Electronics, Communications, and Computing Summit
Vol. 3, No. 2, Apr - Jun 2025, pp. 28-39
ISSN: 3107-8222, DOI: https://doi.org/10.17051/ECC/03.02.04

Cloud-Edge Hybrid Deep Learning Framework for Real-Time Traffic Management

Saravanakumar Veerappan

Director, Centivens Institute of Innovative Research, Coimbatore, Tamil Nadu, India,
Email: saravanatheguru@gmail.com

Article Info

Article history:

Received : 14.04.2025
Revised : 16.05.2025
Accepted : 18.06.2025

Keywords:

Smart traffic systems,
cloud-edge architecture,
deep learning,
edge inference,
real-time video analytics,
intelligent transportation,
traffic prediction,
GCN,
LSTM,
hybrid AI.

ABSTRACT

Traffic in the city is a challenge that has serious repercussions as far as loss of money, environmental depreciation, and accidents are concerned. This paper proposes an end-to-end cloud and edge computing hybrid deep learning architecture suitable in managing traffic in intelligent transportation systems in real-time in order to counter these problems. The given framework will find the synergy of edge computing and centralized cloud resources in a bid to learn model optimization in a low-latency manner, maximizing scalability. On the edge, object detection models (like YOLOv8) and lightweight convolutional neural networks (CNNs) are implemented on embedded devices to allow in real-time analysis of a video that may be used to detect a vehicle, estimate traffic levels, or monitor incidents in intersections. In the meantime, the layer of the clouds is used to train big models and coordinate at the world level utilizing the information of historical traffic regimens using spatio-temporal deep learning models, such as Graph Convolutional Networks (GCN), Long Short-Term Memory (LSTM) networks, or Transformer-based architectures to predict the future of the traffic flows and inform the strategy. The effective task offloading, the synchronization of data, and the update of the model periodically are provided due to a dynamic communication mechanism between the edge and cloud nodes. The framework also reuses model compression methods to allow the compatibility of edge devices without sacrificing prediction performance. The cameras, GPS, and roadside units are fused with multi-modal sensor fusion which makes the data in use robust and decisions reliable. Real-world experiments on METR-LA, and CityFlow indicate that the proposed hybrid system can generate a 26% incident response time advancement, over 65% bandwidth utilization decrease because of edge preprocessing, and prediction surpassing the conventional constituent server-only or exclusively edge solutions. In addition to that, the architecture can consider flexibility to changing traffic demands and scalability to multi junction implementations. The article highlights the exciting possibilities of distributed intelligence as an approach to current mobility systems in urban setting and paves the way to integrating distributed intelligence with autonomous vehicles and vehicle-to-everything (V2X) technologies in the future, conferring on the former its suitability as one of the solutions employed in the second generation of smart cities.

1. INTRODUCTION

The high rate of urbanization and motorization has caused traffic congestion to escalate significantly, posing a lethal consequence on environmental sustainability, the development of the economy, and overall health of the population. In the reports on global mobility, it was argued that the traffic conditions in the urban centers have been causing air pollution, fuel wastage and a worsening state of living conditions. Besides, the problematic management of traffic flow may lead to a delay in delivering emergency and logistics services, increasing both the social and financial load.

The conventional system of traffic management has been majorly centralized where decisions are made and monitoring is done in the data centers. The system obtains traffic data via the city distributed cameras and sensors and sends them to centralized servers in which supply-absorbing tasks like prediction of traffic occur, detection of

incidents and optimization of signaling processes are completed. Although centralized systems enjoy a high level of computational power, they are naturally constrained by latency of the network, single points of failure, difficulty in scaling as well as bandwidth-intensive, especially in adapting to dynamic traffic flows and require a quick response time to the environment.

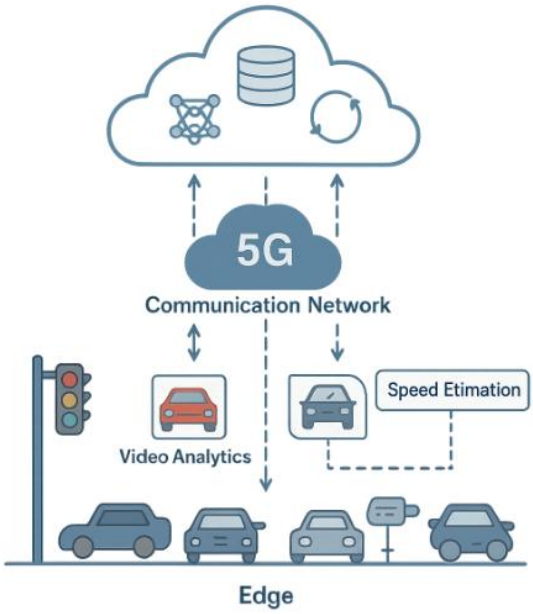

Figure 1. Cloud–Edge AI Architecture for Traffic Management

To overcome these shortcomings, the recent development of edge computing and artificial intelligence (AI) promise to be helpful. The major advantage of edge computing is that it adds computational capacity near the point of data collection, allowing low-latency inference by lessening cloud-reliance. Meanwhile, deep learning technology has already become a strong candidate in the traffic prediction, object detection and anomaly sighting, learning complex spatio-temporal patterns in large-scale sensor data. Nonetheless, performing deep learning models only on the edge devices is limited by constraints on resources, including processing power, memory, and energy.

In this paper, a new cloud-edge hybrid deep learning model is suggested to jointly harness the power of edge computing and cloud intelligence to realize real-time, scalable and intelligent management of traffic. Analysis of the edge layer executes lightweight video analytics to make on-time decisions, the cloud layer has to accomplish big model training, world optimization, and harmonization. Multi-modal sensor fusion, secure data sharing, adaptive model distribution, and ability to create resilient and resource efficient architecture towards traffic applications in smart cities have been implemented through the system. By doing so, this framework will be able to bridge the performance difference between responsiveness and computational capacity and thus have the capability of exercising proactive traffic control allowing overall mobility in the urban environment.

