

USENIX

THE ADVANCED COMPUTING
SYSTEMS ASSOCIATION

AutoHack: A Physically Verified Multi-Bus CAN Dataset for Intrusion Detection System Evaluation

Yuchan Song, *Korea University;* Sejun Ahn, *Kookmin University;*
Seungjin Baek, *Korea University;* Hyeonseong Kim, *Kookmin University;*
Huy Kang Kim, *Korea University;* Sanghoon Jeon, *Kookmin University*

https://www.usenix.org/conference/vehiclesec26/presentation/song

This paper is included in the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy.

August 10–11, 2026 • Baltimore, MD, USA

ISBN 978-1-939133-56-4

Open access to the Proceedings of the
4th USENIX Symposium on Vehicle Security and Privacy
is sponsored by USENIX.



AutoHack: A Physically Verified Multi-Bus CAN Dataset for Intrusion Detection System Evaluation

Yuchan Song*,[1]
thddbcks618@korea.ac.kr

Sejun Ahn*,[2]
s3037444@kookmin.ac.kr

Seungjin Baek[1]
baekseungjin@korea.ac.kr

Hyeonseong Kim[2]
gj0690@kookmin.ac.kr

Huy Kang Kim†,[1]
cenda@korea.ac.kr

Sanghoon Jeon†,[2]
sh.jeon@kookmin.ac.kr

[1]Korea University [2]Kookmin University

Abstract

This paper presents the AutoHack Dataset, a high-fidelity vehicle cybersecurity dataset collected from a 2023 Hyundai vehicle in real-world driving environments. Unlike existing datasets that rely on simulations or simplified periodicity assumptions, AutoHack Dataset provides synchronized multiple CAN buses logs (C-CAN, P-CAN, and B-CAN) and incorporates sophisticated attack scenarios such as UDS-based diagnostics and timing-opaque Replay attacks. We prioritize ground-truth reliability by physically verifying attack impacts, including engine stalls and emergency call activations, to ensure accurate labeling of actual threats. Evaluation results from the AutoHack competition demonstrate that the dataset presents a significant challenge to state-of-the-art intrusion detection systems. Furthermore, we provide a comprehensive performance comparison between the competition-winning models and representative machine learning algorithms. By releasing this dataset alongside these benchmark results and optimized models, we offer a robust and reproducible framework for developing generalized vehicular security solutions. AutoHack Dataset bridges the gap between network-level detection and functional safety in modern automotive architectures.

1 Introduction

Controller Area Network (CAN) serves as the primary network for transmitting control messages between Electronic Control Units (ECUs) in vehicle systems. However, its original design did not sufficiently consider security requirements, leaving the network without inherent features such as authentication, integrity, or confidentiality. This lack of security makes CAN vulnerable to various attacks, including message spoofing, replay, and arbitrary injection [1]. Consequently, researchers have actively developed methods to detect and defend against such threats, with Intrusion Detection Systems (IDS) emerging as a critical technology for vehicle security. However, the detection performance and reliability of an IDS depend significantly on the quality and realism of the datasets used for its training and evaluation. In real vehicle environments, factors such as ECU interactions, varying driving conditions, and event-driven communications complexly alter traffic patterns. Evaluations using datasets that fail to reflect these characteristics often lead to poor performance in real-world applications. Therefore, researchers must construct high-quality datasets that encompass the diverse communication characteristics of actual vehicles to ensure the effectiveness and reproducibility of IDS studies [13].

However, existing public CAN datasets do not sufficiently reflect realistic vehicle environments. Many datasets assume that CAN messages occur periodically and use timing-based features as primary detection criteria. This assumption causes an IDS to misidentify non-periodic normal traffic, which occurs frequently in actual vehicles, as an attack [4]. Furthermore, most public datasets focus on traffic from simulations or simplified environments, failing to capture the complex ECU interactions and exceptional communication traits present in real driving. Even real-vehicle datasets often target only a single CAN bus, which prevents them from reflecting the cross-domain interactions or attack propagation that occur in multi-bus structures connected via gateways. Modern vehicles utilize multiple CAN buses—divided into functional domains such as powertrain, chassis, and body—that interoperate through a gateway. Attacks can occur simultaneously or sequentially across these buses rather than being confined to a single one. Thus, observing only one bus limits the explanation of attack paths and cross-domain dependencies. An IDS trained on single-bus data likely suffers from performance degradation or increased false positives due to partial observ-

*Equal contribution.
†Corresponding authors.

ability within a multi-bus vehicle architecture.

Existing datasets also frequently exclude interactions based on actual vehicle diagnostic protocols, such as Unified Diagnostic Services (UDS). UDS does not merely appear in attack scenarios; it constitutes event-driven traffic that arises during normal operations, including maintenance, inspection, calibration, and diagnostic sessions. These interactions exhibit non-periodic characteristics. Evaluating an IDS with datasets that omit UDS interactions leads to critical limitations, such as misidentifying frequent non-periodic communications as attacks or failing to validate diagnostic-based attacks. Consequently, performance evaluations based on existing datasets risk overestimation by biasing toward specific assumptions like periodicity, resulting in poor generalization when applied to real vehicles [18].

1.1 Dataset and the AutoHack Competition

We constructed the proposed dataset for the AutoHack competition [12, 15], which aimed to enhance vehicle security capabilities through real-vehicle attack execution and IDS development. We collected traffic from three CAN buses (C-CAN, P-CAN, and B-CAN) on a moving vehicle and reflected the complexity of simultaneous attacks on multiple buses by injecting threats into two buses concurrently. This dataset differentiates itself from existing ones by incorporating a high-difficulty traffic composition, implementing and utilizing actual UDS processes for attacks rather than simple message injections.

Figure 1 illustrates the on-site environment of the AutoHack competition. The event was conducted over three days, with ten teams participating, each composed of four members. Participants developed IDS models using the proposed dataset and were granted five official submission opportunities. A real-time leaderboard provided immediate feedback after each submission. This competition-based evaluation framework enabled external validation of the dataset's effectiveness and practical applicability.

A notable observation from the competition was that participant performance remained relatively low and dispersed. Despite using a common dataset and evaluation metric along with five official submission opportunities for iterative model improvement, the final weighted F1-scores ranged from 39% to 66%. Since this distribution was observed across multiple participating teams rather than from a single failed run, and since even the top score remained at 66%, the results suggest that the dataset was not trivially saturated under the competition setting. This observation motivated our attack-wise analysis of detection difficulty in Section 4.

1.2 Contributions of This Study

This study makes the following key contributions:



Figure 1: AutoHack competition.

- **Realistic Multi-bus Configuration and Non-Periodic Traffic Traces**: This dataset provides multiple buses' traffic data collected in real-world driving environments and reflects event-driven, non-periodic communication characteristics. These features enable a more rigorous evaluation of false positive and false negative issues that periodicity-based IDS models face in realistic settings.

- **Physically Verified High-Fidelity Labels**: We secured dataset reliability by physically verifying the impact of attack scenarios on actual vehicle maneuvers and precisely labeling attack intervals based on these observations. This approach offers a level of ground-truth fidelity that simulation-based datasets rarely provide.

- **A Challenging Benchmark for Reproducible Evaluation**: Our dataset serves as a challenging benchmark that tests the limits of state-of-the-art IDS models. By releasing it alongside the winning models from the competition, we support fair performance comparisons and reproducible evaluations in future research.

1.3 Ethical Consideration

We conducted all experiments within a safe, controlled environment and testbed on our University's internal roads, isolated from public traffic. These activities posed no risk to other vehicles or the general public. We provide the collected dataset solely for academic research purposes. Since the vulnerabilities and attack techniques we explore confirm known structural limitations of existing protocols, they present a negligible risk for actual vehicle theft or lethal exploitation. Furthermore, the entire research process adheres to relevant ethical regulations and minimizes the exposure of security vulnerabilities.

Table 1: Comprehensive comparison of open CAN IDS datasets based on six key characteristics.

