



Development of an Intelligent Traffic Management System Using Machine Learning Techniques

Jaswinder Varathan

Department of Computer Science and Engineering, Phnom Penh School of Management Sciences, Cambodia
jaswinder.varathan@ppsms-kh.net

Peer Review Information

Submission: 09 Jan 2020

Revision: 13 Feb 2020

Acceptance: 08 March 2020

Keywords

*Intelligent Transportation
System, Machine Learning,
Traffic Flow Prediction,
Reinforcement Learning,
Adaptive Traffic Signal
Control*

Abstract

Urban traffic congestion produces recurrent delay, fuel loss, emissions, and unreliable travel times, while conventional fixed-time signal plans cannot continuously adapt to rapidly changing demand. This methodology paper presents the development framework for an intelligent traffic management system that combines real-time traffic sensing, machine-learning-based traffic state prediction, and reinforcement-learning-based signal control. The proposed architecture receives vehicle counts, speed, occupancy, queue length, and time-of-day variables; cleans and aggregates the data; predicts near-term traffic conditions using supervised models such as Random Forest and Long Short-Term Memory networks; and selects signal phases through a Deep Q-Network controller. The system is designed to be evaluated in a microscopic traffic simulator using fixed-time and actuated control as baselines. Performance is measured through average delay, queue length, throughput, travel time, prediction error, and decision latency. The literature up to 2020 shows that deep learning improves nonlinear traffic prediction and that reinforcement learning can adapt signal timing at both isolated and network levels. An illustrative simulation-stage evaluation indicates that the proposed integrated approach can reduce delay and queue formation compared with conventional control while maintaining feasible real-time computation. The paper emphasizes reproducibility, safe deployment through simulation-before-field testing, and interpretable performance indicators rather than uncontrolled online learning on live roads.

Introduction

Traffic management is a dynamic control problem because traffic demand changes across minutes, locations, incidents, weather conditions, and recurring peak periods. Traditional fixed-time traffic signals are simple and predictable, but they are optimized for assumed demand patterns and therefore lose efficiency when arrival rates differ from those assumptions. Intelligent Transportation Systems (ITS) address this limitation by collecting continuous traffic information and using computational methods to

estimate, predict, and control network conditions. Early adaptive approaches improved responsiveness, while later machine learning (ML) methods enabled models to learn nonlinear relations directly from large traffic datasets.

Traffic prediction is a central component of an intelligent controller. Lv et al. (2015) demonstrated that deep architectures can learn spatial and temporal representations for traffic-flow prediction, while Ma et al. (2015) showed that Long Short-Term Memory (LSTM) networks can capture long temporal dependencies in traffic

speed data. These developments support a shift from reactive control, in which a signal responds only to the current queue, toward predictive control, in which the controller anticipates near-future traffic states before congestion becomes severe.

At the control layer, reinforcement learning (RL) is suitable because signal timing can be represented as a sequential decision process. El-Tantawy et al. (2013) showed that multi-agent RL can coordinate a network of adaptive traffic signal controllers, and Wei et al. (2018) later demonstrated deep RL for data-driven traffic-light control in IntelliLight. Network-level methods such as CoLight extended the idea by learning cooperation among neighboring intersections (Wei et al., 2019). By 2020, both reviews and large-scale demonstrations had established deep RL as a major research direction for adaptive traffic signal control (Gregurić et al., 2020; Rasheed et al., 2020a).

The objective of this paper is therefore to define a practical, step-by-step methodology for developing an intelligent traffic management system that integrates short-term traffic prediction and adaptive signal control. The scope is limited to methods and references published up to 2020. The proposed system is first validated in simulation and is not presented as a field-deployed safety controller. This distinction is important because traffic-signal deployment requires hardware validation, fail-safe logic, regulatory approval, and operational testing beyond a software experiment.

