Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

BLOCKCHAIN-ENABLED-DEEP-LEARNING-FOR-REAL-TIME-FDI-DETECTION-IN-VEHICULAR-CLOUD-NETWORKS

HANI AL-BALASMEH[1]
[1]Informatics Engineering Department, College of Engineering, University of Technology Bahrain (UTB), Bahrain
E-mail: h.albalasmeh@utb.edu.bh, hbalasmeh04@gmail.com

ABSTRACT
Vehicular Cloud Networks (VCNs) support supercritical services through continuous data exchange among vehicles, Roadside Units (RSUs), edge nodes, and cloud services, but their openness makes them vulnerable to False Data Injection (FDI) attacks. Existing studies typically emphasize either intelligent attack detection or trustworthy evidence management; consequently, they do not fully address the combined requirements of real-time classification, tamper-resistant verification, and adaptive response. This paper proposes a blockchain-enabled CNN-LSTM framework that analyzes mobility, temporal, communication, and consistency-based features and then records compact message hashes and detection metadata on a distributed ledger. Smart contracts validate the transaction structure, verify message integrity, update vehicle trust scores, and trigger restriction actions when repeated malicious behavior is observed. The methodological contribution is a single detection-verification-trust pipeline that connects feature engineering, sequence-based deep learning, hash-only on-chain evidence, and trust-based response. On the reported held-out test set, the framework achieved 98.42% accuracy, 98.11% precision, 98.73% recall, a 98.42% F1-score, and a 1.39% false positive rate. The CNN-LSTM classifier required 12.6 ms per inference, while the complete preprocessing-to-confirmation path required 73.4 ms; 10,000 compact blockchain transactions required 8.2 MB. Comparative and ablation analyses show that engineered temporal features improve detection, the hybrid CNN-LSTM outperforms conventional and standalone models, and blockchain adds traceability and evidence integrity with measurable latency. The findings provide operational evidence that an integrated, metadata-oriented design can strengthen cyber-resilience and data trustworthiness in real-time VCN security, subject to the stated simulation and deployment limitations.
Keywords: *Vehicular Cloud Networks, False Data Injection, CNN-LSTM, Blockchain Verification, Trust Management*

1. INTRODUCTION

Vehicular Cloud Networks (VCNs) combine vehicular communication, edge computing, and cloud services to support collision avoidance, route optimization, traffic management, autonomous-driving assistance, and emergency response. Vehicles continuously report speed, position, acceleration, heading, traffic events, and packet-level information to neighboring vehicles, Roadside Units (RSUs), and cloud-assisted applications. The same openness, mobility, and connectivity that enable these services also expose them to cyberattacks in which manipulated data can directly influence safety-critical decisions [1]-[4].

False Data Injection (FDI) attacks are particularly serious because they target the correctness of the information used by vehicular applications. A malicious vehicle, compromised RSU, or unauthorized node may report false speed, fabricated location coordinates, forged congestion alerts, manipulated sensor readings, or false emergency events. If accepted, such messages can mislead route planning, hide real hazards, create unnecessary warnings, and disrupt cooperative driving [3]-[7].

Signature- and rule-based intrusion detection methods are limited by known attack patterns and rigid decision rules. Conventional machine learning can identify some deviations but may not capture longer temporal dependencies. Deep learning can learn nonlinear mobility and communication patterns, yet detection alone does not guarantee the integrity, traceability, or non-repudiation of the evidence used to make a security decision. Blockchain can provide tamper-resistant logging and decentralized verification, but it also introduces latency, storage, consensus, and computational costs [8]-[13]. The technical problem is therefore not simply to maximize classification accuracy; it is to jointly satisfy detection quality, evidence

integrity, adaptive trust management, and real-time feasibility.

1.1 New Knowledge, Scope, and Delimitations

The new knowledge created by this study is an experimentally evaluated integration strategy in which an engineered CNN-LSTM detector, a hash-only blockchain evidence model, and a smart-contract trust update mechanism operate as one real-time security pipeline. The study quantifies not only classification quality but also end-to-end latency, transaction size, storage overhead, and trust-score behavior. This joint evidence is intended to clarify the practical trade-off between stronger integrity guarantees and the additional delay introduced by blockchain verification.

The scope is delimited to binary classification of normal versus FDI messages in a VCN consisting of vehicles, RSUs, edge nodes, cloud services, and a blockchain layer. The attack model covers manipulated speed, location, acceleration, event reports, and packet behavior. The work evaluates a controlled experimental configuration and does not claim validation on production vehicles, a public road deployment, every possible consensus protocol, or every coordinated cyberattack. Physical vehicle control, hardware failures, privacy-preserving identity management, and multi-class attack attribution are outside the present scope.

1.2 Research Objectives and Hypotheses

The research objectives are to:

- O1: formulate a reproducible VCN and FDI attack model that represents manipulated mobility, event, and communication data;
- O2: design an engineered CNN-LSTM detector that captures local feature patterns and temporal dependencies;
- O3: design a lightweight blockchain evidence model that stores message hashes and detection metadata rather than complete vehicular messages;
- O4: implement smart-contract-based integrity verification, trust updating, and response decisions; and
- O5: evaluate detection performance, comparative effectiveness, convergence, blockchain overhead, latency, trust behavior, and component contributions.

H1: The engineered CNN-LSTM detector will achieve higher F1-score and lower false positive rate than the evaluated conventional machine-learning and standalone deep-learning baselines. H2: Hash-only blockchain verification and smart-contract trust updating will provide tamper-resistant evidence and adaptive response while maintaining an end-to-end latency compatible with the evaluated real-time configuration.

1.3 Research Contributions

- An FDI-specific detection-verification-trust architecture for Vehicular Cloud Networks rather than a general-purpose intrusion detector.
- A feature engineering and sequence-learning pipeline covering speed deviation, location displacement, message frequency, movement consistency, packet behavior, and event reporting.
- A compact blockchain transaction model that records message hashes and essential detection metadata to reduce on-chain storage.
- A smart-contract trust mechanism that rewards consistent normal behavior, penalizes repeated malicious behavior, and triggers restriction decisions below a trust threshold.
- A multi-criteria evaluation that combines classification metrics, ROC and precision-recall analysis, convergence, baseline and related-work comparison, blockchain performance, trust evolution, latency decomposition, and ablation analysis.

The remainder of the paper is organized as follows. Section 2 reviews related work and derives the research gap and hypotheses. Section 3 presents the framework and threat model. Section 4 explains the reproducible methodology. Section 5 presents the evaluation and discussion. Section 6 states limitations, assumptions, and threats to validity. Section 7 concludes the paper and identifies future research directions.

2. RELATED WORK

2.1 Deep Learning-Based Intrusion Detection in Vehicular Networks

Deep learning has been widely adopted for vehicular intrusion detection because it can extract nonlinear patterns from high-dimensional traffic and behavioral data. Lampe and Meng [2] and Luo et al. [14] reviewed CNN, recurrent, LSTM, autoencoder, and hybrid architectures and reported strong potential relative to shallow methods. They also identified limitations involving dataset representativeness, generalization to unseen attacks, computational cost, and deployment in time-sensitive environments. Song et al. [8] demonstrated a DCNN-based CAN intrusion detector, Yang and Effatparvar [15] applied attention-oriented deep learning to CAN security, and Karthiga et al. [9] and Bangui et al. [16]

evaluated learning-based vehicular intrusion detection. These approaches establish the value of learned representations, but much of the literature focuses on in-vehicle or general vehicular attacks rather than cloud-assisted FDI detection with auditable evidence.