#	Dataset	D.C.	A.I.	T.P.	N.T.	P.C.	L.R.	Ref.
1	CAN Intrusion (OTIDS)	Real	Real-time	Periodic-heavy	Single	CAN only	heuristic	[6]
2	Survival Analysis for Automotive IDS	Real	Real-time	Periodic-heavy	Single	CAN only	heuristic	[8]
3	Car Hacking for Intrusion Detection	Real	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[7]
4	SynCAN	Sim	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[5]
5	Automotive CAN Bus Intrusion v2	Real	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[3]
6	Can Log Infector	Real	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[2]
7	ROAD CAN Intrusion Dataset	Real	Real-time	Periodic-heavy	Single	CAN only	P.I.V.[2]	[18]
8	CAN Signal Extraction and Translation	Real	Real-time	Periodic-heavy	Single	CAN only	heuristic	[9]
9	M-CAN Intrusion Dataset	Real	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[11]
10	B-CAN Intrusion Dataset	Real	Offline[1]	Periodic-heavy	Single	CAN only	heuristic	[10]
11	can-train-and-test	Real	Real-time	Periodic-heavy	Single	CAN only	P.I.V.[2]	[14]
12	CAN-MIRGU	Real	Offline[1]	Periodic-heavy	Single	CAN only	P.I.V.[2]	[19]
13	CICIoV2024	Testbed	Real-time	Periodic-heavy	Single	CAN only	heuristic	[16]
14	**AutoHack (Ours)**	**Real**	**Real-time**	**P-H/E-D**	**Multi**	**C&U**	**P.I.V.[2]**	**This Work**

D.C.: Data Collection, **A.I.**: Attack Injection, **T.P.**: Temporal Property, **N.T.**: Network Topology, **P.C.**: Protocol Coverage, **L.R.**: Label Reliability
[1] Offline: Replay/edit/simulated injection [2] P.I.V.: Physical-impact verified P-H/E-D: Periodic-heavy/Event-driven C&U: CAN and UDS

2 Related Works

2.1 Investigation of Vehicle Cybersecurity Research Datasets

Vehicle cybersecurity datasets have evolved from early studies that identified simple attack packet injections to modern research that reflects the physical domain characteristics of actual vehicles. Early-stage datasets, such as OTIDS [6] and the Car Hacking Dataset [7], primarily focused on attacks that cause high bus loads by abnormally surging traffic, such as Denial of Service (DoS) or Fuzzing. While these datasets contributed to verifying the basic mechanisms of IDS, researchers now argue that they lack sufficient discriminative power to represent modern vehicle security threats, as contemporary detection models achieve nearly 100% detection accuracy on such simplified attack patterns [18].

Subsequently, researchers began to project communication characteristics from real driving environments into their datasets. The ROAD dataset [18] enhanced sophistication by including Masquerade attacks that influence physical behavior based on logs collected from actual vehicles. Recent studies continue this trend, such as can-train-and-test [14], which tests model generalization performance across multiple vehicle models, and CAN-MIRGU [19], which reflects the dynamic characteristics of vehicles during real-world driving.

2.2 Dataset Comparison

We categorize and evaluate the quality and utility of vehicle security datasets based on six key characteristics. Table 1 summarizes the comparative analysis of major datasets, including the AutoHack dataset proposed in this work.

2.2.1 Data Collection

We categorize data collection environments into real vehicles, laboratory-scale vehicle testbeds, and simulation environments that mimic virtual vehicles. As shown in Table 1, most datasets, including OTIDS, ROAD, and M-CAN, collect data from real vehicle environments to capture in-vehicle dynamic characteristics. While testbeds or simulation environments prove useful for verifying valid logical operations, they face limitations in fully reflecting irregularities caused by communication delays, bus loads, and exceptions that occur during actual driving. Considering these factors, we collected the AutoHack dataset in a real vehicle driving environment, providing a foundation to validate the reliability of intrusion detection models under actual driving conditions.

2.2.2 Attack Injection

We classify attack injection methods into real-time injection of attack messages and offline injection by editing pre-collected normal data. According to Table 1, several datasets, including Car Hacking, SynCAN, and Automotive v2, adopt the post-editing approach. While this method offers advantages in constructing datasets safely and easily, it fails to reflect the logical responses and interactions between in-vehicle ECUs or the dynamic system changes caused by attack injections. In contrast, the AutoHack dataset adopts the real-time injection method—similar to ROAD, can-train-and-test, and CICIoV2024—to ensure high reliability by allowing models to learn and verify variables that occur in actual threat situations.

2.2.3 Temporal Property

We distinguish the transmission period of messages based on whether they transmit at regular intervals (periodic) or irreg-

ularly based on specific events (event-driven). Most public datasets consist of periodic messages to project normal driving conditions. While useful for verifying detection models that analyze transmission interval deviations, these datasets limit the ability to reflect the complexity of non-periodic messages in actual vehicles. The AutoHack dataset integrates periodic messages with event-driven non-periodic messages arising from specific operating conditions, providing a precise metric to evaluate intrusion detection model performance.

2.2.4 Network Topology

We classify network topology as single-bus or multi-bus environments based on the structure and domain of the CAN bus where data collection occurred. Most public datasets provide data collected from a single CAN bus. While suitable for evaluating IDS models targeting individual buses, this approach limits the reference for inter-bus interactions and attack propagation through gateways across the entire vehicle network. The AutoHack dataset focuses on a multi-bus environment within an actual vehicle, reflecting the complex network architecture to enable performance evaluation for IDS models covering the entire internal CAN network.

2.2.5 Protocol Coverage

As shown in Table 1, most public datasets—including Survival Analysis, SynCAN, and CAN-MIRGU—collect data exclusively for the standard CAN protocol. However, actual vehicles utilize a mixture of various protocols for control and diagnostics. The AutoHack dataset simultaneously collects data using the UDS protocol for real-time diagnostics in addition to standard CAN messages. This provides a sophisticated benchmark for IDS models to detect threats where attackers exploit specific authorized diagnostic services to impact driving safety critically.

2.2.6 Label Reliability

Finally, differences exist in labeling precision and verification methods, which determine dataset reliability. In Table 1, most public datasets employ a post-collection labeling method based on specific patterns in CAN IDs, data fields, or injection times. While easy to implement, this method raises reliability concerns regarding label precision due to network delays or bus loads. Therefore, we performed real-time labeling during the attack injection process for the AutoHack dataset and confirmed the physical changes and impacts caused by the injections to verify the causal relationship within the vehicle system.

3 AutoHack Dataset

This section describes our data collection methodology, the physical verification of attack scenarios, the statistical composition of the dataset, and its distinctive features compared to existing datasets. To facilitate reproducibility and support the research community, the proposed AutoHack Dataset is publicly released through a permanent Zenodo archive [17], which serves as the official access channel for the dataset. The released dataset includes ① synchronized multi-bus CAN logs collected from C-CAN, P-CAN, and B-CAN interfaces, ② metadata, ③ ground-truth labels for the attack scenarios, and ④ accompanying documentation. The data are provided in CSV format for broad compatibility with various data analytics and machine-learning tools. The dataset is intended to support academic research and educational use, and released under the Creative Commons Attribution 4.0 International (CC BY 4.0) license, allowing reuse with appropriate attribution.

3.1 Data Collection Methodology

We designed this dataset to ensure data reliability and temporal precision within real-world driving environments.

Experimental Environment We conducted experiments using a 2023 Hyundai Motor Company S model and constructed an integrated testbed by disassembling vehicle components to tap into the wiring connected to the ECUs behind the Central Gateway (CGW) for the C-CAN, P-CAN, and B-CAN interfaces, as illustrated in Fig. 2. In this setup, C-CAN was persistently monitored across all vehicle states, whereas P-CAN and B-CAN data were selectively captured during driving and parking (stationary) sessions, respectively. To ensure safety, we performed all data collection and attack injections within a controlled environment on a university campus road, isolated from public traffic. We utilized a Dual Channel Kvaser hardware interface for data acquisition and managed the process using custom scripts developed with the Kvaser CANlib SDK on a Windows-based laptop acting as an Attacker PC.

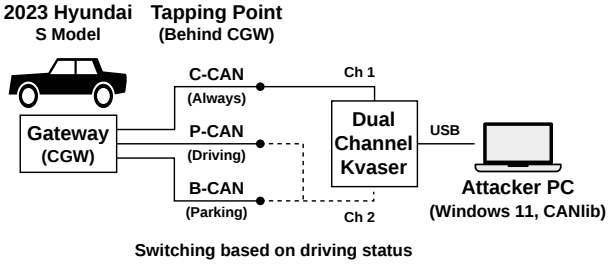


Figure 2: Experimental testbed configuration for data collection and attack injection.