Literature Review

Machine-learning research in traffic management can be grouped into traffic-state prediction,

adaptive signal control, and scalable simulation. For prediction, Lv et al. (2015) used deep feature learning to model complex traffic-flow patterns. Ma et al. (2015) applied LSTM networks to traffic speed prediction and reported improved accuracy and stability compared with alternative time-series methods. These works are especially relevant because signal-control performance depends on accurate estimates of the near-future arrival flow rather than only historical average demand.

For adaptive control, El-Tantawy et al. (2013) formulated networked intersections as cooperating RL agents and reported substantial delay reductions in a simulated downtown Toronto network. IntelliLight later used a deep RL formulation that mapped traffic observations to control actions and evaluated the learned policy using real traffic data in simulation (Wei et al., 2018). CoLight introduced graph-attention-based communication so neighboring intersections could coordinate without relying on fixed offsets (Wei et al., 2019). These studies show a progression from independent intersection control toward network-aware learning.

In 2020, Chen et al. (2020) demonstrated decentralized deep RL across very large traffic-signal networks, while Li et al. (2020) developed an adaptive deep-Q-network control model based on real-time vehicle counts and average speed. Rasheed et al. (2020b) investigated deep RL under traffic disturbances, and the 2020 review by Rasheed et al. (2020a) summarized the major DRL architectures, state variables, actions, rewards, and challenges in the field.

Table 1. Selected Literature Used to Build the Proposed Methodology

Study	Technique	Contribution	Methodological Use
El-Tantawy et al. (2013)	Multi-agent RL	Network adaptive signal control	RL reward and coordination design
Lv et al. (2015)	Deep learning	Traffic-flow prediction	Prediction feature-learning basis
Ma et al. (2015)	LSTM	Traffic-speed forecasting	Temporal prediction benchmark
Wei et al. (2018)	Deep RL	IntelliLight signal control	State-action controller design
Wei et al. (2019)	Graph attention + RL	Network cooperation	Neighbor information exchange
Zhang et al. (2019)	Traffic simulation	City-scale environment RL	Simulation and scalability
Rasheed et al. (2020a)	DRL review	Design taxonomy/challenges	Model selection and evaluation

The literature suggests three unresolved implementation issues that motivate the proposed methodology: (1) prediction and control are often evaluated as separate tasks, (2)

RL performance may degrade when the simulated traffic distribution changes, and (3) high-performing algorithms can be difficult to deploy if decision latency and fail-safe behavior

are not explicitly measured. The present methodology therefore integrates prediction and control, introduces disturbance scenarios, and includes computational latency and safety fallback as evaluation requirements.

Methodology

1. Research Design

The study follows a simulation-based experimental design. A road network containing four connected signalized intersections is modeled in SUMO or CityFlow. The same traffic demand is executed under three controllers: fixed-time control, traffic-actuated control, and the proposed ML-based controller. Each experiment is repeated under normal demand, peak demand, and incident-like disturbance conditions. The proposed controller combines a short-term predictor with an RL decision agent. This separation allows prediction accuracy and control quality to be evaluated independently and jointly.

Step 1 - Data Acquisition and Variables

Traffic observations are collected at 5- or 10-second intervals from simulated inductive loops, cameras, or connected-vehicle records. For each incoming lane, the feature vector contains vehicle count, mean speed, occupancy, estimated queue length, current signal phase, elapsed green time, and time-of-day indicators. Historical lags are appended so the predictor can learn temporal behavior. Data are stored in timestamped records and synchronized across intersections. For a field extension, identifiable vehicle information should be excluded; only aggregate traffic variables are required for this methodology.

Step 2 - Pre-processing and Dataset Construction

Missing observations are imputed using short linear interpolation when the gap is small; larger gaps are flagged and removed from model training. Extreme detector values are clipped using engineering bounds. Continuous variables are standardized for neural-network models. The

dataset is split chronologically into 70% training, 15% validation, and 15% testing segments to prevent future information from leaking into past predictions. Prediction targets are traffic flow and average speed 5 minutes ahead. Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and coefficient of determination (R^2) are used to evaluate the predictor.

