HFedBike: A Hybrid Federated Learning System for Urban Bike-Sharing Demand Forecasting

Nikos Andrianopoulos, Andreas Komninos, Spyros Sioutas, Gerasimos Vonitsanos
Department of Computer Engineering and Informatics, University of Patras, Greece
nick.andrianopoulos.work@gmail.com, {akomninos, sioutas, mvonitsanos}@ceid.upatras.gr

***Abstract*—This study presents a hybrid federated learning system for urban bike-demand forecasting that balances distributed training with controlled coordination. The approach introduces a structured communication scheme in which local model updates are exchanged within small groups and selectively propagated across the network, thereby avoiding the need for a central aggregation point. The system is evaluated on large-scale, real-world mobility and weather data in a realistic multi-node deployment with heterogeneous edge devices. Results show consistent improvements in predictive accuracy over non-collaborative and temporally adaptive approaches, while maintaining stable operation across all evaluation cycles. In addition, the proposed design significantly reduces communication overhead, demonstrating its effectiveness as a practical and scalable solution for real-world smart mobility scenarios.**

***Index Terms*—Federated Learning, Edge Computing, Smart Mobility, Bike-Sharing**

I. Introduction

Urban bike-sharing systems generate continuous streams of station-level data that are essential for short-term demand forecasting and operational planning. Beyond forecasting, prior work has also examined system-level optimization strategies, such as adaptive incentive mechanisms for spatial balancing, demonstrating the importance of decentralized and data-driven decision-making in improving system efficiency [1]. Conventional centralized approaches, however, require collecting all data on a central server, leading to increased communication overhead and limited scalability.

Federated Learning (FL) [2] enables distributed model training without transferring raw data; however, typical FL architectures rely on a central aggregator, while fully decentralized approaches introduce additional coordination complexity [3], [4].

Hierarchical federated learning approaches, such as hFedLAP [5], have explored communication-efficient coordination through structured aggregation schemes. Inspired by this line of work, the proposed architecture builds upon these ideas by introducing a hybrid design that combines a lightweight, non-aggregating router with ring-based neighborhood consensus, aiming to extend existing approaches with an alternative communication and coordination strategy tailored to spatially heterogeneous environments.

This study presents HFedBike, a hybrid federated learning system that organizes clients into fixed neighborhoods for local model propagation. To maintain a controlled training protocol without excessive synchronization complexity, the architecture employs a passive central router for inter-neighborhood communication. Unlike in traditional FL, this router is excluded from the aggregation process and focuses solely on message forwarding between neighborhood leaders.

In this context, the proposed design balances coordination efficiency and distributed learning by limiting centralized dependencies while preserving structured communication, enabling scalable learning across heterogeneous urban environments without excessive synchronization complexity.

Forwarded from the research track.

II. Related Work

Federated Learning (FL) was introduced by McMahan et al. [2] through the Federated Averaging (FedAvg) algorithm, establishing the standard centralized training paradigm. To overcome the scalability and communication limitations of this approach, decentralized and peer-to-peer alternatives have been proposed. For instance, BrainTorrent [3] employs a ring-all-reduce topology to eliminate the need for a central server, while gossip-based methods [4] reduce communication overhead through partial model exchanges.

Hybrid approaches aim to balance decentralization and coordination. Architectures such as hFedLAP [6] organize clients into groups, with designated nodes performing local aggregation and coordinating inter-group communication. Although effective in improving scalability, these methods introduce additional computational responsibilities at the aggregator level. Building on hierarchical approaches such as hFedLAP and BrainTorrent, HFedBike adopts a hybrid design that includes a non-computational routing layer for inter-group communication, simplifying coordination across nodes.

In the context of smart-city applications, federated learning has been applied to mobility and transportation systems, including demand forecasting in bike-sharing networks. These approaches enable distributed learning across stations while avoiding centralized data collection, supporting scalable and adaptive urban services [7].

