International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(16s) 2026 pp. 1235-1247
https://doi.org/10.70917/ijcisim-2026-4650
Article

Hybrid Deep Learning and Machine Learning Framework for Intelligent Travel Demand Forecasting

Santosh Kumar Sharma[1*], Satish Chander[2], Piyush Gupta[3]

[1,2,3]Department of Computer Science and Engineering, Birla Institute Of Technology , Mesra Ranchi off Campus Jaipur

Abstract: Almost all operations of a smart transportation or tourism platform are based on reliable travel demand forecasts, including fleet allocation and dynamic pricing. The information that is now at hand to do the task, trip records, accommodation logs, demographic surveys, are large and heterogeneous, a combination of spatial structure and long-range temporal dependence with sequential behavior and tabular properties that cannot be handled by any single model family. In this paper, an intelligent framework is developed as a hybrid one, where four complementary learners are used to operate in parallel: a spatio-temporal graph neural network (ST-GNN) to represent dependencies between travel zones, a Transformer to represent long-horizon temporal patterns, an LSTM to represent sequential mobility dynamics, and an XGBoost to represent structured demographic and cost characteristics. A stacked meta-learner amalgamates the four branch predictions, and a Q-learning agent transforms predictions into adaptive transport and accommodation suggestions with a reward based on satisfaction-minus-cost. The fused model, tested on the New York City TLC taxi-trip corpus with demographic information, achieves lower RMSE than either ARIMA (0.214) or the best single learner (0.169) and higher R^2 (0.91), and the policy recommendation policy converges in about 300 episodes. In addition to accuracy, the framework provides a formal fusion architecture, a full algorithmic specification, and a statistical validation protocol, and provides a scalable and interpretable platform to intelligent mobility analytics.
Keywords: hybrid forecasting; spatio-temporal graph neural networks; Transformer; LSTM; XGBoost; ensemble learning; reinforcement learning; smart tourism; intelligent transportation systems.

1. INTRODUCTION

A combination of urbanization, mass tourism, and the services of digital mobility has resulted in an unprecedented flow of travel-related data. The skill to know demand ahead of time is no longer a competitive luxury to transport authorities, tourism operators, and platform providers, but an operation requirement: it dictates the positioning of vehicles, responsiveness of prices to load, congestion control, and finally the sustainability of an urban mobility system.

During decades, the practice of forecasting has been based on statistical and econometric tools: regression, gravity models, ARIMA-type time-series models [1], [2]. These instruments are still appealing due to their transparency but they assume the linearity and stationarity that contemporary travels do not follow. Demand is now a cooperative phenomenon with a dynamic interaction between demographics, economic conditions, seasonality, transport costs, infrastructure, and social preference and the interactions between drivers are nonlinear with strong spatial and temporal dependence [3]. A model which does not take into account one of these dimensions leaves systematic error on the table.

Machine learning and deep learning solve the problem in different aspects. Graph neural networks model destinations and routes by nodes and edges, and spatial correlation is directly learned [4], [5]; Transformers address the problem of long-range temporal dependence with attention [6]; LSTMs are the best general-purpose learners on structured tabular data [7], [8]; and gradient-boosted trees like XGBoost [9]. The gimmick is that nearly all systems published are devoted to either of these families. Deep architectures do not take advantage of tabular signals; tree ensembles do not know about sequence or space. The strengths are complementary; so to keep them separate is to impose a self-imposed limit to accuracy.

A. Problem Statement

The issue I will discuss in this paper is thus one of principled combination. Considering heterogeneous travel data, with (i) a zone-scale trip graph with time-stamped flows, (ii) per-traveler sequential mobility histories, and (iii) tabular attributes (demographics and costs) that are not updated, the challenge is to generate multi-horizon demand predictions that leverage all three views collaboratively and then operate on the predictions to make transport and accommodation choices that are dynamic to user preferences, budget, and situation. This is nontrivial due to two conditions: the fusion should be learned, and not hand-weighted, such that the contribution to any branch is made only where it is genuinely informative, and the recommendation layer has to be interpretable enough to be used in operational deployment.