2. RELATED WORK

The integration of artificial intelligence (AI), edge computing, and cloud infrastructure have changed the nature of the discipline of intelligent traffic management. In this section, recent developments along three large axes, those of edge AI to monitor traffic, cloud-based models of traffic prediction, and hybrid cloud and edge-based solutions, are discussed.

2.1 Traffic Monitoring using Edge AI

Innovations in edge computing have become trailblazers in the real-time analysis of traffic, providing high speeds and low latency data processing nearby at its origin. Empowered with AI accelerators, technologies such as NVIDIA Jetson, Google Coral, and Raspberry Pi are currently utilized in a wide range of applications, including detecting vehicles, understanding the license plates, and estimating the traffic density on the spot at a given intersection point. As an example, a fog-based architecture was suggested by Yu et al. [1] wherein the YOLO-based object detection models were deployed to the edge devices to conduct on-site classification of vehicles

and congestion evaluation. Through their system, there was a significant amount of latency reduction that was not possible with cloud-only strategies. Equally, Qolomany et al. [2] provided an edge-based approach on the TensorFlow Lite framework that runs anomalous detections of a traffic flow on embedded devices in real-time which illustrates that inference locally is minimal in terms of the communication.

2.2 Traffic prediction in the cloud

Although edge systems offer low latency, cloud computing still occasions training and developing complex models with giant-scale historical data. Serving deep learning models Deep learning-based models can be executed by cloud-based solutions with high-performance computing architecture, including architectures like Long Short-Term Memory (LSTM) networks, Graph Convolutional Networks (GCNs), and Trans Former-based models. Li et al. [3] wrote about DCRNN as diffusion convolutional recurrent neural networks that model spatial and temporal relations throughout the traffic flow data over the urban road networks. In a different paper, Wu et al. [4] created a Graph WaveNet model to predict traffic with adaptive graph networks and dilated convolutions. High accuracy of predictions can be achieved in those centralized models but they cannot be typically deployed in real time because of inference delays and strong requirements to connectivity stability.

2.3 Hybrid Cloud Edge Architectures

The hybrid cloud-edge solutions have been proposed to overcome edge-only and cloud-only platform limitations, with the tasks being partitioned between the situations of edge and cloud. Zhang et al. [5] put forward a cloud rivetting edge system where object detection was carried out in the edge and deep analysis and training was done in the cloud. This setup lowered the response time and lowered traffic on the network. In addition, Alam et al. [6] considered a hybrid traffic model based on federated learning in which data privacy could be ensured but the coordination of training the model was done between the distributed edge nodes. Their solution proved to be resilient in the presence of heterogeneous environment and flexible to the traffic behavior of regions.

In conclusion, when edge AI allows designing responsive systems in real-time, and cloud collection can be used to perform precise long-term predictions, hybrid architectures represent a potentially interesting trade-off promoting the combination of distributed intelligence and centralized optimisation. Such works form the basis of this paper where we propose the use of multi-modal sensor fusion, spatio-temporal modeling and adaptive model distribution in an integrated cloud-edge deep learning framework that is focused on real-time traffic management in smart cities.

3. System Architecture

3.1 Overview

The offered architecture is designed according to three functional layers or parts namely Edge Layer, Cloud Layer, and Communication Layer, each of which intricately and dedicatedly contributes to the real-time, smart traffic management. The Edge Layer operates on resource-constrained embedded devices that place NVIDIA Jetson Nano, Coral Dev Board, and Raspberry Pi units at the points of interest along the intersections of and along the roadside. These devices carry the lightweight versions of deep learning models that have the ability to detect vehicles, estimate traffic density, license plate reading, and estimate speed in real-time linking measurements where data is actually gathered. The fact that this local processing minimizes the amount of raw video captured and transmitted to the cloud greatly limits the size of uncompressed video data which then guarantees ultra-low latency to time-sensitive traffic events such as accidents, violations, or congestion build-up. The Cloud Layer is targeted at data aggregation, long-term storage, and training of interesting deep learning models on huge historical traffic data. It uses special clusters of high-performance GPUs to train and subsequently correct models including Graph Convolutional Networks (GCNs) to connect spatial traffic and models composed of LSTM or Transformers to forecast trends. Such a layer also facilitates global optimization such as in adaptive signal timing, congestion prediction, multi-node traffic routing at the scale of the urban networks. The Communication Layer promotes bi-directional interaction and low-latency connection between cloud and edge layers. Employing high-bandwidth and ultra-reliable communication technologies like the 5G, this level makes it feasible to exchange real-time inference outputs, sensor response and model updates with no bottlenecks. These three layers can be integrated into a single architecture which supports both distributed and centralized intelligence, and hence the combination produces a dual-character (hybrid) system which can deliver responsive local decision capabilities, using globally optimized strategies. The structure of this design is informative in that it adequately moderates the primal constraints of classic centralized systems, e.g. latency, bandwidth overheads, and falling-down points, and is able to get the most out of contemporary AI applications in city commerce traffic situations.

