Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

A unified threat-scoring detection algorithm for cybersecurity attacks in automotive CAN networks

Ivan Ivanov[1]

[1]Technical University of Varna, 9010 Varna, ul. Studentska 1, Bulgaria

Corresponding author contact: ivan.or.ivanov@tu-varna.bg

Abstract. *Controller Area Network (CAN) is the primary in-vehicle communication backbone but provides no built-in security, leaving vehicles susceptible to message-injection attacks. This study presents a novel algorithm that (1) systematically classifies CAN attack types, (2) computes an impact-weighted probabilistic risk score per attack instance, (3) supplies an on-bus simulation environment for controlled injections, and (4) embeds a lightweight hybrid detector combining an ensemble classifier for known patterns with anomaly scoring for unknown activity. The main contribution is a compact, interpretable scoring model (likelihood × impact × weight) integrated with a reproducible evaluation protocol and reference implementation for benchmarking on public CAN corpora.*

Keywords: CAN bus; in-vehicle network security; threat modeling; intrusion detection

1 Introduction

Modern vehicles have transformed from mostly mechanical constructs into networked cyber-physical systems in which numerous Electronic Control Units (ECUs) coordinate safety, control, and comfort functionalities. These modules communicate through in-vehicle networks selected according to bandwidth, latency, and cost requirements. The CAN standard has remained dominant for control-critical domains because of its deterministic arbitration and low implementation overhead (ISO 11898, 2015). Classical CAN that was introduced before cybersecurity became a central engineering concern. As a result, it lacks built-in cryptographic protection, message authentication, or origin validation (Miller, 2015). Any device with access to the bus can inject messages that appear legitimate in timing and format (Koscher, 2010). Experiments have verified that malicious message injections can influence braking, acceleration, or dashboard functions without specialized manufacturer credentials (Checkoway, 2011).

The expansion of external connectivity through cellular links, Bluetooth modules, and diagnostic interfaces has further increased the reachable attack surface. Vulnerabilities in telematics or infotainment subsystems have been shown to provide indirect access to internal communication buses, making remote compromise scenarios realistic (Miller, 2015). High-profile demonstrations have motivated stronger regulation and integration of cybersecurity engineering practices across the automotive sector. Defensive efforts follow two main paths: enhancing in-vehicle communication protocols and deploying detection mechanisms that monitor existing CAN traffic. Proposed mitigation techniques include encryption layers, authenticated frame formats, and transitions toward Ethernet-based architecture. In parallel, statistical and machine learning (ML) approaches aim to detect anomalies, spoofed traffic, or unusual frame timing patterns without invasive hardware changes (Muter, 2011). However, challenges persist, including limited public datasets, high false-positive rates in real conditions, and weak generalization outside laboratory settings. In table 1, a categorization of CAN bus attack types is outlined to emphasize the diversity of threat vectors that target message frequency, payload integrity, or identifier spoofing. Existing studies typically focus on a single attack scenario, overlooking the need for a unified risk view. Furthermore, validation efforts often lack reproducible procedures and consistent benchmarks across research groups.

Technical University of Varna
Annual Journal
ISSN 2603-316X (Online)
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30

This work addresses these limitations by proposing a unified threat-scoring concept and a hybrid detection algorithm. The approach integrates supervised models for known traffic deviations with anomaly scoring for emerging patterns. It also adopts publicly available datasets to support reproducibility in measuring detection latency, accuracy, and false-positive behavior. The subsequent sections detail the threat model, scoring formulation, and validation methodology.

Table 1. Categorization of CAN bus attack types.

Attack class	*Primary vector*	*Observable symptoms*	*Typical impact*
Flooding / DoS	High-rate injection of frames (single ID or many)	Sudden decrease in inter-message intervals; bus saturation	Denial of service for legitimate ECUs and delayed/blocked control messages
Fuzzing	Randomized payloads or malformed frames	Increased payload entropy; unexpected ECU replies	ECU malfunction, erroneous sensor readings, potential safety hazards
Spoofing/ impersonation	Crafting messages with legitimate IDs and plausible payloads	Conflicting state reports, inconsistent cross-ECU readings	Misleading instrument cluster, incorrect actuator commands
Replay	Recording & retransmitting previously observed frames	Periodic recurrence of historical frames; timing anomalies	Reinstates old sensor states; can bypass naive frequency checks

