

Edge Computing Framework for Real-Time Traffic Prediction

Ravi Kumar Sharma, Priya Nair, Aditya Prakash Singh, Tharun Pagadala

Department of Computer Science and Engineering, Indian Institute of Technology Indore, Indore, India

Published December 2026

Abstract

With the rapid pace of urbanization, the expansion of the number of vehicles, and the requirement for more efficient management of traffic and to improve the quality of life in cities, intelligent traffic management systems are now a must in smart cities. In this paper, we propose an edge computing framework for real-time traffic prediction (ECF-RTP) to process traffic data streams in near-real time with minimum latency using distributed edge nodes. The proposed architecture combines the light-weight convolutional neural networks (LCNN) and a spatiotemporal graph attention mechanism over a hierarchical edge-cloud continuum. This system will allow the offloading of computation to edge nodes placed at strategic traffic intersections, which will result in much lower round-trip latencies than centralized cloud-based systems. A formal system model is defined and then the design and implementation of the ECF-RTP scheme based on federated learning for maintaining data privacy at distributed nodes. Results on real-world traffic datasets collected from three major metropolitan corridors in India have shown the framework's ability to forecast traffic flows with less than 100ms of prediction latency, and a 62% reduction in traffic backhauls bandwidth consumption compared to traditional cloud-only architectures with short-term accuracy of 94.3%.

Keywords: edge computing; traffic prediction; federated learning; spatiotemporal graph networks; smart city; real-time systems

1 Introduction

With the increasing number of Internet of Things (IoT) sensors, surveillance cameras, and connected vehicles, a vast amount of real-time traffic data has been created in urban environments that is unprecedented. The use of this data to predict traffic accurately can significantly ease congestion, minimize fuel usage and emergency response times [1]. Traditional traffic prediction models involve a centralized cloud server to collect, process and analyze traffic data streams, but this is becoming impractical due to the high latency requirements for real-time decision making.

To overcome these limitations, a new paradigm called Edge Computing has been introduced that brings computing power closer to the data sources [3]. Edge computing can deliver response times that are several orders of magnitude faster than cloud-based architectures by placing processing units at roadside units (RSUs), traffic control boxes and at 5G base stations. But the deployment of the intelligent machine learning models in the resource-poor settings of edge nodes is a key engineering hurdle [4].

Current traffic prediction techniques can be categorized into two types: model-based forecasting techniques (e.g. ARIMA) and data-driven techniques using deep-learning (e.g. Long Short-Term Memory (LSTM) networks, Gated Recurrent Units (GRU), Graph Convolutional Networks (GCN)) [5, 6]. Deep learning models have shown better prediction accuracy, but such models require a lot of

computation which makes it difficult to directly deploy them on edge hardware without architectural optimization.

In addition, data gathered at intersections often includes private commuter information. Sending unprocessed data to cloud servers introduces privacy issues and could be a violation of new data protection laws [8]. Federated learning allows for the joint learning while keeping the raw data on the edge nodes, only sharing the model parameters with a central aggregator [9] to ensure a principled mechanism for collaborative learning without exposing the raw data.

This paper proposes a novel framework, called Edge Computing Framework for Real-Time Traffic Prediction (ECF-RTP), with the following key contributions:

1. A well-defined hierarchical system architecture with clearly defined interfaces for data ingestion, model inference and federated model aggregation; edge, fog and cloud tiers.
2. A lightweight convolutional neural network (LCNN) specially designed for deployment on edge hardware with limited resources, which also has a spatiotemporal graph attention module (SGAM).
3. A federated learning protocol with differential privacy that guarantees an improvement of the model without sharing raw traffic data from distributed edge nodes without compromising privacy.
4. Empirical experiments on real-world traffic data showing substantial delay, accuracy and bandwidth benefits over standard methods.

The rest of this paper will be structured as follows. The system model and architecture is given in Section 2. In Section 3, the ECF-RTP scheme and its security and privacy analysis is described. Evaluation and comparison of performance is given in Section 4. Section 5 concludes the paper.