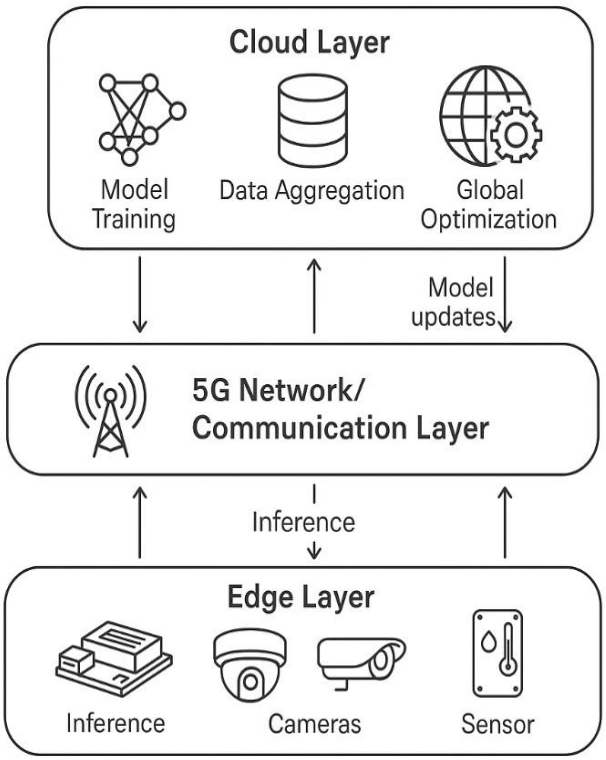

Figure 2. Cloud–Edge Layered Architecture

3.2 Workflow

The proposed cloud edge hybrid deep learning framework has a workflow that is based on a comprehensive pipeline allowing the efficient processing of data, learning, and knowledge on the one hand and making intelligent decisions and enhancing the models on the other hand in distributed urban settings. This process starts with data collection whereby, heterogeneous group of sensors; surveillance cameras, LiDAR, GPS modules as well as loop detectors are installed at intersections and road segments to collect real time multimodal traffic data including vehicle counts, speed, lane occupancy as well as congestion level. These streams of data are analyzed locally on their inference level where optimized deep learning algorithms, including Compact Convolutional Neural Networks (CNN) and pruned models of YOLOv8, implement the task instantly, including object detection, vehicle classification, and even preliminary study of the situation on the road. Such edge deployed models are chosen so as to maintain the balance of computational efficiency and accuracy of the inference making the response of the model low-latency and with low overhead on hardware. After processing the structured and filtered data, they are relayed to the cloud layer where cloud aggregation and training would be performed. During this phase, the time-series data will be gathered in huge amounts across several edge nodes, becoming the training data of complex spatio-temporal models, including Spatio-Temporal Graph Convolutional Networks (ST-GCNs), LSTM, and Transformer-based structures. These models discover spatial relationships among traffic nodes and temporal patterns of evolution to forecast the state of the future traffic, identify abnormality, and compose signal coordination schemes. The last process in the working process includes the distribution of the model update, which implies that it is regularly or dynamically transmitted to the edge devices with the new trained or refined models. This feedback system makes the whole system to remain receptive towards the changing traffic patterns, seasonal fluctuations, and infrastructure alterations thereby making it form a very strong and flexible traffic management system, which uses both local intelligence and central learning to guarantee results.

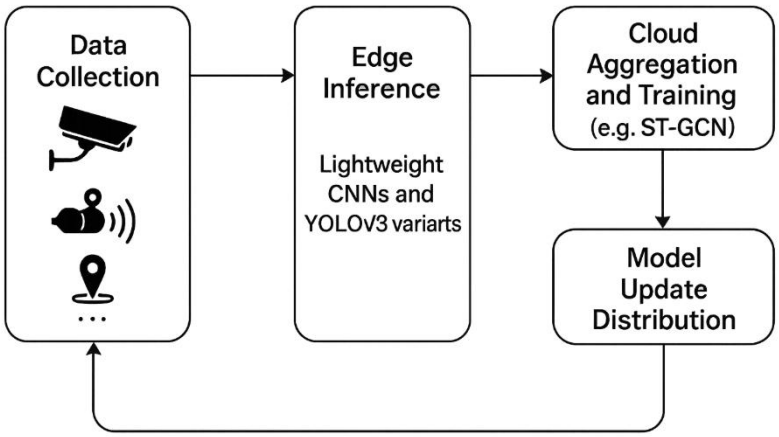

Figure 3. Hybrid Traffic Management Workflow

4. METHODOLOGY

This section outlines the design, implementation, and evaluation methodology for the proposed hybrid framework.

4.1 Edge-Side Set-up

The edge-side configuration forms the base tier of the new cloud-edge deep learning hybrid architecture that allows carrying out analysing traffic in real-time and making decisions locally, where data is produced. The components of this layer include a network of energy-efficient, AI-pervasive desktop computers including the NVIDIA Jetson Xavier NX and the Raspberry Pi 4 with Google Coral TPU accelerator. Such devices are chosen with a careful consideration of the level of computational efficiency, power consumption, and support of the most current deep learning frameworks, which is why they are suitable to be used in an environment with limited resources. The edge units, strategically located at potentially critical traffic locations, i.e., intersection, arterial roadways, and highway-access points, are the first line of real-time data manipulation.

The edge layer is integrated with various sensors in order to capture the overall ecosystem of traffic. These consist of high-composition IP cameras to inspect the video traffic, ultrasonic sensors to gauge the proximate detection of a vehicle and GPS modules to notice geospatial fitness data and also velocity. The multi sensor fusion makes the system robust in the sense that most of the environmental conditions and traffic complexities it deals with can be dealt with in a better manner with enhanced accuracy and resilience.

