D.I.LAB
숭실대학교 분산지능연구실

SEMINAR #01

Anomaly detection on vehicle data

Efficient intrusion detection for the intra-vehicle CAN network with a lightweight deep-learning model.

Ankit Kumar Singh
Ph.D. Student · DILab
Soongsil University, Korea

CO-AUTHORS
M. Ali · S. Moharana · S. Choi · B. J. Choi

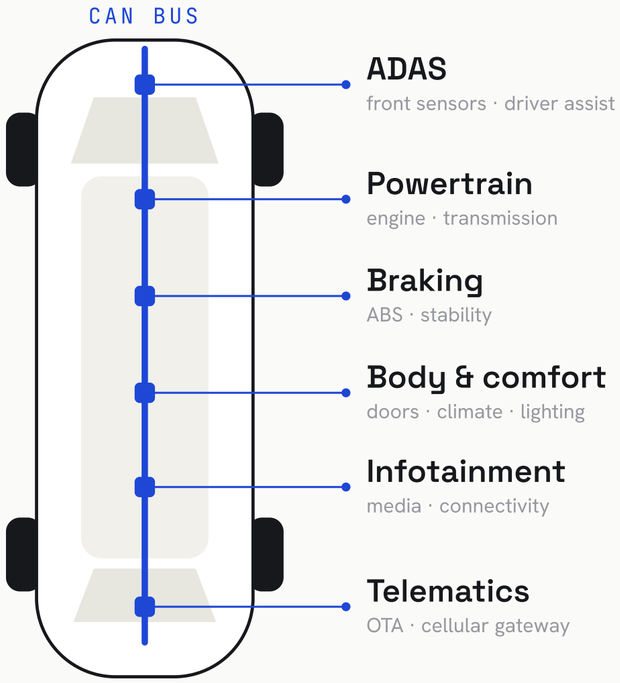
Jun 01, 2026

01 BACKGROUND

Modern vehicles rely on networked ECUs

A modern vehicle runs **70–100+ small computers** — Electronic Control Units (ECUs) — that constantly exchange messages to keep the car running and safe.

They steer, brake, manage the engine, and run driver-assist features. None of it works unless they can talk to each other in real time.



Dozens of ECUs share a single CAN bus.

The CAN bus and its message frame

The **Controller Area Network (CAN) bus** is the shared line every ECU talks on. It's cheap, robust, and fast — the automotive industry standard for decades. Each message is a small, structured frame:

TIMESTAMP	CAN ID	DLC	PAYLOAD
1479121434.850202	0350	8	05 28 84 66 6d 00 00 a2

When the message was seen on the bus.

Which ECU / signal it belongs to (priority).

How long the data field is, in bytes.

The data itself — raw sensor & control values.

03 MOTIVATION

CAN has no built-in authentication or encryption

CAN has **no authentication and no encryption.** Any node on the bus can read every message — and inject its own.

Once an attacker reaches the bus — through infotainment, telematics, or an OTA channel — they can impersonate a real ECU. Researchers have remotely controlled production cars this way.

Injection / DoS

Flood the bus to drown real messages.

Spoofing

Send fake sensor or control values.

Fuzzing

Random payloads to trigger faults.

Masquerade

Mimic a trusted ECU's normal traffic.

04 PROBLEM

Intrusion detection for the CAN bus

Because the protocol cannot be re-secured, an **Intrusion Detection System (IDS)** monitors bus traffic, models normal behavior, and flags deviations as potential attacks.

It runs alongside the existing ECUs as an added monitoring layer — the early-warning capability that authentication and encryption never provided.

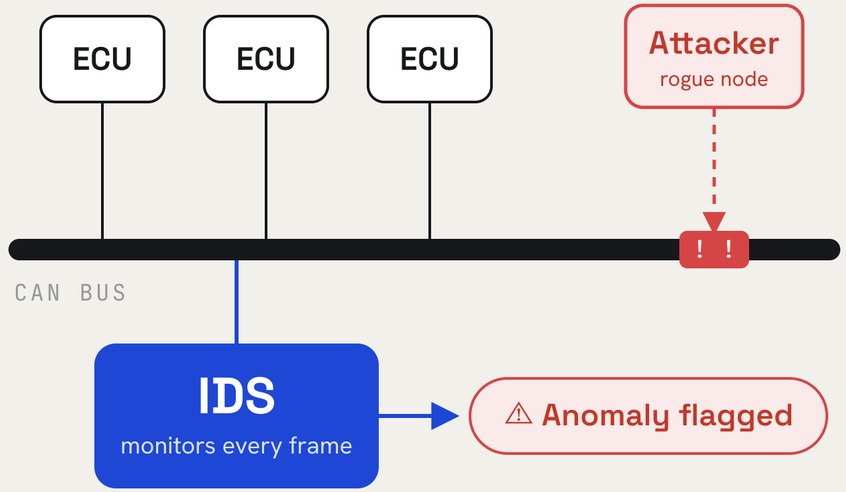


Fig. 1 — IDS flags injected traffic on the bus.

