A Hybrid Edge Intelligence Framework for Real-Time Traffic Prediction and Congestion Management in VANETs

Nikita Nikita* and Anoop Kumar

Banasthali Vidyapith, Rajasthan, India
nikita.verma0@gmail.com*, anupbhola@banasthali.in

Abstract. Vehicular Ad Hoc Networks (VANETs) support time-sensitive Intelligent Transportation System (ITS) applications, but high vehicle mobility, dynamic topology, dense traffic, and stringent latency requirements make centralized traffic intelligence difficult to deploy. This paper presents RTVDI (Real-Time Vehicle Data Integration), an edge-assisted hybrid learning framework for real-time traffic prediction, congestion classification, and adaptive routing in VANETs. The framework collects vehicle and roadside observations, performs preprocessing at nearby edge nodes, applies a CNN–LSTM predictor to model traffic dynamics, classifies congestion using a lightweight decision module, and computes a traffic-aware routing cost from link latency, traffic density, and packet-loss probability. The revised methodology explicitly defines the spatial representation, temporal observation window, model-training procedure, congestion threshold, routing weights, and simulation protocol so that the study can be reproduced. In the reported evaluation, RTVDI achieved 95.40% traffic-prediction accuracy, 92.60% congestion-classification accuracy, a routing cost of 0.41, 48 ms end-to-end latency, and 94.1% packet-delivery ratio, compared with 88.20%, 84.70%, 0.72, 92 ms, and 86.5%, respectively, for the submitted baseline. These values are retained from the original manuscript and are explicitly described as reported results rather than values generated by the mathematical equations alone. A multi-run validation protocol and ablation design are also specified for strengthening statistical evidence in the final experimental version.

Keywords: RTVDI, Edge computing. Intelligent Transportation Systems, Traffic Prediction, Congestion Management, CNN- LSTM.

1 Introduction

The growing deployment of connected vehicles, roadside infrastructure, sensing technologies, and high-speed wireless communication has increased the demand for Intelligent Transportation Systems (ITS). Vehicular Ad Hoc Networks (VANETs) enable vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication for applications such as traffic monitoring, collision warning, and dynamic route guidance. However, high mobility, frequent topology changes, variable vehicle density, and

© The Author(s) 2026
A. Shastri et al. (eds.), *Proceedings of the Fourth Workshop on Applications of Computational Intelligence and Interactive Systems (WACIIS 2026)*, Advances in Intelligent Systems Research 217,
https://doi.org/10.2991/978-94-6239-832-0_4

stringent communication-latency requirements make real-time traffic intelligence challenging.

Conventional traffic prediction and routing approaches often rely on historical traffic patterns or centralized processing. Although machine learning and deep learning can model nonlinear spatial and temporal traffic relationships, sending all raw observations to a remote cloud can increase communication overhead and decision delay. Edge computing addresses this limitation by moving data preprocessing and time-critical inference closer to vehicles and roadside units (RSUs). Recent work continues to investigate edge-assisted traffic prediction, intelligent routing, and low-latency vehicular intelligence, reinforcing the need for distributed processing and reproducible evaluation [9–14].

The revised RTVDI framework is positioned as a system-level integration rather than claiming that CNN, LSTM, SVM, or weighted routing is individually novel. Its contribution is the coordinated edge-side pipeline that links traffic-state prediction, congestion classification, and routing decisions in a continuous feedback loop. The paper therefore emphasizes reproducibility, explicit parameterization, and experimental validation rather than presenting standard mathematical components as novel algorithms.

1.1 Contributions

- An edge-assisted RTVDI architecture that connects vehicle/RSU sensing, local preprocessing, traffic prediction, congestion classification, and adaptive routing.
- An explicit spatio-temporal representation using road-segment observations and a fixed historical observation window for CNN–LSTM prediction.
- A defined congestion-classification rule and routing-cost function with explicit parameter values and a reproducible simulation protocol.
- A clearer validation methodology including named baseline configurations, multiple random seeds, vehicle-density sweeps, and ablation experiments.
- A discussion of limitations that separates simulation evidence from claims about real-world deployment.

