Cybersecurity in the CAN Protocol: A Systematic Mapping of Attacks and Vulnerabilities in Practical Scenarios

Amanda Kasat Baltor[1] **Kalinka Castelo Branco**[1]

[1] Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo – ICMC – USP
São Carlos, SP – Brazil

amanda.kasat@usp.br, kalinka@icmc.usp.br

Abstract. *The technological evolution of vehicles has expanded their attack surface, exposing the CAN (Controller Area Network) protocol to critical vulnerabilities. This paper presents a Systematic Literature Mapping (SLM) analyzing 196 studies focused on practical experiments. Results highlight the prevalence of Spoofing and Injection attacks targeting critical functions like braking and steering. The study identifies a significant lack of standardized attack taxonomy and risk classification in the literature. Furthermore, it distinguishes between legacy architectures and emerging UN R155-compliant vehicles, which implement defenses such as Security Gateways, marking a transition towards secure-by-design automotive ecosystems.*

1. Introduction

1.1. Background

A study by the Brazilian Institute of Planning and Taxation (IBPT) reveals that the vehicle fleet in circulation in Brazil reached approximately 120 million units in December 2023. Consequently, considering a population of 203 million according to the 2022 IBGE Census, there is roughly one vehicle for every 1.7 inhabitants in Brazil [O Tempo 2024]. Although the fleet was significantly smaller years ago, the sophistication level of mass-market vehicles at that time was also vastly different from what is seen on the streets today. From a software perspective, these changes are even more perceptible with the advent of advanced connectivity systems (Wi-Fi, Bluetooth, etc.) and the deployment of various Advanced Driver Assistance Systems (ADAS), such as collision warning and lane keeping assist. While these devices provide user comfort and autonomy, they also leave the automobile susceptible to a range of new vulnerabilities.

This risk is not just theoretical, real-world attacks using communication networks as gateways have already taken place. The most prominent example is the 2015 hack in which researchers Charlie Miller and Chris Valasek remotely took physical control of a Jeep Cherokee, revealing severe security lapses [Kaspersky 2015]. Such incidents, along with the potential for catastrophic consequences resulting from security breaches, drive extensive academic research in Automotive Cybersecurity [Fernandez de Arroyabe et al. 2022]. Findings from these studies help predict and prevent future attacks by identifying common vulnerabilities, force the industry to prioritize security by highlighting human and economic risks, and help define future manufacturing standards by proposing new software and systems.

1.2. Key Concepts

The **CAN module** is a transmission bus network utilized for serial communication among microcontrollers within an automobile. It functions as the backbone of the internal network that connects the various **Electronic Control Units (ECUs)** tasked with managing the vehicle's operations [Bajpai and Enbody 2020]. The CAN bus (Figure 1) transports data frames containing commands and important information for vehicle operation, coordinating functions from airbag deployment to traction control. The CAN module can communicate through different topologies, such as Star, Point-to-Point (P2P), Backbone, among others [Tanenbaum and Wetherall 2011].

In domain-oriented architectures, systems are usually partitioned according to logical function and message priority. Common domains include: Infotainment, Body, Powertrain, and Chassis [Wang et al. 2024]. Buses are selected for each domain to prioritize either speed or data volume.

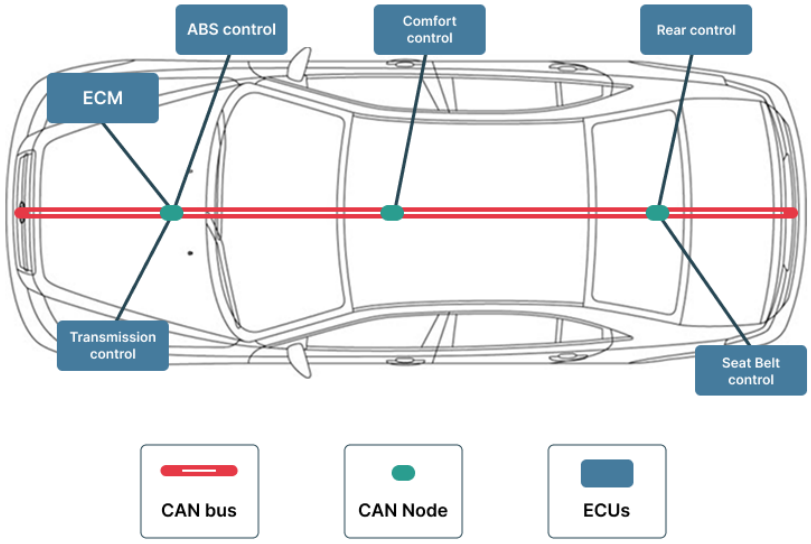


Figure 1. Example of a CAN bus with hybrid topology in a vehicle.

