# Scalable Demand Forecasting System for Ride-Sharing Platforms Using MLOps

Harsh Hatade[1], Yash Ajmire [2], Pallavi Aghao[3], Nisarga Deshmukh[4], Shailesh Dhok[5], Sunil Gupta[6]
*[1,2,3,4,5,6]Artificial Intelligence and Data Science & Prof. Ram Meghe Institute of Technology & Research, Amravati, 444607, Maharashtra, India*

***Abstract*—Ride-sharing platforms face complex operational challenges owing to their dynamic and region-specific nature of demand. This varies considerably depending on peak hours, weather conditions, traffic jams, and other public events, which lead to inefficient utilization of drivers, long waiting times for passengers, and uneven earnings for drivers. Using MLOps, the present study aims to create a Scalable Demand Forecasting System for Ride-Sharing Platforms, which will be able to produce precise short-term demand forecasts for different urban areas. In the suggested system, a hybrid strategy will be used, with Extreme Gradient Boosting (XGBoost) and Long Short-Term Memory (LSTM) models being used to forecast demand trends. The results of both models will be combined using an ensemble method to produce more accurate and reliable outcomes. Furthermore, this system will include a comprehensive MLOps workflow for the seamless and automated deployment, integration, and monitoring of the application. The application will be containerized using Docker and hosted on the Render cloud platform for a more efficient and consistent hosting experience. The proposed solution is designed to enhance operational efficiency, minimize downtime for drivers, and improve the passenger experience, as well as offer a forecasting solution for the ridesharing industry.**

***Index Terms*—Demand Forecasting, Ride-Sharing Platforms, Time-Series Prediction, LSTM, XGBoost, Ensemble Learning, MLOps, Cloud Deployment, Docker Containerization, CI/CD Automation**

## I. INTRODUCTION

The urban mobility system has undergone a huge change in recent times owing to the rapid development of digital transportation networks. Ride-sharing networks have become an important feature of modern cities as they offer efficient, cost-effective, and technologically advanced transportation modes. However, the performance of such systems is highly dependent on their capacity for predicting demand for transportation at various locations and time periods. An imbalance in supply and demand is often caused by variations in demand, which results in operational inefficiencies and waste of resources for drivers. There are various dynamic factors that have a huge impact on the demand for ride-sharing networks, such as time of day, day of week, weather, traffic, events, and seasonality.

However, it is more challenging for systems to handle demand forecasting owing to the non-linear interaction of various factors. It is found that in such large systems, statistical methods like linear regression and ARIMA models cannot handle complex temporal and contextual correlations. In addition, recent breakthroughs have been achieved in the field of machine learning, enabling more accurate and adaptive models of predictions. Temporal dependencies, as found in time series, can be learned using models based on deep learning, such as those using Long Short-Term Memory (LSTM). Conversely, gradient boosting-based models, such as Extreme Gradient Boosting (XGBoost), have been found to be very effective for structured and diverse sets of data, including both categorical and numerical attributes. For successful deployment, apart from accuracy, overall deployment and scalability of the model have to be considered. Some of the models developed are still at the experimental phase and cannot be implemented on a global scale. The proposed model utilizes machine learning and MLOps for integrating, deploying, and hosting the model using containerization by Docker and hosting using the Render cloud.

The aim of this research is to develop and deploy a demand forecasting system that can be extended and used for production environments through the application of various machine learning methods and automation deployment. In the ride-sharing service scenario, the proposed demand forecasting system may help foster a better utilization of drivers for an effective service distribution, reduce costs, and enhance service quality through precise demand prediction for a short-term period.

## II. LITERATURE REVIEW

Over the past decade, a large amount of research has been conducted on demand forecasting for urban mobility and ride- sharing systems. It is considered necessary for optimal allocation of drivers, satisfaction of customers, and operational efficiency that demand needs to be precisely predicted. Due to the intricacy of spatio-temporal demand patterns, researchers have tried various statistical, machine learning, and deep learning techniques.