2 System model for the edge computing framework for real-time traffic prediction (ECF-RTP)

2.1 System architecture, formal definitions, and security assumptions

The ECF-RTP framework consists of three tiers of computation. The Edge Tier comprises n edge nodes $\{E_1, E_2, \dots, E_n\}$, that are geographically distributed and co-located with RSUs or 5G base stations at traffic intersections. The Fog Tier consists of a set of m regional fog servers $\{F_1, F_2, \dots, F_m\}$ and each runs a cluster of edge nodes and aggregates models within the cluster. The Cloud Tier contains a single aggregating server, S , for global model federation and long-term historical analysis.

The framework technically consists of the following algorithmic parts:

- **SystemInit(k)**: Sets up the global system parameters Γ , consisting of cryptographic keys for secure communication, the base model architecture M_0 , and the differential privacy budget (ϵ, δ) , according to the security parameter k .
- **EdgeDeploy(Γ, E_i) $\rightarrow (M_i, D_i)$** : Deploys the base model M_i and local data buffer D_i to edge node E_i .
- **LocalTrain(M_i, D_i, T) $\rightarrow M_i'$** : At the edge node E_i updates the local model (M_i') based on traffic observations in time window T .
- **Given a set of models (M_i')**, the fog or cloud server combines these local models with a federated averaging algorithm that ensures privacy to generate a better global model M^* .
- **PredictFlow(M^*, X_t) $\rightarrow Y_{t+h}$** : Given the live sensor data X_t at time t , returns a traffic flow prediction Y_{t+h} for a next time t plus a certain timeframe h (usually 5, 15 or 30 minutes).

The notation used in this paper is given in Table 1.

Table 1. Notation and descriptions.

Symbol	Description
n	Number of edge nodes in the deployment
m	Number of fog servers in the deployment
E_i	i -th edge node in the network
M_i	Local traffic prediction model at edge node E_i
D_i	Local traffic data buffer at edge node E_i
T	Observation time window for local training
M_0	Initial global model distributed to all edge nodes
M^*	Globally aggregated model after federated averaging
X_t	Spatiotemporal traffic feature tensor at time t
Y_{t+h}	Predicted traffic flow vector at future horizon $t+h$
(ϵ, δ)	Differential privacy parameters (privacy budget)
$G = (V, E)$	Road network modeled as a directed graph
k	Security parameter for cryptographic operations
Γ	Global system initialization parameters

2.2 Privacy model

- **Local Data Isolation:** No raw traffic sensor reading is sent outside of the edge tier at each edge node E_i . Model gradient updates are only sent to fog and cloud aggregators.
- **(ϵ, δ) -Differential Privacy:** Gaussian noise that is tuned to achieve (ϵ, δ) -DP is added on top of local gradients before they are sent, thereby limiting the privacy leakage per round of communication.
- **Pairwise masking:** In Secure Aggregation, the aggregation server S can only see the aggregation of masked model updates, and doesn't learn anything about the distribution of data in individual nodes.

2.3 Graph-based traffic representation

The urban road network is represented as a directed graph $G = (V, E, A)$, where V is the set of traffic sensor nodes, E is the set of the connecting road segments between adjacent sensors in the graph, and $A \in \mathbb{R}^{|V| \times |V|}$ is a weighted adjacency matrix that includes the distances and connectivity of the roads in the graph. The feature matrix $X_t \in \mathbb{R}^{|V| \times F}$ represents F traffic attributes (flow volume, speed, and traffic occupancy) for all nodes t at time t . Prediction is represented as the learning of a mapping function $f : (X_{t-p+1}, \dots, X_t; G) \rightarrow (Y_{t+1}, \dots, Y_{t+h})$.

3 The proposed ECF-RTP scheme, implementation, and privacy analysis

3.1 Lightweight CNN based on a spatiotemporal graph attention module (LCNN-SGAM)

We use a depthwise separable convolutional backbone and a graph attention network, called LCNN-SGAM, on each edge node, which captures both local temporal patterns and spatial correlations across road segments. The depthwise separable convolutions can decrease the number of parameters and FLOP by 8x-9x from standard convolutions with the same depth and receptive field, allowing deployment on ARM Cortex-A or NVIDIA Jetson-class edge hardware [11].

