Urban Transport Systems Optimization Through Real-Time Data Analytics

Nikoloz Patatishvili

MSc (Computer Science), School of Informatics and Engineering, Georgian-American University, Tbilisi, Georgia.
Email: *nikoloz.patatishvili@gau.edu.ge*

Besiki Tabatadze

PhD (Applied Mathematics), Professor, European University, Georgian-American University, Tbilisi, Georgia.
Email: *tabatadze.besik@eu.edu.ge*

Abstract

Urban traffic congestion continues to increase travel time, fuel consumption, and environmental impact in modern cities, and Tbilisi is no exception: traffic intensity at signalized intersections varies sharply depending on time of day, direction, and the workday–weekend cycle. This study proposes an integrated, AI-assisted framework that connects real-time route-level traffic observation, short-term forecasting, and adaptive signal control within a single decision-support pipeline. Traffic data were collected at five-minute intervals for twelve origin–destination route pairs surrounding one critical intersection in Tbilisi using the Google Routes API, with automated acquisition and storage implemented through AWS Lambda, EventBridge, and Amazon S3, producing a structured dataset of 23,556 observations and 75 variables. A LightGBM regression model forecasts short-term delay at two horizons (t+15 and t+30 minutes), while a two-stage multi-class classifier separates normal from congested traffic states and, within the congested subset, distinguishes slow traffic from jam conditions. Forecasts are translated into operational signal-control actions through a rule-based decision layer and evaluated in a SUMO/TraCI microscopic simulation environment against a fixed-time baseline, a max-pressure-style pressure-adaptive controller, a bridge-model hybrid controller, and an experimental Deep Q-Network (DQN) reinforcement-learning controller. Results show that LightGBM achieved $R^2 = 0.82$ and $R^2 = 0.77$ at the two forecasting horizons, while the two-stage classifier reached 90.6% overall accuracy. In simulation, the pressure-adaptive controller reduced mean waiting time by approximately 33.6% relative to the fixed-time baseline, while the DQN controller achieved an average reduction of approximately 23.3% across multiple random seeds. The proposed framework demonstrates a reproducible, data-driven pathway from real-time observation to adaptive traffic-signal decision-making, without removing engineering judgment from the control loop.

Keywords: Artificial Intelligence, Intelligent Transportation Systems, Traffic Signal Control, Real-Time Data Analytics, LightGBM, Reinforcement Learning, Max-Pressure Control, Microscopic Traffic Simulation.

Introduction

Managing traffic flow in modern cities is one of the most difficult engineering and analytical problems facing urban infrastructure today. Congested intersections increase travel time, reduce road throughput, raise fuel consumption, and negatively affect air quality and the overall efficiency of urban mobility. For this reason, the transition from traditional fixed-time signal control toward data-driven, adaptive, and explainable control systems is now a technological necessity rather than a purely theoretical ambition.

This study focuses on one critical urban intersection in Tbilisi, where traffic intensity changes sharply depending on the hour of day, the direction of travel, and the route pair under consideration. The central idea of the research is that a traffic system should first be able to forecast the near-term horizon of traffic conditions, then translate that forecast into a signal-control recommendation, and finally validate the resulting control rules in a microscopic simulation environment together with a reinforcement-learning alternative.

This approach gives the study two complementary directions. The first is predictive analytics: the quantitative and class-based estimation of delay using data accumulated in real time. The second is operational control: using the resulting forecast for adaptive phase reallocation and microsimulation-based control evaluation. These two directions form the methodological axis of the research, and their integration — rather than their separate treatment — is the primary contribution of this work.

The classical literature on traffic control long relied on fixed-time and actuated controllers as the starting point from which adaptive systems evolved; approximate dynamic programming was shown to support real-time dynamic green-time allocation as early as 2009 (Cai, Wong, & Heydecker, 2009), and subsequent work extended this direction toward actor-critic learning for real traffic networks under disruption (Aslani, Mesgari, & Wiering, 2017). In parallel, traffic forecasting research has expanded well beyond univariate time-series prediction to include spatio-temporal and origin–destination prediction categories (Du et al., 2025), while systematic reviews note the growing role of machine learning, ensembles, and deep learning in congestion forecasting, alongside a persistent challenge in transferring these forecasts into operational control (A Systematic Literature Review of Traffic Congestion Forecasting, 2025). Reinforcement learning for traffic signal control has attracted particular attention as research moved from isolated intersections toward multi-intersection network settings (Noaeen et al., 2022; Michailidis, Michailidis, Lazaridis, & Kosmatopoulos, 2025), while decentralized max-pressure control remains a widely used and theoretically grounded adaptive baseline (Varaiya, 2013). Despite this progress, a persistent gap remains between systems that rely on aggregated, route-level observations — the kind of data that is realistically available in many cities — and systems that assume detector-level or full vehicle-trajectory data, and between forecasting and control components that are typically studied in isolation (Noaeen et al., 2022; Kurniawan et al., 2025).