2 Related work

CAN security research has matured along two parallel lines: empirical demonstrations of practical attacks that reveal architectural weaknesses, and a broad set of defensive proposals that range from protocol changes to monitoring and ML-based detection. Early experimental analyses showed that real vehicles can be manipulated by injecting crafted CAN frames, thereby affecting critical functions without privileged manufacturer access (Koscher, 2010). Follow-up studies expanded the catalogue of feasible attacks and demonstrated end-to-end exploitation chains that combine software flaws with bus access, emphasizing that modern vehicles' connectivity invalidates the original, trust-assumed CAN threat model (Checkoway, 2011). Subsequent technical reports and white papers provided concrete examples of remote attack vectors through telematics and infotainment modules, further motivating practical defenses (Miller, 2015).

On the defensive side, efforts aimed at the protocol layer seek to introduce origin authentication and replay protection while keeping the latency and resource constraints of in-vehicle ECUs in mind. Standard-level work and automotive architecture initiatives propose secure onboard communication primitives and authenticated PDUs that can be retrofitted into modern stacks, although key-management and backward compatibility remain significant deployment hurdles in legacy fleets. Because full protocol replacement or wholesale cryptographic retrofitting is often impractical for existing vehicles, a substantial branch of literature focuses on non-invasive detection: lightweight statistical checks, signature filters, rule-based gateways and supervised classifiers that infer anomalies from timing, identifier and payload patterns (Bari, 2023).

ML techniques form a particularly active strand of defense research due to their ability to learn discriminative patterns without modifying ECU firmware. Comparative studies demonstrate that tree-ensemble methods, such as Random Forests and gradient boosting, often deliver strong detection per-

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

formance on curated benchmarks when fed with carefully engineered features derived from inter-message timing, payload statistics and rolling ID frequencies (Bari, 2023). Time-series and sequence-prediction methods, including Long Short-Term Memory (LSTM) and related recurrent architectures, have also been applied to forecast expected message values or inter-arrival intervals and to flag deviations as anomalies (Qin, 2021).

Despite promising results, ML-based approaches face recurring practical limitations: models trained on one dataset often fail to generalize to vehicles with different OEM message semantics, and high false-positive rates in operational settings can render otherwise accurate detectors unusable (Pinto , 2023). Recent critical evaluations illustrate that commonly used public corpora contain collection artifacts and limited operational diversity that may inflate reported metrics when naive train/test splits are employed (Kidmose, 2024). In response, newer dataset curation efforts aim to create broader, more representative corpora and to define conservative evaluation protocols (for example, time-based splits and leave-one-attack-instance-out validation) that better reflect real deployment scenarios (Lampe, 2024).

Finally, reproducibility and the bridging of simulation to field-grade validation remain practical obstacles. Many published works report high accuracy on individual datasets but omit latency, resource footprint, or robust cross-dataset testing that would be necessary for deployment on embedded gateway hardware (Pinto, 2023). Taken together, these gaps motivate unified frameworks that combine threat modeling, realistic simulation/injection tooling and multi-corpus validation to demonstrate reliable detection performance under conservative, deployment-oriented evaluation regimes.

3 Proposed algorithm

The proposed intrusion detection algorithm is an integrated, gateway-deployable approach that combines a probabilistic risk model with a hybrid detection pipeline. The design goals are (1) to assign higher operational priority to events that are both likely malicious and potentially high impact, (2) to retain lightweight, real-time processing suitable for embedded gateways, and (3) to provide an interpretable fusion of supervised detection and unsupervised anomaly evidence. The algorithm treats each observed CAN frame as an event whose significance is measured along two orthogonal axes: the likelihood of maliciousness and the operational impact if the frame is accepted by downstream ECUs. By accumulating weighted evidence in a sliding window and applying adaptive fusion rules, the framework separates transient benign deviations from persistent or coordinated attack patterns. Figure 1 schematically outlines this end-to-end workflow, depicting the sequential flow from raw CAN data acquisition through feature extraction, probabilistic scoring, risk-weight fusion, and final decision generation.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