In the event of an attack, access to the CAN module can occur either physically—requiring hardware manipulation primarily via the OBD-II port or third-party devices —or remotely, exploiting the vehicle's connectivity over long ranges (e.g., Internet or cellular networks) or short ranges (e.g., Bluetooth or Wi-Fi). Once access is gained, various types of attacks known in the literature can be executed [Bajpai and Enbody 2020]. If these attacks target critical ECUs, they can cause severe accidents. Furthermore, Sniffing attacks can collect private data from the driver and passengers via connections with mobile phones.

1.3. Objective

This study aims to systematically analyze the literature on cyberattacks targeting the CAN protocol to identify their types, research methodologies (tools or databases used), main targets, and classification parameters employed. The ultimate goal is to map trends, attack patterns, and research gaps in the field.

1.4. Related Work

To contextualize our Systematic Literature Mapping, it is essential to examine how recent secondary studies have approached the domain of automotive cybersecurity. While existing literature provides valuable macroscopic insights, our study distinguishes itself through a strict focus on the empirical execution of attacks.

In this context, the article by Kifor et al. [Kifor and Popescu 2024] takes a broad view, providing an overview of automotive cybersecurity divided into four main topics: frameworks, regulatory standards, vulnerability management, and testing. In contrast, our mapping has a more specific scope, focusing on vulnerabilities and attack vectors specific to the CAN protocol. Furthermore, our study stands out by limiting the analysis to studies that conducted practical tests and empirical attacks on the CAN module, excluding purely theoretical proposals.

Other related study is the Systematic Literature Review by Luo et al. [Luo et al. 2022], which focuses specifically on cybersecurity testing methods and their testbeds. They categorize the literature based on testing approaches and the level of automation. In contrast, our research focuses on the attacks themselves and their physical targets (such as steering, brakes, and the engine), quantifying in the process the prevalence of attacks such as spoofing and injection in the real world, as well as raising a critical issue such as the lack of standardization in the taxonomy of attacks in the literature.

Another related study is the review by Fakhfakh et al. [Fakhfakh et al. 2021], which categorizes attacks and focuses primarily on listing and classifying defense mechanisms, such as the use of encryption, firewalls, and Intrusion Detection Systems (IDS). In contrast, however, our study does not focus primarily on defense mechanisms but rather on detailing the methodological aspects of the attacks reported in the literature, mapping the tools, public databases used (such as the HCRL and ROAD datasets), and the source of the empirical data.

2. Metodology

In order to understand the state of the art and the evolution of the security landscape in automotive networks, a Systematic Mapping Study (SMS) was conducted. This method was selected as it allows for a rigorous, reproducible, and comprehensive analysis of existing studies. The platform used to organize the process was Parsifal[1], which enables the structuring of the SMS in a standardized manner. The process was structured into several stages, detailed as follows:

2.1. SMS Planning

Four research questions were defined:

RQ1 : What types of cybersecurity attacks on the CAN protocol have been investigated in the study?

RQ2 : What methods, tools, and data sources were used to support the analysis of attacks on the CAN protocol?

[1]Online tool to support systematic reviews and mapping studies. Available at: `https://parsif.al/`.

RQ3 : What are the primary targets (e.g., specific ECUs, network availability, vehicle functions) of the investigated cybersecurity attacks on the CAN protocol?

RQ4 : According to which parameters (e.g., risk, difficulty) were the studied attacks classified?

Once the research questions were defined, commonalities among them were identified, which helped define the keywords for the SMS. Based on these keywords, the initial search string was constructed. After the calibration phase, that is, testing whether the search string was returning expected results and ensuring no essential papers were excluded—the final string presented below was obtained.

("CAN bus" OR "CAN module" OR "CAN protocol") AND ("Attack" OR "Intrusion" OR "Invasion") AND ("Simulation" OR "Implementation" OR "Experiment")

To retrieve the studies, the ACM Digital Library[2], Scopus[3], and IEEE Xplore[4] databases were consultedas they represent the most comprehensive repositories for computer science and engineering literature. The search string was adapted for each search engine, considering the specificities of each library. All searches were conducted on October 15, 2025, yielding a total of 346 studies: 288 from Scopus, 8 from ACM, and 50 from IEEE. A total of 63 duplicate studies were identified, resulting in 283 unique studies.

To finalize the planning phase, inclusion and exclusion criteria were established. The primary inclusion criterion required studies to investigate at least one type of attack on the CAN bus, whether through simulation software, prototype-simulated environments, or a real car. Conversely, studies were excluded if they were not primary studies, were not published in English, or lacked full text access. Consequently, works limited to theoretical proposals or those that did not explicitly describe the execution of attacks on the CAN bus in ground vehicles were also disregarded. This rigorous filtering was essential to distinguish between purely hypothetical vulnerabilities and those empirically proven to pose real risks, ensuring the mapping focuses strictly on practical experiments.

2.2. Study Pre-selection

For the pre-selection phase, the titles, abstracts, and keywords of the studies were examined to determine if there was explicit mention of conducting attacks on the CAN module. It was also verified whether the study was a primary study and published in English. In cases where the abstract was inconclusive regarding the experimental nature of the attack, the introduction and conclusion sections were briefly reviewed to ensure no relevant data was discarded.