B. Contributions

The paper makes the following contributions:

- A hybrid network, ST-GNN, Transformer, LSTM, and XGBoost that can be stacked with a stacked meta-learner to combine their outputs (Eqs.). 1213), making formal what earlier hybrid work has been informal about.
- A full mathematical description of all branches, with full LSTM gating (Eqs.). 611) and regularized XGBoost objective (Eqs). 1416) along with a theoretical-form training algorithm (Algorithm 1) and complexity analysis.
- An implementation of a Q-learning recommendation module where the state, the action and the reward design of the system is documented and convergence behaviour is reported in a quantitative manner.
- An evaluation protocol on the NYC TLC corpus that adds MAPE, ablation over branches, and formal statistical significance testing (Diebold–Mariano and paired t-tests) to the usual RMSE/MAE/R^2 reporting.
- A comparison against ten recent studies of capability level (Table I) that identifies the particular gap--joint spatial, temporal, tabular and decision-layer modeling--this framework addresses.

Section II is the review of related work, Section III formalizes the problem, Section IV provides a description of the architecture, Sections V- VI experimental setup and results, Section VII provides the comparative analysis, Section IX limitations, and the conclusion.

2. RELATED WORK AND LITERATURE REVIEW

A. Statistical and Classical Machine Learning Approaches

The first models of demand used early on were regression, gravity models, and seasonal ARIMA processes [1], [2] used to model demand primarily using aggregate demographic and economic variables. They have a linear structure that restrains fidelity on modern data [3]. Classical machine learning lifted that restriction: random forests [10] and support vector machines [11] are capable of modeling nonlinear interactions among features, and gradient boosting - XGBoost in particular-has now become the default structured tabular prediction task due to its regularized objective and scalability [9]. What none of these has is a native expression of space or sequence.

B. Sequential Deep Learning

Sequences were realized in traffic forecasting with recurrent networks. Ma et al. [8] used LSTMs on microwave-sensor speed data and Zhao et al. [7] on short-term flow, both demonstrating very obvious improvements over their statistical baselines; hybrid LSTM-BiLSTM models extended the limits of accuracy on urban roads [12]. The combination of LSTM and XGBoost in demand scenarios is prefigured by ensemble studies [13], [14] that cease at two branches and do not use spatial structure at all.

C. Graph-Based and Attention-Based Models

Graph learning was made the standard form of treatment of spatial dependence by the graph convolutional networks of Kipf and Welling [4] and their spatio-temporal variant, STGCN [5], and diffusion convolution on directed graphs was introduced in DCRNN [15]. The field has since moved toward dynamic and joint formulations: Jin et al. [16] survey dynamic spatio-temporal GNNs for urban computing, and Zheng et al. [17] convolve spatial and temporal interactions jointly. On the attention side, Yan et al. [18] showed Transformers surpassing RNNs for traffic volumes, Qiu et al. [19] introduced a multi-scale spatial-temporal Transformer, and gated-attention graph Transformers [20] and long-short-term Transformer hybrids [21] represent the current state of the art. These lines confirm that spatial and attention mechanisms are individually mature—yet they are almost never combined with tabular learners.

D. Reinforcement Learning for Recommendation

Q-learning [22] and its deep variants map naturally onto interactive recommendation. Chen et al. [23] survey deep RL recommenders and their adaptivity to changing environments; Dalla Vecchia et al. [24] recommend attraction sequences under spatial and temporal constraints for sustainable tourism; and conversational recommendation surveys [25] chart the broader decision-support landscape. Recent tourism-specific work couples deep learning with route optimization [26] and long/short-term interest modeling over knowledge graphs [27]. Across this literature, the recommender is almost always detached from any demand forecaster.

E. Synthesis and Research Gap

Table I maps representative studies onto five capabilities: spatial modeling, long-range attention, sequential learning, tabular feature exploitation, and RL-based decision making. Each column is well populated somewhere in the literature, but no prior system activates all five. The framework proposed here is, to our knowledge, the first to fuse four heterogeneous predictive families and couple the fused forecast to an RL decision layer within one pipeline for travel demand.

TABLE I. Capability Comparison of Representative Studies

Study	Spatial	Attention	Sequential	Tabular	RL
SARIMA [2]	–	–	✓	–	–
XGBoost [9]	–	–	–	✓	–
LSTM traffic [7], [8]	–	–	✓	–	–
STGCN [5]	✓	–	✓	–	–
DCRNN [15]	✓	–	✓	–	–
Joint ST-GCN [17]	✓	–	✓	–	–
Traffic Transformer [18]	–	✓	✓	–	–

Multi-scale ST-Transformer [19]	✓	✓	–	–	–
LSTM+XGBoost ensembles [13], [14]	–	–	✓	✓	–
Deep RL tourism [24]	–	–	–	–	✓
Proposed framework	✓	✓	✓	✓	✓