2.2 False Data Injection and Misbehavior Detection

FDI and false-message detection studies focus on whether a reported value is consistent with surrounding traffic, neighboring vehicles, or learned behavior. Zhang et al. [3] used traffic data and vehicle consensus; Zhao et al. [17] used cloud-based data fusion; Idris et al. [5] used neighboring public transport vehicles; Uprety et al. [7] and Wang et al. [6] applied federated learning; and Lv et al. [18] combined privacy-preserving federated learning with blockchain. These studies show that spatial and temporal context can improve detection and that distributed learning can reduce direct sharing of raw data. However, the integrity, traceability, latency, and trust-response properties of the resulting detection evidence remain separate or incompletely evaluated concerns.

2.3 Blockchain-Enabled Trust and Intrusion Detection

Blockchain has been investigated as a means of providing decentralized trust, immutability, transparency, and tamper-resistant logging. Liu et al. [10] integrated blockchain and federated learning for collaborative intrusion detection in vehicular edge computing. Abou El Houda et al. [11] further examined blockchain-enabled federated learning, and Abdel-Basset et al. [12] studied federated intrusion detection in blockchain-based smart transportation systems. Yang et al. [19] introduced blockchain-based traffic-event validation and trust verification, while Lv et al. [18] integrated privacy-preserving federated learning and blockchain for VANET misbehavior detection. Al-Quayed et al. [13] surveyed the broader area. These studies demonstrate that blockchain can strengthen accountability and collaborative security, but many address general intrusion detection or isolated trust functions rather than the specific sequence of FDI detection, evidence preservation, trust updating, and response in VCNs.

2.4 Conflicting Perspectives and Design Trade-offs

The literature presents three relevant tensions. First, increasingly complex deep-learning models may improve detection but increase computation and reduce interpretability. Second, blockchain can strengthen integrity and decentralization, yet consensus and on-chain storage may conflict with real-time vehicular latency and scalability. Third, federated or decentralized schemes improve data locality, but model poisoning and unreliable participants may shift rather than eliminate the trust problem [20]. The proposed design adopts a pragmatic position: detection is performed off-chain at the edge, only compact hashes and metadata are recorded on-chain, and trust updates are automated through smart contracts. This design does not remove the trade-offs; it makes them explicit and measurable.

2.5 Research Gap, Problem Statement, and Hypotheses

The reviewed studies leave four connected gaps: limited attention to real-time FDI in cloud-assisted vehicular environments; separation between intelligent detection and tamper-resistant evidence; limited integration of trust-score-based response; and incomplete evaluation of latency, storage, false positives, and deployment constraints. The resulting problem statement is: how can a VCN detect manipulated vehicular messages, preserve verifiable evidence, update source trust, and respond within a measurable real-time budget? H1 and H2 in Section 1.2 directly operationalize this problem and are evaluated in Sections 5.4-5.9.

Table 1: Comparative Analysis with Related Vehicular Security Frameworks

Framework	Detection Method	Environment	Blockchain	FDI-Specific	Principal Limitation
Lampe and Meng [2]	DL IDS survey	Automotive networks	No	Partial	Survey; no integrated VCN evidence pipeline.
Zhang et al. [3]	Traffic data + consensus	Internet of Vehicles	No	Yes	No blockchain evidence integrity.
Liu et al. [10]	Blockchain + FL	Vehicular edge	Yes	No	Collaborative IDS; not FDI-specific.
Abou El Houda et al. [11]	Blockchain-enabled FL	Vehicular edge	Yes	No	No complete FDI response path.
Wang et al. [6]	Federated learning	Internet of Vehicles	No	Yes	No tamper-resistant evidence/trust scoring.
Yang et al. [19]	Blockchain event validation	VANET	Yes	Partial	No deep FDI detector or end-to-end latency.
Lv et al. [18]	FL + blockchain	VANET	Yes	Yes	Different VANET FL-blockchain pipeline.
Proposed framework	Blockchain CNN-LSTM	Vehicular Cloud Network	Yes	Yes	Detection, evidence, trust, and response.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3. PROPOSED FRAMEWORK

The proposed framework combines intelligent attack detection, decentralized evidence verification, tamper-resistant logging, and dynamic trust management. It contains six components: vehicular data generation; V2V/V2I communication and collection; preprocessing and feature extraction; CNN-LSTM-based FDI detection; blockchain verification and trust management; and secure vehicular cloud services. Manipulated messages are therefore analyzed and verified before they can influence route planning, safety applications, traffic management, or other cloud-assisted services.

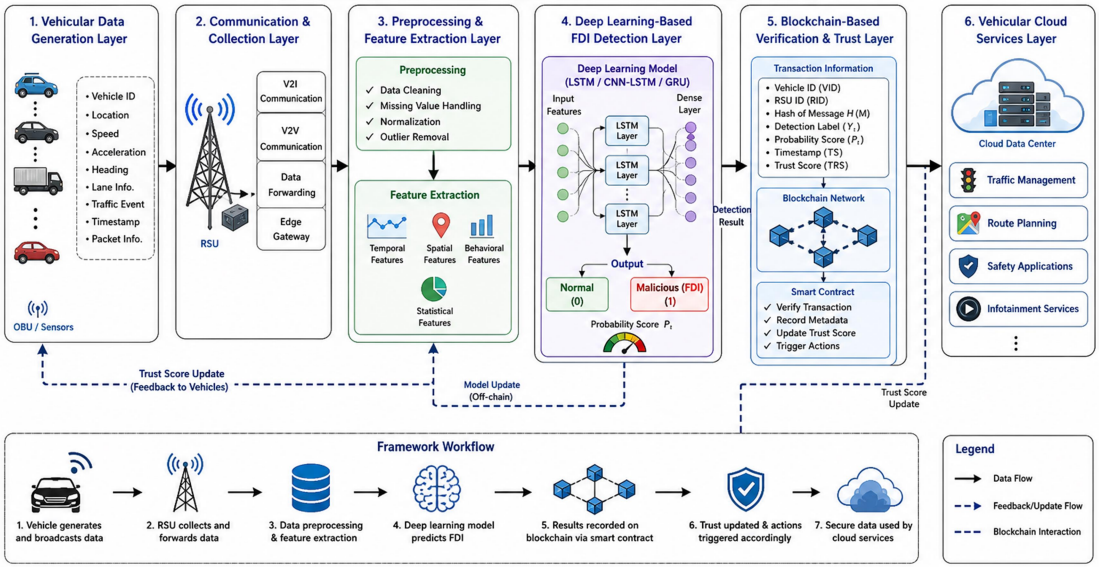

Figure 1: Proposed Blockchain-Enabled Framework

3.1 Framework Overview

Vehicles generate identity, speed, location, acceleration, heading, event, timestamp, and packet-related data. RSUs and edge gateways collect the messages and apply cleaning, normalization, missing-value handling, outlier detection, and feature construction. The CNN-LSTM detector outputs a class label and malicious probability. A normal message is forwarded, whereas a suspicious message is converted into a blockchain transaction containing identifiers, the message hash, detection metadata, timestamp, and trust score. The smart contract verifies the transaction and updates the vehicle status.

3.2 Threat Model and FDI Attack Scenario

The VCN contains legitimate and malicious vehicles and may also contain a compromised RSU or communication link. An attacker attempts to lower the reliability of cloud services by modifying one or more mobility, event, or packet features. The attacker is assumed capable of injecting or changing message content but not of breaking the cryptographic hash function or rewriting a confirmed ledger record. This assumption is stated explicitly because blockchain protects recorded evidence; it does not independently prove that a source measurement was truthful before detection [4], [17].