Many articles cited keywords such as 'attack' and 'CAN module' but had distinct focuses, such as the development of multiphysics models or vehicle stability analysis. These were excluded for failing to meet the criterion of active attack investigation, limiting themselves to considering theoretical threats to validate engineering parameters. Furthermore, only works published in English were considered in this review.

Following this phase, 30% of the studies were excluded for not meeting the inclusion requirements, leaving 196 papers to be analyzed in the data extraction stage.

[2]Association for Computing Machinery Digital Library. Accessed via: `https://dl.acm.org/`.

[3]Elsevier Digital Library. Accessed via: `https://www.scopus.com/`.

[4]Institute of Electrical and Electronics Engineers Digital Library. Accessed via: `https://ieeexplore.ieee.org/`.

2.3. Data Extraction

The spreadsheet containing information on data extraction and the vehicles used in the experiments is available on the OSF. In addition to basic bibliographic information and research questions, the table was expanded to include further details such as the datasets and tools used in simulations, the nature of the data (real or simulated), the main conclusions of each study, and, in the case of real data, the specific vehicle from which the information was collected.

In this stage, data was extracted and tabulated. It was noted that some articles lacked a well-detailed methodology, which was the primary focus of this SMS, and therefore could not answer all research questions. Despite this limitation, such studies remain in the mapping as they offer relevant information for the general analysis.

2.4. Threats to Validity

Any Systematic Literature Mapping (SLM) is subject to limitations that may affect the accuracy and generalizability of its findings. To ensure the methodological rigor of this study, potential threats to validity were identified and mitigated across four primary dimensions: construct, internal, external, and conclusion validity.

Construct Validity There is always a risk that the search string will exclude some relevant studies due to the omission of a term that is important for identifying studies. To address this, several tests were conducted using different keywords and their synonyms, focusing primarily on articles retrieved from the Scopus virtual library. After several tests, the final search string was determined. It was also streamlined by excluding synonyms that did not increase the number of retrieved articles. In addition, there is also a language bias: by excluding studies published in languages such as Japanese or Mandarin, we have excluded work from countries that are actively conducting relevant research in the field of automotive safety. However, the difficulty in translating these languages into English, as well as the fact that English is the standard language for publication, justify this limitation.

Internal Validity During the mapping process, a lack of standardized terminology for attacks was observed; to facilitate the grouping and analysis of the most common attacks, it was necessary to catalog them, a process that may introduce biases. To mitigate this vulnerability, strategies were adopted as search for these attacks in forums to facilitate grouping by definition, as well as peer discussion in cases that were more difficult to categorize.

External Validity The study mapped various attacks carried out in testbeds; in this context, one concern is that attacks that are successful in a controlled environment may not work as well in real vehicles, where other external factors are involved. In addition, it was also noted that most critical vulnerabilities affect models manufactured before 2020. This opens up opportunities for future research—both practical and theoretical—on more modern vehicles compliant with the UN R155 standard.

Conclusion Validity To ensure the reproducibility of the study, the data extraction sheet containing the responses to the research questions and details of the vehicles used in the experiments has been made publicly available on OSF. This transparency provides the traceability required in a systematic literature review.

3. Results and Discussion

Before proceeding to the data extraction results, it is worth noting that the analyzed studies were highly diverse, with different authors, universities, and countries researching the topic of automotive security. This diversity enriches not only the current state of the art but also the SMS conducted herein, as it reduces the likelihood of bias and provides a comprehensive overview. Given this context, the discussion regarding the results found in relation to the Research Questions follows.

3.1. RQ1 : What types of cybersecurity attacks on the CAN protocol have been investigated in the study?

Upon listing each of the attacks performed by the authors, an immediate issue was observed: the lack of standardized nomenclature for the conducted attacks. This caused variants of the same term, such as '**Masquerade Attack**' and '**Masquerading Attack**', to be recorded as distinct categories. To proceed with the analysis, naming conventions were standardized. This facilitated the recognition of attacks well-known in the literature, as well as the identification of unique or lesser-known attacks.

The identified unique attacks, such as "**Sinkhole**" and "**Hello Floods**", appear in much smaller numbers compared to traditional ones like **Spoofing** and **Replay**. It was also observed that some of these attacks were minor, more specific variants of well-known attacks, such as the "**RPM Spoofing Attack**" and the "**Gear Spoofing Attack**", since both represent the same attack type but with different targets. Furthermore, it was noted that some studies used unfamiliar names for attacks that, by definition, were identical to well-known attacks; consequently, these attacks were reclassified under the name established in the literature.

Therefore, the attacks were initially divided into subcategories according to their specific characteristics and subsequently grouped into broader categories, as shown in Table 1. It is important to point out that the vast majority of papers conducted more than one type of attack on the CAN module, which explains why the total count of attacks is significantly higher than the number of studies. Additionally, attacks such as "**RPM Spoofing**" and "**Gear Spoofing**" were counted as two distinct **Spoofing** attacks, given that they possess different targets and potentially involve different code and simulation setups.