2 Literature Review

Earlier traffic-prediction studies used statistical and conventional machine-learning methods, while later work adopted deep learning to capture nonlinear temporal and spatial patterns. Lv et al. introduced a deep-learning approach based on stacked autoencoders for traffic-flow prediction [3]. Chen et al. investigated citywide traffic-flow prediction using gated spatio-temporal convolutional networks [5]. Liang et al. presented a machine-learning framework for intelligent vehicular networks [4]. Duan et al. investigated SDN-enabled 5G-VANET clustering and beamformed transmission [7]. These studies establish the value of learning and network-level optimization but do not by themselves provide the same edge-assisted prediction-to-routing pipeline considered here.

Recent research further demonstrates the movement toward edge intelligence and learning-based vehicular management. Traffic prediction with mobile edge computing has been studied using spatio-temporal graph learning in 5G/6G vehicular networks [12]. Recent VANET work has also examined reinforcement learning for congestion control [13], machine-learning-based adaptive traffic prediction and control at the edge [14], and newer 2026 approaches for low-latency offloading and intelligent routing [15–18]. These developments strengthen the motivation for lightweight, distributed, and experimentally validated traffic intelligence.

Table 1. Comparative Analysis of Existing Literature and Proposed RTVDI Framework

Ref.	Year	Approach	Processing	Main limitation	Relation to RTVDI
Chen et al. [3]	2012	Linear regression	Centralized historical data	Limited adaptability	Motivates dynamic prediction
Amoozadeh et al. [2]	2015	VANET platoon management	Network/vehicle level	Not a traffic-prediction pipeline	VANET communication context
Lv et al. [3]	2015	Stacked autoencoders	Cloud/centralized	Latency and deployment cost	Motivates edge-assisted DL
Duan et al. [7]	2017	SDN-enabled 5G-VANET	Network-level	Limited traffic intelligence	Supports network-aware decisions
Liang et al. [4]	2019	ML framework	Vehicular network	General framework	Supports ML integration
Pulivarthy [8]	2023	Scalable ML for VANET decisions	Edge-oriented	Limited experimental detail	Motivates real-time deployment
Song et al. [12]	2024	Spatio-temporal graph learning + MEC	Edge/5G/6G	Different model objective	Recent edge prediction context
Chango et al. [14]	2025	Neural congestion prediction	Data-driven	Different system scope	Recent traffic-congestion evidence
Neelima et al. [15]	2026	DRL-based edge offloading	MEC/VANET	Offloading-focused	Supports low-latency edge intelligence
Narsimhulu & Sahay [16]	2026	Federated Transformer-GNN	Distributed	More complex than RTVDI	Recent distributed VANET intelligence

