CAN-MIRGU: A Comprehensive CAN Bus Attack Dataset from Moving Vehicles for Intrusion Detection System Evaluation

Sampath Rajapaksha*, Garikayi Madzudzo†, Harsha Kalutarage*
Andrei Petrovski* and M.Omar Al-Kadri‡,
*Robert Gordon University, †Horiba Mira Ltd, ‡University of Doha for Science and Technology
{s.rajapaksha, h.kalutarage, a.petrovski}@rgu.ac.uk, garikayi.madzudzo@horiba-mira.com, omar.alkadri@udst.edu.qa

***Abstract*—The Controller Area Network (CAN Bus) has emerged as the de facto standard for in-vehicle communication. However, the CAN bus lacks security features, such as encryption and authentication, making it vulnerable to cyberattacks. In response, the current literature has prioritized the development of Intrusion Detection Systems (IDSs). Nevertheless, the progress of IDS research encounters significant obstacles due to the absence of high-quality, publicly available real CAN data, especially data featuring realistic, verified attacks. This scarcity primarily arises from the substantial cost and associated risks involved in generating real attack data on moving vehicles. Addressing this challenge, this paper introduces a novel CAN bus attack dataset collected from a modern automobile equipped with autonomous driving capabilities, operating under real-world driving conditions. The dataset includes 17 hours of benign data, covering a wide range of scenarios, crucial for training IDSs. Additionally, it comprises 26 physically verified real injection attacks, including Denial-of-Service (DoS), fuzzing, replay, and spoofing, targeting 13 CAN IDs. Furthermore, the dataset encompasses 10 simulated masquerade and suspension attacks, offering 2 hours and 54 minutes of attack data. This comprehensive dataset facilitates rigorous testing and evaluation of various IDSs against a diverse array of realistic attacks, contributing to the enhancement of in-vehicle security.**

I. Introduction

Advancements in automotive technology have equipped modern vehicles with advanced features designed to enhance the comfort and safety of both drivers and passengers. These include features such as automated parking assistance, lane departure warning, adaptive cruise control, and infotainment systems [1]. Modern automobiles incorporate a large number of electronic control units (ECUs) to facilitate these advanced features. Effective communication among these ECUs is essential for exchanging real-time information. This necessitates a unified network that facilitates near real-time data transmission, ensuring sufficient bandwidth and reliable performance [2]. The CAN bus, a message-based communication protocol, fulfils these requirements due to its advantages, including low cost, support for a maximum bus speed of 1000 kbps, lightweight design, and robustness [3]. Therefore, the CAN bus has become the de facto standard for in-vehicle communication. However, the CAN bus lacks security features such as encryption and authentication [4]. Moreover, its utilization of an ID-based priority mechanism and broadcast transmission makes the CAN bus susceptible to cyberattacks [5]. Researchers in automobile cybersecurity have demonstrated the capability of exploiting weaknesses in the CAN bus across various vehicle brands [6]–[8]. Gaining physical control of the vehicle by an attacker not only jeopardizes vehicle functions but also compromises the safety of vehicle passengers.

In response to numerous vulnerabilities and the increasing threat of cyberattacks on modern automobiles, significant efforts have been directed towards safeguarding vehicles from such attacks. Approaches involving detection and prevention mechanisms are employed to identify or prevent in-vehicle cyberattacks. Nevertheless, detection strategies are deemed more feasible considering operational and economic considerations [9]. Consequently, as a reactive security measure, current literature has prioritized the development of IDSs explicitly for the CAN bus. Based on the detection strategy, IDSs can be categorized as signature-based detection systems and anomaly-based detection systems [10]. Due to the constraints in signature-based IDS, such as the challenge of detecting novel attacks and the need for frequent updates to the known attack database, the majority of previous works have focused on anomaly-based detection approaches [5]. Anomaly detection-based IDSs can be further classified as statistical approaches, frequency or time-based methods, and machine learning approaches [10]. These anomaly-based IDSs model the normal behaviour of CAN bus data using known benign datasets and utilize learned patterns or statistical metrics to identify anomalies, thereby highlighting their reliance on the availability and quality of CAN bus datasets.

Despite the recent increase in focus and publication of IDSs on the CAN bus [5], [11] the advancement of IDS research faces significant obstacles due to the lack of high-quality, publicly available real CAN data that includes realistic attack scenarios [12]. This is mainly due to the considerable cost and associated risks involved in generating real attack data on moving vehicles. The use of a real CAN dataset for model training, validation, and testing is crucial for the development of an effective IDS capable of detecting a wide range of attacks in real-world conditions. However, many proposed IDSs rely on self-collected datasets that are not accessible to other researchers [11]. Furthermore, the widely used car hacking

Symposium on Vehicles Security and Privacy (VehicleSec) 2024
26 February 2024, San Diego, CA, USA
ISBN 979-8-9894372-7-6
https://dx.doi.org/10.14722/vehiclesec.2024.23043
www.ndss-symposium.org

dataset [13], despite being a popular public benchmark, has a significant drawback: benign data was collected during vehicle movement, while attack data was collected when the vehicle was stationary [5]. In an attempt to address these challenges, a more advanced dataset has been introduced in [12]. However, it is important to note that this dataset focuses on a limited number of IDs, and the vehicle was on a dynamometer during the collection of attack data.

To the best of the author's knowledge, there is currently no publicly available CAN bus dataset that includes physically verified attacks collected during real-world driving conditions. In light of this gap, we present CAN-MIRGU, a real CAN bus dataset obtained from a modern automobile aiming to propel advancements in IDS research within in-vehicle networks. The primary contributions of this paper can be outlined as follows:

1) **Generating a CAN bus attack dataset while the vehicle is in motion under real-world conditions:** This paper introduces CAN-MIRGU, a novel and publicly available CAN bus attack dataset collected from a modern automobile equipped with autonomous driving capability, operating under real-world driving conditions. This dataset includes physically verified attacks, addressing the existing gap in publicly accessible datasets featuring realistic attacks in dynamic driving scenarios.
2) **Comprehensive training and testing dataset:** The dataset includes 17 hours of benign data collected under diverse driving conditions to train IDSs with ample and varied data, enhancing their capacity to recognize normal driving behaviour. Moreover, it incorporates attack data with extended duration to assess IDS resilience under adversarial learning.
3) **In-depth dataset analysis:** This includes a thorough analysis of the dataset, offering insights to better understand both the benign and attack data. The availability of this detailed analysis provides valuable information for researchers and practitioners to gain a comprehensive understanding of the dataset's characteristics.

The rest of this paper is structured as follows: Section II provides the preliminaries. Section III presents the publicly available CAN datasets. Section IV introduces the CAN-MIRGU dataset along with the analysis. The discussion of the observed behaviours during the experiments is presented in section V. Finally, section VI concludes the paper.

II. Preliminaries

A. Controller area network (CAN bus)

CAN operates as a lightweight broadcast-based communication protocol, and a CAN data frame comprises seven fields that facilitate data transmission. These fields include Start of Frame (SOF), Arbitration Field (CAN ID), Control Field (DLC), Payload (data), Cyclic Redundancy Code (CRC), Acknowledgment (ACK), and End of Frame (EOF), as illustrated in Figure 1 along with their respective bit-lengths. Among these, CAN ID and payload hold particular significance within the CAN frame for attack detections [14]. The CAN ID functions as a message identifier, prioritizing messages based on their ID values, where lower IDs receive higher priority and vice versa. This prioritization is used to manage concurrent messages on the CAN bus. CAN payload values contain the information intended for transmission over the network and support data transmission of up to 64 bits (8 bytes). The specifications of the CAN frame are stored in a file known as CAN DataBase (DBC), which is not publicly available [15]. Furthermore, these specifications vary based on the vehicle's make, model, year, and trim [14].