This paper proposes an AI-assisted, forecasting-to-control framework that addresses this gap by combining real-time route-level data acquisition, short-term delay forecasting, two-stage traffic-state classification, a rule-based decision layer, and microscopic simulation with an experimental reinforcement-learning controller. The remainder of this paper is organized as follows. Section 1 reviews the background and current challenges of traffic signal control in urban networks. Section 2 presents the proposed framework and describes its architecture. Section 3 describes the case-study data, infrastructure, and experimental design. Section 4 presents the experimental results and discusses their implications, while the final section summarizes the main findings and outlines directions for future research.

1. Traffic Signal Control in Urban Networks: Background and Challenges

1.1. Traditional and Adaptive Signal Control Approaches

Fixed-time and actuated controllers represent the classical starting point of traffic signal management, from which more sophisticated adaptive strategies have since developed. Real-time dynamic allocation of green-phase duration was shown to be achievable using approximate dynamic programming as early as 2009, when the objective was to distribute green time, tune parameters, and update signal plans quickly in response to changing demand (Cai, Wong, & Heydecker, 2009). Subsequent research shifted toward actor-critic learning architectures; an adaptive actor-critic controller evaluated on a real road network demonstrated strong robustness under multiple types of traffic disruption and disturbance (Aslani, Mesgari, & Wiering, 2017), moving the control problem from a one-shot optimization exercise toward a continuously learning process. Among rule-based adaptive strategies, decentralized max-pressure control has become one of the most influential approaches: it balances queue lengths across upstream and downstream links using only local information at each intersection, and offers a theoretically grounded stability guarantee that has made it a common benchmark for both classical and learning-based traffic-signal research (Varaiya, 2013).

1.2. Data-Driven Traffic Forecasting

Contemporary traffic-forecasting research is no longer limited to univariate time-series prediction. A 2025 systematic review covering close to 350 studies groups traffic prediction into three main categories — time-series prediction, spatio-temporal prediction, and origin–destination prediction — and shows that the type of available data and its spatio-temporal dependencies directly determine which modeling approach is most appropriate (Du et al., 2025).

Other systematic reviews confirm the growing importance of machine learning, ensembles, and deep learning for congestion forecasting, while noting that translating these forecasts into operational control decisions remains a persistent challenge, which is precisely where the integration of forecasting and control within a single framework becomes necessary (A Systematic Literature Review of Traffic Congestion Forecasting, 2025). Within this landscape, gradient-boosted tree ensembles such as LightGBM (Ke et al., 2017) provide an efficient and practical alternative to large sequential deep-learning models when the available data is aggregated, feature-engineered, and tabular in structure — conditions that closely match the kind of route-level observation used in the present study.

1.3. Reinforcement Learning in Traffic Signal Control

Reinforcement learning became especially relevant to traffic signal control research once the literature began considering multi-intersection, network-level environments rather than isolated single-node settings. A 2022 systematic review summarizing 160 peer-reviewed studies concludes that the use of deep learning and non-commercial microsimulators has grown markedly within reinforcement-learning-based urban network traffic-signal control, although the field still lacks unified testbeds and a closer connection to transportation engineering practice (Noaeen et al., 2022). More recent reviews emphasize that current reinforcement-learning-based traffic-signal-control research places particular attention on reward design, multi-agent architectures, fair comparison against baselines, safety, and real-world deployment, and that bounded or rule-shielded hybrid systems often integrate more easily into existing infrastructure than fully autonomous reinforcement-learning policies (Michailidis, Michailidis, Lazaridis, & Kosmatopoulos, 2025; Kurniawan et al., 2025). The connection between real-time adaptive control and connected vehicle environments is well illustrated by a systematic review showing that adaptive traffic-signal-control research is primarily oriented toward delay minimization and queue optimization, while coordinated control, heterogeneous traffic, and incomplete real-world data remain open problems (Jing, Huang, & Chen, 2017). Building on this direction, real-time reinforcement-learning-based adaptive control has been shown to reduce queue length and stop delay, particularly when signal plans are combined with speed guidance within a microsimulation environment (Maadi, Stein, Hong, & Murray-Smith, 2022), and deep Q-network agents using a discrete traffic-state encoding have been applied directly to signal-timing optimization within the SUMO microsimulator (Genders & Razavi, 2016).

1.4. Research Gap and Positioning of the Study

A comparative reading of the literature reveals four recurring limitations. First, many studies rely either on detector-level data or on complete vehicle-trajectory information, whereas real cities frequently provide only aggregated route-level or origin–destination observations. Second, forecasting and control are frequently studied as separate research problems rather than as stages of a single pipeline. Third, the transfer of simulation-based results into real operating environments remains limited. Fourth, reinforcement-learning-based systems often perform well on a single metric but remain difficult to interpret in practical terms (Noaeen et al., 2022; Michailidis, Michailidis, Lazaridis, & Kosmatopoulos, 2025; Kurniawan et al., 2025).

The present study is positioned to address this gap directly. Its predictive component is built entirely on real-time, aggregated route-level data; its control component consists of three complementary stages — a rule-based, max-pressure-inspired pressure-adaptive controller, a bridge-model hybrid controller, and an experimental Deep Q-Network reinforcement-learning line. This design positions the proposed framework as an intermediate step between classical adaptive traffic control and fully autonomous, learning-based traffic-signal control.