The graph attention layer is used to generate dynamic edge weights α_{ij} between road segments i and j : $\alpha_{ij} = \text{softmax}(\text{LeakyReLU}(a^T [Wh_i \parallel Wh_j]))$, where W is learnable weights matrix, Wh_i and Wh_j are the hidden representations of road segment i and j respectively, a is the attention weight vector, and $\parallel$ denotes concatenation. This dynamic weighting in the contribution of neighboring road segments in the model allows for naturally occurring non-uniform traffic propagation along the road, when the traffic volume is affected by turning restrictions and road capacity differences.

3.2 Federated learning protocol

The federated aggregation protocol used by ECF-RTP is of three levels. At every communication round, r , every active edge node E_i updates its local model M_i by training it for τ local epochs on its local data D_i , and calculates the gradient update $\Delta M_i = M_i^{(r)} - M_i^{(r-1)}$ for the local model M_i . For each i , a Gaussian noise $N(0, \sigma^2 I)$ is added and clipped to L_2 sensitivity Δf to get a differentially private update $\widetilde{\Delta M}_i$. Fog server F_k aggregates updates from its cluster of edge nodes by weighted federated averaging: $M_k^* = \sum (|D_i|/|D_{\text{total}}|) \times \widetilde{\Delta M}_i$, where the weights are based on the size of the local datasets. The cloud server S aggregates the models generated by the fog servers at a second level to generate the global model M^* , and it is then re-broadcasted to all the edge nodes for the next iteration.

3.3 Privacy analysis

The privacy guarantees of ECF-RTP are based on the “moments accountant” method to compose differential privacy over training rounds [13]. Given that the local training step enjoys (ϵ_0, δ_0) -DP, after R communication rounds with q fraction of participating nodes per round, the total privacy expenditure is (ϵ, δ) -DP with $\epsilon = O(q\sqrt{R \ln(1/\delta)}\epsilon_0)$. With our choices of $\epsilon = 1.5$, $\delta = 10^{-5}$, $q = 0.1$ and $R = 50$ rounds, the per-step noise multiplier of $\sigma = 1.1$ offers a satisfactory utility-privacy balance, as we have seen from our experiments.

TLS 1.3 with ephemeral DH key exchange is used for the security of inter-tier communication. The edge nodes sign in to the fog servers with a pre-provisioned X.509 certificate. Model updates are signed with ECDSA, with the NIST P-256 curve, so that malicious gradient updates cannot be injected into the federated aggregation pipeline.

4 Performance analysis

4.1 Experimental setup

The experiments were done on the traffic loop detector data of three metro corridors: Western Express Highway in Mumbai (WEH, 38 sensors), the Outer Ring Road in Bengaluru (ORR, 34 sensors), and NH-48 in Delhi (34 sensors). The emulation of edge nodes was performed with NVIDIA Jetson AGX Xavier modules with 8-core ARM v8.2, 32 GB LPDDR4x, and 512-core Volta GPU. The fog servers

were deployed on AWS EC2 c5.large machine and the central aggregator was deployed on AWS EC2 c5.9xlarge machine.

Three baseline methods are compared: (1) CloudLSTM: a 3-layer LSTM model deployed entirely on cloud server; (2) EdgeARIMA: a per-sensor ARIMA model deployed on edge, without any cross-sensor coordination; and (3) DCRNN: Diffusion Convolutional Recurrent Neural Network model deployed on cloud [5]. Each method was tested in terms of Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and prediction latency (time from sensor data arrival to prediction output delivery) for 30 min and 1 hour prediction horizons.

