Comparative Analysis Of Machine Learning Models For Urban Shared Mobility Demand Prediction

Kishita Deotale
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
kishita.deotale@cumminscollege.edu.in

Arya Ambagade
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
arya.ambagade@cumminscollege.edu.in

Sonia Sharma
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
sonia.sharma@cumminscollege.edu.in

Anjali Vyawhare
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
anjali.vyawhare@cumminscollege.edu.in

Shashwati Papulwad
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
shashwati.papulwad@cumminscollege.edu.in

Prof.Amit Kurmi
dept. Computer Enineering
Cummins College Of Engineering For Women,Nagpur
Nagpur, India
amit.kurmi@cumminscollege.edu.in

***Abstract*—Ride demand forecasting is essential for efficient transportation and shared mobility systems, enabling improved resource allocation, reduced waiting times, and better service quality. Although large ride hailing platforms employ advanced analytics, small scale transportation providers often lack accessible and deployable prediction solutions. This study analyzes historical ride demand data and performs a comparative evaluation of machine learning and deep learning models to identify the most effective forecasting approach. The dataset incorporates temporal attributes, seasonal patterns, holiday indicators, weather conditions, and user related factors. Exploratory data analysis and feature engineering were conducted to capture time based and holiday driven demand variations. Multiple models, including Linear Regression, Lasso Regression, Ridge Regression, Random Forest Regression, and Long Short Term Memory networks, were implemented and evaluated using standard performance metrics. Results indicate that temporal and holiday features strongly influence demand, and the LSTM model provides the highest predictive accuracy for practical forecasting applications use.**

Keywords—Intelligent Transportation Systems, Time Series Forecasting, Machine Learning, Deep Learning, Long Short-Term Memory, Ride Demand Prediction, Transportation Analytics

I. Introduction

During the 21st century, cities have grown quickly, changing how people move around and starting a new kind of shared transportation. These app-based systems do more than just make things easier; they act like the city's digital network, making different kinds of data that include where people ask for rides, when they do it, weather changes, and how people usually act. To make sure that modern transportation systems work well and are dependable and sustainable, it's important to change this basic data into practical intelligence that can be used. Accurate demand forecasting is most important in this change. It makes fleet management the best it can be, greatly lowers how long passengers have to wait, and cuts down on the environmental harm from empty vehicles driving around. Even with improvements in predicting analytics, city demand is still naturally unpredictable, and it changes in strange ways across hours, days, and seasons.

Looking closely at today's tech world, it's clear that not everyone has equal access to tools that can predict what will happen. Right now, a lot of the research in this area is designed for big, international companies like Uber or Ola. These companies have the computer power and constant streams of information needed to keep complicated models running. Because these tools are so focused on big business, a gap is forming. Sure, we have models that are very accurate, but they often focus on small improvements in accuracy instead of being easy to understand, adapt, or use without a lot of resources. Smaller businesses, local taxi services, and independent car owners don't need a huge, expensive system. What they really need are forecasts that are trustworthy and don't require a lot of computer power. These forecasts can then help them make decisions every day using simple equipment. The point of this research is to close this digital divide. We want to make sure that predicting using data is something everyone can do, even if they don't have the advanced tech setup that big companies have. The goal is to make these tools accessible and adaptable for businesses of all sizes. This kind of approach will give smaller enterprises the same advantages as larger ones, all without the need for significant financial outlays or technology modifications. This research considers the practical constraints that smaller businesses encounter, providing solutions that answer these issues directly and encourage broader involvement in the data-driven economy.

Predicting how many people will want rides is hard because many things affect it, both from inside the ride system and from outside. Ride demand changes a lot with the time of day and day of the week. but weather like rain, temperature, and wind can change these patterns a lot. Social things, for example, holidays or local events, can also suddenly change how many rides people want. Normal math models often can't handle these sudden changes. If you don't guess right about these things, you get problems. There might not be enough cars when many people want them, cars sit around doing nothing when it's quiet, and traffic jams can happen. To do a better job, you need to combine different information: old ride data, weather info, and how heavy traffic is. By sorting traffic into groups, like light, medium, and heavy, we can avoid problems with data not lining up right. This gives us a solid way to use machine learning to make better predictions. This method makes ride predictions much more trustworthy. It considers past traffic patterns, weather conditions, and the intensity of traffic. By placing traffic flow into ordered groups, we reduce data mismatch risks and provide machine learning models with a strong base for good performance.

