

A Machine Learning Framework for Short-Term Traffic Flow Prediction and Adaptive Traffic Signal Optimization Using Internet of Things (IoT) Sensor Data

Zixuan Xie

*Computer Science and Technology, Capital University of Economics and Business, Beijing, China
18516998599@163.com*

Abstract. Traffic demand at urban intersections fluctuates drastically across different signal phases, while conventional fixed-time signal control strategies fail to dynamically adjust green light durations in response to short-term traffic variations. This study addresses two core research questions: whether higher single-point traffic prediction accuracy can guarantee superior signal control performance, and how phase-level prediction uncertainty can be integrated into adaptive green time optimization. One-minute aggregated traffic and signal operation data collected from an urban signalized intersection in Zurich are adopted for model training and verification. Lagged traffic volumes, rolling statistical features, temporal indicators, and signal control parameters are constructed as model input variables. Under consistent chronological train-validation-test data partitioning, four mainstream prediction approaches are comprehensively compared: XGBoost, BiLSTM, rolling-window ARIMA, and the proposed Hurdle-Based Uncertainty-Aware Multi-Task Adaptive Signal Control model (HUMAS-TSC). The proposed HUMAS-TSC model simultaneously predicts traffic demand for signal groups sg2, sg4 and sg6. It leverages hurdle modelling to handle sparse zero-flow observations and incorporates prediction uncertainty quantification to support stable green time allocation. Experimental results reveal that XGBoost achieves the optimal single-point prediction accuracy, with a Mean Absolute Error (MAE) of 0.2925 and a Root Mean Squared Error (RMSE) of 0.3885. In contrast, HUMAS-TSC delivers the best signal control performance: it cuts the average vehicle delay by 25.804% relative to fixed-time control, reduces the high saturation ratio to merely 0.107%, and completely eliminates saturation cap incidents. The core conclusion indicates that single-point prediction accuracy alone cannot determine the overall efficiency of signal control systems, instead, uncertainty-aware phase-level optimization can simultaneously enhance traffic operation efficiency and operational stability. Theoretically, this research clarifies the intrinsic correlation among single-point prediction precision, phase-level predictive uncertainty, and downstream signal control performance. From an engineering perspective, it provides an uncertainty-driven adaptive signal timing solution for urban intersections, which balances traffic passing efficiency, phase service reliability and system operational stability.

Keywords: Adaptive traffic signal control, IoT traffic data, Webster delay, HUMAS-TSC, Prediction uncertainty

1. Introduction

Urban signalized intersections act as recurring bottlenecks within road traffic networks, as traffic demand varies significantly across different approaching lanes and time windows. Traditional fixed-time signal schemes are calibrated based on historical average traffic volumes, which leads to inefficient green time allocation when real-time traffic deviates from preset historical patterns. Adaptive signal control offers a more responsive alternative, yet its operational performance heavily relies on the integrity and reliability of real-time traffic state data [1]. For this reason, extensive research has developed short-term traffic prediction algorithms, covering classical statistical time series models, traditional machine learning methods, and deep neural networks [2]. However, little evidence shows that lower point-prediction error necessarily improves downstream signal control. Moreover, three critical factors have not been sufficiently explored: phase-level sparse traffic observations, predictive uncertainty, and the stability of continuous signal timing adjustments [3].

To address these limitations, this study uses the traffic signal and loop detector dataset collected at a fully actuated urban intersection in Zurich, Switzerland. A machine-learning framework is developed to combine one-step-ahead traffic demand prediction with adaptive green-time optimization. XGBoost, BiLSTM, Rolling ARIMA, and the proposed HUMAS-TSC model are evaluated under the same data split. HUMAS-TSC directly predicts the future demands of signal groups sg2, sg4, and sg6, applies hurdle-based modelling to sparse phase data, and estimates prediction uncertainty. The resulting forecasts are then used in an optimization process that balances demand-related delay against changes in consecutive green-time plans. The framework links phase-level forecasting, uncertainty modelling, and stable signal control using IoT traffic data.