Synchronization between multiple buses To achieve precise time synchronization across the functional buses, we utilized a single-device Dual Channel Kvaser interface, where both channels share a unified hardware timebase. While C-CAN was persistently monitored on the first channel, the P-CAN and B-CAN interfaces were captured on the second channel during their respective driving and stationary sessions. Since all incoming frames from both ports are timestamped by the same internal oscillator at the moment of reception, the system maintains a strictly consistent global timeline. This architecture ensures that the temporal relationship between the persistent C-CAN traffic and the alternating P-CAN and B-CAN data remains synchronized at the microsecond (μs) level, inherently eliminating software-induced time offsets or drifts between the channels.

Strict Ground-Truth Labeling To maximize labeling accuracy, we define the starting point as the moment the actual bus completes the transmission of the first attack frame, rather than the time the attack script initiates. Specifically, we apply a strict criterion by labeling reactive messages or jitter as normal data, as the attacker did not directly generate these packets. We also defined the attack termination point as the completion of the last injected packet's transmission.

3.2 Attack Scenarios & Physical Verification

We constructed scenarios that pose threats to actual vehicle control and verified them physically, going beyond simple packet injection.

3.2.1 Attack Injection Scenarios

This dataset includes the following five major attack types:

DoS: We utilize the 0x000 ID, which holds the highest priority, to flood messages at a 0.2ms interval and seize network bandwidth.

Fuzzing: We induce system malfunctions by injecting random combinations of IDs, DLCs, and payloads at $1ms$ intervals. To explore a comprehensive range of potential vulnerabilities, the attack covers the entire valid 11-bit CAN ID spectrum from $0x000$ to $0x7FF$. This broad scope intentionally includes IDs of $0x700$ and above, extending the fuzzing targets to the UDS range. By including diagnostic communication channels, the attack increases the probability of triggering unintended diagnostic states or system-level exceptions.

Spoofing: We select valid IDs directly associated with critical vehicle functions, such as the Electronic Parking Brake (EPB) and Parking Assist System (PAS), to ensure the attack remains indistinguishable from legitimate traffic. To address potential security and proprietary concerns of the vehicle manufacturer, we specifically disclose the detailed specifications of the C-CAN interface as representative cases in Table 2. The attack involves selecting a random integer between 1 and 10 and injecting that specific number of messages consecutively at $1ms$ intervals in a burst fashion to disguise them as normal signals.

Replay: We neutralize timing-analysis-based IDS by retransmitting normal data pre-collected from the C-CAN, P-CAN, and B-CAN interfaces during both driving and parking states. To ensure protocol and architectural consistency, the retransmission is strictly mapped to the corresponding bus type, such as replaying C-CAN logs specifically onto the C-CAN interface. These messages are synchronized with their original hardware timestamps to replicate legitimate traffic intervals and patterns, effectively deceiving detection systems that rely on frequency analysis.

UDS Spoofing: We execute sophisticated attacks by re-implementing legitimate diagnostic sequences extracted from an authentic Original Equipment Manufacturer (OEM) diagnostic tool, enabling the forceful control of ISO 14229-compliant sessions in core ECUs. Similar to the spoofing attacks, we exclusively disclose UDS targets associated with the C-CAN interface to address security and proprietary concerns of the vehicle manufacturer. As summarized in Table 3, these attacks focus on safety-critical components on the C-CAN bus, such as the front camera, power steering, and surround radars. By mimicking authorized diagnostic requests, we can transition these ECUs into restricted diagnostic modes or deactivate specific functions, effectively influencing the vehicle's functional safety beyond simple bus injection.

3.2.2 Physical Impact Verification

We directly verified the effectiveness of all attacks through changes in the physical state of the vehicle.

DoS: We confirmed that data continuity breaks cause warning lights and system error alarms in major ECUs.

Table 2: Detailed specifications of targeted C-CAN IDs for Spoofing attacks.

ID (Hex)	Function	Physical Response
0x366	Engine RPM	RPM gauge fluctuation
0x390	Warning Alarm	Unexpected alert trigger
0x490	EPB Error	EPB warning indicator
0x4F4	Parking Assist	PAS logic error
0x50C	Fuel Warning	Fuel warning trigger

Table 3: Detailed specifications of targeted UDS-based IDs on C-CAN for Spoofing attacks.

ID (Hex)	Targeted ECU	Physical Response
0x755	Right Radar	Diagnostic state entry
0x7B7	Left Radar	Sensor session lock
0x7C4	Front Camera	Camera mode deactivation
0x7D4	Power Steering	Steering assist limitation

Fuzzing: We captured critical defects, such as the vehicle suddenly stopping in Drive (D) mode or an automatic connection to the emergency call center without an accident.

Spoofing: We confirmed that the injection of messages specified in Table 2 leads to functional malfunctions, which are manifested as warning lights or "Check System" messages on the instrument cluster. These physical responses were directly verified through experiments, confirming that spoofing critical control IDs effectively triggers unintended system behaviors.

Replay: We verified control interference by retransmitting pre-collected normal data synchronized with original timestamps. This allowed us to arbitrarily reproduce or neutralize specific functions, such as turn signals and window control, by playing back historical driving sequences.

UDS Diagnostic Session Manipulation: We can change the ECU state or temporarily limit certain vehicle functions by forcibly starting unauthorized diagnostic sessions and performing specific service requests. For example, specific functions may deactivate or diagnostic modes may activate; this demonstrates that UDS-based attacks influence actual vehicle behavior beyond simple bus injection.

These physical verification results demonstrate that all attack scenarios, including diagnostic-based attacks, exert observable impacts on the actual vehicle state, proving that the AutoHack dataset reflects real threat environments rather than simple packet-level variations.

3.3 Dataset Composition & Statistics

The collected data follows a systematic statistical structure for objective performance evaluation of detection models.

Session-based Data Splitting To prevent bias from temporal continuity, we separated the Train Set and Test Set into independent driving sessions at a ratio of approximately 6.6:3.4. and Table 4 provide detailed configuration metrics.

Spoofing ID Separation Strategy To prevent the model from overfitting to specific IDs, we assigned spoofing IDs for the training and test sets mutually exclusively, as shown in Table 6. In particular, we assigned new IDs to the test set for B-CAN to rigorously measure generalization performance against unseen attacks.

Table 4: Detailed attack statistics by interface and session.

Bus Interface	Attack Type	Train Set	Test Set
C-CAN	Total Frames	4,026,488	1,678,489
	Fuzzing	38,191	13,147
	Replay	24,123	16,876
	Spoofing	22,088	5,384
	DoS	14,419	8,490
	UDS Spoofing*	580	420
P-CAN	Total Frames	2,655,149	883,323
	Fuzzing	39,952	10,319
	DoS	34,441	7,647
	Replay	25,797	16,746
	Spoofing	23,634	5,188
	UDS Spoofing*	887	344
B-CAN	Total Frames	240,963	149,685
	Fuzzing	57,459	29,698
	DoS	28,244	19,492
	Spoofing	12,846	11,698
	Replay	5,589	8,408
	UDS Spoofing*	580	460

*Aggregated counts of various UDS-based spoofing

3.4 Distinctive Features

The AutoHack Dataset possesses three core strengths that distinguish it from existing datasets:

- **Overcoming Limitations of Periodicity Assumption**: Most public CAN datasets assume messages occur at fixed periods, limiting detection to simple timing analysis. This dataset includes non-periodic (event-driven) communications like UDS and precisely restores the $Inter-arrival time$ of original traffic during Replay attacks. This sophisticated configuration neutralizes the detection logic of existing IDS models relying on temporal regularity and provides a challenging evaluation environment requiring more advanced detection algorithms.
- **High Complexity and Entropy Based on Real Driving Environments**: Unlike datasets collected in limited labs, we collected this dataset through actual driving on campus roads, naturally including various driving states such as acceleration, deceleration, turning, and stopping. This increases the variability of message distribution and results in high entropy, making anomaly detection

Table 5: AutoHack Dataset statistics.

Metric	Train Set	Test Set
Total Frames	6,922,600	2,711,497
Driving Duration (sec)	1,400.8182	583.1505
Unique CAN IDs	2,048	2,048
Normal Frames	6,593,770	2,557,180
Attack Frames	328,830	154,317

Table 6: Summary of target CAN IDs for Spoofing attacks by interface and dataset.