05 CHALLENGES

Four challenges for in-vehicle IDS

01

Tiny computers

ECUs are low-power microcontrollers. The detector has to be small.

02

Hard real-time

At 100 km/h, a few milliseconds of delay can be fatal.

03

No false alarms

Flagging normal driving as an attack erodes all trust.

04

Stealthy attacks

Masquerade & adversarial attacks look almost normal.

06 RELATED WORK

Prior approaches trade accuracy for deployability

Rule-based

Fixed checks on each message.

`Light & explainable`

`Misses new attacks`

Statistical

Watches timing & frequency.

`Lightweight`

`High false positives`

Deep learning

Learns patterns from data.

`High accuracy`

`Too heavy for ECUs`

Transformer-based deep models reach high accuracy but are **too large for resource-constrained ECUs.** Our work targets this accuracy–efficiency gap with a lightweight model.

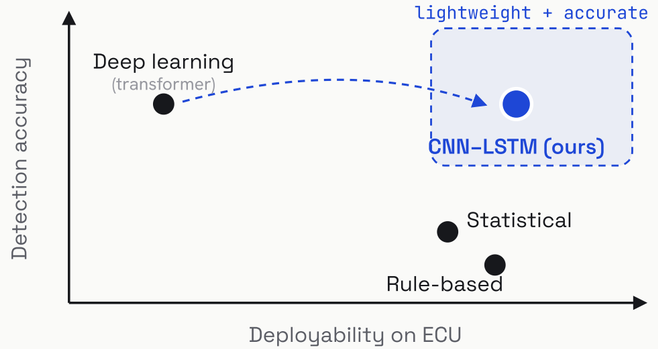


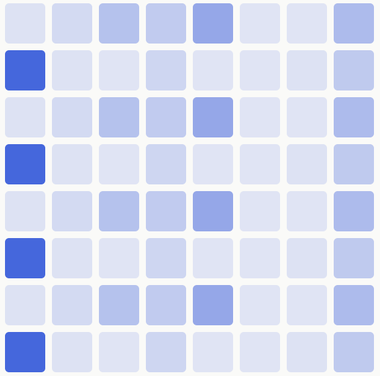
Fig. — The accuracy–deployability gap our model targets.

07 METHOD

Feature engineering and payload encoding

TIMESTAMP	CAN ID	DLC	PAYLOAD → 8×8
1479121434.850202	0350	8	05 28 84 66 6d 00 00 a2

Payload splitting. The 8-byte data field is unfolded into an **8×8 image** — so the CNN can see structure the way it sees pixels.



Feature engineering. We add temporal fingerprints that attacks tend to disturb:

- Time delta
- ID frequency
- Hamming distance
- Information entropy
- Bus load

Each captures a different tell — sudden timing shifts, flooding, or impossible jumps in a sensor value.

08 PROPOSED METHOD

A lightweight CNN–LSTM detector

CNN features from the payload image are **fused** with engineered temporal features; an LSTM + linear head then emits a binary or multiclass verdict.

