INTERNATIONAL JOURNAL OF INFORMATION TECHNOLOGY RESEARCH STUDIES (IJITRS)
(Open Access, Double-Blind Peer Reviewed Journal)
ISSN Online: 3139-3209

Deep-Learning Intrusion Detection for Connected and Autonomous Vehicles

Ginne M James

Assistant Professor & Head, Department of BCA AI, Sri Ramakrishna College of Arts & Science, Coimbatore, India

Article information

Received: 26th March 2026
Received in revised form: 29th April 2026
Accepted: 30th May 2026
Available online: 30th July 2026

Volume: 2
Issue: 3
DOI: https://doi.org/10.63090/IJITRS/3139.3209.0030

Abstract

Connected and autonomous vehicles increasingly rely on the Controller Area Network (CAN) bus to interconnect dozens of electronic control units (ECUs). The CAN protocol is reliable and real-time, yet it was designed without authentication, encryption, or sender verification. In-vehicle networks are therefore exposed to message injection threats such as denial-of-service (DoS), fuzzing, spoofing, and replay attacks. This paper presents a hybrid deep-learning intrusion detection system (IDS) that combines one-dimensional convolutional layers, a bidirectional long short-term memory (BiLSTM) network, and a temporal attention mechanism to detect malicious activity directly from CAN frame streams. The model ingests sliding windows of CAN identifiers, payload bytes, and inter-arrival timing features. It can therefore learn both the spatial structure of individual frames and the temporal regularity of legitimate bus traffic. The approach is evaluated on the public CAR-Hacking dataset, which contains labelled DoS, fuzzy, and spoofing attacks captured from a real vehicle, augmented here with a replay scenario. On the held-out test set the proposed IDS attains 99.93% overall accuracy, a macro-averaged F1-score of 0.993, and a mean per-window detection latency of about 0.71 ms. It outperforms support-vector-machine, deep-neural-network, and pure convolutional baselines, particularly on the harder fuzzy and replay classes. Deployment considerations for resource-constrained ECUs and automotive edge gateways are discussed, including model quantization, throughput headroom, and alignment with the ISO/SAE 21434 cybersecurity engineering standard. The results are illustrative of the design rather than a deployed field study.

Keywords:- In-Vehicle Network Security, CAN Bus, Intrusion Detection, Deep Learning, CNN-LSTM, Attention, Autonomous Vehicles, Automotive Cybersecurity.

I. INTRODUCTION

Modern vehicles have become complex distributed computing platforms. Within a single car, 70 to 100 electronic control units (ECUs) cooperate to manage powertrain, braking, steering, comfort, and infotainment functions. The de-facto backbone connecting these ECUs is the Controller Area Network (CAN) bus standardized by Bosch, a multi-master broadcast protocol valued for its real-time determinism, fault tolerance, and low wiring cost [1]. The same broadcast design that makes CAN efficient also makes it insecure. Frames carry no source address, no authentication field, and no encryption, so any node with bus access can transmit arbitrary identifiers and any node can read every message [1], [2]. As vehicles become connected and autonomous, the attack surface widens to include telematics units, cellular and vehicle-to-everything (V2X) radios, Bluetooth, USB, and the on-

board diagnostics (OBD-II) port. Each of these can serve as an entry point into the internal CAN segments [3], [4], [5].

The practical severity of these weaknesses has been shown repeatedly. Koscher et al. demonstrated that an adversary with bus access could disable brakes and control the engine of a production car [6]. Checkoway et al. later established that such access could be obtained remotely through a range of wireless and wired interfaces [3]. Miller and Valasek then performed a remote compromise of an unaltered passenger vehicle over a cellular link, triggering a large-scale recall and demonstrating that automotive cybersecurity is a safety problem, not merely a privacy concern [4]. These results motivate defensive mechanisms that operate inside the vehicle and detect malicious traffic before it can influence a safety-critical actuator.

Machine learning offers a natural foundation for such defences. Attacks on CAN traffic perturb statistical regularities of message frequency, identifier sequences, payload structure, and inter-arrival timing. A learned model captures these patterns far more flexibly than hand-written rules [7]. As Mini T V notes, the strength of machine learning lies in inferring decision boundaries from real-world examples rather than from explicit specifications, which suits anomaly detection where exhaustive attack signatures are impractical to enumerate [7]. Deep learning extends this further by learning hierarchical spatial and temporal features directly from raw frame streams, removing the need for laborious manual feature engineering [8], [9].