Bus Interface	Train Set (IDs)	Test Set (IDs)
C-CAN	0x366, 0x390, 0x4F4	0x490, 0x50C
P-CAN	0x171, 0x374, 0x5CF	0x080, 0x2B0
B-CAN	0x18E, 0x1BC, 0x510	0x012, 0x08C, 0x102, 0x103, 0x107, 0x1C2, 0x560

more realistic and difficult. To maximize experimental reliability, we separated the Train and Test sets into independent driving sequences at different times rather than arbitrarily splitting a single session.

- **Providing Synchronized Multi-bus data**: Unlike existing studies focused on a single CAN bus, the AutoHack dataset provides synchronized logs of C-CAN, P-CAN, and B-CAN at the hardware level. This allows for analyzing message characteristic changes through the gateway and attack propagation across multiple buses, reflecting modern vehicle architectures rarely addressed in existing datasets.

4 Benchmarking and Evaluation

Section 3 introduced the AutoHack Dataset and its design choices for realistic in-vehicle network modeling. In this section, we validate these claims through controlled benchmarking experiments and analyze whether the dataset provides a reproducible and fair evaluation environment for diverse IDS models.

4.1 Benchmark Setup

4.1.1 IDS Competition Structure: AutoHack

This study establishes the benchmark configuration based on the operational structure and evaluation framework of the AutoHack competition conducted in conjunction with the AutoHack Dataset. The competition was designed to enable undergraduate students with a strong interest in intrusion detection systems to develop and evaluate diverse detection models under a unified dataset and common evaluation metrics. This setting allowed for the observation of how the dataset is utilized in a practical research context and how performance distributions emerge across different modeling approaches.

4.1.2 Baselines for Benchmarking

The baseline configuration is not intended to highlight the superiority of any specific model. Rather, it aims to establish a reference environment to verify whether the proposed dataset supports fair and reproducible evaluations across diverse training conditions. To achieve this, we adopted widely recognized models from existing IDS studies that also emerged as top performers in the AutoHack competition. This approach allows us to examine the performance distribution of algorithms validated in both academic research and competitive settings, thereby ensuring the objectivity and comparability of our evaluation results.

Regarding evaluation metrics, F1-score was selected as the primary metric in consideration of the fact that accuracy can be overly optimistic in class-imbalanced settings. In addition, from the perspective of attack detection, recall for each attack class was reported to quantitatively reflect the risk of false negatives.

Controlled Evaluation Protocol To ensure that the benchmark results reflect genuine model generalization rather than artifacts of data leakage or memorization, we strictly adopt the controlled data configuration described in Section 3. First, the train and test sets are separated at the driving-session level to prevent temporal continuity between splits, thereby mitigating performance inflation caused by shared temporal context. This session-based split enables a more rigorous evaluation of generalization across independent driving conditions.

Second, since the feature configuration includes Arbitration_ID, we enforce mutually exclusive spoofing target IDs between the training and test sets. This prevents the model from memorizing specific attack identifiers and instead evaluates its ability to generalize to previously unseen spoofing patterns. In particular, assigning new spoofing IDs to the B-CAN test set allows us to explicitly assess cross-ID generalization capability. Under these controlled settings, the observed performance distributions can be interpreted as reflecting the structural characteristics of the dataset rather than unintended biases in data partitioning.

Baseline Models In this study, tree-based models (Random Forest and XGBoost), which demonstrated strong performance in the competition and are widely adopted in IDS research, were selected as the baseline configuration. These models are capable of effectively capturing both statistical

and temporal features, thereby providing an appropriate reference for analyzing how the structural complexity and class imbalance inherent in the dataset are reflected in performance metrics. Rather than using the baseline models as evidence of practical applicability by themselves, we use this baseline setting as a controlled and reproducible reference protocol for analyzing IDS-relevant characteristics of the AutoHack Dataset. Under this protocol, the dataset exposes several benchmark-relevant properties: heterogeneous detection difficulty across attack types, decision-boundary changes caused by non-periodic UDS traffic, and generalization challenges in a time-synchronized multi-bus setting. These observations provide empirical support that the AutoHack Dataset can serve as a realistic and challenging benchmark for IDS evaluation. To support reproducibility and transparency, we archive the IDS reference implementation on Zenodo together with the dataset-related artifact [17]. This release enables independent verification and replication of the experimental setup, feature configuration, data preprocessing, and evaluation procedures.

Feature Configuration To ensure a fair evaluation of the AutoHack Dataset, an identical feature set was applied to all baseline models. The selected features do not rely on vehicle-specific signal interpretation; instead, they consist of reproducible attributes that generally represent the structural information and temporal patterns of CAN messages. Specifically, the feature set includes 'Basic & Temporal' features and frequency-based statistical features ('Frequency & Rolling'). The detailed composition is presented in Table 7.

4.2 Benchmark Results and Key Observations

Observation 1: Variations in Detection Difficulty Across Attack Types The AutoHack Dataset is not constructed around a single level of attack difficulty. Rather, it simultaneously includes highly distinguishable attacks and stealthy attacks that are highly correlated with normal traffic. This design is significant from a benchmarking perspective, as it avoids bias toward a specific attack type and instead encompasses a broad spectrum of difficulty levels. In other words, the dataset provides a structure that enables the evaluation of IDS performance under diverse conditions by incorporating both relatively distinguishable attacks and high difficulty attacks that distributionally overlap with normal traffic.

These characteristics were consistently observed in the competition results. In particular, Replay and Spoofing attacks exhibited relatively lower performance across participating models, thereby forming a higher level of detection difficulty. The consistent degradation observed across heterogeneous model families suggests that this difficulty is not model-specific but reflects an intrinsic structural property of the dataset. Table 8 presents the classification performance (F1-score) for each attack type in the AutoHack Dataset, quantitatively demonstrating that detection difficulty varies structurally across attack categories. While relatively separable attacks achieved high F1-scores, Replay and Spoofing showed comparatively lower F1-scores, indicating greater difficulty in distinguishing them from the normal class.

The confusion matrix in Figure 3 further illustrates this distributional characteristic. Spoofing and Replay were misclassified as Normal 7,770 times (34.89%) and 8,784 times (20.90%), respectively, which are markedly higher than those of other attack types. In addition, Normal samples were misclassified as Spoofing (2,346 cases) and Replay (13,051 cases) more frequently than for other attack classes. As described in Section 3, these findings support the interpretation that Replay and Spoofing attacks exhibit structural similarity to normal traffic, since they are generated by retransmitting legitimate driving messages or by leveraging valid control IDs, thereby resulting in distributional characteristics that closely resemble normal communication.

In summary, the AutoHack Dataset is structured to encompass varying degrees of separability across attack types and to simultaneously include attacks with different levels of detection difficulty. This design enhances the suitability of the dataset as a benchmarking resource by avoiding bias toward a single difficulty level and enabling evaluation across diverse attack scenarios. The baseline results serve as a reference framework for analyzing these structural characteristics, confirming that the dataset provides an evaluation environment that is not skewed toward overly separable or uniformly simple attack patterns.

Importantly, the observed performance variations should not be interpreted merely as limitations of the evaluated models. Rather, they reflect the realistic nature of the AutoHack Dataset, in which the distributional boundary between normal and attack traffic is inherently ambiguous. By incorporating attacks that overlap with normal communication patterns, the dataset presents a detection environment that more closely resembles real-world vehicular networks. Consequently, compared to existing public datasets, it offers a more challenging and practically representative benchmark for IDS evaluation.

Observation 2: Impact of Non-Periodic (UDS) Communication on Detection Boundaries The AutoHack Dataset is designed to include not only periodic CAN communication but also non-periodic diagnostic communication based on UDS, which occurs in real vehicle environments. This design reflects the fact that in-vehicle networks are not merely composed of periodic signals, but rather exhibit a composite structure in which event-driven and diagnostic messages coexist. In practical vehicle systems, non-periodic diagnostic traffic constitutes a legitimate part of normal operation; therefore, incorporating UDS messages enables a more realistic representation of the structural diversity of benign communication patterns.

To analyze the impact of this data configuration, we com-

Table 7: Baseline feature set for multi-bus intrusion detection

Category	Feature	Description
Basic & Temporal	Arbitration_ID	CAN ID (decimal format)
	DLC	Data Length Code (bytes)
	Prev_Interval	Time interval from previous message
	ID_Prev_Interval	Inter-arrival time (same CAN ID)
	Data_Prev_Interval	Inter-arrival time (same CAN ID and payload)
Frequency & Rolling	ID_Frequency	ID occurrence count (10 second time window)
	Data_Frequency	ID + payload occurrence count (10 second time window)
	Frequency_diff	ID frequency difference

Table 8: Per-attack performance of baseline IDS.