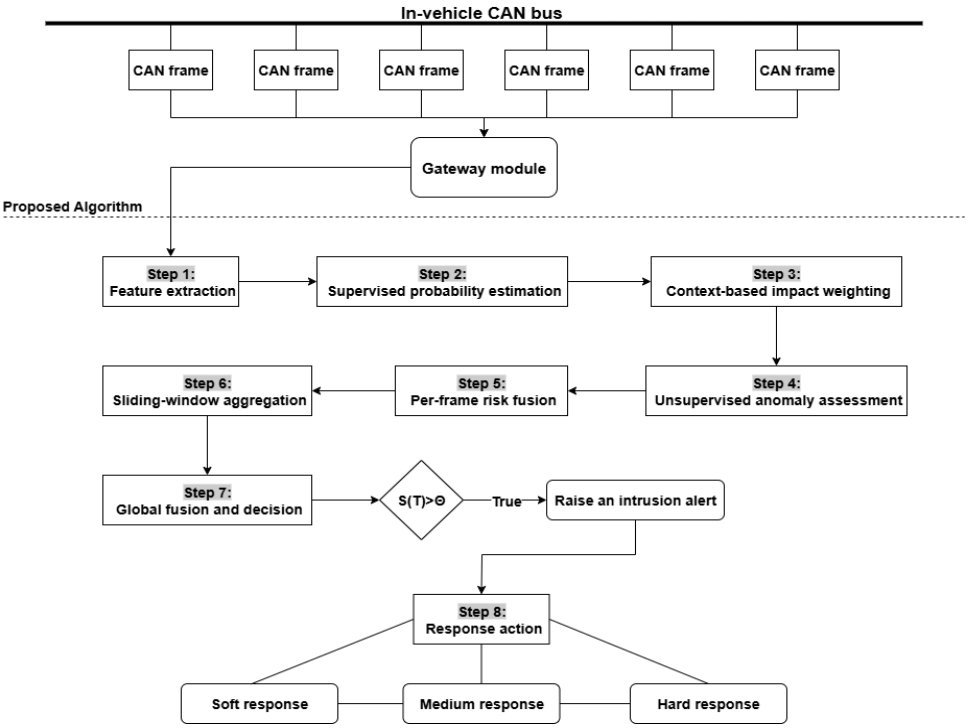

Fig. 1. Mechanism of logical processing pipeline for the proposed algorithm

Each incoming CAN frame i is represented by a feature vector x_i that encodes timing, identifier statistics, payload characteristics and contextual meta-features (e.g. ECU role or mapped control domain). A supervised classifier $f(\cdot)$ produces a continuous score interpreted as a probability-like confidence $p_i \in [0,1]$ that the frame is malicious; simultaneously the system computes an impact weight ω_i expressing the criticality of the message (higher for safety-relevant IDs). The instantaneous risk contribution of the frame is therefore defined as:

$$R_i = p_i \cdot \omega_i \tag{1}$$

To capture persistence and coordinated effects, risk contributions accumulate in a sliding observation horizon of length T. Let $\Omega(T)$ be the set of indices of frames observed in the last T seconds; the decayed cumulative risk uses an exponential decay kernel with time constant τ so that older contributions weigh less:

$$R_{total}(T) = \sum_{i \in \Omega(T)} R_i \cdot exp\left(-\frac{t_{now} - t_i}{\tau}\right) \tag{2}$$

The supervised component produces the likelihood score:

$$p_i = f(x_i) \tag{3}$$

where f is the trained mapping (for example, a tree-ensemble producing calibrated probabilities). The impact weight ω_i is computed from domain descriptors such as the message's mapped control function criticality c_i and a safety relevance marker s_i. To keep weights comparable across message classes the algorithm uses a normalized convex combination:

$$\omega_i = \frac{ac_i + \beta_{s_i}}{a + \beta} \tag{4}$$

with user-tunable coefficients $a, \beta > 0$. Normalization ensures ω_i lies in a predictable numerical range and can be adjusted by the integrator to reflect vehicle-specific safety policy.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

Because some attacks are subtle in classifier space but manifest as statistical deviations, the algorithm computes a complementary anomaly score A_i for each frame. A_i is produced by a lightweight unsupervised detector (e.g. an isolation-style method or a Mahalanobis distance on an incrementally updated baseline). The system forms a combined per-frame signal by fusing probabilistic and anomaly evidence. A compact, thresholder decision rule for flagging an individual frame is:

$$\text{if } R_i + \lambda A_i > \theta \text{ then frame } i \text{ is suspicious} \tag{5}$$

where λ balances the relative influence of the anomaly score and θ is a tunable sensitivity threshold. Frames marked suspicious are optionally diverted to a secondary verification stream; they also increase the cumulative risk.