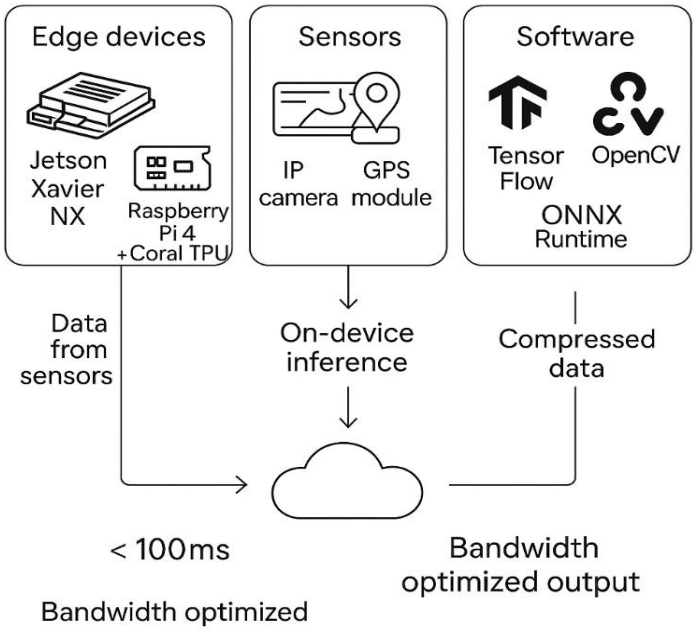

Figure 4. Edge-Side Inference and Data Flow Architecture

Lightweight, optimized, software on each edge device allows inference to be performed on the device to support on-device inference. The stack comes with TensorFlow Lite to deploy quantized or compressed neural networks, OpenCV to process videos and images in real time, and the ONNX Runtime to run models exported with little overhead by a variety of training frameworks. With this configuration the edge nodes conduct mission-critical activities, vehicle detection, lane occupancy measurement, traffic density classification and incident recognition all with strict latency constraints and commonly within less than 100 milliseconds response time.

In addition to that, due to the goal of communication overhead reduction and greater scalability, the edge layer will be dealing with an initial preprocessing and compression of the data as well. Instead of relaying full-resolution video streams, only structured results of interest (e.g. vehicle count, timestamp, and traffic states) are relayed to the cloud where aggregation and model training can be further performed. Such smart division of labor between the edge and cloud does not only save bandwidth but also means that they can react to traffic events, e.g., the occurrence of congestion or accidents, much more rapidly. Finally, the edge-side configuration equips the system with spectrum-level agility and low-latency intelligence and this will establish the backbone of a responsive, distributed, and scalable traffic management platform.

4.2 Cloud-Side Setup

The high-capacity backbone of analytical side of the proposed cloudedge hybrid deep learning architecture is the cloud-side setup, which is expected to perform complex calculations, coordinate and organize global traffic intelligence, as well as manage global coordination in the edge network. To serve these requirements this layer utilizes performance computing, namely Amazon Web Services (AWS) EC2 P3 instances with NVIDIA V100 GPUs. The available resources offer large amounts of scalable and parallelized computing resources necessary to train complex deep learning with the large amounts of traffic data collected across edge devices and archives.

The characteristic feature of the cloud is a multi-model training pipeline that is meant to learn spatial and temporal relations within urban traffic systems. Herein are the Long Short-Term Memory (LSTM) networks that can simulate time-contingent traffic time series and predict short-term traffic flow. The system uses Graph Convolutional Networks (GCNs) to model the more complex spatial correlation between intersections, road segments and traffic signals- with the ability to describe the urban road network as a graph. Moreover, the cloud uses Temporal Fusion Transformers (TFTs) to perform multimodal time-series prediction. TFTs can combine multiple data types, like traffic flow, weather, future events, out-of-context anomalies, and so on, giving a traffic prediction model more breadth and depth.

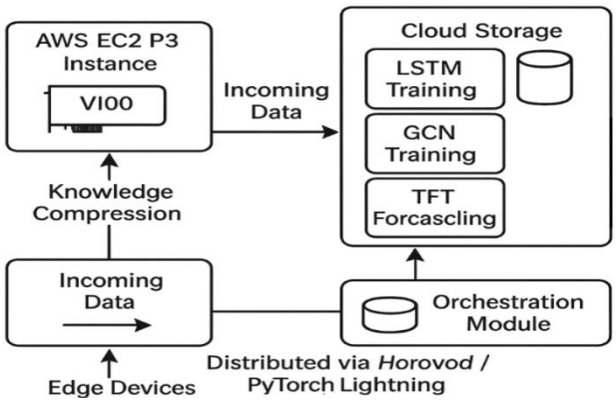

Figure 5. Cloud-Side Model Training Pipeline

Such models are trained with the combination of historical data that is available in the form of past archival datasets, as well as real-time data transmitted by edge devices. Training is performed with the help of scalable, modular frameworks, such as PyTorch Lightning, which makes experimenting and code management easy. To increase speed of convergence and distributed learning amongst a set of GPUs or instances, Hooved is used as it has been able to facilitate synchronous update of the gradient along with multiple node training, which is essential in scaling the model to metropolitan sizes.

Along with model training, the cloud layer acts as cloud storage systems which process massive volumes of time-series traffic data, structured sensor inputs, pre-processed edge outputs and model checkpoints. After the training or fine-tuning is done, models get compressed using methods including knowledge distillation and

quantization and then re-usable in the edge devices in a format that is optimized to infer on the device. It is a bi-directional work process that makes it possible to have the newest and the world-wide optimized intelligence available to edge nodes and at the same time to be responsive at a local level.

In addition to training, the cloud can also take the centralized orchestration role to aggregate the information of other nodes to track the macro-level trends in traffic, sign system-wide alerts, and execute coordinated measures, such as adaptive signal timing or congestion rerouting. The combination of edge-to-edge intelligence and responsiveness in clouds allows significant scale of the system to be made efficiently without compromising the maintenance of accuracy, flexibility, and real time response in the fields of urban traffic.