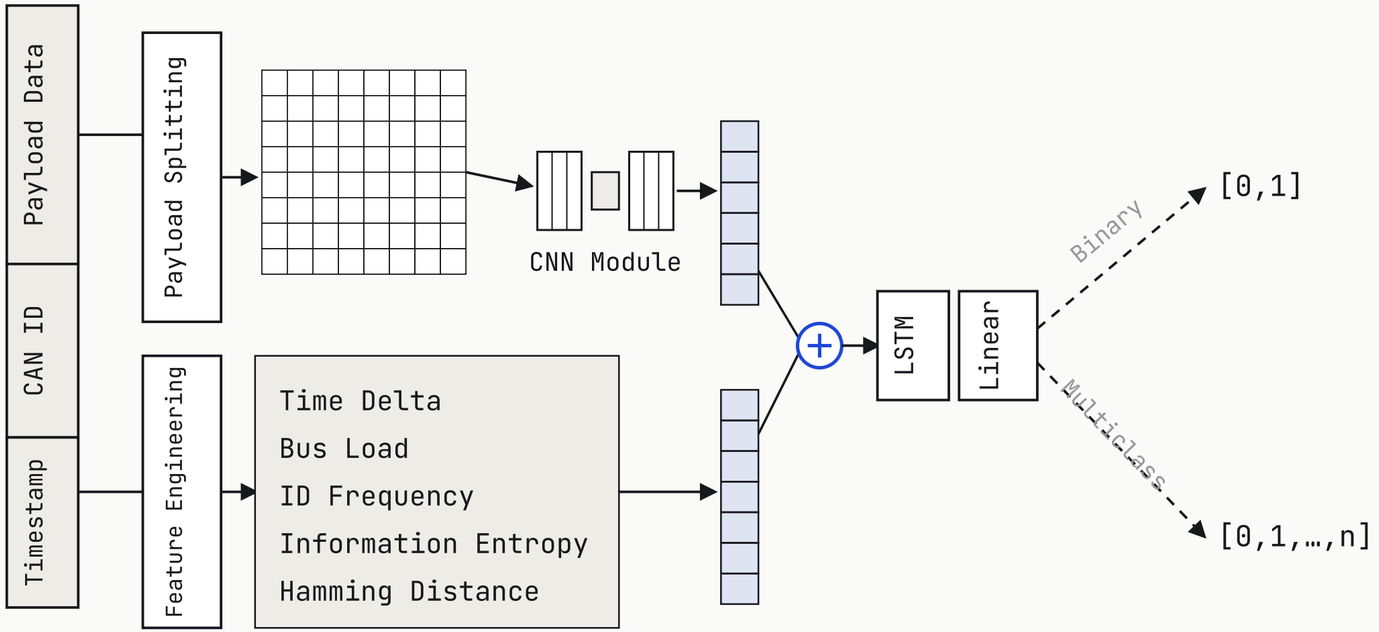


Fig. 2 — CNN-LSTM with feature extraction and model fusion.

11 EFFICIENCY

Smaller and faster than the transformer

MODEL SIZE — PARAMETERS

Transformer baseline

457,777 params

CNN-LSTM (ours)

38,597 params

8.4%

of the parameters — about **12× smaller**.

INFERENCE LATENCY — PER SAMPLE

Transformer baseline

0.024–0.027 ms

CNN-LSTM (ours)

0.019–0.020 ms

~28%

faster inference, on average.

Same 99.99% accuracy on Car-Hacking — with a fraction of the compute, built to run on a real ECU.

12 RESULTS

At a glance — ours vs. the transformer

F1 across both detection tasks and both datasets — the easy benchmark (Car-Hacking) and the hard one (ROAD, masquerade + fabrication).

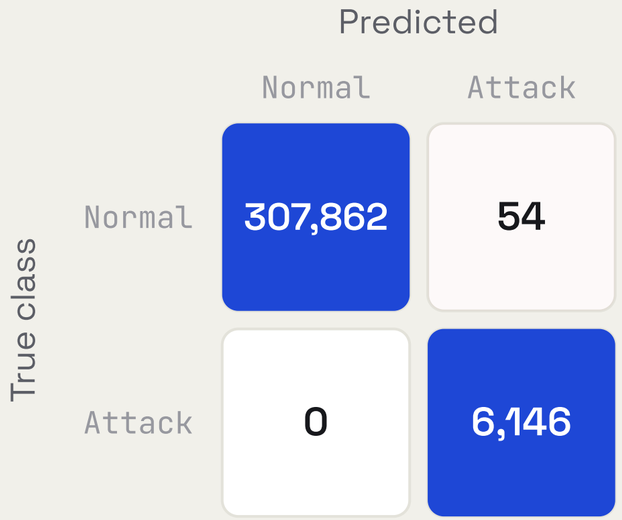
	MULTICLASS · F1		BINARY · F1		
METHOD	CAR-HACKING	ROAD	CAR-HACKING	ROAD	PARAMS
Transformer	0.9999	0.7410	0.9999	0.9442	≈ 457K
CNN-LSTM (ours)	1.0000	0.8221	1.0000	0.9270	≈ 38K

All values are F1 scores; ROAD = combined masquerade + fabrication. CNN-LSTM ties on Car-Hacking and wins the harder multiclass-ROAD set; the transformer leads only on binary-ROAD — at **≈ 12× the parameters**. Parameter counts shown approximately (binary / multiclass variants differ slightly).