III. Methodology

A. System Overview

The proposed architecture, HFedBike, is a partially decentralized Federated Learning (FL) framework designed to improve demand prediction in bike-sharing systems while reducing communication overhead. In contrast to centralized FL,

where a single server aggregates all client updates, and fully decentralized approaches that often introduce synchronization complexity, HFedBike adopts a hierarchical structure based on static neighborhoods.

This design differentiates the proposed approach from prior hierarchical FL architectures, where intermediate nodes typically undertake aggregation tasks. To ensure reproducibility and facilitate deployment, the system is fully containerized using Docker and docker-compose, enabling seamless execution across heterogeneous edge devices.[1].

For evaluation, three models are considered: (i) a static baseline trained on bike usage data from 2023 along with corresponding hourly weather features, applied without further updates, (ii) a bias-adapted baseline that updates only its output-layer bias on a weekly basis, and (iii) the proposed federated model. The bias-adapted model serves as an online comparator, isolating the effect of temporal adaptation from that of collaborative learning, as it follows the same update schedule without inter-node information exchange. Statistical significance of performance differences is assessed using the Wilcoxon signed-rank test [8].

B. Datasets and Preprocessing

The primary datasets used in this study consist of Citi Bike trip [9] history data for New York City and hourly weather data obtained from OpenWeatherMap [10]. The data cover two periods: the full year of 2023 for model training (approximately 35 million trips) and January–February 2024 for testing (approximately 4 million trips), ensuring a realistic evaluation while maintaining computational efficiency.

The bike-sharing data include trip-level information such as station locations and timestamps, which were aggregated at an hourly level to produce time-indexed usage counts per station. Weather data, including temperature, humidity, wind speed, and precipitation, were temporally aligned with the usage data. For the 2024 evaluation period, stations were partitioned using K-means clustering into four regions, each further divided into four subzones, resulting in 16 subzones corresponding to federated nodes.

Following spatial partitioning, bike usage and weather data were merged using hourly timestamps, and the dataset was shifted by one time step to predict the next-hour demand. Temporal features were encoded using sine and cosine transformations, numerical variables were normalized via Z-score normalization [11] and categorical weather variables were one-hot encoded [12]. Missing weather values were handled using linear interpolation for continuous variables and forward filling for categorical conditions, preserving temporal consistency. Model performance is evaluated on a weekly basis across successive time windows.

To ensure consistency across nodes in the federated setting, all subzones share a common feature space and model configuration, while training is performed exclusively on local data partitions. This design enables each node to capture localized demand patterns while remaining compatible with model aggregation. Furthermore, the weekly evaluation protocol reflects a realistic deployment scenario in which models are updated incrementally over time, allowing the system to adapt to evolving usage trends without requiring full retraining.

[1]Source code and deployment instructions are available at: https://github.com/Nikolis2002/p2p-federated-learning-demand-forecasting

C. ML Predictive Models

1) Baseline Model: The baseline model is trained on the hourly aggregated dataset from 2023 and is used to predict bike usage for the subsequent hour at the city level. The input consists of temporal and meteorological features, while the target variable corresponds to the bike usage at the next time step.

A feedforward neural network [13] with a fixed structural design is employed, consisting of an input layer, three fully connected hidden layers, and an output layer with a single unit. The hidden layers utilize the ReLU activation function, while L2 regularization is applied to all hidden layers to mitigate overfitting. The training process incorporates early stopping and a dynamic learning rate to ensure stable convergence.

To identify an appropriate model configuration, an extensive hyperparameter search was conducted. Specifically, multiple combinations of network architectures were evaluated, considering 32, 64, 128, and 256 neurons per hidden layer, along with different optimization algorithms, including Nadam, Adam, and SGD with momentum and Nesterov acceleration. Additional parameters, such as learning rate, regularization strength, and number of training epochs, were also varied.

Model selection was performed using a leave-some-out cross-validation strategy with five folds, where stratification was applied based on the month variable to preserve seasonal patterns. The selection criterion was the Mean Squared Error (MSE) [14], and the best-performing configuration across folds was retained. In total, 1,734 different configurations were evaluated.

