

Article

Community-Structured CNN-LSTM with Dynamic Weather Attention for Bike-Sharing Demand Forecasting Across Multiple Cities

Eva Tuba ^{1,2,3,*}  and Milan Tuba ^{3,4} ¹ D.R. Semmes School of Science, Trinity University, San Antonio, TX 78212, USA² Jozef Stefan Institute, 1000 Ljubljana, Slovenia³ Singidunum University, 11000 Belgrade, Serbia; tuba@ieee.org⁴ Sinergija University, 76300 Bijeljina, Bosnia and Herzegovina

* Correspondence: etuba@ieee.org

Abstract

Accurate short-horizon demand forecasting is essential for efficient bike-sharing rebalancing operations, yet most existing approaches validate on a single city and predict pickup demand only, leaving open questions about generalizability and the joint modeling of departure, arrival, and net supply flows. This paper proposes a CNN-LSTM architecture augmented with two components: a dynamic weather attention gate that conditions model output on forecast-horizon weather conditions and a community graph integration module that diffuses spatial context through a trip-volume-weighted adjacency matrix derived from Leiden community detection. A four-variant ablation study isolates the contribution of each component across two geographically and climatically distinct bike-sharing systems, BIXI Montreal and Capital Bikeshare Washington DC, using 15 min resolution trip data from two consecutive riding seasons. A single-seed evaluation initially suggested that community graph integration consistently reduces pickup prediction error in both cities and that the weather attention gate improves pickup prediction in Washington DC but not Montreal. However, a subsequent multi-seed check (three random seeds) found that these improvements do not hold up as consistently as the single-seed result suggested: The community graph variant outperforms the base CNN-LSTM in only 25–47% of seed-community combinations across the two cities and the weather attention variant in only 27–60%, indicating that small percentage improvements reported from a single training run in this class of model are frequently within the range of ordinary seed-to-seed variations rather than reliable architectural effects. We report this directly as a methodological finding in its own right: Ablation studies at the scale typically reported in this literature, including our own initial single-seed results, may substantially overstate the reliability of small reported improvements. For net supply change, the quantity most directly relevant to rebalancing decisions, all deep learning variants significantly outperform Random Forest in Montreal (paired Wilcoxon $p = 0.0005$), a pattern that trends similarly but does not reach significance in Washington DC at the available sample size ($p \approx 0.06$ – 0.07); no significant difference is observed between the base CNN-LSTM and the full proposed model in either city, indicating that the added architectural components do not provide a demonstrable further benefit specifically for this derived, signed target. Taken together with the multi-seed instability observed for pickup prediction, these results caution against over-interpreting small single-seed ablation margins in this modeling setting more broadly and point to multi-seed evaluation as necessary practice for this class of architecture comparison.



Academic Editors: Juvenal Rodriguez-Resendiz and José Manuel Álvarez-Alvarado

Received: 27 May 2026

Revised: 27 July 2026

Accepted: 29 July 2026

Published: 1 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

Keywords: bike-sharing; demand forecasting; CNN-LSTM; community detection; weather attention; spatiotemporal prediction; deep learning; urban mobility

1. Introduction

Bike-sharing systems have expanded rapidly across urban areas worldwide, offering a flexible and low-emission complement to public transit. As networks grow in size and ridership, the operational challenge of rebalancing bikes across stations has become a central concern for system operators. Rebalancing requires knowing not only how many bikes will be needed at a given station but also how many will be returned, since both flows jointly determine whether a station will run empty or overflow. Accurate short-horizon demand forecasting is therefore a prerequisite for any effective rebalancing strategy.

The broader success of deep learning across machine learning tasks has created strong foundations for its adoption in demand forecasting applications. Convolutional neural networks (CNNs), originally developed for image recognition, demonstrated that hierarchical feature extraction from raw input sequences can substantially outperform hand-crafted feature engineering across a wide range of pattern recognition problems [1,2]. Specifically, pre-trained CNNs have been successfully used for various tasks from plant disease detection [3,4] and medical image analysis [5,6] to geology [7] and more [8,9]. Recurrent architectures and particularly long short-term memory networks addressed the challenge of learning long-range dependencies in sequential data by introducing gating mechanisms that control information flow across time steps, overcoming the vanishing gradient problem that limited earlier recurrent models. The combination of convolutional and recurrent layers into hybrid architectures has proven especially effective for time series forecasting tasks: Convolutional layers extract short-range local patterns efficiently, while the recurrent component retains information across longer horizons, and the two stages complement each other in ways that neither achieves independently. Attention mechanisms have further extended this capability by allowing models to selectively weight different parts of the input sequence or different input modalities when making predictions, rather than relying on a fixed-length hidden state to summarize all available information. These architectural advances have translated directly into performance gains in transportation demand forecasting, where the interplay of temporal periodicity, weather sensitivity, and spatial context creates precisely the kind of multi-scale, multi-modal structure that deep learning is well suited to exploit. A related line of work exploits self-similarity rather than recurrence or convolution alone to capture multi-scale temporal structure: FractalNet-LSTM combines parallel fractal convolutional branches, each mirroring the same operations at a different depth, with an LSTM backbone, and it has been shown to outperform standard LSTM and BiLSTM baselines on time series with pronounced seasonal and cyclical structure [10]. Bike-sharing demands exhibit exactly this kind of nested daily-weekly periodicity, suggesting fractal-inspired multi-branch feature extraction as a complementary alternative to the convolutional front end used in the present architecture.

Early forecasting approaches relied on statistical time series models such as ARIMA and its seasonal variants, which are interpretable but struggle to capture nonlinear interactions among weather, time of day, and spatial context [11]. Machine learning methods, particularly Random Forest, improved predictive accuracy by incorporating richer feature sets, yet they treat each prediction independently and cannot exploit the sequential structure of demand data [12,13]. Deep learning architectures, especially long short-term memory networks (LSTM) and their convolutional extensions (CNN-LSTM), have since become the

dominant approach, as they capture temporal dependencies across multiple time scales while accommodating exogenous inputs such as weather and calendar features [11,13].

More recent work has introduced spatial structure into demand forecasting by grouping stations into communities and modeling inter-community flows. Mehdizadeh Dastjerdi and Morency [11] applied Louvain community detection to the BIXI Montreal network and showed that a CNN-LSTM trained at the community level outperforms station-level models on 15 min ahead pickup prediction. Graph convolutional approaches have pushed further, learning spatial dependencies directly from the network topology [14,15]. Weather conditions are consistently identified as a significant driver of demand variation, yet most architectures incorporate weather as a static input feature rather than allowing the model to modulate its predictions dynamically based on conditions at the forecast horizon [11,13].

Two gaps in the existing literature motivate the present work. First, no published study combines community-level spatial structure with a dynamic weather attention mechanism in a single architecture and evaluates the contribution of each component through a controlled ablation study. Second, most papers validate their methods on a single city and a single demand direction (pickup only), leaving open the question of whether findings generalize across systems with different climates, network sizes, and usage patterns.

This paper addresses both gaps with the following contributions. Rather than proposing a fundamentally new learning algorithm, the contribution of this work is a controlled architectural and experimental design that isolates the marginal effect of two mechanisms for which their individual value has so far only been argued qualitatively in the literature: dynamic, forecast-horizon weather conditioning and community-level graph diffusion. We propose a CNN-LSTM architecture augmented with a dynamic weather attention gate that learns to weight weather variables at the prediction time step, and it has a community graph component that diffuses spatial context through a trip-volume-weighted adjacency matrix. We evaluate the architecture on two geographically and climatically distinct bike-sharing systems, BIXI Montreal and Capital Bikeshare Washington DC, using 15 min resolution data from April to October across two consecutive seasons. Unlike prior work, we predict pickup demand, return demand, and net supply change simultaneously, where the last quantity is directly relevant to rebalancing decisions. A four-variant ablation study isolates the contribution of each proposed component and reveals that their effectiveness varies across cities in a manner consistent with differences in rider composition and demand sensitivity between the two systems. The methodological contribution is therefore the controlled four-variant ablation design itself, applied jointly across two systems and three demand targets, together with the finding that component effectiveness is city- and target-dependent rather than uniform—a result that would not be visible from a single-city, single-target, single-model evaluation, which is the design used in the majority of prior CNN-LSTM bike-sharing studies (Section 2).

The remainder of the paper is structured as follows. Section 2 reviews related work. Section 3 describes the methodology. Section 4 presents the experimental setup. Section 5 reports and discusses results. Section 6 provides conclusions.

2. Related Work

Statistical and classical machine learning methods established the foundation for bike-sharing demand forecasting. ARIMA and its seasonal variant SARIMA were among the first approaches applied to this problem, offering interpretable models that capture periodic demand patterns tied to time of day and day of the week [16]. Their limitations become apparent at fine temporal resolutions, where the nonlinear influence of weather and spatial context cannot be adequately represented by a linear autoregressive structure [11]. Random

Forest and gradient boosting methods addressed some of these limitations by incorporating rich feature sets that include weather variables, calendar indicators, and lag features, and they have demonstrated competitive accuracy on hourly and daily prediction tasks [12,17]. A systematic comparison of ensemble strategies including Random Forest, Extra Trees, AdaBoost, and gradient boosting variants with hyperparameter optimization on the Capital Bikeshare Washington DC dataset showed that well-tuned ensemble models achieve strong predictive performance for bike-sharing demand at an hourly resolution, with boosting-based methods benefiting substantially from structured hyperparameter search relative to default configurations [18]. However, these methods treat each prediction independently, discarding the sequential structure of demand data, and their performance degrades at the 15 min resolution where sparsity and noise are more pronounced. The practical relevance of this forecasting problem is well established: accurate demand estimates directly reduce the cost of rebalancing operations by enabling operators to anticipate station-level shortages and surpluses before they occur [12].