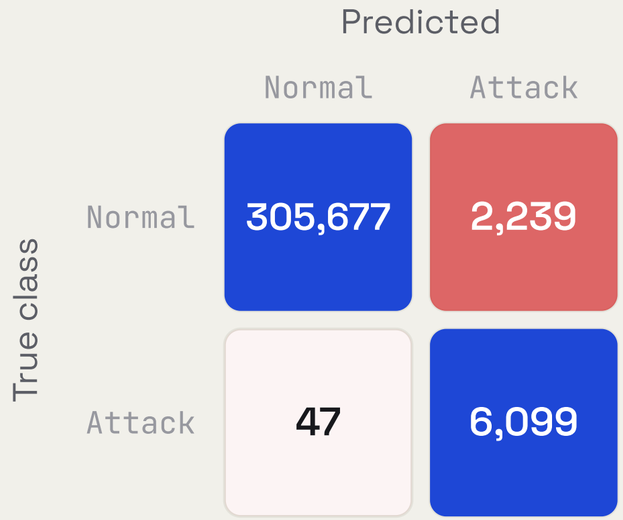
A7 ERROR ANALYSIS

Where the errors go — ROAD-Masquerade

Binary detection (Normal vs. attack). Both models hold a near-perfect diagonal — but the transformer raises **2,239 false alarms** against the CNN-LSTM's 54, and misses 47 attacks the CNN-LSTM catches.



CNN-LSTM — 54 false alarms, 0 missed attacks. F1 = 0.9976, FPR = 0.0001.



Transformer — 2,239 false alarms, 47 missed attacks. F1 = 0.9192, FPR = 0.0072.

Absolute frame counts (Table 14, ROAD-Masquerade). Diagonal = correct; off-diagonal error cells are shaded by volume. The transformer's false-positive load drives its lower precision (0.8657 vs 0.9955) and F1 (0.9192 vs 0.9976).

D.I.LAB
숭실대학교 분산지능연구실

— THANK YOU

Questions & discussion

Happy to discuss the method, the datasets, the results, or directions for future work.

Ankit Kumar Singh
Ph.D. Student · DILab
Soongsil University, Korea

ankit@soongsil.ac.kr

Singh et al., *Efficient Intrusion Detection for Intra-Vehicle CAN Networks* — IEEE Access, 2026 · github.com/0eai/CANomaly

D.I.LAB

— BACKUP SLIDES

Appendix

Supporting detail on feature definitions, rule-based detection, datasets, and full results.

A1 FEATURE ENGINEERING

Engineered temporal features

Time delta · Δt

Inter-arrival gap between consecutive messages of the same arbitration ID.

$$\Delta t_i = t_i - t_{i-1}$$

ID frequency

Rolling mean & standard deviation of Δt per ID over a rolling window.

$$\bar{x}_W = (1/W) \sum x_j$$

$$\sigma_W = \sqrt{[\ (1/(W-1)) \sum (x_j - \bar{x}_W)^2\ ]}$$

Hamming distance

Number of differing bits between two consecutive 64-bit payloads.

$$D_H(B_1,B_2) = \sum_{k=1}^{64} (B_{1,k} \oplus B_{2,k})$$

Information entropy

Shannon entropy of the CAN-ID stream over the rolling window.

$$H(X) = -\sum_{i=1}^{n} P(x_i) \log_2 P(x_i)$$

Bus load

Count of all messages over a fixed period — overall traffic density on the bus.

$$L = \#\{\ \text{messages in } \Delta T\ \}$$

A2 RULE-BASED DETECTION

Seven complementary static checks

Formality

ID must belong to the known legitimate set.

$$ID \in S_{valid}$$

Frequency

Inter-arrival time within the normal profile.

$$\mu - k\sigma \leq \Delta t \leq \mu + k\sigma$$

Range

Each payload byte within learned bounds.

$$q_{lo} \leq b_i \leq q_{hi}$$

DLC

Data length must be valid for that ID.

$$DLC = DLC_{exp}(ID)$$

Payload invariance

Invariant payload bytes must not change.

$$b_i = const$$

Hamming

Bit-change magnitude within the normal range.

$$|D_H - \mu| \leq k\sigma$$

Entropy

Rolling ID entropy within the normal band.