Attack Type	Precision	Recall	F1-score	AUC
Normal	0.9940	0.9928	0.9934	0.9952
DoS	1.0000	1.0000	1.0000	1.0000
Spoofing	0.8328	0.7449	0.7864	0.9908
Replay	0.6498	0.7128	0.6799	0.9854
Fuzzing	0.9706	0.9943	0.9823	0.9999
UDS_Spoofing	0.9498	0.9126	0.9308	0.9992

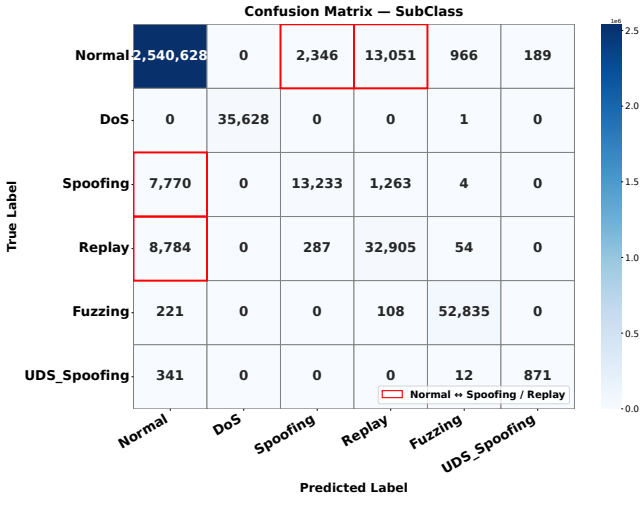


Figure 3: Confusion matrix of the Random Forest model.

pared two experimental settings: one including UDS messages and the other excluding them. The comparison was conducted under identical preprocessing procedures, feature configurations, model settings, and evaluation protocols, with the inclusion of UDS traffic as the only varying factor. As shown in Table 9, removing UDS results in a macro F1-score of 0.8634, suggesting that aggregate performance metrics exhibit no substantial difference. However, a closer examination of the confusion matrix in Figure 4 reveals meaningful distributional changes that are not captured by the overall metric.

In particular, when UDS messages are excluded, the number of Normal samples misclassified as Fuzzing increases from 966 to 5,296—representing more than a fivefold rise. This change indicates a significant restructuring of the decision boundary between specific classes rather than a simple fluctuation in overall accuracy.

This phenomenon is closely related to the structural characteristics of the Fuzzing attack defined in Section 3. As described therein, the Fuzzing attack generates random CAN IDs within the range 0x000 to 0x7FF, randomly selects DLC values between 0 and 8, and injects arbitrary payloads at 1 ms intervals to induce unexpected system malfunctions. Such traffic lacks periodicity and exhibits high variability and randomness, forming a statistically irregular distribution.

UDS traffic, while benign, also exhibits non-periodic behavior, as it follows a request–response pattern and is triggered under specific conditions or events. From a temporal and

Table 9: Per-attack performance of baseline IDS (Excluding UDS).

Attack Type	Precision	Recall	F1-score	AUC
Normal	0.9933	0.9916	0.9925	0.9948
DoS	0.9999	1.0000	1.0000	1.0000
Spoofing	0.8003	0.6187	0.6979	0.9898
Replay	0.6465	0.7122	0.6777	0.9846
Fuzzing	0.9074	0.9946	0.9490	0.9999
Macro Avg	0.8695	0.8634	0.8634	0.9938
Weighted Avg	0.9848	0.9844	0.9844	0.9938

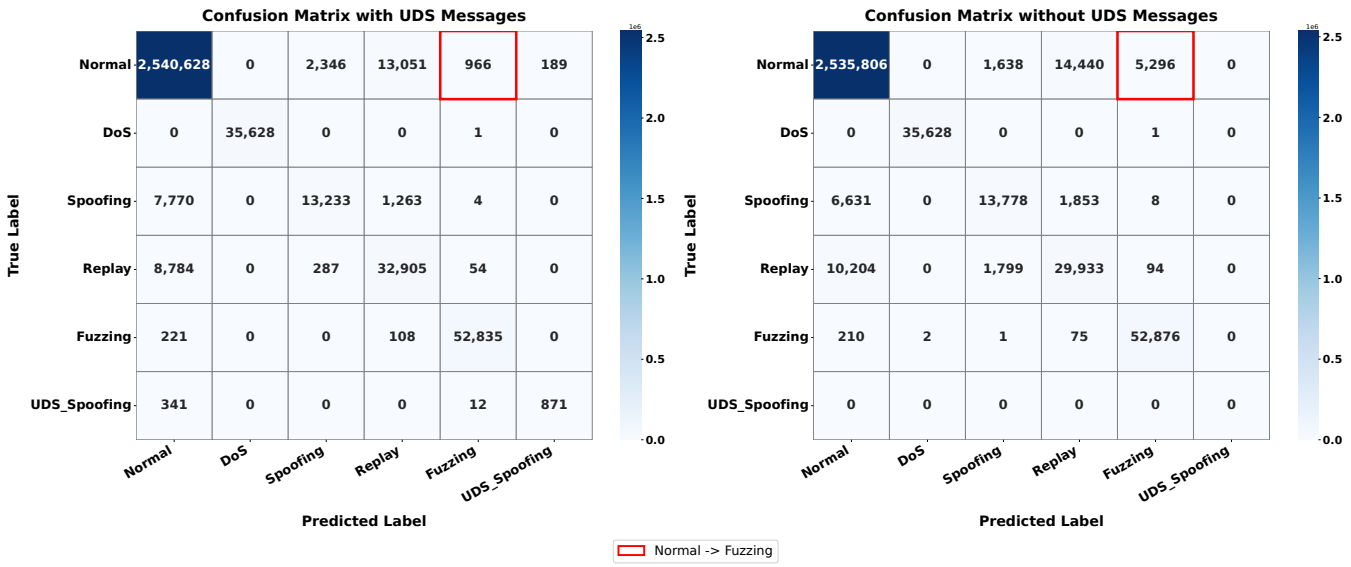


Figure 4: Confusion matrices of the Random Forest with and without non-periodic (UDS) messages.

statistical perspective, UDS shares partial similarities with Fuzzing in terms of non-periodicity and variability. When UDS is included, the model learns that irregular communication patterns do not necessarily indicate malicious activity, enabling it to form a more refined decision boundary between stochastic attack traffic and legitimate diagnostic communication. In contrast, when UDS is removed, the Normal class becomes relatively simplified and centered around periodic patterns. Consequently, the model may develop a stronger association between non-periodic behavior and malicious activity, increasing the likelihood that boundary samples are misclassified as Fuzzing.

Although removing UDS inevitably alters the total number of samples and class distribution to some extent, misclassification changes for other attack types (*e.g.*, DoS, Spoofing, and Replay) remain relatively limited. In contrast, the Normal-to-Fuzzing misclassification exhibits a pronounced increase. This suggests that the observed performance variation is better interpreted not as a simple consequence of reduced data volume, but rather as a structural weakening of the distributional boundary between Normal and Fuzzing due to the exclusion of non-periodic benign traffic.

In summary, the inclusion of UDS traffic should be understood not merely in terms of numerical performance differences, but as a factor that enhances the structural richness and realism of the dataset. By incorporating non-periodic diagnostic communication that occurs in real vehicle environments, the AutoHack Dataset more faithfully represents the complex distributional characteristics of in-vehicle networks and provides a more realistic and structurally challenging benchmark for IDS evaluation.

Observation 3: Empirical Characteristics of the Time-Synchronized Multi-Bus Setting The AutoHack Dataset differs from conventional single-bus datasets in that it simultaneously captures C-CAN, P-CAN, and B-CAN traffic under time synchronization. This configuration preserves the coexistence of multiple functional domains within the same vehicle and enables integrated analysis of bus-specific and cross-bus traffic characteristics that cannot be directly examined in single-bus datasets.