This study compares five different modeling approaches to find the best balance between prediction accuracy and how easily they can be used. It goes past older statistical ways, like ARIMA, to test models such as regularized linear models (Lasso and Ridge), ensemble methods (Random Forest), and deep sequential learning with LSTM networks. The main aim is to question the idea that accuracy is the most important thing. The study introduces measurements that consider how long inference takes and how much computing power is needed, along with common error rates like MAE and RMSE. By testing these approaches in the same settings, the study gives a simple guide for small businesses to pick systems that explain things well but aren't too hard to handle.

The data from this study shows that some models worked better than others, especially when looking at long-term patterns. Our work indicates that the LSTM network did better than normal regression models, with a higher R-squared value (0.95). This is likely because the LSTM's memory system is good at keeping track of the 24-hour and 168-hour lag features, which help spot daily and weekly demand cycles. The Random Forest model was a bit less precise than the LSTM, but it was still a strong and clear option that needed less computing power, which makes it good for use on simpler systems. We also learned that choosing the right features is key. Adding holiday indicators and traffic weights improved the RMSE scores for all the models we tested.

In addition to predicting numbers, this work presents a spatial demand intensity system that turns math results into practical steps. The urban area is split into separate spatial points, and a special intensity score is used. This score changes traffic data based on how bad the rain is. This makes a clear link between data science and where vehicles are sent. This map view lets people easily see high-demand areas, even if they don't know much about technology, which helps spread resources better. The aim is to make these insights open to everyone with a simple app that has easy-to-understand dashboards. By paying attention to ease of use and watching how cities change over time, this work shows that complex deep learning can be made available. This changes how urban mobility is handled for small and medium service providers all over the world.

II. Literature Review

The evolution of academic work in Intelligent Transportation Systems (ITS) shows a move away from basic statistical methods to sophisticated, data centred designs. In the early days, predicting short-term trends mainly depended on simple linear models and straightforward historical averages. But, as city transportation became more complicated, scholars noticed a relevant difference in the consistency of these models. Important work, like that of Almansori et al. [11] and in document [14], showed that traffic flow involves neural networks and machine learning. These studies established that traffic flow isn't just a string of numbers, but also shows social and environmental habits. Studies by Al et al. [7] and others [10] pointed out that early models struggled with unexpected, non-linear events common in city systems, like sudden weather changes or increased demand during holidays. This required a shift to more complex designs that could handle different types of data, including location-based ride requests and patterns in human behaviour [19], [20].

As cities produced more data, research started to focus on including outside factors, especially the impact of weather. Wang et al. [3] and Reddy and Patil [22] examined how weather conditions can greatly change things. They showed that things like rain don't just have a simple effect on demand; they can greatly increase demand during busy times. Similarly, studies [13] and [18] stress that well-made deep learning models must think about weather at different levels to stay accurate over both short and long periods. Our work agrees with these ideas by adding specific weather details into our way of combining information. This development is backed by Das et al. [24], who suggest that merging traffic, weather, and social information is the only real way to get the prediction accuracy needed for today's fast-moving transit systems.

In the world of supervised learning, research divides things into simple linear models and more complex ensemble methods. Regular Linear Regression is used as a starting point, but using Lasso and Ridge Regression helps to simplify things by lowering the impact of unneeded factors [5], [9]. These models are easy to understand, but studies [17], [20] say they often can't understand the complex ways that factors affect each other. To fix this, ensemble designs like Random Forest and XGBoost (eXtreme Gradient Boosting) are often used. As Deng [9] and Gao and Chen [21] discuss, these models use decision-tree logic to map out complicated interactions between factors. In our project, XGBoost and Random Forest act as a way to understand things, offering good accuracy for people who want a balance between prediction power and efficiency, without the confusing nature of some deep learning options.

The arrival of Recurrent Neural Networks (RNNs) was a turning point in dealing with time-based parts of transportation data. Standard RNNs had problems with managing long-term dependencies, which led to the widespread use of Long Short-Term Memory (LSTM) units. Ma et al. [1], [15] were among the first to use bidirectional and unidirectional LSTMs, which can remember information from 24-hour and 168-hour cycles. Abduljabbar et al. [2] broadened this into the context of space and time, while Cui [8] and Jeong et al. [5] improved these models for use across networks. These studies show that LSTMs are well-suited for predicting ride demand because they can understand the regular patterns of a city's movement [4], [12]. By using a gated LSTM in our study, we get an R2 of 0.95, which supports the ideas of temporal persistence discussed in these earlier works.