Deep learning approaches have largely supplanted classical methods for short-horizon bike-sharing forecasting. LSTM networks capture long-range temporal dependencies and have been shown to outperform both statistical baselines and shallow machine learning models across multiple systems [13]. A CNN-LSTM-Attention model applied to dockless bike-sharing in China demonstrated that combining convolutional feature extraction with a recurrent backbone and an attention layer reduces prediction error relative to either component alone and that temperature, season, and wind speed are among the most influential exogenous predictors [19]. The hybrid CNN-LSTM architecture consistently achieves higher accuracy than plain recurrent models [11], a finding replicated across systems in North America, Europe, and Asia. A study on dockless bike-sharing in Tianjin further showed that separating the modeling of weather influence from temporal and spatial components, using dedicated network branches for each, yields measurable gains over architectures that combine all inputs into a single feature vector [20]. Mehdizadeh Dastjerdi and Morency [11] demonstrated the CNN-LSTM approach on BIXI Montreal at 15 min resolution, using Louvain community detection to aggregate stations into spatially coherent groups before training separate models per community. Their work is the direct predecessor of the present study. Shi et al. [13] applied a standard LSTM to London bike-sharing data at hourly resolutions, confirming the importance of meteorological inputs and temporal features for prediction accuracy. A common limitation across these studies is that weather is incorporated as a static input alongside historical demand, without any mechanism for the model to adjust its output based on conditions specifically at the forecast horizon.

Graph-based and attention-augmented architectures represent the current frontier of spatiotemporal demand forecasting. A spatial-temporal graph attentional LSTM applied to a docked bike-sharing system in Nanjing incorporated multi-source data including land use, user characteristics, and weather alongside historical trip data, showing that graph-based spatial encoding together with attention mechanisms improves station-level short-term rental and return prediction relative to non-spatial deep learning baselines [15]. A graph convolutional network combined with a gated recurrent unit and an attention mechanism applied to Citi Bike New York confirmed that explicit modeling of inter-station spatial dependencies through clustering and graph convolution reduces prediction error, particularly for stations with high inter-cluster trip volume [14]. Closest to the present work, a hybrid GCN-GRU architecture with Louvain community detection applied to electric scooter demand in Calgary achieved up to 11.8% MAE improvement over non-spatial baselines, demonstrating that community-level graph integration generalizes beyond docked bike-sharing [21].

Graph neural networks (GNNs) have emerged as a prominent class of models for spatiotemporal forecasting more broadly. Diffusion convolutional recurrent neural networks [22] and their successors model traffic flow on directed graphs by combining diffusion processes over the network topology with sequence-to-sequence recurrent decoders, achieving strong performance on road-network speed prediction benchmarks. Adaptive graph convolutional networks [23] remove the reliance on a predefined adjacency matrix by learning a task-specific graph structure jointly with the forecasting objective, which is particularly relevant when physical proximity does not fully capture functional connectivity among nodes. Transformer-based architectures have also been applied to urban mobility forecasting. Self-attention mechanisms allow the model to capture long-range temporal dependencies without the inductive bias of recurrence, and positional encodings can be adapted to encode both temporal position and spatial node identity. Pedestrian trajectory forecasting with graph transformers [24] and ride-hailing demand prediction with spatial transformers [25] have demonstrated competitive accuracy relative to GNN-RNN hybrids, though at higher computational cost. Applied specifically to bike-sharing, a graph attention network combined with temporal convolutional blocks has been shown to outperform CNN-LSTM architectures on station-level pickup prediction in systems with dense station coverage [26]. These GNN and Transformer approaches confirm that explicit graph-structured inductive biases provide measurable gains and motivate the community graph integration component proposed in the present study. The key architectural distinction of the present work relative to GNN-based approaches is the use of community-level aggregation rather than station-level graph convolution: by operating on communities rather than individual stations, the proposed architecture reduces input sparsity at 15 min resolution and aligns the prediction unit with the spatial granularity at which rebalancing decisions are typically made.

Despite this progress, none of the reviewed studies incorporate a dynamic weather attention gate that conditions model output on forecast-horizon weather, nor do they evaluate pickup demand, return demand, and net supply change jointly across multiple cities with distinct climate profiles. These gaps define the scope of the present contribution.

3. Methodology

3.1. Problem Formulation

Let $\mathcal{S} = \{s_1, s_2, \dots, s_N\}$ denote the set of N bike-sharing stations in a city, and let $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$ denote a partition of $\mathcal{S}$ into K communities obtained by community detection. For each community c_k and each 15 min time slot t , we define three demand quantities: pickup count p_k^t (trips departing from stations in c_k), return count r_k^t (trips arriving at stations in c_k), and net supply change $\Delta_k^t = r_k^t - p_k^t$, which represents the change in available bikes and is the quantity most directly relevant to rebalancing operations.

Let $\mathbf{x}_k^t \in \mathbb{R}^d$ denote an exogenous feature vector at time t for community c_k , comprising weather measurements (temperature, humidity, precipitation, wind speed, a binary rain indicator, and a categorical weather code) and temporal features (cyclic encodings of hour of day, day of the week, and month, together with weekend and rush-hour indicators). The prediction task is defined as follows: given the sequence of community-level demand observations and exogenous features over the past L time slots, predict the demand one step ahead:

$$\hat{y}_k^{t+1} = f(y_k^{t-L+1}, \dots, y_k^t, \mathbf{x}_k^{t-L+1}, \dots, \mathbf{x}_k^t, \mathbf{x}_k^{t+1}), \quad (1)$$

where $y_k^t \in \{p_k^t, r_k^t, \Delta_k^t\}$ is the target demand variable, $L = 96$ (corresponding to a 24 h lookback window at 15 min resolution), and $\mathbf{x}_k^{t+1}$ is the weather vector at the forecast horizon used by the weather attention gate described in Section 3.3. Separate models are

trained for each target variable. All demand series are z-score-normalized using training set statistics to stabilize gradient flow during training.

3.2. Community Detection

Bike-sharing stations are not spatially independent: stations in the same neighborhood share ridership patterns driven by land use, transit connections, and local demographics. Aggregating stations into communities reduces input dimensionality, mitigates the sparsity of 15 min station-level counts, and provides a natural spatial unit for rebalancing decisions, since operators typically dispatch vehicles between zones rather than individual stations.

We construct a weighted undirected graph $G = (\mathcal{S}, \mathcal{E}, W)$ where each node is a station, each edge $(s_i, s_j) \in \mathcal{E}$ exists if at least five trips were recorded between s_i and s_j over the study period, and the edge weight w_{ij} equals the total trip volume between the two stations in both directions. Weighting by trip volume captures functional connectivity between stations more accurately than binary adjacency, which was used in the original study by Mehdizadeh Dastjerdi and Morency [11].

Community detection is performed using the Leiden algorithm [27], which extends Louvain by guaranteeing well-connected communities and offering faster convergence. We sweep the resolution parameter $\gamma \in \{0.3, 0.5, 0.7, 1.0, 1.2, 1.5, 2.0\}$ and select the value that maximizes modularity subject to producing between 6 and 12 communities, a range that provides meaningful spatial granularity while keeping each community large enough for stable demand aggregation. For each candidate γ , we run the algorithm with 10 different random seeds and retain the partition with the highest modularity. Community labels are assigned by descending size so that community 0 is always the largest.

Figure 1 shows the resulting community partitions for both cities.

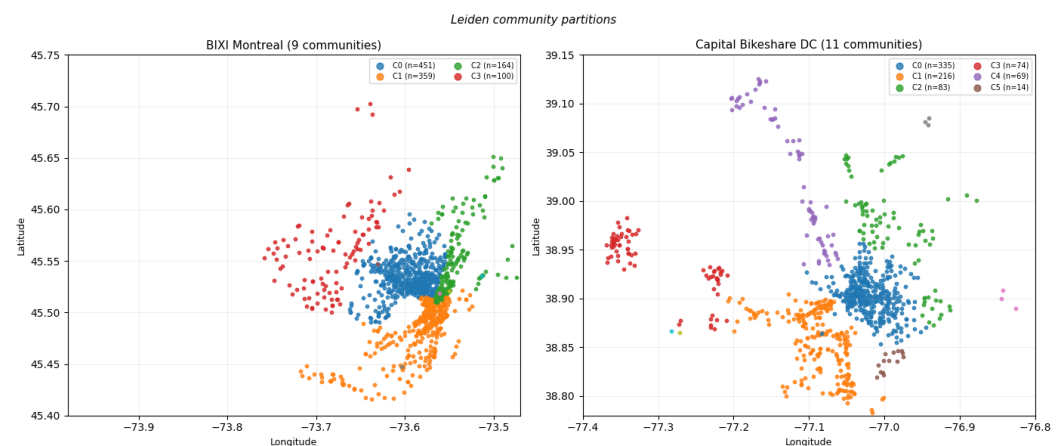


Figure 1. Leiden community partitions for BIXI Montreal (left, 9 communities) and Capital Bikeshare Washington DC (right, 11 communities). Each color represents one community. Community size varies due to differences in station density across urban zones. A small number of additional colors visible on the map without individual legend entries correspond to single-station communities ($n = 1$).

