

Edge Insights

On-device CAN anomaly analytics — from development to production.

CATEGORY On-device CAN analytics	PLATFORMS Windows · Linux x86-64 · Linux arm64	CAN INTERFACES Any SocketCAN device · Kvaser CANlib	BUS / PROTOCOLS CAN · CAN FD · DBC · J1939 · NMEA 2000
DETECTION Timing · Congestion · Error frames · Unknown ID	DEPLOYMENT Laptop · PC · Embedded arm64 (Kvaser Edge)	DATA Fully local — no cloud required	LICENSING Per-device · free during Beta · EUR 600/year

1 Overview

Edge Insights is an on-device analytics tool for CAN networks. It records bus traffic, decodes it against your DBC, learns what your bus normally looks like, and flags timing, congestion, error-frame and unknown-identifier anomalies.

It runs on an engineer's Windows or Linux PC during development and test, and on an embedded Linux device in the vehicle for production monitoring. Processing and storage stay on the machine. In one line: **it turns raw CAN traffic into reviewable anomalies, locally, without scripting.**

2 Processing pipeline

Stage	Function
Capture	Reads frames via SocketCAN (Linux) or Kvaser CANlib (Windows). Classic CAN and CAN FD, timestamped to local storage.
Decode	DBC-driven signal decode: scaling, byte order, multiplexing. In J1939 and NMEA 2000 mode, messages match the DBC by PGN.
Teach — baseline	A configurable learn window records identifiers and per-message timing, then freezes a baseline. Direct, Learn and Detect modes; the baseline survives a restart.
Detect	Analyzers score live traffic against the baseline and rules (identifier allow-list, DBC). Deviations become anomaly events.
Surface	Anomalies, aggregates and decoded signals are stored on the device and presented in dashboards, logs and an HTTP API.

3 Anomaly detection

Class	Detects	Method
Timing	Messages arriving late, early, or with missed cycles	Against each identifier's learned period range. Periodic, aperiodic and burst are classified automatically.
Congestion	Bus-load approaching saturation; micro-bursts	Windowed bus-utilization measurement.
Error frames	Elevated bus-error activity	Per-window error-frame counts against a threshold.
Unknown identifier	Identifiers absent from the learned allow-list or DBC	Membership test against the baseline allow-list, seeded from the DBC and declared identifiers.

Learned, not rule-shipped. The baseline comes from *your* observed traffic, with DBC and allow-list rules layered on top. It is deterministic and inspectable. Learned IDs can be pruned, and timing thresholds adjusted by re-teaching.

4 Data, signals & storage

Protocols	Classic CAN, CAN FD; generic DBC; J1939 and NMEA 2000 (single-frame PGNs). Multi-frame transport (J1939 TP / DMI DTCs, NMEA 2000 Fast Packet) is roadmap.
Interfaces	Linux: any SocketCAN device. Windows: Kvaser CANlib. Single channel today; multi-channel on roadmap.
Raw retention	Stored on the device; size-based quota (oldest data retired first).
Analytics retention	Stored on the device; time-based (hours) or retained indefinitely.
Upgrades	Recorded logs, learned baselines and anomaly history survive an update.
Edge data reduction	Anomalies are kept permanently; routine data is pruned as storage fills up. The raw context around a flagged anomaly can be downloaded; automatic retention around events is roadmap.
Flash-aware I/O	Storage is written in a flash-friendly way, to extend the life of eMMC and SD cards.
Offline re-analysis	Import a log and re-run the same pipeline on a PC. Formats: MDF v4.11 (.mf4), ASC (.asc), Kvaser (.txt / .kme), candump (.log).

5 Deployment & data governance

PC role	Windows (Kvaser CANlib) or Linux (SocketCAN). Inspect-and-close use on an engineer's machine. The portal is local-only by default.
Embedded role	Always-on appliance on any arm64 Linux device; validated on Kvaser Edge (Debian). The portal is reachable over the network when installed with network access. Turnkey embedded deployment on roadmap.
Data locality	All capture, decode, analysis and storage happen on the device. Nothing leaves it: no bus traffic, no decoded signals, no analytics. The portal binds to the local machine by default; network access is an explicit choice at install time.
Cloud	None required. Optional cloud replication, opt-in and off by default.
Access control	Authentication, TLS, role-based access control and encryption at rest are roadmap.