The final selected model corresponds to a fully connected architecture with three hidden layers consisting of 64, 128, and 128 neurons, respectively. Each hidden layer employs the ReLU activation function and applies L2 regularization with a coefficient of 0.0001, while no dropout is used. The model is trained using the Nadam optimizer with a learning rate of 0.001. Although training was configured for 1,100 epochs, early stopping resulted in convergence after 69 epochs. This configuration provided the best trade-off between predictive performance and model complexity.

2) Federated Model: The federated model [15] simulates a distributed learning environment in which 16 nodes, corresponding to spatial subzones, collaboratively train a predictive model without sharing raw data. Each node processes a local stream of hourly bike usage along with corresponding weather features, which are assumed to be globally consistent.

At each time step, nodes construct input feature vectors and generate next-hour predictions. Training is performed incrementally, and after processing one week of data (168 hourly

samples), local models are updated, and their parameters are prepared for aggregation.

To address statistical heterogeneity, local outputs are normalized using a running mean and variance estimation based on Welford's algorithm [16], enabling stable normalization in a streaming setting. The normalization is applied to the target variable, while input features remain unscaled. At the end of each weekly cycle, input normalization is reset, whereas target normalization is retained to ensure stability.

The nodes are organized into four fixed neighborhoods of four nodes each. Within each neighborhood, model parameters are exchanged via a ring-based iterative consensus protocol, in which nodes update their models by pairwise averaging with their neighbors, converging to a neighborhood-level representation maintained by a designated leader.

Each neighborhood leader transmits its aggregated model to a central coordinator following a star-topology communication pattern. The coordinator acts solely as a lightweight router, redistributing models to other leaders without performing aggregation, while optionally attaching routing metadata.

Finally, each leader integrates its local model with those received from other neighborhoods using a federated averaging strategy and broadcasts the updated model within its neighborhood. This hierarchical process enables efficient communication while capturing both local and global usage patterns.

IV. Results

A. Experimental Setup

To evaluate the proposed architecture under realistic deployment conditions, two experimental configurations were implemented.

In the first configuration, all 16 subzones were deployed as isolated Docker containers, each representing a logical microdevice. The containers were interconnected via a macvlan network, enabling direct TCP/IP communication and closely emulating a distributed edge environment.

In the second configuration, a hybrid deployment was used to assess performance under heterogeneous hardware conditions. To further validate the proposed architecture under realistic conditions, one neighborhood was deployed on resource-constrained hardware, including a 2 GB Raspberry Pi 4. The system was successfully executed over the full evaluation period without failures, demonstrating the robustness of the approach in edge environments. All nodes communicated over the same IP-based network.

This setup allows the evaluation of scalability, communication efficiency, and hardware heterogeneity in a realistic edge computing scenario. This configuration demonstrates that the proposed system remains functional under constrained hardware conditions, supporting its scalability and applicability to heterogeneous edge environments.

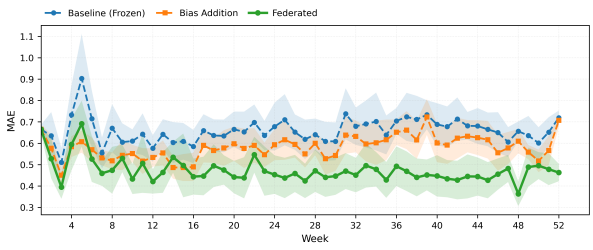


Fig. 1. Comparison of MAE between the baseline model and the federated learning model.

B. Performance Improvement

The performance of the three models is evaluated across 52 common evaluation points, each corresponding to a fixed window of 168 consecutive hourly observations.

Figure 1 presents the MAE across 52 evaluation weeks for the three models. The federated model consistently achieves lower error than both the frozen and bias-adapted models, while the bias-adapted model provides moderate improvement over the baseline. The federated approach demonstrates more stable and accurate performance over time.