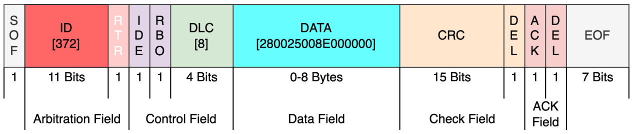


Fig. 1: CAN bus data frame with example values for ID and payload fields

B. CAN bus attacks

The broadcast nature of the CAN bus results in all ECUs receiving unencrypted messages transmitted across the bus. This characteristic makes the CAN bus susceptible to a sniffing attack, wherein an attacker can eavesdrop on all messages, recording them for subsequent analysis of the CAN data. The absence of authentication further amplifies the vulnerability, allowing any node to transmit a frame to the CAN network. This allows an attacker to inject malicious frames by exploiting a compromised ECU. In addition to these vulnerabilities, attackers can leverage the ID-based priority mechanism to inject frames with higher priority IDs, potentially leading to a DoS attack.

Attacks on the CAN bus can be mainly classified into three categories as injection (fabrication), suspension and masquerade (impersonation) attacks [1]. Injection attacks involve introducing new malicious frames into the CAN bus. Common injection attacks on the CAN bus include:

- **DoS attacks:** DoS attacks try to make communication services unavailable by sending a large number of frames. In the context of the CAN bus, attackers can continuously transmit frames with low CAN IDs, particularly those assigned the highest priority. Figure 2a illustrates a DoS attack on the CAN bus. The presence of the high-priority CAN ID 0x000 may introduce delays for frames with CAN ID 0x372, transmitted by ECU B. Such delays have the potential to induce unexpected behaviour in the vehicle.
- **Fuzzing attacks:** In a fuzzing attack, a malicious node floods the network with a large number of frames, employing randomly generated IDs and malicious payloads to mimic legitimate frames. Two variations of this attack exist: injecting CAN IDs that appear during normal traffic (valid IDs) and injecting entirely new, randomly generated CAN IDs. The fuzzing attack on the CAN bus is depicted in Figure 2b. The attacker, ECU A, transmits randomly generated CAN IDs 0x450 and 0x460, causing the receiver ECU C to read and utilize information from these malicious frames. Attackers may execute this attack with prior knowledge of CAN frames, acquired through CAN

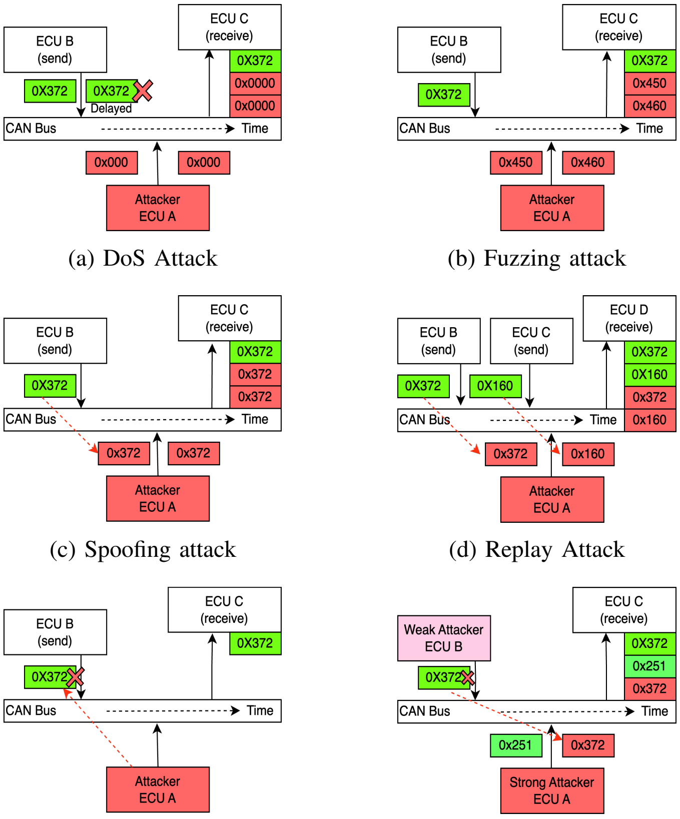
bus sniffing or as a black-box attack without prior knowledge of CAN frames.

- **Spoofing attacks (Targeted ID attack):** In a spoofing attack, the attacker targets specific CAN IDs to introduce malicious messages. Figure 2c illustrates the spoofing attack that attacker ECU A targets CAN ID 0x372 of ECU B. In this scenario, alongside the legitimate frame transmitted by ECU B, ECU C receives additional frames with the same ID but manipulated payloads. Consequently, ECU C might respond based on the malicious data.
- **Replay attacks:** In replay attacks, attackers capture and resend previously valid frames at different times. For instance, previously recorded speedometer values may be transmitted at a later time. Figure 2d illustrates a replay attack where attacker ECU A transmits CAN IDs belonging to both ECU B and ECU C.

Executing DoS and fuzzing attacks with randomly generated CAN IDs does not necessitate prior knowledge of the CAN bus. However, executing a fuzzing attack with existing CAN IDs and replay attacks requires limited prior knowledge, which can be acquired through the CAN bus sniffing. In contrast, a spoofing attack demands advanced knowledge of the CAN specification, particularly when targeting specific vehicle functionalities. One significant challenge with injection attacks is message confliction [12], [16]. As the attacker injects malicious frames, legitimate ECUs continue to send messages, leading to conflicts. The ECU's response to message confliction varies; simpler ECUs, like speedometers, might react based on straightforward algorithms, such as considering the last received message or utilizing queuing algorithms. Complex ECUs, on the other hand, might choose to ignore conflicting messages or disable certain features if they are not safety-critical [16]. An effective approach to overwrite legitimate messages with the target ID involves injecting malicious frames with the same ID immediately after the appearance of the legitimate frame, a technique known as flam delivery [12]. In terms of detection, DoS and fuzzing attacks are generally more easily detectable with frequency-based IDSs since these attacks alter the inter-arrival time of frames or disrupt the sequential behaviour of CAN IDs. Detecting spoofing and replay attacks might pose more challenges based on the attacker's strategy, as these attacks do not require frequent injections compared to DoS and fuzzing attacks.

In contrast to injection attacks, both suspension and masquerade attacks do not introduce additional frames into the CAN bus during the attack period. Suspension attacks involve compromising an ECU, preventing it from transmitting messages for a specific duration, as depicted in Figure 2e. In this scenario, the attacker compromises ECU B, suspending its transmission of frames with ID 0x372, thereby disrupting other ECUs reliant on messages from ECU B. The detection capability of suspension attacks may depend on the targeted IDs and the suspension time. In a masquerade attack, the attacker suspends an ECU and then utilizes a strongly compromised ECU to transmit malicious frames with the same ID and frequency. For instance, as illustrated in Figure 2f, the attacker can monitor and learn about message IDs and their frequencies from the weak attacker ECU B (ID 0x372). Subsequently, the attacker suspends ECU B's message transmission, allowing ECU A to transmit a fabricated message representing ECU B. Masquerade attacks avoid message conflict by preventing the appearance of multiple frames with the same ID, thus adhering to the frequency behaviour of the ID. Executing a masquerade attack demands an expert level of hacking expertise and comprehensive knowledge about the vehicle [12]. Consequently, no publicly available real CAN masquerade attack dataset exists. Similar to the complexity of the attack, detecting masquerade attacks may necessitate an advanced IDS utilizing the payload field to discern these sophisticated intrusions.