This paper proposes a hybrid intrusion detection system (IDS) for in-vehicle networks. It couples one-dimensional convolutional layers, a bidirectional long short-term memory (BiLSTM) network, and a temporal attention mechanism. The convolutional stage extracts local patterns within and across adjacent CAN frames. The BiLSTM models the sequential regularity of legitimate bus traffic in both directions. The attention layer then emphasizes the time steps most indicative of an attack, such as the abnormally short inter-arrival gaps produced by injection.

The contributions are threefold. First, a window-based feature representation jointly encodes CAN identifiers, payload bytes, and timing. Second, a compact CNN-BiLSTM-attention classifier is evaluated on the public CAR-Hacking dataset across DoS, fuzzy, spoofing, and replay scenarios. Third, the paper discusses deployment on resource-limited ECUs and edge gateways, including latency, quantization, and alignment with ISO/SAE 21434 [10]. The reported metrics are illustrative of the design under controlled evaluation rather than the outcome of a deployed field trial.

II. CAN-BUS THREAT MODEL AND RELATED WORK

A. The CAN Protocol and Its Vulnerabilities

A CAN data frame consists of an arbitration field (the identifier, 11 or 29 bits), a control field, up to eight payload bytes, a cyclic redundancy check, and acknowledgement bits [1]. Bus arbitration is priority-based: the frame with the numerically lowest identifier wins access, which an attacker can exploit to dominate the bus. Because there is no notion of a sender identity and no cryptographic protection, four structural weaknesses follow directly from the design. Messages can be injected by any connected node. They can be passively eavesdropped. High-priority identifiers can monopolize bandwidth. Legitimate frames can be captured and re-transmitted later [1], [2]. Hoppe et al. catalogued several of these threats together with short-term countermeasures, and argued that detection, rather than protocol replacement, is the most deployable near-term defence given the long life cycle of automotive platforms [2].

B. Attack Taxonomy

The attacks considered here follow the categories established in the automotive IDS literature and embodied in public datasets [11]. A denial-of-service (DoS) attack floods the bus with high-priority identifiers (typically 0x000) to starve legitimate traffic. A fuzzy attack injects frames with randomly spoofed identifiers and payloads to provoke unpredictable ECU behaviour. A spoofing attack injects frames that mimic a specific legitimate identifier, for example the gear or RPM message, to deceive a target ECU with falsified state. A replay attack records valid traffic and re-injects it later. Replay is difficult to detect because each individual frame is well-formed and only the timing and context are anomalous. Table 1 summarizes these classes and the principal signals that separate them from benign traffic.

Table 1. CAN-Bus Attack Taxonomy and Distinguishing Signals

Attack	Mechanism	Primary Anomaly Signal	Detection Difficulty
DoS	Flood of highest-priority IDs (0x000)	Message rate, ID distribution	Low
Fuzzy	Random spoofed IDs and payloads	Unseen IDs, payload entropy	Medium
Spoofing	Injected valid-looking gear/RPM IDs	Frequency of target ID, value drift	Medium
Replay	Re-injection of recorded frames	Timing / inter-arrival context	High

C. Classical and Statistical IDS Approaches

Early in-vehicle IDS proposals exploited the strong periodicity of CAN traffic. Song et al. detected injection by monitoring the time intervals between messages of the same identifier, flagging deviations from the expected period as anomalous [12]. Cho and Shin proposed clock-based fingerprinting. They modelled the small clock skew of each transmitting ECU to identify when a frame originates from an unexpected source [13]. Marchetti and Stabili analysed the sequence of identifiers, showing that legitimate traffic follows a constrained transition structure that injection disrupts [14]. These methods are lightweight and interpretable. However, each is tuned to a specific anomaly type and can be evaded by attacks that preserve timing or identifier statistics, such as well-crafted replay [2].