Observing Table 1, the **Spoofing & Injection** category (224 occurrences) is the most prevalent, stemming from the CAN module's lack of native authentication. It should be explicitly clarified that the number of attacks in this category (224) exceeds the total number of primary studies in the SML (196). This occurs because if a single study executed multiple distinct types of spoofing or injection attacks, each variant was counted individually. Since any node has write access, injection becomes a prime target. This category is critical because injecting fabricated commands can induce dangerous vehicle behaviors. The second category, **Availability Threat** (DoS) (165), causes serious damage by disabling functions like brake assists. Furthermore, DoS attacks are simple to execute, often requiring only message flooding without complex reverse engineering.

A low incidence of **Direct Manipulation** (26) was also observed, likely due to the difficulty in conducting them given the broadcast nature of the CAN module and the

Table 1. Categorization and Quantification of Vehicular Cyberattacks

Category	Total	Goal	Subcategories and Key Attacks (Count)
1. Spoofing & Injection	224	Deceive the system with false data or forged identities.	**Signal Spoofing:** Generic (66), RPM (20), Gear (19), Target ID (6), Stand-still Attack (2), ACK (1). **Impersonation:** Masquerade (29), Generic (20), MitM (4), Sybil (1), Denning-Sacco (1). **Fabrication:** Injection (37), Fabrication (11), UDS/Diagnostic (3), FDI (3), Deception Attack (1).
2. Availability (DoS)	165	Prevent vehicle functions or disrupt ECU communication.	**Resource Exhaustion:** DoS Attack (107), Flooding (14), Hello Floods (1), Jamming (1), Surge Attack (1). **Protocol/Bus:** Suspension (16), Bus-off (8), Shutdown (1), Delete (1), Select Forward (1), Bus Possession (1), Frame-less DoS (1), All-Zero ID Attack (1), Jitter-Induced Disruption (1). **Others:** Malfunction (8), Disruption (1), Sinkhole (1).
3. Replay & Timing	77	Use legitimate messages out of context or alter temporal flow.	**Replay:** Generic (73). **Temporal:** Interval Attack (2), Frequency-appearance modification (1), Complex Series Order Attack (1).
4. Reconnaissance	114	Discover vulnerabilities or map network traffic.	**Scanning/Fuzzing:** Fuzzy Attack (95), Reconnaissance (1). **Sniffing/Monitoring:** Sniffing (13), Information Gathering (3), Intercept (1). **Analysis:** Side-channel Analysis - SCA (1).
5. Manipulation	26	Alter physical state, internal code, or message content.	**Content Modification:** Generic (12), Field/Bit Manipulation (4), Data Corruption (2), Replacement (1). **Actuator/Sensor:** Accelerator (2), Steering (2), DC Motor (1), Light Sensor (1), ACC (1).
Others	15	Miscellaneous or unclassified attacks.	Buffer Overflow (3), Generic (2), Obfuscation (2), Additive (2), SOME (2) , Hybrid (1) RCE (1), Physical Intrusion (1), Malicious CAN Messages (1).

** Note: Totals reflect the sum of all individual instances provided in the dataset.*

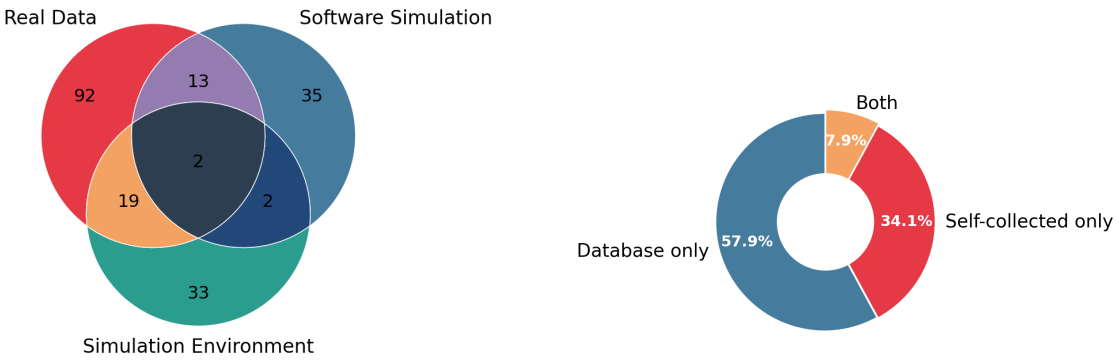
bus speed. Therefore, altering the content of a frame on-the-fly (as in a classic Man-in-the-Middle attack) is extremely difficult on CAN. It is much easier for an attacker to ”override” the legitimate message with Spoofing than to attempt to manipulate a specific bit within milliseconds.