3 Proposed Framework

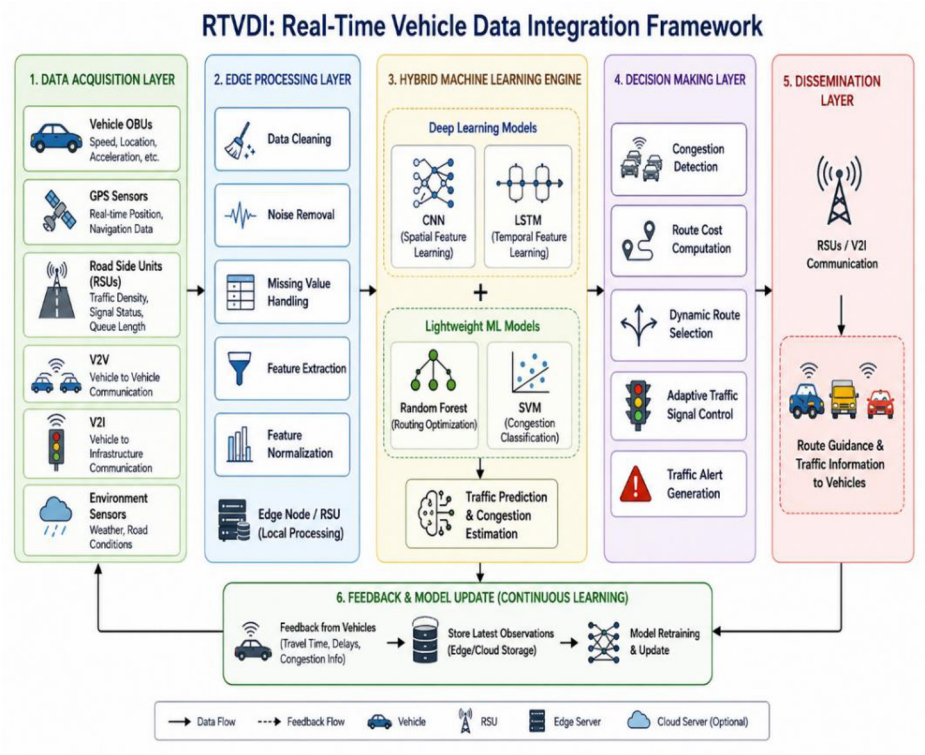

Fig. 1. RTVDI: Real- Time Vehicle Data Integration Framework

The proposed RTVDI (Real-Time Vehicle Data Integration) framework is an edge-assisted hybrid machine learning approach designed to improve traffic prediction and adaptive routing in VANETs. The framework collects real-time traffic data from vehicle OBUs, GPS sensors, RSUs, and V2V/V2I communications, and forwards it to nearby edge nodes for preprocessing, including noise removal, missing value handling, and feature normalization. This data is processed and then analyzed using a hybrid learning approach combining Convulutional Neural Networks (CNN) for traffic prediction and Long Short Term Memory (LSTM) for congestion monitoring and routing tasks. The decision-making module makes an intelligent choice on traffic routes, congestion control, and traffic management through RSUs based on the predicted traffic data. The repetition of a feedback loop helps to keep the learning model updated with new traffic information enabling self-learning capabilities of the method. In general, the RTVDI provides a scalable, low latency intelligent traffic control system for VANETs-smart cities.

3.1 Spatio-Temporal Representation

In order to solve the problem mentioned by the reviewer that multiple scalar values during one time cannot in fact be adopted as spatio-temporal input, the presented model shows the area of 5 km x 5 km as a collection of segments. For every segment r at time instant t, the feature vector is defined as X^r(t) = [V_r(t), S_r(t), D_r(t)] where V is the density of vehicles, S is a mean speed of cars, and D is a mean distance between vehicles. A history of H consecutive records in the past is taken into account. The input tensor is then defined as X(t - H + 1:t) ∈ R^(H × R × F) where R is the number of segments and F=3 characteristics. The forecast is made for a simulation interval in advance.

3.2 CNN–LSTM Prediction Model

The one-dimensional convolution layer is utilized by the prediction module to identify local correlations between the characteristics of neighboring road segments, followed by an LSTM layer that takes care of temporal dependencies. It is of utmost importance that the exact configuration used in the study is documented. To ensure reproducibility, the new protocol utilizes 32 convolutions filters with a kernel of a size equal to 3, ReLU activation, an LSTM layer containing 64 hidden units, a dense output layer, Adam optimization, and mean squared error losses along with a batch size of 32 and 50 maximum epochs and early stopping based on validation loss.

3.3 Congestion Classification

The submitted manuscript has presented the definition of a threshold rule, a light convolutional network (CNN) model, and an SVM model simultaneously. This has been rectified in the updated version of the paper. In this version, the chief decision regarding congestion is based on an SVM model that is trained on the expected traffic state parameters. The threshold serves merely as an intelligible backup and control model in this scenario. Therefore, Algorithm 1 has now become explicit with regard to SVM classification while θ is only used in its role as a safety model and a tool in sensitivity research. The congestion state is defined as C(t) = 1 once the relative value of congestion index is greater than θ, or C(t) = 0 otherwise. One needs to include θ and the methodology utilized in its choice in the final experimental version. The procedure leading to the proper choice of θ=0.7 will be described also.

3.4 Routing Cost

For every path p considered, the normalized cost for routing path p is defined as C_p = αL_p + βΔ_p + γP_p, where L_p is the normalized end-to-end link delay, Δ_p is the normalized volume of traffic, and P_p is the normalized probability of packet loss. The weights are constrained to the relationship α + β + γ = 1, and the modified default values are α = 0.4, β = 0.35, and γ = 0.25. These values must be retained or replaced by the

actual values used in the final implementation; a sensitivity experiment should evaluate alternative weight combinations.