3.3 System Workflow

The workflow begins when a vehicle broadcasts a message to an RSU or edge node. After preprocessing, the engineered feature sequence is submitted to the trained detector. A normal result permits forwarding to cloud services. A malicious result creates a blockchain transaction, invokes the smart contract, reduces trust, records an alert, and may flag, isolate, or restrict the source. Feedback from the trust layer affects subsequent decisions, enabling adaptive rather than one-time security control.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

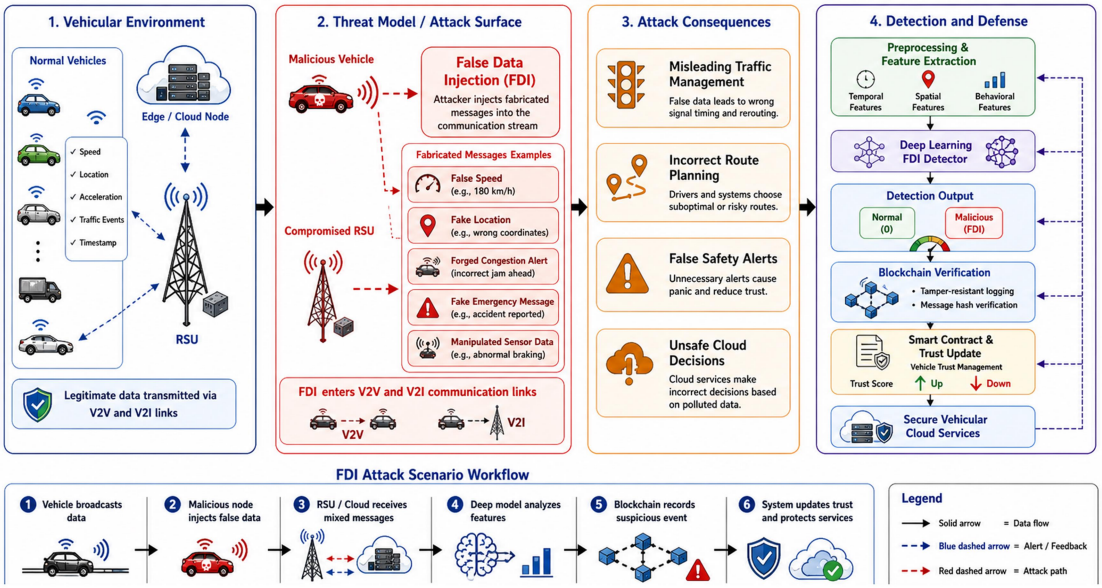

Figure 2: Threat Model and FDI Attack Scenario in Vehicular Cloud Networks

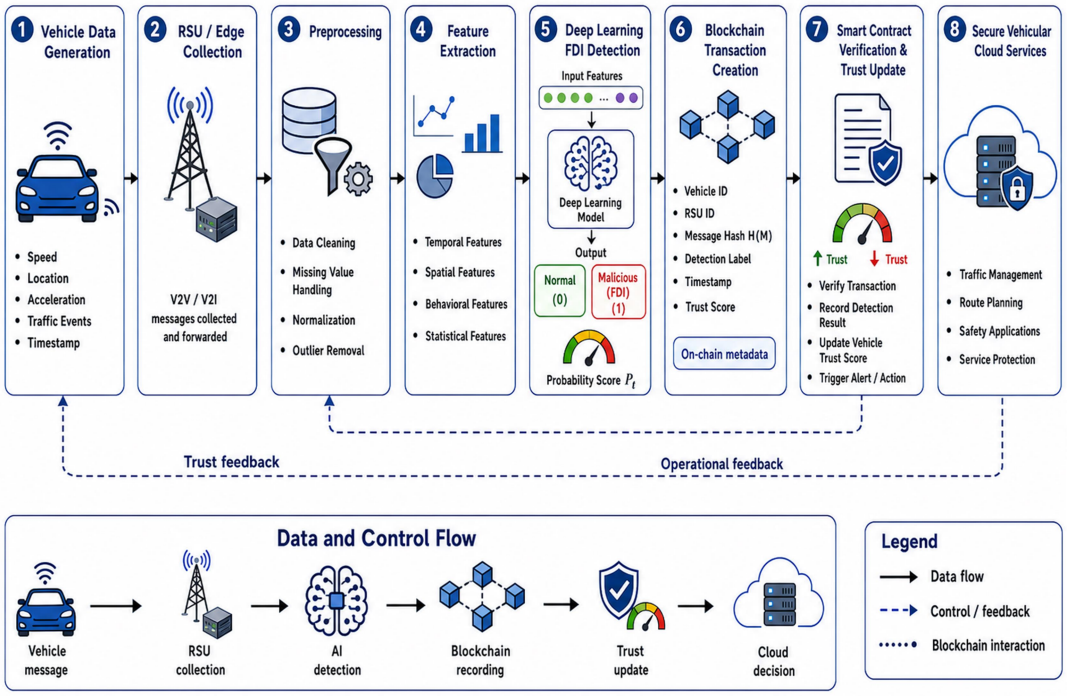

Figure 3: System Workflow of the Proposed Real-Time FDI Detection Framework

3.4 Deep Learning-Based Detection Layer

Let the vehicular feature vector at time t contain speed v, location l, acceleration a, heading h, message frequency f, packet behavior p, distance-related features d, and event-reporting information e:

$$X_t = \{v_t, l_t, a_t, h_t, f_t, p_t, d_t, e_t\} \quad (1)$$

$$Y_t = F(X_t) \quad (2)$$

$$Y_t \in \{0, 1\} \quad (3)$$

$$P_t = P(Y_t = 1 \mid X_t) \quad (4)$$

$$Y_t = \begin{cases} 1, & P_t \geq \theta \\ 0, & P_t < \theta \end{cases} \quad (5)$$

A lower threshold increases sensitivity but may increase false alarms, whereas a higher threshold reduces false alarms but risks missed attacks. The threshold is therefore selected on validation data and held fixed for testing.

3.5 Blockchain Verification and Trust Layer

$$Tx = \{\mathrm{VID}, \mathrm{RID}, H(M), Y_t, P_t, \mathrm{TS}, \mathrm{TRS}\} \quad (6)$$

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Only the message hash and essential metadata are recorded on-chain. This avoids the storage cost and privacy exposure associated with recording complete vehicular messages [10], [18], [19]. The trust score is updated according to the detection result:

$$\mathrm{TRS}_{new} = \begin{cases} \mathrm{TRS}_{old} + \alpha, & Y_t = 0 \\ \mathrm{TRS}_{old} - \beta, & Y_t = 1 \end{cases} \quad (7)$$

$$\mathrm{TRS}_{new} = \min(1, \max(0, \mathrm{TRS}_{new})) \quad (8)$$

$$\mathrm{Status(VID)} = \begin{cases} \text{Suspicious}, & \mathrm{TRS}_{new} < \gamma \\ \text{Trusted}, & \mathrm{TRS}_{new} \geq \gamma \end{cases} \quad (9)$$

3.6 Smart Contract Verification Flow

The smart contract checks required fields, validates the message hash, records detection metadata, updates trust, and triggers a response. Normal behavior increases or maintains trust; malicious behavior decreases trust and creates an alert. A source whose trust falls below γ may be restricted or subjected to additional verification.

