International Journal of Scientific Research & Engineering Trends
Volume 12, Issue 5, Sep-Oct-2026, ISSN (Online): 2395-566X

Edge AI For Real-Time Prediction in Autonomous Vehicles: A Review of Trajectory Forecasting Models and Onboard Optimization

Bhalerao Onkar Rohidas[1] Prof. M. A. Auti[2] Prof. M. C. Kolse[3] Prof. S. K. Said[4]
Department Of Artificial Intelligence And Data Science
Jaihind College Of Engineering
Pune, Maharashtra, India

Abstract—Autonomous vehicles must predict how nearby vehicles, cyclists and pedestrians will move within the next few seconds in order to plan safe actions. Sending raw sensor data to the cloud for this prediction introduces latency and dependence on network connectivity that can jeopardize safety. This paper reviews Edge AI approaches that perform perception and motion prediction on onboard embedded hardware. It surveys trajectory forecasting models, from social pooling of recurrent networks and generative adversarial prediction to graph-based models that encode high-definition maps and agent interactions, and multi-modal predictors that output several likely futures. Large-scale motion forecasting datasets are discussed. The review then examines techniques for running these models within strict latency and power budgets, including pruning, quantization to integer arithmetic, knowledge distillation and hardware–software co-design on automotive accelerators. A five-layer onboard architecture covering sensing, perception, edge prediction, planning and vehicle control is proposed, with research gaps, objectives and a comparison of prediction models. The review concludes that compact, quantized multi-modal predictors running at the edge can deliver the low-latency, reliable forecasts that safe autonomous driving requires.

Index Terms—Edge AI, autonomous vehicles, trajectory prediction, motion forecasting, model compression, quantization, knowledge distillation, embedded systems

I. INTRODUCTION

A self-driving car travelling at 60 km/h covers more than sixteen metres every second. Within that time it must detect surrounding road users, predict their movements and decide how to respond. Any delay in this loop directly reduces the margin for safe action.

Cloud computing offers abundant processing power but adds network round-trip delays and fails entirely in tunnels or rural areas. Edge computing [1] moves computation close to the data source, and for autonomous vehicles that means onto the vehicle itself. Liu *et al.* [2] analysed the computing requirements of autonomous driving and argued that onboard edge processing is essential for real-time safety.

This paper reviews motion prediction models and the techniques used to run them efficiently on vehicle hardware, and proposes an onboard architecture.

II. BACKGROUND AND FUNDAMENTAL CONCEPTS

2.1 Motion Prediction

Given the past positions of each agent and a map of the road, a predictor outputs future trajectories over a horizon of typically three to eight seconds. Because human behaviour is uncertain, good predictors output several possible trajectories with probabilities. Fig. 1 classifies the main model families and optimization techniques.

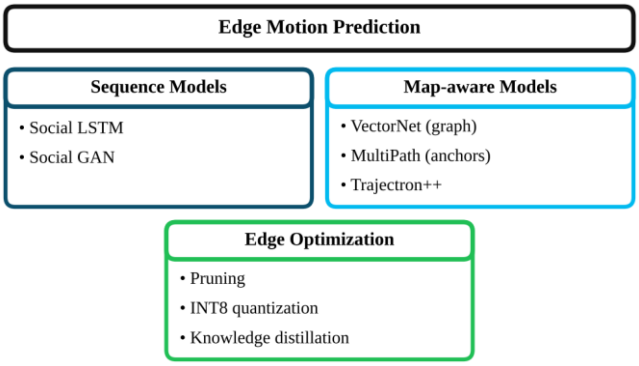


Fig. 1. Prediction models and edge optimization techniques.

2.2 Edge Constraints

Onboard computers must meet hard latency deadlines, operate within limited power and thermal budgets, and satisfy automotive safety standards. Model size and arithmetic precision therefore matter as much as accuracy.

III. LITERATURE SURVEY

Alahi *et al.* [3] proposed Social LSTM, which shares hidden states among nearby pedestrians through a pooling layer so that predicted paths respect social interactions. Gupta *et al.* [4] used generative adversarial networks in Social GAN to

© 2026 IJSRET

produce multiple socially plausible trajectories, addressing the multi-modal nature of human motion.

For vehicles, map context is crucial. Gao *et al.* [5] represented lanes and agent histories as vectors and used a graph network, VectorNet, which was both more accurate and more efficient than rasterized image inputs. Chai *et al.* [6] introduced MultiPath, which predicts a distribution over a fixed set of anchor trajectories, and Trajectron++ [7] produced dynamically feasible forecasts from heterogeneous inputs.