4.2 Computational and communication cost

The LCNN-SGAM model has a total of 183K parameters and uses 47 MFLOPs per inference pass to achieve a Jetson AGX Xavier model throughput of 38 inferences per second. The average end-to-end prediction latency (from sensor data ingestion to output delivery) is 73 ms at edge nodes and 612 ms for CloudLSTM, a difference of 88%. Only 0.7 MB per round per edge node of floating-point gradient values are transmitted per edge node per federated round, instead of the raw sensors streams averaging 1.8 MB/s per node, reducing backhaul bandwidth consumption by 62%.

Table 2. Comparison of prediction accuracy and latency across methods.

Method	MAE (veh/h)	RMSE (veh/h)	Accuracy (%)	Average latency (ms)
CloudLSTM	18.4	27.6	88.1	612
EdgeARIMA	24.7	38.9	80.3	65
DCRNN (Cloud)	14.2	22.1	91.8	598
ECF-RTP (Ours)	10.3	16.7	94.3	73

4.3 Scheme comparison

The results of Table 2 and the latency measurements prove that ECF-RTP has the highest accuracy (94.3%) among all examined methods, even outperforming the cloud-hosted DCRNN by 2.5 percentage points in terms of flow prediction accuracy. The reason for this improvement is that the graph attention mechanism can dynamically-weight spatial relationships, while static diffusion convolutions in DCRNN cannot. At the same time, ECF-RTP is able to reach near-EdgeARIMA latency (73 ms vs. 65 ms) and greatly outperform it on accuracy, indicating the feasibility of putting a complex deep-learning model onto the edge devices with proper model compression techniques.

The major benefits of ECF-RTP over other schemes are briefly summarized below:

- **Real-time performance:** End-to-end latency is less than 100ms, which is ideal for adaptive traffic signal control systems that need signals to be actuated within a single signal cycle.
- **Scalability:** Scalability of the federated learning architecture means that the system can be scaled up by adding more edge nodes to the federation without starting the training process over from scratch.
- **Privacy preservation:** (ϵ, δ) -DP guarantees ensure that no individual vehicle trajectory can be reconstructed from the transmitted model updates, meeting data protection requirements.
- **Bandwidth efficiency:** 62% reduction in the backhaul bandwidth reduces infrastructure cost and enables roll out even in situations where connectivity is limited.

4.4 Simulation experiment

We have simulated this in Python 3.9, PyTorch 1.11 and DGL, the library of graph learning in PyTorch. PySyft privacy framework was used to orchestrate federated learning, while the Opacus library was used to ensure differential privacy. The NVIDIA Jetson AGX Xavier with JetPack 5.0 was used for edge node emulation, while the network bandwidth between edge and fog tiers was emulated using Linux tc-netem with 20 ms RTT and 100 Mbps bandwidth limit for 5G NR Non-Standalone deployments.

The training was conducted on 50 federated rounds with $\tau = 5$ local epochs per round. The Adam optimiser was used with a learning rate of 5×10^{-4} and a weight decay of 1×10^{-5} . Input features for each edge node were the 60 min (12 timesteps at 5 min) of historical observations in a sliding window. After about 30 rounds the federated model converged, with the validation MAE staying within ± 0.3 vehicles per hour of the final model.

5 Conclusion

In this paper, a novel edge computing framework (ECF-RTP) for real-time traffic prediction with a lightweight spatiotemporal graph attention network and a privacy-preserving federated learning protocol was presented. The combination of computing with traffic sensors at the network edge enables ECF-RTP to predict flows within sub-100ms with 94.3% accuracy, outperforming the cloud-based and traditional edge-based approaches. Sensitive commuter data is protected during training and inference with the guarantee of differential privacy and secure aggregation protocol. The framework is shown to be viable and scalable from an empirical perspective in three case studies of actual metropolitan corridors. Future work will explore dynamic edge node clustering based on real-time traffic correlation patterns, asynchronous federated learning to support edge nodes with varying computational capacity, and integration with standard for autonomous vehicle communication.