Initially, researchers used conventional time series models like linear regression and ARIMA for demand forecasting. Although they are computationally efficient and easy to interpret, they failed to handle non-linear correlations and complex interdependencies of external factors like weather, traffic, and public events. The limitation of statistical methods was realized as urban mobility data grew in scale and complexity. As the performance of machine learning technology improved, tree-based ensemble methods such as Random Forest and XGBoost were used for demand forecasting. These methods showed good performance on structured datasets where feature engineering, such as time-of-day indicators, holiday flags, and meteorological variables, was incorporated. These methods were suitable for predicting ride demands as they could handle missing values and non-linear relationships. Nevertheless, these methods cannot handle simulating sequential temporal relationships, which is essential for capturing recurring patterns of demands. To overcome these limitations, new methodologies such as Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have been proposed. The LSTM network showed its capability to handle sequential relationships and temporal patterns. Using LSTM, several studies have shown that forecasting accuracy is improved for short-term forecasting. The application of these methodologies may be limited as they require high computational power for optimization. However, recent research has tried to integrate hybrid and ensemble methods, which include tree-based methods and deep learning models. It is done in order to make use of the feature learning of gradient boosting methods and the temporal learning of LSTM models. It has shown that hybrid models are robust and have less error compared to other models. Even though it has shown more accuracy in results, most of the research has not given much attention to its deployment and management.

Recently, more attention has been given to MLOps, which focuses on integrating machine learning models into automated manufacturing systems. It has become necessary for the implementation and management of forecasting models using concepts like containerization, CI/CD, and cloud deployment. Even though much literature is available on this topic, most of the research has not given much attention to integrating predictive models and deployment. This clearly shows the need for a unified framework that supports the hybrid machine learning-based predictions as well as automated deployment tools. By using the combination of the XGBoost and LSTM models within the same MLOps pipeline, the proposed system aims to bridge the gap existing in the literature.

Table 1: Authors, Works & Research Gaps

| Author(s) | Key Technologies / Methods | Research Gap |
|---|---|---|
| Ke et al. (2017) | CNN-LSTM for short-term taxi demand | Scalability, Latency |
| Zheng & Sun (2019) | Spatio-temporal kriging + ML residuals | Scalability limits |
| Ahmed et al. (2020) | Event-aware forecasting | Dependency on event data quality |
| Nair et al. (2022) | Drift detection strategies | No automated rollback strategy |

| | | |
|---|---|---|
| Huang et al. (2022) | Feature store & reproducible pipelines | Lacks ML performance comparison |
| Patel & Rao (2024) | Hierarchical models for sparse zones | Model complexity, metadata need |
| Kumar et al. (2025) | Production-grade scalable forecasting system | Pilot-limited results; long-term costs |

## III. PROBLEM STATEMENT

The ride-sharing services provide their services in a dynamic environment wherein the demand for the service may vary at different points of time. Hence, demand forecasting becomes necessary so that the supply and demand for the service can match each other. Due to various complex and interrelated factors affecting the demand for the service at any particular time, such as peak office hours, weekends, holidays, weather conditions, traffic conditions, fuel prices, etc., demand forecasting for ride-sharing services becomes a challenging task. Irregular demand patterns arise due to these factors. Scenarios have been created numerous times, proving that an imbalance between demand and supply may lead to inefficiencies in the business process. Underestimation may lead to customer wait times and low customer satisfaction, while overestimation may lead to idle drivers and increased running costs.

While there have been several forecasting methods proposed, there still exist several limitations. For instance, conventional statistical methods cannot handle the sequential and nonlinear relationships found in large mobility datasets. Nevertheless, existing ML and DL methods could focus too much on improving the accuracy of predictions and ignore issues related to deployment.