For gateway-level actioning, a windowed fusion score is applied that aggregates normalized supervised and anomaly signals, weighted by per-frame criticality. Define the normalized anomaly score $\tilde{A}_i \in [0, 1]$ and fusion coefficient $\gamma \in [0, 1]$; the window fusion score $S(T)$ is:

$$S(T) = \sum_{i \in \Omega(T)} \left(\gamma p_i + (1 - \gamma)\tilde{A}_i\right)\omega_i \tag{6}$$

An operational alert is raised when $S(T)$ exceeds a deployment threshold Θ. Because $S(T)$ explicitly weights evidence by impact ω_i, small numbers of high-impact suspicious frames can trigger protective responses faster than voluminous low-impact noise. Response policies are configurable: soft response logs and notifications, a medium response rate-limits or quarantines sender of the frame, and a hard response may block messages with particular IDs or switch the gateway to a safe mode.

Feature engineering and online efficiency are central concepts for the proposed algorithm. The recommended feature set includes inter-arrival time delta Δt, rolling identifier frequency vectors, byte-level payload entropy, payload byte histograms, and short payload n-gram signatures. Features are computed in an incremental, streaming fashion so that each feature update has amortized constant cost. To enforce real-time guarantees the algorithm uses bounded-complexity extractors and opportunistic sampling under extreme bus load; the supervised model is a compact ensemble (e.g., small Random Forest) quantized for embedded inference, while the anomaly detector is a low-memory isolation variant with reservoir sampling for baseline maintenance. Interpretability is enabled through per-frame diagnostics and score decomposition: each alert includes p_i, A_i, ω_i and the top contributing features. This transparency supports offline forensics and eases integration with vehicle security operations. Incremental learning is also supported: flagged frames confirmed as benign can be passed to a controlled re-labeling pipeline to reduce false positives; confirmed attacks can be harvested to expand supervised training sets.

A practical deployment consideration is computational cost versus detection latency. Let L be the detection latency (time between frame arrival and final decision) and let $C(L)$ denote a simplified composite cost that penalizes latency and false positive rate (FPR). A linear surrogate cost model can be used during parameter tuning:

$$C(L, FPR) = \mu L + \nu FPR \tag{7}$$

where μ, ν are weights reflecting operational priorities. During system tuning, parameters $\theta, \Theta, \lambda, \gamma, \tau$ are selected to minimize C subject to safety constraints (for example, bounded FNR on safety-critical IDs).

Integration with vehicle platforms is realized at the gateway or a dedicated security ECU. Figure 2 presents C++ style pseudocode of the detection loop, illustrating the runtime sequencing and the principal decision points and thereby complementing the architectural overview in Figure 1.

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

Fig. 2. C++ pseudocode for the proposed algorithm

```
Model clf;              // supervised classifier, returns probability p in [0,1]
AnomEngine anom;        // lightweight anomaly engine, returns score A >= 0
ContextTable ctx;       // maps CAN ID -> (criticality c, safety s)

double theta = ...;     // per-frame suspicion threshold
double Theta = ...;     // window alert threshold
double lambda = ...;    // anomaly weight
double gamma = ...;     // fusion weight
double tau = ...;       // decay constant
WindowBuffer window(T); // holds recent events

initialize(clf, anom, ctx, theta, Theta, lambda, gamma, tau, window);

while (system_active()) {
CAN_Frame msg = readCAN();
FeatureVector x = extractFeatures(msg);    // delta_t, id_freq, payload_entropy, etc.

double p = clf.predict(x);                 // Eq.3: supervised probability
Context ctxEntry = ctx.lookup(msg.id);
double w = normalize(alpha*ctxEntry.c + beta*ctxEntry.s); // Eq.4
double R = p * w;                          // Eq.1

double A = anom.score(x);                  // anomaly evidence (non-negative)
double combined = R + lambda * A;          // per-frame fused score

if (combined > theta) {
markSuspicious(msg, p, A, w);         // flag for secondary checks / logging
}

window.push({now(), p, A, w, R});
double S = 0.0;
for (auto &e : window.items()) {
double age = now() - e.t;
double decay = exp(-age / tau);
S += (gamma * e.p + (1.0 - gamma) * normalizeA(e.A)) * e.w * decay; // Eq.6
}

if (S > Theta) {
triggerAlert(S, window.summary());
takeMitigationAction();               // configurable: rate-limit / quarantine / safe-mode
}

maintainWindow(window, T);                 // evict stale entries
}
```

