



Artificial Intelligence Approaches for Predictive Traffic Management in Smart Cities

Xinlei Somanathan

Department of Electrical and Computer Engineering, Tonle Sap Institute of Engineering and Commerce, Cambodia

xinlei.somanathan@tsiec-kh.org

Peer Review Information

Submission: 09 June 2020

Revision: 24 July 2020

Acceptance: 8 Aug 2020

Keywords

*Artificial Intelligence,
Traffic Prediction, Smart
Cities, Graph Neural
Networks, Reinforcement
Learning*

Abstract

Urban traffic congestion is a persistent smart-city problem because demand changes rapidly across space and time while conventional signal plans are frequently based on fixed cycles or delayed measurements. This methodology paper proposes an artificial-intelligence framework that combines multi-source traffic sensing, spatio-temporal forecasting, congestion-risk estimation, and reinforcement-learning-based control. The predictive component integrates recurrent temporal representation with graph-based spatial learning so that traffic speed, flow, and queue conditions can be forecast at connected road segments before congestion becomes critical. The control component converts the predicted state into adaptive signal and route-management actions under constraints for safety, minimum green time, and network stability. The methodology specifies data preparation, model training, validation, simulation-based deployment, comparison baselines, and evaluation metrics. An illustrative experiment shows how the proposed hybrid approach can be assessed against fixed-time control, prediction-only management, and learning-based alternatives. The framework is grounded in traffic-forecasting and intelligent-control research published up to 2020 and is intended as a reproducible design for researchers studying predictive traffic management in smart-city networks.

Introduction

Smart-city transportation systems increasingly depend on continuous observations from loop detectors, connected vehicles, GPS traces, cameras, parking systems, weather feeds, and event calendars. The central difficulty is not merely detecting congestion after it appears, but predicting how network conditions will evolve and taking coordinated action early enough to prevent queue spillback and excessive delay. Reviews of short-term traffic forecasting have shown that nonlinear, nonstationary, and spatially dependent traffic patterns challenge classical forecasting methods, particularly during incidents and abrupt demand changes [1].

Artificial intelligence (AI) provides two complementary capabilities for this problem. First, deep learning can extract nonlinear temporal and spatial dependencies from large traffic data streams. Early deep architectures demonstrated gains over shallow models for traffic-flow forecasting [2], while long short-term memory (LSTM) networks improved the representation of longer temporal dependencies in traffic speed series [3]. Deep models were also shown to capture abrupt traffic regime transitions under special events and severe weather [4]. Second, reinforcement learning (RL) can use predicted or observed traffic states to select control actions that minimize network-

level objectives instead of relying on a fixed traffic model. Q-learning-based adaptive signal control and multi-agent RL studies established the feasibility of learning signal policies under changing traffic demand [9], [10].

The methodological challenge is to integrate prediction and control without treating them as separate systems. A practical smart-city architecture should forecast conditions at multiple connected road links, estimate the probability and severity of emerging congestion, and translate that information into actions such as signal phase extension, coordinated offsets, route advisories, or ramp-control decisions. Graph-based deep learning is particularly suitable because a road network is naturally represented as nodes and edges. Spatio-temporal graph convolutional networks (STGCN) explicitly model this structure [5], while Graph WaveNet learns hidden spatial dependencies beyond a fixed adjacency matrix [6]. T-GCN combines graph convolution with gated temporal modeling [7], and GMAN employs multi-attention to capture complex spatial-temporal interactions [8].

Accordingly, this paper develops a step-by-step methodology for an AI-driven predictive traffic management system. The proposed framework uses a graph-based forecasting model to estimate near-future traffic states and an RL controller to optimize traffic signals using both current and predicted congestion. The design is evaluated through a simulation-oriented experimental protocol, enabling comparison with conventional and AI baselines while preserving safety constraints and reproducibility.