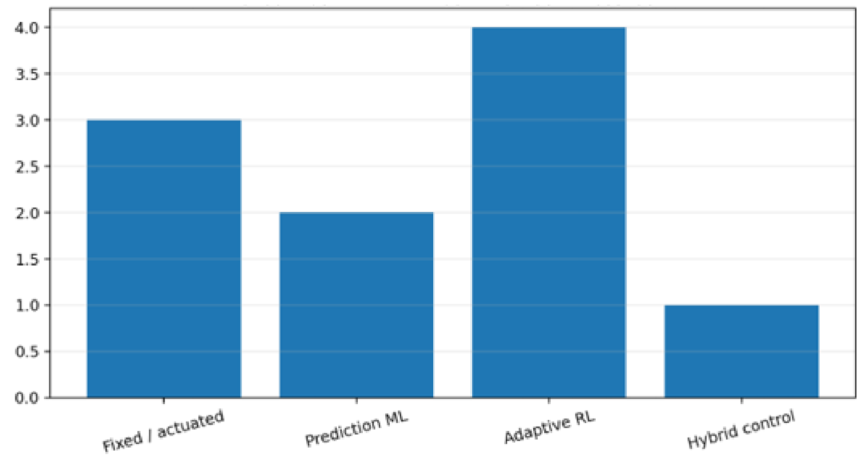

Figure 1. Thematic emphases of the reviewed literature on adaptive traffic signal control and forecasting.

2. Proposed AI-Assisted Forecasting-to-Control Framework

2.1. Overall Framework Architecture

The proposed framework connects five interdependent stages into a single decision-support pipeline: (i) real-time data acquisition from route-level traffic observations, (ii) preprocessing and feature engineering, (iii) short-term forecasting, expressed both as a regression task (delay estimation) and a classification task (traffic-state estimation), (iv) a rule-based decision layer that translates forecasts into signal-control recommendations, and (v) a SUMO/TraCI-based microscopic simulation environment used to evaluate rule-based and reinforcement-learning control strategies under identical conditions. Rather than replacing traffic engineering judgment, the framework is designed to provide transparent, testable, and interpretable control recommendations that can be evaluated before any real-world deployment is considered. The Python-based implementation of the pipeline itself was developed following AI-assisted programming practices, in line with recent conceptual and data-informed observations on integrating AI assistance into practical, reproducible research-software workflows (Tabatadze, 2026).

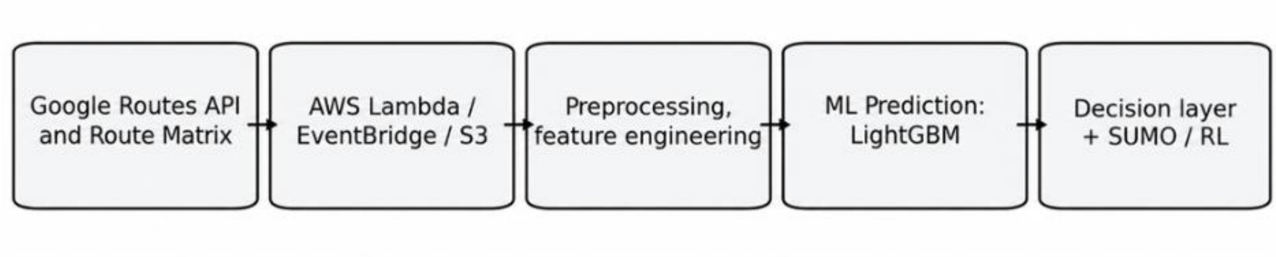

Figure 2. Integrated architecture connecting real-time data, forecasting, decision layer, and control.

2.2. Real-Time Data Acquisition Infrastructure

Traffic observations were collected automatically at five-minute intervals using a serverless cloud pipeline built around AWS Lambda, with scheduled execution managed by Amazon EventBridge and results stored in Amazon S3, ensuring reproducibility, scheduled operation, and consistent data accumulation. At the API level, two complementary methods of the Google Routes API were used. The computeRouteMatrix method processes origin×destination combinations simultaneously and returns distance, duration, and staticDuration values for each combination (Google for Developers, 2025a, 2025b). The traffic-aware routing preference options were used because official documentation indicates that TRAFFIC_AWARE and TRAFFIC AWARE OPTIMAL modes reflect real traffic conditions more accurately, at the cost of additional latency (Google for Developers, 2025c). Where needed, the computeRoutes method was also used; under traffic-aware conditions it returns duration and staticDuration, and, when combined with traffic-aware polylines, additionally provides speedReadingIntervals describing congestion along individual route segments (Google for Developers, 2026). This combination made it possible to construct both aggregate delay variables and segment-level congestion indicators.

2.3. Feature Engineering and Preprocessing

Preprocessing included chronological ordering, normalization of time fields, construction of route-direction identifiers, removal of duplicate records, consistent type definitions, and control of anomalous values. Temporal features included local hour, weekday, peak-hour indicators, and workday/weekend and holiday variables. Because short-term forecasting depends heavily on historical context, lagged delay variables (delay_lag_1, delay_lag_2, delay_lag_3, delay_lag_6, and delay_lag_12) were constructed together with rolling mean and rolling standard-deviation statistics across several time windows. These features allow the model to capture not only the current traffic state but also its short-term dynamics — whether congestion is increasing, stabilizing, or decreasing.