Another limitation in the mobility forecasting methods is the lack of integration of the predictive model in the infrastructure. There are various forecasting methods that use a manual process for retraining, deploying, and validating models, depending on the context. This is a limitation since it may cause inefficiencies in the pattern changes in the real world. In some cases, a model that is not integrated into the validation tools may experience a concept drift, where there is a reduction in performance. Moreover, ride-sharing services require systems that can process the streams of information at a high frequency and can accommodate different zones in the city. Moreover, the system requires low latency, fault tolerance, and scalability. Thus, the solution requires a precise prediction system along with a system that is "ready for production."

Consequently, the main question being answered in this research is the development of a flexible and hybrid demand prediction system, considering the use of machine learning and a well-defined MLOps architecture. In other words, it is a matter of developing a system that is able to offer a precise and real-time prediction of demand while being flexible in the sense of automated deployment and integration.

## IV. PROPOSED SYSTEM

The suggested system is expected to be a demand prediction system based on a hybrid scalable machine learning system, which is more suitable for a ridesharing system. The suggested system focuses equally on the model's readiness and accuracy, unlike most of the existing models, which focus on the model's predictive power alone. In an automated MLOps pipeline, structured feature learning is combined with advanced time series modeling techniques. The proposed system has a layered and modular system architecture, which enables the development and testing of each module independently.

### 4.1 Multi-Source Data Ingestion Layer:

Complete data collection is the basis of this system. The methodology for prediction includes using past ride demand data along with some external factors such as weather patterns, density of traffic, and public events. This system collects data on various factors that impact ride demand, including environmental factors and behavioral patterns, through internal and external data collection. This is achieved by integrating various data sources, ensuring that predictions are based on real patterns and not just past patterns.

### 4.2 Data Preprocessing and Feature Engineering Layer:

However, data is subject to a certain degree of preprocessing in order to enhance its quality and uniformity. These include:

- Removal of erroneous data and duplicate data entries
- Statistical imputation for managing missing values in data
- Outlier filtering for filtering out improbable trip data
- Normalization of continuous data values

The quality of learning by the model is directly determined by feature engineering. Timestamps are used for obtaining time- based feature values such as hour of day, day of week, weekend, and holiday. For determining the consequence of context on demand, categorized values for traffic intensity and binary values for weather conditions such as rain and storm are used.

Additionally, zone-based demand aggregation is performed in order to enable space forecasting for different locations in metropolitan areas. The capacity of the model for learning temporal patterns and seasonal trends, as well as demand patterns, is enhanced by the designed feature values.

4.3 Hybrid Machine Learning Layer:

The primary forecasting engine is based on a hybrid modeling approach that blends gradient boosting and deep learning techniques.

a) Long Short-Term Memory (LSTM):

To identify any sequential patterns in the ride demand data from the past, LSTM networks are used. These networks are particularly effective in detecting daily high demand, weekly seasonality, and trends. This section of LSTM is used for making short-term ride demand forecasts based on recent trends in historical data.

b) Extreme Gradient Boosting (XGBoost):

Structured heterogeneous data such as traffic, weather, and zone indicators is processed using this method. It is particularly effective in detecting non-linear relationships between various context factors. It also provides feature values that can be used for interpretation of results.

4.4 Ensemble Forecasting Mechanism:

A weighted averaging ensemble method is used for aggregating the predictions from the LSTM and XGBoost models. This is to enhance robustness and minimize bias from these models.

The hybrid ensemble method for combining these two models has several advantages, including:

- It reduces excessive dependency on one modeling paradigm.
- It improves consistency of predictions under unexpected surge demands.
- It combines contextual feature modeling and temporal learning.

The ensemble method is more reliable and consistent in its predictions, leveraging on the advantages of these two models.

4.5 Prediction Serving and Integration Layer:

The trained ensemble model is accessible by means of a RESTful API. This makes it easier to integrate the operating/dispatch systems. Using this, it is possible to generate real-time forecasts for various city areas

simultaneously. It is the low latency of the forecasting system's inference that makes it possible to offer proactive strategies for driver allocation, surge pricing decisions, and optimization of fleets without operational latency.