Large datasets enabled this progress: Argoverse [8] provided rich maps and forecasting scenarios, and the Waymo Open Motion Dataset [9] offered large-scale interactive scenarios.

For efficient deployment, Han *et al.* [10] combined pruning, quantization and coding to compress networks greatly with little accuracy loss. Jacob *et al.* [11] developed integer-only quantization for efficient inference, and Hinton *et al.* [12] showed that knowledge distillation transfers knowledge from a large teacher to a small student network. Table I summarizes these works.

TABLE I Summary of Literature Survey

Ref.	Authors	Year	Technique Used	Merits	Demerits
[3]	Alahi et al.	2016	Social LSTM with pooling	Socially aware pedestrian prediction	No map context
[10]	Han et al.	2016	Pruning + quantization + coding	Large compression, little loss	Evaluated on vision models
[4]	Gupta et al.	2018	Social GAN	Multiple plausible trajectories	Training instability
[11]	Jacob et al.	2018	Integer-only quantization	Efficient inference on hardware	Some accuracy loss
[6]	Chai et al.	2019	MultiPath anchor trajectories	Probabilistic multi-modal output	Fixed anchor set
[5]	Gao et al.	2020	VectorNet graph encoding	Accurate and efficient map encoding	Needs vectorized HD maps
[7]	Salzmann et al.	2020	Trajectron++	Dynamically feasible forecasts	Complex model
[9]	Ettinger et al.	2021	Waymo Open Motion Dataset	Large interactive scenarios	US driving only

IV. RESEARCH GAP

- Most prediction research optimizes accuracy on benchmarks without reporting onboard latency or power.
- Compression methods are rarely evaluated on multi-modal trajectory predictors.
- Prediction failures in rare or unusual scenarios are poorly characterized.
- Indian mixed traffic with two-wheelers, auto-rickshaws and pedestrians is under-represented in datasets.
- Uncertainty estimates are not always calibrated for downstream planning.

4.1 Objectives

- To review trajectory prediction models for autonomous driving.
- To examine model optimization techniques for edge deployment.
- To propose an onboard Edge AI architecture for real-time prediction.
- To compare models on accuracy, multi-modality and computational cost.
- To identify challenges and future directions.

4.2 Proposed System Architecture

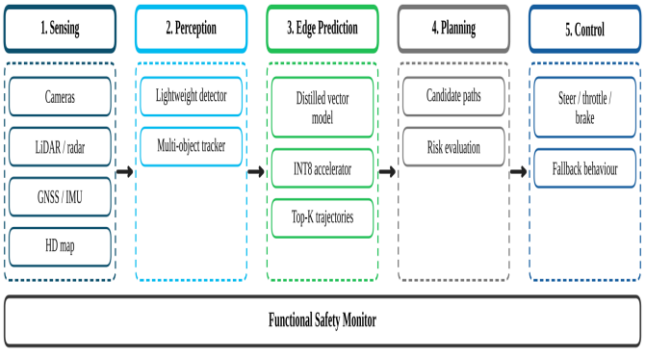


Fig. 2 shows the proposed onboard architecture.

Fig. 2. Proposed onboard Edge AI architecture for real-time prediction.

© 2026 IJSRET

4.3 Sensing Layer

Cameras, LiDAR, radar and GNSS/IMU provide synchronized data, and a high-definition map supplies lane geometry.

4.4 Perception Layer

Lightweight detectors identify and track agents; tracks provide position, velocity and heading histories.

4.5 Edge Prediction Layer

A distilled, INT8-quantized vector-based predictor outputs the top K trajectories with probabilities for each agent, running on an automotive AI accelerator.

4.6 Planning Layer

The planner evaluates candidate ego paths against predicted futures, choosing safe and comfortable manoeuvres.

4.7 Control Layer

Steering, throttle and braking commands are issued, with a safety monitor enforcing limits and fallback behaviour.

V. METHODOLOGY AND WORKING

Fig. 3 describes the real-time loop. Prediction accuracy is measured by minimum average displacement error over K predicted trajectories:

$$minADE_K = min_k \, (1/T) \, \Sigma_t \, \|\hat{y}_t^k - y_t\| \quad (1)$$

End-to-end latency is the sum of sensing, perception, prediction and planning times, and must stay below the control cycle period, typically around 100 ms.