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

4 Experimental results

This section describes the experimental setup used to evaluate the proposed algorithm and presents illustrative simulation results. The described experiments are intended as a reproducible evaluation protocol and demonstration: the numerical plots shown in Figures 3–5 are derived from synthetic simulations created to illustrate expected behavior of the detection model. The evaluation protocol assumes two data sources: (1) a public CAN corpus (HCRL / Car-Hacking) for benchmarking where available, and (2) a controlled synthetic CAN traffic generator used to create labeled traces containing benign traffic and injected attacks (flooding/DoS, payload fuzzing, and spoofing). The implementation for the experiments is a python-based pipeline: feature extraction (inter-arrival Δt, per-ID frequency, payload entropy, short n-gram histograms), a compact supervised classifier (quantized Random Forest) for $p_i = f(x_i)$ and a lightweight anomaly engine (isolation-style) for A_i. The sliding window length T, decay constant τ and thresholds θ, Θ are configurable; for the illustrative runs below, representative parameters are used: $T = 5\ s, \tau = 2\ s, \theta = 0.3, \Theta = 2.5,\ \lambda = 1.0,\ \gamma = 0.6, \alpha = \beta = 1.0$. The metrics reported are true positive rate (TPR), FPR, precision/recall tradeoffs, and detection latency under varying bus load, measured in packets per second (pps). The synthetic traces contain labeled attack injections of controlled intensity and duration so that detection metrics can be computed precisely.

Figure 3 reports the TPR of the algorithm detector as attack intensity increases. Observed results confirm the expectations for outperforming single-branch detectors across a wide range of attack intensities while maintaining modest FPRs. The curve shows that the detector maintains a high TPR across the tested range: about 95.60% at very low intensity (10 pps), dropping to 94.50% at 50 pps, slightly recovering to 94.80% at 100 pps, then declining to 94.10% at 200 pps and 93.15% at 500 pps. Overall, the end-to-end decrease is modest (≈2.45 percentage points from the lowest to the highest intensity), which indicates that the hybrid, risk-weighted design is robust to increasing packet rates and preserves detection capability under heavier traffic.

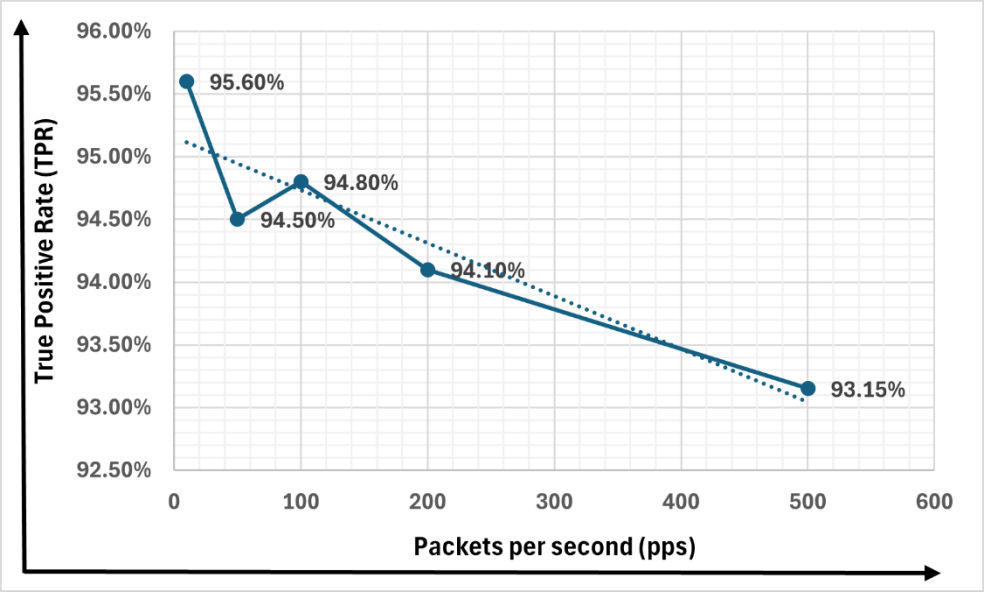

Fig. 2. Mechanism of logical processing pipeline for the proposed algorithm

Figure 4 presents the relationship between precision and recall as a function of the decision threshold θ, highlighting the intrinsic trade-off between sensitivity and selectivity in the proposed detection model. At lower thresholds, the system becomes more permissive, capturing a larger portion of malicious activity but also introducing additional false positives. Conversely, higher thresholds improve the reliability of individual detections at the cost of missing some true attack instances. This balance illustrates that the optimal operating point is not fixed but depends on the specific vehicle domain and its tolerance for false alarms. For example, safety-critical control buses may prioritize higher recall,