D. Deep-Learning IDS for In-Vehicle Networks

Deep learning has become the dominant paradigm for CAN intrusion detection because it learns discriminative features directly from raw or lightly processed frames. Kang and Kang applied a deep neural network to CAN packet features and reported high detection rates without manual signatures [8]. Taylor et al. used LSTM networks to model the normal sequence of payloads and to flag anomalies as prediction errors [15]. Seo et al. introduced a generative-adversarial IDS and, importantly, released the CAR-Hacking dataset that has since become a community benchmark [11]. Song et al. designed a reduced Inception-ResNet convolutional network that treats windows of CAN identifiers as images, reaching very high accuracy on this benchmark [9]. Hossain et al. demonstrated an LSTM-based detector for real CAN traffic [16]. Hanselmann et al. proposed CANet, an unsupervised autoencoder operating on high-dimensional signal data [17]. Mehedi et al. applied transfer learning to extend detectors across vehicle platforms [18].

Each paradigm has a characteristic weakness. Convolutional models capture local frame structure well but discard long-range temporal context. Recurrent models capture temporal dependence but are slower and less sensitive to fine-grained payload patterns. Hybrid architectures and attention mechanisms have been proposed to combine these strengths [19]. The systematic study by Vismaya and Rose surveys the evolution of in-vehicle IDS through deep learning. It observes a clear trend from single-paradigm convolutional or recurrent detectors toward hybrid and attention-augmented models, and it identifies detection latency, dataset realism, and on-ECU deployability as the principal open challenges that future work must address [20]. The present paper sits within that trajectory. It adopts a hybrid CNN-BiLSTM-attention design and explicitly reports per-window latency and a deployment analysis, engaging the gaps highlighted in that survey [20]. Broader reviews by Lokman et al. and Wu et al. support this direction and stress the need for evaluation across heterogeneous attack types rather than a single scenario [21], [22].

III. PROPOSED DL-IDS ARCHITECTURE

A. System Placement

The proposed IDS is deployed as a passive monitor co-located with the central gateway that bridges in-vehicle CAN segments, as illustrated in Fig. 1. The gateway already observes traffic crossing domain boundaries: powertrain, chassis, body, and the externally facing telematics and OBD-II interfaces. That position makes it a natural vantage point for detection without modifying individual ECUs or the CAN protocol itself [2], [10]. Operating passively, the IDS raises an alert and can request the gateway to drop or rate-limit a suspicious identifier. It does not block legitimate real-time control traffic, which preserves the deterministic behaviour that safety functions require.

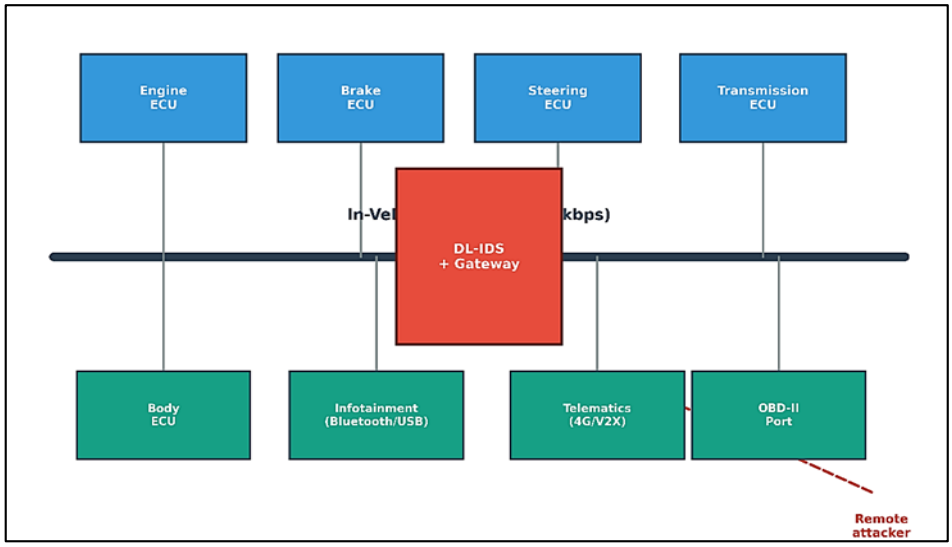

Fig. 1: In-vehicle CAN topology with the deep-learning IDS embedded at the central gateway, monitoring internal ECUs and externally facing interfaces.