2.4. Short-Term Delay Forecasting Module

The regression component targets quantitative delay forecasting at two horizons, t+15 and t+30 minutes. Dummy, Ridge, Random Forest, and LightGBM (Ke et al., 2017) models were compared, and a chronological train/validation/test split was used so that the evaluation scenario would closely resemble real operational use and information leakage from the future into the training data would be avoided. This choice is consistent with evidence that tree-based gradient-boosting methods tend to perform robustly on aggregated, feature-engineered tabular transportation data (Du et al., 2025).

2.5. Two-Stage Traffic-State Classification

The classification task was formalized in two stages. In the first stage, the model determines whether the forecast traffic state remains normal or transitions into a congested subgroup. In the second stage, applied only within the congested subgroup, the model distinguishes between slow-moving traffic and jam conditions. This division reflects the observation that the boundary between normal and congested states is considerably more stable than the boundary between slow and jam states, and that separating the two decisions produces a more robust overall classifier than a single-step multi-class formulation.

2.6. Decision Layer: From Predictions to Signal-Control Actions

The practical value of a forecast is realized only once it is converted into an operational control rule. For this purpose, a decision layer was constructed that allocates the total cycle duration across signal phases according to the predicted traffic class: a normal state implies minimal intervention, a slow state implies moderate prioritization, and a jam state implies a more aggressive, though still bounded, reallocation of green time. This approach transforms the forecast into an interpretable and controllable signal plan, which is important for practical deployment: the system is not a fully opaque black box, since each predicted class corresponds to an explicit control rule that can also be verified in isolation. The pressure-adaptive controller implemented in this study is inspired by the max-pressure principle, allocating priority according to the difference between upstream and downstream queue pressure at each phase (Varaiya, 2013), while a bridge-model hybrid controller was additionally built to test whether a data-informed, learned controller can be integrated into a live simulation loop.

2.7. SUMO/TraCI Simulation Integration and Reinforcement Learning Environment

The microscopic simulation component of the framework is built on SUMO (Simulation of Urban MObility), a widely used open-source traffic simulator (Lopez et al., 2018). Official SUMO documentation confirms that traffic lights can be controlled through TraCI, a TCP-based client/server interface in which SUMO acts as the server and an external script acts as the controller (SUMO Documentation, 2026a, 2026b, 2026c). This capability made it possible to read current signal phases from Python, identify controlled lanes, modify green-time durations, and perform frame-based visual analysis. Within a realistic network built on OpenStreetMap data, one key signalized intersection was selected, its controllable lanes and phases were identified, and a comparative evaluation was carried out among a fixed-time baseline, the pressure-adaptive controller, the bridge-model hybrid controller, and a Deep Q-Network reinforcement-learning controller (Genders & Razavi, 2016; Maadi, Stein, Hong, & Murray-Smith, 2022). The reinforcement-learning state vector included phase pressures, total waiting time, halting count, queue length, average speed, per-lane vehicle counts, the current phase index, and the most recently applied action; the action space was defined as a discrete reallocation of total green time between two green phases. The reward function was initially based on absolute congestion metrics but was later redefined in a delta-based form that evaluates improvement relative to the previous control step rather than the absolute state alone, which reduced policy collapse and improved the diversification of selected actions.

3. Case Study: Data, Infrastructure, and Experimental Design

3.1. Study Node and Origin–Destination Scheme

The empirical component of the study is built around one critical traffic node in Tbilisi, for which twelve directional origin–destination route pairs were defined. This formalization was important because, rather than relying on full network-wide data, the study focuses on the routes that practically reflect the principal load variations at the node. Representing the problem at the route level makes it possible to simultaneously observe travel duration and delay as well as the differences between specific directions. The choice of a single, high-detail study node is also methodologically appropriate as a step preceding the construction of a citywide adaptive control system.

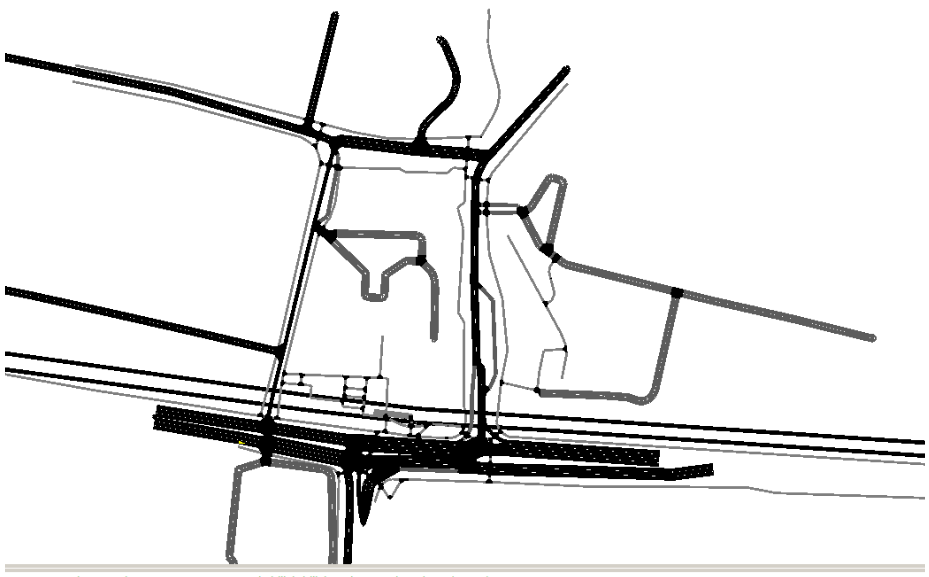

Figure 3. Study intersection represented within the SUMO simulation environment, showing its controllable lanes.

