high-performance ECU necessary to establish and adapt the models according to the driver or environmental conditions. Apart from the voltage, signal characteristics are less dependent on the situation or driver and thus much easier to keep up-to-date. On the other hand, these systems are also able to detect attacks based on the payload of the frames, even if the identifier used is sent by a legitimate ECU. Therefore, we consider package-inspection as a complementary security measure to sender identification.

VII. Resource Requirements

The approach presented in this paper reduces the requirements, enabling the usage of comprehensive signal characteristics on resource-constrained platforms in real time. Previous approaches have considerably higher requirements, are not real-time capable even with high-end PC hardware [11] or leave this issue unclear [34], [12]. In order to compare the requirements, we consider the necessary memory to store the measurements in order to generate one fingerprint and the computational effort required for the calculation of the mean, a feature used by the compared approaches. In the following we assume a bus with a baud rate of 500 kb/s and a frame with a payload of 8 bytes.

In the approach of Choi *et al.* [11] the extended identifier is used and its signals are recorded with 2.5 GS/s and 12 bit accuracy. Due to the recommended extended identifier, three stuff bits are transmitted in addition, resulting in a total of 21 bits. Thus, only for the storage of the measurements 153.81 kB are needed, which is already 20 % more than the available memory of the platform used in this paper. The VoltageIDS [12] uses the signals which are transmitted after the arbitration phase. This results in 86 bits, recorded with at least 250 MS/s and 8 bit accuracy. Possible stuff bits are not considered here. Altogether, this results in a memory requirement of 41.99 kB. Scission [34] is comparable to the VoltageIDS, but operates with a lower sample rate of 20 MS/s, resulting in a memory requirement of 3.36 kB. Simple [17] samples initially the whole frame with 50 MS/s, including the bit fields before the payload in order to extract the identifier. Since this can easily be avoided by using the CAN controller, we also assume 86 bits here, which corresponds to a memory requirement of 8.4 kB. EASI samples ten edges two times, resulting in 20 measurements respectively a memory requirement of 20 bytes, which is a reduction by a factor of 168 compared to Scission [34].

An exact statement about the required computational resources is difficult to determine as the approaches are implemented in different programming languages, have been tested on different platforms or are not available. However, since the greatest effort lies in the calculation of the features, we consider the number of cycles required to calculate the mean value. This gives an estimation of how strong the calculation depends on the number of measurements. The results are shown in Table V, where it can be seen that our approach reduces the computational effort by a factor of 142 compared to Scission [34]. As mentioned in Section V a frame with 8 bytes of payload occupies the bus for a minimum of 222 μs at a bandwidth of 500kB/s. The calculation of the mean value with Scission alone requires 332 μs with the embedded platform used here, almost one and a half the time the bus is occupied by the frame. For comparison, our approach is completed with the entire feature calculation and classification after 96.62 μs. Since Simple [17] does not calculate comparable features, we cannot give a comparison for this metric. To ensure that this performance advantage is not ignored, we highlight Simple positively in the table.

TABLE V: Comparison of the considered approaches.

Approach	*Choi et al.* [11]	VoltageIDS [12]	Scission [34]	Simple [17]	EASI
Identification (%)	96.48	93.54	99.85	99.1	99.98
False positives (%)	3.52	6.46	0	0.9	0
Sampling rate (MS/s)	2500	250	20	50	2
Improvement factor	1250	125	10	25	-
FP footprint (kB)	153.81	41.99	3.36	8.4	0.02
Improvement factor	7691	2150	168	420	-
Computation (cycles)	2.7 M	0.75 M	60 K	+	420
Improvement factor	6443	1782	142	+	-

Another part of the cost of implementing an approach for the use of comprehensive signal characteristics is defined by the required ADC. Higher sampling rates usually require an external ADCs, which leads to additional costs for the circuit and also for the actual ADC. While gigasample ADCs cost several 100 dollars, ADCs with few megasample are in the 10 dollar range. EASI can work without additional ADCs, since the required sample rate is often supported by current MCUs [30], [49], [59].