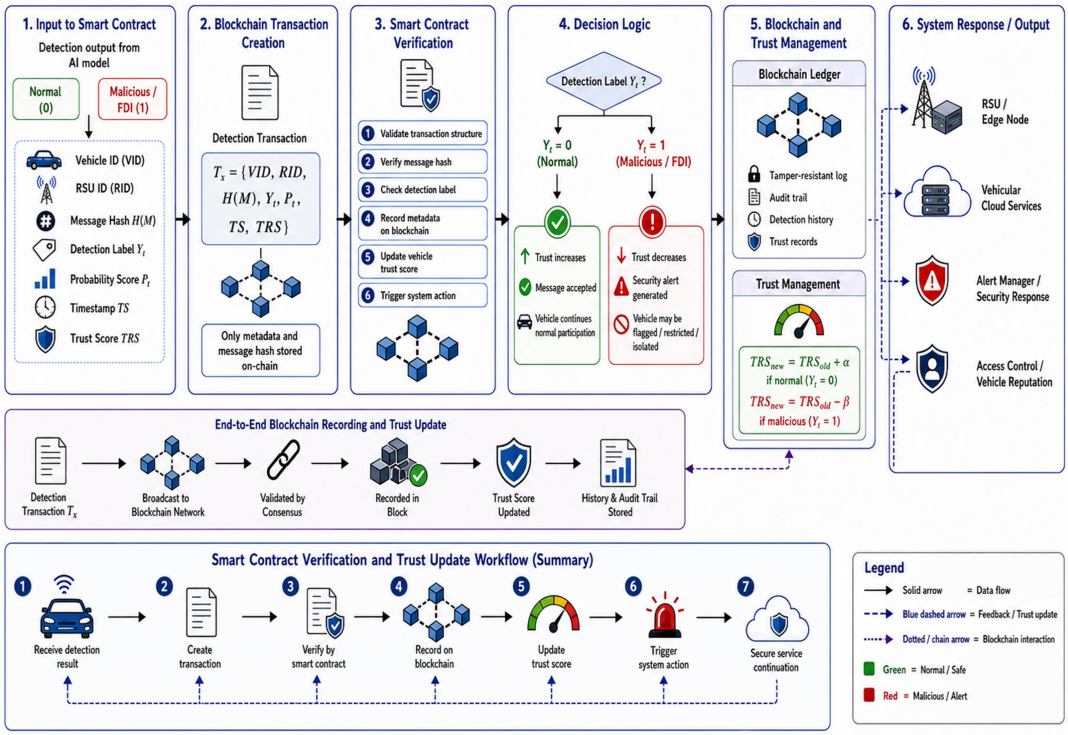

Figure 4: Smart Contract-Based Verification and Trust Update Flow in Vehicular Cloud Networks

4. METHODOLOGY

The methodology links mathematical system and attack models, preprocessing and feature engineering, CNN-LSTM training and inference, blockchain evidence creation, smart-contract trust updating, integrated response, computational analysis, and multi-criteria evaluation. The sequence is stated so that another researcher can implement the same logical experiment with the reported controls.

4.1 System Model

$$S = \{V, R, E, C, B\} \quad (10)$$

$$V = \{v_1, v_2, \ldots, v_n\} \quad (11)$$

$$M_i^t = \{\mathrm{VID}_i, \mathrm{Loc}_i^t, \mathrm{Sp}_i^t, \mathrm{Acc}_i^t, \mathrm{Dir}_i^t, \mathrm{TS}_i^t, \mathrm{Event}_i^t, \mathrm{Pkt}_i^t\} \quad (12)$$

Here V, R, E, C, and B denote vehicles, RSUs, edge nodes, cloud services, and the blockchain network. Each message is transformed into a structured vector before classification.

4.2 FDI Attack Model

$$\widehat{M_i^t} = M_i^t + \Delta_i^t \quad (13)$$

$$\Delta_i^t = \{\Delta\mathrm{Loc}_i^t, \Delta\mathrm{Sp}_i^t, \Delta\mathrm{Acc}_i^t, \Delta\mathrm{Event}_i^t, \Delta\mathrm{Pkt}_i^t\} \quad (14)$$

$$\widehat{\mathrm{Sp}_i^t} = \mathrm{Sp}_i^t + \delta_{sp} \quad (15)$$

$$\widehat{\mathrm{Loc}_i^t} = \mathrm{Loc}_i^t + \delta_{loc} \quad (16)$$

The detector must distinguish legitimate M_i^t from manipulated M̂_i^t before the data are consumed by vehicular cloud services.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.3 Preprocessing and Feature Engineering

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{17}$$

$$z = \frac{x - \mu}{\sigma} \tag{18}$$

$$|z| > \tau \tag{19}$$

$$\mathrm{SD}_i^t = \left|\mathrm{Sp}_i^t - \mathrm{Sp}_i^{t-1}\right| \tag{20}$$

$$\mathrm{LD}_i^t = \sqrt{\begin{matrix}(\mathrm{Lat}_i^t - \mathrm{Lat}_i^{t-1})^2 + \\ (\mathrm{Lon}_i^t - \mathrm{Lon}_i^{t-1})^2\end{matrix}} \tag{21}$$

$$\mathrm{MF}_i = \frac{N_i}{T} \tag{22}$$

$$\mathrm{CS}_i^t = 1 - \frac{|\mathrm{LD}_i^t - \widehat{\mathrm{LD}}_i^t|}{\mathrm{LD}_i^t + \varepsilon} \tag{23}$$

$$X_i^t = \{x_1, x_2, \ldots, x_k\} \tag{24}$$

$$S_i^t = \{X_i^{t-w+1}, \ldots, X_i^t\} \tag{25}$$

Algorithm 1: Vehicular Data Preprocessing and Feature Engineering

Require: raw vehicular message set M; Ensure: normalized feature matrix X.
1. Remove duplicate, incomplete, and inconsistent records; verify timestamp order.
2. Impute missing numerical values using the selected statistical rule.
3. Compute z using (18); mark or exclude values for which $|z| > \tau$.
4. Normalize numerical attributes using (17).
5. Compute speed deviation (20), location displacement (21), message frequency (22), and consistency score (23).
6. Construct X_i^t, append it to X, and create sliding-window sequences S_i^t using (25).

4.4 CNN-LSTM Detection Model

$$F_\theta : S_i^t \to Y_i^t \tag{26}$$

$$Y_i^t = \begin{cases} 0, & \text{normal message} \\ 1, & \text{FDI message} \end{cases} \tag{27}$$

$$Z_i^t = \mathrm{CNN}(S_i^t) \tag{28}$$

$$H_i^t = \mathrm{LSTM}(Z_i^t) \tag{29}$$

$$P_i^t = \sigma(W_o H_i^t + b_o) \tag{30}$$

$$\widehat{Y}_i^t = \begin{cases} 1, & P_i^t \geq \theta_d \\ 0, & P_i^t < \theta_d \end{cases} \tag{31}$$

$$\mathcal{L}_{BCE} = -\frac{1}{m}\sum_{i=1}^{m}[Y_i \log(P_i) + (1 - Y_i)\log(1 - P_i)] \tag{32}$$

$$\theta^* = \arg\min_\theta \mathcal{L}_{BCE} \tag{33}$$

Algorithm 2: Training Procedure for the Deep Learning FDI Detector

Require: D_train, D_val, window w, learning rate η, and epochs E; Ensure: F_θ*.
1. Construct time-window sequences according to (25) and initialize θ.
2. For each epoch and mini-batch, compute CNN features (28), LSTM states (29), probability (30), and BCE loss (32).
3. Update θ through gradient-based optimization.
4. Evaluate on D_val; retain θ* when validation performance improves.
5. Select the detection threshold on D_val and keep it fixed for test evaluation.

4.5 Blockchain Transaction and Evidence Model

$$Tx_i^t = \{\mathrm{VID}_i, \mathrm{RID}_j, H(M_i^t), \widehat{Y}_i^t, P_i^t, \mathrm{TS}_i^t, \mathrm{TRS}_i^t\} \tag{34}$$

$$H(M_i^t) = \mathrm{Hash}(M_i^t) \tag{35}$$

$$\mathrm{Verify}(M_i^t) = \begin{cases} \text{Valid}, & H(M_i^t) = H(M_i^t)' \\ \text{Invalid}, & \text{otherwise} \end{cases} \tag{36}$$

$$0 \leq \mathrm{TRS}_i^t \leq 1 \tag{37}$$

A changed message produces a different hash. The ledger therefore verifies the integrity of the recorded message representation without storing the complete payload.