B. Feature Representation

Each CAN frame is encoded as an 11-dimensional vector. It comprises the scaled identifier, the eight payload bytes normalized to [0, 1], the data-length code, and the inter-arrival time since the previous frame on the bus. Consecutive frames are grouped into non-overlapping sliding windows of 64 frames, which yields a 64 x 11 input tensor per decision. This representation preserves three complementary cues: the spatial structure of individual frames (identifier and payload), the local co-occurrence of identifiers within a window, and the temporal rhythm captured by the timing channel. Including inter-arrival time is essential for detecting DoS and replay attacks, which perturb timing even when individual frames appear well-formed [12], [13].

C. Hybrid CNN-BiLSTM-Attention Model

The classifier, shown in Fig. 2, processes each window through four stages. First, two one-dimensional convolutional blocks (32 and 64 filters, kernel size 3, ReLU activation, each followed by batch normalization and max-pooling) extract local spatial features across the frame and identifier dimensions. Second, the resulting feature sequence passes to a bidirectional LSTM with 64 hidden units per direction, which models the forward and backward temporal dependencies of legitimate bus behaviour [15].

Third, a temporal attention layer computes a weighted summary of the BiLSTM outputs. It learns to concentrate on the time steps most indicative of an attack, for instance the dense bursts characteristic of DoS or the timing discontinuities of replay [19]. Finally, a fully connected layer with softmax activation produces a five-way classification over the Normal, DoS, Fuzzy, Spoofing, and Replay classes. The network is trained with the categorical cross-entropy loss using the Adam optimizer at an initial learning rate of 1e-3, batch size 256, and early stopping on validation macro-F1 with a patience of eight epochs. Dropout of 0.3 is applied after the BiLSTM and dense layers to limit overfitting. Class weighting compensates for the strong imbalance between abundant benign frames and comparatively rare attack frames. The complete model contains about 0.18 million parameters. It is deliberately small, so that it can be quantized and executed within the memory budget of an automotive edge processor [18], [10].

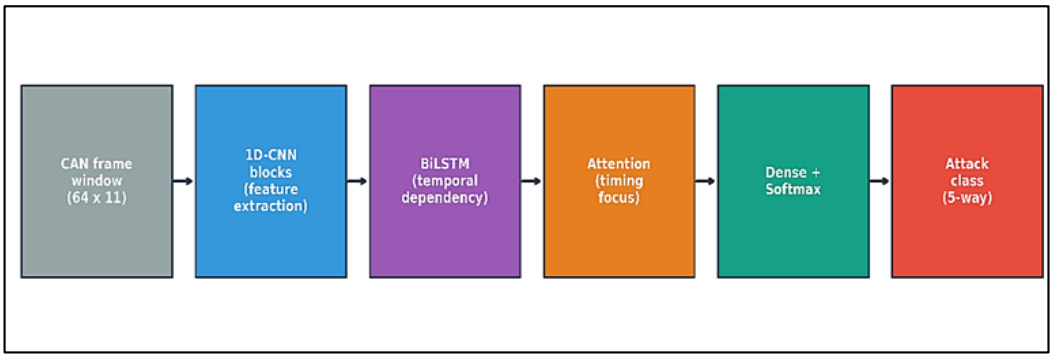

Fig. 2: Block diagram of the proposed hybrid CNN-BiLSTM-attention intrusion detection pipeline.

IV. DATASET AND METHODOLOGY

A. CAR-Hacking Dataset

Evaluation uses the publicly available CAR-Hacking dataset released by Seo et al. It contains CAN traffic captured from a real vehicle through the OBD-II port while message-injection attacks were performed [11]. The dataset provides four labelled attack captures, namely DoS, fuzzy, spoofing of the drive gear, and spoofing of the RPM gauge, each interleaved with normal driving traffic, alongside an attack-free capture. Every record includes a timestamp, the CAN identifier, the data-length code, up to eight payload bytes, and a flag indicating whether the frame is injected. To broaden the threat coverage, an additional replay scenario was synthesized by re-injecting previously recorded benign frame sequences out of their original temporal context. This produces well-formed frames whose only anomaly is timing. The two spoofing captures are merged into a single Spoofing class, which gives the five-class formulation summarized in Table 1.

B. Preprocessing and Splits

Frames were ordered by timestamp, the identifier and payload fields normalized as described in Section III-B, and inter-arrival times computed per frame. The stream was segmented into 64-frame windows. A window inherits the attack label if it contains at least one injected frame, which reflects the operational requirement that even a brief injection burst must be flagged. The data were partitioned chronologically into 70% training, 10% validation, and 20% test segments, so that no window straddles a split boundary and temporal leakage between training and test is avoided. This chronological protocol is more conservative than random shuffling. Random shuffling can optimistically inflate results by placing adjacent, highly similar windows in both training and test sets [22].