The results demonstrate a consistent improvement pattern across the majority of subzones. The transition from the frozen baseline to the bias-adapted model yields moderate gains, typically ranging between approximately 3% and 11%, indicating that simple temporal adaptation provides limited but stable improvements.

TABLE I
Relative MAE improvement (%) and Wilcoxon p-values

Sub	B/Frz	Fed/Bias	Fed/Frz	B/Fr	Fed/Bias	Fr/Fed
A	4.1	38.4	41.0	.001	.000	.000
B	11.0	11.9	21.6	.005	.002	.000
C	2.4	47.0	48.3	.038	.000	.000
D	4.8	37.7	40.6	.149	.000	.000
E	4.6	40.3	43.1	.001	.000	.000
F	7.3	28.1	33.4	.000	.000	.000
G	6.3	31.6	35.9	.000	.000	.000
H	5.5	35.5	39.1	.000	.000	.000
L	6.6	30.9	35.4	.000	.000	.000
P	3.6	48.3	50.2	.043	.000	.000
Q	7.3	28.8	34.0	.000	.000	.000
R	6.6	31.3	35.9	.000	.000	.000
W	31.8	-48.1	-1.1	.000	.000	.606
X	26.1	-27.9	5.5	.000	.000	.363
Y	20.8	-24.7	1.3	.000	.000	.903
Z	34.7	-61.1	-5.3	.000	.000	.160

A substantially stronger effect is observed when comparing the federated model to the bias-adapted model, as presented the Table I In most subzones, the federated approach achieves significant reductions in error, frequently exceeding 30% and reaching up to 48.3% in subzone P and 47.0% in subzone C. These improvements are also reflected in the absolute MAE values, as depicted in Table I, where the federated model consistently outperforms both baselines (e.g., subzone C: 0.552 $\rightarrow$ 0.285, subzone P: 0.519 $\rightarrow$ 0.258).

The comparison against the frozen baseline further confirms this trend, with improvements exceeding 40% in several subzones and reaching up to 50.2%. The corresponding Wilcoxon signed-rank tests indicate statistically significant differences in nearly all cases, demonstrating the robustness of the observed gains.

TABLE II
PER-SUBZONE MAE ACROSS MODELS

Subzone	Freeze	Bias	Federated
A	0.687	0.659	**0.406**
B	0.458	0.408	**0.359**
C	0.552	0.539	**0.285**
D	0.485	0.462	**0.288**
E	0.633	0.604	**0.360**
F	0.796	0.738	**0.530**
G	0.759	0.712	**0.487**
H	0.694	0.656	**0.423**
L	0.750	0.700	**0.484**
P	0.519	0.500	**0.258**
Q	0.775	0.718	**0.511**
R	0.741	0.692	**0.475**
W	0.874	**0.596**	0.883
X	0.592	**0.438**	0.560
Y	0.556	**0.440**	0.548
Z	0.683	**0.446**	0.719

However, this behavior is not uniform. Table II depicts a small subset of subzones (W, X, Y, and Z) exhibits limited or negative improvement when transitioning from the bias-adapted to the federated model, with reductions reaching -61.1%. This is also evident in the absolute MAE values, where the federated model performs worse than the bias-adapted baseline in these cases. The lack of statistical significance in these subzones suggests that the degradation is not systematic but rather linked to localized data characteristics.

These findings indicate that while the federated approach effectively captures shared spatial patterns across most regions, it may introduce noise in subzones with distinct local dynamics. This behavior is consistent with the use of uniform aggregation, ignoring node heterogeneity.

The results demonstrate that the proposed architecture achieves substantial and statistically significant improvements in the majority of subzones, while highlighting the importance of heterogeneity-aware aggregation strategies as a direction for future work.

V. CONCLUSIONS AND FUTURE WORK

This study presented a hierarchical federated learning architecture for bike-sharing demand prediction under realistic smart city conditions. The proposed approach combines neighborhood-level collaboration with lightweight coordination, enabling efficient distributed training across spatially distributed subzones. The results indicate improved performance over both static and temporally adaptive baselines, highlighting the ability of the model to capture spatial dependencies beyond local updates.