4.6 Smart Contract Verification and Trust Management

$$\mathrm{TRS}_i^0 = \mathrm{TRS}_{init} \tag{38}$$

$$\mathrm{TRS}_i^{t+1} = \begin{cases} \mathrm{TRS}_i^t + \alpha(1 - \mathrm{TRS}_i^t), & \widehat{Y}_i^t = 0 \\ \mathrm{TRS}_i^t - \beta \mathrm{TRS}_i^t, & \widehat{Y}_i^t = 1 \end{cases} \tag{39}$$

$$\mathrm{Status}(v_i) = \begin{cases} \text{Trusted}, & \mathrm{TRS}_i^{t+1} \geq \gamma \\ \text{Suspicious}, & \mathrm{TRS}_i^{t+1} < \gamma \end{cases} \tag{40}$$

Algorithm 3: Real-Time FDI Detection Using the Trained Model

1. Receive M_i^t and apply Algorithm 1.
2. Construct S_i^t and submit it to F_θ*.
3. Compute P_i^t; set Ŷ_i^t according to θ_d.
4. Return Ŷ_i^t and P_i^t.

Algorithm 4: Blockchain Verification and Smart-Contract Trust Update

1. Compute H(M_i^t), create Tx_i^t, and submit it to the smart contract.
2. Reject incomplete transactions; verify the hash using (36).
3. If valid, record metadata, update trust using (39), and determine status using (40).
4. Return the blockchain record and vehicle status.

4.7 Integrated Detection and Response Procedure

Algorithm 5: Integrated Blockchain-Enabled FDI Detection

1. Receive the message; preprocess and classify it through Algorithms 1 and 3.
2. Execute blockchain verification and trust updating through Algorithm 4.
3. Accept and forward a normal message from a trusted source.
4. Flag a malicious message from a source whose trust remains above γ and request additional verification.
5. Restrict the source and notify the RSU/cloud security manager when trust falls below γ.
6. Record the final decision in the local security log.

4.8 Computational Complexity

$$O_{pre} = O(mk) \tag{41}$$

$$O_{dl} = O(mwh^2) \tag{42}$$

$$O_{bc} = O(n_b T_v) \tag{43}$$

$$O_{total} = O_{pre} + O_{dl} + O_{bc} \tag{44}$$

The principal controllable costs are sequence processing and blockchain validation. Hash-only

metadata recording reduces storage and transaction payload relative to full-message on-chain designs.

4.9 Evaluation Metrics

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (45)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (46)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (47)$$

$$F_1 = 2\frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \quad (48)$$

$$\text{FPR} = \frac{FP}{FP + TN} \quad (49)$$

$$L_d = T_{out} - T_{in} \quad (50)$$

$$L_b = T_{confirm} - T_{submit} \quad (51)$$

$$L_{total} = L_{pre} + L_d + L_{create} + L_{contract} + L_b \quad (52)$$

$$O_s = N_t \cdot S_t \quad (53)$$

4.10 Reproducibility Protocol and Experimental Controls

The repeatable procedure is: (1) construct normal VCN messages using the fields in (12); (2) create FDI samples by manipulating the attributes listed in (14)-(16); (3) apply Algorithm 1; (4) split data into mutually exclusive training, validation, and held-out test subsets; (5) tune model and threshold choices only on training/validation data; (6) train for 30 epochs, save the best validation model, and freeze it before testing; (7) evaluate the same held-out records with every detection baseline; (8) create blockchain transactions only after classification; and (9) evaluate trust and latency using the same thresholds and transaction definitions. The reported confusion matrix contains 2,000 held-out records, and the trust experiment uses $\gamma = 0.45$. To support exact replication, implementations should archive the data-generation seed, split indices, software versions, CNN/LSTM layer widths, optimizer, learning rate, batch size, and consensus configuration with the code. These implementation constants are not inferred or fabricated here; they must be the values actually used by the author.

Table 2: Reported Experimental Configuration and Controls

Parameter	Reported Configuration
Network environment	VCN with vehicles, RSUs, edge nodes, cloud layer, and blockchain layer.
Attack attributes	Speed, location, acceleration, event reports, and packet behavior.
Input representation	Mobility, temporal, spatial, communication, behavioral, and consistency-based features.
Detection model	Hybrid CNN-LSTM; binary normal/FDI classification.
Held-out test records	2,000 records, as reported by the confusion matrix.
Training horizon	30 epochs; best model selected by validation performance.
On-chain content	Message hash and essential detection metadata only.
Trust threshold	$\gamma = 0.45$ in the reported trust-threshold experiment.
Evaluation families	Classification, discrimination, convergence, comparison, blockchain, trust, latency, and ablation.

5. RESULTS AND ANALYSIS

5.1 Justification of the Analysis Criteria

The result criteria are selected to match the dual security and operational objectives. Recall and false negative behavior address the safety risk of missed attacks; precision and FPR address the operational cost of false alarms; F1-score balances these errors; ROC-AUC and average precision assess threshold-independent discrimination; training and validation curves assess convergence and overfitting; baseline and related-work comparisons test H1 and novelty; blockchain latency, transaction size, and storage test H2 and scalability; trust evolution tests adaptive response; and ablation isolates the contribution of features, temporal learning, and the complete integration. No single accuracy value is treated as sufficient evidence.

5.2 Detection Performance

The proposed framework achieved 98.42% accuracy, 98.11% precision, 98.73% recall, a 98.42% F1-score, and a 1.39% false positive rate. The confusion matrix contains 982 correctly classified normal records, 995 correctly classified attack records, 14 false positives, and 9 false negatives. The high recall is important for safety-sensitive services, while the low FPR limits unnecessary alerts.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: Detection and Response Performance of the Proposed Framework

Metric	Value
Accuracy	98.42%
Precision	98.11%
Recall	98.73%
F1-score	98.42%
False positive rate	1.39%
CNN-LSTM inference latency	12.6 ms
Model-to-confirmation latency	51.0 ms
Complete end-to-end latency	73.4 ms

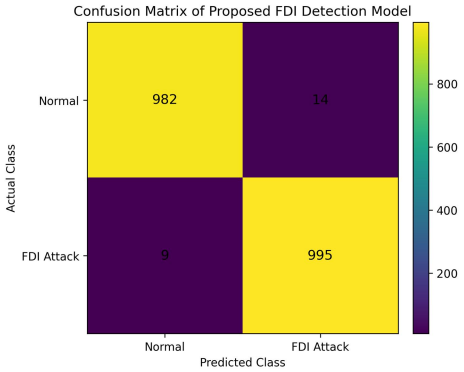

Figure 5: Confusion Matrix of the Proposed FDI Detection Model

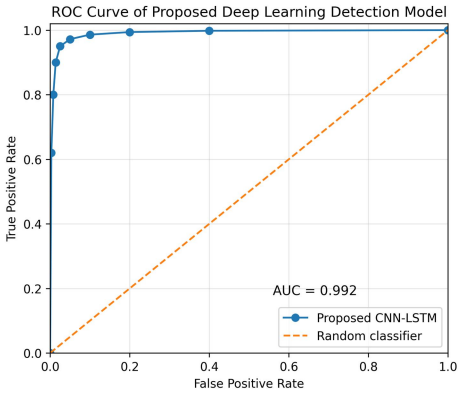

Figure 6: ROC Curve of the Proposed Deep Learning Detection Model (AUC = 0.992)

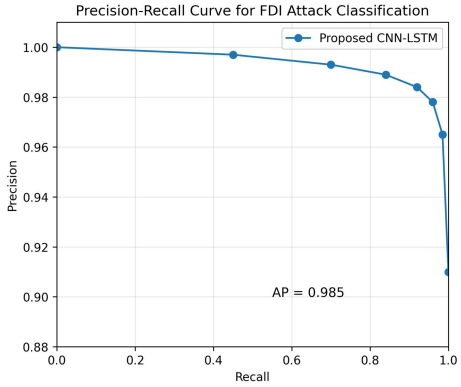

Figure 7: Precision-Recall Curve for FDI Attack Classification (AP = 0.985)

The ROC-AUC of 0.992 and average precision of 0.985 confirm strong discrimination across thresholds. These curves complement the point estimates in Table 3 and reduce the risk of drawing conclusions from a single operating threshold.