VIII. Discussion

A. Size of Frames

A disadvantage of the proposed procedure is that a minimum number of rising edges is required to acquire a sufficient number of samples. In order to obtain a representative signal curve comparable to a curve sampled with 20 MS/s, at least 10 rising edges are required at a sampling rate of 2 MS/s. The number of edges depends primarily on the transmitted data. However, the minimum available and usable edges can be determined so that a minimum sampling rate can be specified with regard to the data lengths. If the entire data space is known for a communication system, the required sampling rate can be determined.

The minimum number of usable rising edges depending on the length of the payload and the resulting minimum sampling rate is given in Table VI. Only the rising edges which are present after arbitration were counted, since previous ones may have been influenced by other bus participants. In addition to the data length and the actual data, the CRC field is also available, which in the worst case contains two rising edges. If these parameters are not fulfilled, the senders of messages containing too few rising edges may only be determined with limited accuracy. Thus, with 2 MS/s in the worst case, only frames containing at least 5 bytes of payload can be clearly assigned.

TABLE VI: Required sampling rate in MS/s dependent on the length of the payload in bytes.

Payload	0	1	2	3	4	5	6	7	8
Edges	2	4	6	8	9	11	11	13	15
Rate	10	5	3.4	2.5	2.3	1.9	1.9	1.6	1.4

B. Using an External ADC

Although we have presented a procedure which avoids the need for an additional ADC depending on the platform, it can still offer advantages to use an external ADC with a high sample rate in order to record a full edge directly. This is particularly the case if several buses are to be analyzed simultaneously by the same system, for example, if EASI is integrated in a gateway that connects several bus segments. Thus, one ADC can then be utilized for the buses, since the time of its usage per frame would be much smaller and the triggering would also require a less precise behavior. Except for the signal gathering phase, no changes would be necessary.

C. Falling Edge

In addition to the rising edge, the falling edge also provides characteristics which can be used for classification. Although the characteristics are not sufficient to achieve comparable results, they can be used to increase the detection rate and robustness. Depending on the selected features, however, the required computing power and memory utilization increases.

D. Limitations

The system detects intrusions when a malicious ECU sends messages with identifiers for which it is not authorized. When an attack can be carried out without this injury, the system presented is not able to detect it. This also applies if an attack is fed into the monitored bus via a gateway and the messages with the identifiers used can generally be sent via the gateway. This is based on the fact that the signals of the original sender are replaced by the signals of the gateway. Further, simple additional connected devices are recognized by detecting changes in the bus, as this significantly changes the characteristics of the known ECUs. If it is possible for an attacker to prevent this change, no alarm is triggered. Although this requires access to a specific vehicle, but the attacker has access to a similar vehicle of the same model with the same equipment, he can obtain some information about the characteristics of the target vehicle in advance. Following, it is possible with correspondingly complex hardware and effort to imitate special signals and thus respective characteristics. For this reason, in addition to the monitoring of the signal characteristics, the use of classical IDS, which work on the basis of frequency and packet analyses, is recommended.

E. Field of Application

By drastically reducing the resource requirements and the associated costs, it is possible to integrate the presented system at several points of a bus. Due to the varying measuring positions in the bus, different characteristics occur for the same frame, as shown in Figure 7. If several systems are operated in parallel, their results can be compared which leads to a reduction of wrong decisions In addition to being used as extensions for classic IDSs, the approach can also be integrated into gateways. This allows messages to be checked before being forwarded and discarded in the event of an attack. Thus it is possible to prevent the propagation of attacks over several segments without the need for further communication over potentially compromised bus segments to notify other ECUs.