To execute these attacks, an attacker needs to gain access to the CAN bus. In practice, this can be achieved by connecting an On-Board Diagnostic (OBD-II) dongle while the vehicle is parked or by obtaining remote access [16]. Remote attack surfaces include the anti-theft system, tire pressure monitoring system (TPMS), remote keyless entry, Bluetooth, radio systems, Wi-Fi, and telematics units [16], [17]. Various previous experimental research studies have demonstrated the feasibility of these attacks in real-world conditions [6], [8], [16], [17]



(a) DoS Attack

(b) Fuzzing attack

(c) Spoofing attack

(d) Replay Attack

(e) Suspension Attack

(f) Masquerade attack

Fig. 2: CAN bus attacks

III. Related Work

This section introduces publicly available CAN datasets featuring attacks. It includes descriptions of the attacks with the benefits and drawbacks of each dataset.

A. Car hacking dataset for the intrusion detection (HCRL CH) [13]

This was released by the Hacking and Countermeasure Research Lab for academic use. The dataset comprises only 500-second benign data along with four datasets corresponding to distinct attack types: DoS, fuzzing, and two spoofing attacks (RPM and gear). Each attack dataset includes 300 instances of message injection lasting 3-5 seconds, captured over 30-40 minutes. These attacks significantly altered ID frequencies, making them easily detectable through frequency-based or sequence-based approaches. Experimental results from various studies consistently demonstrate high accuracy, achieving an F1-score of over 99% for all attacks due to the simplicity and unrealistic nature of the data [3], [18], [19]. While benign data collection occurred during driving, signal decoding revealed that the car was stationary during attack data collection [12]. Additionally, benign data and attack data are stored in different file formats. These limitations make this dataset unsuitable for evaluating an IDS, particularly those developed using AI methods.

B. CAN dataset for intrusion detection (HCRL OTIDS) [20]

This dataset, created by HCRL in conjunction with their remote frame-based CAN IDS [21], utilizes a KIA SOUL vehicle for collecting benign, DoS, fuzzy, and impersonation (masquerade) attack data. It is the only publicly accessible CAN dataset featuring remote frames and responses. However, unlike the car hacking dataset, it lacks labels (ground truth) as an attribute. Instead, the documentation provides attack injection intervals, though these are deemed inaccurate [12] and are insufficient for labelling fuzzy and impersonation attacks due to a lack of details such as injected IDs. Furthermore, based on their documentation, the masquerade attack in this dataset does not align with actual masquerade attacks, as it involves message injection.

C. Survival analysis dataset for automobile IDS (HCRL SA) [22]

HCRL released this dataset with their frequency-based CAN IDS [23]. Notably, it stands as the only publicly available CAN dataset featuring real attacks on multiple vehicles, namely the HYUNDAI YF Sonata, KIA Soul, and CHEVROLET Spark. For each vehicle, the dataset encompasses benign data and three distinct attack types: flooding (DoS), fuzzing, and malfunction (spoofing) attacks. However, it's essential to note that these attacks are basic and can be easily detected using frequency-based or sequence-based IDS due to their impact on significant frequency changes. Furthermore, the benign datasets related to each vehicle are limited to 60-90 seconds, which may not be sufficiently large for training a robust IDS.

D. Car hacking attack and defence challenge (HCRL CHDC) [24]

HCRL collected this dataset utilizing a Hyundai Avante CN7 for a competition focused on advancing attack and detection methodologies for CAN bus systems. The dataset comprises benign, flooding (DoS), spoofing, replay, and fuzzing attacks, with timestamp, ID, Data Length Code (DLC), payload, label, and sub-class (indicating attack type) as data attributes. Unlike other HCRL datasets where attack datasets were stored in separate files, here, both benign and four types of attacks coexist in the same file. Despite the presence of benign data interspersed between attacks, the benign dataset is notably limited and may not offer sufficient data for effective algorithm training.

E. SynCAN dataset [25]

This synthetic dataset was released with the CAN IDS CANet [26]. The primary objective of this dataset is to train unsupervised CAN IDS. Widely utilized in the literature for evaluating unsupervised payload-based IDSs [1], [26], [27], it stands out by providing signal values without the raw CAN data. This characteristic makes it particularly suitable for testing signal-based IDSs. The dataset comprises training data and six test datasets, featuring one normal dataset and five attack datasets. The attacks are categorized as plateau, continuous, playback, suppress, and flooding. Notably, these attacks are synthetic and cannot be validated for their impact on a real vehicle. It's worth mentioning that this dataset encompasses only 10 CAN IDs with a maximum of four signals, which is relatively limited compared to modern vehicles.

F. TU Eindhoven CAN bus intrusion dataset [28]

This dataset, released by the Department of Mathematics and Computer Science at Eindhoven University of Technology, utilizes two cars (Opel Astra and Renault Clio) and a CAN bus prototype to gather benign data. Attacks are synthetic and consist of diagnostic, fuzzing, replay, suspension, and DoS attacks. However, the manipulation of CAN message timestamps during the post-processing stage makes this dataset unsuitable for testing CAN IDSs that rely on time as a critical feature.

G. Real ORNL Automotive Dynamometer (ROAD) CAN intrusion dataset [12]

This real dataset includes an advanced set of attacks, comprising 13 unique attacks and 12 benign datasets covering various driving scenarios. The data were collected using a single vehicle and included fuzzing, targeted ID, and accelerator attacks. Fuzzing attacks injected random IDs, while targeted ID attacks utilized four variations: correlated signal, max speedometer, max engine coolant temperature, and reverse light. Accelerator attacks induced a compromised mode in the ECU. Masquerade attack versions were generated for targeted ID attacks by removing legitimate messages during post-processing. Although labels are unavailable, attack IDs and intervals are provided, aiding in identifying attack messages. Regarded as one of the most comprehensive CAN datasets, it enables the evaluation and comparison of CAN IDSs against realistic attacks. Despite its advantages, this dataset has several drawbacks. The benign data collection utilized both roads and a dynamometer. However, when collecting attack data, the vehicle was exclusively on a dynamometer. This difference in data collection environments may introduce variations compared to actual road driving scenarios. In the obfuscation process, intentional changes were made to the order of CAN IDs, resulting in the removal of priority information. This limitation restricts the applicability of this dataset for IDSs

Dataset	Real/Synthetic	Attacks	DoS	Fuzzing	Replay	Spoofing	Suspension	Masquerade	Benign duration	Attack duration	Labeled
HCRL CH	Real	4	✓	✓	-	✓	-	-	0h 8m 20s	7h 21m 57s	Yes
HCRL OTIDS	Real	3	✓	✓	-	-	-	✓	0h 17m 17s	0h 18m 56s	No
HCRL SA	Real	9	✓	✓	-	✓	-	-	0h 3m 31s	0h 8m 53s	Yes
HCRL CHDC	Real	4	✓	✓	✓	✓	-	-	-	0h 23m 23s	Yes
SynCAN	Synthetic	5	✓	✓	✓	✓	✓	-	-	-	Yes
TU Eindhoven	Synthetic	5	✓	✓	✓	-	✓	-	0h 19m 20s	0h 8m 17s	Yes
ROAD	Real	13	-	✓	-	✓	-	✓	3h 0m 32s	0h 27m 10s	No
CAN-MIRGU	Real	36	✓	✓	✓	✓	✓	✓	17h 8m 10s	2h 54m 56s	Yes