It is worth noting that 4 studies did not specify the attacks studied. It is also important to highlight the lack of nomenclature standardization, with the coexistence of multiple terms describing identical attack types. This hinders the establishment of standards and underscores the pressing need for a unified taxonomy.

It is also important to note that the prevalence of these attacks reflects the scientific community's focus and the ease with which they can be replicated in controlled environments. To understand the actual frequency of attacks in real-world scenarios, it is necessary to cross-reference this academic data with Cyber Threat Intelligence (CTI) sources, which monitor industry incidents and usually reveal criminal trends that differ from the strictly academic perspective.

3.2. RQ2: What methods, tools, and data sources were used to support the analysis of at- tacks on the CAN protocol?

The initial analysis regarding data extraction yielded highly comprehensive results. Tools were grouped into three main categories based on data origin: real vehicles (empirical data), simulation software (synthetic data), and controlled prototyping environments (using Arduino, protoboards, and/or specific hardware). Several studies employed more than one data type, such as collecting data from a vehicle and testing it in a simulated environment, among other combinations, witch allows researchers to validate synthetic findings against empirical evidence. Therefore, to provide a better visualization, a Venn diagram was created (Figure 2). Regarding studies based on real data, a further distinction was made: some studies utilized public datasets containing real data, others relied solely on self-collected data, and some employed both approaches.



(a) Data Source Intersection.

(b) Details: Source of Real Data.

Figure 2. Classification of data collection approaches: overview and details

Starting with the studies that analyzed real data from public datasets, an indisputable dominance of the HCRL (Hacking and Countermeasure Research Lab) [Song et al. 2020] can be observed, whose datasets, specifically the Car-Hacking Dataset and the CAN-intrusion dataset, are practically a literature standard. The ROAD dataset

(ORNL) [Verma et al. 2021] and CICIDS2017 [Wang et al. 2024] appear as the second most listed in the studies. For data processing, the use of machine learning libraries such as Scikit-learn[5], TensorFlow[6], and Keras[7] for the Python language is noted. Even when relying on pre-existing repositories, it is observed that many studies also replicate their own test environments using both hardware and software tools.

Next, analyzing the group of real data collected by the researchers themselves, a wide variety of tools used for reading vehicle data is perceived, ranging from simple, low-cost options like ELM327 adapters, Arduino, and Raspberry Pi, to complex and expensive ones like Vector (CANcaseXL)[8], PEAK System (PCAN-USB)[9], and PicoScope oscilloscopes[10].. Regardless, the vast majority of these tools utilize the same access point: the OBD-II port. Tools such as Wireshark[11] and Vehicle Spy 3[12] are frequently employed to read the raw data obtained. Almost all experiments also specified the vehicle used; however, this analysis is detailed in Subsection 3.5.

The next analysis concerns the Software Simulated Data group. Due to the absence of physical hardware, these studies rely on mathematical modeling and network simulation environments. MATLAB/Simulink[13] stands out as the central tool. However, traffic and network simulators such as OMNeT++[14], NS-3[15] and SUMO[16] are also quite popular. Virtual sniffing tools, such as SocketCAN virtual interfaces (vcan), allow researchers to employ the same real-world network analysis tools (like Wireshark and can-utils[17]) within the simulated environment.

Finally, there is the group of data obtained through simulated environments, which includes Hardware-in-the-Loop and Testbeds. This block represents a hybrid approach, where test benches simulate the vehicle utilizing real physical components outside the chassis. In this scenario, there is intensive use of microcontrollers emulating ECUs, with a prominence of Arduino, STM32, and specific automotive chips such as the Infineon AURIX[18]. Furthermore, the use of FPGAs is notable for prototyping dedicated security

[5]Open source library for predictive data analysis and machine learning, built on top of NumPy, SciPy, and matplotlib. Official documentation: `https://scikit-learn.org/`.

[6]Plataforma completa de código aberto para aprendizado de máquina e computação numérica, desenvolvida pelo Google Brain. Documentação oficial: `https://www.tensorflow.org/`.

[7]Interface de programação de aplicações (API) de alto nível para redes neurais e aprendizado profundo. Documentação oficial: `https://keras.io/`.

[8]High-performance bus interface for development and testing of automotive networks (CAN, LIN, FlexRay). Manufacturer: Vector Informatik GmbH. Available at: `https://www.vector.com/`.

[9]CAN-to-USB interface adapter, industry standard for monitoring vehicle networks. Manufacturer: PEAK-System Technik. Available at: `https://www.peak-system.com/`.

[10]PC-based oscilloscopes for advanced automotive diagnostics and electrical signal analysis. Manufacturer: Pico Technology. Available at: `https://www.picotech.com/`.

[11]Network protocol analyzer for packet capture. Official site: `https://www.wireshark.org/`.

[12]Intrepid Control Systems software for vehicle network analysis, simulation, and engineering. Official site: `https://www.intrepidcs.com/`.

[13]MathWorks platform for numerical computing and model-based simulation. Official site: `https://www.mathworks.com/`.