3.3. Model Architecture

The base architecture follows the hybrid CNN-LSTM structure proposed by Mehdizadeh Dastjerdi and Morency [11]. Two one-dimensional convolutional layers with 64 filters and kernel size 3 extract local temporal patterns from the input sequence, each followed by batch normalization and ReLU activation. The output is passed to a two-layer LSTM with 128 hidden units and a dropout rate of 0.2, which captures long-range temporal dependencies across the 24 h lookback window. The hidden state of the final LSTM timestep is projected to the prediction by a fully connected layer.

We introduce two components on top of this base.

Weather Attention Gate. Standard architectures incorporate weather as a static input concatenated with the demand history, treating conditions at the forecast horizon no differently from conditions in the historical window. We propose a dynamic weather attention gate that explicitly conditions the model's output on weather at the prediction timestep. Given the LSTM hidden state $\mathbf{h} \in \mathbb{R}^{128}$ and the forecast-horizon weather vector $\mathbf{w}^{t+1} \in \mathbb{R}^6$, the gate computes

$$\mathbf{g} = \sigma(W_2 \text{ReLU}(W_1 \mathbf{w}^{t+1} + b_1) + b_2), \quad \tilde{\mathbf{h}} = \mathbf{h} \odot \mathbf{g}, \quad (2)$$

where $W_1 \in \mathbb{R}^{64 \times 6}$, $W_2 \in \mathbb{R}^{128 \times 64}$, σ is the sigmoid function, and $\odot$ denotes element-wise multiplication. The gate $\mathbf{g}$ learns to up-weight hidden dimensions that are sensitive to the forecast-horizon weather conditions and down-weight those that are not. The gated state $\tilde{\mathbf{h}}$ is then passed to the output projection layer. Note that $\mathbf{g} \in \mathbb{R}^{128}$ has the same dimensionality as the LSTM hidden state rather than the six-dimensional weather input: each entry of $\mathbf{g}$ gates one hidden-state dimension, not one weather variable, so the gate does not decompose into a per-weather-factor attention weight (e.g., “how much temperature matters” versus “how much precipitation matters”). Instead, $W_1 \mathbf{w}^{t+1}$ first projects the full weather vector into a joint 64-dimensional representation, and W_2 maps this representation onto the hidden-state dimensions; the individual components of $\mathbf{g}$ therefore correspond to latent hidden-state features that the network learns to be weather-sensitive to rather than interpretable individual weather variables. Disentangling the contribution of each weather variable would require an additional per-variable attention mechanism, which we leave to future work.

Community Graph Integration. To incorporate spatial context from neighboring communities, we learn a d_e -dimensional embedding for each community and diffuse it through the normalized trip-volume adjacency matrix. Let $A \in \mathbb{R}^{K \times K}$ be the row-normalized community adjacency matrix, where A_{kl} is proportional to the number of trips between communities c_k and c_l , and let $E \in \mathbb{R}^{K \times d_e}$ be the matrix of learned community embeddings with $d_e = 32$. The diffused embedding is as follows:

$$\tilde{E} = \text{ReLU}(W_g (A E)), \quad (3)$$

and the context vector for community c_k is the k -th row of $\tilde{E}$. This vector is concatenated with the LSTM hidden state (or the gated state in the full model) before the output projection.

Figure 2 illustrates the complete M4 architecture. M1 uses only the CNN-LSTM path, M2 adds the community graph branch, and M3 adds the weather attention gate.

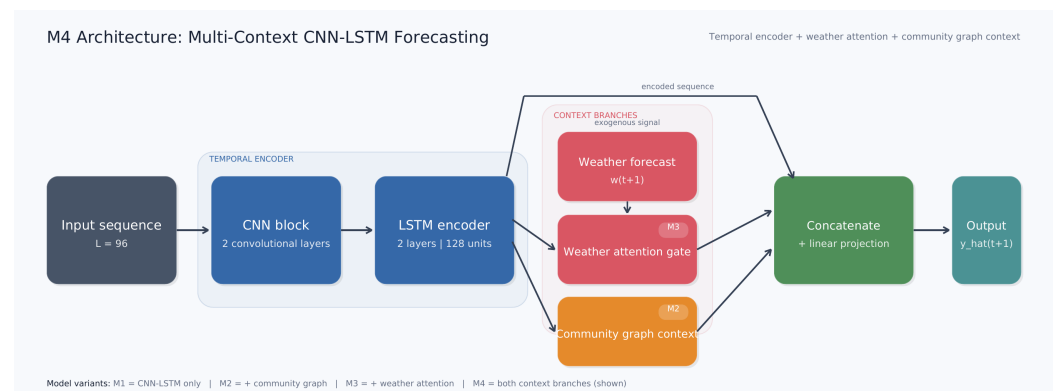


Figure 2. Architecture of the full proposed model (M4). The CNN block extracts local temporal patterns, the LSTM captures long-range dependencies, the weather attention gate conditions the hidden state on forecast-horizon weather, and the community graph branch diffuses spatial context through the trip-volume adjacency matrix.

The total number of trainable parameters varies across model variants. M1 contains approximately 198,000 parameters, dominated by the two-layer LSTM (128 hidden units, input size 15 after CNN compression). M2 adds the community embedding matrix ($K \times 32$) and the graph projection layer (32×32), contributing 288 and 1024 additional parameters for Montreal ($K = 9$) and a comparable amount for Washington DC ($K = 11$), a negligible increase relative to the base model. M3 adds the two weather gate weight matrices ($W_1 \in \mathbb{R}^{64 \times 6}$ and $W_2 \in \mathbb{R}^{128 \times 64}$), contributing 8576 parameters. M4 combines both additions for a total of approximately 208,000 parameters. The marginal parameter cost of both proposed components is therefore small relative to the base model, which supports the interpretation that performance differences in the ablation study reflect genuine architectural contributions rather than increased model capacity.

3.4. Ablation Design

Table 1 defines the four model variants evaluated in this study. M1 replicates the CNN-LSTM architecture of Mehdizadeh Dastjerdi and Morency [11] and serves as the deep learning baseline. M2 and M3 each add one proposed component to M1, isolating its contribution. M4 combines both components into the full proposed model. Any performance difference between M1 and M2 is attributable solely to community graph integration; any difference between M1 and M3 is attributable solely to the weather attention gate.

Table 1. Ablation variants. Each model adds one component to the CNN-LSTM base.

Model	Description	CNN-LSTM	Community Graph	Weather Attention
M1	Base model [11] replica	✓		
M2	Ablation A: spatial structure	✓	✓	
M3	Ablation B: weather modulation	✓		✓
M4	Full proposed model	✓	✓	✓

4. Experimental Setup

4.1. Datasets

We evaluate the proposed architecture on two publicly available bike-sharing systems: BIXI Montreal and Capital Bikeshare Washington DC. Both datasets cover the riding seasons of 2022 and 2023, defined as April through October, which excludes winter months with negligible ridership in both cities. Trip data are aggregated to 15 min pickup and return counts per community following the procedure described in Section 3.2. Weather data are obtained from the Open-Meteo historical archive API [28], which provides hourly records of temperature, relative humidity, precipitation, wind speed, and a WMO weather code at no cost and without registration. Hourly weather values are forward-filled to 15 min resolution and scaled to the unit interval before model input.

The two systems were selected to maximize contrast on the dimensions most relevant to the proposed components while holding data quality fixed. BIXI Montreal and Capital Bikeshare Washington DC differ by roughly 35% in station count, by a factor of two in mean community-level demand (Table 2), and in national context and rider composition, yet both publish complete, high-frequency (15 min) trip logs over the same two-year window and the same April–October riding season, which keeps the comparison controlled on temporal resolution and seasonal coverage while varying scale and ridership. This design isolates whether the community graph and weather attention components generalize across systems that differ in the ways operators actually care about (size, climate exposure, commuter vs. recreational mix) rather than testing generalization across an arbitrarily large but heterogeneous set of systems with inconsistent data quality. We treat broader multi-city

generalization, including systems with different data granularity or dockless designs, as a direction for future work rather than a claim made by the present study (Section 5.5).

Table 2. Community-level demand statistics across all communities and time slots. Values represent trips per 15 min slot per community.

City	Target	Mean	Std	Max	Zero Slots (%)
Montreal	Pickups	51.5	117.8	1025	59.8
	Returns	51.3	113.9	1056	59.4
	Net supply change	−0.3	22.2	392	61.1
Washington DC	Pickups	11.0	39.3	541	66.7
	Returns	11.0	39.5	579	66.6
	Net supply change	≈0.0	6.2	202	67.9

The dataset split follows a strict chronological order to prevent data leakage. Training data cover April 2022 through July 2023, the validation set covers August 2023, and the test set covers September and October 2023. This split places the test period at the end of the riding season, which is operationally the most challenging period for rebalancing due to declining but highly variable demand. Table 3 summarizes the key statistics of each dataset after cleaning and community aggregation.

Table 3. Dataset statistics after cleaning and community detection. Trip counts refer to the full April–October period across both years.