TABLE I: Publicly available CAN attack datasets. **Attacks:** indicating the count of distinct attack captures available in the dataset. For the SynCAN dataset, the duration of both benign and attack periods cannot be accurately determined using the provided timestamps.

that rely on ID priority information. The dataset comprises 106 CAN IDs in the vehicle, yet during targeted ID attacks, only two high-priority IDs and one low-priority ID were focused on. This limitation hinders the evaluation of IDS capability to detect attacks on various IDs, particularly those of low and medium frequency. Given the 106 ECUs, acquiring a large dataset is necessary to effectively learn the normal behaviour of the vehicle, surpassing the available 3-hour benign dataset. Additionally, attack datasets last only a few seconds for each targeted ID attack, imposing constraints on the thorough evaluation of an IDS.

Each publicly available CAN dataset has its own set of limitations. These include a lack of sufficient data to effectively learn normal behaviour for anomaly detection, a focus on only a few CAN IDs (ECUs) during attacks, significant differences in driving conditions between benign data collection and attack data collection, and the use of high-frequency injection for attacks, making them easily detectable even with a simple time-based detector. Importantly, none of the existing attack datasets were generated by targeting a moving vehicle in authentic driving scenarios. Our dataset addresses these drawbacks by providing a diverse range of attacks that target different high, medium, and low-frequency IDs, all conducted under real-world driving conditions using a modern automobile. Additionally, it includes a large benign dataset suitable for advanced IDS training. The comparative table for publicly available CAN attack datasets is presented in Table I.

IV. CAN-MIRGU DATASET

This section details the experimental setup employed for collecting both benign and attack data in our dataset, named CAN-MIRGU. It delineates the procedures for the attacks, describes the vehicle's responses to each attack, and offers an analysis of both benign and attack data. The dataset is accessible through the following link: https://github.com/sampathrajapaksha/CAN-MIRGU.

A. Dataset collection setup

We utilized a modern automobile manufactured in 2016, and while we do not disclose the specific make and model, it is a fully electric vehicle equipped with full autonomous driving capabilities. To mitigate risks associated with executed attacks, the autonomous driving mode was deactivated, and professionally trained drivers were engaged for both attack and benign data collection. The CAN data was captured using SocketCAN utilities[1] on a Linux laptop, employing the candump command. For data logging, a Kvaser Memorator 2xHS v2 was connected to the laptop using a standard USB 2.0 cable. In contrast to previous CAN data collection methods that involved connecting the CAN data logger directly to the OBD-II port [14], [29], we encountered limitations as only diagnostic messages were accessible through the OBD-II port of the vehicle in use. Consequently, the CAN data logger was directly connected to the CAN gateway to facilitate comprehensive data collection and injection. The candump speed was configured to 500 Kbps to align with the high-speed CAN bus. For injecting attack frames, Python-can[2] along with the cansend command in can-utils were employed, utilizing Python 3.9.

For the benign data collection, the vehicle was driven mimicking the normal driving behaviour of an average driver on public roads in the UK to include various benign driving activities. Significantly, these benign datasets were collected over a six-week period to account for any normal variations in the data arising from diverse conditions or the natural wear and tear of vehicle components. As a result, our benign dataset offers a more realistic representation of normal driving behaviours compared to other publicly available datasets. Given the inherent risks of these injection attacks, the vehicle was driven at a maximum speed of 30 mph on the 750-acre proving ground belonging to our industry partner Horiba MIRA during the attack data collection. Safety protocols were rigorously adhered to during the attacks, especially in situations affecting critical functions like steering.

B. Attack scenarios

Injection attacks, including DoS, fuzzing, spoofing, and replay, were executed for specific IDs. To ensure a comprehensive evaluation of IDS across various IDs, we targeted five high-frequency IDs, including 2B0, 160, 251, 371, 372, five low-frequency IDs, including 07F, 50C, 559, 541, 593, and three medium-frequency IDs, including 381, 386, 394. Prior to and after each injection attack, benign datasets were collected, allowing for the evaluation of IDS performance on both benign and attack data. Below are descriptions of the attack scenarios. Comprehensive details for each attack and message timing analysis are listed in Table II and Table III in subsection IV-C.

1) DoS attack: Given that CAN ID 0x000 holds the highest priority, it was employed to execute the DoS attack using the maximum payload (FFFFFFFFFFFFFFFF). Frames with ID 0x000 were injected every 0.001s. However, no reactions to the attack were observed during this period. This lack of

[1] https://github.com/linux-can/can-utils

[2] https://python-can.readthedocs.io/en/stable/

response could stem from CAN ID 0x000 not being a valid ID for this vehicle or possibly from a violation of the checksum mechanism used by this vehicle.

2) Fuzzing attack: There are two variations of the fuzzing attack, each performed with different IDs. In the fuzzing attack with random IDs, an ID was randomly selected from the range of 0x000 to 0x255 and injected with the maximum payload, similar to the DoS attack, which is a valid payload according to the CAN specification. This led to the observation of a few warning lights on the dashboard and occasional warning sounds. In the other variation, randomly selected valid IDs were injected with the maximum payload, resulting in more warning messages. For both variations, frames were injected every 0.02s.

3) Replay attack: This dataset includes three replay attacks, where previously transmitted payloads were injected into unusual contexts using flam delivery. In these instances, the injected frames were placed in situations or sequences that deviated from their original context or intended use. These include steering angle replay attack, Engine Management System (EMS) replay attack, and EMS replay long attack. No visible changes were observed during the replay attacks.

4) Spoofing attacks: The majority of the attacks in the CAN-MIRGU dataset are spoofing attacks. Both flam delivery and time-based injection were employed for different attacks depending on the targeted ID and payload. These attacks encompass various scenarios such as steering angle, brake and fog light, brake warning, drive mode changing, Forward Collision Avoidance Assist (FCA) warning, power steering, max speedometer, three variations of min speedometer, wiper warning, EMS, parking brake, two variations of gear shifter, and door open warning attacks. All of these attacks involve a single attack window that spans over a few seconds or a few minutes. Additionally, there are four attack datasets with multiple attack windows, including fuzzing valid IDs and DoS attacks with two attack windows, reverse speedometer and fuzzing attacks with two attack windows, and two variations of multiple attacks with three and six attack windows.

5) Suspension attack: Using small benign datasets, we simulated five suspension attacks by removing legitimate target ID frames for a specific period of time. This simulation replicates the suspension of an ECU. The selected IDs for these attacks are 160, 371, 386, 541, and 07F, covering high, medium, and low-frequency IDs.

6) Masquerade attacks: This was simulated by employing five selected real spoofing attack captures that utilized flam delivery as the attack technique. Similar to the approach used in [12], we removed the legitimate target ID frames preceding each injected frame to create more advanced versions of the attacks. This approach eliminates message confliction in the data, creating the appearance that only the spoofed messages are present during the injection interval. The selected spoofing attacks used to produce masquerade attack versions are break warning, steering angle, wiper warning, min speedometer, and break and fog light attacks. While masquerade attacks are simulated through post-processing, the impact of the malicious frames employed in these attacks was physically verified during real attacks. Consequently, it is expected to yield a similar effect to the real attack.