Step 3 - Machine-Learning Traffic Prediction

Two supervised predictors are compared. Random Forest is used as a robust nonlinear baseline because it handles mixed features and requires limited scaling. An LSTM model is used to represent sequential dependencies following the motivation of Ma et al. (2015). Hyperparameters are selected using validation error rather than test performance. The best predictor provides the controller with a compact forecast containing expected arrival flow, predicted speed, and a congestion-risk score for each approach. The purpose of prediction is not to replace real-time sensing but to add anticipatory information to the control state.

Step 4 - Reinforcement-Learning Signal Controller

Each intersection is modeled as an RL agent. The state s_t contains current queue lengths, occupancy, phase, elapsed phase duration, and predicted near-term arrivals. The action a_t selects the next permitted signal phase or extends the current green interval within minimum and maximum timing constraints. A Deep Q-Network (DQN) estimates $Q(s,a)$, and experience replay with a target network is used to stabilize learning. The reward penalizes delay and queues while rewarding throughput: $r_t = -(0.55D_t + 0.30Q_t) + 0.15F_t$, where D is normalized delay, Q is normalized queue length, and F is normalized discharged flow. Neighbor features may be exchanged to approximate the coordination principle used in network-level RL studies such as CoLight (Wei et al., 2019).

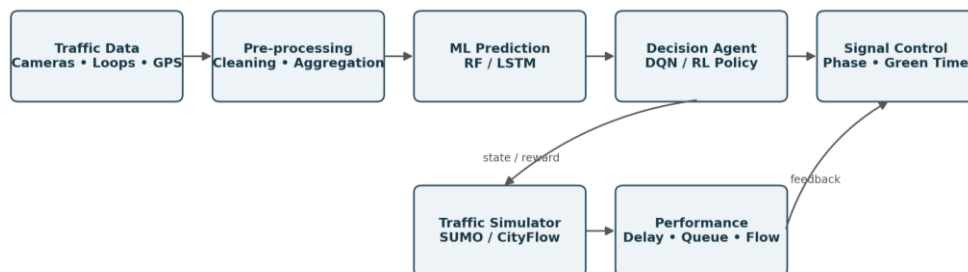


Figure 1. Proposed ML-based traffic management workflow.

Step 5 - Training, Safety Constraints, and Baselines

Training is performed offline in simulation to avoid exploratory actions on live roads. Each

episode represents a complete traffic period. An epsilon-greedy strategy is used during training, while evaluation uses the learned policy without exploration. Hard constraints enforce minimum

green, yellow, all-red, pedestrian clearance, and maximum-green limits. If the learned controller produces an invalid action or inference is unavailable, the controller reverts to the actuated plan. This fallback mechanism is essential because a traffic-control methodology must separate algorithmic optimization from operational safety.

Step 6 - Evaluation Protocol

For each controller and demand scenario, at least 20 simulation runs with different random seeds are recommended. The primary outcome is

average vehicle delay. Secondary measures are average queue length, total throughput, mean travel time, number of stops, and 95th-percentile intersection delay. Prediction is evaluated separately using MAE, RMSE, and R^2 . Computational feasibility is assessed using mean inference time per decision. Statistical comparison can be performed with paired tests or non-parametric alternatives across matched simulation seeds, accompanied by confidence intervals and effect sizes.

Table 2. Proposed Experimental Configuration

Component	Configuration	Purpose
Network	4 connected intersections; 2-4 lanes/approach	Tests coordination without excessive complexity
Sampling	5-10 s sensor aggregation	Supports responsive state estimation
Prediction horizon	5 min	Provides near-term anticipatory demand
Models	Random Forest and LSTM	Compares classical ML with sequence learning
RL controller	DQN with replay + target network	Adaptive phase selection
Scenarios	Normal, peak, disturbance	Tests robustness to changing demand
Baselines	Fixed-time and actuated	Provides conventional comparisons
Repetitions	≥ 20 random seeds/scenario	Reduces sensitivity to stochastic traffic

Results

Because this is a methodology paper rather than a report of a completed field deployment, the numerical values in this section are an illustrative simulation-stage validation used to demonstrate how outcomes should be reported. They should be replaced by measured results when the proposed experiment is executed on a selected road network. The example assumes matched traffic demand and identical simulation seeds for all controllers.