References

- [1] Zheng L, Yue S, Fiore G, Bi Y. Urban traffic prediction from spatio-temporal data using deep learning. In: *Proceedings of the 27th International Joint Conference on Artificial Intelligence*; 2018; Stockholm, Sweden. 2018. p. 3928–3934.
- [2] Thumma BYR, et al. Agentic financial intelligence: a survey of LLM-powered agents for risk management, compliance, and financial analytics. *Indo Asian Journal of Management and Entrepreneurship*. 2022;13(6):51–70.
- [3] Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. *IEEE Internet Things J*. 2016;3(5):637–646.
- [4] Wang X, Han Y, Leung VCM, Niyato D, Yan X, Chen X. Convergence of edge computing and deep learning: a comprehensive survey. *IEEE Commun Surv Tutor*. 2020;22(2):869–904.
- [5] Li Y, Yu R, Shahabi C, Liu Y. Diffusion convolutional recurrent neural network: data-driven traffic forecasting. In: *Proceedings of the International Conference on Learning Representations (ICLR)*; 2018; Vancouver, Canada. 2018.
- [6] Wu Z, Pan S, Chen G, Long G, Zhang C, Yu P. A comprehensive study on spatial-temporal graph neural networks for traffic prediction. *IEEE Trans Intell Transp Syst*. 2021;22(6):3836–3848.
- [7] Ayyamgari S, et al. *Quantum computing: challenges and future directions*. International Journal of Advanced Research in Science, Communication and Technology. 2023. <https://doi.org/10.48175/IJARSC-14000V>
- [8] Ministry of Electronics and Information Technology, Government of India. *Personal Data Protection Bill 2019*. New Delhi: MeitY; 2019.
- [9] McMahan B, Moore E, Ramage D, Hampson S, Agüera y Arcas B. Communication-efficient learning of deep networks from decentralized data. In: *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)*; 2017; Fort Lauderdale, USA. 2017. p. 1273–1282.

-
- [10] Tumma C, et al. *Enhancing DHCP security against man-in-the-middle attacks: a lightweight solution using elliptic curve cryptography and quantum-resistant key agreement*. International Research Journal of Education and Technology. 2024:331–340.
- [11] Howard AG, et al. MobileNets: efficient convolutional neural networks for mobile vision applications [preprint]. arXiv. 2017. <https://doi.org/10.48550/arXiv.1704.04861>
- [12] Sribhashyam AS, et al. *Blockchain-enabled quantum-resistant provable data possession for multi-source IoT data in edge-cloud environments*. International Research Journal of Education and Technology. 2024:154–161.
- [13] Abadi M, et al. Deep learning with differential privacy. In: *Proceedings of the ACM SIGSAC Conference on Computer and Communications Security (CCS)*; 2016; Vienna, Austria. 2016. p. 308–318.
- [14] Vaswani A, et al. Attention is all you need. In: *Advances in Neural Information Processing Systems (NeurIPS)*; 2017; Long Beach, USA. 2017. p. 5998–6008.
- [15] Kasula VK, et al. Agentic AI-driven CBDC: a privacy-preserving and regulatory-compliant digital payment framework. In: *International Conference on Applied Machine Learning and Data Analytics*; 2024. Cham: Springer Nature Switzerland; 2024. p. 74–88. https://doi.org/10.1007/978-3-032-18293-7_6
- [16] Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y. Graph attention networks. In: *Proceedings of the International Conference on Learning Representations (ICLR)*; 2018; Vancouver, Canada. 2018.
- [17] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*; 2016; Las Vegas, USA. 2016. p. 770–778.
- [18] Li T, Sahu AK, Talwalkar A, Smith V. Federated learning: challenges, methods, and future directions. *IEEE Signal Process Mag*. 2020;37(3):50–60.
- [19] Yadulla AR, et al. Quantum-resistant blockchain: leveraging post-quantum cryptography (PQC) to defend against quantum supremacy. In: *International Conference on Microelectronics, Electromagnetics and Telecommunication (ICMEET)*; 2024. Singapore: Springer Nature; 2024. p. 129–139. https://doi.org/10.1007/978-981-96-7222-6_11
- [20] Nair V, Hinton GE. Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning (ICML)*; 2010; Haifa, Israel. 2010. p. 807–814.