The architecture was also validated under heterogeneous deployment settings, including resource-constrained devices, demonstrating its applicability in edge environments. These findings suggest that combining local adaptation with controlled inter-node collaboration can effectively model urban mobility patterns while maintaining efficient communication.

Future work will include comparisons with clustering-based non-IID federated learning methods and hierarchical FL approaches such as hFedLAP, as well as evaluation against a centralized online baseline to better assess performance trade-offs. Additional directions involve systematic ablation studies of key design choices, including neighborhood formation, update scheduling, and coordination mechanisms.

Furthermore, a more detailed analysis of communication efficiency, including bandwidth savings and additional performance metrics such as latency and resource utilization, is planned for future work. Finally, extensions toward production-level deployment will address robustness through fault-tolerant leader mechanisms and scalability to larger and more dynamic network configurations.

REFERENCES

[1] A. Komninos, G. Vonitsanos, I. Simou, and J. Garofalakis, "Adaptive and mixed incentive strategies for spatial balancing in bike-sharing systems," in *2025 IEEE International Smart Cities Conference (ISC2)*. IEEE, 2025, pp. 1–6.

[2] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Artificial intelligence and statistics*. Pmlr, 2017, pp. 1273–1282.

[3] A. G. Roy, S. Siddiqui, S. Pölsterl, N. Navab, and C. Wachinger, "Braintorrent: A peer-to-peer environment for decentralized federated learning," *arXiv preprint arXiv:1905.06731*, 2019.

[4] C. Hu, J. Jiang, and Z. Wang, "Decentralized federated learning: A segmented gossip approach," *arXiv preprint arXiv:1908.07782*, 2019.

[5] L. Luo, C. Zhang, H. Yu, G. Sun, S. Luo, and S. Dustdar, "Communication-efficient federated learning with adaptive aggregation for heterogeneous client-edge-cloud network," *IEEE Transactions on Services Computing*, vol. 17, no. 6, pp. 3241–3255, 2024.

[6] S. Luo, X. Chen, Q. Wu, Z. Zhou, and S. Yu, "Hfel: Joint edge association and resource allocation for cost-efficient hierarchical federated edge learning," *IEEE Transactions on Wireless Communications*, vol. 19, no. 10, pp. 6535–6548, 2020.

[7] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 10, no. 2, pp. 1–19, 2019.

[8] R. F. Woolson, "Wilcoxon signed-rank test," *Wiley encyclopedia of clinical trials*, pp. 1–3, 2007.

[9] Citi Bike NYC, "Citi bike system data," https://citibikenyc.com/system-data, 2023, accessed: 2025-03-20.

[10] OpenWeather, "Openweathermap historical weather data," https://openweathermap.org/history, 2023, accessed: 2025-03-20.

[11] S. Patro and K. K. Sahu, "Normalization: A preprocessing stage," *arXiv preprint arXiv:1503.06462*, 2015.

[12] L. Yu, R. Zhou, R. Chen, and K. K. Lai, "Missing data preprocessing in credit classification: One-hot encoding or imputation?" *Emerging Markets Finance and Trade*, vol. 58, no. 2, pp. 472–482, 2022.

[13] M. H. Sazlı, "A brief review of feed-forward neural networks," *Communications Faculty of Sciences University of Ankara Series A2-A3 Physical Sciences and Engineering*, vol. 50, no. 01, 2006.

[14] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation," *Peerj computer science*, vol. 7, p. e623, 2021.

[15] C. Zhang, Y. Xie, H. Bai, B. Yu, W. Li, and Y. Gao, "A survey on federated learning," *Knowledge-Based Systems*, vol. 216, p. 106775, 2021.

[16] B. P. Welford, "Note on a method for calculating corrected sums of squares and products," *Technometrics*, vol. 4, no. 3, pp. 419–420, 1962.