In Observation 3, we employ an expanded 38-feature configuration. This full configuration extends the feature set in Table 7 by adding 30 additional descriptors, including payload byte-level features, payload statistical features, per-ID rolling-window statistics, and payload entropy. We leverage this expanded representation to investigate how a wider range of features is reflected in the multi-bus setting. The complete feature list and extraction definitions for the expanded configuration are provided in the `feature_configuration.txt` document included in the Zenodo benchmark archive [17].

We first compare two training regimes using the same 38-feature representation. In the *single-bus* regime, a model is trained and evaluated on data from the same bus. In the *multi-bus* regime, a single model is trained on the union of all three buses and evaluated on the corresponding full test set; the predictions are then partitioned by the `Interface` field to report per-bus performance. As shown in Table 10, the multi-bus setting yields lower per-bus macro F1-scores than the single-bus setting across all three buses for both Random Forest and XGBoost. We additionally compared feature importance between the single-bus and multi-bus settings under the same 38-feature configuration, and the top-ranked features remained broadly similar across both regimes. Specifically, ID-aware

temporal and frequency features such as `ID_Prev_Interval` and `ID_Frequency` frequently appeared in the highest ranks. This suggests that the observed F1-score reduction is difficult to attribute to a significant shift in feature importance alone.The corresponding feature-importance summary is provided in Appendix **??**.

Since feature importance alone does not fully explain the observed reduction in F1-scores, we next examine an ID-level structural property specific to the integrated multi-bus setting. Although ID-aware temporal and frequency features remain highly ranked in both regimes, this very consistency suggests that ID context continues to play a pivotal role in classification. Rather than searching for shifts in feature importance, we investigate how IDs are organized across buses, prompting a closer look at structural ID-level properties. Specifically, we focus on overlapping CAN IDs for two reasons. First, given that ID-aware features remain dominant in both regimes, ID context is a natural candidate for further analysis. Second, unlike message-rate or payload-statistical features—which can be computed within an individual bus—overlapping CAN IDs are only observable when multiple buses are analyzed jointly. We therefore utilize overlapping CAN IDs as a structural variable to examine how the multi-bus setting influences error composition, rather than as the sole explanation for performance degradation.

One distinguishing characteristic of the time-synchronized multi-bus design of the AutoHack Dataset is that the same CAN identifier can be observed across different buses. We refer to such identifiers as *overlapping CAN IDs*. After excluding Fuzzing attacks, which span the entire ID space by construction, 40 overlap IDs are identified: seven between P-CAN and B-CAN, thirty-three between P-CAN and C-CAN, and four between B-CAN and C-CAN. A small subset of these IDs also exhibits label inconsistency across buses. Rather than treating overlapping IDs as a single determining factor, we present them as a structural characteristic that can arise in integrated multi-bus data and that is difficult to observe directly in conventional single-bus datasets.

To further examine how this structural characteristic appears in the multi-bus setting, we compare the proportion of overlap-ID messages in the full test set with that in the subset misclassified by the multi-bus model. The resulting pattern differs by class. As shown in Table 11, for Random Forest, the overlap-ID ratio increases for the Normal class (46.12% $\rightarrow$ 53.14%), remains nearly unchanged for Spoofing (25.47% $\rightarrow$ 25.75%), and decreases for Replay (41.20% $\rightarrow$ 38.52%). These results show that overlap-ID messages do not appear with a single common tendency across classes in the integrated multi-bus setting. In particular, the opposite directions observed for Normal and Replay suggest that the overlap-ID structure is reflected differently in the error composition of each class. In this sense, while domain shift may still contribute to the overall performance change, the observed class-wise variation indicates that the multi-bus setting cannot be understood solely in terms of a single uniform effect.

DoS, Fuzzing, and UDS were excluded from this comparison because their specific traffic characteristics render the overlap-ID ratio less informative. Specifically, DoS attacks yield an insufficient number of misclassified samples for analysis, Fuzzing attacks inherently span the entire ID space, and UDS involves non-periodic diagnostic traffic with no overlapping IDs in this dataset.

In summary, these results show that the time-synchronized multi-bus design of the AutoHack Dataset enables analysis of structural properties, such as overlapping CAN IDs, that are difficult to observe in conventional single-bus datasets. This is also meaningful in light of future vehicle architectures: as E/E systems evolve from functionally separated P-, B-, and C-CAN domains toward zonal architectures, bus organization may be restructured not only by function but also by physical location and centralized control. In such settings, interactions across previously distinct domains may become more tightly coupled, making multi-domain analysis increasingly relevant for IDS design and evaluation.

4.3 Dataset Fidelity and Benchmark Validity

The quantitative analyses and benchmark results presented in this study demonstrate that the AutoHack Dataset possesses structural characteristics and a level of realism that distinguish it from existing vehicular network datasets. As observed in Observations 1–3, the dataset incorporates varying levels of attack difficulty, the influence of non-periodic communication, and interactions arising from a multi-bus integrated systems.

In particular, unlike existing public datasets that primarily rely on periodic message patterns within a single bus, the Au-

Table 10: Macro F1-score of single-bus versus multi-bus training.

Model	Bus	Single-bus	Multi-bus	Δ
RF	B-CAN	0.8411	0.6840	−0.1572
	C-CAN	0.8368	0.7299	−0.1069
	P-CAN	0.8880	0.7775	−0.1104
XGBoost	B-CAN	0.7879	0.7339	−0.0540
	C-CAN	0.8613	0.7505	−0.1109
	P-CAN	0.8236	0.7791	−0.0445

Table 11: Comparison of overlap-ID ratios by class.

Class	r_{test}	r_{mis}	Direction
Normal	46.12%	53.14%	$\uparrow$
Spoofing	25.47%	25.75%	$\approx$
Replay	41.20%	38.52%	$\downarrow$

Table 12: Cyber-physical impact taxonomy: mapping attack types to physical domains and functional safety relevance.

Attack Type	Target Bus	Affected Domain	Physical Impact	Severity	Safety Relevance
DoS	C-CAN	Multi-domain	ECU warning lights, system error alarms	Medium	Degraded situational awareness
Fuzzing	P-CAN, B-CAN	Powertrain, Body	Engine stall in D-mode, unintended emergency call	Critical	Direct driving safety hazard
Spoofing	C-CAN	Chassis, Body	EPB warning, PAS logic error, false fuel warning	Medium–High	Misleading driver information
Replay	C-CAN, P-CAN, B-CAN	Control, Body	Arbitrary turn signal / window activation	Medium	Unintended actuator control
UDS Spoofing	C-CAN	ADAS, Steering	Camera deactivation, steering assist limitation	Critical	ADAS function loss during driving

toHack Dataset includes a complex communication structure across multiple buses as well as non-periodic messages such as UDS. These characteristics more faithfully reflect communication patterns observed in real vehicular environments and introduce a degree of practical complexity that has not been sufficiently addressed in prior datasets.

Furthermore, the benchmark results reveal that identical models exhibit performance variations depending on attack type and bus configuration. This indicates that distributional overlap, class imbalance, and inter-bus interactions within the dataset directly influence IDS performance. Such phenomena should not be interpreted merely as differences in model capability, but rather as consequences of the dataset's inherent structural properties.

In conclusion, the AutoHack Dataset provides a more challenging and realistic evaluation environment compared to existing datasets and can serve as a benchmark for more rigorously assessing the generalization capability of IDS models. These findings underscore the importance of incorporating realistic vehicular network conditions into future automotive security research and evaluation frameworks.

5 Discussion

5.1 Bridging Cyber Attacks and Physical Malfunctions

Beyond evaluating IDS performance, the proposed dataset introduces a new research direction by enabling the analysis of how in-vehicle network attacks affect the physical behavior of the vehicle. Automotive cybersecurity is increasingly recognized not merely as a network-level anomaly detection problem, but as a cyber-physical challenge in which the impact of cyber attacks on vehicle dynamics and operational safety must be jointly considered.

Various anomalous phenomena observed during the experiments illustrate this characteristic. When attacks were injected, abnormal changes in vehicle states were identified across multiple functional domains, including driving assistance and control systems, powertrain components, body control modules, and convenience features. These observations indicate that the attacks extend beyond communication-layer anomalies and are directly associated with the vehicle's control logic. In particular, certain attacks were found to influence safety-related functions, suggesting that automotive security concerns transcend traditional information security and are closely intertwined with functional safety.