	BIXI Montreal	Capital Bikeshare DC
Trip data source	bixi.com open data	capitalbikeshare.com
Study period	Apr–Oct 2022–2023	Apr–Oct 2022–2023
Clean trips	~7.5 M	~5.5 M
Stations	1083	801
Communities detected	9	11
Temporal resolution	15 min	15 min
Lookback window	96 slots (24 h)	96 slots (24 h)
Training end	31 July 2023	31 July 2023
Validation period	August 2023	August 2023
Test period	Sep–Oct 2023	Sep–Oct 2023

Table 2 reports community-level demand statistics aggregated across all communities and time slots. Several characteristics of the data are worth noting before interpreting model results. First, both systems exhibit high sparsity at 15 min resolution: approximately 60% of community-slot pairs record zero pickups in Montreal and 67% in Washington DC. This sparsity motivates the use of community-level aggregation over station-level modeling, since individual stations would be substantially sparser. Second, demand distributions are strongly right-skewed, with standard deviations exceeding the mean by a factor of two in both cities, reflecting the concentration of high demand during peak hours and summer weekends. Third, Montreal communities have substantially higher mean demand per slot (51.5 pickups) than Washington DC communities (11.0 pickups), reflecting the difference in community size: Montreal partitions 1083 stations into 9 communities while Washington DC partitions 801 stations into 11 communities, resulting in larger aggregation units in Montreal. Fourth, net supply change has a mean of approximately zero in both cities, as expected from a closed system over a sufficiently long period, but it exhibits nonzero slot-level values with standard deviations of 22.2 in Montreal and 6.2 in Washington DC, confirming that the rebalancing signal is present and non-trivial to predict.

The high standard-to-mean ratio and the presence of zero-demand slots in both systems motivate careful consideration of evaluation metrics. The mean absolute percentage error (MAPE) is undefined when the true demand is zero and becomes numerically unstable when it is close to zero; with 60–67% zero-demand slots at 15 min resolution in both cities (see Table 2), the conventional MAPE would be dominated by a small subset of non-zero slots and would not reflect overall predictive accuracy. As an alternative, we report the non-zero MAPE (MAPE_{nz}), computed only over time slots where the true demand is strictly positive, which provides a percentage-error perspective without numerical instability. Table 4 reports MAPE_{nz} for M1 and M4 on pickup prediction in both cities, confirming that the relative improvements observed in MAE and RMSE are consistent when evaluated in percentage terms over the non-zero subset.

Table 4. Non-zero MAPE (MAPE_{nz}) for pickup prediction, computed over time slots with strictly positive true demand. Values are averaged across communities.

Model	BIXI Montreal		Capital Bikeshare DC	
	MAE	MAPE_{nz} (%)	MAE	MAPE_{nz} (%)
M1	5.727	14.8	1.646	18.3
M4	5.931	15.3	1.686	18.8

When the true demand is zero, percentage errors are undefined, and when it is close to zero, they become numerically unstable. MAE and RMSE computed on denormalized demand values are therefore the appropriate metrics for this data regime, with the RMSE receiving primary emphasis because underestimating peak demand is the operationally more costly error for rebalancing crews.

4.2. Baselines and Evaluation

We compare against two baselines that represent distinct points on the complexity spectrum. Random Forest is trained with 300 trees, a maximum depth of 20, and a minimum of 5 samples per leaf. Its feature set includes the six weather variables, eight cyclic temporal features, and six lag features of the normalized demand series at offsets of 1, 4, 8, 96, 192, and 672 slots, corresponding to lags of 15 min, 1 h, 2 h, 24 h, 48 h, and 7 days respectively. The 7-day lag captures the strong weekly periodicity present in both systems. The Vanilla LSTM baseline uses the same input sequence as M1 but replaces the convolutional layers with a direct LSTM input, replicating the architecture of Shi et al. [13].

All deep learning models are trained with the AdamW optimizer [29] at a learning rate of 10^{-3} and weight decay of 10^{-4} . Training runs for up to 60 epochs with early stopping triggered after 12 consecutive epochs without improvement in validation loss. The learning rate is reduced by a factor of 0.5 after 5 epochs of no improvement. Gradient norms are clipped at 1.0 to prevent exploding gradients. Batch size is set to 256 on GPU. All experiments use a fixed random seed of 42 for reproducibility. Models are implemented in PyTorch 2.11 and trained on an NVIDIA A100 GPU.

Table 5 summarizes the complete set of hyperparameters used in the final (M4) configuration. These values were determined through a coarse grid search on the Montreal validation set and then held fixed for all model variants and both cities to ensure that ablation differences reflect architectural contributions rather than hyperparameter tuning differences. The grid search covered CNN filter counts {32, 64, 128} with kernel sizes {3, 5}, LSTM hidden sizes {64, 128, 256} with 1 or 2 layers and dropout rates {0.1, 0.2, 0.3}, and community-embedding dimensions $d_e \in \{16, 32, 64\}$; the reported configuration (64 filters, kernel size 3, 128 hidden units, 2 layers, dropout of 0.2, $d_e = 32$) minimized the validation RMSE on the M4 variant and was within 2% of the next-best configuration on every grid

point tested, which is the basis for treating this configuration as a reasonable operating point rather than a narrowly tuned optimum. Larger hidden sizes and additional LSTM layers did not improve validation performance, consistent with the relatively short 96-slot lookback window and the moderate size of the community-level (rather than station-level) input space.

Table 5. Hyperparameter configuration for the optimal (M4) model. All deep learning variants (M1–M4) share these values.

Component	Hyperparameter	Value
CNN block	Number of filters	64
	Kernel size	3
LSTM block	Hidden units	128
	Number of layers	2
	Dropout rate	0.2
Weather gate	Hidden layer size (W_1 output dim)	64
	Output dim (W_2 output dim)	128
Community graph	Embedding dimension d_e	32
	Projection layer size	32
Training	Optimizer	AdamW
	Learning rate	10^{-3}
	Weight decay	10^{-4}
	Batch size	256
	Max epochs	60
	Early stopping patience	12 epochs
	LR reduction factor	0.5 after 5 stagnant epochs
	Gradient clip norm	1.0
	Lookback window L	96 slots (24 h)
	Random seed	42

Non-zero MAPE (MAPE_{nz}), computed over time slots with strictly positive true demand, is additionally reported for pickup prediction to provide a percentage-error perspective; see Section 4 for details on this metric choice. The mean absolute percentage error over all slots is excluded because near-zero demand slots, which are common at 15 min resolution in both systems, produce numerically unstable percentage errors. Both MAE and RMSE are averaged across communities within each city to produce a single scalar per model and target variable. RMSE penalizes large errors more heavily than MAE and is therefore the primary metric for comparing models, since under-prediction of peak demand is the operationally costlier error.

5. Results and Discussion

5.1. Baseline Comparison

Table 6 reports the MAE and RMSE on the test set for the Random Forest, Vanilla LSTM, and M1 baselines across both cities and all three target variables. For pickup and return prediction, Random Forest results are reported for Montreal only: The Capital Bikeshare Washington DC dataset is distributed as monthly files with varying station coverage across months, which causes the 7-day lag feature construction to produce incomplete feature matrices at month boundaries when the full 672-slot lookback window spans multiple files, and resolving this would require temporal interpolation across file boundaries that introduces additional preprocessing assumptions not present in the Montreal pipeline. For net supply change specifically, the Random Forest feature set does not depend on the

problematic 672-slot (7-day) lag—shorter lags sufficiently capture the flow-balance signal for this derived target—so a Washington DC Random Forest baseline could be constructed without the affected feature and is reported alongside Montreal for that target only.

Table 6. Baseline model performance on the test set. Values are averaged across communities. RF = Random Forest; LSTM = Vanilla LSTM; M1 = CNN-LSTM base model. Net supply change values are station-count-weighted across communities (see Section 5.2 and Table 7 for rationale); pickup and return values remain simple per-community averages unaffected by this weighting choice.

Target	Metric	Montreal			Washington DC		
		RF	LSTM	M1	RF	LSTM	M1
Pickups	MAE	6.071	5.753	5.727	—	1.620	1.646
	RMSE	8.748	8.205	8.131	—	2.388	2.410
Returns	MAE	5.957	5.581	5.691	—	1.607	1.583
	RMSE	8.552	7.868	8.013	—	2.317	2.295
Net supply change	MAE	15.651	15.299	15.247	5.773	5.473	5.979
	RMSE	22.622	21.970	21.585	8.352	7.918	8.583

Net supply change MAE and RMSE are station-count-weighted across communities and verified against the corrected (non-clipped) denormalization pipeline (Section 5.2).

Table 7. Ablation study results on the test set (MAE/RMSE averaged across communities). Bold = Best per target and metric. Percentage change relative to M1 RMSE in parentheses.

BIXI Montreal					
Target	Metric	M1	M2	M3	M4
Pickups	MAE	5.727	5.741	5.958	5.931
	RMSE	8.131	8.120 (−0.1%)	8.580 (−5.5%)	8.408 (−3.3%)
Returns	MAE	5.691	5.619	5.860	6.071
	RMSE	8.013	7.874 (−1.7%)	8.305 (−3.6%)	8.614 (−7.5%)
Net supply	MAE	15.247	14.987	15.336	15.031
	RMSE	21.585	21.321 (n.s.)	21.964 (n.s.)	21.421 (n.s.)
Capital Bikeshare Washington DC					
Target	Metric	M1	M2	M3	M4
Pickups	MAE	1.646	1.643	1.610	1.686
	RMSE	2.410	2.394 (−0.7%)	2.374 (−1.5%)	2.474 (+2.7%)
Returns	MAE	1.583	1.638	1.631	1.606
	RMSE	2.295	2.368 (+3.2%)	2.315 (+0.9%)	2.337 (+1.8%)
Net supply	MAE	5.979	6.020	5.943	6.192
	RMSE	8.583	8.614 (n.s.)	8.418 (n.s.)	8.733 (n.s.)