2. Methodology

2.1. Data description and preprocessing

The data used in this study were obtained from the ETH Zurich Research Collection and were originally recorded at a four-leg signalized intersection in Zurich, Switzerland. A detailed description of the dataset is provided by Genser et al. [4]. The intersection operates under a fully actuated signal control system, and the dataset therefore reflects actual variations in both traffic demand and signal operation. The raw data contain timestamped inductive-loop events and signal-state changes for sg2, sg4, and sg6. Since records are generated only when a detector or signal state changes, the observations are irregularly spaced.

The event-based records were aggregated at one-minute intervals. Detector transitions were used to estimate phase-level demand, while signal records provided green duration and green-time ratio. After timestamp normalization, chronological ordering, duplicate checking, and invalid-event removal, the aligned variables formed a continuous multivariable time series.

2.2. Feature engineering

Following Table 1 summarizes the four core categories of input features, their key components, and primary roles for traffic flow prediction and signal optimization.

Table 1. Summary of core categories and description

Feature Category	Key Component	Core Role
Historical Traffic Flow Lags Features	Traffic flow at 5, 10, and 15 minutes before prediction	Capture short-term temporal dependence and recent demand changes
Rolling Statistic Features	Rolling mean, maximum, minimum, and standard deviation	Describe recent traffic levels and variability
Temporal Lag Features	Hour of day, day of week, and weekend indicator	Represent recurring daily and weekly traffic patterns
Signal-control Features	Green duration and green ratio of each signal group	Characterize recent signal operation and distinguish demand variation from discharge effects

2.3. Traffic flow prediction models

2.3.1. Model framework and time-series extensions

Four models are compared in this study: XGBoost, BiLSTM, Rolling ARIMA, and HUMAS-TSC. All models use the same chronological training, validation, and test periods and generate one-step-ahead predictions for the traffic demands of signal groups sg2, sg4, and sg6.

XGBoost is a feature-based machine-learning method that combines multiple regression trees through gradient boosting [5]. Each new tree is trained to reduce the residual errors produced by the preceding trees. Its objective function is expressed as:

$$L_{\text{XGB}} = \sum_{t=1}^N l(q_t, \hat{q}_t) + \sum_{k=1}^K \Omega(f_k) \quad (1)$$

Here, $l(q_t, \hat{q}_t)$ measures prediction error, while $\Omega(f_k)$ controls tree complexity. Using lagged flow, rolling statistics, temporal variables, and signal-control features, XGBoost captures nonlinear interactions in structured data. However, its temporal representation depends on engineered features, and it does not explicitly model cross-phase relationships or prediction uncertainty.

BiLSTM learns nonlinear temporal patterns from historical sequences through gated memory units and bidirectional processing [6]. Although it can represent complex time dependence, its performance is sensitive to data volume, sequence length, and hyperparameter selection [7]. The conventional structure also lacks explicit mechanisms for sparse zero demand and predictive uncertainty.

Rolling ARIMA fits each signal group separately and updates the historical sequence as new observations become available [3]. It provides an interpretable benchmark for locally stable autoregressive series, but its linear and phase-independent formulation limits nonlinear modelling and information sharing across signal groups.

To address these limitations, HUMAS-TSC combines multi-task phase prediction, sparse-demand modelling, uncertainty estimation, and temporal-change learning. A shared encoder extracts information from traffic demand, rolling statistics, temporal features, and previous signal operation, while task-specific heads predict sg2, sg4, and sg6. A total-demand head and consistency loss align the phase-level outputs with the predicted intersection demand.

For signal groups with frequent zero observations, HUMAS-TSC uses a hurdle-based prediction head. It separately estimates the probability of positive demand and the expected demand magnitude when vehicles are present [8, 9]:

$$\hat{q}_{i,t+1} = P(q_{i,t+1} > 0) \cdot E(q_{i,t+1} \mid q_{i,t+1} > 0) \quad (2)$$

This structure separates positive-demand occurrence from its conditional magnitude, reducing the influence of frequent zeros while retaining non-zero demand information. Phases with relatively continuous flow use a standard regression head.