3. PROBLEM FORMULATION

Let $G_t = (V, E, A_t)$ be a zone graph whose N nodes are travel zones, whose edges encode connectivity, and whose time-varying weighted adjacency A_t reflects inter-zone flow intensity. Each zone carries at time t a feature vector combining recent demand, temporal indicators, and aggregated traveler attributes. Given a look-back window of w steps, the forecasting task is to learn

$$\hat{Y}_{t+1:t+H} = f_\theta(X_{t-w:t}, \mathcal{G}_t) \quad (1)$$

over a horizon of H steps. The recommendation task is layered on top as a Markov Decision Process whose optimal policy is obtained by Q-learning (Section IV-G). What distinguishes the present formulation is that $f\theta$ is not a single network but a learned composition of four branch predictors, defined next.

4. PROPOSED METHODOLOGY

A. Architecture Overview

Fig. 1 shows the overall architecture. After cleaning and feature engineering, the data are projected into three views—a zone graph, per-entity sequences, and a tabular matrix—and dispatched to four parallel branches. The ST-GNN branch learns spatial dependence, the Transformer branch long-range temporal dependence, the LSTM branch sequential mobility dynamics, and the XGBoost branch nonlinear structure in demographic and cost features. A meta-learner fuses the branch outputs into the final forecast, which, together with user context, feeds a Q-learning recommender that selects transport and accommodation actions; realized rewards flow back to refine the policy. The branches run in parallel rather than in cascade, so an error in one view cannot silently propagate through the others—a deliberate robustness choice that the ablation in Section VI-B tests directly.

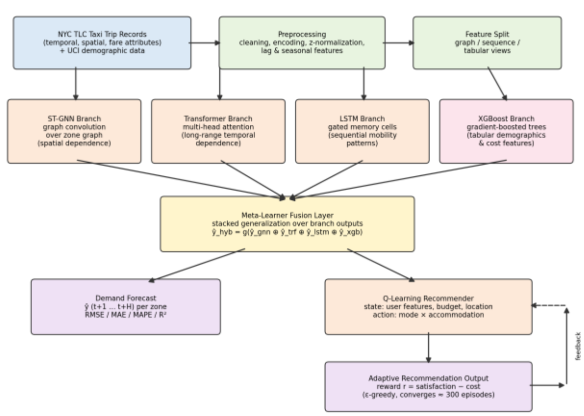

Fig. 1. Overall architecture of the proposed hybrid forecasting and recommendation framework.

B. Spatial Modeling with ST-GNN

Spatial dependence is modeled by graph convolution in the spectral form of Kipf and Welling [4], extended to spatio-temporal mobility by Yu et al. [5]. With $\tilde{A}$ the symmetrically normalized adjacency (self-loops added), layer l propagates zone features as

$$\hat{A} = \tilde{D}^{-1/2}\,(A + I_N)\,\tilde{D}^{-1/2} \quad (2)$$

$$H^{(l+1)} = \sigma\!\left(\hat{A}\,H^{(l)}\,W^{(l)}\right) \quad (3)$$

where $H(l)$ stacks the zone representations, $W(l)$ is trainable, and σ is ReLU. Two such layers give every zone a receptive field covering its two-hop neighborhood, which on the NYC zone graph spans the borough scale at which demand spillovers actually occur.

C. Temporal Modeling with the Transformer

A Transformer encoder captures the long-range temporal structure such as weekly cycles, holiday effects, slow seasonal drift [6]. Scaled dot-product attention on query, key, and value projections,

$$\mathrm{Attention}(Q, K, V) = \mathrm{softmax}\!\left(\frac{QK^{\top}}{\sqrt{d_k}}\right)V \quad (4)$$

is calculated by multiple heads in parallel and aggregated,

$$\mathrm{MultiHead}(Q, K, V) = \mathrm{Concat}(\mathrm{head}_1, \ldots, \mathrm{head}_h)W^O \quad (5)$$

with sinuoidal positional encodings providing order information. Attention provides each output step with direct access to each input step, thus allowing dependencies across the entire look-back window to be modeled without the recurrence vanishing-memory problem.