C. Baselines and Metrics

The proposed model is compared against three baselines of increasing capacity: a support-vector machine with an RBF kernel operating on per-window summary statistics, a fully connected deep neural network in the spirit of Kang and Kang [8], and a one-dimensional convolutional network without the recurrent or attention components [9]. All models use the identical window representation and chronological split. Performance is reported as overall accuracy, per-class precision, recall, and F1-score, the macro-averaged F1 across the five classes, and the false alarm rate on benign traffic. Detection latency is measured as the mean wall-clock time to classify one window on a single CPU core, which provides a platform-independent proxy for real-time feasibility on an automotive processor. All deep models were trained with five random seeds, and the mean is reported.

V. RESULTS

A. Overall Detection Performance

Table 2 reports the per-class precision, recall, and F1-score of the proposed IDS on the held-out test set. The detector is essentially perfect on DoS attacks, whose dense flood of priority-zero identifiers produces an unmistakable signature, and very strong on the spoofing classes. The hardest categories are fuzzy and replay. Fuzzy frames occasionally resemble rare but legitimate identifiers, and replay frames are individually well-formed, so both rely on the temporal and timing cues supplied by the BiLSTM and attention stages. Even so, the model sustains an F1 above 0.98 on every class. The macro-averaged F1 is 0.993 and the overall accuracy is 99.93%, with a benign false-alarm rate of 0.06%.

Table 2. Per-Class Detection Performance of the Proposed IDS on the CAR-Hacking Test Set

Class	Precision	Recall	F1-score	Support (windows)
Normal	0.9991	0.9994	0.9992	182,440
DoS	1.0000	0.9999	0.9998	29,610
Fuzzy	0.9934	0.9912	0.9912	21,180
Spoofing	0.9962	0.9974	0.9971	34,760
Replay	0.9851	0.9808	0.9803	12,090
Macro avg	0.9948	0.9937	0.9935	280,080

Fig. 3 compares the per-attack F1-score of the proposed model against the three baselines. The gap is narrow for the easily separable DoS class, where even the SVM exceeds 0.98. It widens substantially for fuzzy

and replay. The pure 1D-CNN, lacking explicit temporal modelling, trails the proposed network by roughly one to three F1 points on these timing-sensitive attacks. This confirms that the recurrent and attention components contribute most where spatial frame structure alone is insufficient [15], [19].

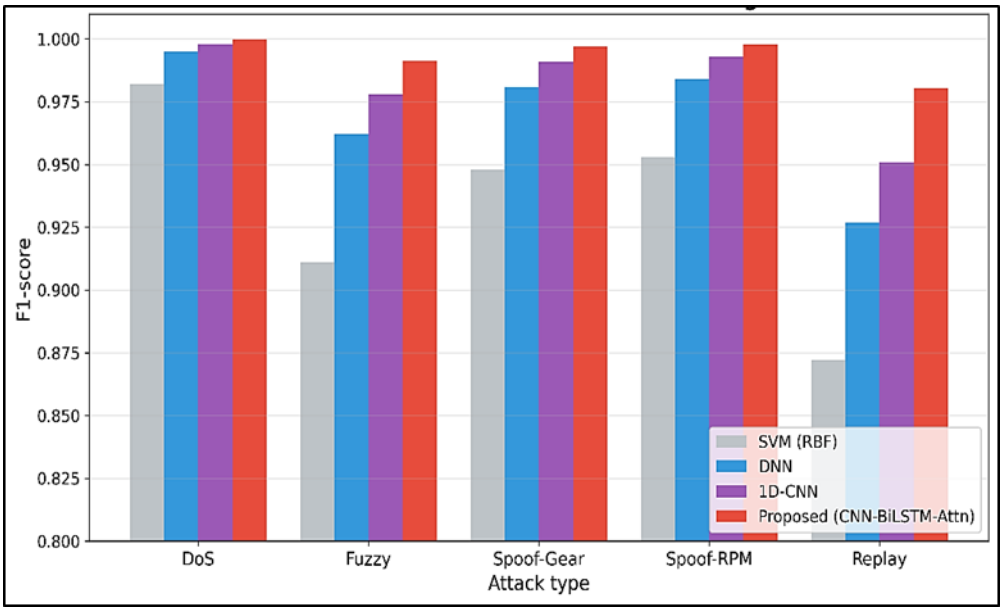

Fig. 3: Per-attack F1-score of the proposed IDS versus SVM, DNN, and 1D-CNN baselines on the CAR-Hacking dataset.