Besides the automotive sector, CAN is also used in other areas, such as automation, medicine and rail. There are also several higher protocols based on CAN, such as CANopen [6] or SafetyBUS [51]. In principle, the same problems exist here, which makes EASI relevant for these areas as well. In addition, since the bandwidth provided by CAN will not be sufficient for future developments, CAN-FD [54] was introduced in 2012. CAN-FD enables an increased bandwidth of up to 2.5 Mbit/s in the automotive sector [22] and a payload between 8 and 64 bytes per message. Since the functionality remains unchanged, the proposed approach is also suitable for the next development stage of CAN. The higher payload also increases the number of rising edges, which can be used to reduce the sample rate.

IX. Conclusion

Intrusion Detection Systems are a promising technology to increase security in connected vehicles. Due to the missing sender identification of the CAN, the most used bus system, the IDSs miss the important information from which ECU received messages were sent. Also the evolution, CAN-FD, which will be introduced with the next vehicle generation, does not change this and can therefore benefit from this information as well. Besides the automotive sector, CAN is also used in other areas, such as automation, medicine and rail. The approach presented in this paper allows the sender to be identified by differences in signal characteristics caused by hardware. The evaluation based on a prototype and two production vehicles proved that a sender identification of over 99.9 % is also possible with resource-constrained hardware. Thus, it could also be shown for the first time that classification is also possible at low cost on the basis of comprehensive signal characteristics, while real-time requirements could be met. This was mainly achieved by strong simplifications of the individual phases, the reduced sampling rate and the small amount of data which has to be processed by the system. Nevertheless, we were able to further increase the identification rates, as the presented approach focuses on the characteristics which contain the most information for sender identification. In addition, we have specified an update approach and shown by simulation that it is able to adapt the model to potential changes. Besides attacks using compromised ECUs, the presented IDS is also able to detect intrusions by unmonitored and additional ECUs. It was also demonstrated to be robust against attackers who can influence the voltage level of compromised ECUs and the energy supply of the vehicle. Finally, we did a one-week test drive to demonstrate the robustness of the system under changing conditions and active consumers. Considering that an attacker can control all vehicle functions as soon as he has access to the internal communication, the approach presented here offers considerable potential for increasing the security and thus the safety of connected vehicles. Overall, EASI is the first sender identification approach exploiting comprehensive signal characteristics which enables the implementation on a realistic automotive platform.

References

[1] Arm Limited, "Cmsis dsp software library," https://github.com/ARM-software/CMSIS_5, 2018, version 5.4.0.

[2] AUTOSAR Development Partnership, *Specification of Diagnostic Communication Manager*, no. 4.3.1. [Online]. Available: https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_DiagnosticCommunicationManager.pdf

[3] ——, "Specification of module secure onboard communication," https://www.autosar.org/fileadmin/user_upload/standards/classic/4-3/AUTOSAR_SWS_SecureOnboardCommunication.pdf, Nov. 2016.

[4] O. Avatefipour, A. Hafeez, M. Tayyab, and H. Malik, "Linking received packet to the transmitter through physical-fingerprinting of controller area network," in *2017 IEEE Workshop on Information Forensics and Security (WIFS)*, Dec 2017, pp. 1–6.

[5] Z. Cai, A. Wang, W. Zhang, M. Gruffke, and H. Schweppe, "0-days & mitigations: Roadways to exploit and secure connected bmw cars," *Black Hat USA*, vol. 2019, p. 39, 2019.

[6] CAN in Automation, "Canopen the standardized embedded network," https://www.can-cia.org/canopen/.

[7] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno, "Comprehensive experimental analyses of automotive attack surfaces," in *Proceedings of the 20th USENIX Conference on Security*, ser. SEC'11. Berkeley, CA, USA: USENIX Association, 2011, pp. 6–6.

[8] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 1044–1055. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978302

[9] ——, "Fingerprinting electronic control units for vehicle intrusion detection," in *25th USENIX Security Symposium (USENIX Security 16)*. Austin, TX: USENIX Association, 2016, pp. 911–927. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[10] ——, "Viden: Attacker identification on in-vehicle networks," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '17. New York, NY, USA: ACM, 2017, pp. 1109–1123. [Online]. Available: http://doi.acm.org/10.1145/3133956.3134001

[11] W. Choi, H. J. Jo, S. Woo, J. Y. Chun, J. Park, and D. H. Lee, "Identifying ecus using inimitable characteristics of signals in controller area networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 6, pp. 4757–4770, 2018.