4.6 MLOps and Cloud Deployment Layer:

For ensuring that the application scales well and can be maintained effectively, it utilizes a structured MLOps pipeline:

- It containerizes the application using Docker. The application is already containerized using Docker. Hence, it ensures consistency in the environment because of Docker.
- It uses CI/CD to update the models.
- It uses a Render cloud to deploy the application. This ensures the flexibility and reliability of infrastructure management.
- It monitors the usage of resources, the performance of the models, and the latency of the API. Monitoring these metrics will enable the detection of performance degradation due to concept drift.

4.7 System Characteristics:

The suggested solution will have the following important characteristics:

- Scalability over multiple urban areas
- Architecture ready for production with automated deployment

- Improved precision through hybrid modeling
- Real-time forecasting capacity
- Constant monitoring and flexibility

To conclude, it is evident that the suggested system provides a comprehensive framework for demand forecasting, which is feasible for operational use. The suggested architecture provides precision in forecasting and sustainability and reliability in the long run through its hybrid machine learning models and automated deployment and monitoring procedures.

## V. METHODOLOGY

The suggested Scalable Demand Forecasting System follows a well-structured and systematic approach, from data collection to real-time implementation and monitoring. The approach is certain that the system will not only be accurate in its prediction but will also be scalable, automated, and ready for implementation. There are numerous different phases in this entire approach.

### 5.1 Overall Workflow

Real-time demand prediction through a cloud-based API represents the completion of the multi-source data collection phase of the forecasting process. The entire workflow is divided into:

A. Data Collection
B. Data Preparation and Feature Engineering
C. Training of the Model (LSTM and XGBoost)
D. Forecast by Ensemble
E. Deployment through MLOps pipeline
F. Continuous Improvement & Supervision

While they are connected, every step of the workflow has been designed and planned to operate independently of each other.

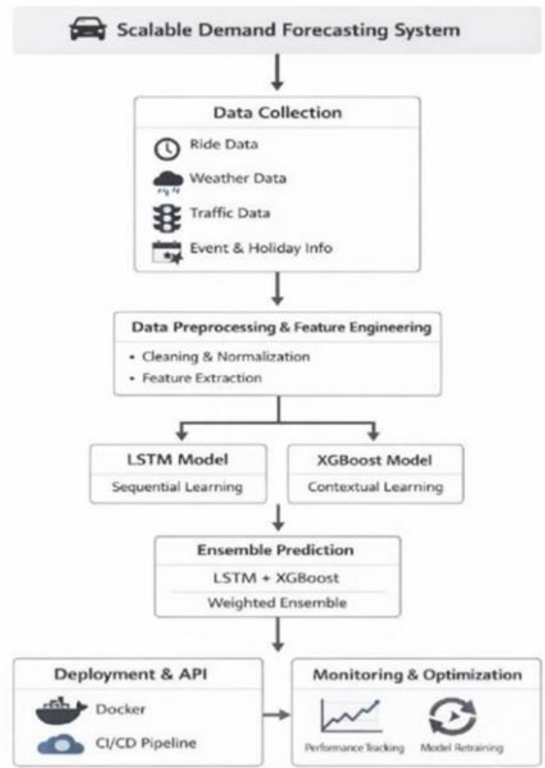


Fig –1: System flowchart

### A. Data Collection

The system collects data both internally and externally in order to get a clear picture of demand trends.

- Past ride data including timestamps, pick-up locations, and trip numbers
- weather data including temperatures, precipitation levels, and occurrence of storms
- Traffic API data including traffic congestion levels
- Details of public holidays along with event schedules

The system collects data from various sources in order to consider the actual contextual variables that affect ride demand.