5.3 Training and Convergence

Training and validation accuracy increase steadily across 30 epochs, while both losses decrease and remain close. The curves do not show a large divergence, which supports stable learning in the reported configuration. Nevertheless, the absence of repeated-seed confidence intervals is acknowledged as a limitation in Section 6.

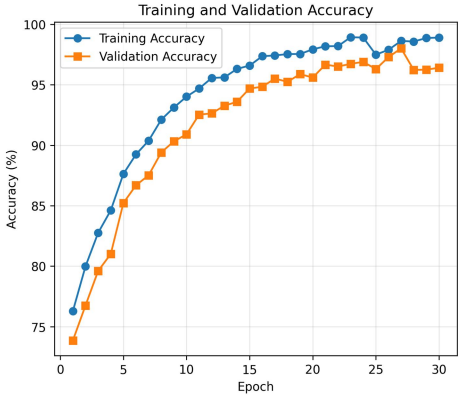

Figure 8: Training and Validation Accuracy of the Proposed CNN-LSTM Model

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

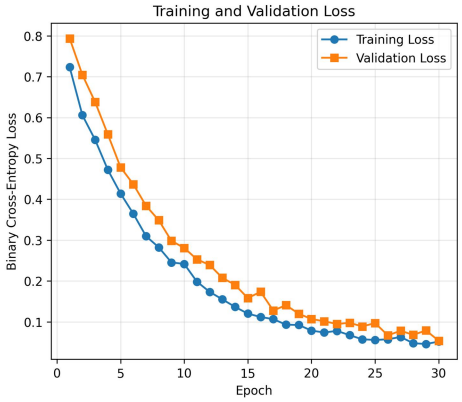

Figure 9: Training and Validation Loss of the Proposed CNN-LSTM Model

5.4 Comparison with Baseline Models

All baselines are evaluated on the same reported classification task. The proposed framework improves F1-score by 7.39 percentage points relative to SVM, 5.44 points relative to Random Forest, 2.74 points relative to standalone LSTM, 3.42 points relative to standalone CNN, and 0.89 points relative to CNN-LSTM without blockchain. Its FPR is also the lowest. The gain over the CNN-LSTM without blockchain should be interpreted carefully: blockchain does not directly retrain the classifier; the complete framework adds verified decision handling and trust response, while the ablation configuration also represents the full engineered pipeline used in the integrated system.

Table 4: Performance Comparison with Baseline Detection Models

Model	Accuracy	Precision	Recall	F1-score	FPR
SVM	91.64%	90.87%	91.20%	91.03%	6.82%
Random Forest	93.28%	92.91%	93.06%	92.98%	5.41%
LSTM	95.76%	95.32%	96.05%	95.68%	3.28%
CNN	95.14%	94.89%	95.11%	95.00%	3.74%
CNN-LSTM without blockchain	97.61%	97.22%	97.84%	97.53%	2.05%
Proposed blockchain-enabled CNN-LSTM	98.42%	98.11%	98.73%	98.42%	1.39%

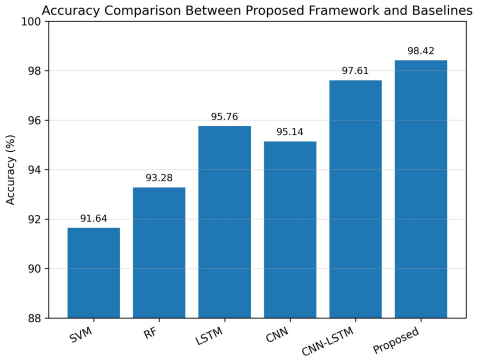

Figure 10: Accuracy Comparison between the Proposed Framework and Baseline Models

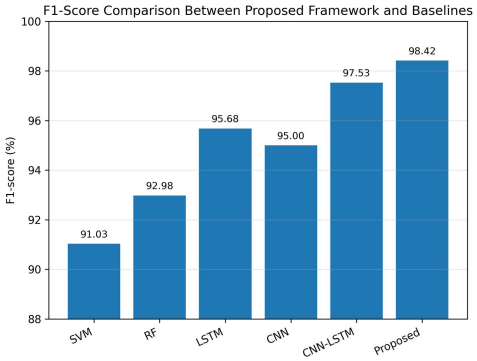

Figure 11: F1-Score Comparison between the Proposed Framework and Baseline Models

5.5 Comparison with Related Research Directions

The contribution differs from deep-learning IDS surveys and detectors [2], [8], [9], [14]-[16] because it targets cloud-assisted FDI and evaluates evidence handling. It differs from false-message and federated FDI studies [3]-[7], [17], [18] because it adds tamper-resistant records and trust response. It differs from blockchain/federated vehicular IDS frameworks [10]-[13], [18], [19] because it centers the pipeline on FDI-specific temporal features and reports classifier, blockchain, trust, and end-to-end latency evidence together. The comparison therefore addresses both research contribution and achievement of objectives in the context of prior literature.

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: Contribution Relative to Related Research Directions

Research Direction	Strength	Remaining Gap	Contribution of This Study
Deep-learning vehicular IDS [2], [8], [9], [14]-[16]	Strong nonlinear detection	Often in-vehicle/general IDS; evidence integrity not central	FDI-specific CNN-LSTM plus verified evidence and trust response.
FDI/misbehavior detection [3]-[7], [17], [18]	Targets manipulated vehicular information	Limited tamper-resistant logging and trust updating	Adds hash-only blockchain evidence and smart contracts.
Blockchain/federated vehicular IDS [10]-[13], [18], [19]	Distributed trust and collaboration	Often general IDS; overhead not tied to an FDI pipeline	Reports FDI detection, latency, storage, and adaptive response jointly.

5.6 Blockchain Latency and Storage

The metadata-oriented blockchain design achieved 38.4 ms average confirmation latency, 0.82 KB average transaction size, 11.7 ms average smart-contract execution time, 4.3 ms average trust-update time, and 8.2 MB storage for 10,000 transactions. Latency rises with transaction load, but storing only hashes and essential fields avoids the substantially larger burden of recording raw vehicular messages.

Table 6: Blockchain Performance Analysis

Metric	Value
Average transaction confirmation latency	38.4 ms
Average transaction size	0.82 KB
Average smart-contract execution time	11.7 ms
Average trust-update time	4.3 ms
Storage overhead for 10,000 transactions	8.2 MB

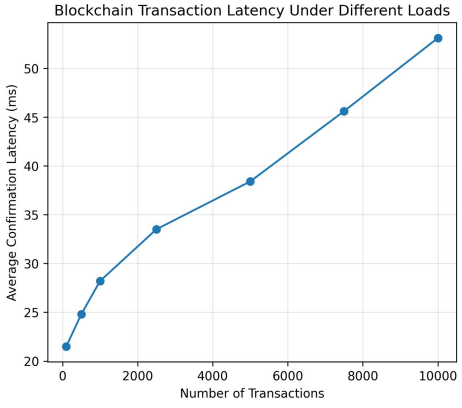

Figure 12: Blockchain Transaction Latency under Different Transaction Loads

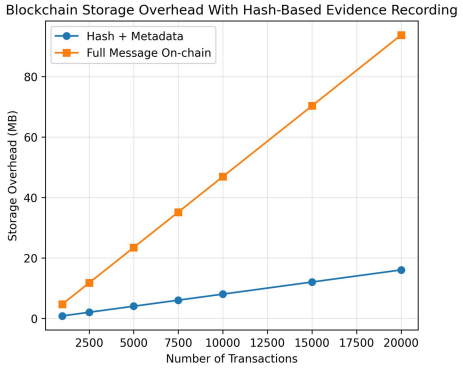

Figure 13: Blockchain Storage Overhead with Hash-Based Evidence Recording

5.7 Trust-Score Analysis

Normal vehicles retain or gradually increase trust, while repeated malicious detections decrease trust. At $\gamma = 0.45$, sources below the threshold are treated as suspicious and may be restricted. This behavior demonstrates that the framework does not rely on a single classification event; it incorporates repeated behavior into the response decision.