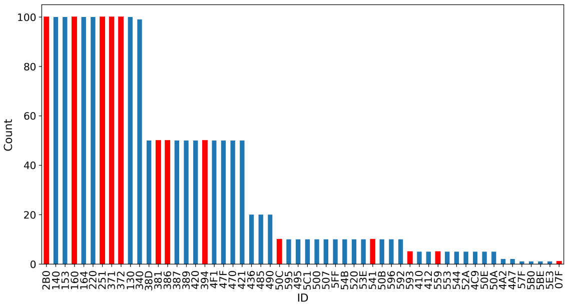


Fig. 3: Average number of ID counts for one second driving. Targeted IDs for attacks are shown in red bars.

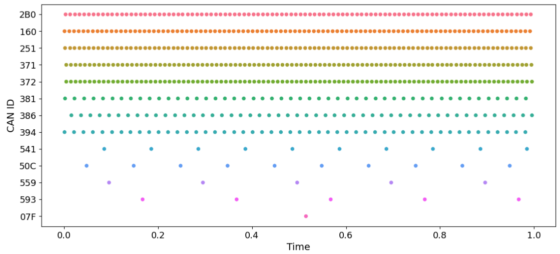


Fig. 4: Frame transmission over one second for the targeted IDs

Table II and Table III present a comprehensive summary of attacks along with visualizations of message timing. These visualizations illustrate the inter-message arrival times between all messages and the transmission of frames for the injected messages. They provide insights into how the malicious frames impact these times based on the attack technique, whether flam or time-based injection. The targeted ID message timing plots reveal changes only in the transmission of injected frames, while other ID transmissions remain unchanged. Therefore, frequency or time-based IDSs should leverage ID-level information for enhanced detection rates. For masquerade attacks, the transmission of targeted ID messages mirrors that of benign messages, posing a challenge for time-based IDSs, which might struggle to detect these attacks (see the targeted ID message timing for break warning masquerade attack). Conversely, in the case of suspension attacks, the targeted ID's frame transmission halts during the attack period, as depicted in the targeted ID message timing for the ID 160 suspension attack. These findings offer valuable insights for designing IDSs capable of detecting attacks with lower latency and higher detection rates.

C. Benign and attack data analysis

CAN-MIRGU dataset comprises 26 real injection attacks and 10 simulated attacks for suspension and masquerade attacks, totalling 36 attacks that targeted 13 IDs out of the total 56 CAN IDs. The real injection attack captures span over a duration of 2 hours, 9 minutes, and 16 seconds, while suspension attacks span for 26 minutes and 16 seconds, and masquerade attacks span for 19 minutes and 24 seconds. Additionally, the dataset includes 17 hours of benign data, providing a substantial dataset for training an IDS to learn

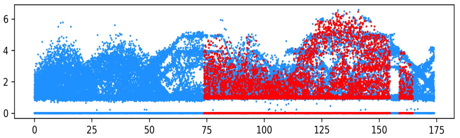
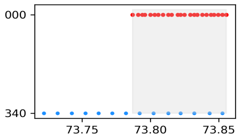
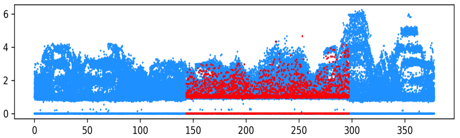
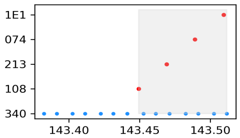
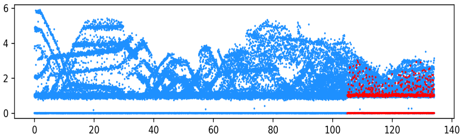
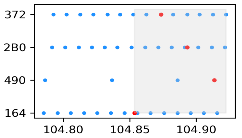
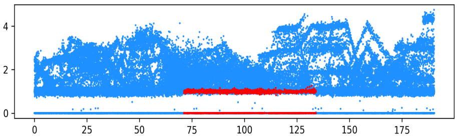
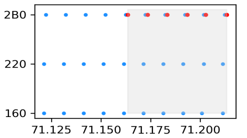
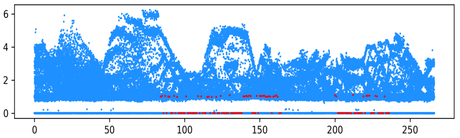
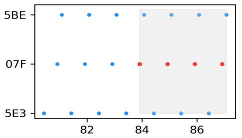
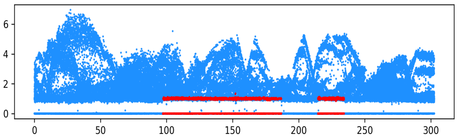
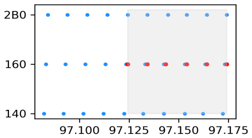
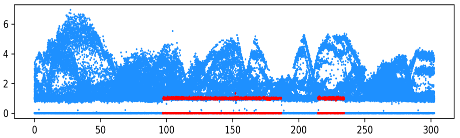
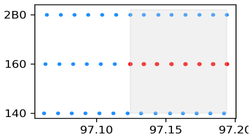
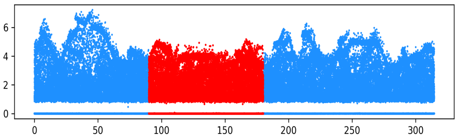
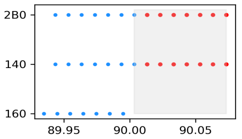
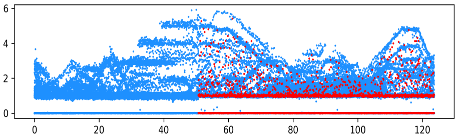
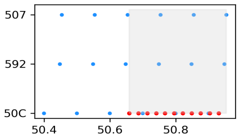
Attack	Observations	Message Timing	Targeted ID Message Timing
DoS ★ 000#FFFFFFFFFFFFFFFF 153.704584 Injecting every 0.02s	No visible changes.		
Fuzzing random IDs ★ XXX#FFFFFFFFFFFFFFFF 153.704584 Injecting every 0.02s	Few warning lights on the dashboard and occasional warning sounds.		
Fuzzing valid IDs ★ XXX#FFFFFFFFFFFFFFFF 134.016241 Injecting every 0.02s	'Check FCA (Forward Coll. Avoidance Assist)' warning message, parking brake, and ABS indicators on the dashboard. 'Harness Relay Malfunction' warning message on the lane detection display and continuous warning sounds.		
Steering angle ★ 2B0#XXAAXXXXXX 190.264094 Flam	'Check FCA(Forward Coll Avoidance Assist)' warning message on the dashboard and continuous warning sounds.		
Break and fog light ★ 07F#XXC3XXXXXXXXXXXX 266.008802 Flam	'Check brake light' and 'Check fog light' warning messages on the dashboard and continuous warning sounds.		
Break warning ★ 160#02AAXXXXXXXXXXXX 266.008802 Flam	'Stop vehicle and check breaks' warning message on the dashboard and continuous warning sounds.		
Break warning masquerade ★ 160#02AAXXXXXXXXXXXX 266.008802 Masquerade	Simulated attack. It is expected to have a comparable impact to the brake warning attack.		
ID 160 suspension ★ 160#XXXXXXXXXXXXXXXX 314.518042 Suspension	Simulated attack.		
Drive mode changing ★ 50C#FF05FFFF24FFFFE0 123.605818 Injecting every 0.02s	Continuously switching between normal, sport, eco and eco+ driving modes for a few seconds and stabled at eco+.		