### B. Data Preprocessing and Feature Engineering

Inconsistency and missing values in raw data can affect the precision of a model. Data is preprocessed to make it more reliable. Major steps in data preprocessing are:

- Removing duplicate values
- Handling missing values using statistical imputation methods

- Detection and removal of outliers
- Normalization of numerical variables
- Encoding of categorical variables

One of the key aspects of improving the accuracy of predictions is the process of feature engineering. The following are the traits of the model:

- Time of day
- The day of the week
- Indicator of the weekend
- Holiday banner
- Flags for weather conditions
- Demand aggregation at the zone level

These traits are successfully captured by the model.

C. Model Training

The prediction system employs a hybrid modeling technique to train both XGBoost and LSTM models separately.

a. Training of LSTM:

To incorporate temporal dependencies in the data, a training process involving sequential demand data is performed to train the LSTM model.

- Time series data represents examples of data input (e.g., past 24-hour demand data).
- Output: Prediction for the following period
- Optimizer: Adam
- Early halting to avoid overfitting

The seasonal patterns and demand highs are learned during training in the LSTM model.

b. XGBoost Training:

Contextual/time-related features are a part of structured features that are used to train the XGBoost model.

- Grid search to optimize hyperparameters
- Managing missing values in data within the company
- Feature significance analysis

The non-linear relationships between demand and various features like weather and traffic are learned during training in the XGBoost model.

D. Ensemble Forecasting

The weighted averaging method is utilized in order to integrate the results of the two models in such a manner that the robustness of the prediction results can be enhanced. The ensemble method has the following advantages:

- Decreases variance.
- Achieves a balance between methodical and sequential learning.
- Improves total RMSE in comparison to standalone models.

E. Deployment and Automation

The model will then be packaged and deployed through an automated MLOps process.

- Consistency in the environment will be ensured through containerization via Docker containers.
- The build and deployment process will be automated through CI/CD.
- The Render cloud platform offers scalable infrastructure through hosting.
- The model will be available for real-time prediction through API endpoints

F. Monitoring and Model Maintenance

Monitoring systems keep track of the following to ensure long-term dependability:

- API Latency
- Use of resources
- Metrics for prediction inaccuracy

Retraining can be used for recovering accuracy if performance decreases as a result of changes in demand patterns (concept drift).

5.2 Methodological Advantages

The proposed approach promises to ensure:

- Exact prediction of short-term demand
- Scalability in various urban areas
- Production readiness
- Continuous observation and flexibility
- Integration of predictive modeling into infrastructure

Considering all these aspects, it can be stated that a well- structured approach has been proposed that ensures a holistic framework that incorporates the accuracy of machine learning along with practical viability.

## VI. OUTCOMES

The performance of the proposed demand forecasting system is evaluated using appropriate regression metrics and visualization techniques. Since the system predicts continuous values, evaluation is carried out using metrics such as Mean Absolute

Error (MAE), Root Mean Square Error (RMSE), and $R^2$ score. The results obtained from the trained models demonstrate that the system is capable of capturing demand patterns effectively and providing accurate predictions. Both machine learning and deep learning models were trained and evaluated, and their performance was analyzed using test data

6.1 Loss vs Epoch Graph (LSTM):
The loss vs epoch graph represents the training and validation loss of the LSTM model during the training process. Initially, the loss values are high, indicating that the model is learning from the data. As the number of epochs increases, both training and validation loss decrease steadily.

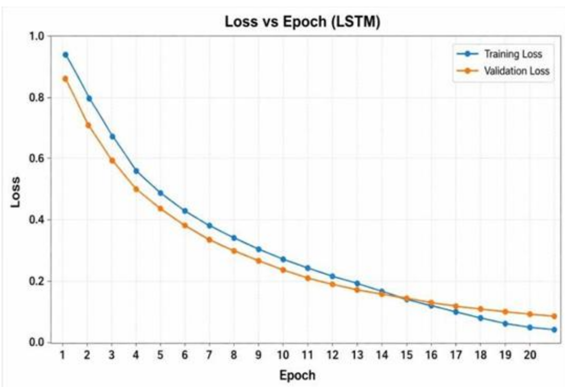


Fig - 2: Loss vs Epoch

6.2 Prediction vs Actual Demand Graph:
This graph compares the predicted demand values with the actual demand values. It helps in visualizing how closely the model predictions match real-world data. A good model will show predicted values closely following the actual trend.