HUMAS-TSC also estimates the uncertainty of each phase-demand prediction [10]. For each signal group, the model outputs both the expected demand and its predictive variance:

$$\left(\hat{q}_{i,t+1}, \sigma_{i,t+1}^2 \right) \quad (3)$$

Larger predictive variance represents lower confidence. A heteroscedastic loss with logarithmic regularization prevents unrealistic uncertainty estimates, which are subsequently used to adjust demand smoothing and green-time changes [10]. A temporal-change term further aligns predicted variations with observed traffic changes.

The complete training objective is written as

$$L = L_{\text{hurdle}} + \lambda_1 L_{\text{magnitude}} + \lambda_2 L_{\text{total}} + \lambda_3 L_{\text{cons}} + \lambda_4 L_{\text{temporal}} + \lambda_5 L_{\text{unc}} \quad (4)$$

Here, L_{hurdle} evaluates positive-demand occurrence, $L_{\text{magnitude}}$ measures non-zero demand error, and L_{total} evaluates total-flow prediction. L_{cons} maintains consistency between phase-level and total demand, L_{temporal} captures traffic changes, and L_{unc} trains the uncertainty output. The coefficients λ_1 to λ_5 are selected using the validation dataset.

2.3.2. Evaluation metrics

The four models are evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), Symmetric Mean Absolute Percentage Error (SMAPE), and Filtered Mean Absolute Percentage Error (Filtered MAPE). MAE measures average absolute deviation, whereas RMSE gives greater weight to large peak-period errors. SMAPE supports comparisons across signal groups with different flow levels. Filtered MAPE excludes zero-demand observations that would produce undefined or excessive percentage errors; the threshold is set to 1 vehicle per minute for all models and phases. Metrics are calculated for each signal group and averaged to obtain the phase-level results reported in Table 2.

2.4. Traffic signal optimization method

2.4.1. Problem formulation

The traffic signal timing problem is formulated as a constrained optimization task under a fixed cycle length [8]. At each control interval, the decision variables are the green times assigned to sg2, sg4, and sg6. The optimization objective considers both traffic demand and the stability of consecutive timing plans:

$$\min_{g_t} \sum_{i \in \{2,4,6\}} \bar{q}_{i,t} \left(1 - \frac{g_{i,t}}{C}\right) + \lambda_t \sum_{i \in \{2,4,6\}} (g_{i,t} - g_{i,t-1})^2 \quad (5)$$

Here, $\bar{q}_{i,t}$ denotes processed phase demand, C is the fixed cycle length, and $g_{i,t-1}$ is the previous green time. The demand term penalizes insufficient service for high-demand movements, whereas the stability term discourages abrupt timing changes. The coefficient λ_t balances responsiveness and stability.

2.4.2. Demand-based signal allocation

The allocation process starts with the predicted demands of signal groups sg2, sg4, and sg6. HUMAS-TSC provides both a phase-demand prediction $\hat{q}_{i,t}$ and an uncertainty estimate $\sigma_{i,t}$. Before the green times are calculated, the predicted demand is adjusted using uncertainty-adaptive smoothing:

$$\bar{q}_{i,t} = \alpha_{i,t} \hat{q}_{i,t} + (1 - \alpha_{i,t}) \bar{q}_{i,t-1} \quad (6)$$

Here, $\bar{q}_{i,t-1}$ is the demand estimate retained from the previous interval, and $\alpha_{i,t}$ controls the contribution of the current prediction. The smoothing coefficient is determined by:

$$\alpha_{i,t} = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \exp(-\gamma \sigma_{i,t}) \quad (7)$$

where $\alpha_{\min}$ and $\alpha_{\max}$ define the permitted smoothing range, and γ controls the sensitivity of the smoothing process to uncertainty. When the prediction uncertainty is low, $\alpha_{i,t}$ increases and the processed demand follows the current forecast more closely. When the uncertainty is high, greater weight is assigned to the previous demand estimate, reducing the influence of unreliable short-term fluctuations.

The green-time stability coefficient is also adjusted according to the combined uncertainty of the three signal groups:

$$\lambda_t = \lambda_0 + \rho \sum_{i \in \{2,4,6\}} \sigma_{i,t} \quad (8)$$

where λ_0 is the basic stability coefficient and ρ determines the influence of prediction uncertainty. A higher uncertainty level increases λ_t , making the timing plan more conservative. When the forecasts are more reliable, the controller can respond more actively to changes in traffic demand.