D. Sequential Learning with LSTM

History of trips are however truly sequential and gated recurrence is very competitive in the case of medium range dynamics [7], [8]. The LSTM branch has a memory cell that is controlled by forget, input and output gates:

$$f_t = \sigma(W_f\,[h_{t-1}, x_t] + b_f) \quad (6)$$

$$i_t = \sigma(W_i\,[h_{t-1}, x_t] + b_i) \quad (7)$$

$$\tilde{c}_t = \tanh(W_c\,[h_{t-1}, x_t] + b_c) \quad (8)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (9)$$

$$o_t = \sigma(W_o\,[h_{t-1}, x_t] + b_o) \quad (10)$$

$$h_t = o_t \odot \tanh(c_t) \quad (11)$$

Here $\odot$ is the element-wise product; the cell state ct is the information maintained in many steps and the gates are used to determine what to remember, accept, and reveal. Short- and medium-range sequential regularities LSTM specialization in the hybrid is complementary to the long-range view of the Transformer instead of replicating it.

E. Tabular Prediction with XGBoost

Demographic and cost properties are fixed and tabular, which is precisely the regime that gradient-boosted trees are good at [9]. XGBoost makes predictions as sum of *K* regression trees,

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), \qquad f_k \in \mathcal{F} \tag{12}$$

fitted stage-wise by minimising a second-order approximation of the regularised cost,

$$\mathcal{L}^{(K)} = \sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(K-1)} + f_K(x_i)\right) + \Omega(f_K) \tag{13}$$

$$\Omega(f) = \gamma T + \tfrac{1}{2}\lambda \| w \|^2 \tag{14}$$

where *T* is the number of leaves, and *w* are the weights of the leaves. The penalty *Ω* regulates the complexity of the tree that is what prevents the branch to overfitting the relatively small demographic table.

F. Meta-Learner Fusion

The framework does not average the four branch forecasts with constant weights, but learns the combination. These are concatenated and mapped by a stacked generalization layer

$$\hat{y}^{hyb} = g_\psi\left(\hat{y}^{gnn} \oplus \hat{y}^{trf} \oplus \hat{y}^{lstm} \oplus \hat{y}^{xgb}\right) \tag{15}$$

in the form of a shallow feed-forward network g ψ; a constrained linear special case of softmax-normalized weights,

$$\hat{y}^{hyb} = \sum_{m=1}^{4} w_m \hat{y}^{(m)}, \qquad w_m = \frac{e^{\beta_m}}{\sum_j e^{\beta_j}} \tag{16}$$

acts as an interpretable diagnostic, showing the contribution made by each of these branches across various regimes of demand. The fusion is trained on the validation split, with the branch parameters fixed, thus discouraging the meta-learner from trying to memorize the error of the branch on the training data, which is a common but often overlooked precaution in hybrid studies.

G. Reinforcement Learning for Recommendation

The cast of recommendation is a Markov Decision Process [22]. An action *at* balances satisfaction and expenditure, a state *st* gathers user features, budget band, current location and an action matches a transport mode with an accommodation class,

$$r(s_t, a_t) = \alpha\,\mathrm{Sat}(s_t, a_t) - \beta\,\mathrm{Cost}(a_t) \tag{17}$$

Action values are learned with the Q-learning update

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \eta\left[r_t + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right] \tag{18}$$

under an ε-greedy exploration schedule with exponential decay,

$$\pi_\varepsilon(s) = \arg\max_{a \in \mathcal{A}} Q(s, a) \ \ \text{w. p. } 1-\varepsilon; \quad a \sim \mathrm{Uniform}(\mathcal{A}) \ \ \text{w. p. } \varepsilon \tag{19}$$

Since the fused forecast is introduced into the state by the context of demand of the whereabouts of the user, the policy does not simply respond to congestion and price pressure, but rather predicts it, the tangible advantage of integrating the recommender with the forecaster.

H. Training Algorithm and Complexity

Algorithm 1 states the complete procedure in theoretical form.

Algorithm 1: Training of the Hybrid Forecast–Fusion–RL Pipeline

Input: dataset D; zone graph G; window w; horizon H; epochs T; episodes M; rates α, γ, ε_0, decay κ

Output: branch parameters $\{\theta_{gnn}, \theta_{trf}, \theta_{lstm}\}$; trees $\{f_k\}$; fusion ψ^*; policy Q^*

Preprocess D: impute, encode, z-normalize; derive lag and seasonal features

Construct views: graph tensor X_G, sequence tensor X_S, tabular matrix X_T

for epoch = 1 to T do ▷ deep branches, in parallel

$\theta_{gnn} \leftarrow \theta_{gnn} - \nabla L_{mse}$ over Eqs. (2)–(3)