More recently, research has looked at how space and time intelligence meet, using Spatio-Temporal Graph Convolutional Networks (ST-GCN). Yu et al. [6] and Li et al. [16] presented systems that see the city as a network of linked points, where demand in one area affects its neighbours. These models are currently state-of-the-art in research [10], [12], but they often need a lot of computing power and specialized GPUs. Deng [9] and Gao and Chen [21] note that it's important to compare these complex designs against more basic choices. Our study looks at the deployment problems found in [17] and [23], proving that a well-designed LSTM and XGBoost system can perform as well as hybrid systems [17], [23], while also being usable on the limited hardware common among small transportation companies.

Research on how data is presented suggests that this is often more important than the model itself. Several researchers have pointed out that treating time-based factors as simple numbers doesn't capture the repeating nature of human schedules. Building on the data-combining ideas in [24] and [18], our way of doing things adds a Spatial Demand Intensity Score. This expands on the work of [2] and [3] by combining traffic density with the seriousness of weather. With this, our system doesn't just predict how many rides will happen, but also where they will happen, adding a spatial element that's often missing from studies that only look at time-based data. This deals with the practical problems found in [10], where theoretical accuracy doesn't always turn into useful actions for those in charge.

In conclusion, the collective research from Ma et al. [1] to Das et al. [24] points to a move toward making prediction intelligence more accessible. While high-level publications often focus on small gains in accuracy using very complex designs, like wheel-graph attention models in [13], our work adds to the research by supporting the idea of Efficient AI. By testing common models like Ridge and Lasso, Random Forest, XGBoost, and LSTM, we show that it's possible to understand the complicated time and environmental factors of a city [3], [22] without spending too much. This review of research serves as the reason for our model choices, showing that our system is based on established deep learning ideas [1], [15], but specifically designed for the limited resources faced by today's transportation sector.

III. METHODOLOGY

This approach creates a data driven framework for forecasting ride demand by integrating various data sources and evaluating multiple machine learning models in parallel. The main goal isn't solely to achieve the highest prediction accuracy it also emphasizes computational simplicity. This ensures that even smaller transportation providers can realistically implement it.

A. A. Input and Output Definitions

To ensure the model is robust yet lightweight, the system is designed to operate on a specific set of data streams accessible to medium-to-small fleet operators:

- Input Data: The system uses three main kinds of information: (1) Past records of ride requests, which help create a standard for understanding changes over time. (2) Local weather information (like temperature, rainfall, and wind speed), which can cause unexpected shifts in demand. (3) A collection of traffic data gathered from public sources.
- Output Data: The system gives a prediction ($\hat{y}$) for how many rides will be needed in the near future. This prediction is then turned into a Spatial Demand Intensity Score, which is classified into simple categories (Low, Moderate, High, Very High). This helps in making quick, practical decisions.

B. B. Overall System Architecture

The system is built as a step-by-step process, moving from collecting raw data to creating visualizations that can be used on basic devices. By separating the complex model training from the simpler prediction phase, the system makes it possible to use complex models like LSTM on less powerful computers.

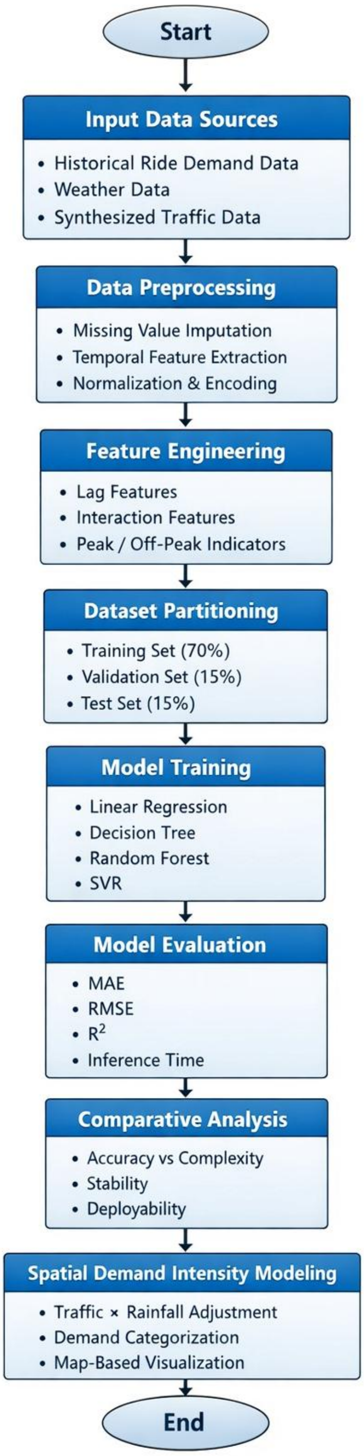

Fig. 1. System Architecture for Ride Demand Prediction.