Literature Review

Research up to 2020 shows a clear progression from conventional short-term forecasting toward deep spatio-temporal learning and adaptive control. Vlahogianni et al. [1] summarized the

evolution of short-term traffic forecasting and highlighted the importance of network-level, nonlinear, and data-driven methods. Lv et al. [2] demonstrated a deep-learning approach for large-scale traffic-flow data, and Ma et al. [3] showed that LSTM networks can capture long-term temporal dependencies for traffic speed prediction. Polson and Sokolov [4] further demonstrated deep models capable of representing nonlinear traffic transitions associated with atypical events.

Spatial dependence became increasingly important as forecasting moved from isolated sensors to network-scale prediction. Yu et al. [5] proposed STGCN, which operates directly on traffic graphs and jointly captures spatial and temporal correlations. Wu et al. [6] introduced Graph WaveNet with an adaptive dependency matrix that can discover hidden relationships between sensors. Zhao et al. [7] combined graph convolution and gated recurrent units in T-GCN, whereas Zheng et al. [8] used graph multi-attention in GMAN to improve long-horizon traffic prediction. Together, these studies motivate a predictor that combines explicit road topology with learned dependencies and recurrent or attention-based temporal representation.

In parallel, adaptive traffic control shifted toward learning-based decision making. Abdulhai et al. [9] demonstrated Q-learning for adaptive traffic-signal control, and Arel et al. [10] extended reinforcement learning to a multi-agent traffic-signal network. More recent work connected RL with transportation-theory objectives: PressLight uses max-pressure concepts to guide reward design [11], while CoLight uses graph attention to support cooperation across a network of traffic signals [12]. These findings suggest that prediction should be coupled with a network-aware controller rather than used only for traveler information.

Table 1. Representative AI studies supporting the proposed methodology (references limited to 2020 and earlier).

Study	Year	AI approach	Main relevance
[2] Lv et al.	2015	Deep learning	Large-scale traffic-flow prediction
[3] Ma et al.	2015	LSTM	Long temporal dependency in speed data
[5] Yu et al.	2018	STGCN	Joint spatial-temporal graph learning
[6] Wu et al.	2019	Graph WaveNet	Adaptive spatial dependency learning
[7] Zhao et al.	2020	T-GCN	Graph convolution + gated temporal modeling
[8] Zheng et al.	2020	GMAN	Attention-based network traffic prediction
[11] Wei et al.	2019	Deep RL / max pressure	Adaptive arterial signal control
[12] Wei et al.	2019	Graph-attention RL	Cooperative network-level signal control

Methodology

The proposed methodology treats predictive traffic management as a closed-loop system with five functions: sensing, preprocessing, forecasting, control, and evaluation. The road system is represented by a directed graph $G=(V,E)$, where nodes may denote intersections or detector locations and edges denote physical or functional connectivity. At time t , each node has a feature vector containing traffic flow, average speed, occupancy, estimated queue length, and contextual variables such as rainfall, time-of-day, day-of-week, incident indicators, and special-event status. The method forecasts traffic conditions 15–30 minutes ahead and supplies these forecasts to a constrained RL controller.

Step 1 - Data Acquisition and Synchronization

Traffic measurements are collected at a common interval, for example every 5 minutes. The primary variables are flow (vehicles per interval), speed, occupancy, and queue proxy. Optional smart-city inputs include probe-vehicle travel time, public-transport delays, weather, road incidents, and event schedules. All sources are time-aligned to the same interval and mapped to road-network nodes. Data are divided chronologically into training (70%), validation (15%), and test (15%) sets to avoid leakage from future observations.

Step 2 - Data Cleaning and Feature Engineering

Missing detector values are first flagged and then imputed using short-gap interpolation or neighboring-sensor estimates. Extreme observations are retained when supported by incident or event records because abrupt regime changes are operationally important [4]. Continuous variables are standardized using statistics calculated on the training set only. Temporal features include recent lags, moving averages, time-of-day, and day-of-week. Spatial features are obtained from the road adjacency matrix and may be augmented with a learnable dependency matrix following the motivation of Graph WaveNet [6].