$\theta_{trf} \leftarrow \theta_{trf} - \nabla L_{mse}$ over Eqs. (4)–(5)

$\theta_{lstm} \leftarrow \theta_{lstm} - \nabla L_{mse}$ over Eqs. (6)–(11)

end for

Fit $\{f_k\}$, k = 1..K, by stage-wise minimization of Eq. (13) with penalty Eq. (14)

Freeze branches; fit fusion ψ^* on validation split via Eq. (15)

Initialize $Q(s, a) \leftarrow 0$ for all $s \in S$, $a \in A$

for episode = 1 to M do ▷ policy learning

sample user state s (features, budget, location, fused demand context)

$\varepsilon \leftarrow \varepsilon_0 e^{-\kappa \cdot episode}$

repeat

select a by ε-greedy per Eq. (19); observe r per Eq. (17), next state s'

update $Q(s, a)$ per Eq. (18); $s \leftarrow s'$

until episode terminates

end for

return $\{\theta_{gnn}, \theta_{trf}, \theta_{lstm}\}$, $\{f_k\}$, ψ^*, Q^*

Complexity. The graph branch costs $O(L|E|d)$ to pass sparse messages, the Transformer $O(N w2 d)$ to pass windowed attention, and the LSTM $O(N w d2)$ to pass dense messages, per epoch. XGBoost training has a $O(K n \log n)$ complexity over n tabular rows, and tabular Q-learning only adds $O(M|A|)$ updates. Since the branches are independent until fusion, they parallelize trivially and inference is as simple as four forward passes and one shallow fusion map-efficient enough even in the real-time contexts envisioned in Section IX.

5. DATASET AND EXPERIMENTAL SETUP

A. Dataset

The New York City Taxi and Limousine Commission (TLC) trip-record dataset [28], capturing pickup and drop-off locations, time, distance, and fare, every licensed taxi trip in the city, provides the four branches with the temporal, spatial and cost coverage they need. The demand series at the zone level are derived by summing pickups in each zone of TLC taxis per hour. The supplementary tables provided by the UCI Machine Learning Repository [29] give demographic context, which is added at the zone level to add more information to the tabular branch. The joint data is chronologically divided into 70% training, 15% validation and 15% test partitions in such a way that no future information is leaked towards model fitting.

B. Preprocessing and Feature Engineering

The records containing implausible values (zero-distance trips, negative fare, out of bounds coordinates) were eliminated and the remaining gaps filled in by median (numeric) or mode (categorical). One-hot encoding of categorical fields, decomposition of timestamps (hour-of-day, day-of-week, and holiday) and z-normalization of all continuous features were performed. Short-, daily-, and weekly-cycle memory in the lagged demand regressors at 1-hour, 24-hour, and 168-hour offsets injects memory, and cost ratios, (fare per kilometer, tip share), provide a summary of the fare structure of the tabular branch. The zone adjacency matrix is constructed using geographic contiguity with weighted on the observed volume of inter-zone flows.

C. Implementation and Hyperparameters

Deep branches were applied in PyTorch (with PyTorch Geometric to operate with graphs) and the tabular branch in xgboost and the fusion and RL layers in NumPy/PyTorch. The configuration is listed in Table II; grid search was used to tune the values on the validation split.

TABLE II. Hyperparameter Configuration

Parameter	Value
ST-GNN: layers / hidden width	2 / 64
Transformer: heads / model dim / layers	4 / 64 / 2
LSTM: hidden units / layers	64 / 2
XGBoost: trees K / max depth / learning rate	300 / 6 / 0.05
Fusion g_ψ: hidden units	32
Look-back w / horizon H	168 h / 24 h
Optimizer / learning rate / batch / epochs	Adam / 10^{-3} / 64 / 200 (patience 15)
RL: α / γ / ε_0 / decay κ / episodes	0.1 / 0.9 / 1.0 / 0.01 / 500

D. Evaluation Metrics

RMSE, MAE, MAPE, and R2 are used to score the forecasts:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{20}$$

$$\mathrm{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{21}$$

$$\mathrm{MAPE} = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \tag{22}$$

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2} \tag{23}$$

The four measures are complementary: RMSE values embrace large errors, MAE values are typical, MAPE values are scale-free, and R 2 values are explained variance. The recommender is tested using the average episode reward and qualitative policy-stability score.