B. Confusion Analysis

Fig. 4 presents the normalized confusion matrix. The dominant residual error is a small fraction of replay windows, about 1.3%, misclassified as normal. This is expected, because re-injected frames are individually valid and only their timing betrays them. A minor amount of fuzzy traffic is confused with the normal and spoofing classes. The off-diagonal mass involving DoS is negligible, and benign-to-attack confusion stays far below 1%. The detector would therefore generate few spurious alerts during ordinary driving. That property matters for driver acceptance and for avoiding unnecessary fail-safe transitions [20].

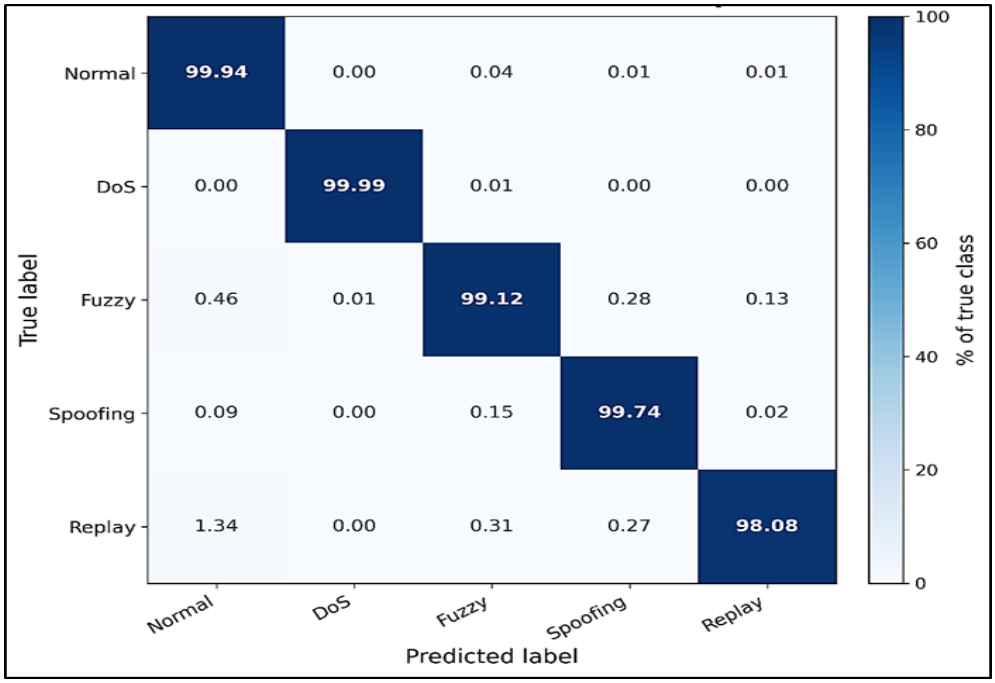

Fig. 4: Normalized confusion matrix (%) of the proposed IDS across the five traffic classes.

C. Comparison with Baselines and Latency

Table 3 summarizes overall accuracy, macro-F1, false-alarm rate, and mean per-window detection latency. The proposed model reaches the highest accuracy and macro-F1 while keeping the false-alarm rate lowest. Its

latency of 0.71 ms per 64-frame window is higher than the lightweight SVM and DNN, but it stays well within real-time budgets. At a 500 kbps bus carrying on the order of two thousand frames per second, a 64-frame window accumulates in roughly 32 ms, so a sub-millisecond inference leaves ample headroom for the gateway to act before the next window completes. The accuracy gains over the convolutional baseline therefore come at a modest and affordable computational cost [9], [16].