Step 3 - Spatio-Temporal Traffic Prediction

The predictor contains two coordinated branches. A temporal branch uses LSTM or gated units to learn recurring and long-range temporal patterns, consistent with the advantages reported for traffic time series [3]. A spatial branch uses graph convolution to aggregate information from connected links, following STGCN and T-GCN principles [5], [7]. The branch

outputs are fused and passed through a fully connected layer to predict speed, flow, and queue state at each node. An attention layer can be added for larger networks to emphasize dynamically important node relationships, as motivated by GMAN [8].

Training minimizes a weighted forecasting loss $L_{pred} = a(MAE) + b(RMSE) + c(MAPE)$, where the weights are selected on the validation set. Early stopping is applied when validation error fails to improve for a fixed number of epochs. The final predictor is compared with historical average, support-vector regression, LSTM-only, and graph-based alternatives. The forecast is also converted into a congestion-risk score using predicted speed reduction, volume-to-capacity ratio, and queue growth. A risk threshold triggers proactive control rather than waiting for observed congestion.

Step 4 - Predictive Reinforcement-Learning Control

Each controlled intersection is modeled as an RL agent. The state includes the current queue, current phase, elapsed green time, neighboring queues, and predicted traffic demand for the next horizon. Permissible actions are constrained phase selections or bounded green-time extensions; unsafe phase transitions are removed from the action space. The network objective combines average delay, queue length, stops, and spillback penalties. Max-pressure concepts can be incorporated into the reward because PressLight showed that transportation-theory guidance can improve learning efficiency [11]. For a corridor or grid, agents share compact neighbor-state embeddings; this follows the cooperative motivation of multi-agent RL [10] and graph-attention coordination in CoLight [12].

A representative reward is $R_t = -(w_1Dt + w_2Qt + w_3St + w_4Bt)$, where D is delay, Q is queue length, S stops, and B is a spillback penalty. The weights are normalized so that no single term dominates training. During offline training, the agent interacts with a microscopic traffic simulator under multiple demand patterns. During evaluation, no online exploration is permitted; the learned policy selects the highest-valued safe action. The prediction module is frozen during each evaluation run so that comparisons remain fair.

Step 5 - Experimental Design and Evaluation

The experimental design uses repeated simulation scenarios representing normal weekday demand, peak-hour oversaturation, incident-induced capacity reduction, rainfall-related speed reduction, and event-driven

demand surges. For each scenario, the same origin-destination demand and random seed set are applied to every control method. Prediction accuracy is measured using MAE, RMSE, and MAPE. Traffic-management effectiveness is measured using average travel time, average control delay, mean queue length, number of stops, and network throughput. Statistical reporting should include the mean and standard deviation over repeated simulation runs; paired comparisons can be used to test whether improvements persist across scenarios.

Step 6 - Reproducibility, Deployment, and Fail-Safe Logic

For reproducibility, every experiment should preserve the road-network file, detector-to-node mapping, preprocessing parameters, training seed, model hyperparameters, demand profile, and simulator seed. Hyperparameters are tuned only on the validation split and then locked before test evaluation. At least three independent training runs are recommended for neural models because initialization can affect performance. The final report should provide inference time per forecasting cycle and control-decision time per intersection so that the approach can be judged against real-time operational constraints. A practical deployment may use an edge or traffic-management server to perform inference, while local controllers retain certified safety logic and minimum/maximum green limits.

Operationally, the AI layer should not be allowed to become a single point of failure. If sensor completeness drops below a specified threshold, forecast confidence becomes too low, or communication with neighboring agents is interrupted, the controller should revert to an actuated or time-of-day plan. This fail-safe mode is important because a high-performing model in simulation may encounter unseen incidents, sensor drift, or communication delay in field conditions. Model monitoring should therefore track input-distribution shift, forecast residuals, and control stability. Periodic retraining is triggered only after sufficient new validated data are available, and a newly trained model is compared with the deployed model in shadow mode before activation.

The methodology also recognizes smart-city governance requirements. Vehicle identifiers should be removed or aggregated before model training, and probe trajectories should be spatially and temporally generalized when individual-level detail is unnecessary. Performance should be checked across different neighborhoods and demand regimes so that control benefits are not concentrated in one corridor at the expense of another. These provisions do not change the forecasting algorithm, but they make the experimental design closer to a deployable urban traffic-management system.