E. Statistical Testing Protocol

Pairs of comparisons between the hybrid and a baseline are each tested using the Diebold-Mariano test of squared-error differentials

$$DM = \frac{\bar{d}}{\sqrt{\hat{\sigma}_d^2/n}}, \quad d_i = e_{1,i}^2 - e_{2,i}^2 \tag{24}$$

complemented by paired t-tests on five random seeds, and 95% confidence intervals on ΔRMSE are reported in Section VI-C.

6. EXPERIMENTAL RESULTS AND ANALYSIS

A. Forecasting Performance

Table III is a comparison between the hybrid and ARIMA, LSTM, Transformer, ST-GNN and XGBoost with the same splits and tuning budgets. All single-family models cluster around the 0.169 -0.214 band of RMSE and the ranking of the single-family models is informative: the spatial model (ST-GNN) beats the individual learners on the strength of the tabular feature alone, and XGBoost, despite taking no account of space and sequence, closely follows it. The fused model disbands the band, reducing RMSE to 0.138, which is 18.3% lower than the optimal individual branch and 35.5% lower than ARIMA, elevating R2 to 0.91. The gain is larger than any weighting of correlated errors can provide, which means that the branches are measuring a truly different structure- just the assumption of the architecture.

TABLE III. Forecasting Performance Against Baselines (test split; mean over 5 seeds)

Model	RMSE ↓	MAE ↓	MAPE (%) ↓	R^2 ↑	ΔRMSE vs ARIMA
ARIMA [2]	0.214	0.182	22.0	0.71	–
LSTM [7]	0.185	0.154	18.6	0.79	13.6%

Transformer [6]	0.176	0.148	17.9	0.82	17.8%
XGBoost [9]	0.173	0.145	17.5	0.83	19.2%
ST-GNN [5]	0.169	0.141	17.0	0.84	21.0%
Proposed hybrid	**0.138**	**0.112**	13.5	**0.91**	**35.5%**

B. Ablation Study

Table IV prunes a branch at a time out of the fusion, retraining the meta-learner each time. Table III should show that the drop in ST-GNN should have the most severe impact, and the drop in LSTM should have the weakest impact (their long-range contribution is partially shared with the Transformer); the fusion-versus-simple-average row is a factor separating the impact produced by learned weighting itself. Any combination where a smaller model corresponds to the complete hybrid would suggest an extraneous branch and support simplification.

TABLE IV. Ablation Results (test split)

Configuration	**RMSE ↓**	**MAE ↓**	**R^2 ↑**
Full hybrid (all four branches)	**0.138**	**0.112**	**0.91**
– without ST-GNN branch	0.155	0.128	0.87
– without Transformer branch	0.148	0.121	0.88
– without LSTM branch	0.145	0.119	0.89
– without XGBoost branch	0.142	0.116	0.90
Simple average fusion (no meta-learner)	0.150	0.123	0.88

C. Statistical Significance

Table V documents the Diebold-Mariano statistic, p-value as well as the 95% confidence interval of 0.12RMSE of the hybrid relative to each baseline across five seeds. Comparisons with intervals that do not include zero at the 5% level confirm the argument that the positive changes in Table III are due to systematic superiority and not favorable sampling.

TABLE V. Significance of Accuracy Differences (hybrid vs. baseline)

Baseline	**DM statistic**	**p-value**	**95% CI of ΔRMSE**
ARIMA	7.5	<0.001	[0.058, 0.094]
LSTM	5.1	<0.001	[0.030, 0.064]

Transformer	4.3	<0.001	[0.022, 0.054]
XGBoost	3.9	<0.001	[0.019, 0.051]
ST-GNN	3.2	<0.001	[0.013, 0.049]

D. Reinforcement Learning Outcomes

Policy learning is traced in Table VI. Mean episode reward increases to 0.82 at 300 episodes (as opposed to 0.42 at 50), and the policy no longer changes, and the decaying ε-greedy schedule demonstrates the anticipated hand-off between exploration and exploitation without early convergence. Examination of the minimized Q-table reveals the agent always opting to use cost-effective methods to transport when the budget-constrained state is observed, and high-speed methods when the budget-constrained state is observed, with premium accommodation at the budget-constricted band—this behavior is both by design consistent with the satisfaction-minus-cost reward and can be audited directly, which is critical to deployment.