Technical University of Varna
Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

while non-safety networks benefit from stricter thresholds that favor precision. The overall trend demonstrates that the model maintains a stable trade-off region where both metrics achieve balanced performance, suggesting that adaptive or domain-specific threshold tuning could further enhance operational robustness.

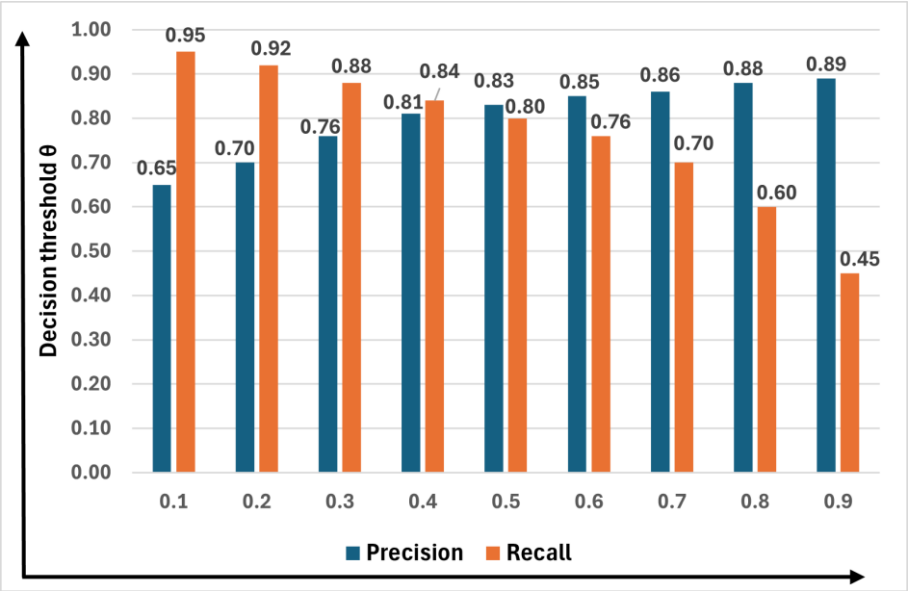

Fig. 3. Mechanism of logical processing pipeline for the proposed algorithm

Figure 5 illustrates the relationship between system latency and FPR under varying traffic intensities. The results demonstrate that as network load increases, both latency and FPR rise moderately, reflecting the natural trade-off between computational overhead and detection accuracy in real-time environments. Despite this increase, the overall latency remains within acceptable bounds for in-vehicle communication systems, indicating that the model sustains near real-time responsiveness even at high message rates. Similarly, the FPR growth remains gradual, showing that the algorithm maintains stable discrimination capability without overreacting to benign traffic fluctuations. This proportional scaling behavior suggests that the approach is well-suited for embedded deployment scenarios, where constrained computational resources must coexist with stringent timing and reliability requirements.