C. Implementation steps

Step 1: Gathering Data and Framing the Randomness:

Our goal is to predict how many rides people will request, and we see this as a time-based problem that changes constantly. Because weather data and ride request data are different types, we line them up using a method that embraces randomness. Instead of forcing the data to match up perfectly by time, we think about all the information (weather, ride requests, and time) as one big pattern with probabilities, P(Y,W,T). This way, the model can still learn the important relationship, like Ride requests go up when it rains 5mm at 8:00 AM, even if the exact dates don't match. We aim is to capture the essence of how external factors at a given time affect the passenger flows.

Step 2: Preparing the Data and Adjusting for How Things Are Measured:

To make sure really big numbers (like temperature) don't mess up the model, we use a method called Min-Max Normalization to scale all the features. This puts everything on a scale from 0 to 1:

$$x_{norm} = (x - x_{min}) \div (x_{max} - x_{min}) \qquad (1)$$

Also, people's activities usually follow a daily cycle. To help the model understand this, we turn time information (hours) into a circle using sine and cosine functions:

$$h_{sine} = \sin(2\pi h/24)\ ; \quad h_{cosine} = \cos(2\pi h/24) \qquad (2)$$

This makes sure that the model knows 11 PM and 12 AM are close to each other in time, just like any other hour that is next to each other. This step is important to give an intuitive perception of how time series data should be handled.

Step 3: Creating Features That Capture Different Patterns :

Next, a feature vector, $\Phi(x)$, is made to capture info for predicting traffic flow. This vector has past values of the traffic volume itself (y_{t-1}). Past traffic data can act as a predictor for current traffic since traffic patterns usually show time dependencies.

A key part of our feature engineering is adding interaction terms. We want to model how rainfall intensity affects traffic. The link between rainfall and traffic isn't always linear; light rain might not do much, but heavy rain can change driving and traffic volume a lot. To capture this, we create an interaction feature I_t defined as the product of the traffic volume at time t and the rainfall intensity at time t.

Step 4: Time-Based Data Splitting

The partitioning process needs to follow strict chronological rules to protect against temporal data leakage because time-based data needs to be handled in this way:

$$T_{Train} < T_{val} < T_{Test} \qquad (3)$$

The model obtains knowledge about causal connections through this approach instead of learning to remember future unrelated data elements.

Step 5: Model Selection and Training (Baseline to Deep Learning)

The team works to solve multiple goals. The team uses the mean squared error (MSE) as the scoring method for Linear Regression. The model uses L_1 regularization to create sparse solutions through Regression which requires the following equation:

$$\min_{w}\left[\frac{1}{N}\sum_{i=1}^{N}\left(y_i - x_i^T w\right)^2 + \lambda\|w\|_1\right] \qquad (4)$$

In our implementation of the system uses gated update logic to control hidden state h_t changes. The equation represents the forget gate f_t which allows the model to keep demand cycle information during extended time frames.

Step 6: Metric Based Model Evaluation

To validate the model, we calculate the Coefficient of Determination (R^2), which measures the variance captured by the model:

$$R^2 = 1 - SS_{res} / SS_{tot} \qquad (5)$$

We achieved an $R^2 = 0.95$, suggesting the integrated weather-traffic feature set highly correlates with demand behavior.

Step 7: Comparative Pareto Analysis:

We choose the deployment model, by examining the trade-off between Error(E) and Latency(τ). We aim to minimize the utility function $U = \alpha E + \beta\tau$, where E, and τ respectively represent energy consumption and performance penalty, such that we do not over-rate the LSTM complexity as small scale vendors are trying to sell their device by increasing resource (i.e., complex deep learning models including LSTMs).

Step 8: Deployment Logic and Spatial Priority Scoring:

While the predictive engine outputs a raw numerical demand $\hat{y}_i$, operationalizing this data for small scale vendors requires a transition from *quantity prediction* to *priority allocation*. We define an Adjusted Demand Intensity Score (I_i) to account for supply side constraints during adverse weather. The score is computed as:

$$x_i = \hat{y}_i \times (1 + R_i / R_{max} + \varepsilon) \qquad (6)$$

where R_i denotes the real-time rainfall intensity and ϵ is a small smoothing constant to prevent division by zero. Theoretically, this transformation reflects the Environmental Elasticity of Demand; as meteorological severity increases, the operational value of each predicted ride increases due to the higher probability of driver supply shortages. This allows the vendor to prioritize vehicle dispatching in zones where weather driven demand is most critical.