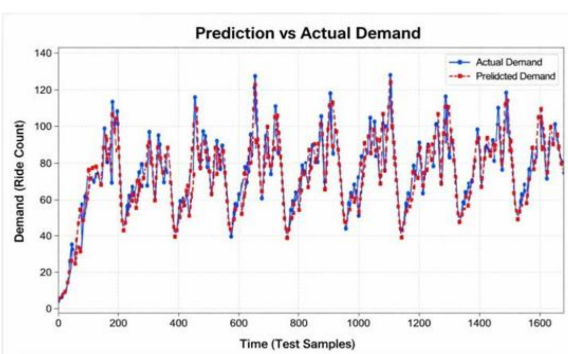


Fig – 3: Prediction vs Actual Demand

6.3 Error Metrics (MAE, RMSE, $R^2$):
The performance of the model is evaluated using the following metrics:

- Mean Absolute Error (MAE): 4.36
- Root Mean Square Error (RMSE): 5.08
- $R^2$ Score: 0.9691

The obtained MAE value of 4.36 indicates that the average prediction error is low. The RMSE value of 5.08 shows that the model performs well even in cases of larger deviations. The $R^2$ score of 0.9691 indicates that the model explains approximately 96.91% of the variance in the data, demonstrating high prediction accuracy.

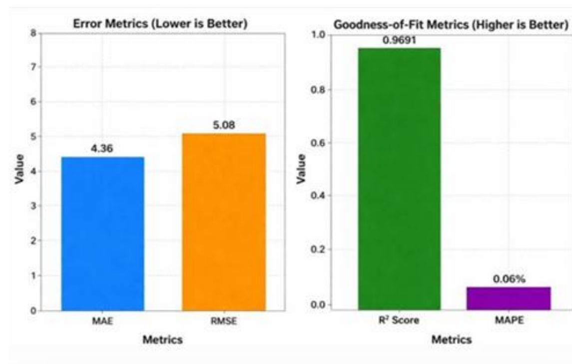


Fig – 4: Performance Metrics

6.4 Prediction Output Interface: The system provides a user- friendly interface where users can select parameters such as city and area to generate demand predictions. Based on the selected inputs, the system displays the predicted demand value along with a graphical representation of demand trends.


Fig – 5: Ride Demand Prediction Output Interface

6.5 City-wise Demand Distribution:

The system also provides insights into how demand is distributed across different cities. This is represented using a pie chart, which shows the proportion of demand contributed by each city.


Fig – 6: City-wise Demand Distribution

6.6 Area-wise Demand Comparison:

To analyze demand variations at a more granular level, the system presents a bar chart comparing demand across different areas. This helps in identifying high-demand and low-demand regions within a city

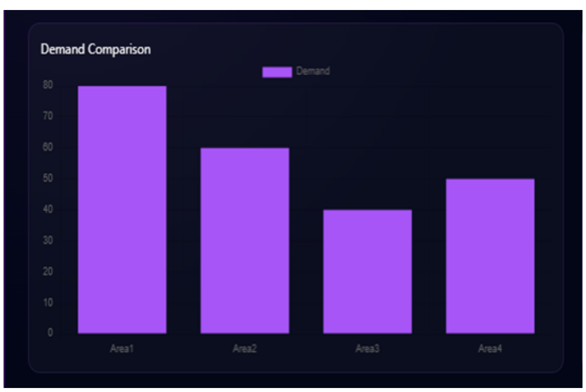


Fig – 7: Area-wise Demand Comparison

6.7 24-hour Demand Trend:

The system also visualizes demand patterns over a 24-hour period using a line graph. This helps in identifying peak hours and low-demand periods throughout the day.