The smoothed demands and stability coefficient are inserted into Equation (5) to determine the green times of sg2, sg4, and sg6. Each solution becomes the reference for the next interval, enabling adaptive but stable control.

2.4.3. Constraint design

The signal allocation scheme applies operational constraints to ensure practical feasibility. Each signal group is assigned minimum and maximum green times. The lower bound prevents low-demand movements from losing service, while the upper bound avoids excessive allocation to one movement. These limits preserve basic service for sg2, sg4, and sg6 while allowing demand-responsive adjustments.

The total effective green time remains fixed within each cycle and is distributed among the three signal groups. Yellow, all-red, and other transition intervals are reserved for safety and excluded from this budget [8]. The optimization therefore redistributes only the available green time without changing the cycle structure. These constraints prevent infeasible or extreme timing plans and maintain compatibility with intersection operation.

3. Experimental results

3.1. Prediction performance

The dataset was divided chronologically into 72% for training, 8% for validation, and 20% for testing to prevent future information leakage. All models generated one-minute-ahead phase-demand predictions over the same test period. Hyperparameters were selected using the validation set. Rolling ARIMA adopted an order of (200), BiLSTM was trained for 50 epochs, and validation-based early stopping was applied to XGBoost and HUMAS-TSC. Prediction performance was evaluated using MAE, RMSE, Filtered MAPE, and SMAPE, as reported in Table 2.

Table 2. Four models phase flow prediction indicators

Model	MAE	RMSE	Filtered MAPE (%)	SMAPE (%)
XGBoost	0.2925	0.3885	8.8559	17.6399
Rolling ARIMA	0.4100	0.5485	12.2948	21.9538
HUMAS-TSC	0.4221	0.5713	12.6383	21.6912
BiLSTM	0.4471	0.5946	13.8064	22.8838

XGBoost achieved the lowest error across all four metrics. Relative to HUMAS-TSC, it reduced MAE and RMSE by approximately 30.7% and 32.0%, indicating that engineered lagged, rolling, temporal, and signal-control features were effective for tree-based short-term forecasting. Rolling ARIMA ranked second in MAE, RMSE, and Filtered MAPE, confirming clear short-term autoregressive dependence. HUMAS-TSC produced slightly higher absolute errors than Rolling ARIMA but achieved a lower SMAPE, while BiLSTM recorded the largest errors, possibly because the available data were insufficient for its more complex parameterization.

Although XGBoost provided the best point-prediction accuracy, lower average error does not necessarily yield better downstream signal control. Green-time allocation is particularly sensitive to phase imbalance, peak-period errors, and short-term fluctuations. HUMAS-TSC addresses these factors through phase-level modelling, uncertainty-aware smoothing, and stable timing adjustment. Figures 1 and 2 compare the observed demand with XGBoost and HUMAS-TSC predictions over the same test period and are consistent with the numerical results.