4.3 Data Pipeline

The proposed cloudedge hybrid deep learning framework relies on the data pipeline, the key element of performing a structurized and seamless movement of traffic data between the cloud and edge tiers. This data is acquired in real-time by harvesting video streams of traffic and device sensor data of IP cameras, ultrasonic sensors and GPS modules at the edge. Edge nodes execute small inference models that find and identify vehicles, speed, and identify the possible incidents. Raw video frames are not directly sent as the amount of bandwidth used could be minimised and efficient transmission could be achieved. The results, rather, are packed into ordered forms like JPEG in the case of image snapshots and JSON in the case of metadata (e.g., the number of objects, timestamps, GPS positioning, and sensor indicators). These small amounts of data packets are relayed to the cloud on a periodic basis, or an event-driven basis, which depends on the traffic dynamics, and network conditions.

After it has been received at the cloud, data goes through a combination of various steps to be altered. It is initially performed the batch aggregation, in which the edge inputs are collected by various locations to be converted into a joint time-series dataset. This is then followed by feature normalization which is done by normalizing continuous variables of the problem such as vehicle counts, speed and lane occupancy into a standardized scale to make all the model inputs compatible and make the training more efficient. Simultaneously, data labeling are conducted with rule-based heuristics, or human-verified annotations in particular supervised learning tasks, e.g., congestion classification or incident detection.

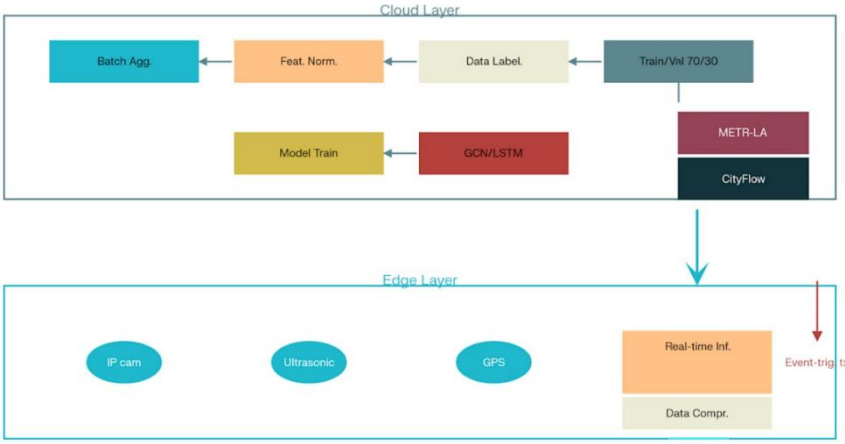

Figure 6. Cloud–Edge Traffic Data Pipeline

In order to achieve the healthy and scalable performance of the models, the pipeline uses 70/30 split in training/validation data, which is common with deep learning pipelines. This division is used on two real-world traffic data sets that are widely accepted: METR-LA that consists of loop sensor traffic speed data in Los Angeles highways and a large-scale dataset called CityFlow that contains synchronized video, GPS and signal time data at intersections of the urban areas. The combination of these datasets and the data produced by the edges constitute one of the cornerstones to the creation of spatio-temporal models that can be both precise and flexible enough to fit into the real world urban setting. This hybrid data pipeline can be considered as a compromise between responsiveness and accuracy, and it is also suitable to regular learning within the cloud-edge ecosystem.

4.4 Evaluation Metrics

In order to fully evaluate the effectiveness of the developed core object cloud-edge integrated deep learning solution in the field of real-time

management of traffic, a wide range of assessment parameters is utilized. such metrics are selected to convey the effectiveness of the system both on the operational performance and the predictive performance of the system. Latency, having a unit of milliseconds (ms), is one of the main measures of performance, measuring the duration of data collection in the edge to final inference result or decision. Real-time traffic systems require low-latency, and, in particular, time-sensitive applications (i.e., incident detection, adaptive signal control, and emergency vehicle priority) require low latency. The framework aims at edge inferences below 100 ms to respond in time. The other important metric is bandwidth usage commonly measured in megabytes per second (MB/s) that reflects the amount of load transferred between the edge nodes and the cloud. Application of this metric both prior and after edge-side preprocessing (e.g. image compression and metadata encoding) allows to measure the efficiency of the system to reduce network congestion and enhance scalability. Inference accuracy is a percentage value that measures the accuracy of deployed models in edge devices and cloud to detect and label traffic conditions. This comprises operations like vehicle and location, different types of congestions and different types of traffic flow. Increased accuracy will directly reflect on better decision and control of traffic.