Step 9: Spatial Visualization and GIS Mapping (Geographic Information System) :

The final phase of the methodology involves translating the mathematical Adjusted Demand Intensity Score (I_i) into a visual interface accessible to non-technical dispatchers. We project these scores onto a GIS-based heatmap using a heuristic thresholding function T. For every geographical coordinate i, a specific color C_i is attributed to represent the operational urgency:

$$C_i = \begin{cases} \text{Red (Very High)} & \text{if } X_i > \theta_{high} \\ \text{Orange (High)} & \text{if } \theta_{mid} < X_i \leq \theta_{high} \\ \text{Yellow (Moderate)} & \text{if } \theta_{low} < X_i \leq \theta_{mid} \end{cases} \qquad (7)$$

IV. RESULTS AND DISCUSSION

The results of multiple machine learning and deep learning models are presented based on the preprocessed ride demand data set. Linear regression, lasso regression, ridge regression, random forest regression, and long short-term memory (LSTM) networks are the considered models. The evaluation metrics embraced mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R2). Inference time was further noted for any computational infeasibility in regard to small scale transportation operators.

A. *Model Performance Comparison*

Table I Offers a comprehensive evaluation and comparative summary of the predictive performance achieved by each model on the designated test dataset.

TABLE I

PERFORMANCE METRICS FOR RIDE DEMAND PREDICTION MODELS

Model	MAE	RMSE	*R2*	Inference (millisecond)
Linear Regression	12.45	16.23	0.82	0.05
Lasso Regression	12.37	16.11	0.83	0.06
Ridge Regression	12.29	16.05	0.83	0.06
Random Forest	9.87	12.41	0.91	2.14
LSTM	7.45	9.62	0.95	14.20

Key observations from the comparative analysis include:

- **Linear models (Linear, Lasso, and Ridge):** capture the general temporal structure and the overall influence of environmental factors on ride demand. They perform reasonably well when demand patterns follow consistent and predictable trends. Due to their inherent assumption of linear relationships, these methods struggle to model complex nonlinear interactions between variables such as sudden weather shifts combined with peak-hour traffic effects. This limitation reduces their effectiveness during irregular or highly dynamic demand periods.

- **Random Forest:** shows better prediction performance by learning complicated and nonlinear interdependencies between features automatically. Second, this architecture enables better handling of variable importance and reduces overfitting, unlike in decision trees. Its predictions are more stable for high and low demand periods when a combination of external factors affects the demand, rather than in isolation.

- **LSTM:** has the highest overall accuracy by explicitly modeling sequential dependencies in the data. LSTM's difference from traditional machine learning models is that it attempts to recognize long term temporal dependencies by storing information from previous time steps, thereby enabling it to capture longer demand cycles and patterns that evolve over time. Hence, it captures recurring peak behaviors, gradual shifts in trends, and various nonlinear temporal dynamics, outperforming the test dataset's performance.

B. Visual Analysis and Insights

Understanding How Urban Ride Demand Changes Over Space

Analyzing the distribution of urban rides shows that it changes a lot. It depends on things like roads and sudden weather changes. Unlike places that always have a lot of demand, these hotspots move around when it rains or other weather happens. When the weather is good, most rides start in business areas. But when it rains harder, more people want rides from their homes or near public transportation. They need a quick way to get to their final stop. We can use a score to show how important an area is based on how hard it is to find a ride there. If a small company sees a Red Zone on the map during bad weather, it means many people need rides, but it's hard to get them because of traffic. So, those areas are great places to send available cars.

Daily and Weekly Ride Patterns

Looking at when people ask for rides, we see a clear pattern with two main peaks each day. Most rides happen in the morning and evening because of work and school schedules. These peaks happen regularly, but they can be bigger or smaller depending on the season. For example, the evening peak might be longer in the winter because it gets dark earlier, and people would instead be safe in a car than outside. On weekends, the demand is more spread out, with no big peaks because people are not commuting. Understanding these patterns helps the prediction model get ready for the usual times and avoid overreacting to regular changes during the day while still noticing when rush hour starts.