3.2. Dataset Structure and Variables

The final dataset contains 23,556 observations and 75 variables. Each record describes a specific origin–destination pair at a specific time and includes core route-level metrics (distance, duration, staticDuration, delay) as well as additional analytical features: congestion ratio, slow/jam segment indicators, peak-hour indicators, workday/weekend variables, lag features, and rolling-window statistics. An important property of the dataset architecture is that it does not contain individual vehicle trajectories; the study is based on route-level snapshots rather than a vehicle-level GPS stream. This is simultaneously a limitation and a practical strength of the approach: the chosen solution must operate under the conditions in which a city has aggregated, but not fully granular, traffic data — which reflects the actual data-availability constraints faced by many municipal transportation authorities.

Indicator	Value
Number of observations	23,556
Initial number of variables	75
Final modeling features	66
Categorical variables	6
Numerical variables	60
Number of OD pairs	12
Collection interval	5 minutes
Infrastructure	AWS Lambda + EventBridge + S3
Data source	Google Routes API

Table 1. Main characteristics of the dataset.

3.3. Experimental Design and Evaluation Metrics

Experiments followed a chronological split principle: the training set comprised 70% of the data, validation 15%, and testing 15%. This design was chosen so that the evaluation would correspond as closely as possible to a real operational scenario; models did not use future information during training, which was one of the key methodological requirements of the study. Regression tasks were evaluated using MAE, RMSE, and R^2. Classification tasks were evaluated using Accuracy, Precision, Recall, F1, Macro F1, and Weighted F1. Control-module comparison relied on mean waiting time, mean halting count, mean queue length, mean average speed, and mean vehicle count. For the reinforcement-learning line, an additional multi-seed evaluation was carried out to minimize the influence of any single simulation run.

4. Results and Discussion

4.1. Forecasting and Classification Performance

The regression experiments confirmed that tree-based gradient-boosting models are best suited to the task of short-term route-level delay forecasting. LightGBM achieved the lowest error and the highest R^2 at both horizons: MAE = 6.4659, RMSE = 12.1669, and R^2 = 0.8196 at t+15, and MAE = 7.4263, RMSE = 13.3441, and R^2 = 0.7694 at t+30. This result aligns with literature indicating that, for aggregated transportation data, feature-engineered boosting models are often more stable than smaller sequential deep-learning models (Du et al., 2025; A Systematic Literature Review of Traffic Congestion Forecasting, 2025). The classification line showed that the two-stage approach better reflects the nature of traffic classes: the first stage effectively separates congestion cases, while the second stage attempts to differentiate the internal structure of congestion — slow and jam states. Although the second stage is inherently more difficult, its results are sufficiently strong to support the decision layer. The best first-stage LightGBM classifier reached Accuracy = 0.9317, Precision = 0.7034, Recall = 0.9353, and F1 = 0.8030 on the test set; the second-stage slow-vs-jam task achieved Accuracy = 0.7263, Precision = 0.6561, Recall = 0.6667, and F1 = 0.6613. Combining both stages, the final multi-class system reached Accuracy = 0.9057, Macro F1 = 0.7150, and Weighted F1 = 0.9086.

Task	Model	Metrics
Delay forecasting t+15	LightGBM	MAE=6.4659; RMSE=12.1669; R^2=0.8196
Delay forecasting t+30	LightGBM	MAE=7.4263; RMSE=13.3441; R^2=0.7694
Stage 1 classification	LightGBM	Accuracy=0.9317; F1=0.8030
Stage 2 classification	LightGBM	Accuracy=0.7263; F1=0.6613

Final multiclass	Two-stage LightGBM	Accuracy=0.9057; Macro F1=0.7150; Weighted F1=0.9086

Table 2. Main results of delay forecasting and classification.