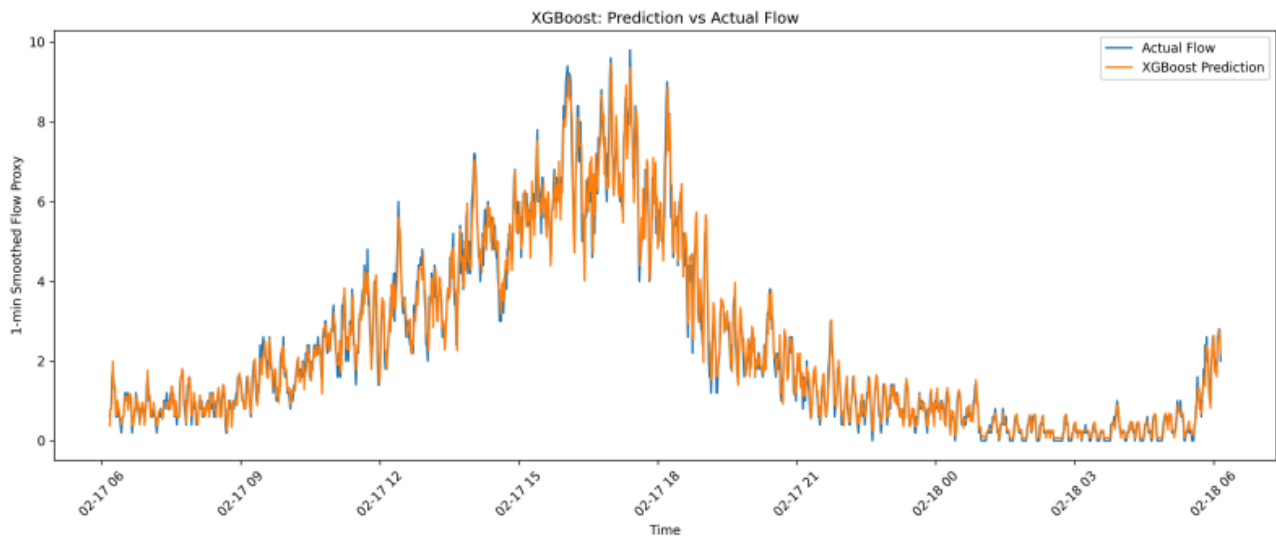


Figure 1. Observed and predicted traffic demand using XGBoost

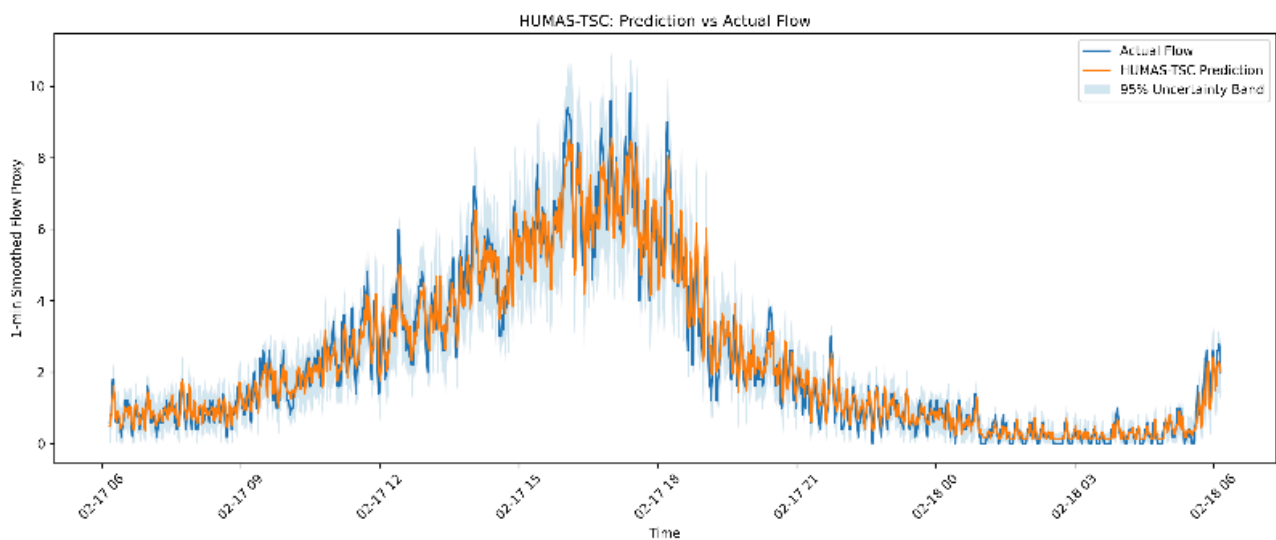


Figure 2. Observed and predicted traffic demand using HUMAS-TSC

3.2. Optimization results

Signal optimization was evaluated on the same test period used for traffic prediction. The total effective green time was fixed at 31 s for all control methods. Minimum and maximum green-time limits and a maximum inter-cycle change of 5 s were imposed. All timing parameters and safety thresholds were calibrated using only the training and validation data. The optimization used a demand-based delay proxy, while final delay was evaluated using the Webster formulation. High-saturation and saturation-cap rates measure threshold exceedance and upper-limit occurrence, respectively.