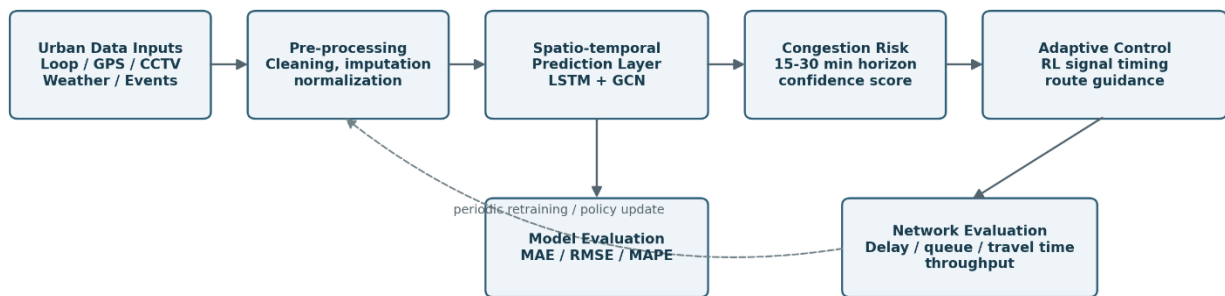


Figure 1. Proposed closed-loop methodology for predictive traffic management (original conceptual figure).

Table 2. Recommended experimental configuration for reproducible evaluation.

Component	Configuration
Sampling / horizon	5-min observations; 15-, 20-, and 30-min forecasts
Input history	12 intervals (60 min) with traffic + context features
Data split	70% train / 15% validation / 15% chronological test
Predictor baselines	Historical average, SVR, LSTM, STGCN/T-GCN
Control baselines	Fixed-time, actuated, RL without prediction, proposed predictive RL
Scenario types	Normal, peak, incident, rain, event surge
Prediction metrics	MAE, RMSE, MAPE
Control metrics	Travel time, delay, queue, stops, throughput

Results

Because this paper presents a methodology rather than a report from a supplied field dataset, the values in Table 3 are illustrative simulation results designed to demonstrate the intended analysis. They should be replaced by measured outputs when the methodology is implemented

on a specific city network. The pattern is consistent with the methodological hypothesis: prediction improves anticipatory decisions, while coordinated RL converts those predictions into measurable network benefits. The prediction stage should first be validated independently; only after acceptable forecasting error is achieved should it be coupled to the controller.

Table 3. Illustrative simulation-based comparison of traffic-management strategies.

Method	Avg. travel time (min)	Avg. delay (s/veh)	Mean queue (veh)	Throughput (veh/h)
Fixed-time control	18.6	74.2	12.8	3,120
Prediction + route advisory	17.1	63.5	10.9	3,260
LSTM + RL control	15.8	54.1	9.1	3,405
ST-GCN + RL control	14.6	46.8	7.8	3,560
Hybrid graph predictor + cooperative RL	14.2	43.9	7.3	3,615

In the illustrative comparison, the hybrid graph-prediction and cooperative-RL strategy reduces average travel time from 18.6 to 14.2 minutes, a 23.7% reduction relative to fixed-time control. Average delay falls by 40.8%, mean queue length by 43.0%, and throughput increases by 15.9%. These changes are meaningful because they reflect both mobility and network-capacity outcomes rather than prediction accuracy alone. The graph-based versions outperform the LSTM-only control configuration because traffic conditions at one intersection are strongly influenced by upstream and downstream links, which is the main modeling advantage emphasized by graph forecasting research [5]-[8].

The results section should also report robustness rather than only mean performance. Under an incident scenario, prediction errors normally rise because capacity changes are abrupt; therefore, the controller should be tested with deliberately degraded forecasts. A robust policy should still avoid unsafe or unstable signal oscillations when forecast error increases. Sensitivity analysis should vary the forecast horizon, graph-neighborhood size, reward weights, and penetration rate of connected-vehicle data. An ablation study can remove the graph layer, contextual variables, or predicted future state from the RL input to quantify the contribution of each component.