$$|H - \mu_H| \leq k\sigma$$

A3 LABELING APPROACH

Per-message label matching

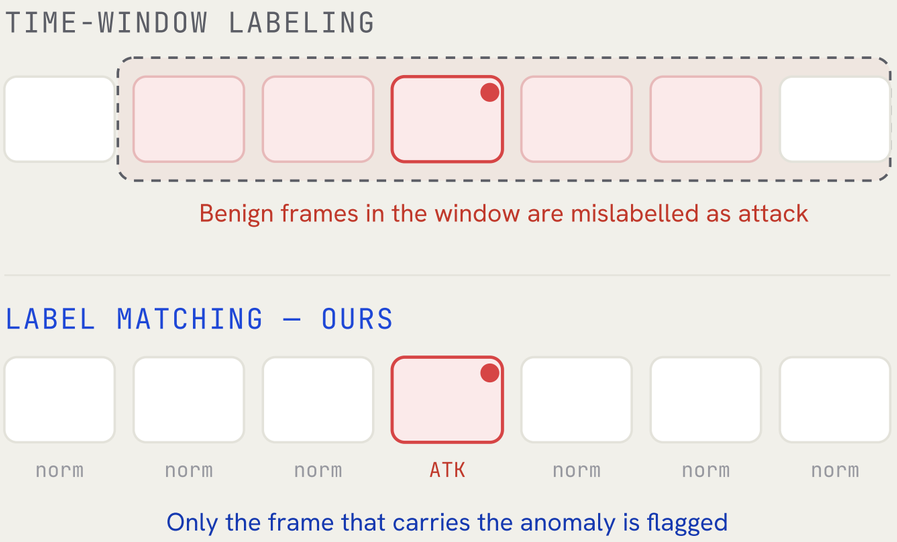


Fig. 3 — Label-matching scheme.

Time-window labeling can mislabel individual frames within a window, masking stealthy attacks.

Label matching instead assigns a verdict to **each set of attributes**, aligning labels with the messages that actually carry the anomaly.

A4 DATASETS

Two real-vehicle CAN datasets

HCRL Car-Hacking

- Hyundai YF Sonata, captured via OBD-II
- Attacks: DoS, fuzzy, gear & RPM spoofing
- Remote-frame capture at 0.01 s intervals
- Raspberry Pi 3 + PiCAN2 collection

ORNL ROAD

- 3.5 h of single-vehicle dynamometer testing
- 33 distinct attack types
- Fabrication, masquerade, advanced & fuzzing
- ~3 h benign + 30 min attack data

09 RESULTS

Multiclass detection performance

100%

Accuracy, F1 & ROC-AUC

HCRL Car-Hacking dataset

0.9973

F1 — masquerade attacks

ORNL ROAD · multiclass

0.8531

F1 — fabrication attacks

ORNL ROAD · multiclass

AGAINST THE TRANSFORMER · MULTICLASS

	Transformer	CNN-LSTM (ours)
Car-Hacking · F1	99.99%	**100%**
ROAD combined · F1	0.7410	**0.8221**

Car-Hacking is perfectly separable (1.0 on every metric). ROAD headline figures are per-attack-family; the comparison row uses the harder combined ROAD setting.

10 RESULTS

Binary detection performance

The deployment-facing task: flag each frame as **benign or attack**.

1.000

F1 — perfect detection

HCRL Car-Hacking · Acc 0.9999

0.9976

F1 — masquerade attacks

ORNL ROAD · FPR 0.0001

0.9578

F1 — fabrication attacks

ORNL ROAD · per-attack

AGAINST THE TRANSFORMER · BINARY F1

	Transformer	CNN-LSTM (ours)
ROAD masquerade	0.9192	**0.9976**
ROAD fabrication	0.9545	**0.9578**
ROAD combined	0.9442	**0.9270**

CNN-LSTM leads on each attack family; the transformer edges ahead only on the combined set — at ~12× the parameters.

A5 DETAILED RESULTS

Multiclass detection by model

MODEL	CAR-HACKING F1	ROAD F1	ROAD ROC-AUC	PARAMETERS
CNN	0.9289	0.5542	0.9727	223,653
LSTM	1.0000	0.7022	0.9959	215,941
Transformer	0.9999	0.7410	0.9989	457,777
CNN–LSTM (ours)	1.0000	0.8221	0.9950	38,597

ROAD figures are the combined masquerade + fabrication multiclass setting. Per-attack-family multiclass F1: 0.9973 (masquerade), 0.8531 (fabrication).

A6 DETAILED RESULTS

Binary detection by model

MODEL	CAR-HACKING	ROAD M+F	ROAD-MASQ	ROAD-FAB
Rule-based	0.9860	0.7316	0.7612	0.7152
Hybrid	0.7470	0.5736	0.5164	0.6173
Transformer	0.9999	0.9442	0.9192	0.9545
CNN-LSTM (ours)	1.0000	0.9270	0.9976	0.9578

All values are F1 scores (binary = benign vs. attack). CNN-LSTM wins both ROAD attack subcategories; the transformer edges the combined ROAD set (0.9442 vs 0.9270) but at ≈12× the parameters and a higher false-positive rate.