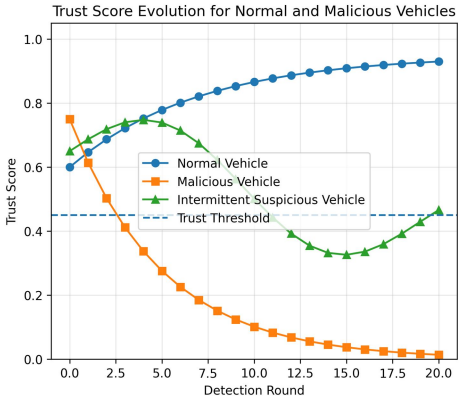

Figure 14: Trust-Score Evolution for Normal and Malicious Vehicles

Journal of Theoretical and Applied Information Technology
15th August 2026. Vol.104. No.15

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

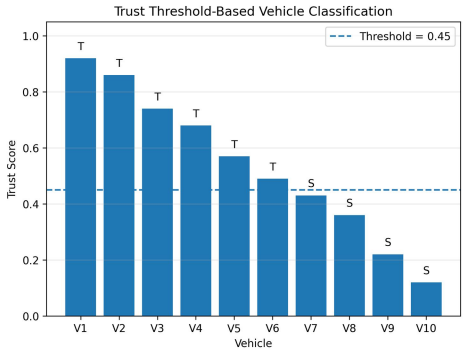

Figure 15: Trust-Threshold-Based Vehicle Classification ($\gamma = 0.45$)

5.8 Latency and Real-Time Feasibility

Two latency totals are reported and are now explicitly distinguished. The 51.0 ms value is the model-to-confirmation time obtained by adding CNN-LSTM inference (12.6 ms) and transaction confirmation (38.4 ms). The complete end-to-end path is 73.4 ms because it additionally includes preprocessing and feature extraction (6.8 ms), transaction creation (3.9 ms), and smart-contract execution (11.7 ms). The complete total, rather than 51.0 ms, is used when assessing the full operational pipeline.

Table 7: Latency Breakdown of the Proposed Framework

Processing Stage	Average Latency
Preprocessing and feature extraction	6.8 ms
Deep-learning inference	12.6 ms
Blockchain transaction creation	3.9 ms
Smart-contract execution	11.7 ms
Transaction confirmation	38.4 ms
Complete end-to-end response	73.4 ms

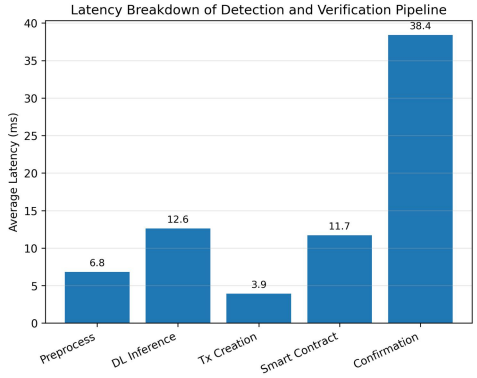

Figure 16: Latency Breakdown of the Proposed Detection and Verification Pipeline

Transaction confirmation is the largest latency component. Consequently, consensus optimization, batching, and edge-assisted verification are the highest-priority directions for reducing response time.

5.9 Ablation Study

Feature engineering improves performance relative to raw features, temporal CNN-LSTM learning provides a further gain, and the complete framework adds verified evidence and adaptive response. The results support the design choice to combine behavioral features, sequence learning, and blockchain-based decision handling.

Table 8: Ablation Study of the Proposed Framework Components

Configuration	Accuracy	Precision	Recall	F1-score	FPR
Raw features + SVM	91.64%	90.87%	91.20%	91.03%	6.82%
Engineered features + RF	93.28%	92.91%	93.06%	92.98%	5.41%
Engineered features + CNN-LSTM	97.61%	97.22%	97.84%	97.53%	2.05%
Complete proposed framework	98.42%	98.11%	98.73%	98.42%	1.39%

5.10 Objective and Hypothesis Evaluation

Table 9 connects the reported evidence to each stated objective. H1 is supported in the evaluated configuration because the proposed model has the highest F1-score and lowest FPR among all listed baselines. H2 is supported conditionally: blockchain provides hash-verifiable evidence, compact storage, and trust-based response, while the complete measured path is 73.4 ms. This result establishes feasibility for the evaluated setup, not a universal latency guarantee for every VCN or consensus mechanism.

Table 9: Achievement of Research Objectives and Hypotheses

Item	Evidence	Assessment
O1 - system and attack models	Equations (10)-(16) define entities, messages, and injected manipulations.	Achieved
O2 - CNN-LSTM detector	98.42% F1-score; AUC 0.992; AP 0.985.	Achieved in the reported test configuration
O3 - lightweight blockchain evidence	0.82 KB average transaction; 8.2 MB for 10,000 records.	Achieved
O4 - trust-based response	Normal/malicious trust trajectories and $\gamma = 0.45$ classification.	Achieved in simulation
O5 - multi-criteria evaluation	Metrics, curves, baselines, related-work comparison, latency, storage, trust, and ablation.	Achieved
H1	Highest listed F1-score and lowest listed FPR.	Supported for evaluated baselines
H2	Hash verification, smart-contract response, and 73.4 ms end-to-end latency.	Conditionally supported for evaluated setup

6. LIMITATIONS, ASSUMPTIONS, AND THREATS TO VALIDITY

Controlled data and external validity. The evaluation uses a controlled VCN/FDI configuration rather than a large, independently collected real-road dataset. Performance may change with different mobility, traffic density, sensor noise, communication loss, and regional driving patterns.

Attack coverage. The classifier is binary and focuses on manipulated speed, location, acceleration, event, and packet behavior. Coordinated attacks, Sybil behavior, adversarial examples, model poisoning, privacy attacks, and previously unseen attack families were not separately evaluated.

Implementation disclosure. The logical methodology, equations, algorithms, test-set size, epoch count, trust threshold, and reported metrics are documented. Exact replication also requires the original seed, split indices, layer widths, optimizer, learning rate, batch size, software versions, hardware, and blockchain consensus settings. These values should accompany the final code package.

Statistical uncertainty. The paper reports point estimates and learning curves but not repeated-seed confidence intervals or statistical significance tests. Therefore, small differences - especially the 0.89 percentage-point F1 gain over the CNN-LSTM configuration - should not be interpreted as proof of universal superiority.

Blockchain assumptions. Hash integrity and ledger immutability are evaluated under an assumed functioning blockchain. Consensus faults, validator compromise, network partition, key management, privacy leakage from metadata, and smart-contract vulnerabilities require separate security analysis.

Latency and scalability. The 73.4 ms result applies to the evaluated transaction load and configuration. Larger validator sets, congested networks, or different consensus protocols may increase confirmation delay.

Truth versus integrity. Blockchain verifies that recorded evidence has not been altered; it does not prove that a source measurement was true. Detection quality and sensor provenance remain essential.

These limitations explain the boundary of the claims and reconcile the principal conflicting perspectives in the literature: stronger models and distributed verification can improve security, but

they introduce complexity, latency, interpretability, privacy, and deployment risks.