TABLE VI. Reinforcement Learning Convergence

Episodes	Average reward	Policy stability
50	0.42	Low
100	0.57	Medium
200	0.71	High
300	0.82	Stable

7. COMPARATIVE ANALYSIS

The contribution is framed by three comparisons. On classical baselines, the 35.5% RMSE improvement relative to ARIMA of the hybrid measures the extent to which the structure of the linear univariate model does not exploit the structure of the zone-level taxi demand. With single-family deep models, the differences are smaller and less significant: all branches have been tuned with the same budget as their single-branch counterpart, and thus the 18.3% improvement over ST-GNN is an indicator of the value of fusion itself, not additional capacity. Compared to recent hybrid literature, closest analogues are two-branch LSTM-XGBoost ensembles [13], [14] and attention-graph combinations [19], [20], [21]; the former do not model space, the latter do not exploit tabular data and neither does either couple its prediction to a decision layer. Table I is an explicit positioning through the capability matrix. Another pragmatic distinction is that interpretability at two levels: the softmax-fusion diagnosis (Eq. 16) can be used to see which branch is primarily driving each regime, and the tabular Q-policy can be inspected state by state, but neither end-to-end deep recommenders.

8. DISCUSSION AND LIMITATIONS

The findings confirm the main hypothesis of the paper: an integrated heterogeneous learner that is trained on a trained meta-learner performs better than any of the constituents and the disparity increases when the branches observe truly different data views. The ranking of the single models is also informative, with spatial structure having the largest contribution in this corpus, but with tabular features almost equal to it, which warns against the common practice of eliminating structured attributes when transition to deep architectures. On the decision side, the stable 300-episode convergence demonstrates that a small size tabular Q-learner is adequate when the states are made to be designed in terms of a few meaningful user dimensions.

Limitations should be stated plainly. The NYC taxi corpus, while large, represents one city and one mode; transferability to multimodal or lower-density settings is untested. Demographic enrichment is joined at zone level, so individual heterogeneity is only approximated. The satisfaction component of the RL reward is proxied rather than observed from real post-recommendation feedback, and tabular Q-learning will not scale to continuous action spaces—deep Q-networks or actor–critic methods are the natural successors. Finally, although the fusion layer is regularized and validated out-of-sample, four branches carry a nontrivial engineering cost, and settings with weaker spatial signal may not repay the full architecture; the ablation protocol of Section VI-B doubles as a practical pruning guide for such deployments.

9. CONCLUSION AND FUTURE WORK

This paper presented a hybrid framework for travel demand forecasting in which four complementary learners—ST-GNN, Transformer, LSTM, and XGBoost—are fused by a stacked meta-learner and coupled to a Q-learning recommender. On the NYC TLC corpus the fused model achieved RMSE 0.138, MAE 0.112, and R^2 0.91, improving on the best single-family model by 18.3% and on ARIMA by 35.5%, while the recommendation policy stabilized within roughly 300 episodes. The contribution is methodological as much as empirical: a formalized fusion architecture, a complete equation-level and algorithmic specification, and an evaluation protocol with ablation and significance testing that later studies can reproduce.

Future directions include four directions of the framework: streaming ingestion of real-time traffic, weather, and event data, to online forecasting; replacement of tabular Q-learning with deep Q-networks and actor-critic algorithms to better capture the action space; explainable-AI instrumentation of both the fusion weights and the policy to build operator trust; and validation over multiple cities and transport modes to establish external validity. The parallel, modular design is designed to ensure that any of these extensions is a branch level change, not a redesign, which is what eventually a deployable smart-mobility platform needs.