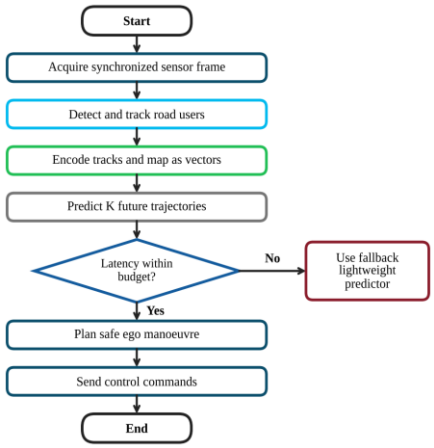


Fig. 3. Real-time onboard prediction loop.

Distillation trains the student to match the teacher's output distribution, and quantization-aware training reduces accuracy loss when weights and activations are converted to 8-bit integers.

VI. COMPARATIVE ANALYSIS

Table II compares prediction models. Recurrent models are simple but ignore maps; GAN-based models capture multiple futures; vector-based graph models are accurate and efficient; anchor-based models offer predictable computation suited to edge deployment.

TABLE II Comparison of Prediction Models

Model	Multi-modal	Map-aware	Edge Cost
Social LSTM	No	No	Low
Social GAN	Yes	No	Medium
VectorNet	With decoder	Yes	Low–Medium
MultiPath	Yes	Yes	Low
Trajectron++	Yes	Yes	Medium–High

VII. CHALLENGES AND FUTURE SCOPE

Table III lists challenges. Future work includes joint perception–prediction models that share computation, transformer predictors optimized for automotive chips, uncertainty-aware planning, and datasets representing Indian traffic.

TABLE III Challenges and Mitigation

Challenge	Mitigation
Latency deadlines	Quantization; accelerator-aware design
Accuracy loss from compression	Distillation; quantization-aware training
Rare scenarios	Scenario mining; simulation
Indian mixed traffic	Local datasets and fine-tuning
Uncertainty calibration	Probabilistic outputs; calibration checks

VIII. CONCLUSION

Real-time motion prediction is a cornerstone of safe autonomous driving, and performing it on the vehicle avoids the latency and connectivity risks of the cloud. The reviewed literature shows that graph-based, multi-modal predictors provide accurate forecasts, and that pruning, quantization and distillation make them small enough for onboard hardware. Reporting latency alongside accuracy and testing on diverse traffic will be essential next steps.

© 2026 IJSRET

8.1 Acknowledgment

The authors express their sincere gratitude to the Department of Artificial Intelligence and Data Science, Jaihind College of Engineering, Pune, for providing the facilities, guidance and encouragement needed to carry out this work. The authors also thank the faculty members and colleagues whose valuable suggestions helped improve the quality of this review.

REFERENCES

1. W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," IEEE Internet Things J., vol. 3, no. 5, pp. 637–646, 2016.
2. S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, and W. Shi, "Edge computing for autonomous driving: Opportunities and challenges," Proc. IEEE, vol. 107, no. 8, pp. 1697–1716, 2019.
3. A. Alahi, K. Goel, V. Ramanathan, A. Robicquet, L. Fei-Fei, and S. Savarese, "Social LSTM: Human trajectory prediction in crowded spaces," in Proc. IEEE CVPR, 2016, pp. 961–971.
4. A. Gupta, J. Johnson, L. Fei-Fei, S. Savarese, and A. Alahi, "Social GAN: Socially acceptable trajectories with generative adversarial networks," in Proc. IEEE/CVF CVPR, 2018, pp. 2255–2264.
5. J. Gao et al., "VectorNet: Encoding HD maps and agent dynamics from vectorized representation," in Proc. IEEE/CVF CVPR, 2020, pp. 11525–11533.
6. Y. Chai, B. Sapp, M. Bansal, and D. Anguelov, "MultiPath: Multiple probabilistic anchor trajectory hypotheses for behavior prediction," in Proc. Conf. Robot Learn. (CoRL), 2019.
7. T. Salzmann, B. Ivanovic, P. Chakravarty, and M. Pavone, "Trajectron++: Dynamically-feasible trajectory forecasting with heterogeneous data," in Proc. ECCV, 2020, pp. 683–700.
8. M.-F. Chang et al., "Argoverse: 3D tracking and forecasting with rich maps," in Proc. IEEE/CVF CVPR, 2019, pp. 8748–8757.
9. S. Ettinger et al., "Large scale interactive motion forecasting for autonomous driving: The Waymo Open Motion Dataset," in Proc. IEEE/CVF ICCV, 2021, pp. 9710–9719.
10. S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding," in Proc. ICLR, 2016.
11. B. Jacob et al., "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE/CVF CVPR, 2018, pp. 2704–2713.
12. G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," arXiv:1503.02531, 2015.

© 2026 IJSRET