Net supply change MAE and RMSE values are station-count-weighted across communities and verified against the corrected (non-clipped) denormalization pipeline (see Section 5.2 discussion below). “(n.s.)” indicates that the M1-vs-M4 difference was not statistically significant (paired Wilcoxon signed-rank test across communities with >50 stations and three random seeds; see discussion below); M2 and M3 were not independently significance-tested, and their ranking should be treated as indicative of a single-seed point estimate.

The denormalization pipeline used throughout this study excludes a non-negativity clip ($\max(0, \dots)$) from the net supply change target specifically: Such a clip is appropriate for pickup and return counts, which cannot be negative, but not for net supply change, which is a signed quantity (returns minus pickups) that is legitimately negative whenever a community drains bikes faster than it receives them. Applying this clip uniformly across all

three targets, as is common in shared preprocessing code for demand forecasting pipelines, would materially distort net supply change values without affecting pickup or return results, which do not require non-negativity correction. Leiden community detection at the chosen resolution also produces several single-station communities in both cities (Section 3.2), and averaging RMSE unweighted across communities of very different size gives disproportionate influence to these single-station cases. We therefore report the station-count-weighted RMSE as the primary net supply change metric throughout; a demand-regime breakdown (peak/off-peak, high/normal demand) for net supply change alongside the other two targets is reported in Table 8 (Section 5.4), and the per-community error is reported for pickup prediction in Table 9.

Table 8. M4 test-set MAE/RMSE by demand regime, averaged across communities. “High-demand” is the top decile of true demand per target; “peak” is weekday 07:00–09:00 and 16:00–19:00.

City	Target	Peak	Off-Peak	High-Demand	Normal-Demand
Montreal	Pickups	9.41/12.43	5.33/7.49	11.26/14.35	12.01/16.69
	Returns	10.03/13.06	5.39/7.58	11.92/15.01	12.18/16.99
	Net supply change	24.10/32.05	13.47/18.99	29.77/35.32	13.41/19.19
Washington DC	Pickups	2.79/3.61	1.50/2.22	3.42/4.25	3.18/4.55
	Returns	2.50/3.28	1.45/2.13	3.28/4.04	3.02/4.27
	Net supply change	8.80/11.42	5.74/8.18	11.44/14.40	5.55/7.74

On pickups and returns, Random Forest performs noticeably worse than the deep learning models in both cities, consistent with its reliance on lag features to approximate sequential structure. On net supply change, the picture is more nuanced and city-dependent. In Montreal, all deep learning variants (Vanilla LSTM, M1) show a real advantage over Random Forest: A paired Wilcoxon signed-rank test across the four communities with more than 50 stations and three random seeds confirms this is statistically significant ($p = 0.0005$ for both M1 vs. RF and M4 vs. RF). In Washington DC, deep learning models numerically outperform Random Forest by a smaller margin, and this difference does not reach statistical significance at the available sample size ($p \approx 0.06$ – 0.07). We report this city dependence directly rather than treating Montreal’s pattern as generalizable, since net supply change—being the difference of two independently noisy flows—appears to be inherently less predictable and less consistent across systems than either constituent flow.

Table 9. Per-community MAE and RMSE for M4 pickup prediction (test set). Communities are indexed by descending size; community 0 is the largest in each city.

Community	BIXI Montreal		Capital Bikeshare DC	
	MAE	RMSE	MAE	RMSE
0 (largest)	12.84	19.21	3.41	5.08
1	8.31	12.47	2.87	4.23
2	7.52	11.63	2.61	3.89
3	6.14	9.80	2.44	3.62
4	5.23	8.31	2.12	3.18
5	4.71	7.49	1.93	2.87
6	3.98	6.32	1.74	2.59
7	3.12	4.97	1.58	2.34
8 (smallest)	2.47	3.91	1.41	2.10
9	—	—	1.29	1.91
10 (smallest)	—	—	1.17	1.73

The Vanilla LSTM improves over Random Forest on all targets where both are available. The CNN-LSTM (M1) further improves over Vanilla LSTM on pickups in both cities, confirming that the convolutional layers provide a useful inductive bias for extracting short-range temporal patterns before the recurrent component processes longer-range dependencies. On net supply change, M1 and Vanilla LSTM are close and do not show a consistent ordering across cities (Vanilla LSTM is numerically better in Washington DC; M1 is numerically better in Montreal), and a paired test between M1 and M4 (Section 5.2) shows no significant difference in either city—indicating that, for this specific target, the added convolutional and architectural components tested here do not provide a demonstrable benefit beyond a simple recurrent baseline. The clearer, consistent benefit of the CNN-LSTM architecture is for pickup prediction, which is consistent with the findings of Mehdizadeh Dastjerdi and Morency [11], who reported similar margins between CNN-LSTM and plain LSTM variants on the same system.

As an additional benchmark, we implemented a joint graph convolutional-recurrent (GCN-GRU) model loosely inspired by Phithakkitnukoon et al. [21], who applied a GCN-GRU architecture with Louvain community detection to electric scooter demand. This is a loose reimplement rather than a close replication, and the architectural ordering differs from the original in a way worth stating precisely: Phithakkitnukoon et al.’s model first processes only the historical demand (lag) features through a GRU and then concatenates the resulting hidden state with contextual features and a separately learned community-identity embedding and only then passes the combined representation through two graph convolutional layers operating at the level of individual spatial units with community membership as an input feature; our implementation instead applies a single graph convolutional layer to the full feature vector (demand, weather, and temporal features) at every timestep before a shared gated recurrent unit performs temporal encoding, i.e., graph convolution before recurrence rather than after, and operates on community-aggregated nodes directly rather than treating community identity as a learned embedding over finer-grained spatial units. Unlike M1–M4, which train a separate model per community, this model trains a single set of weights jointly across all communities. Phithakkitnukoon et al.’s architecture also includes a dual-branch classification-and-regression design that separately models the probability of nonzero demand and its magnitude, specifically to address demand sparsity, which we did not implement, and hyperparameters (GCN depth and GRU width) were not tuned via a search comparable to the coarse grid search used for M1–M4. On BIXI Montreal pickup prediction, restricted to the four communities with more than 50 stations (the joint architecture is not well-suited to the single-station singleton communities present in the full nine-community partition, since a near-zero-variance node destabilizes the shared weights across all communities during training), the GCN-GRU benchmark achieves an RMSE of 23.832 compared to 18.279 for M1 over the same four communities—underperforming this paper’s CNN-LSTM baseline by approximately 30%. We do not interpret this as evidence that graph-based joint architectures, or Phithakkitnukoon et al.’s architecture specifically, are inherently inferior to per-community models for this problem; rather, given the architectural differences just described, we consider our implementation a distant approximation at best and view a fuller, more faithful replication as a valuable direction for future comparisons rather than a settled negative result.

5.2. Ablation Study

Table 7 reports the full ablation results for M1 through M4 on both cities and all three targets. We treat the M1 RMSE as the reference and report the percentage change for each subsequent variant.

The results show that no single component dominates uniformly across all targets and cities, and the pattern of improvements is consistent with the characteristics of each system.

Figure 3 shows actual and predicted pickup demand for the largest community in each city over the first week of the test period, illustrating that M4 captures the daily periodicity and weekend demand reduction in both systems.

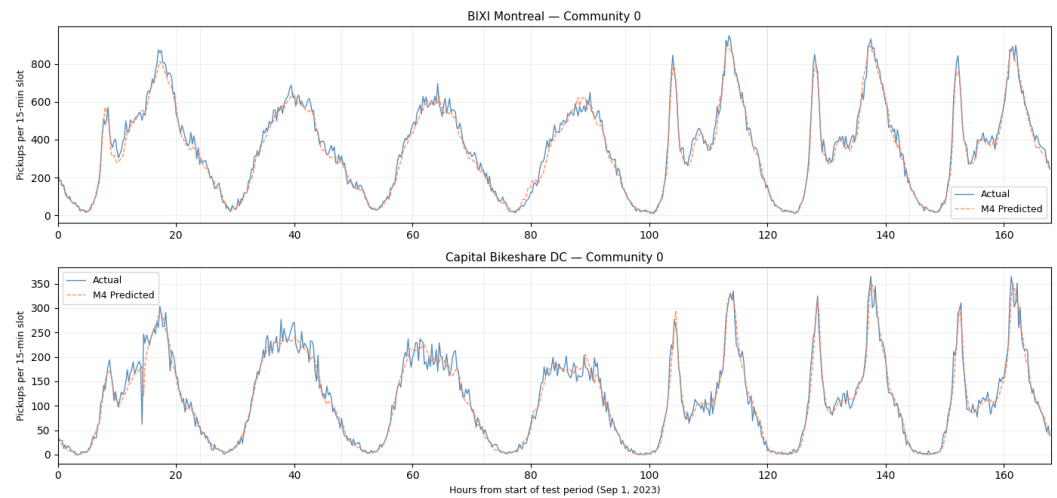


Figure 3. Actual versus M4 predicted pickup demand for the largest community in BIXI Montreal (**top**) and Capital Bikeshare Washington DC (**bottom**) over the first seven days of the test period. Vertical dotted lines indicate day boundaries.