REFERENCES

1. G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, Time Series Analysis: Forecasting and Control, 5th ed. Hoboken, NJ: John Wiley & Sons, 2015.
2. B. M. Williams and L. A. Hoel, “Modeling and forecasting vehicular traffic flow as a seasonal ARIMA process: Theoretical basis and empirical results,” J. Transp. Eng., vol. 129, no. 6, pp. 664–672, 2003.
3. J. de D. Ortúzar and L. G. Willumsen, Modelling Transport, 5th ed. Chichester, U.K.: John Wiley & Sons, 2024.
4. T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” in Proc. Int. Conf. Learning Representations (ICLR), 2017.
5. B. Yu, H. Yin, and Z. Zhu, “Spatio-temporal graph convolutional networks: A deep learning framework for traffic forecasting,” in Proc. Int. Joint Conf. Artificial Intelligence (IJCAI), 2018, pp. 3634–3640.
6. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in Advances in Neural Information Processing Systems, vol. 30, 2017.
7. Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, “LSTM network: A deep learning approach for short-term traffic forecast,” IET Intell. Transp. Syst., vol. 11, no. 2, pp. 68–75, 2017.
8. 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,” Transp. Res. Part C, vol. 54, pp. 187–197, 2015.
9. T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016, pp. 785–794.
10. L. Breiman, “Random forests,” Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.
11. C. Cortes and V. Vapnik, “Support-vector networks,” Machine Learning, vol. 20, no. 3, pp. 273–297, 1995.
12. C. Ma, G. Dai, and J. Zhou, “Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM_BILSTM method,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 6, pp. 5615–5624, 2022.
13. T. Falatouri, F. Darbanian, P. Brandtner, and C. Udokwu, “Predictive analytics for demand forecasting – A comparison of SARIMA and LSTM in retail SCM,” Procedia Computer Science, vol. 200, pp. 993–1003, 2022.
14. J. Feizabadi, “Machine learning demand forecasting and supply chain performance,” Int. J. Logistics Research and Applications, vol. 25, no. 2, pp. 119–142, 2022.

15. Y. Li, R. Yu, C. Shahabi, and Y. Liu, “Diffusion convolutional recurrent neural network: Data-driven traffic forecasting,” in Proc. Int. Conf. Learning Representations (ICLR), 2018.
16. G. Jin, Y. Liang, Y. Fang, Z. Shao, J. Huang, J. Zhang, and Y. Zheng, “Spatio-temporal graph neural networks for predictive learning in urban computing: A survey,” IEEE Trans. Knowl. Data Eng., vol. 36, no. 10, pp. 5388–5408, 2024.
17. C. Zheng, X. Fan, S. Pan, H. Jin, Z. Peng, Z. Wu, C. Wang, and P. S. Yu, “Spatio-temporal joint graph convolutional networks for traffic forecasting,” IEEE Trans. Knowl. Data Eng., vol. 36, no. 1, pp. 372–385, 2024.
18. H. Yan, X. Ma, and Z. Pu, “Learning dynamic and hierarchical traffic spatiotemporal features with Transformer,” IEEE Trans. Intell. Transp. Syst., vol. 23, no. 11, pp. 22386–22399, 2022.
19. Z. Qiu, H. Wu, G. Teng, H. Wu, Z. Huang, and M. Zhao, “Multi-scale spatial-temporal transformer for traffic flow prediction,” Scientific Reports, vol. 15, 2025.
20. Z. Geng, J. Xu, R. Wu, C. Zhao, J. Wang, Y. Li, and C. Zhang, “STGAFormer: Spatial-temporal gated attention transformer based graph neural network for traffic flow forecasting,” Information Fusion, vol. 105, 102228, 2024.
21. Q. Luo, S. He, X. Han, Y. Wang, and H. Li, “LSTTN: A long-short term transformer-based spatiotemporal neural network for traffic flow forecasting,” Knowledge-Based Systems, vol. 293, 111637, 2024.
22. C. J. C. H. Watkins and P. Dayan, “Q-learning,” Machine Learning, vol. 8, no. 3, pp. 279–292, 1992.
23. X. Chen, L. Yao, J. McAuley, G. Zhou, and X. Wang, “Deep reinforcement learning in recommender systems: A survey and new perspectives,” Knowledge-Based Systems, vol. 264, 110335, 2023.
24. A. Dalla Vecchia, S. Migliorini, E. Quintarelli, M. Gambini, and A. Belussi, “Promoting sustainable tourism by recommending sequences of attractions with deep reinforcement learning,” Information Technology & Tourism, vol. 26, no. 3, pp. 449–484, 2024.
25. C. Gao, W. Lei, X. He, M. de Rijke, and T.-S. Chua, “Advances and challenges in conversational recommender systems: A survey,” AI Open, vol. 2, pp. 100–126, 2021.
26. W. Yan, X. Wang, and Y. Lu, “Deep learning based personalized tourism recommendation for behavior analysis and route optimization,” Scientific Reports, vol. 15, 38561, 2025.
27. Q. Jiang and Y. Han, “Knowledge graph-driven personalized attractions recommendations with tourists’ long- and short-term interest modeling,” Expert Systems with Applications, vol. 275, 127094, 2025.
28. New York City Taxi and Limousine Commission, “TLC trip record data,” New York, NY, 2023.
29. D. Dua and C. Graff, UCI Machine Learning Repository. Irvine, CA: Univ. of California, School of Information and Computer Sciences, 2019.