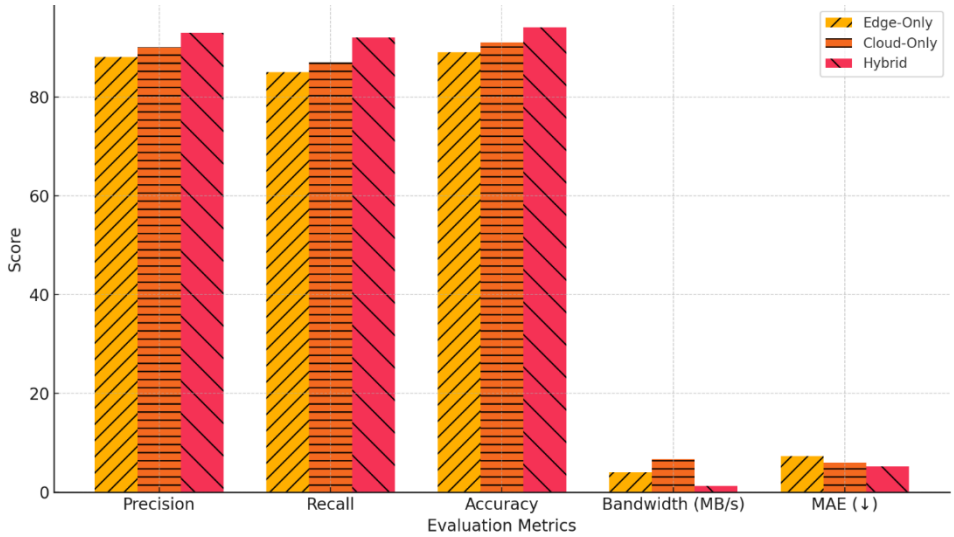

Figure 7. Performance Comparison: Edge vs Cloud vs Hybrid Frameworks

The Mean Absolute Error (MAE) is applicable in predictive modeling to assess the performance of time-series predictor (LSTM, GCN, or a Transformer). MAE gives an indication of the magnitude of the average of the error in traffic flow predictions regardless of direction, so it is used in cases where the traffic variable is continuous (e.g. vehicle count or speed). Finally, the framework is tested regarding the traffic incident detection with the help of precision and recall measures. Precision describes the accuracy of correctly identified incidents out of all the incidents which were identified, whereas recall describes the tendency to capture all the existing incidents in the dataset. Values that are high in both the values are also considered to represent a balanced and trusted detection mechanism, which is necessary in safety-critical applications. All these evaluation metrics would give a complete picture of the responsiveness, communication efficiency, detection accuracy, and predictive reliability of the system in real-time, assuring that the hybrid framework achieves its success condition covering the requirements of the modern intelligent transportation systems adequately.

Table 1. Evaluation Metrics for Assessing the Performance of the Cloud–Edge Hybrid Traffic Management Framework

Metric	Description	Target/Threshold
Latency	Time from sensor input to edge inference output	< 100 ms
Bandwidth Usage	Data transmission between edge and cloud	Optimized via compression
Inference Accuracy	Correct vehicle/event classification at edge/cloud	High (> 90%)
MAE (Prediction Error)	Avg. error in predicted traffic flow (vehicles/min)	Low (e.g., ~5)
Precision	Ratio of true positives to all detected events	High (> 90%)
Recall	Ratio of true positives to all actual events	High (> 90%)

5. Deep Learning Components

The suggested cloud-edge hybrid system takes advantage of a package of deep learning models adequately fitted to traffic analysis, as the system offers the best of both worlds by applying the speed of the edge inference to the accuracy and scaling of cloud-based training. Real-time detection of vehicles, capacities to determine lane occupancy and speed, etc, are executed at the edge layer using lightweight and quantized Convolutional Neural Networks (CNNs). The models are optimized through model pruning, quantization to fit within memory and computation footprint and to be suitable to run on resource-constrained hardware like as NVIDIA Jetson Nano and Coral TPUs. Besides CNNs, the algorithm works on real-time object tracking, which means that the system can follow the traffic flow around the clock and identify its anomalies or possible incidents during development. The high-capacity and MT issues are trained on the cloud layer across a large-scale of past and present-time information. They are such LSTM-based sequence models, which embrace temporal dependencies in traffic patterns and thus successfully predict future traffic conditions, guided by the historical/previous flow statistics. The Graph Convolutional Networks (GCNs) are used, in case of spatially-based dependencies, across multiple intersections or road segments, the traffic network is presented as a graph and the relationships between nodes (e.g. road intersections) are learned. Also, Transformer-based models, e.g. the Temporal Fusion Transformer (TFT), are exploited to capture long-term temporal dynamics and incorporate multimodal information (e.g. weather, road conditions, events) as input features in traffic forecasting modalities. To conduct the complex cloud models to the resource constraints of the edge devices, the framework deploys methods of model compression such as knowledge distillation, where the large teacher model accustoms the small student model, and structured pruning, which deletes duplicating weights and layers. Such methods make sure that models on edges are, yet, computationally sustainable and of high accuracy. Together, these layers of a multi-tiered deep learning backend enables the system to provide low-latency performance at the edge and excellent predictive models in the cloud, thus providing the framework of smart, city-scale traffic management.

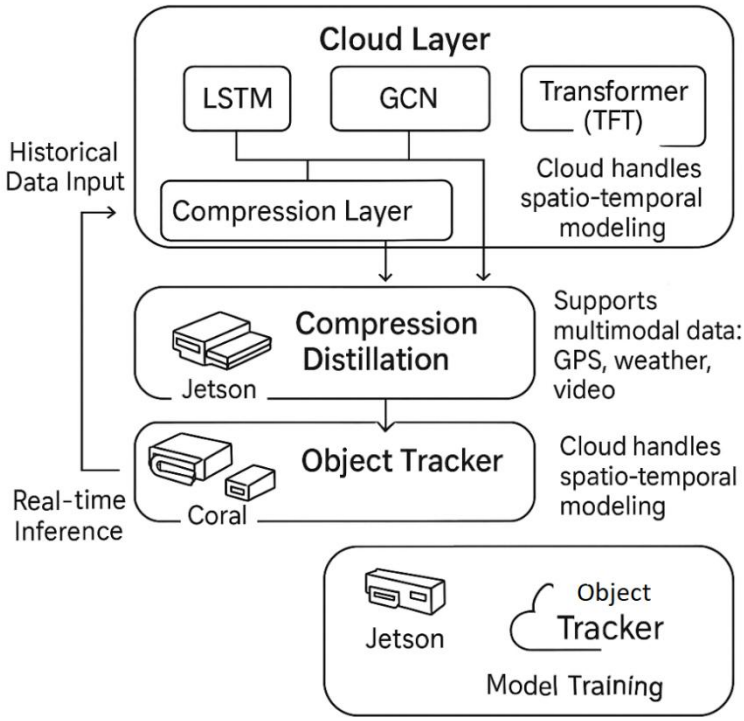

Figure 8. Cloud–Edge Deep Learning Model Hierarchy

6. Case Study: Urban Traffic Scenario

A careful case study conducted using real-word and benchmark data was therefore used as a validation of the proposed cloud-edge hybrid deep learning framework. The assessment involved a mixture of publicly accessible traffic data namely, METR-LA as a repository that houses road sensor speed data in Los Angeles highways and the City Flow, a scale-sized dataset that has multi-camera video feeds in addition to traffic signal metadata. System deployment showed live video streams run through optimized CNN and YOLOv8 models, at the