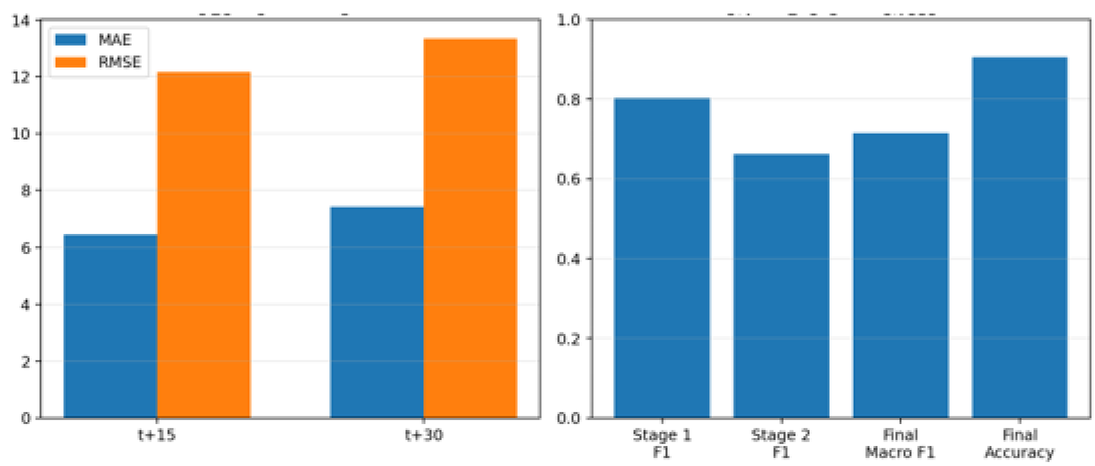

Figure 4. Main results of the forecasting models.

4.2. Comparison of Control Strategies

In the control experiments, the fixed-time baseline produced a mean waiting time of 2572.60 units. The pressure-adaptive method reduced this to 1709.02 units in the same simulation environment — an improvement of approximately 33.6% — while mean queue length and halting count decreased by roughly 8.3% on average. This result shows that phase-aware pressure logic proved, in practice, to be the most stable controller with respect to the primary waiting-time KPI. The bridge-model hybrid controller also showed improvement relative to the fixed-time baseline, although its waiting-time reduction reached only around 10%. This was nonetheless an important intermediate step, as it confirmed that a data-trained controller can be integrated into a live SUMO control loop, even though its strength at this stage remained below that of the pressure-based rule. Multi-seed evaluation of the DQN controller showed that the learning-based controller significantly outperforms the fixed-time baseline: mean waiting time decreased by approximately 23.3% on average across seeds, mean halting count and queue length decreased by approximately 13.4%, and mean average speed increased by approximately 10.1%.

However, compared with the pressure-adaptive method, DQN still lagged on the primary waiting-time metric, despite showing competitive results on secondary metrics.

Controller	Mean waiting time	Mean halting	Mean avg. speed	Mean queue
Fixed-time	2572.60	13.66	10.37	2.28
Pressure-adaptive	1709.02	12.53	9.66	2.09
Hybrid Stage 2	2315.46	12.65	10.04	2.11
DQN-RL (multi-seed mean)	1972.73	11.83	11.41	1.97

Table 3. Final comparison of control strategies.

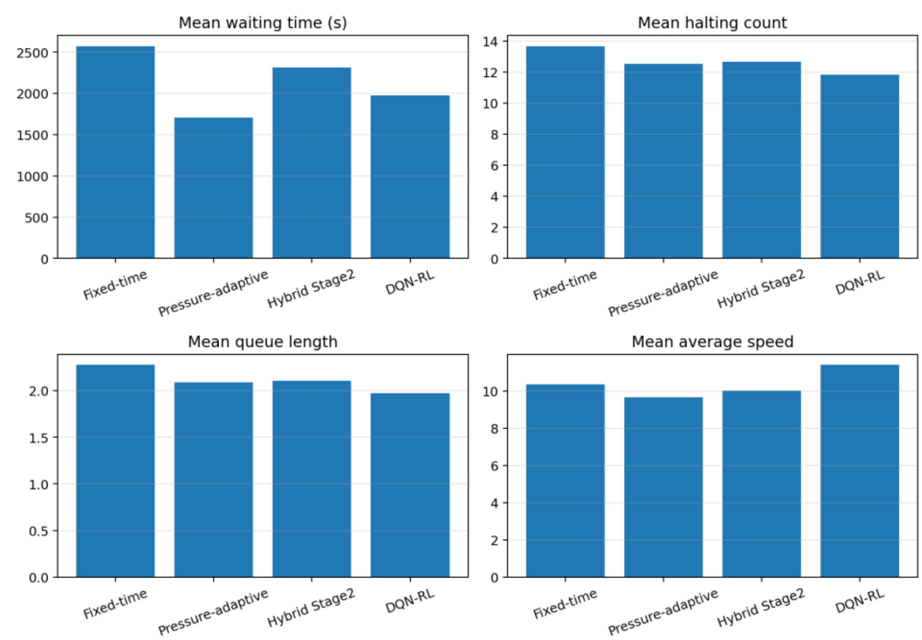

Figure 5. Comparison of the fixed-time, pressure-adaptive, hybrid Stage 2, and DQN-RL controllers.

4.3. Multi-Seed Evaluation of the DQN Controller

Multi-seed evaluation of the DQN line revealed two important observations. First, complete policy collapse was not observed: several distinct action shares appeared in the action distribution, indicating that the agent did not converge to a single repeated action. Second, meaningful variation exists across seeds, indicating that the reinforcement-learning module remains more sensitive to initial conditions and reward design than the rule-based pressure controller. This leads to a practically important conclusion: under the conditions of the present study, the pressure-adaptive controller can be regarded as the best current operational controller, while DQN represents the most promising learning-based direction — one that clearly improves on the baseline but still requires additional tuning to surpass the pressure-based method on the primary waiting-time metric.