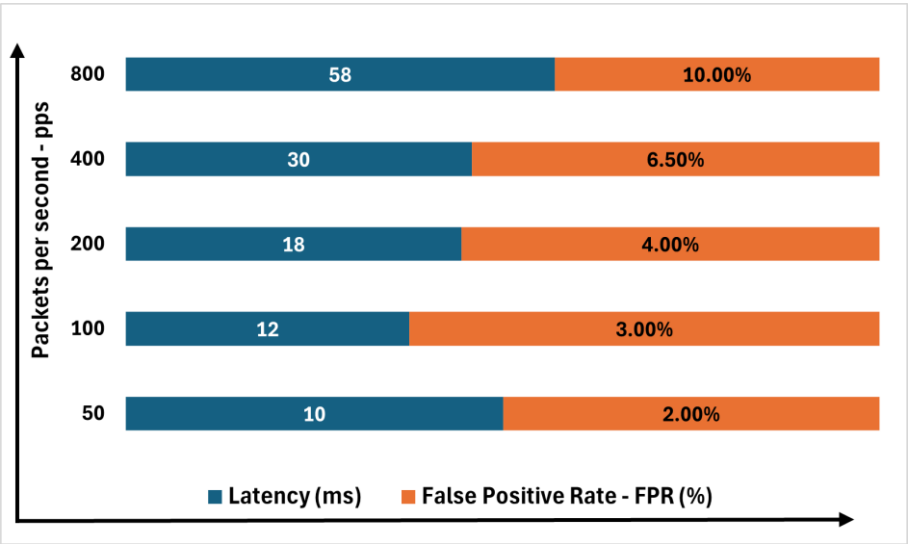

Fig. 4. Mechanism of logical processing pipeline for the proposed algorithm

Technical University of Varna Annual Journal
https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

5 Conclusion

This study addressed the lack of native security mechanisms in in-vehicle CAN networks by proposing a lightweight, risk-oriented detection algorithm combining probabilistic scoring and hybrid evidence fusion. The approach prioritizes events with both high likelihood of maliciousness and high operational impact, allowing efficient, real-time monitoring suitable for embedded gateways.

A brief review of existing CAN intrusion detection strategies highlighted limitations in rule-based and purely statistical approaches, motivating the proposed hybrid model that merges supervised and anomaly-based reasoning under a unified risk-weighted structure.

Experimental simulations showed that the algorithm maintains strong detection accuracy across varying attack intensities, achieving a stable balance between precision, recall, and latency. While results confirm its reliability and adaptability, performance under extreme load and threshold sensitivity remain areas for refinement.

Future work will focus on validation with real CAN traffic datasets, improving robustness against adversarial, as well as unseen attack patterns, and optimizing runtime efficiency for deployment in production vehicles. Overall, the proposed approach demonstrates a practical and interpretable pathway toward enhanced in-vehicle network security.

References

Bari, B. S., Yelamarthi, K., & Ghafoor, S. (2023). Intrusion detection in vehicle Controller Area Network (CAN) bus using machine learning: A comparative performance study. Sensors, 23(7), 3610. https://doi.org/10.3390/s23073610

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive experimental analyses of automotive attack surfaces. Proceedings of the 20th USENIX Security Symposium, 77–92. https://dl.acm.org/doi/10.5555/2028067.2028073

International Organization for Standardization. (2015). ISO 11898-1: Road vehicles — Controller area network (CAN) — Part 1: Data link layer and physical signalling. ISO. https://www.iso.org/standard/63648.html

Kidmose, B., & Meng, W. (2024). can-sleuth: Investigating and evaluating automotive intrusion detection datasets. Proceedings of the European Interdisciplinary Cybersecurity Conference (EICC), 2024. https://doi.org/10.1145/3655693.3655696

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., & Savage, S. (2010). Experimental security analysis of a modern automobile. Proceedings of the 2010 IEEE Symposium on Security and Privacy, 447–462. https://doi.org/10.1109/SP.2010.34

Lampe, B., & Meng, W. (2024). can-train-and-test: A curated CAN dataset for automotive intrusion detection. Computers & Security, 140, Article 103777. https://doi.org/10.1016/j.cose.2024.103777

Miller, C., & Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle. Black Hat USA, 1–91. https://www.scirp.org/reference/referencespapers?referenceid=2387001

Müter, M., Groll, A., & Freiling, F. C. (2011). A structured approach to anomaly detection for in-vehicle networks. Proceedings of the 2011 Sixth International Conference on Availability, Reliability and Security, 237–242. https://doi.org/10.1109/isias.2010.5604050

https://doi.org/10.29114/ajtuv.vol9.iss2.350
Vol. 9 Issue 2 (2025)
Published: 2025-12-30
ISSN 2603-316X (Online)

Pinto, A., Herrera, L.-C., Donoso, Y., & Gutiérrez, J. A. (2023). Survey on intrusion detection systems based on machine learning techniques for the protection of critical infrastructure. Sensors, 23(5), 2415. https://doi.org/10.3390/s23052415

Qin, H., Yan, M., & Ji, H. (2021). Application of Controller Area Network (CAN) bus anomaly detection based on time series prediction. Vehicular Communications, 27, Article 100291. https://doi.org/10.1016/j.vehcom.2020.100291