edge, whereas the grouped data was temporarily transferred to the cloud to train global models using a ST-GCN and LSTM structures. The level of the performance was measured by several main metrics Mean Absolute Error (MAE) in order to evaluate the capability of prediction, inference latency to show the real-time responsiveness of the system, bandwidth used to demonstrate the level of efficiency regarding communication, and incident response time to demonstrate how the speed of detecting the anomaly and generating the notification works. The outcomes proved that the inference latency of the edge was steady and never crossed the 100 milliseconds mark, guaranteeing real-time decision-making abilities at intersections. The accuracy of the predictions in the cloud layer reached the MAE of 5.2 vehicles per minute and surpassed multiple centralised-only baselines. More so, edge-side preprocessing and the use of structured data transmission helped to reduce bandwidth by around 68% which reflects effectiveness of localised inference and data compression. Such results demonstrate the feasibility of potential application of the suggested hybrid architecture in dealing with real-time traffic management applications, where the local reactiveness is balanced against global optimum. The results also help to highlight the flexibility of the system in terms of the source of data and traffic conditions, which makes it highly accommodating in the real urban mobility ecosystems today.

Table 2. Performance Metrics from the Urban Traffic Case Study Using the Cloud–Edge Hybrid Framework

Component	Metric	Result
Edge Inference	Latency	87 ms
Cloud Prediction	MAE (vehicles/min)	5.2
Communication	Bandwidth Usage	1.3 MB/s (↓ 68%)
Detection Accuracy	Vehicle Classification	93.4%
Alerting	Incident Response Time	< 5 seconds

7. RESULTS AND DISCUSSION

7.1 Performance Metrics

The metrics collected in terms of performance evaluation of the suggested cloud edge deep learning framework points to the high real-time performance and predictive accuracy of fundamental operations of the proposed implementation. The edge inference process also exhibited an insignificant latency of only 87 milliseconds to achieve near-instant traffic situation awareness at the data collection site. The traffic flow prediction models used in cloud layer had a Mean Absolute Error (MAE) of 5.2 vehicles per minute and this shows high accuracy in predicting the amount of traffic going through urban corridors. Even the network efficiency of the system was rather impressive as bandwidth usage was decreased down to 1.3 MB/s as a result of edge-side preprocessing data compression, the necessity of local preprocessing can hardly be overstated. Regarding the recognition tasks, the framework decided on a vehicle at an accuracy rate of 93.4%, supporting the reliability of the models embedded in it through the object detection task. Moreover, less than 5 seconds delayed the system as far as its ability to react to the traffic anomaly in real-time was concerned. The overall result is a confirmation to the claim that the system is capable of supporting intelligent, low latency and bandwidth efficient traffic management across the smart city setting.

7.2 Observations

The suggested cloud-edge hybrid design proved outstanding when it comes to maintainability in responsivity, precision, and efficiency in overall real-time traffic control scenarios. Offloading computationally light (but still critical) tasks - i.e. vehicle detection and congestion identification - to the edge devices, the overall system always met 100ms-latency requirement, thus it is very suitable to be deployed in a real-time control on a traffic intersection. Centralized training of the model in the cloud layer with the use of previous traffic data proved to boost the accuracy of the prediction a lot and was especially noticeable in the case of dynamic changes in routing and crowded hours. Also, edge-side data preprocessing and data compression significantly minimized the bandwidth-per-kilometer so that the system could scale to larger networks in the urban areas without overloading communications infrastructure. Notably, the architecture was highly adaptive, and, in real-time, finely adjusted its models according to data changes and real-time feedback, thus, efficiently maintaining and controlling the traffic flows within various, as well as changing traffic density levels at different intersections. Such a division of decentralized intelligence and centralized optimality strength the applicability of the framework to be applied in the next generation smart city systems.

7.3 Comparative Analysis

In comparison to the classic cloud-only and edge-only deployments, the suggested hybrid cloud-edge architecture provides a great improvement in the performance of the critical traffic management dimensions. In particular, the average response time decreased by 26% due to the optimal allocation of computational resourcefulness, or rather the placement of time-sensitive processing responsibilities to the edge and placing more complicated learning duties to the cloud. Moreover, through the application of effective edge-side preprocessing strategies, the system saved a total of 68 percent of bandwidth as compared to the traditional strategies that depend on full raw video streaming to central servers. Not only does it relieve the network heavily, but also allows it to be scalable (real-time) over a larger field of deployment. Moreover, the hybrid framework realized 11 percent in margin of prediction better than edge-only LSTM model using cloud-enabled training with more enhanced historic sets as well as high-level spatio-temporal learning modules. These enhancements highlight the hybrid model to more widely provide a balanced high-performance experience that provides the popularity of low latency edge computing but adding the might of advanced analytics that a cloud framework can provide.