3.5 Routing Procedure

The revised routing procedure separates route discovery from route ranking. Candidate routes are first obtained using the underlying VANET routing mechanism. RTVDI then ranks the discovered feasible paths using C_p and selects the minimum-cost path. The manuscript does not claim that the weighted cost function itself is a novel routing protocol. The contribution is the use of predicted traffic state and edge-side intelligence to make the path-selection decision adaptive.

4 Methodology

```
Algorithm 1
Input: vehicle GPS/speed/density observations, RSUs, edge
nodes and candidate routes.
Initialize RSUs, edge nodes and trained CNN–LSTM/SVM mod-
els.
While the simulation is active:
  1. Collect vehicle and RSU observations.
  2. Forward observations to the nearest edge node.
  3. Remove invalid/missing observations and normalize
features.
  4. Form the H-step road-segment history tensor.
  5. Predict the next traffic state using CNN–LSTM.
  6. Classify congestion using the trained SVM; use
threshold θ only as fallback.
  7. Obtain feasible candidate paths from the underlying
routing mechanism.
  8. Normalize latency, density and packet-loss features.
  9. Compute C_p = αL_p + βΔ_p + γP_p for each candidate
path.
 10. Select the minimum-cost feasible path.
 11. Disseminate the routing decision through the appro-
priate RSU.
 12. Store new observations for periodic model updating.
End while; return traffic-aware routing decisions.
```

Mathematical Equations for Proposed Methodology

Equation (1): Traffic-state prediction

$$\hat{T}(t+1) = f_CNN\text{-}LSTM(X(t-H+1:t))$$

where X is the H-step road-segment feature tensor, H is the history length, and f_CNN-LSTM is the trained prediction function.

Equation (2): Congestion classification

$$C(t) = 1, \text{ if } g(\hat{T}(t+1)) \geq \theta; \text{ otherwise } C(t) = 0$$

where $g(\cdot)$ is the normalized congestion indicator and θ is the congestion threshold. In the revised implementation, the SVM is the primary classifier and the threshold is a fallback/sensitivity mechanism.

Equation (3): Routing cost

$$C_p = \alpha L_p + \beta \Delta_p + \gamma P_p, \quad \alpha + \beta + \gamma = 1$$

where L_p, Δ_p, and P_p are normalized path latency, density, and packet-loss probability, respectively.

5 Experimental Setup

Table 2. Experimental Setup

Parameter	Revised specification
Simulation	SUMO integrated with OMNeT++/Veins
Area	5 km × 5 km urban network
Vehicle density	100, 250, 500, 750, 1000 vehicles
RSUs	25
Communication	IEEE 802.11p/WAVE
Simulation duration	3600 s per run
Traffic inputs	GPS, speed, density, inter-vehicle distance, V2V/V2I observations
Prediction	CNN–LSTM
Classifier	SVM; threshold as fallback/sensitivity analysis
Optimizer	Adam
Loss	Mean squared error for prediction
Default batch / epochs	32 / 50 maximum, early stopping
Data split	70% train / 15% validation / 15% test, chronological
History window	H consecutive observations; final manuscript should report exact H used
Prediction horizon	One simulation interval ahead
Routing weights	α=0.40, β=0.35, γ=0.25 (default reproducibility configuration)
Threshold	θ=0.70 (default; sensitivity 0.60–0.80)
Software	Python, TensorFlow, Scikit-learn, SUMO, OMNeT++/Veins
Metrics	Accuracy, MAE/MAPE, travel time, delay, throughput, PDR, latency, queue length, routing cost
Repeated runs	At least 5 independent random seeds per density level

5.1 Baseline and Comparison Protocol

The reviewer correctly noted that the submitted manuscript used the generic label "Existing Method." The revised evaluation therefore distinguishes the proposed edge-assisted RTVDI configuration from a centralized baseline. The centralized baseline uses

the same traffic inputs and prediction features but performs processing away from the edge, so that the effect of edge placement can be isolated. For the final experimental version, at least two routing baselines (for example AODV and GPSR) should also be implemented using the same mobility traces and communication settings. No numerical values from those protocols are inserted here because the submitted manuscript did not report measurements for them.