To systematically formalize these observations, Table 12 presents a cyber-physical impact taxonomy that maps each attack type to its targeted bus, affected functional domain, physical impact, and relevance to functional safety. As illustrated, attacks such as Fuzzing on the P-CAN and B-CAN directly induce critical safety hazards, including engine stalls in D-mode and unintended emergency calls. Furthermore, UDS Spoofing on the C-CAN compromises the ADAS and steering domains, resulting in severe consequences like camera deactivation and steering assist limitation during driving. Even attacks classified with medium severity, such as DoS or Replay, cause degraded situational awareness and unintended actuator control. This taxonomy explicitly demonstrates that network-level intrusions in modern vehicles inevitably cascade into physical safety risks, providing a structured framework for assessing the functional safety implications of automotive cyber threats.

Accordingly, future vehicle security research should move beyond network-level detection and adopt an integrated perspective that accounts for cyber-physical interactions. Moreover, quantitatively establishing the causal relationships underlying these cyber-physical interactions remains an important open research challenge. Since the AutoHack Dataset includes both real attack injections and the corresponding physical effects observed in an actual vehicle, it provides a foundation for analyzing cyber-physical correlations that have been difficult to investigate using existing datasets.

5.2 Limitations and Future Directions

Although the proposed dataset was collected from a real vehicle platform, it is currently limited to a single vehicle model and a finite set of driving conditions. Therefore, the benchmark results obtained from the current AutoHack Dataset should not be interpreted as direct evidence of vehicle-independent generalization. CAN identifiers, payload semantics, gateway routing policies, diagnostic implementations, and message timing characteristics may vary across vehicle manufacturers, vehicle models, and E/E architectures. In addition, the included attack scenarios focus on representative attack types and do not comprehensively cover all possible attack vectors.

Nevertheless, this limitation concerns the current coverage of the dataset rather than the extensibility of the proposed dataset-construction methodology. Prior CAN IDS datasets show that comparable CAN-level attributes, attack families, and evaluation protocols can be applied across different vehicle platforms, even though the resulting identifiers, payload distributions, and functional semantics remain vehicle-specific. In this sense, AutoHack should be viewed as an initial high-fidelity multi-bus benchmark and a transferable data-construction protocol, rather than as a final vehicle-independent benchmark. The core components of our methodology—hardware-level synchronized multi-bus logging, real-time attack injection, physically verified labeling, UDS-aware diagnostic traffic collection, session-level train/test separation, and cross-ID spoofing evaluation—can be replicated on additional vehicle platforms.

Future work will extend the dataset by applying the same collection and evaluation protocol to multiple vehicle models, manufacturers, and driving environments. Such an extension would enable systematic cross-vehicle, cross-bus, and cross-manufacturer evaluation of IDS models, thereby providing a stronger basis for assessing the scalability and practical applicability of multi-bus vehicular IDS benchmarks in real deployment scenarios.

6 Conclusion

In this paper, we introduced AutoHack Dataset, an automotive cybersecurity dataset collected from a real vehicle platform. AutoHack provides a higher degree of realism compared to existing datasets by incorporating synchronized communication data from multiple CAN buses (C-CAN, P-CAN, and B-CAN), various attack scenarios, and validated physical impact information derived from actual vehicle behavior.

We conclude that IDS performance is strongly influenced by attack type, communication characteristics, and bus configuration, with aperiodic traffic and multiple buses settings revealing clear limitations of conventional periodicity- or single-bus–based approaches. These findings were further supported by external evaluation conducted as part of the AutoHack competition, demonstrating the dataset's benchmark validity and practical relevance for IDS research. Overall, our results provide empirically grounded evidence that automotive cybersecurity must be considered in close integration with functional safety, as network-level attacks can directly impact vehicle behavior.

Future work should focus on expanding the dataset to include diverse vehicle platforms and driving environments, as well as developing IDS architectures capable of effectively modeling the interactions and aperiodic communication patterns in multiple buses. AutoHack Dataset is expected to serve as a realistic and challenging benchmark to advance research in automotive cybersecurity.

References

[1] Mehmet Bozdal, Mohammad Samie, and Ian Jennions. A survey on can bus protocol: Attacks, challenges, and potential solutions. In *2018 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*, pages 201–205. IEEE, 2018.

[2] Irina Chiscop, András Gazdag, Joost Bosman, and Gergely Biczók. Detecting message modification attacks on the CAN bus with temporal convolutional networks. Proceedings of the 7th International Conference on Vehicle Technology and Intelligent Transport Systems, 2021.

[3] Guillaume Dupont, Alexios Lekidis, Jerry den Hartog, and Sandro Etalle. Automotive controller area network (CAN) bus intrusion dataset v2. 4TU.ResearchData, 2019. https://doi.org/10.4121/uuid:b74b4928-c377-4585-9432-2004dfa20a5d.

[4] Lorenzo Guerra, Linhan Xu, Paolo Bellavista, Thomas Chapuis, Guillaume Duc, Pavlo Mozharovskyi, and Van-Tam Nguyen. Ai-driven intrusion detection systems (ids) on the road dataset: A comparative analysis for automotive controller area network (can). In *Proceedings of the 2024 Cyber Security in CarS Workshop*, pages 39–49, 2023.

[5] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer. Canet: An unsupervised intrusion detection system for high dimensional can bus data. *IEEE Access*, 8:58194–58205, 2020.

[6] HCRL. CAN Intrusion Dataset (OTIDS). http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset, 2017.

[7] HCRL. Car Hacking for Intrusion Detection. http://ocslab.hksecurity.net/Datasets/CAN-intrusion-dataset, 2018.

[8] HCRL. Survival Analysis for Automotive IDS. http://ocslab.hksecurity.net/Datasets/survival-ids, 2018.

[9] HCRL. CAN Signal Extraction and Translation Dataset. https://ocslab.hksecurity.net/Datasets/can-signal-extraction-and-translation-dataset, 2021.

[10] HCRL. B-CAN Intrusion Dataset. https://ocslab.hksecurity.net/Datasets/b-can-intrusion-dataset, 2022.

[11] HCRL. M-CAN Intrusion Dataset. https://ocslab.hksecurity.net/Datasets/m-can-intrusion-dataset, 2022.

[12] Kookmin University. Kookmin University Successfully Hosts 'CO-SHOW AutoHack 2025'. https://english.kookmin.ac.kr/kookmin/newsplus/616, 2025. Kookmin University News Plus. Accessed: 2026-05-04.

[13] Brooke Lampe and Weizhi Meng. Intrusion detection in the automotive domain: A comprehensive review. *IEEE Communications Surveys & Tutorials*, 25(4):2356–2426, 2023.

[14] Brooke Lampe and Weizhi Meng. can-train-and-test: A curated can dataset for automotive intrusion detection. *Computers Security*, 140:103777, 2024.

[15] Mobility Cybersecurity Lab. AutoHack2025. https://mose.kookmin.ac.kr/mose/AutoHack2025.do, 2025. Kookmin University Mobility Cybersecurity Lab. Accessed: 2026-04-30.

[16] Euclides Carlos Pinto Neto, Hamideh Taslimasa, Sajjad Dadkhah, Shahrear Iqbal, Pulei Xiong, Taufiq Rahman, and Ali A Ghorbani. Ciciov2024: Advancing realistic ids approaches against dos and spoofing attack in iov can bus. *Internet of Things*, 26:101209, 2024.

[17] Yuchan Song, Sejun Ahn, Seungjin Baek, Hyeonseong Kim, Huy Kang Kim, and Sanghoon Jeon. AutoHack Dataset and Benchmark Code: A Physically Verified Multi-Bus CAN Dataset for Intrusion Detection System Evaluation. Zenodo. https://doi.org/10.5281/zenodo.20321379, 2026. Includes the AutoHack dataset archive and the benchmark implementation code.

[18] Miki E. Verma, Robert A. Bridges, Michael D. Iannacone, Samuel C. Hollifield, Pablo Moriano, Steven C. Hespeler, Bill Kay, and Frank L. Combs. A comprehensive guide to can ids data and introduction of the road dataset, 2024.

[19] Armin Wasicek et al. CAN-MIRGU: A Dataset for Moving Vehicle Intrusion Detection. In *Proceedings of the 31st Network and Distributed System Security (NDSS) Symposium*, 2024.

A Artifact Appendix

A.1 Description & Requirements

A.1.1 How to access

The **AutoHack artifact** is permanently archived on Zenodo (DOI: https://doi.org/10.5281/zenodo.20321379), containing both **Component 1** (*AutoHack Dataset*) and **Component 2** (*AutoHack Benchmark Code*). After downloading the archive, evaluators should consult the `README.md` file in the root directory for environment setup, recommended execution order, and reproduction steps.