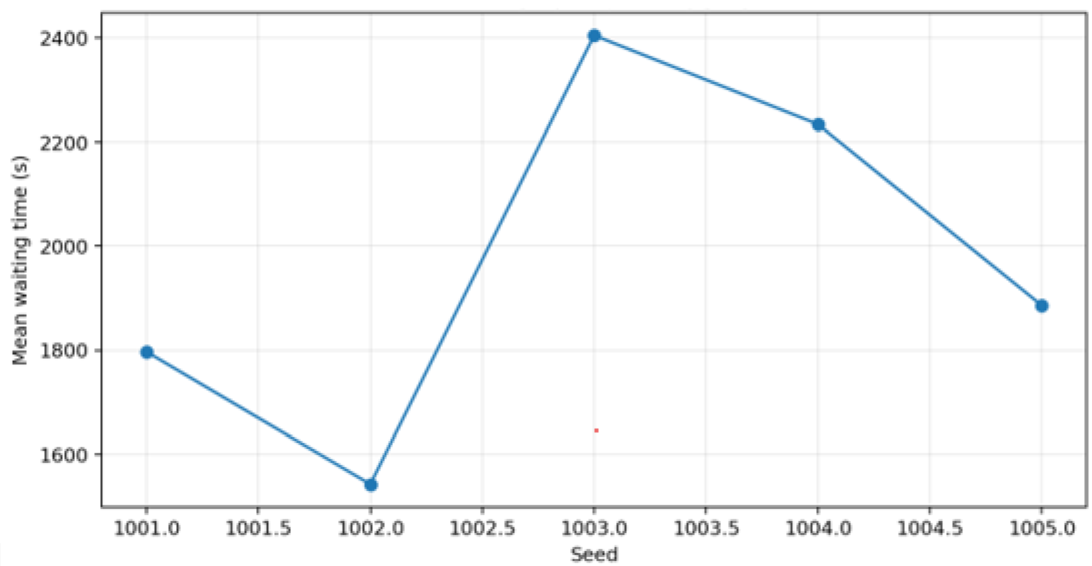

Figure 6. Multi-seed evaluation of the DQN controller by waiting time.

4.4. Visual Analysis of Real Data and Simulation

The visual analysis component was organized into two independent layers. The first concerns the aggregate picture derived from real data: the temporal dynamics of delay and congestion-class changes across route pairs. The second concerns a frame-based comparison of the fixed-time, pressure-adaptive, and DQN controllers within the SUMO environment. A methodological clarification is important here: because the collected data consists of route-level snapshots rather than complete vehicle-trajectory information, an exact vehicle-by-vehicle animation of real traffic was not possible. Real-traffic visualization was therefore performed using a proxy approach reflecting the temporal dynamics of aggregate delay, whereas the SUMO environment allowed for detailed microscopic visualization of the specific node using screenshots and generated GIFs. The visual comparison showed that the pressure-adaptive controller clears accumulated queues more quickly and leaves less residual congestion on problematic approaches, while the DQN controller often achieves a better balance between queue length and speed, though in some scenarios it reduces total waiting time more slowly. This difference is consistent with the quantitative results and shows that different controllers optimize for different key performance indicators.

Conclusion

The continuing growth of urban traffic volumes and the resulting increase in congestion create a growing need for reliable, data-driven tools that can support both the forecasting and the control stages of traffic management. This study proposed an integrated AI-assisted framework for urban traffic-signal optimization that connects real-time route-level data acquisition, short-term delay forecasting, two-stage traffic-state classification, a rule-based decision layer, and SUMO/TraCI-based simulation evaluation, including an experimental reinforcement-learning controller.

The strongest forecasting performance was obtained with a two-stage LightGBM system, which achieved high forecasting accuracy at both the t+15 and t+30 horizons alongside a practically usable traffic-class estimation. In the control stage, the strongest practical result was achieved by the pressure-adaptive controller, which reduced mean waiting time by approximately 33.6% relative to the fixed-time baseline, thereby fulfilling the study's principal operational objective most effectively. The bridge-model hybrid controller and the DQN reinforcement-learning line together demonstrated that learning-based control can indeed be integrated into a microsimulation environment and can meaningfully outperform the fixed-time baseline.

The main contribution of this study is the integration of real-time route-level data, predictive analytics, and adaptive control into a single, transparent, and reproducible framework, rather than treating forecasting and control as separate research problems. The dataset architecture deliberately reflects the aggregated, route-level nature of data that is realistically available in many cities, rather than assuming full vehicle-trajectory access, which strengthens the practical relevance of the proposed approach.

The study nonetheless has several limitations. The temporal coverage of the collected data is sufficient for short-term forecasting and initial control comparisons but does not fully capture seasonal variation, weather effects, incidents, or special-event anomalies. The spatial scope is limited to one critical node and twelve route pairs; while this design enables high analytical detail, network-level generalization would require the inclusion of additional nodes, corridors, and interdependent signals. In addition, the route-level snapshots used in this study do not allow reconstruction of individual vehicle trajectories, which limits the use of some advanced reinforcement-learning state representations, although this constraint closely mirrors the aggregated data conditions available to many municipal transportation authorities. Finally, the reinforcement-learning component shows greater sensitivity to initial conditions and reward design than the rule-based controllers.