TABLE II: Description of attacks. **Attack:** This column provides information on the attack name, attack severity, injected ID and payload, attack duration in seconds, and attack technique. Severity of the attack is categorized with ★ for no impact, ★ for warnings, and ★ for significant behavior alteration. **Message Timing:** The subplots display inter-message arrival time between all messages, where the x-axis represents time in seconds and the y-axis represents inter-message arrival time in milliseconds (ms). Blue dots and red dots indicate benign and attack frames, respectively. **Targeted ID Message Timing:** The subplots represent the transmission of frames for the injected ID near the attack start. The attack area is shaded. The x-axis represents the inter-arrival time for the injected ID, and the y-axis represents CAN IDs, using two same frequency CAN ID (blue dots) for comparison. For DoS and fuzzing random IDs attacks, where no particular ID was targeted, the same ID (340) is used for the comparison.

the normal behaviour of the vehicle. The average number of ID counts for one second of benign driving data is illustrated in Figure 3. The IDs selected for the attacks are highlighted in red bars. Based on this, the selected targeted IDs range from the highest frequent ID 2B0 to the lowest frequent ID 07F. This facilitates the evaluation of IDS performance against

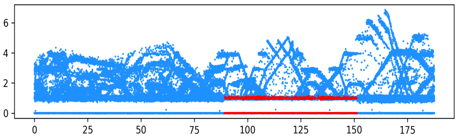
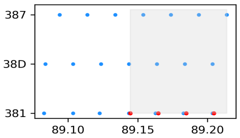
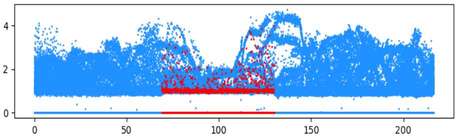
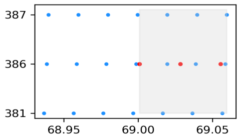
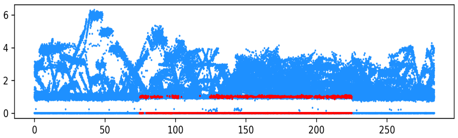
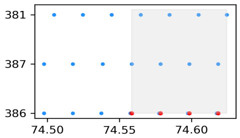
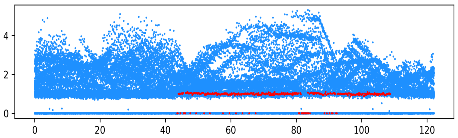
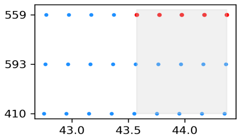
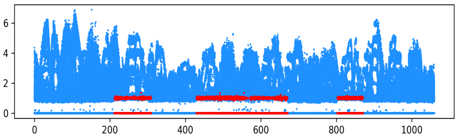
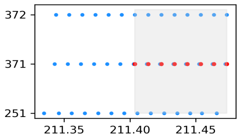
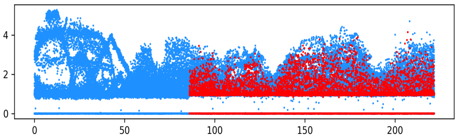
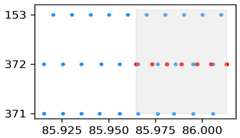
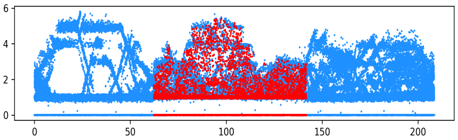
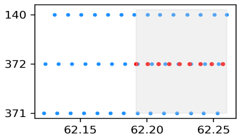
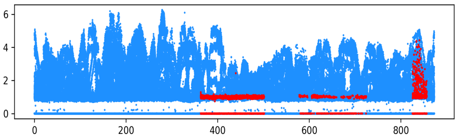
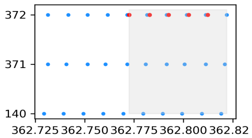
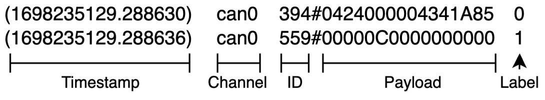
Attack	Observations	Message Timing	Targeted ID Message Timing
Power steering ★ 381#FFB73FXXXXXXXXXX 187.484292 Flam	'Check motor-driven power steering' warning message on the dashboard, slightly less control of the steering wheel.		
Max speedometer ★ 386#FFFFFFFFFFFFFFFF 216.432840 Injecting every 0.02s	Speedometer jumps to 159 mph while driving at 30 mph		
Min speedometer 1 ★ 386#FF027002F9821D42 283.422522 Flam	Speedometer jumps to 15 mph while driving at 30 mph		
Wiper warning ★ 559#XXXXXCXXXXXXXXXX 122.107031 Flam	Set the front wiper speed to 2 on the dashboard. No physical movement of the wiper.		
EMS replay long ★ 371#2E1E000000000010 1058.974900 Flam	No visible changes		
Gear shifter attack 1 ★ 372#800001000000AA05 221.688076 Injecting every 0.001s	'Shifting not possible due to overheating' warning message, Steering wheel becomes stiffer.		
Gear shifter attack 2 ★ 372#000001000000AA05 208.340552 Injecting every 0.001s	'Shifting not possible due to overheating' warning message. Steering wheel became too loose.		
Multiple attacks 1 ★ 372#XXFFXXXXXXXXXXXX 559#XXXXXCXXXXXXXXXX 386#00000000F982FFFF 872.579076 Flam and Injecting every 0.02s	Changed driving mode into 2WD certification mode for ID 372 attack, set the front wiper speed to 2 on the dashboard for ID 559 attack, speedometer jumps to 19 mph while driving at 30 mph.		

TABLE III: Description of attacks. Description for all attacks are available in https://github.com/sampathrajapaksha/CAN-MIRGU

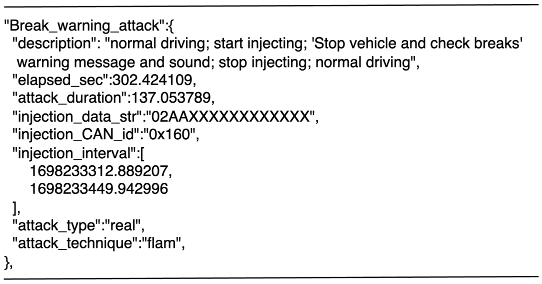
different frequent IDs, as detection capability might depend on the characteristics of each ID. Figure 4 depicts the frame transmission of normal driving over one second for the targeted 13 IDs. These frequent patterns are expected to change during the injection and suspension attacks due to the introduction of additional frames or suspension of frames. In contrast, masquerade attacks do not change this pattern, as they do not introduce any new frames.



```
(1698235129.288630)   can0   394#0424000004341A85   0
(1698235129.288636)   can0   559#00000C0000000000   1
```