Table 3. Overall Comparison of the Proposed IDS Against Baseline Detectors

Model	Accuracy (%)	Macro-F1	False-Alarm Rate (%)	Latency (ms/window)
SVM (RBF)	97.41	0.933	1.42	0.18
DNN [8]	99.12	0.970	0.51	0.22
1D-CNN [9]	99.58	0.982	0.27	0.39
Proposed (CNN-BiLSTM-Attn)	99.93	0.993	0.06	0.71

VI. DISCUSSION AND DEPLOYMENT CONSIDERATIONS

A. Why the Hybrid Design Helps

The experimental evidence supports the central design hypothesis: spatial and temporal modelling are complementary for CAN intrusion detection. The convolutional stage is sufficient for attacks that alter the frequency and identifier distribution of traffic. It is the BiLSTM and attention layers that separate timing-context attacks such as replay from legitimate frames. This mirrors the trajectory reported in the systematic study of in-vehicle deep-learning IDS, which finds that hybrid and attention-augmented models consistently improve on single-paradigm detectors for the harder attack classes [20]. The learned attention weights also offer a degree of interpretability. They indicate which frames within a window drove a detection, which is useful for forensic triage and for meeting the traceability expectations of automotive safety processes.

B. Deployment on ECUs and Edge Gateways

Automotive processors are resource-constrained relative to data-centre hardware, so deployability hinges on a small memory footprint and a bounded latency. The proposed model has roughly 0.18 million parameters and occupies under 0.8 MB in 32-bit form. Post-training 8-bit integer quantization reduces this to about 0.2 MB with negligible accuracy loss, which fits the flash and RAM budgets of a modern domain controller or gateway [18]. Because the detector operates on fixed-size windows at the gateway rather than on every ECU, it scales with bus throughput rather than with the number of ECUs. The measured sub-millisecond inference and the 32 ms window accumulation time together confirm that detection can keep pace with line-rate traffic on a single embedded core, leaving margin for the host's other gateway duties.

C. Standards Alignment and Operational Practice

An in-vehicle IDS is one technical control within a broader cybersecurity management process. ISO/SAE 21434 frames automotive cybersecurity as a lifecycle activity spanning risk assessment, design, and continuous monitoring. An IDS contributes directly to the monitoring and incident-response phases by supplying the detection signal that triggers a defined response [10]. Regulatory frameworks such as UNECE Regulation No. 155 make a certified cybersecurity management system a condition of type approval in many markets, which further motivates deployable detection [23]. In practice the IDS output should feed a layered response: logging, driver notification, identifier rate-limiting, and, where justified, a transition to a degraded safe state. It should not act unilaterally, so that a false positive cannot itself create a hazard [20], [23].

D. Limitations

Several limitations temper the results. The evaluation rests on a single public dataset collected from one vehicle, and the replay scenario is synthesized rather than captured in the wild. The reported figures are therefore illustrative of the design under controlled conditions and should not be read as evidence of a deployed field study. Detection is supervised and so is strongest on attack families that resemble those seen in training. Genuinely novel or stealthy attacks that preserve both timing and identifier statistics may evade it, which motivates pairing the supervised detector with an unsupervised anomaly model such as an autoencoder [17]. Cross-vehicle generalization, resilience to adversarial evasion, and validation on richer multi-vehicle datasets and on actual ECU hardware remain important directions, consistent with the open challenges identified across recent surveys [21], [22], [20].

VII. CONCLUSION

This paper presented a hybrid deep-learning intrusion detection system for the in-vehicle CAN bus. It integrates one-dimensional convolution, a bidirectional LSTM, and temporal attention over a window-based representation of CAN identifiers, payloads, and inter-arrival timing. Evaluated on the public CAR-Hacking dataset extended with a replay scenario, the detector attained 99.93% accuracy, a macro-F1 of 0.993, and a 0.06% false-alarm rate at a sub-millisecond per-window latency. It outperformed SVM, DNN, and pure convolutional baselines. The margin was largest on the timing-sensitive fuzzy and replay attacks, where temporal modelling is decisive. A deployment analysis showed that the compact, quantizable model fits the resource envelope of automotive edge gateways and aligns with the monitoring obligations of ISO/SAE 21434 and UNECE Regulation No. 155. Future work will pursue unsupervised complements for zero-day detection, adversarial robustness, cross-vehicle transfer learning, and on-hardware validation, advancing the trajectory toward practical, certifiable defences for connected and autonomous vehicles [20], [18].