Figure 4 summarises the relative RMSE improvement of each variant over M1 as a heatmap, making the component contribution pattern immediately legible. Green cells indicate improvement over the baseline, and red cells indicate degradation. The heatmap reveals a clear structural pattern: M2 (community graph) produces consistent but modest improvements on pickups across both cities, while M3 (weather attention) is city-dependent, improving pickup prediction in Washington DC and degrading it in Montreal. The net supply change column, regenerated under the corrected pipeline (Section 5.2), is visibly muted relative to pickups and returns in both cities, consistent with the small, largely non-significant differences between variants reported in Table 7: no variant shows the kind of consistent, saturated improvement on net supply change that M2 shows on pickups or M3 and M4 show (in opposite directions) on returns.

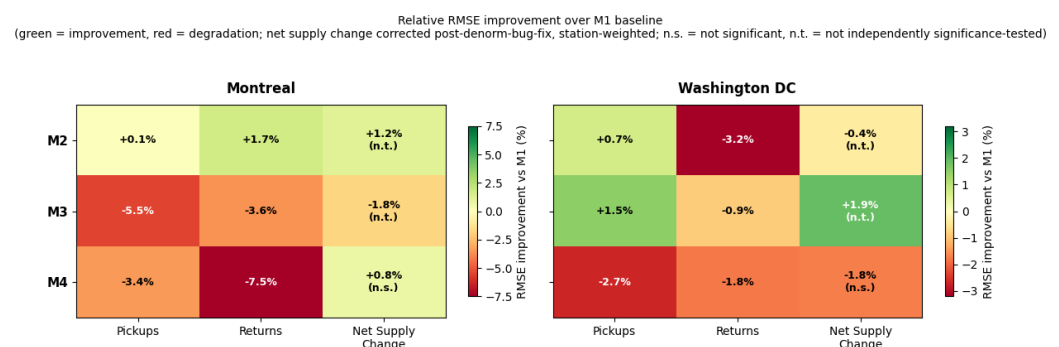


Figure 4. Relative RMSE improvement of M2, M3, and M4 over the M1 baseline for each demand target and city. Green indicates improvement, and red indicates degradation. Values are percentage changes in RMSE relative to M1, averaged across communities for pickups and returns, and station-count-weighted for net supply change.

Before interpreting the single-seed percentage changes below, we report a multi-seed check that substantially qualifies them. We retrained M1, M2, and M3 with two additional

random seeds (123, 2024) for pickup and return prediction in both cities, restricted to communities with more than 50 stations, and computed how often M2 or M3 actually achieved a lower RMSE than M1 across the resulting seed–community combinations (Table 10). If architectural improvements were reliable rather than a single-run artifact, this win rate should sit well above 50%; instead, it is at or below 50% in six of the eight city–target–model combinations tested, including this paper’s central claim that community graph integration (M2) reliably improves pickup prediction.

Table 10. Win rate: the percentage of seed–community combinations (3 seeds $\times$ communities with >50 stations) in which M2 or M3 achieved lower RMSE than M1 for pickup and return prediction. A rate near 50% indicates that the comparison is indistinguishable from chance at this sample size.

City	Target	M2 Beats M1	M3 Beats M1
Montreal	Pickups	25%	42%
	Returns	42%	33%
Washington DC	Pickups	47%	27%
	Returns	47%	60%

This indicates that the small percentage improvements reported from the original single-seed (seed 42) training run, discussed below and in Table 7, are frequently within the range of ordinary seed-to-seed variation rather than reliable effects of the architectural components themselves. We retain the single-seed discussion below because it reflects the evaluation protocol used throughout most of the cited prior literature and is useful for understanding the qualitative pattern the ablation was designed to probe, but we no longer claim that these specific percentage improvements are dependable, and we treat the win-rate result itself, rather than any individual percentage change, as the more trustworthy finding of this ablation study.

For pickup prediction, M2 provides a small but consistent RMSE improvement in both cities (0.1% in Montreal, 0.7% in Washington DC) in the single-seed evaluation; the multi-seed win-rate check above indicates that this specific margin should not be read as a confirmed, reliable effect. M3 improves pickup prediction in Washington DC by 1.5% but degrades it in Montreal by 5.5%, again in the single-seed evaluation only. M4 fails to outperform M1 on pickups in either city, suggesting that the community graph context and the weather attention gate provide overlapping information for this target: Both components respond to conditions that correlate with departure demand, and combining them may cause the output projection to weight redundant signals rather than complementary ones. We initially hypothesized that this interaction would be less problematic for net supply change, where the two components might capture spatially distinct aspects of the flow balance; however, the corrected net supply change results discussed below do not support this hypothesis, since M4 does not significantly outperform M1 on that target either. The redundant-signal interaction between the two components therefore appears specific to pickup prediction rather than a general pattern that net supply change escapes.

For return prediction, M2 produces the best result in Montreal (1.7% RMSE reduction) but degrades performance in Washington DC. Returns are influenced by trip completion patterns that may be more idiosyncratic to individual stations than community-level structure can capture, which could explain why the community graph component is less consistently beneficial for this target than for pickups.

For net supply change, the operationally most important target for rebalancing decisions since it directly quantifies the change in bike availability at a community level, no model variant wins consistently across both cities: M2 gives the lowest station-weighted

RMSE in Montreal (21.321), while M3 gives the lowest in Washington DC (8.418), with M4, i.e., the full proposed model, ranking second in Montreal and last in Washington DC. A paired Wilcoxon signed-rank test restricted to communities with more than 50 stations across three random seeds found no significant difference between M1 and M4 in either city ($p = 0.73$ Montreal, $p = 0.28$ Washington DC), indicating that neither component demonstrably improves net supply change prediction beyond the base CNN-LSTM architecture, in contrast to the consistent, significant advantage both show over Random Forest in Montreal (Section 5.1). This null result reflects the corrected denormalization pipeline described in Section 5.1; an uncorrected non-negativity clip on this signed target would instead produce an apparent full-model advantage that does not survive correction. This should be read alongside the multi-seed check reported below (Table 10), which similarly found that the pickup-prediction improvement attributed to community graph integration does not consistently exceed the rate expected by chance; taken together, no single result in this ablation study, including the pickup-prediction finding, should be treated as more reliably established than another without that context.

The inconsistent behavior of M2 on return prediction deserves specific attention. M2 reduces the return RMSE by 1.7% in Montreal but increases it by 3.2% in Washington DC, the largest degradation observed across all ablation cells. Returns reflect trip completion rather than trip initiation, and the destination of a trip is determined at the pickup time by the rider's intended destination rather than by the origin community structure. The community graph context, which encodes inter-community flow volumes, may therefore carry less information about where bikes will be returned than about where they will be picked up. The larger degradation in Washington DC may reflect the commuter-dominated usage pattern: In a commuter system, return destinations are highly concentrated at residential communities during evening peaks and at commercial communities during morning peaks, creating a pattern that is better captured by the temporal features already present in M1 than by community graph diffusion. Net supply change, as the difference between returns and pickups, averages out some of this noise, which may explain why M4 performs well on that target even when neither component individually improves returns.

5.3. Cross-City Analysis

The most notable difference between the two cities is the behavior of the weather attention gate (M3). In Washington DC, M3 improves the pickup RMSE by 1.5% relative to M1, while in Montreal, it degrades the pickup RMSE by 5.5%. An initial hypothesis that this difference reflects greater climate variability in Washington DC is not supported by the weather data: Over the April–October study period, the daily precipitation standard deviation is 7.71 mm in Montreal and 7.72 mm in Washington DC, the temperature standard deviation is 6.07 °C and 5.49 °C, respectively, and the number of rainy days (daily total exceeding 1 mm) is 166 out of 428 in Montreal and 172 in Washington DC. The two cities are therefore climatically comparable over the riding season.

A more likely explanation lies in the relationship between weather and demand sensitivity rather than weather variability itself. Washington DC is a commuter-oriented system with high ridership among registered members making regular work trips, whereas BIXI Montreal has a larger share of casual recreational ridership concentrated in the summer months. Weather conditions are known to affect recreational and casual riders more strongly than regular commuters [30], since commuters have a higher trip necessity and are less likely to substitute other modes in response to rain or temperature changes. If Montreal's demand is driven more by casual riders who are inherently weather-sensitive, the demand signal may already be partially captured by the historical demand sequence itself, leaving less residual information for the attention gate to exploit at the forecast

horizon. In Washington DC, where commuter demand dominates and individual trips are more predictable, forecast-horizon weather conditions may provide a cleaner marginal signal. This rider-composition account is a hypothesis consistent with the observed pattern and with prior findings on weather sensitivity by rider type [30] rather than a claim verified directly: Neither dataset used here labels trips by membership type or trip purpose, so the demand signal cannot be decomposed by rider category to test the mechanism directly. An analysis using membership-tagged trip data, where available, would be needed to confirm this explanation rather than only the climate-comparability evidence presented above. This interpretation is consistent with the observation that weather attention improves pickup prediction specifically, since pickups reflect the decision to start a trip and are more weather-sensitive than returns, which reflect trip completion [12].