A.1.2 Hardware dependencies

The artifact supports two evaluation tiers. The **Minimal** configuration for Component 1 dataset verification requires 2+ CPU cores, 8 GB RAM, and 2 GB storage. The **Full** configuration for Components 1+2 benchmark reproduction requires 8+ CPU cores, 32 GB RAM, and 30 GB SSD storage. The full benchmark was tested on a 24-core CPU with 32 GB RAM and SSD storage using Docker flags `--cpus=20` and `--memory=28g`; evaluators may lower `--cpus` to match their available cores. All evaluations are CPU-bound, and no GPU, FPGA, proprietary hardware, hardware counters, or remote SSH access is required.

A.2 Set-up

A.2.1 Installation

To set up the artifact, download the two archives from the Zenodo record 10.5281/zenodo.20321379: `Autohack2025_Dataset.zip` (Component 1) and `AutoHack-Benchmark.zip` (Component 2). Extract both archives in the working directory. For local execution, place the extracted `Autohack2025_Dataset/` directory under `AutoHack-Benchmark/`. Detailed Linux and Windows commands are provided in `README.md`.

A.2.2 Basic Test

To verify the dataset integrity and ensure all statistical metrics match the paper, run the following command from the root directory:

```
python check_dataset.py
```

The script is pre-configured to load `autohack_train_both_interface.csv` and `autohack_test_both_interface.csv` from the `Interface` directory. It performs a comprehensive validation of schema consistency, null values, temporal integrity, and DLC ranges. The test is successful if the output reproduces the statistical tables and concludes with the `[PASSED]` status.

Expected Output. Upon successful execution, the terminal should display the following logs. Please confirm that the frame counts for each bus match the statistics reported in Table 4 and Table 5 of the main paper:

```
Loading Train Set from: Interface\train\autohack_train_both_interface.csv
[Train Set] Verifying data integrity...
Integrity Check PASSED for Train Set.

Loading Test Set from: Interface\test\autohack_test_both_interface.csv
[Test Set] Verifying data integrity...
Integrity Check PASSED for Test Set.

============================================================
Table 4: Detailed attack statistics by interface and session.
============================================================
[C-CAN]
Attack Type     |       Train Set |        Test Set
-----------------------------------------------------
Total Frames    |       4,026,488 |       1,678,489
Fuzzing         |          38,191 |          13,147
... (omitted for brevity) ...

============================================================
FINAL STATUS: [PASSED] Dataset is valid and Ready-to-Use.
============================================================
```

A.3 Evaluation workflow

A.3.1 Major Claims

Component 1 — Dataset Claims:

The AutoHack dataset is designed to provide a realistic and challenging environment for vehicular IDS evaluation, supported by the following claims:

(C1): **Hardware-level Synchronized Multi-bus Realism.** AutoHack provides microsecond-level synchronized logs from three separate CAN interfaces (C/P/B-CAN) and incorporates realistic non-periodic UDS interactions, distinguishing it from existing periodicity-reliant datasets.

(C2): **Physically Verified Reliability and Benchmark Challenge.** The ground-truth labels are secured through direct physical verification on a real vehicle, providing a challenging benchmark where sophisticated attacks exhibit high distributional similarity to legitimate traffic.

Component 2 — Benchmark Claims:

(C3): The dataset serves as a realistic and challenging IDS benchmark with three structural properties:

- **(C3-i)** Replay and Spoofing are harder to detect than DoS, Fuzzing, and UDS_Spoofing because they overlap more with normal traffic (RF F1: 0.6799 and 0.7864 vs. ≥ 0.93 for the others).
- **(C3-ii)** Removing non-periodic UDS traffic mainly shifts the Normal–Fuzzing boundary: Normal→Fuzzing errors increase from 966 to 5,296, while Macro/Weighted F1 remain about 0.86/0.98.
- **(C3-iii)** Joint multi-bus training lowers per-bus macro-F1 on all three buses for both Random Forest and XGBoost, while key features remain similar. Overlap-ID ratios

shift by class: Normal rises (46.12% → 53.14%), Spoofing stays stable (25.47% ≈ 25.75%), and Replay falls (41.20% → 38.52%).

A.3.2 Experiments

The following experiments reproduce the three IDS observations in Section 4.2 of the main paper. All commands are run under `AutoHack-Benchmark/`; detailed OS-specific commands are provided in `README.md`.

(D1) Observation 1 — Per-Attack Detection Difficulty
[10 human-min + ∼8 compute-min + 15GB disk]. Supports **C3(i)** and reproduces Table 8 and Figure 3. Run `python preprocess.py` and `python observation1.py`. The experiment passes if all three conditions hold: (a) Normal, DoS, Fuzzing, and `UDS_Spoofing` achieve F1 $\geq$ 0.93; (b) Spoofing and Replay are the two lowest-F1 classes and both fall below 0.80; (c) the three largest off-diagonal cells are Replay→Normal, Spoofing→Normal, and Normal→Replay, each exceeding 5,000.

(D2) Observation 2 — Non-Periodic UDS Impact
[10 human-min + ∼5 compute-min + 15GB disk]. Supports **C3(ii)** and reproduces Table 9 and Figure 4. Run `python observation2.py` using the 8-feature files from (D1). The experiment passes if (a) the Normal→Fuzzing cell increases by at least 2× compared with (D1), and (b) all other off-diagonal cells (Normal→Spoofing, Normal→Replay, Spoofing→Normal, Replay→Normal) change by less than 1.3× in either direction.

(D3) Observation 3 — Multi-Bus Characteristics
[20 human-min + ∼100 compute-min + 30GB disk]. Supports **C3(iii)** and reproduces Tables 10–11. Run `python preprocess38f.py` and `python observation3.py`. The experiment passes if all three conditions hold: (a) all RF/XGBoost single-bus→multi-bus Macro-F1 deltas are negative across B-CAN, C-CAN, and P-CAN; (b) `ID_Prev_Interval` and `ID_Frequency` remain among the top features in both single- and multi-bus settings; (c) RF overlap-ID ratios show diverging class trends: Normal increases (∼46% → ∼53%), Spoofing stays stable (∼25% ≈ ∼26%), and Replay decreases (∼41% → ∼39%).

Docker execution. D1–D3 can also be reproduced with the provided Docker image using the resource flags in Section A.1.2. Use `--cpus=8` and `--memory=24g` for the minimum full configuration, or `--cpus=20` and `--memory=28g` for the tested configuration. The same pass criteria above apply.

Determinism. All benchmark scripts fix `random_state=42`; minor numerical variation may occur across machines due to floating-point and thread-scheduling non-determinism in scikit-learn / XGBoost, but the qualitative trends supporting C3 are stable.

A.4 Notes on Reusability

To facilitate broader research in automotive security, this artifact is designed with modularity and extensibility in mind.

Reuse Scenarios for Component 1 (AutoHack Dataset).

- **Cross-Domain Analysis:** The synchronized C-, P-, and B-CAN logs support multi-domain IDS research, including cross-bus correlation and attack propagation analysis.
- **Non-Periodic Traffic Analysis:** UDS-based event-driven messages enable evaluation of anomaly detection methods in realistic non-periodic vehicle traffic.
- **Feature Engineering Benchmarking:** The raw CSV logs allow researchers to design and evaluate alternative feature sets on the same ground truth.

Reuse Scenarios for Component 2 (IDS Baselines).

- **Model Comparison:** New machine learning or deep learning models can be integrated and compared directly against the provided Random Forest and XGBoost baselines.
- **Multi-Bus Research:** The benchmark supports studies on cross-bus transfer learning, domain shift, zonal-architecture IDS design, and overlap-ID analysis.
- **Custom Feature Engineering and Education:** The raw five-column log format and reference preprocessing pipelines (`preprocess/`) support both custom feature design and educational use in reproducible IDS experiments.

License. The AutoHack Dataset is licensed under the **Creative Commons Attribution 4.0 International (CC BY 4.0)** license. The source code for evaluation scripts and IDS baselines is released under the **MIT License**.

A.5 Version

Based on the LaTeX template for Artifact Evaluation V20260101. Submission, reviewing and badging methodology followed for the evaluation of this artifact can be found at https://secartifacts.github.io/vehiclesec2026/.