7. CONCLUSION AND FUTURE RESEARCH

This paper presented a blockchain-enabled CNN-LSTM framework for real-time FDI detection in Vehicular Cloud Networks. The framework connects engineered mobility and communication features, temporal classification, hash-only blockchain evidence, smart-contract verification, trust-score updating, and security response. The reported model achieved 98.42% accuracy, 98.73% recall, a 98.42% F1-score, and a 1.39% FPR; it also achieved an ROC-AUC of 0.992 and average precision of 0.985. The complete operational path required 73.4 ms, and 10,000 compact blockchain records required 8.2 MB.

In relation to the hypotheses, H1 is supported for the evaluated baselines because the proposed model produced the highest F1-score and lowest FPR. H2 is conditionally supported because hash-based evidence and trust updates were provided with measurable and acceptable overhead in the evaluated configuration. These findings answer the problem statement by showing that detection, evidence preservation, and adaptive response can be combined in one measurable VCN security pipeline, while the limitations prevent over-generalization.

Future work should validate the framework with public and real-world vehicular datasets; release code, seeds, and full implementation metadata; evaluate repeated runs with confidence intervals; test multi-class, coordinated, and adversarial attacks; compare consensus mechanisms and validator configurations; apply transaction batching and off-chain channels; examine privacy-preserving identities and federated learning; and add explainability and formal smart-contract verification.

8. DECLARATIONS

Ethical considerations: The reported study is based on simulated/controlled vehicular and network data and does not involve human participants, animals, clinical records, or personally identifiable information.

Funding: No specific external grant was reported in the materials supplied for this manuscript.

Conflict of interest: No conflict-of-interest statement was present in the supplied manuscript materials. The author must confirm and replace this sentence with the applicable declaration before submission.

Data and code availability: The data-generation rules and evaluation procedure are described in the paper. The final reproducibility package should include the implementation code, original random seed, split indices, model hyperparameters, and blockchain configuration used to generate the reported results.

AI-assisted language editing: Generative AI tools were used to support language editing, organization, and formatting. The author reviewed the technical content, numerical results, citations, and final manuscript and remains fully responsible for the work.

REFERENCES

[1] M. Whaiduzzaman, M. Sookhak, A. Gani, and R. Buyya, "A survey on vehicular cloud computing," Journal of Network and Computer Applications, Vol. 40, pp. 325-344, 2014.

[2] B. Lampe and W. Meng, "A survey of deep learning-based intrusion detection in automotive applications," Expert Systems with Applications, Vol. 221, Article 119771, 2023.

[3] Y. Zhang, C. Cheong, S. Li, Y. Cao, X. Zhang, and D. Liu, "False message detection in Internet of Vehicle through machine learning and vehicle consensus," Information Processing and Management, Vol. 61, No. 6, Article 103827, 2024.

[4] C. Zhao, J. S. Gill, P. Pisu, and G. Comert, "Detection of False Data Injection Attack in Connected and Automated Vehicles via Cloud-Based Sandboxing," IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 7, pp. 9078-9088, 2022.

[5] H. A. Idris, K. Ueda, B. Mokhtar, and S. A. Elsagheer Mohamed, "Machine Learning Based Misbehavior Detection System for False Data Injection Attack in Internet of Vehicles Using Neighbor Public Transport Vehicle Approach," International Journal of Computer Networks and Applications, Vol. 11, No. 2, pp. 159-176, 2024.

[6] H. Wang, J. Yang, J. Sun, Z. Wang, Q. Liu, and S. Luo, "FedIFD: Identifying False Data Injection Attacks in Internet of Vehicles Based on Federated Learning," Big Data and Cognitive Computing, Vol. 9, No. 10, Article 246, 2025.

[7] A. Uprety, D. B. Rawat, and J. Li, "Privacy Preserving Misbehavior Detection in IoV Using Federated Machine Learning," in Proc. 2021 IEEE 18th Annual Consumer Communications

and Networking Conference (CCNC), pp. 1-6, 2021.

[8] H. M. Song, J. Woo, and H. K. Kim, "In-vehicle network intrusion detection using deep convolutional neural network," Vehicular Communications, Vol. 21, Article 100198, 2020.

[9] B. Karthiga, D. Durairaj, N. Nawaz, T. K. Venkatasamy, G. Ramasamy, and A. Hariharasudan, "Intelligent Intrusion Detection System for VANET Using Machine Learning and Deep Learning Approaches," Wireless Communications and Mobile Computing, Vol. 2022, Article 5069104, 2022.

[10] H. Liu, S. Zhang, P. Zhang, X. Zhou, X. Shao, G. Pu, and Y. Zhang, "Blockchain and Federated Learning for Collaborative Intrusion Detection in Vehicular Edge Computing," IEEE Transactions on Vehicular Technology, Vol. 70, No. 6, pp. 6073-6084, 2021.

[11] Z. Abou El Houda, H. Moudoud, B. Brik, and L. Khoukhi, "Blockchain-Enabled Federated Learning for Enhanced Collaborative Intrusion Detection in Vehicular Edge Computing," IEEE Transactions on Intelligent Transportation Systems, Vol. 25, No. 7, pp. 7661-7672, 2024.

[12] M. Abdel-Basset, N. Moustafa, H. Hawash, I. Razzak, K. M. Sallam, and O. M. Elkomy, "Federated Intrusion Detection in Blockchain-Based Smart Transportation Systems," IEEE Transactions on Intelligent Transportation Systems, Vol. 23, No. 3, pp. 2523-2537, 2022.

[13] F. Al-Quayed, N. Tariq, M. Humayun, F. Aslam Khan, M. Attique Khan, and T. S. Alnusairi, "Securing the Road Ahead: A Survey on Internet of Vehicles Security Powered by a Conceptual Blockchain-Based Intrusion Detection System for Smart Cities," Transactions on Emerging Telecommunications Technologies, Vol. 36, No. 4, Article e70133, 2025.

[14] F. Luo, J. Wang, X. Zhang, Y. Jiang, Z. Li, and C. Luo, "In-vehicle network intrusion detection systems: A systematic survey of deep learning-based approaches," PeerJ Computer Science, Vol. 9, Article e1648, 2023.

[15] H. Yang and M. Effatparvar, "A deep learning based intrusion detection system for CAN vehicle based on combination of triple attention mechanism and GGO algorithm," Scientific Reports, Vol. 15, Article 19462, 2025.

[16] H. Bangui, M. Ge, and B. Buhnova, "A hybrid machine learning model for intrusion detection in VANET," Computing, Vol. 104, pp. 503-531, 2022.

[17] C. Zhao, G. Comert, and P. Pisu, "Secure Connected and Automated Vehicles against False Data Injection Attack using Cloud-based Data Fusion," IFAC-PapersOnLine, Vol. 54, No. 20, pp. 638-643, 2021.

[18] P. Lv, L. Xie, J. Xu, X. Wu, and T. Li, "Misbehavior Detection in Vehicular Ad Hoc Networks Based on Privacy-Preserving Federated Learning and Blockchain," IEEE Transactions on Network and Service Management, Vol. 19, No. 4, pp. 3936-3948, 2022.

[19] Y.-T. Yang, L.-D. Chou, C.-W. Tseng, F.-H. Tseng, and C.-C. Liu, "Blockchain-Based Traffic Event Validation and Trust Verification for VANETs," IEEE Access, Vol. 7, pp. 30868-30877, 2019.

[20] T. D. Nguyen, P. Rieger, M. Miettinen, and A.-R. Sadeghi, "Poisoning Attacks on Federated Learning-Based IoT Intrusion Detection System," in Proc. Workshop on Decentralized IoT Systems and Security (DISS), pp. 1-7, 2020.