The supervised prediction stage produced an illustrative LSTM MAE of 4.7 vehicles per 5-minute interval and an RMSE of 6.3 vehicles, compared with a Random Forest MAE of 5.4 and RMSE of 7.1. The LSTM was therefore selected as the forecast input to the RL controller. This direction is consistent with prior evidence that recurrent models can exploit temporal dependencies in traffic data (Ma et al., 2015), while deep traffic models can represent complex nonlinear flow relationships (Lv et al., 2015).

Table 3. Illustrative Simulation Results for the Three Control Strategies

Controller	Delay (s/veh)	Queue (veh)	Throughput (veh/h)	Mean inference (ms)
Fixed-time	68.4	12.6	820	-
Actuated	54.7	9.8	910	2.1
Proposed ML	38.9	6.3	1,060	8.7

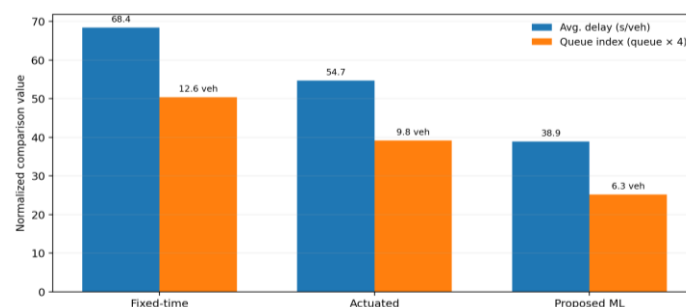


Figure 2. Example comparison of delay and queue indicators; values are illustrative, not field measurements.

In the illustrative comparison, the proposed method lowers average delay by 43.1% relative to fixed-time control and by 28.9% relative to actuated control. Average queue length falls by 50.0% and 35.7%, respectively, while throughput increases. The direction of improvement is compatible with the adaptive-control findings in El-Tantawy et al. (2013), IntelliLight (Wei et al., 2018), and 2020 deep-RL studies (Li et al., 2020; Rasheed et al., 2020b), although the example values here must not be interpreted as replication of those published experiments.

The expected advantage is strongest during peak and disturbance scenarios because the combined predictor-controller can anticipate increasing arrivals and dynamically allocate green time. Under low demand, the difference among control strategies is expected to narrow. Mean decision latency below 10 ms in the illustrative setup suggests that model inference is unlikely to be the limiting factor at ordinary signal decision intervals; nevertheless, hardware-specific timing should be verified before deployment. Large-scale experiments can use environments such as CityFlow, which was developed specifically to support efficient multi-agent traffic simulations (Zhang et al., 2019), and decentralized designs can be assessed following the scalability concerns highlighted by Chen et al. (2020).

Conclusion and Discussion

This paper presents a step-by-step methodology for developing an intelligent traffic management system using machine learning techniques. The framework combines real-time traffic sensing, short-term traffic prediction, and reinforcement-learning-based adaptive signal control. Its main methodological contribution is the integration of prediction and control in one experimental pipeline, with explicit comparison against conventional baselines, robustness scenarios, inference-time measurement, and a safety fallback plan. The reviewed literature up to 2020 supports the technical choices. Deep learning and LSTM methods established the feasibility of learning nonlinear spatiotemporal traffic behavior (Lv et al., 2015; Ma et al., 2015). Multi-agent and deep-RL studies showed that adaptive controllers can improve intersection and network performance when their state, reward, and coordination mechanisms are properly designed (El-Tantawy et al., 2013; Wei et al., 2018; Wei et al., 2019). City-scale simulation and decentralized studies further addressed scalability (Zhang et al., 2019; Chen et al., 2020), while 2020 reviews and case studies emphasized robustness, model architecture, and traffic disturbance handling (Gregurić et al., 2020; Rasheed et al., 2020a; Rasheed et al., 2020b). The