[12] 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 Transactions on Information Forensics and Security*, vol. 13, no. 8, pp. 2114–2129, Aug 2018.

[13] Consumer Watchdog, "Kill switch why connected cars can be killing machines and how to turn them off," https://www.consumerwatchdog.org/sites/default/files/2019-07/KILL%20SWITCH%20%207-29-19.pdf, 2019.

[14] T. Dagan and A. Wool, "Parrot, a software-only anti-spoofing defense system for the CAN bus."

[15] Federal Office for Information Security, "Tr-02102-1 cryptographic mechanisms: Recommendations and key lengths," Jan. 2018.

[16] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692

[17] M. Foruhandeh, Y. Man, R. Gerdes, M. Li, and T. Chantem, "Simple: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks," in *Proceedings of the 35th Annual Computer Security Applications Conference*, ser. ACSAC '19. New York, NY, USA: ACM, 2019, pp. 229–244. [Online]. Available: http://doi.acm.org/10.1145/3359789.3359834

[18] I. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures," in *9th USENIX Workshop on Offensive Technologies (WOOT 15)*. Washington, D.C.: USENIX Association, 2015. [Online]. Available: https://www.usenix.org/conference/woot15/workshop-program/presentation/foster

[19] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.

[20] B. Groza and S. Murvay, "Efficient protocols for secure broadcast in controller area networks," *IEEE Transactions on Industrial Informatics*, vol. 9, no. 4, pp. 2034–2042, Nov 2013.

[21] F. Guo, Z. Wang, S. Du, H. Li, H. Zhu, Q. Pei, Z. Cao, and J. Zhao, "Detecting vehicle anomaly in the edge via sensor consistency and frequency characteristic," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5618–5628, June 2019.

[22] F. Hartwich, "Can with flexible data-rate," *CAN in Automation*, 2012.

[23] A. Higbee, "Hack your car for boost and power!" DEF CON 15 Hacking Conference, Aug. 2007.

[24] T. Hoppe, S. Kiltz, and J. Dittmann, "Adaptive dynamic reaction to automotive it security incidents using multimedia car environment," in *2008 The Fourth International Conference on Information Assurance and Security*. New York, NY, USA: ACM, Sept 2008, pp. 295–298.

[25] ——, "Security threats to automotive can networks – practical examples and selected short-term countermeasures," in *Computer Safety, Reliability, and Security*, M. D. Harrison and M.-A. Sujan, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 235–248.

[26] L. Hoyong, "Automotive security challenges in autonomous driving systems," http://www.krnet.or.kr/board/data/dprogram/2260/H1_1_%C0%CC%C8%A3%BF%EB.pdf, ESCRYPT - Embedded Security KOREA, 2018, kRnet Conference.

[27] L. Huang, A. D. Joseph, B. Nelson, B. I. Rubinstein, and J. D. Tygar, "Adversarial machine learning," in *Proceedings of the 4th ACM workshop on Security and artificial intelligence*. ACM, pp. 43–58.

[28] P. J. Huber, *Robust Estimation of a Location Parameter*. New York, NY: Springer New York, 1992, pp. 492–518. [Online]. Available: https://doi.org/10.1007/978-1-4612-4380-9_35

[29] A. G. Illera, "Dude, wtf in my car?" DEF CON 21 Hacking Conference, Aug. 2013.

[30] Infineon Technologies, "Aurix^{TM} 32-bit microcontrollers for automotive and industrial applications," https://www.infineon.com/dgdl/Infineon-TriCore_Family_BR-BC-v01_00-EN.pdf?fileId=5546d4625d5945ed015dc81f47b436c7, 2019.