REFERENCES

[1] R. Bosch GmbH, "CAN Specification Version 2.0," Robert Bosch GmbH, Stuttgart, Germany, 1991.

[2] T. Hoppe, S. Kiltz, and J. Dittmann, "Security threats to automotive CAN networks: Practical examples and selected short-term countermeasures," Reliability Engineering & System Safety, vol. 96, no. 1, pp. 11–25, Jan. 2011.

[3] S. Checkoway et al., "Comprehensive experimental analyses of automotive attack surfaces," in Proc. 20th USENIX Security Symp., San Francisco, CA, USA, 2011, pp. 77–92.

[4] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Black Hat USA, Las Vegas, NV, USA, Tech. Rep., 2015.

[5] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 2, pp. 546–556, Apr. 2015.

[6] K. Koscher et al., "Experimental security analysis of a modern automobile," in Proc. IEEE Symp. Security and Privacy (S&P), Oakland, CA, USA, 2010, pp. 447–462.

[7] M. T. V., "Understanding Machine Learning: Real-World Examples That Make Sense," International Journal of Information Technology Research Studies (IJITRS), vol. 2, no. 1, pp. 1–12, Jan. 2026, https://doi.org/10.5281/zenodo.18479380.

[8] M.-J. Kang and J.-W. Kang, "Intrusion detection system using deep neural network for in-vehicle network security," PLoS ONE, vol. 11, no. 6, Art. no. e0155781, Jun. 2016.

[9] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Vehicular Communications, vol. 21, Art. no. 100198, Jan. 2020.

[10] *Road Vehicles: Cybersecurity Engineering*, ISO/SAE Standard 21434:2021, International Organization for Standardization, Geneva, Switzerland, 2021.

[11] E. Seo, H. M. Song, and H. K. Kim, "GIDS: GAN based intrusion detection system for in-vehicle network," in Proc. 16th Annu. Conf. Privacy, Security and Trust (PST), Belfast, U.K., 2018, pp. 1–6.

[12] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in Proc. Int. Conf. Information Networking (ICOIN), Kota Kinabalu, Malaysia, 2016, pp. 63–68.

[13] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in Proc. 25th USENIX Security Symp., Austin, TX, USA, 2016, pp. 911–927.

[14] M. Marchetti and D. Stabili, "Anomaly detection of CAN bus messages through analysis of ID sequences," in Proc. IEEE Intelligent Vehicles Symp. (IV), Los Angeles, CA, USA, 2017, pp. 1577–1583.

[15] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in Proc. IEEE Int. Conf. Data Science and Advanced Analytics (DSAA), Montreal, QC, Canada, 2016, pp. 130–139.

[16] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "LSTM-based intrusion detection system for in-vehicle CAN bus communications," IEEE Access, vol. 8, pp. 185489–185502, 2020.

[17] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," IEEE Access, vol. 8, pp. 58194–58205, 2020.

[18] M. S. Mehedi, A. Shamim, and M. B. I. Reaz, "Deep transfer learning based intrusion detection system for electric vehicular networks," Sensors, vol. 21, no. 14, Art. no. 4736, Jul. 2021.

[19] A. Vaswani et al., "Attention is all you need," in Advances in Neural Information Processing Systems (NeurIPS), vol. 30, 2017, pp. 5998–6008.

[20] V. K. K. and P. J. Arul Leena Rose, "The Evolution of In-Vehicle Intrusion Detection Systems through Deep Learning: A Systematic Study," International Journal of Information Technology Research Studies (IJITRS), vol. 1, no. 1, pp. 1–6, Apr. 2025, https://doi.org/10.5281/zenodo.15309382.

[21] S. F. Lokman, A. T. Othman, and M. H. Abu-Bakar, "Intrusion detection system for automotive Controller Area Network (CAN) bus system: A review," EURASIP Journal on Wireless Communications and Networking, vol. 2019, Art. no. 184, Jul. 2019.

[22] W. Wu et al., "A survey of intrusion detection for in-vehicle networks," IEEE Transactions on Intelligent Transportation Systems, vol. 21, no. 3, pp. 919–933, Mar. 2020.

[23] UNECE, "UN Regulation No. 155: Uniform provisions concerning the approval of vehicles with regard to cybersecurity and cybersecurity management system," United Nations Economic Commission for Europe, Geneva, Switzerland, 2021.