Table 3. Signal optimization performance

Control method	Average delay (s/vehicle)	Delay improvement (%)	High-saturation rate (%)	Saturation-cap rate (%)
Fixed-time	34.578	0.000	4.941	2.420
XGBoost-based	28.619	16.547	0.515	0.515
HUMAS-TSC	25.655	25.804	0.107	0.000

As shown in Table 3, average delay decreased from 34.578 s/vehicle under fixed-time control to 28.619 with XGBoost-based control and 25.655 with HUMAS-TSC. The high-saturation rate fell from 4.941% to 0.515% and 0.107%, while the saturation-cap rate decreased from 2.420% to 0.515% and finally to zero. These consistent trends show that prediction-based control reduces both delay and severe phase overload, with HUMAS-TSC providing the strongest improvement.

XGBoost offers accurate nonlinear forecasting from engineered features but does not explicitly represent phase-level uncertainty. HUMAS-TSC combines sparse-demand modelling, uncertainty-aware smoothing, and timing stability, making its outputs more suitable for constrained green-time allocation. Its maximum adjustment during stable operation was 0.25 s, and the initial transition of 4.417 s remained within the 5 s limit. The control gains were therefore achieved without excessive timing variation.

4. Discussion

Overall, the findings indicate that downstream signal-control performance depends on prediction reliability and timing stability in addition to point-prediction accuracy. However, the study is limited to a single urban intersection and relies mainly on aggregated traffic data, a delay proxy, and the Webster delay model, which cannot fully represent vehicle interactions and driver behaviour[8]. Future work will examine multi-intersection data, microscopic simulation, field testing, and online model updating to improve the robustness of HUMAS-TSC.

5. Conclusion

This study developed an iot-based framework linking phase-level traffic prediction with adaptive signal optimization. under the same chronological data split, xgboost achieved the highest point-prediction accuracy, whereas HUMAS-TSC produced the strongest control performance. through hurdle modelling, uncertainty-aware smoothing, and stable green-time allocation, HUMAS-TSC reduced average delay by 25.804% and substantially limited high-saturation conditions. the findings demonstrate that the most accurate forecast does not necessarily yield the best control decision because prediction reliability and timing stability also influence downstream performance. the study is limited by its single-intersection dataset and analytical delay evaluation. future work will focus on multi-intersection validation, microscopic simulation, field testing, online updating, and coordinated signal control.

References

- [1] Qadri, S. S. S. M., Gökçe, M. A., & Öner, E. (2020). State-of-art review of traffic signal control methods: Challenges and opportunities. *European Transport Research Review*, 12, 55. <https://doi.org/10.1186/s12544-020-00439-1>
- [2] Abduljabbar, R., Dia, H., & Liyanage, S. (2025). Machine learning traffic flow prediction models for smart and sustainable traffic management. *Infrastructures*, 10, 155. <https://doi.org/10.3390/infrastructures10070155>
- [3] Lippi, M., Bertini, M., & Frasconi, P. (2013). Short-term traffic flow forecasting: An experimental comparison of time-series analysis and supervised learning. *IEEE Transactions on Intelligent Transportation Systems*, 14, 871–882. <https://doi.org/10.1109/TITS.2013.2247040>
- [4] Genser, A., Makridis, M. A., Yang, K., Leclercq, L., & Menendez, M. (2023). A traffic signal and loop detector dataset of an urban intersection in Zurich, Switzerland. *Data in Brief*, 48, 109117. <https://doi.org/10.1016/j.dib.2023.109117>
- [5] Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). ACM. New York, NY, USA. <https://doi.org/10.1145/2939672.2939785>
- [6] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9, 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- [7] Graves, A., & Schmidhuber, J. (2005). Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18, 602–610. <https://doi.org/10.1016/j.neunet.2005.06.042>
- [8] Webster, F. V. (1958). *Traffic signal settings* (Road Research Technical Paper No. 39). Road Research Laboratory. pp. 1–44. London, UK.
- [9] Mullahy, J. (1986). Specification and testing of some modified count data models. *Journal of Econometrics*, 33, 341–365. [https://doi.org/10.1016/0304-4076\(86\)90002-3](https://doi.org/10.1016/0304-4076(86)90002-3)
- [10] Kendall, A., & Gal, Y. (2017). What uncertainties do we need in Bayesian deep learning for computer vision? In *Advances in Neural Information Processing Systems* (pp. 5574–5584).