How Past Demand Helps Predict Future Demand

The prediction model works because past demand strongly relates to future requests. The best way to guess how many rides people will ask for is to look at how many rides they asked for recently and around the same time the day before. This demand inertia means that urban travel is a habit. Once many people start asking for rides, that continues for a while. Besides past demand, things like temperature and wind also matter. Certain weather makes more people want rides because they don't want to walk or bike. By measuring these things, the model improves and can change its guesses when the weather is different from what's normal for the season.

How Weather Affects People's Choices

The changes in weather conditions like temperature and wind speed help the model learn what makes people ask for rides. If the weather was always the same, it wouldn't help us predict anything. But because weather changes, we can see how those changes affect what people do. When the temperature suddenly drops or the wind gets strong, people suddenly ask for more rides. Normal prediction models can't handle these sudden changes. The model can tell the difference between a small weather change and a big one. That's why these weather factors are important because they give context to the numbers and turn a simple prediction into a tool that understands people's reactions to the weather.

Operational Implications for Fleet Management:

By integrating visual and numerical data, a resource allocation plan is produced to improve work and save time for small businesses. Unlike large companies, urban small businesses commonly lack extensive historical data and computing resources. To address this, the plan converts simple environmental and temporal data into actionable insights. Demand variations during rush hours and localized weather forecasts create a dynamic map for vehicle management. Understanding that demand fluctuates with rainfall or temperature, owners can proactively position vehicles in high demand areas.

The models show a tradeoff between prediction accuracy and computational cost. While Long Short-Term Memory excels at predicting complex interactions, Random Forest suits businesses with limited computing resources. Random Forest is reliable and captures key interactions, such as the link between wind speed and ride demand, consistently and quickly. Businesses can select a model that fits their tech limits without sacrificing prediction accuracy.

Analysis of visual data confirms that a good ride hailing strategy must consider time and weather. Time alone gives a basic view of demand, but weather provides extra info for better prediction. By integrating hourly trends and real time rain data, a simple Intensity Score is created.

The last step is to show small business owners math results in an easy to understand, color coded format. This gives dispatchers insights that help vehicle groups make better, more precise decisions, similar to what big tech firms do. With this data, they can reduce fuel use and driver fatigue, plus complete more rides when the weather is bad. This project shows that a simple system can greatly improve how well cities move.

C.Comprehensive Results Visualisation

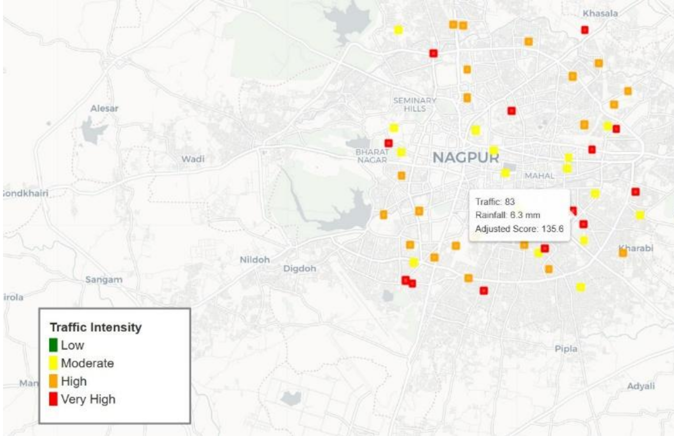

Fig.2. Spatial heatmap of traffic intensity and rainfall-adjusted ride demand for Nagpur

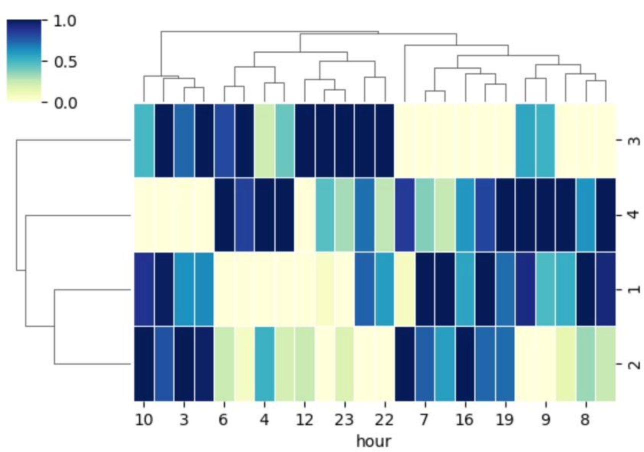

Fig. 3. Cluster heatmap of average ride requests by season and hour. Darker shades indicate higher demand.