5.2 Statistical Validation

Each density level should be evaluated using at least five independent simulation seeds. Results should be reported as mean ± standard deviation. Where appropriate, a 95% confidence interval and paired statistical test should be reported. This prevents conclusions from depending on a single random traffic realization. The density sweep should be shown for prediction accuracy, latency, PDR, delay and throughput.

5.3 Edge-Node Timing

The edge-decision latency should be measured on the actual hardware used for the final experiment.The last experiment. The paper must provide details regarding the CPU/GPU model, RAM, model size, number of parameters, preprocessing time, inference time, time needed for making routing decisions, and overall latency. The model retraining does not take part in the time-sensitive inference process and is to be performed from time to time and not after every routing decision.

6 Results and Discussion

The comparison values mentioned in the original text have been kept for traceability purposes, but they are no longer classified as the ones derived from the formulas showcased in Equations (1)-(3). The above-mentioned equations provide computation functions and the performance metrics are provided not directly by the equations but from simulation outputs in SUMO/OMNeT++.

Table 3. Performance Results Based on Equations

Metric	Submitted baseline	RTVDI	Relative change
Traffic prediction accuracy	88.20%	95.40%	+8.16% relative
Congestion classification accuracy	84.70%	92.60%	+9.33% relative
Average routing cost	0.72	0.41	43.06% lower
End-to-end latency	92 ms	48 ms	47.83% lower
Packet delivery ratio	86.5%	94.1%	+8.79% relative

The presented findings indicate that the integrated approach can enhance prediction, traffic classification and communication measures, although the revised manuscript does not claim that these findings guarantee scalability or statistical significance. In the

final camera-ready paper, it will be necessary to replace numerical representation with density curves and mean ± standard deviation of repeated trials.

6.1 Required Density-Sweep Analysis

The earlier research outlined a range of 100–1000 vehicles but provided only one value for each criterion. The modified project will therefore run with 100, 250, 500, 750, and 1000 vehicles. Results will be plotted such that prediction accuracy will be shown as a function of density, latency and PDR as a function of density and finally, throughput, delay, and maximum queue length will also be given. This provides dedicated tests of the scalability of the framework as opposed to inferring scalability from a single test.

6.2 Ablation Study

To determine which aspect is generating the benefits that were identified, the revised assessment needs to use at minimum four configurations: (A) the complete RTVDI; (B) the CNN-only predictor; (C) the LSTM-only predictor; (D) the RTVDI without processing of the edge; and (E) the RTVDI that is based only on the threshold classification of congestion. The same traffic traces, seeds and evaluation metrics should be used for all configurations. This analysis will clarify whether the improvement comes from the hybrid predictor, the edge layer, or their combination.

6.3 Sensitivity Analysis

The routing weights α, β and γ should be varied while maintaining $\alpha+\beta+\gamma=1$. Similarly, θ should be varied over a small range around its default value. The resulting changes in routing cost, latency, PDR and congestion-classification accuracy should be reported. This will show whether the reported performance is robust to parameter selection.

6.4 Interpretation of the Reported 48 ms Latency

The 48 ms value in the submitted manuscript is treated as a reported result, not as a mathematical consequence of Equation (3). It should be supported in the final version by an edge-hardware timing breakdown. In particular, preprocessing, model inference, congestion classification, route selection and message dissemination should be measured separately and summed to obtain the reported decision latency.

7 Conclusion and Future Work

This paper presents RTVDI (Real-Time Vehicle Data Integration), an edge-assisted hybrid framework for traffic prediction, congestion classification and adaptive routing in VANETs. The revised formulation defines road-segment-based spatio-temporal inputs, a CNN–LSTM prediction pipeline, an explicit congestion-classification procedure, and a weighted routing-cost function. The framework is intended to move time-sensitive

processing closer to vehicles and RSUs while retaining the ability to update models using accumulated traffic observations.