[31] International Organization for Standardization, *ISO 11898-1:2015 Road vehicles – Controller area network (CAN) – Part 1: Data link layer and physical signalling*, 2nd ed.

[32] ——, *ISO 11898-2:2016 Road vehicles – Controller area network (CAN) – Part 2: High-speed medium access unit*, 2nd ed.

[33] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *2018 IEEE Symposium on Security and Privacy (SP)*, New York, NY, May 2018, pp. 19–35.

[34] M. Kneib and C. Huth, "Scission: Signal characteristic-based sender identification and intrusion detection in automotive networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 787–800. [Online]. Available: http://doi.acm.org/10.1145/3243734.3243751

[35] P. Koopman, J. Morris, and P. Narasimhan, "Challenges in deeply networked system survivability," *NATO SECURITY THROUGH SCIENCE SERIES D-INFORMATION AND COMMUNICATION SECURITY*, vol. 2, p. 57, 2006.

[36] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, and S. Savage, "Experimental security analysis of a modern automobile," in *2010 IEEE Symposium on Security and Privacy*. New York, NY: IEEE, May 2010, pp. 447–462.

[37] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *2008 IEEE Intelligent Vehicles Symposium*, June 2008, pp. 220–225.

[38] D. Larsson and J. Hemlin, "Exploring the generic timer modules feasibility for truck powertrain control," http://publications.lib.chalmers.se/records/fulltext/219127/219127.pdf, 2015.

[39] C. W. Lin and A. Sangiovanni-Vincentelli, “Cyber-security for the controller area network (can) communication protocol,” in *2012 International Conference on Cyber Security*. New York, NY: IEEE, Dec 2012, pp. 1–7.

[40] Microchip Technology Inc., “Mcp2515 stand-alone can controller with spi interface,” http://ww1.microchip.com/downloads/en/DeviceDoc/MCP2515-Stand-Alone-CAN-Controller-with-SPI-20001801J.pdf, Apr. 2005, revision J.

[41] ——, “Mcp2551 high-speed can transceiver,” http://ww1.microchip.com/downloads/en/devicedoc/21667e.pdf, Jan. 2007, revision E.

[42] C. Miller and C. Valasek, “Adventures in automotive networks and control units,” DEF CON 21 Hacking Conference, Aug. 2013.

[43] ——, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, p. 91, 2015.

[44] S. Mittal, “A survey of architectural techniques for managing process variation,” *ACM Comput. Surv.*, vol. 48, no. 4, pp. 54:1–54:29, Feb. 2016. [Online]. Available: http://doi.acm.org/10.1145/2871167

[45] H. Mori, Y. Suzuki, N. Maeda, H. Obata, and T. Kishigami, “Novel ringing suppression circuit to increase the number of connectable ecus in a linear passive star can,” in *International Symposium on Electromagnetic Compatibility - EMC EUROPE*. New York, NY: IEEE, Sept 2012, pp. 1–6.

[46] P. S. Murvay and B. Groza, “Source identification using signal characteristics in controller area networks,” *IEEE Signal Processing Letters*, vol. 21, no. 4, pp. 395–399, April 2014.

[47] M. Müter, A. Groll, and F. C. Freiling, “A structured approach to anomaly detection for in-vehicle networks,” in *2010 Sixth International Conference on Information Assurance and Security*. New York, NY: IEEE, Aug 2010, pp. 92–98.

[48] N. Nowdehi, W. Aoudi, M. Almgren, and T. Olovsson, “Casad: Can-aware stealthy-attack detection for in-vehicle networks,” 2019.

[49] NXP Semiconductors, “Mpc5775k data sheet,” https://www.nxp.com/docs/en/data-sheet/MPC5775KDS.pdf, 2016, rev. 9.1 10/2016.

[50] P. J. Pupalaikis, “Random interleaved sampling,” http://cdn.teledynelecroy.com/files/whitepapers/wp_ris_102203.pdf.

[51] D. Reinert and M. Schaefer, *Sichere Bussysteme fuer die Automation*. Huethig, 2001.