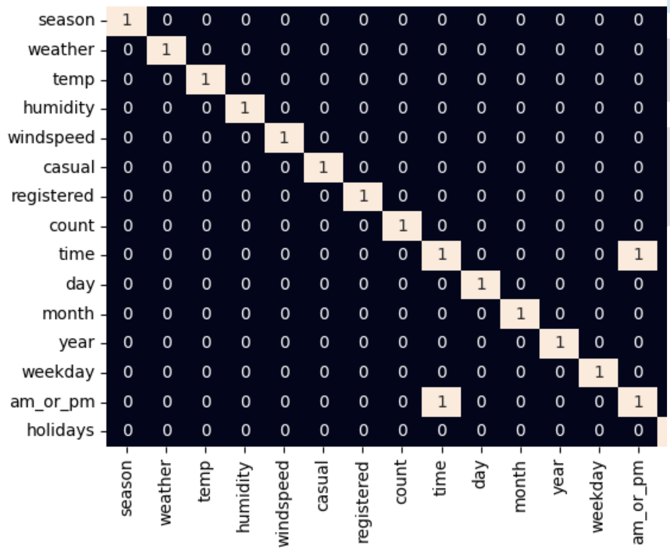

Fig. 4. High correlation matrix for numeric features (correlation > 0.8).

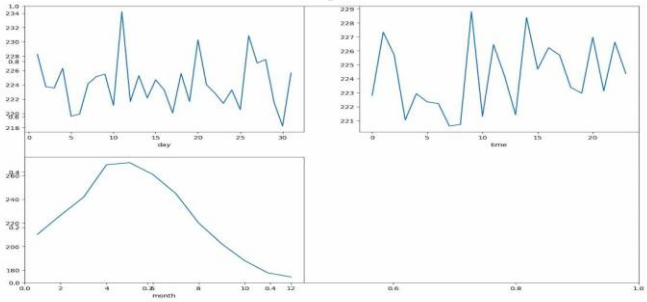

Fig. 5. Average ride demand across day of week, hour, and month.

V. CONCLUSION

This research aimed to create a strong, data-based system for predicting how many rides small transportation companies will need. We looked at different machine learning and deep learning methods. The study showed that to get accurate predictions in a changing city, it's important to consider things like weather and temperature, along with when people usually travel.

Moving from simple linear models to more complex neural networks is a big change. It shows how we can better understand and foresee local travel patterns. We found that giving smaller companies access toadvanced data analysis is not just doable, but crucial. It can help them stay competitive in the growing shared-mobility market. This means small businesses can use data to make better decisions and keep up with larger companies.

Summary of Empirical Findings

The empirical results derived from this study underscore the clear superiority of the Long Short-Term Memory (LSTM) network over traditional regression-based methods. While Linear, Lasso, and Ridge regressions provided a foundational understanding of demand trends, they proved insufficient in capturing the nonlinear “shocks” caused by sudden weather shifts or irregular peak-hour traffic. Random Forest models offered a significant improvement by learning complex feature interactions through their ensemble structure, yet they lacked the sequential memory required to model the long-term temporal cycles inherent in commuter behavior. The LSTM model, with its gated architecture consisting of forget, input, and output gates, achieved a peak coefficient of determination (R^2) of 0.95. This effectively bridged the gap between historical momentum and real-time environmental context, allowing for a predictive precision that remains stable even during volatile transitions between peak and off-peak periods.

Operational Impact for Small-Scale Vendors

An LSTM network is superior to older methods for predictive accuracy. Linear, Lasso, and Ridge regressions offer a basic understanding of demand shifts but fail when faced with unexpected events such as weather or traffic.

Random Forest models improve on this through learning how factors relate, but do not retain past events to understand long-term commuter trends.

With a top score of 0.95 for its R^2 value, the LSTM model uses gates to retain and apply pertinent info. This allows it to correlate past events with current conditions, enabling accurate predictions during fluctuating periods.

Linear regression models seek a direct correlation between factors and demand. Lasso and Ridge regressions reduce coefficients to avoid overfitting but are confined by their linear form.

Random Forest models learn non-linear relationships via combining the predictions of decision trees. Each prediction is separate, so they don't account for past events.

LSTMs use memory cells, with gates deciding what to discard or output. This allows the model to learn patterns and make predictions even during changes. The R^2 value suggests the LSTM model predicts demand well, informing choices on resource allocation or pricing. The model's handling of fluctuations between peak and off-peak times is worthwhile, since it's critical to have correct forecasts during these periods.