[14]Simulator dedicated to communication network modeling. Official site: `https://omnetpp.org/`.

[15]Discrete-event network simulator for internet systems. Official site: `https://www.nsnam.org/`.

[16]Microscopic and multimodal traffic simulation. Official site: `https://eclipse.dev/sumo/`.

[17]Collection of userspace utilities for the Linux CAN subsystem (SocketCAN). Official repository: `https://github.com/linux-can/can-utils`.

[18]Multicore microcontroller family developed by Infineon for safety-critical automotive applications and

hardware. From a software perspective, Vector CANoe[19] is by far the most widely used tool to orchestrate these environments, enabling the simulation of the ”vehicle” while interacting with the physical ECUs on the test bench.

Studies situated at the intersection of two or more groups employ a combination of the tools described above. These studies are therefore able to overcome certain limitations inherent to each individual data collection method. For example, integrating software simulation with controlled environments (testbeds) enables the massive execution of destructive attacks without physical risks prior to final validation on real hardware.

3.3. RQ3: What are the primary targets (e.g., specific ECUs, network availability, vehicle functions) of the investigated cybersecurity attacks on the CAN protocol?

A vast variation in attack targets was observed across the studies, with individual studies often conducting attacks on distinct focal points. The data was categorized into six groups based on their type (specific function, ECU, access points) and the associated risks:

a) **Critical Driving and Safety Functions:** These are directly responsible for vehicle operation. Compromising these targets yields the most severe consequences, directly impacting the life and physical integrity of occupants and third parties. Examples: Brakes, Steering, Acceleration, Powertrain/Transmission, among many others.
b) **Display and Vehicle Auxiliary Functions:** These are responsible for the driver interface and comfort. Although not directly critical, manipulation can cause distractions due to unexpected vehicle behavior. Examples: Air conditioning system, Speedometer, RPM (Revolutions Per Minute), Gear Indicator, Lights (headlights, brake lights, turn signals, reverse lights), and others.
c) **Electronic Control Units (ECUs) and Specific Modules:** These represent the vehicle's electronic ”brains,” managing the specific functions listed in items (a) and (b). Attacks may seize control of ECUs, disable them, modify their behavior, or use them as a vector to target specific functions. Examples: Engine ECU, Body Control Module (BCM), Brake Control Module, Steering Wheel ECU, Gateway ECU, and others.
d) **CAN Bus Network and Communication:** Some attacks solely aim to disrupt, overload, and/or interrupt the normal operation of the CAN module without targeting a specific function or ECU. In this case, the communication infrastructure itself and the CAN protocol logic are the targets. This can cause serious damage, as legitimate and critical messages may be lost or delayed.
e) **Data Integrity and Confidentiality:** The focus here is on the manipulation, falsification, theft, or leakage of information trafficking the network, directly attacking the privacy of the driver or passengers. This type of target may also indicate initial vehicle reconnaissance for future attacks.
f) **Vectors and Access Points:** These are the physical or logical means by which an attacker can gain access to the CAN bus to initiate attacks. Examples: Diagnos-

ADAS systems. Official site: `https://www.infineon.com/`.

[19]Software environment for the development, testing, and analysis of individual ECUs or entire systems networks. Developed by Vector Informatik. Official site: `https://www.vector.com/`.

tic ports (OBD-II), Wireless interfaces (Wi-Fi, Bluetooth, V2X), Over-The-Air (OTA) update modules, and others.

From the studies reviewed, it was observed that the most targeted attacks were directed at the Engine Control (62 mentions), Steering System (44 mentions), and Brake System (40 mentions).

3.4. RQ4: According to which parameters (e.g., risk, difficulty) were the studied attacks classified?

A significant portion of the studies (148) did not perform an explicit or systematic classification of the attacks. This includes articles that focused on classifying algorithms, adversaries, or vehicles, as well as those that simply did not mention any form of attack classification. In contrast, only a small group utilized classifications already established in the literature, such as the STRIDE threat model[20] (**S**poofing, **T**ampering, **R**epudiation, **I**nformation Disclosure, **D**enial of Service, and **E**levation of Privilege) and published works [Nilsson et al.].

Therefore, the second most common group consists of studies that developed their own taxonomy. These publications employ varied criteria, including: implementation technique, detection difficulty, adversary profile, and systemic impact. Regarding technique, a distinction is observed between the injection of complete frames versus specific bits, as well as a separation between attacks that are transparent or opaque to network timing. Regarding detection, attacks such as Fuzzy and Impersonation are classified as high complexity due to their resemblance to valid messages, whereas DoS is cited as an attack simple to implement and easily identified. Regarding the profile, distinctions are made between hackers with more or less experience, with the former being considered more dangerous due to their ability to conduct complex, hard-to-detect attacks. Finally, regarding systemic impact, the division lies between potential harm to passenger safety/life versus solely material damage, such as increased fuel consumption and premature wear.