CanEduDev is a Kvaser reseller. We can supply the Kvaser interface or edge hardware, or help you deploy on hardware you already own.

6 Outputs & integration

- **Dashboards.** Extendable Grafana views for bus health, timing and anomaly review.
- **Unified anomaly view.** Every anomaly class in one view.
- **Log export.** Raw or decoded, by time range or around a flagged anomaly. Verbatim, candump text or zipped, with a size estimate up front.
- **HTTP API.** Pull interface state, logs and export status into your own scripts. Versioned, so integrations keep working. Today's endpoints are read-only.
- **Help and support in the portal.** Open a support request with the application log attached.
- **Alerting.** Email, SMS and webhook are roadmap. Anomalies are recorded, not pushed.

7 Feature & availability matrix

Capability	Status
Timing, congestion, error-frame, unknown-ID detection	AVAILABLE
Learned baseline / Teach mode (Direct · Learn · Detect)	AVAILABLE
DBC decode; J1939 & NMEA 2000 single-frame PGNs	AVAILABLE
Classic CAN & CAN FD	AVAILABLE
Any SocketCAN interface (Linux); Kvaser CANlib (Windows)	AVAILABLE
arm64 embedded (validated on Kvaser Edge)	AVAILABLE
Grafana dashboards, log export, offline log re-analysis	AVAILABLE
HTTP API for scripted access	AVAILABLE
Signal-value anomaly detection	ROADMAP
Multi-channel CAN monitoring	ROADMAP
J1939 Transport Protocol / DMI DTCs; NMEA 2000 Fast Packet	ROADMAP
Real-time alerting (email / SMS / webhook)	ROADMAP
Optional cloud replication	ROADMAP
Auth, TLS, RBAC, encryption at rest	ROADMAP
Turnkey embedded deployment; fleet management; OTA updates	ROADMAP

8 Scope — what Edge Insights is *not*

- * **Not a real-time control or safety system** — it observes and analyzes, outside the control loop.
- * **Not a physical-layer instrument** — frames and signals, not volts and edges.
- * **Not a cloud service** — no mandatory cloud, account or upload.
- * **Not a DBC editor or bus/rest-bus simulator** — it reads your DBC and observes traffic.
- * **Not an ECU diagnostic or flashing suite** — no UDS testing, no ECU memory or firmware access.

9 Specifications & licensing

Operating systems	Windows and Linux on x86-64; Linux on arm64 (validated on Kvaser Edge, Debian).
CAN backends	SocketCAN — any compatible interface (Linux); Kvaser CANlib (Windows).
Bus types	Classic CAN, CAN FD.
Protocols	Generic DBC; J1939 and NMEA 2000 (single-frame PGNs).
Channels	Single channel today. Multi-channel on roadmap.
Access	Desktop application on Windows and Linux. On an embedded unit, the same interface opens in a browser over the network. HTTP API for scripted access on both.
Distribution	Signed Windows installer, Debian package, or tarball for other Linux distributions. Sign up at canedudev.com for the download link and license key.
Licensing	Per-device license, tied to the machine it is activated on and transferable to another. Free during the Beta program; EUR 600 per device per year thereafter, indicative until commercial launch. Current pricing at canedudev.com . Activation is online; the unit runs offline afterward.

CanEduDev — a Swedish engineering team with 50 years of combined CAN experience. Edge Insights is co-financed by Vinnova, Sweden's innovation agency. Complete product documentation is published at docs.cededgeinsights.com and updated continuously with the product.

Roadmap items are indicative and not a commitment to a delivery date.



Edge-first analytics for CAN data.

Automated CAN analysis at the edge — raw bus traffic turned into reviewable anomalies, locally, without scripting.

WEB

cededgeinsights.com

EMAIL

info@canedudev.com

Complete product documentation is published at docs.cededgeinsights.com and updated continuously with the product.