Future work should extend the temporal and spatial coverage of the dataset, construct a more realistic multi-phase SUMO topology based on the specific geometry of the study node, and evaluate safer, hybrid reinforcement-learning approaches in which the action space is constrained by rule-based safety layers (Kurniawan et al., 2025). Overall, the proposed framework provides a practical and reproducible foundation for advancing intelligent transportation systems, bridging classical adaptive traffic control and fully autonomous, learning-based traffic-signal control while preserving the transparency and interpretability that real-world deployment requires.

REFERENCES

Aslani, M., Mesgari, M. S., & Wiering, M. (2017). Adaptive traffic signal control with actor-critic methods in a real-world traffic network with different traffic disruption events. *Transportation Research Part C: Emerging Technologies, 85*, 732–752. https://doi.org/10.1016/j.trc.2017.09.020

Cai, C., Wong, C. K., & Heydecker, B. G. (2009). Adaptive traffic signal control using approximate dynamic programming. *Transportation Research Part C: Emerging Technologies, 17*(5), 456–474. https://doi.org/10.1016/j.trc.2009.04.005

Du, K., Guo, X., Li, L., Song, J., Shi, Q., Hu, M., & Fang, J. (2025). Traffic prediction in time series, spatial-temporal, and OD data: A systematic survey. *Journal of Traffic and Transportation Engineering (English Edition), 12*(3), 666–700. https://doi.org/10.1016/j.jtte.2025.03.001

Genders, W., & Razavi, S. (2016). *Using a deep reinforcement learning agent for traffic signal control* (arXiv:1611.01142). arXiv. https://arxiv.org/abs/1611.01142

Google for Developers. (2025a). *Routes API overview*. https://developers.google.com/maps/documentation/routes

Google for Developers. (2025b). *Method: computeRouteMatrix | Routes API*. https://developers.google.com/maps/documentation/routes/reference/rest/v2/TopLevel/computeRouteMatrix

Google for Developers. (2025c). *Set the level of traffic data | Routes API*. https://developers.google.com/maps/documentation/routes/config_trade_offs

Google for Developers. (2026). *Method: computeRoutes | Routes API; Request route polylines*. https://developers.google.com/maps/documentation/routes/reference/rest/v2/TopLevel/computeRoutes

Jing, P., Huang, H., & Chen, L. (2017). An adaptive traffic signal control in a connected vehicle environment: A systematic review. *Information, 8*(3), Article 101. https://doi.org/10.3390/info8030101

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. *Advances in Neural Information Processing Systems, 30*, 3146–3154.

Kurniawan, F., Agustian, H., Dermawan, D., Nurdin, R., Ahmadi, N., & Dinaryanto, O. (2025). Hybrid rule-based and reinforcement learning for urban signal control in developing cities: A systematic literature review and practice recommendations for Indonesia. *Applied Sciences, 15*(19), Article 10761. https://doi.org/10.3390/app151910761

Lopez, P. A., Behrisch, M., Bieker-Walz, L., Erdmann, J., Flötteröd, Y.-P., Hilbrich, R., Lücken, L., Rummel, J., Wagner, P., & Wießner, E. (2018). Microscopic traffic simulation using SUMO. In *Proceedings of the 21st IEEE International Conference on Intelligent Transportation Systems (ITSC)* (pp. 2575–2582). IEEE. https://doi.org/10.1109/ITSC.2018.8569938

Maadi, S., Stein, S., Hong, J., & Murray-Smith, R. (2022). Real-time adaptive traffic signal control in a connected and automated vehicle environment: Optimisation of signal planning with reinforcement learning under vehicle speed guidance. *Sensors, 22*(19), Article 7501. https://doi.org/10.3390/s22197501

Michailidis, P., Michailidis, I., Lazaridis, C. R., & Kosmatopoulos, E. (2025). Traffic signal control via reinforcement learning: A review on applications and innovations. *Infrastructures, 10*(5), Article 114. https://doi.org/10.3390/infrastructures10050114

Noaeen, M., Naik, A., Goodman, L., Crebo, J., Abrar, T., Hossein Abad, Z. S., Bazzan, A. L. C., & Far, B. (2022). Reinforcement learning in urban network traffic signal control: A systematic literature review. *Expert Systems with Applications, 199*, Article 116830. https://doi.org/10.1016/j.eswa.2022.116830

SUMO Documentation. (2026a). *Traffic lights*. https://sumo.dlr.de/docs/Simulation/Traffic_Lights.html

SUMO Documentation. (2026b). *TraCI*. https://sumo.dlr.de/docs/TraCI/index.html

SUMO Documentation. (2026c). *TraCI4Traffic Lights tutorial*. https://sumo.dlr.de/docs/Tutorials/TraCI4Traffic_Lights.html

A systematic literature review of traffic congestion forecasting: From machine learning techniques to large language models. (2025). *Forecasting, 7*(4), Article 142.

Tabatadze, B. (2026). AI-assisted programming in practice: Conceptual foundations and data-informed observations. *Computational and Applied Science, 1*(1), 84–100.

Varaiya, P. (2013). Max pressure control of a network of signalized intersections. *Transportation Research Part C: Emerging Technologies, 36*, 177–195.