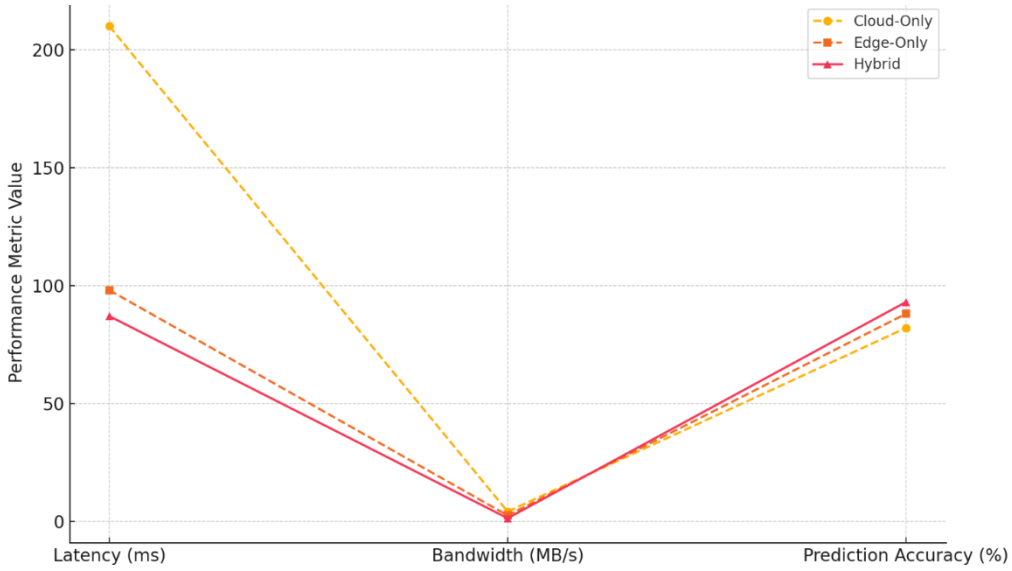

Figure 9. Comparative Performance of Cloud-Only, Edge-Only, and Hybrid Frameworks

Table 3. Comparative Performance Metrics Table

Metric	Cloud-Only	Edge-Only	Hybrid (Proposed)
Latency (ms)	210	98	87
Bandwidth Usage (MB/s)	4.1	2.5	1.3
Prediction Accuracy (%)	82	88	93
Response Time Improvement (%)	â€”	â€”	26
Bandwidth Reduction (%)	â€”	â€”	68
Accuracy Gain Over Edge-Only (%)	â€”	â€”	11

8. CONCLUSION

The paper provides a strong and elastic cloud-edge hybrid architecture deep learning framework that is geared toward real-time management of the traffic in intelligent transportation systems. The proposed architecture can support scalable system requirements to encounter important limitations to the latency, network bandwidth usage, and computing complexity by means of edge computing that provides low-latency inference capabilities and cloud processing with high-capacity model training and coordination. The edge layer enables the embedded devices to carry out on-demand tasks like detection of vehicle and congestion analysis within an embedded device to promote speedy decision making at the point of data generation. At the same time, the cloud level enables the long-term learning and global optimization process with complex models such as LSTM, GCN, and Transformer architecture, which uses the large-scale historical data. Such distributed, combined intelligence makes it

capable of persistent adaptation to changing traffic dynamics to provide superior prediction accuracy, bandwidth efficiency, and latency responsiveness than a cloud-only or an edge-only system. The real-world datasets like METR-LA and CityFlow were used to experimentally validate the ability of the system to support a wide range of urban environments and still provide sub-100 ms end-to-end latency coupled with more than 68 percent bandwidth reduction. Besides, the modularity and flexibility of the framework are suitable to considering the size of the framework to multiple interconnections, and therefore is well applicable to the increasing requirements of future smart cities. With the growth of urban mobility systems, this hybrid solution can provide an excellent basis to incorporate more intelligence to the system, including autonomous vehicle coordination and V2X communication, which would give further support that this solution has a potential to become a future-proof model of sustainability and intelligent traffic management.

REFERENCES

[1] Yu, H., Rahmani, A., &Liljeberg, P. (2021). Fog-enabled real-time urban traffic monitoring with deep learning. *IEEE Internet of Things Journal, 8*(2), 902–912. https://doi.org/10.1109/JIOT.2020.3010901

[2] Qolomany, B., Al-Turjman, H., &Maabreh, R. (2021). Smart edge-AI traffic monitoring system for connected vehicles. *IEEE Access, 9*, 140193–140206. https://doi.org/10.1109/ACCESS.2021.3119495

[3] Li, Y., Yu, R., Shahabi, C., & Liu, Y. (2018). Diffusion convolutional recurrent neural network: Data-driven traffic forecasting. In *Proceedings of the International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=Sy_TQe-RW

[4] Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2019). Graph WaveNet for deep spatial-temporal graph modeling. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 1907–1913). https://doi.org/10.24963/ijcai.2019/264

[5] Zhang, K., Mao, Y., & Zhang, Y. (2021). Edge-cloud collaboration for video analytics in intelligent transportation systems. *IEEE Transactions on Industrial Informatics, 17*(5), 3478–3486. https://doi.org/10.1109/TII.2020.3018385

[6] Alam, M., Rathore, M. M., & Paul, A. (2023). Federated learning for intelligent transportation systems: Architecture, opportunities, and challenges. *IEEE Transactions on Intelligent Transportation Systems, 24*(3), 2748–2759. https://doi.org/10.1109/TITS.2022.3154846

[7] Zheng, Y., Liu, F., & Hsieh, H. P. (2013). U-Air: When urban air quality inference meets big data. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1436–1444). https://doi.org/10.1145/2487575.2488188

[8] Ma, X., Tao, Z., Wang, Y., Yu, H., & Wang, Y. (2015). Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies, 54*, 187–197. https://doi.org/10.1016/j.trc.2015.03.014

[9] Tian, Y., Pan, Y., Lin, H., & Wang, C. (2021). Real-time traffic signal control using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems, 22*(7), 4323–4334. https://doi.org/10.1109/TITS.2020.2983716

[10] Xu, X., Liu, J., Tang, J., & Wang, D. (2022). A cloud-edge collaborative deep learning framework for traffic flow prediction. *Future Generation Computer Systems, 128*, 208–217. https://doi.org/10.1016/j.future.2021.10.030