3.5. Empirical Data Analysis and Test Vehicles

As previously mentioned, additional information beyond the Research Questions was collected from the studies. A relevant example is the vehicle used in the experiments, where data collection may have originated either from the research team itself or from public datasets. In total, 47 unique vehicles were cataloged. It is worth noting that many studies did not provide all primary information to identify the car, such as make, model, and year of manufacture. However, since many studies conducted attacks on more than one vehicle, the diversity and completeness of the information were not compromised.

Based on the primary data, additional information was collected and categorized into General Vehicle Information (category, propulsion, and cost segment), Technical Specifications related to the CAN module and network (CAN protocol, ECU estimation, network topology, Gateway isolation level, Domain Separation, other secondary buses, network architecture, connectivity, and OBD-II interface), and Secu-

[20]Threat modeling methodology developed by Microsoft for the systematic identification of security vulnerabilities. Official documentation: `https://learn.microsoft.com/azure/security/develop/threat-modeling-tool-threats`.

rity Assessments (average security index, security protocol, and UN R155 compliance [United Nations Economic Commission for Europe 2021]).

Regarding general information, it is observed that various vehicle models from the last 20 years were covered. Among these, models released from 2020 onwards (Hyundai Avante CN7, Genesis G80, XPeng G9, P5, Tesla Model 3, LeapMotor C10) mark a radical shift in architectures, reflecting rapid innovation and the response to new regulations, such as the UN R155 standard. Vehicles such as the Hyundai Avante CN7, Genesis G80, and XPeng P5/G9 explicitly mention CAN buses dedicated to ADAS and the integration of sensor and camera data.

Prior to 2020, for example, most vehicles had exposed OBD-II ports, allowing for the direct injection of CAN messages. Gateways of that era, also unsecured, allowed jumping from the Infotainment domain to the Powertrain, posing significantly higher risks. However, 2020 marked a major milestone in automotive security with the emergence and popularization of more sophisticated hardware and software, such as Firewalls, Service-Oriented Architecture, and Secure Gateways (SGW), which make direct access to the CAN module via the OBD-II port much more difficult. It is also worth highlighting specific architectures such as the XPeng G9's X-EEA 3.0 (Central Supercomputer (C-DCU) and Zone Controllers (Z-DCU)) and the Tesla Model 3's central computer (MCU), which handle high-performance processing. Software-Defined Vehicle (SDV) security systems have also been proposed, implementing "smart fuses (e-Fuses)" for real-time detection/isolation of intruder nodes and encryption layers for information passing through the bus [Xin et al. 2025].

Regarding protocols, CAN 2.0 predominates in nearly all vehicles up to the beginning of the current decade. Variations such as GMLAN (GM) and SAE J1939 (trucks) are also noted. However, in 2020, a new protocol emerged: CAN FD, driven by the need for greater bandwidth for ADAS and infotainment systems.

Regarding the safety index, it is worth noting that there is currently no rating system related to an automobile's vulnerability to a hacker attack. Therefore, the collision safety rating was considered to provide an idea of the car's average safety. In general, almost all have high ratings, with 4/5 stars. Although there is no specific index, there are important metrics currently available, such as compliance with UN R155, a global regulation that makes cybersecurity a mandatory requirement for the sale of vehicles, requiring manufacturers to implement a management system to detect and mitigate digital attacks.

3.6. Temporal Trends and Field Maturity

The literature reveals a clear temporal evolution in automotive cybersecurity. Pre-2020 studies mostly exploited physical vulnerabilities (e.g., exposed OBD-II ports) using straightforward Spoofing and DoS attacks. Post-2020, driven largely by the UN R155 regulation, research shifted towards emerging architectures like Secure Gateways (SGWs) and Over-The-Air (OTA) exploitation. Despite this evolution, Spoofing and DoS remain the most prevalent attacks in practical scenarios, reflecting the enduring lack of native encryption in legacy CAN networks.

Regarding experimental methodologies, we also observed a significant evolution. Older studies usually relied on testing directly on physical vehicles using basic, low-

cost adapters like the ELM327. However, more recent papers show a preference for safer and more complex setups. It is now common to see a hybrid approach, combining Hardware-in-the-Loop (HIL) testbeds with software simulators (such as CANoe) to safely test destructive attacks. In addition, the growing use of public datasets—like HCRL and ROAD—and Machine Learning models for Intrusion Detection Systems (IDS) shows that the community is moving away from simple exploratory hacking. Instead, the focus is now on much more organized and data-driven security evaluations.

4. Conclusions

Technological evolution in automobiles has brought numerous benefits that increase comfort and autonomy, but conversely, it has also drastically expanded the attack surface. In this scenario, once an attacker gains access to the CAN bus, they can manipulate vital car functions and cause severe damage.

To understand the state of the art and the evolution of the security landscape in automotive networks, a Systematic Literature Mapping was conducted. By excluding purely theoretical proposals and requiring that the included studies have conducted at least one simulated or real attack, the review ensures that the results reflect the technical reality of the studied scope. The research questions were designed to encompass the typology, tools and methods, targets and attack surfaces, and subsequent classification.