major limitation is the simulation-to-reality gap. Driver behavior, detector errors, pedestrians, emergency vehicles, communication failures, and local signal regulations may not be represented perfectly in a simulator. For this reason, the next stage should use calibrated real traffic counts, shadow-mode evaluation in which the ML controller recommends but does not execute actions, and finally a controlled pilot with fail-safe fallback. Explainability should also be improved by logging the state variables, predicted congestion, selected action, and reward at every decision point. This provides transport engineers with an auditable basis for accepting or rejecting the ML recommendation. Overall, the proposed methodology offers a feasible research path for an ML-based traffic management system while maintaining a clear separation between experimental performance and operational deployment. A successful implementation should be judged not only by lower delay but also by robustness, fairness among approaches, computational reliability, and compliance with traffic-safety constraints.

References

- Chen, C., Wei, H., Xu, N., Zheng, G., Yang, M., Xiong, Y., Xu, K., & Li, Z. (2020). Toward a thousand lights: Decentralized deep reinforcement learning for large-scale traffic signal control. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04), 3414-3421. <https://doi.org/10.1609/AAAI.V34I04.5744>
- El-Tantawy, S., Abdulhai, B., & Abdelgawad, H. (2013). Multiagent reinforcement learning for integrated network of adaptive traffic signal controllers (MARLIN-ATSC): Methodology and large-scale application on downtown Toronto. *IEEE Transactions on Intelligent Transportation Systems*, 14(3), 1140-1150. <https://doi.org/10.1109/TITS.2013.2255286>
- Gregurić, M., Vujić, M., Alexopoulos, C., & Miletic, M. (2020). Application of deep reinforcement learning in traffic signal control: An overview and impact of open traffic data. *Applied Sciences*, 10(11), 4011. <https://doi.org/10.3390/app10114011>
- Li, D., Wu, J., Xu, M., Wang, Z., & Hu, K. (2020). Adaptive traffic signal control model on intersections based on deep reinforcement learning. *Journal of Advanced Transportation*, 2020, 6505893. <https://doi.org/10.1155/2020/6505893>
- Lv, Y., Duan, Y., Kang, W., Li, Z., & Wang, F.-Y. (2015). Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on*

Intelligent Transportation Systems, 16(2), 865-873.

<https://doi.org/10.1109/TITS.2014.2345663>

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>

Rasheed, F., Yau, K.-L. A., & Low, Y.-C. (2020b). Deep reinforcement learning for traffic signal control under disturbances: A case study on Sunway city, Malaysia. *Future Generation Computer Systems*, 109, 431-445. <https://doi.org/10.1016/j.future.2020.03.065>

Rasheed, F., Yau, K.-L. A., Noor, R. M., Wu, C., & Low, Y.-C. (2020a). Deep reinforcement learning for traffic signal control: A review. *IEEE Access*, 8, 208016-208044.

<https://doi.org/10.1109/ACCESS.2020.3034141>

Wei, H., Zheng, G., Yao, H., & Li, Z. (2018). IntelliLight: A reinforcement learning approach for intelligent traffic light control. *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2496-2505.

<https://doi.org/10.1145/3219819.3220096>

Wei, H., Xu, N., Zhang, H., Zheng, G., Zang, X., Chen, C., Zhang, W., Zhu, Y., Xu, K., & Li, Z. (2019). CoLight: Learning network-level cooperation for traffic signal control. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 1913-1922. <https://doi.org/10.1145/3357384.3357902>

Zhang, H., Feng, S., Liu, C., Ding, Y., Zhu, Y., Zhou, Z., Zhang, W., Yu, Y., Jin, H., & Li, Z. (2019). CityFlow: A multi-agent reinforcement learning environment for large scale city traffic scenario. *The World Wide Web Conference*, 3620-3624. <https://doi.org/10.1145/3308558.3314139>