Fig. 5: CAN bus data format

All CAN data files are logged using the candump command in can-utils. In addition to the standard fields of candump, labels are assigned as 0 for benign frames and 1 for attack frames, as illustrated in Figure 5. For the benign datasets, all labels are set to 0 since there are no instances of attack frames present. For suspension attacks, where frames associated with the targeted ID are removed throughout the attack period, leading to the absence of malicious frames, the entire attack window is labelled as 1. This labelling is necessary as IDSs must identify the attack window in the context of suspension attacks. Metadata is provided in JSON format for each attack capture, including the attack name, description, the length of the capture in seconds, attack duration in seconds, injected ID



```
"Break_warning_attack":{
  "description": "normal driving; start injecting; 'Stop vehicle and check breaks'
   warning message and sound; stop injecting; normal driving",
  "elapsed_sec":302.424109,
  "attack_duration":137.053789,
  "injection_data_str":"02AAXXXXXXXXXXXX",
  "injection_CAN_id":"0x160",
  "injection_interval":[
     1698233312.889207,
     1698233449.942996
  ],
  "attack_type":"real",
  "attack_technique":"flam",
},
```

Fig. 6: Snapshot of metadata for one attack

and payload, injection interval (start, end), attack type (real or synthetic), and attack technique (flam or time-based injection). An example of the provided metadata is shown in Figure 6. In the injection_data_str of the metadata, the wildcard character ‘X’ is used to indicate that these positions are not changed during flam delivery. Instead, the payload values of the last transmitted same ID are used for these positions. Similarly, ‘X’ in the injection_CAN_id field in fuzzing attacks indicates that no particular single ID is targeted. Random numbers are used in fuzzing random ID attacks, and a set of valid IDs are used in fuzzing valid ID attacks.

V. Discussion

Our attacks targeted 13 IDs based on ECU functionalities. For instance, the steering angle attack targeted the ID associated with steering-related data, while the EMS attack targeted the ID associated with the engine management system. The malicious payloads for the attacks were chosen by leveraging the limited knowledge of the DBC file of this vehicle and conducting repeated experiments. After selecting the highest payload value of ’FF’ for a byte, certain observations, such as warning lights or sounds, were noted compared to lower payload values. Consequently, ’FF’ was utilized in the some of malicious payloads, without resorting to random payloads, to maximize the impact of the attacks. However, for certain attacks listed in Table II and Table III, we did not observe any noticeable changes. This lack of observation could be attributed to potential changes that are not visible or the CAN bus actively ignoring inconsistent messages, possibly as a safety measure [16]. Another potential reason could be that the payloads used during the attacks violated the message checksum employed by this vehicle. Generally, packets with incorrect checksums are entirely disregarded by the ECUs on the CAN Bus for which the message is intended [30]. Time-based injection attacks targeting high-frequency IDs, such as fuzzing valid IDs and max-min speedometer attacks, often resulted in bus-off situations. In the creation of the ROAD dataset, the use of the maximum payload during fuzzing attacks aimed to prevent accidental ECU bus-offs [12]. Nevertheless, our experiments indicated that the occurrence of bus-off situations primarily depends on the injection frequency rather than the payload used. Consequently, we mitigated this issue by reducing the injection frequency based on repeated experiments. In cases of bus-off, we had to disconnect and reconnect the CAN data logger to re-establish the connection. It’s worth noting that a bus-off situation was observed for drive mode changing attacks, which targeted a low-frequency ID, 50C. In the DoS attack, the message timing plot in Table II shows a short period within the attack window where no attack frames are available, while frames with ID 0x000 continue to inject continuously throughout the attack period. This pattern is also evident in break warning, break and fog lights attacks, indicating a potential tendency of the CAN bus to temporarily ignore malicious frames.

The reactions to the majority of attacks were observed as warning messages, illuminated dashboard lights, and continuous warning sounds. While most attacks triggered non-critical responses, some can be classified as safety-critical, posing risks to the vehicle and its passengers. Max and min speedometer attacks, focusing on ID 386 associated with four-wheel speeds, share similarities with the correlated signal attack aimed at manipulating the speeds of the four wheels in the ROAD dataset [12]. In the ROAD dataset, this attacks led to the immobilization of the car due to varying, unrelated speeds among the wheels. By injecting different speed values into the respective bytes of the payload, max and min speedometer attacks did not result in physical changes to the vehicle but displayed inaccurate speedometer values. Nevertheless, this could be exploited by adversaries to deceive drivers, particularly in speed-regulated areas, posing potentially serious consequences. Two gear shifter attacks targeted the ECU associated with gear control, each employing distinct payloads. In the first case (payload: 800001000000AA05), a ’Shifting not possible due to overheating’ warning message continuously appeared on the dashboard, accompanied by warning sounds. Simultaneously, the driver experienced a stiff steering wheel, necessitating significant force to turn. In the second case (payload: 000001000000AA05), with only a slight change in the first nibble of the payload, the same warning message and sounds were present, but the steering exhibited looseness, making the vehicle overly responsive to minimal steering adjustments. Both attacks posed a risk of losing control over the vehicle. The noteworthy aspect is that this vehicle is equipped with full autonomous driving capabilities. Given that the vehicle is trained to navigate based on the curvature of the road, aided by the lane detection system, attacks of this nature on an autonomous vehicle could result in the vehicle deviating from its lane. Targeting the drive mode-associated ID 50C led to continuous switching between normal, sport, eco, and eco+ driving modes, resulting in unstable vehicle behaviour and jerking. However, the attacker node entered a bus-off mode shortly after the attack, a safety measure that, while preventing prolonged damage, still allowed for a potentially significant impact during the attack duration.

Certain vehicle functions require input from multiple CAN IDs with specific data to activate the functionality [16]. This was evident in the wiper warning attack. During this attack, we specifically altered the nibble of the payload associated with wiper position 2. However, despite the display on the dashboard indicating that the front wiper was set to level 2, there was no physical movement of the wiper. This occurrence suggests that additional changes to other associated IDs with specific data values may be required to activate the actuators. Some of these results can be observed in the demonstration video available at https://youtu.be/CufiACr2Zs8

ML-based IDSs are vulnerable to adversarial attacks such as model poisoning and data poisoning attacks. While recent attention has been on ML-based IDSs for in-vehicle networks, only one study has focused on adversarial attacks against Artificial Intelligence (AI)-based in-vehicle network IDSs [31]. This research demonstrated a drop in accuracy for attack detection due to data poisoning during model training. The proposed solution emphasizes robust model training encompassing poisoned and benign samples. However, there is a lack of publicly available datasets for evaluating IDSs against such adversarial attacks. To address this gap, we have included a comprehensive set of attacks, including EMS replay long and two multiple attack captures (multiple attack 1 and multiple attack 2), designed for assessing IDS resilience against the model and data poisoning attacks. Despite the absence of visible changes during EMS replay attacks, these types of attacks can be leveraged by adversaries to poison training datasets. Consequently, it is crucial to evaluate IDS resilience under adversarial learning conditions.

As the CAN-MIRGU dataset incorporates unaltered raw CAN data for both benign and attack instances, it is suitable for testing a range of IDS. This allows for the evaluation of IDSs employing various features, including timing, ID sequences, and payload data. It's important to highlight that the intended alterations to vehicle functionality were physically verified for all included injection attacks. For almost all injection attacks performed, the observations were instantaneous. Therefore, any IDS designed for the CAN bus should prioritize detecting the first instance of an attack within the shortest possible time. This focus on detection latency is crucial for implementing prompt countermeasures. Typically, detecting attacks like DoS and fuzzing is relatively straightforward. Nonetheless, we have included them in our dataset to ensure its comprehensiveness, encompassing various types of attacks. However, it is advisable to assess IDSs using all given attacks to conduct a thorough evaluation, rather than relying solely on simple attacks like DoS or fuzzing. This approach allows for a more comprehensive evaluation across different levels of attack difficulty.