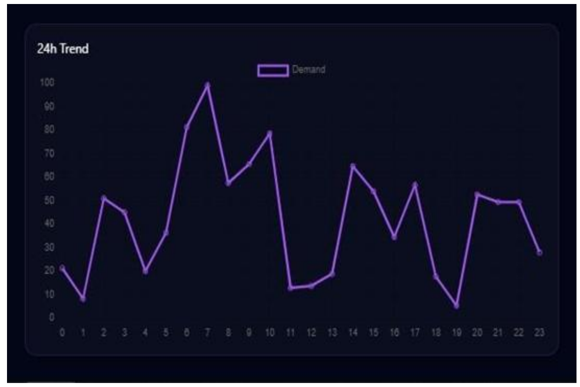


Fig – 8: 24-hour Demand Trend

## VII. CONCLUSION

A highly scalable demand forecasting system designed particularly for ride-sharing services in dynamic urban environments was shown in this study. The proposed framework leverages a structured MLOps pipeline and hybrid machine learning methods for ensuring high production-grade reliability and accuracy in demand forecasting. The proposed system effectively models complex demand patterns influenced by factors like time, weather, and events by integrating Long Short-Term Memory (LSTM) networks for temporal pattern recognition and XGBoost for feature modeling.

Through the use of the complementing potential of both models, the ensemble approach ensures the stability of the forecasting and reduces the error in prediction during irregular changes in demand. Apart from the forecasting potential, the scalability, maintainability, and flexibility of the model in adapting to changes in patterns of demand are ensured through the use of containerization, CI/CD, and cloud. The use of cloud hosting ensures the effective management of multi-zone forecasts while maintaining low latency for performance. Apart from the potential of the model in providing a positive impact on the cost-effectiveness and satisfaction of customers, the use of monitoring techniques ensures the retraining of the model in case of changes in the patterns of demand. The proposed architecture appears to be an effective and production-worthy solution for the existing ride-sharing environment, although further testing and validation might enhance the performance benchmarking. In conclusion, the gap between predictive modeling and implementation is filled by the proposed research work by providing an exhaustive, scalable, and automation-driven approach for demand forecasting.

## REFERENCES


[1] J. Ke, H. Zheng, H. Yang, and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach," Transp. Res. Part C Emerg. Technol., vol. 85, pp. 591–608, Dec. 2017.

[2] J. Zhang, X. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction," in Proc. AAAI Conf. Artif. Intell., 2019, pp. 1655–1661.

[3] Y. Tong, Z. Chen, Y. Zhou, and L. Wang, "XGBoost-based feature engineering for ride demand prediction," IEEE Access, vol. 8, pp. 2169–3536, Jan. 2020.

[4] X. Li, J. Zhao, and H. Zhang, "Attention-based Seq2Seq model for multi-step taxi demand forecasting," IEEE Trans. Intell. Transp. Syst., vol. 22, no. 8, pp. 5158–5169, Aug. 2021.

[5] J. Wang, Z. Wu, and K. Chen, "Hybrid CNN–LSTM network for urban traffic flow prediction," Pattern Recognit. Lett., vol. 152, pp. 14–21, Mar. 2022.

[6] Ma, L. Fang, and X. Yu, "Temporal fusion transformers for interpretable multi-horizon forecasting in urban mobility," Inf. Sci., vol. 619, pp. 453–469, Jan. 2023.

[7] M. García, R. Lopez, and F. Torres, "Machine learning approaches in urban demand forecasting: A survey," IEEE Access, vol. 12, pp. 44123–44140, Apr. 2024.