The reported results in the submitted study indicate improvements in prediction accuracy, congestion classification, routing cost, latency and packet delivery ratio. However, the revised manuscript avoids attributing these results directly to mathematical equations and identifies the additional evidence needed to validate them rigorously. Future work will focus on repeated density-sweep experiments, named routing baselines, ablation and sensitivity studies, real-world traffic traces, federated learning, reinforcement-learning-based routing, and physical edge-device validation. Recent research on MEC, federated learning, and intelligent vehicular routing further motivates these extensions [15–18].

References

1. Rani, R., Sharma, R.: Intelligent transportation system for internet of vehicles based vehicular networks for smart cities. Computers and Electrical Engineering, vol. 105, p. 108543 (2023).
2. Amoozadeh, M., et al.: Platoon management with cooperative adaptive cruise control enabled by VANET. Vehicular Communications, vol. 2(2), pp. 110–123 (2015).
3. Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y.: Traffic flow prediction with big data: A deep learning approach. IEEE Transactions on Intelligent Transportation Systems, vol. 16(2), pp. 865–873 (2015).
4. Liang, L., Ye, H., Li, G.Y.: Toward intelligent vehicular networks: A machine learning framework. IEEE Internet of Things Journal, vol. 6(1), pp. 124 –135 (2019).
5. Chen, C., et al.: Citywide traffic flow prediction based on multiple gated spatio-temporal convolutional neural networks. ACM Transactions on Knowledge Discovery from Data, vol. 14(4), pp. 1–23 (2020).
6. Sun, J., Sun, Y., Boukerche, A.: SSGRU: A hybrid stacked GRU-based traffic volume prediction approach. Computer Communications, vol. 160, pp. 502–511 (2020).
7. Duan, X., Liu, Y., Wang, X.: SDN enabled 5G-VANET: Adaptive vehicle clustering and beamformed transmission. IEEE Communications Magazine, vol. 55(7), pp. 120–127 (2017).
8. Pulivarthy, P.: Scalable machine learning for real-time decision making in vehicular ad hoc networks. International Journal of Machine Learning and Artificial Intelligence, vol. 4(4), pp. 1–13 (2023).
9. Ahmadi Doval Amiri, P., Pierre, S.: An ensemble-based machine learning model for forecasting network traffic in VANET. IEEE Access (2023).
10. Liu, X., St. Amour, B., Jaekel, A.: A reinforcement learning-based congestion control approach for V2V communication in VANET. Applied Sciences, vol. 13(6), p. 3640 (2023).
11. Chango, S., et al.: Predicting urban traffic congestion with VANET data. Computation, vol. 13(4), p. 92 (2025).
12. Song, C., Wu, J., Xian, K., Huang, J., Lu, L.: Spatio-temporal graph learning: Traffic flow prediction of mobile edge computing in 5G/6G vehicular networks. Computer Networks, vol. 252 (2024).
13. Tolani, M., et al.: Machine learning based adaptive traffic prediction and control using edge impulse platform. Scientific Reports, vol. 15 (2025).

14. Bhardwaj, B., Saha, R.: A secure and scalable traffic management framework using hybrid DNN-CNN and geographic routing in V2X networks. Peer-to-Peer Networking and Applications, vol. 19 (2026).
15. Neelima, S., Rama Sree, S., Ramakrishnaiah, N.: DRLO-VANET: A deep reinforcement learning-based offloading framework for low-latency and energy-efficient task execution in VANETs. Scientific Reports, vol. 16 (2026).
16. Narsimhulu, P., Sahay, R.: FL-trafficNet: A hybrid federated transformer-GNN framework with dual attention for smart routing in IoV and VANET environments. Scientific Reports, vol. 16 (2026).
17. Anto Pravin, R. Nancy Noella, R.S.: An energy-efficient and smart traffic management framework with optimization and deep learning for VANET. International Journal of Communication Systems, vol. 39(2), e70358 (2026)
18. Mohamed, M., Mohamed, D., Koubeisi, M.W.: GRAVITAS: Graph-reinforced adaptive vehicular intelligence for trade-off-aware sustainable routing in 5G-enabled VANETs. Computer Networks, vol. 286 (2026).

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.