While this dataset offers notable advantages, there are certain limitations to consider. The maximum speed of the vehicle during attack collection was 30 mph. Although the benign dataset encompasses driving scenarios at various speeds including 30 mph, executing attacks for other higher speeds used in benign driving was not feasible. Additionally, it's important to note that our simulated masquerade and suspension attacks may not perfectly mimic real-world scenarios. For instance, a real masquerade attack would be composed of additional packets to silence the target ECUs. Such additional packets can potentially provide valuable input for detection mechanisms.

VI. Conclusion

Despite the recent surge in focus and publication of IDSs for the CAN bus, advancing IDS research encounters significant hurdles due to the absence of high-quality, publicly available real CAN data that incorporates realistic attacks. This is mainly due to the substantial cost and associated risks linked to generating real attack data on moving vehicles. To overcome this challenge, we present a novel and publicly available CAN bus attack dataset collected from a modern automobile equipped with autonomous driving capabilities operating under real-world driving conditions. This dataset encompasses physically verified attacks, effectively filling the existing gap in publicly accessible CAN datasets featuring realistic attacks within dynamic driving scenarios. This, in turn, facilitates the thorough testing of various techniques presented in the literature. The availability of this dataset promises to enhance the comparison and validation of proposed IDS solutions.

References

[1] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, and G. Madzudzo, "Beyond vanilla: Improved autoencoder-based ensemble in-vehicle intrusion detection system," *Journal of information security and applications*, vol. 77, p. 103570, 2023.

[2] Z. Bi, G. Xu, G. Xu, C. Wang, and S. Zhang, "Bit-level automotive controller area network message reverse framework based on linear regression," *Sensors*, vol. 22, no. 3, p. 981, 2022.

[3] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, G. Madzudzo, and A. V. Petrovski, "Keep the moving vehicle secure: Context-aware intrusion detection system for in-vehicle can bus security," in *2022 14th International Conference on Cyber Conflict: Keep Moving!(CyCon)*, vol. 700. IEEE, 2022, pp. 309–330.

[4] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "Long short-term memory-based intrusion detection system for in-vehicle controller area network bus," in *2020 IEEE 44th Annual Computers, Software, and Applications Conference (COMPSAC)*. IEEE, 2020, pp. 10–17.

[5] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, G. Madzudzo, and M. Cheah, "Ai-based intrusion detection systems for in-vehicle networks: A survey," *ACM Comput. Surv.*, vol. 55, no. 11, feb 2023. [Online]. Available: https://doi.org/10.1145/3570954

[6] C. Miller, "Lessons learned from hacking a car," *IEEE Design & Test*, vol. 36, no. 6, pp. 7–9, 2019.

[7] 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, no. 39, p. 6, 2019.

[8] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, no. 1, p. 16, 2017.

[9] S. N. Narayanan, S. Mittal, and A. Joshi, "Obd_securealert: An anomaly detection system for vehicles," in *2016 IEEE International Conference on Smart Computing (SMARTCOMP)*, 2016, pp. 1–6.

[10] E. Aliwa, O. Rana, C. Perera, and P. Burnap, "Cyberattacks and countermeasures for in-vehicle networks," *ACM Computing Surveys (CSUR)*, vol. 54, no. 1, pp. 1–37, 2021.

[11] F. Luo, J. Wang, X. Zhang, Y. Jiang, Z. Li, and C. Luo, "In-vehicle network intrusion detection systems: a systematic survey of deep learning-based approaches," *PeerJ Computer Science*, vol. 9, p. e1648, 2023.

[12] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay, and F. L. Combs, "Road: The real ornl automotive dynamometer controller area network intrusion detection dataset (with a comprehensive can ids dataset survey & guide)," *arXiv preprint arXiv:2012.14600*, 2020.

[13] Hacking and C. R. Lab, "Car-hacking dataset for the intrusion detection," 2020, retrieved August 2021 from https://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset.

[14] M. Verma, R. Bridges, and S. Hollifield, "Actt: Automotive can tokenization and translation," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*. IEEE, 2018, pp. 278–283.

[15] S. Rajapaksha, H. Kalutarage, M. O. Al-Kadri, A. Petrovski, and G. Madzudzo, "Improving in-vehicle networks intrusion detection using on-device transfer learning," in *Symposium on Vehicles Security and Privacy (VehicleSec) 2023*.

[16] C. Miller and C. Valasek, "Can message injection," *OG Dynamite Edition*, 2016.

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

[18] H. K. Kalutarage, M. O. Al-Kadri, M. Cheah, and G. Madzudzo, "Context-aware anomaly detector for monitoring cyber attacks on automotive can bus," in *ACM Computer Science in Cars Symposium*. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–8. [Online]. Available: https://doi.org/10.1145/3359999.3360496

[19] H.-C. Lin, P. Wang, K.-M. Chao, W.-H. Lin, and J.-H. Chen, "Using deep learning networks to identify cyber-attacks on intrusion detection for in-vehicle networks," *Electronics*, vol. 11, no. 14, p. 2180, 2022.

[20] Hacking and C. R. Lab, "Can dataset for intrusion detection (otids)," 2020, retrieved August 2021 from https://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset.

[21] H. Lee, S. H. Jeong, and H. K. Kim, "Otids: A novel intrusion detection system for in-vehicle network by using remote frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, vol. 00, 2017, pp. 57–5709.

[22] Hacking and C. R. Lab, "Survival analysis dataset for automobile ids," 2020, retrieved August 2021 from https://ocslab.hksecurity.net/Datasets/survival-ids.

[23] M. L. Han, B. I. Kwak, and H. K. Kim, "Anomaly intrusion detection method for vehicular networks based on survival analysis," *Vehicular communications*, vol. 14, pp. 52–63, 2018.

[24] Hacking and C. R. Lab, "Car hacking attack and defense challenge," 2020, retrieved August 2021 from https://ocslab.hksecurity.net/Datasets/carchallenge2020.

[25] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "Syncan dataset," 2020, retrieved August 2021 from https://github.com/etas/SynCAN/blob/master/README.md.

[26] M. Hanselmann, T. Strauss, K. Dormann, and Ulmer, "Canet: An unsupervised intrusion detection system for high dimensional can bus data," *Ieee Access*, vol. 8, pp. 58 194–58 205, 2020.

[27] E. Novikova, V. Le, M. Yutin, M. Weber, and C. Anderson, "Autoencoder anomaly detection on large can bus data," *Proceedings of DLP-KDD*, 2020.

[28] D. o. M. TU Eindhoven and C. Science, "Tu eindhoven can bus intrusion dataset," 2020, retrieved August 2021 from https://doi.org/10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d.

[29] M. L. Han, B. I. Kwak, and H. K. Kim, "Event-triggered interval-based anomaly detection and attack identification methods for an in-vehicle network," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 2941–2956, 2021.

[30] C. Miller and C. Valasek, "Adventures in automotive networks and control units," *Def Con*, vol. 21, no. 260-264, pp. 15–31, 2013.

[31] Y. Li, J. Lin, and K. Xiong, "An adversarial attack defending system for securing in-vehicle networks," in *2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2021, pp. 1–6.