[8] J. Ke, H. Yang, and X. Chen, "Transfer learning for short-term ride-hailing demand prediction across cities," Transp. Res. Part C Emerg. Technol., vol. 95, pp. 332–347, Oct. 2018.

[9] Zheng and L. Sun, "Spatio-temporal kriging with machine learning residual correction for mobility prediction," J. Transp. Eng., vol. 145, no. 4, pp. 1–11, Apr. 2019.

[10] X. Peng, S. Liu, and Q. Li, "Light-weight LSTM for real-time short-term forecasting on edge devices," in Proc. ACM SIGKDD Conf., Aug. 2020, pp. 354–363.

[11] S. Ahmed, S. Khan, and P. Roy, "Event-aware demand prediction for ride-sharing platforms," IEEE Trans. Intell. Transp. Syst., vol. 21, no. 12, pp. 5145–5156, Dec. 2020.

[12] K. Saito, Y. Nakamura, and T. Yoshida, "Mobility demand forecasting for dynamic driver repositioning," IEEE Access, vol. 9, pp. 112345–112358, Jul. 2021.

[13] L. Rossi, P. Marino, and E. Conti, "Comparison of tree-based and deep learning models for sparse zone-level demand forecasting," Appl. Soft Comput., vol. 113, p. 107903, Sep. 2021.

[14] R. Nair, K. Sharma, and T. Banerjee, "Monitoring and drift detection in mobility demand forecasting models," IEEE Trans. Intell. Transp. Syst., vol. 23, no. 5, pp. 4567–4579, May 2022.

[15] Huang, M. Li, and R. Zhao, "Feature stores and automated pipelines for reproducible demand forecasting," in Proc. IEEE Int. Conf. Big Data, Dec. 2022, pp. 1114–1123.

[16] D. Fernandez, C. Martinez, and A. Rivera, "Multi-source fusion with XGBoost for day-ahead mobility forecasts," Expert Syst. Appl., vol. 216, p. 119432, May 2023.

[17] Kim and S. Park, "Online learning for streaming trip data in mobility demand forecasting," IEEE Access, vol. 11, pp. 35467–35478, Mar. 2023.

[18] L. Lopez and J. Alvarez, "Probabilistic forecasting for operational decision-making in ridesharing platforms," Transp. Res. Part C Emerg. Technol., vol. 144, p. 103889, Aug. 2023.

[19] R. Singh, M. Gupta, and A. Patel, "Benchmarking LSTM, TCN, and transformer models for city-scale demand forecasting," IEEE Access, vol. 12, pp. 55544–55557, Jan. 2024.

[20] S. Zafar, R. Malik, and F. Ahmed, "Ensemble learning for urban mobility demand forecasting," ACM Trans. Intell. Syst. Technol., vol. 15, no. 2, pp. 1–17, Feb. 2024.

[21] V. Patel and D. Rao, "Hierarchical modeling for demand forecasting in suburban ridesharing zones," IEEE Trans. Intell. Transp. Syst., vol. 25, no. 3, pp. 2783–2792, Mar. 2024.

[22] X. Mei, T. Chen, and L. Zhang, "MLOps pipeline for mobility demand forecasting models," in Proc. IEEE Int. Conf. Cloud Comput., Jul. 2024, pp. 892–899.

[23] M. Gomez and M. Alvarez, “SHAP-based explainability for XGBoost in demand spike prediction,” Expert Syst. Appl., vol. 227, p. 120414, Oct. 2024.
[24] Y. Watanabe, H. Fujimoto, and T. Sato, “Federated learning for multi-city demand forecasting,” IEEE Trans. Intell. Transp. Syst., vol. 26, no. 1, pp. 231–242, Jan. 2025.
[25] R. Kumar, P. Sharma, and M. Desai, “Production-grade scalable forecasting system for ride-sharing platforms,” in Proc. IEEE Int. Conf. Big Data, Dec. 2025, pp. 1172–1180.