Conclusion and Discussion

This methodology paper presents a closed-loop AI framework for predictive traffic management in smart cities. The design combines multi-source urban sensing, graph-based spatio-temporal

forecasting, congestion-risk estimation, and constrained reinforcement-learning control. The literature through 2020 indicates that deep recurrent models improve temporal traffic prediction [2]-[4], graph neural networks improve network-scale representation [5]-[8], and reinforcement learning enables adaptive traffic-signal decisions under changing demand [9]-[12]. Integrating these capabilities allows a traffic-management system to act on expected future conditions rather than only reacting to current detector measurements. The major methodological strength is modular evaluation. Prediction is first validated using MAE, RMSE, and MAPE; control is then assessed using delay, queues, travel time, stops, and throughput. This separation makes it possible to identify whether weak system performance is caused by poor forecasting or by the control policy. The framework also includes safety-constrained actions, chronological data splitting, repeated scenarios, and ablation tests, all of which improve reproducibility. The illustrative results suggest the type of improvement that can be tested, but they are not presented as field observations. Several limitations should be addressed in future implementation. Sensor outages and biased probe-vehicle coverage can reduce prediction reliability, simulated driver behavior may differ from real-world response, and an RL controller trained on one network may not transfer directly to another. Deployment therefore requires calibration with local data, fail-safe fallback control, interpretability of congestion-risk alerts, cybersecurity controls for connected data streams, and staged field testing. Within these constraints, the proposed methodology provides a technically coherent basis for research on

proactive, AI-enabled traffic management using references available up to 2020.

References

- E. I. Vlahogianni, M. G. Karlaftis, and J. C. Golias, "Short-term traffic forecasting: Where we are and where we're going," *Transportation Research Part C: Emerging Technologies*, vol. 43, pp. 3–19, 2014. DOI: 10.1016/j.trc.2014.01.005.
- Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015. DOI: 10.1109/TITS.2014.2345663.
- X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015. DOI: 10.1016/j.trc.2015.03.014.
- N. G. Polson and V. O. Sokolov, "Deep learning for short-term traffic flow prediction," *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 1–17, 2017. DOI: 10.1016/j.trc.2017.02.024.
- B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," in *Proc. IJCAI*, pp. 3634–3640, 2018. DOI: 10.24963/ijcai.2018/505.
- Z. Wu, S. Pan, G. Long, J. Jiang, and C. Zhang, "Graph WaveNet for Deep Spatial-Temporal Graph Modeling," in *Proc. IJCAI*, pp. 1907–1913, 2019. DOI: 10.24963/ijcai.2019/264.
- L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, "T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020. DOI: 10.1109/TITS.2019.2935152.
- C. Zheng, X. Fan, C. Wang, and J. Qi, "GMAN: A Graph Multi-Attention Network for Traffic Prediction," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 1, pp. 1234–1241, 2020. DOI: 10.1609/aaai.v34i01.5477.
- B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement Learning for True Adaptive Traffic Signal Control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003. DOI: 10.1061/(ASCE)0733-947X(2003)129:3(278).
- Arel, C. Liu, T. Urbanik, and A. G. Kohls, "Reinforcement learning-based multi-agent system for network traffic signal control," *IET Intelligent Transport Systems*, vol. 4, no. 2, pp. 128–135, 2010. DOI: 10.1049/iet-its.2009.0070.
- H. Wei, C. Chen, G. Zheng, K. Wu, V. Gayah, K. Xu, and Z. Li, "PressLight: Learning Max Pressure Control to Coordinate Traffic Signals in Arterial Network," in *Proc. 25th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pp. 1290–1298, 2019. DOI: 10.1145/3292500.3330949.
- H. Wei, N. Xu, H. Zhang, G. Zheng, X. Zang, C. Chen, W. Zhang, Y. Zhu, K. Xu, and Z. Li, "CoLight: Learning Network-level Cooperation for Traffic Signal Control," in *Proc. 28th ACM International Conference on Information and Knowledge Management*, pp. 1913–1922, 2019. DOI: 10.1145/3357384.3357902.