[52] M. Ring, D. Frkat, and M. Schmiedecker, “Cybersecurity evaluation of automotive e/e architectures,” *2. ACM Computer Science in Cars Symposium*, 2018.

[53] Robert Bosch GmbH, “Can specification v2.0,” http://esd.cs.ucr.edu/webres/can20.pdf, 1991.

[54] ——, “Can with flexible data-rate specification version 1.0,” https://can-newsletter.org/assets/files/ttmedia/raw/e5740b7b5781b8960f55efcc2b93edf8.pdf, 2012, version 1.0.

[55] S. U. Sagong, X. Ying, A. Clark, L. Bushnell, and R. Poovendran, “Cloaking the clock: Emulating clock skew in controller area networks,” in *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS ’18. Piscataway, NJ, USA: IEEE Press, 2018, pp. 32–42. [Online]. Available: https://doi.org/10.1109/ICCPS.2018.00012

[56] T. C. Smith and E. Frank, *Statistical Genomics: Methods and Protocols*. New York, NY: Springer, 2016, ch. Introducing Machine Learning Concepts with WEKA, pp. 353–378. [Online]. Available: http://dx.doi.org/10.1007/978-1-4939-3578-9_17

[57] STMicroelectronics, “Reference manual rm0361 generic timer module specification,” https://www.st.com/content/ccc/resource/technical/document/reference_manual/group0/a7/9e/ba/61/bf/4e/4c/36/DM00091883/files/DM00091883.pdf/jcr:content/translations/en.DM00091883.pdf, 2016.

[58] ——, “Spc58eex, spc58nex 32-bit power architecture$^{\circledR}$ microcontroller for automotive asil-d applications,” https://www.st.com/resource/en/datasheet/spc58ne84c3.pdf, 2017.

[59] ——, “Stm32$^{\text{TM}}$32-bit mcu family leading supplier of arm$^{\circledR}$ cortex$^{\circledR}$-m microcontrollers,” https://www.st.com/resource/en/brochure/brstm32.pdf, 2018.

[60] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaniche, and Y. Laarouchi, “Survey on security threats and protection mechanisms in embedded automotive networks,” in *2013 43rd Annual IEEE/IFIP Conference on Dependable Systems and Networks Workshop (DSN-W)*, June 2013, pp. 1–12.

[61] C. Szilagyi and P. Koopman, “Flexible multicast authentication for time-triggered embedded control network applications,” in *2009 IEEE/IFIP International Conference on Dependable Systems Networks*, June 2009, pp. 165–174.

[62] Tencent Keen Security Lab, “Experimental security assessment of bmw cars: A summary report,” https://keenlab.tencent.com/en/whitepapers/Experimental_Security_Assessment_of_BMW_Cars_by_KeenLab.pdf, 2018.

[63] A. Tomlinson, J. Bryans, and S. Shaikh, “Towards viable intrusion detection methods for the automotive controller area network,” *2. ACM Computer Science in Cars Symposium*, 2018.

[64] Vector CANtech Inc., “Common high speed physical layer problems,” https://assets.vector.com/cms/content/know-how/_application-notes/AN-ANI-1-115_HS_Physical_Layer_Problems.pdf, 2003.

[65] M. Wolf and T. Gendrullis, “Design, implementation, and evaluation of a vehicular hardware security module,” in *Information Security and Cryptology - ICISC 2011*, H. Kim, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 302–318.

[66] M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the art: Embedding security in vehicles,” *EURASIP Journal on Embedded Systems*, vol. 2007, no. 1, p. 074706, Jun 2007. [Online]. Available: https://doi.org/10.1155/2007/74706

[67] X. Ying, G. Bernieri, M. Conti, and R. Poovendran, “Tacan: Transmitter authentication through covert channels in controller area networks,” in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, ser. ICCPS ’19. New York, NY, USA: ACM, 2019, pp. 23–34. [Online]. Available: http://doi.acm.org/10.1145/3302509.3313783