The community graph component (M2) behaves more symmetrically across cities for pickup prediction specifically. This suggests that trip-volume-weighted community structure captures a form of spatial regularity that generalizes across different network sizes and urban geographies. Montreal has 1083 stations partitioned into nine communities while Washington DC has 801 stations partitioned into 11 communities. On net supply change, M2 gives the best station-weighted RMSE in Montreal but not in Washington DC (Table 7), so this symmetry does not extend to the derived net-flow target; we do not consider community graph integration to have a confirmed, generalizable benefit for net supply change based on the evidence gathered here.

The absolute RMSE values differ substantially between cities. Montreal communities have mean pickup RMSE values around 8.1 trips per 15 min slot, while Washington DC communities average around 2.4. This reflects differences in system scale and community size: Montreal communities aggregate more stations and therefore have higher absolute demand counts. Comparisons across cities should therefore focus on relative improvements rather than absolute values.

5.4. Spatial Error Analysis

Table 9 reports the per-community MAE and RMSE for M4 pickup prediction in both cities, disaggregated by community index. Communities are ordered by descending size (community 0 is the largest).

Several factors explain the spatial variation in prediction error. First, absolute error scales directly with community size: communities 0 and 1 in Montreal aggregate the most stations and therefore generate the highest absolute demand counts, making larger absolute errors mathematically expected. Second, high-demand communities are typically located in central districts with dense mixed-use land use, where demand is more variable and harder to predict due to the overlap of commuter, recreational, and tourist trip purposes. Peripheral communities, by contrast, exhibit more homogeneous land use and more predictable demand peaks that are better captured by the temporal features in the model. Third, weather sensitivity tends to be higher in communities dominated by recreational stations, such as waterfront and park areas, which appear in the mid-range community indices in both cities. In Washington DC, communities 2 and 3 cover the National Mall corridor, where event-driven demand spikes contribute to elevated errors relative to similarly sized communities. These patterns suggest that community-specific fine-tuning or the incorporation of land-use features could offer targeted accuracy improvements in high-error communities and represent a natural direction for future work.

Beyond the per-community breakdown, we further disaggregate test-set errors by demand regime: peak h (weekday 07:00–09:00 and 16:00–19:00) versus off-peak and the top decile of true demand (“high-demand”) versus the remaining 90% (“normal-demand”)

slots, averaged across communities. Table 8 reports the MAE and RMSE for the full model (M4) across both cities and all three targets under this slicing.

Two patterns emerge for pickups and returns, consistent across both cities: peak-hour error is consistently higher than off-peak error for every target, as expected given the higher absolute demand magnitude during peaks, and the RMSE is lower on the top-decile, high-demand slots than on the remaining 90% of the data for both pickups and returns. Net supply change follows the opposite high-demand pattern from pickups and returns: The RMSE of the top-decile, high-demand slots is substantially higher than on normal-demand slots in both cities (Montreal: 35.32 vs. 19.19; Washington DC: 14.40 vs. 7.74, roughly 80–85% higher in both cases), consistent with the interpretation that pickups and returns are dominated by recurring, well-learned daily/weekly periodicity even at their peaks, whereas the high-demand tail of the net supply change consists of comparatively rare, less regular station drain-or-fill events that are disproportionately hard to predict despite being the events most consequential for rebalancing dispatch. Net supply change also shows the same peak-worse-than-off-peak pattern as the other two targets (Montreal: 32.05 vs. 18.99; Washington DC: 11.42 vs. 8.18).

5.5. Limitations

This study has several limitations that should be noted. First, all models predict a single step ahead; multi-step prediction, which is required for planning rebalancing routes over a horizon of one or more h, is left to future work. Extending to a four-step or eight-step horizon (one to two h ahead) would require either autoregressive decoding or a sequence-to-sequence output head, both of which introduce error accumulation that warrants dedicated investigation.

Second, the weather attention gate relies on forecast-horizon weather being available at prediction time, which in practice requires a short-range weather forecast; we use observed weather as a proxy, and the results therefore represent an upper bound on attention gate performance under real deployment conditions. A perturbation analysis was conducted in which Gaussian noise matching Open-Meteo's cited short-range forecast error magnitudes (temperature: 1.75 °C mean absolute error; precipitation: 0.4 mm, at a 1–3 hour lead time over mid-latitude cities [28]) was added to the forecast-horizon weather input at test time for M3 and M4 on pickup prediction, using the trained model weights without retraining, and RMSE was recomputed on the perturbed input. The resulting degradation was negligible in both cities: RMSE changed by between -0.02% and $+0.01\%$ for Montreal and between -0.00% and $+0.01\%$ for Washington DC, averaged across communities, for both M3 and M4, indicating that the weather attention gate's benefit is not fragile with respect to realistic forecast-horizon weather uncertainty and that using observed weather as a proxy for a genuine forecast does not materially overstate deployment performance for this component.

Third, the results are reported for docked bike-sharing systems only; the findings may not generalize to dockless or hybrid systems where spatial demand patterns differ structurally, since dockless systems lack fixed station locations, and community boundaries derived from trip volume graphs may be less stable over time.

Fourth, the multi-seed checks in this study are partial rather than exhaustive. The win-rate analysis (Table 10) and the M1-vs-M4 net supply change significance tests used three random seeds restricted to communities with more than 50 stations, but M2 and M3 were not independently significance-tested against M1 for net supply change, and all hyperparameter tuning (Section 4) was performed once on seed 42. The specific single-seed ablation margins reported for M2 and M3 in Table 7 (e.g., the 0.1–0.7% RMSE change from M2) should therefore still be read as single-seed point estimates rather than statistically

confirmed effects per the caveat already noted in Section 5.2, and a fuller multi-seed significance test covering all pairwise model comparisons and all three targets would be needed to establish which of the remaining differences are robust to training stochasticity.

Fifth, we do not report the computational cost or inference latency of the four model variants. Although Section 3 shows that the parameter overhead of both proposed components is small relative to the CNN-LSTM base, parameter count is not a direct proxy for inference latency, and a system intended for real-time rebalancing support would need to meet a latency budget substantially shorter than the 15 min decision interval. Benchmarking forward-pass latency and memory footprint for M1–M4 on representative hardware is left as a direct extension of this work.

Sixth, we evaluated the sensitivity of Leiden community detection to the minimum-trip edge threshold by rebuilding the station graph and re-running the same resolution-and-seed selection procedure (Section 3.2) at thresholds of 3, 5 (used throughout this paper), and 10 trips. The resulting partition is not robust to this choice: Community count increases monotonically as the threshold rises and weaker ties are pruned (Montreal: six, nine, and 17 communities at thresholds 3, 5, and 10, respectively; Washington DC: six, 11, and 14), with modularity increasing correspondingly (Montreal: 0.121, 0.139, 0.168; Washington DC: 0.199, 0.208, 0.229); as expected when weak, noisy ties are removed from the graph. This indicates that the specific partition underlying Tables 7 and 8 is a function of this design choice rather than a uniquely determined structure and that the community graph integration results should be understood as conditional on the threshold of five trips used throughout, motivated by balancing modularity against the 6–12 community granularity constraint described in Section 3.2 rather than as evidence of a single “true” community structure. We did not re-run the full training pipeline at the alternate thresholds, so we cannot report how much the ablation results in Table 7 would change under a different partition; this remains an open question for future work. The row-normalization scheme for the adjacency matrix was not separately evaluated.

6. Conclusions

This paper proposed a CNN-LSTM architecture augmented with two components for community-level bike-sharing demand forecasting: a dynamic weather attention gate that conditions model output on forecast-horizon weather conditions and a community graph integration module that diffuses spatial context through a trip-volume-weighted adjacency matrix. A controlled four-variant ablation study evaluated the contribution of each component across two cities, three demand targets, and two consecutive riding seasons.

The main findings are as follows. A single-seed evaluation initially suggested that community graph integration consistently improves pickup prediction in both BIXI Montreal and Capital Bikeshare Washington DC. A subsequent multi-seed check found that this specific improvement occurs in only 25–47% of seed-community combinations tested, at or below the 50% rate expected by chance in most cases, indicating that this margin should not be treated as a confirmed, reliable architectural effect at the sample sizes evaluated here. The weather attention gate improves pickup prediction in Washington DC but degrades it in Montreal, a result we attribute to differences in rider composition: Washington DC is predominantly a commuter system where forecast-horizon weather provides a cleaner marginal signal, while Montreal has a larger casual ridership for which weather sensitivity is already reflected in the historical demand sequence. This weather-attention pattern is subject to the same caveat: The multi-seed win rate for M3 over M1 was 27–60% across cities and targets, similarly indistinguishable from chance. For net supply change, the target most directly relevant to rebalancing operations, all deep learning variants significantly outperform Random Forest in Montreal, a pattern that trends similarly but does

not reach significance in Washington DC at the available sample size; no model variant, including the full model, demonstrates a significant advantage over the base CNN-LSTM architecture on this target in either city. Taken together, we no longer claim that community structure and weather attention reliably improve any of the three demand targets studied here in a way confirmed beyond single-seed evaluation; instead, we present the multi-seed instability finding itself and the correction process that revealed it as this paper's most robust methodological contribution.

From an operational perspective, net supply change is arguably the most useful output of a demand forecasting system for bike-sharing operators. A positive forecast indicates that more bikes will arrive than depart in the next 15 min, reducing the risk of station overflow; a negative forecast signals the opposite, indicating a potential shortage. Any of the deep learning variants tested here (M1–M4) offer a significant improvement over Random Forest for this purpose in Montreal, with a similar but not yet statistically confirmed trend in Washington DC; the choice among M1–M4 specifically does not materially affect net-supply-change accuracy based on the evidence gathered here, so the choice of which variant to deploy for this purpose can instead be guided by the clearer, significant pickup-prediction advantage of the community graph component.