Regarding the research questions, the popularity of Spoofing and Injection attacks (RQ1), correlated with the prevalence of targets in critical driving systems such as braking and steering (RQ3), demonstrates that critical vulnerabilities, such as the lack of native authentication and encryption in the CAN module, can pose serious risks to integrity. The ease with which low-cost tools (such as Arduinos and ELM327 adapters, identified in RQ2) are capable of manipulating vital vehicle functions demonstrates that, for the vast majority of the circulating fleet (pre-2020 models), security depends almost exclusively on obscurity and physical isolation. Furthermore, the absence of standardization in attack nomenclature (RQ1) and systematic risk classifications (RQ4) reveals a research gap. In this scenario, unified classification frameworks could facilitate the comparison between attacks and also aid in proposing global defense metrics.

The analysis of the studied vehicles also reveals a clear difference between two generations of automobiles: the older one, pre-2020, characterized by distributed architectures, exposed OBD-II ports, and vulnerable CAN 2.0 buses; and the modern one, driven by the UN R155 regulation and the adoption of zonal architectures, Automotive Ethernet, CAN-FD, and Security Gateways (SGW). However, although defenses are evolving, the lack of cybersecurity ratings analogous to NCAP[21] demonstrates that despite the evolution, there are still many areas for improvement on this topic, especially regarding its popularization and standardization.

Future works based on this SLM include conducting attack simulations focused specifically on vehicles compliant with the UN R155 regulation. The objective is to investigate the robustness of new Security Gateway (SGW) implementations, verifying whether domain segmentation is sufficient to prevent the propagation of critical attacks initiated via OBD-II or wireless interfaces.

[21]Global New Car Assessment Program that rates performance in collisions and assistance systems. Official site: `https://www.globalncap.org/`.

References

Bajpai, P. and Enbody, R. (2020). Towards effective identification and rating of automotive vulnerabilities. In *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, AutoSec '20, page 37–44, New York, NY, USA. Association for Computing Machinery.

Fakhfakh, F., Tounsi, M., and Mosbah, M. (2021). Cybersecurity attacks on can bus based vehicles: a review and open challenges. *Library Hi Tech*, 40(5):1179–1203.

Fernandez de Arroyabe, I., Watson, T., and Angelopoulou, O. (2022). Cybersecurity in the automotive industry: A systematic literature review (slr). *Journal of Computer Information Systems*.

Kaspersky (2015). Black hat usa 2015: The full story of how miller and valasek hacked a jeep. `https://www.kaspersky.com/blog/blackhat-jeep-cherokee-hack-explained/9493/`. Accessed: 29 jan. 2026.

Kifor, C. V. and Popescu, A. (2024). Automotive cybersecurity: A survey on frameworks, standards, and testing and monitoring technologies. *Sensors*, 24(18).

Luo, F., Zhang, X., Yang, Z., Jiang, Y., Wang, J., Wu, M., and Feng, W. (2022). Cybersecurity testing for automotive domain: A survey. *Sensors*, 22(23).

Nilsson, D., Phung, P., and Larson, U. *Vehicle ECU classification based on safety-security characteristics*, page 102.

O Tempo (2024). Frota brasileira fecha 2023 em 119.227.657; um veículo para cada 1,7 habitante. Accessed: 29 jan. 2026.

Song, H. M., Woo, J., and Kim, H. K. (2020). In-vehicle network intrusion detection using deep convolutional neural networks. In *Proceedings of the 7th ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet)*, pages 10–17, New York, NY, USA. Association for Computing Machinery.

Tanenbaum, A. S. and Wetherall, D. J. (2011). *Redes de computadores*. Pearson Prentice Hall, São Paulo, 5 edition.

United Nations Economic Commission for Europe (2021). UN Regulation No. 155: Uniform provisions concerning the approval of vehicles with regards to cyber security and cyber security management system. Regulation 155, UNECE.

Verma, M. E., Iannaccone, M. D., Bridges, R. A., Hollifield, S. C., Kay, B., and Combs, F. L. (2021). ROAD: The real open automotive datasets for modern vehicles. In *Proceedings of the 3rd ACM Workshop on Automotive and Autonomous Vehicle Security (AutoSec '21)*, New York, NY, USA. Association for Computing Machinery.

Wang, W., Guo, K., Cao, W., et al. (2024). Review of electrical and electronic architectures for autonomous vehicles: Topologies, networking and simulators. *Automotive Innovation*, 7:82–101.

Xin, Y., Wang, X., Lu, L., Zhuo, S., Jiang, Y., Singh, A. K., Ren, K., Yang, M., and Wu, K. (2025). LUFT-CAN: A lightweight unsupervised learning based intrusion detection system with frequency-time analysis for vehicular CAN bus. *Journal of Systems Architecture*.