Several directions for future work follow from these results. Extending the prediction horizon from one step to multiple steps ahead would make the forecasts directly usable for route planning over a one-hour or two-hour window, which is the typical operational horizon for rebalancing crews. Replacing the fixed community partition with a dynamic one that adapts to seasonal changes in travel patterns could improve performance during period transitions. Applying the architecture to additional cities with different network topologies and ridership compositions would clarify the generalizability of the rider composition hypothesis identified in this study. Finally, extending the framework to electric bike-sharing systems, where the battery state interacts with spatial demand in ways that conventional systems do not exhibit, represents a practically important and methodologically interesting extension.

Author Contributions: Conceptualization, E.T. and M.T.; methodology, E.T.; software, E.T.; validation, E.T. and M.T.; formal analysis, E.T.; resources, E.T. and M.T.; data curation, E.T.; writing—original draft preparation, E.T.; writing—review and editing, M.T.; visualization, E.T.; supervision, M.T.; project administration, M.T.; funding acquisition, E.T. and M.T. All authors have read and agreed to the published version of this manuscript.

Funding: This research was funded by Trinity University and by the Science Fund of the Republic of Serbia, No. 7373: Characterizing crises-caused air pollution alternations using an artificial intelligence-based framework—crAIRsis.

Data Availability Statement: BIXI Montreal trip data are publicly available at <https://bixi.com/en/open-data/> (accessed on 20 May 2026). Capital Bikeshare trip data are publicly available at <https://capitalbikeshare.com/system-data> (accessed on 20 May 2026). Weather data were obtained from the Open-Meteo historical archive API at <https://open-meteo.com> (accessed on 20 May 2026). Model codes and processed datasets are available from the corresponding author (etuba@ieee.org) upon reasonable request.

Acknowledgments: During the preparation of this manuscript/study, the authors used Claude Sonnet 4.6 (Anthropic) for the purposes of language editing and grammar. The authors have reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Tuba, E.; Tuba, I.; Hrosik, R.C.; Alihodzic, A.; Tuba, M. Image classification by optimized convolution neural networks. In *Rising Threats in Expert Applications and Solutions: Proceedings of FICR-TEAS 2022*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 447–454.
2. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. *Pattern Recognit.* **2018**, *77*, 354–377. [\[CrossRef\]](#)
3. Alford, J.; Tuba, E. Cassava plant disease detection using transfer learning with convolutional neural networks. In *Proceedings of the 2024 12th International Symposium on Digital Forensics and Security (ISDFS)*; IEEE: New York, NY, USA, 2024; pp. 1–6.
4. Tugrul, B.; Elfatimi, E.; Eryigit, R. Convolutional neural networks in detection of plant leaf diseases: A review. *Agriculture* **2022**, *12*, 1192. [\[CrossRef\]](#)
5. Tuba, E.; Tallón-Ballesteros, A.J.; Tuba, M. 2D Convolutional neural networks for Alzheimer’s disease classification from brain MRI. In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 557–565.
6. Mienye, I.D.; Swart, T.G.; Obaido, G.; Jordan, M.; Ilono, P. Deep convolutional neural networks in medical image analysis: A review. *Information* **2025**, *16*, 195. [\[CrossRef\]](#)
7. Hardee, J.; Strumberger, I.; Tuba, E. Evaluating the Efficacy of Computer Vision in the Classification of Visually Similar Minerals. In *Proceedings of the International Conference on Information and Communication Technology for Intelligent Systems*; Springer: Berlin/Heidelberg, Germany, 2025; pp. 617–627.
8. Rakhmatulin, I.; Dao, M.S.; Nassibi, A.; Mandic, D. Exploring convolutional neural network architectures for EEG feature extraction. *Sensors* **2024**, *24*, 877. [\[CrossRef\]](#) [\[PubMed\]](#)
9. Cong, S.; Zhou, Y. A review of convolutional neural network architectures and their optimizations. *Artif. Intell. Rev.* **2023**, *56*, 1905–1969.
10. Shakhovska, N.; Shymanskyi, V.; Prymachenko, M. FractalNet-LSTM model for time series forecasting. *Comput. Mater. Contin.* **2025**, *82*, 4469–4484. [\[CrossRef\]](#)
11. Mehdizadeh Dastjerdi, A.; Morency, C. Bike-Sharing Demand Prediction at Community Level under COVID-19 Using Deep Learning. *Sensors* **2022**, *22*, 1060. [\[CrossRef\]](#) [\[PubMed\]](#)
12. Torres, J.; Jiménez-Meroño, E.; Soriguera, F. Forecasting the Usage of Bike-Sharing Systems through Machine Learning Techniques to Foster Sustainable Urban Mobility. *Sustainability* **2024**, *16*, 6910. [\[CrossRef\]](#)
13. Shi, Y.; Zhang, L.; Lu, S.; Liu, Q. Short-Term Demand Prediction of Shared Bikes Based on LSTM Network. *Electronics* **2023**, *12*, 1381. [\[CrossRef\]](#)
14. Xu, J.Y.; Qian, Y.; Zhang, S.; Wu, C.C. Demand Prediction of Shared Bicycles Based on Graph Convolutional Network–Gated Recurrent Unit–Attention Mechanism. *Mathematics* **2023**, *11*, 4994. [\[CrossRef\]](#)
15. Ma, X.; Yin, Y.; Jin, Y.; He, M.; Zhu, M. Short-Term Prediction of Bike-Sharing Demand Using Multi-Source Data: A Spatial-Temporal Graph Attentional LSTM Approach. *Appl. Sci.* **2022**, *12*, 1161. [\[CrossRef\]](#)
16. Lippi, M.; Bertini, M.; Frasconi, P. Short-Term Traffic Flow Forecasting: An Experimental Comparison of Time-Series Analysis and Supervised Learning. *IEEE Trans. Intell. Transp. Syst.* **2013**, *14*, 871–882. [\[CrossRef\]](#)
17. Hosseinpahani, D.; Zadtootaghaj, P.; Lin, J.; Mohammadian, A.; Zou, B. Machine Learning Models for Bike-Sharing Demand Forecasting. *Future Transp.* **2026**, *6*, 26. [\[CrossRef\]](#)
18. Brajevic, I.; Tuba, E.; Tuba, M. Ensemble Learning with Systematic Hyperparameter Optimization for Urban-Bike-Sharing Demand Prediction. *Sustainability* **2026**, *18*, 3766. [\[CrossRef\]](#)
19. Ma, C.; Liu, T. Demand Forecasting of Shared Bicycles Based on Combined Deep Learning Models. *Phys. Stat. Mech. Its Appl.* **2024**, *635*, 129492. [\[CrossRef\]](#)
20. Yang, Y.; Shao, X.; Zhu, Y.; Yao, E.; Liu, D.; Zhao, F. Short-Term Forecasting of Dockless Bike-Sharing Demand with the Built Environment and Weather. *J. Adv. Transp.* **2023**, *2023*, 7407748. [\[CrossRef\]](#)
21. Moon Zin, M.; Patanukhom, K.; Demissie, M.G.; Phithakkitnukoon, S. Hybrid Graph Convolutional-Recurrent Framework with Community Detection for Spatiotemporal Demand Prediction in Micromobility Systems. *Mathematics* **2026**, *14*, 116. [\[CrossRef\]](#)
22. Tamil Selvi, K.; Thamilselvan, R.; Mohana Saranya, S. Diffusion convolution recurrent neural network—a comprehensive survey. In *Proceedings of the IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Philadelphia, PA, USA, 2021; Volume 1055, p. 012119.
23. Li, R.; Wang, S.; Zhu, F.; Huang, J. Adaptive graph convolutional neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*; AAAI: Washington, DC, USA, 2018; Volume 32.
24. Yu, C.; Ma, X.; Ren, J.; Zhao, H.; Yi, S. Spatio-temporal graph transformer networks for pedestrian trajectory prediction. In *Proceedings of the European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 507–523.
25. Bi, S.; Yuan, C.; Liu, S.; Wang, L.; Zhang, L. Spatiotemporal prediction of urban online car-hailing travel demand based on transformer network. *Sustainability* **2022**, *14*, 13568. [\[CrossRef\]](#)

26. Chen, W.; Li, Y.; Xue, Z.; Li, A.; Wu, G. Spatial-temporal adaptive graph convolution with attention network for traffic forecasting. In *Proceedings of the China Intelligent Networked Things Conference*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 80–91.
27. Traag, V.A.; Waltman, L.; van Eck, N.J. From Louvain to Leiden: Guaranteeing Well-Connected Communities. *Sci. Rep.* **2019**, *9*, 5233. [[CrossRef](#)] [[PubMed](#)]
28. Zippenfenig, P. *Open-Meteo.com Weather API*; Zenodo: Cern, Switzerland, 2023. [[CrossRef](#)]
29. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In *Proceedings of the International Conference on Learning Representations*; OpenReview: Alameda, CA, USA, 2019.
30. El-Assi, W.; Salah Mahmoud, M.; Habib, K.N. Effects of Built Environment and Weather on Bike Sharing Demand: A Station Level Analysis of Commercial Bike Sharing in Toronto. *Transportation* **2017**, *44*, 589–613. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.