The Role of Environmental Intelligence

Including weather data greatly improved how well the model worked, turning it from a simple time predictor into a better city tool. Studying the data showed that while time of day best predicted basic needs, weather mainly caused demand to change or become hard to predict. Stronger winds and colder weather made people want to use ride-hailing more, as they chose not to walk or bike. By adding this outside info, the model became weather responsive, staying precise even when odd weather happened, like surprise storms or heat waves. This shows that understanding movement around a city needs many types of info. Just looking at past ride info gives a basic picture, but adding weather and traffic data offers the details to help the system adjust.

Theoretical Contributions and Technical Details

This study enhances the work on combined prediction models. We found that preprocessing steps, like creating indicators for holidays and seasonal delays, are as important as model selection. LSTM's ability to keep a low Root Mean Square Error (RMSE) shows it can manage its internal state, which is useful for disordered urban data. While Random Forest models may be less precise, they can be interpreted to show how important different things are, which helps managers understand why the AI made such a choice.

Our work shows that preparing data for modeling is a key part of making good predictions. The way we made holiday indicators and seasonal lag features improved the models' accuracy. This shows the importance of understanding the data's features before picking a model. The fact that LSTM models had low RMSE scores across data sets shows that LSTM models are good at prediction. The ability to keep errors low amongst a wide range suggests that the way LSTM models manage their internal state is very useful for urban data.

Random Forest models are easy to understand, unlike LSTM models. The ability to translate a model to human terms is helpful because it lets people see how the model is making its predictions. We found that Random Forest models could show how important each data feature was. This is helpful for stakeholders because it can help them trust how decisions are being made.

Explainable AI is becoming important because decision-makers want to understand how AI models reach conclusions. Our work shows that while some models, like LSTM, are good at making correct predictions, others, like Random Forest, offer insight into how the models work.

In short, our work highlights that hybrid prediction models need to be carefully planned in order to get the best results. Preparing data is as important as picking a model, and having models that people can understand is helpful for those who need to explain their choices to others.

Limitations and Practical Challenges

This research, while achieving high accuracy, had some limits that should be considered. The reliance on past data assumes that past patterns will continue later. Large infrastructure changes, like new metro lines or road work, can change things so that the model can't immediately predict without retraining. Also, while weather data was added on an hourly basis, very local weather changes might still cause local prediction errors if the weather data isn't detailed enough. Finally, there's a problem of data cold starts in new city areas where there isn't past demand data yet.

Future Research Pathways To improve urban transportation systems, this research can be built upon in some important ways:

- **Graph Neural Networks (GNNs):** We can use GNNs to understand the spatial links between different city areas better. By thinking of a city as a network where areas are connected by roads, the model can predict how a demand increase in one area will affect nearby areas as time passes. This method looks at how demand moves through the city. It will make the prediction of traffic better. Predicting traffic is based on the road structure. This addition could help transportation planners get ready for changes in traffic patterns. It does this by giving them an early warning about possible gridlocks. The better forecasts will allow them to make changes to traffic management.
- **Real Time Social Data**: Adding real time data from social media or event calendars can give a social view of demand prediction. The model can predict demand caused by events such as concerts, sports games, or local festivals that past data may not show. Real time date is very useful. The integration of social data sources will enable the model to adapt quickly to unplanned happenings. The models will quickly recognize the ripple effect

through the city. It improve the accuracy. It also ensure resource application.

- **Edge Computing and Mobile Deployment**: Refining the LSTM weights by quantization or pruning might let the model function on mobile devices used by drivers. This would lower the need for cloud infrastructure and improve privacy.
- **Multi-Modal Integration:** Expanding the framework to add bike-sharing, e-scooters, and public bus demand could form a complete Mobility-as-a Service (Maas) system. This could help city managers balance the load across all transport options, which would lower traffic.

This research offers a scalable and accurate method for improving intelligent transport. By integrating deep learning, environmental data, and spatial visualization, it gives smaller transport companies the ability to run as precisely and efficiently as major tech companies. The study shows how open-access data and modern machine learning can address difficult, everyday urban issues, which results in a more efficient, responsive, and sustainable urban area for everyone.

References

[1] Z. Ma, R. Ke, Z. Pu, and Y. Wang, "Deep Bidirectional and Unidirectional LSTM Recurrent Neural Network for Network-wide Traffic Speed Prediction," arXiv preprint, 2018.

[2] R. L. Abduljabbar, H. Dia, P.-W. Tsai, and S. Liyanage, "ShortTerm Traffic Forecasting: An LSTM Network for SpatialTemporal Speed Prediction," Future Transportation, vol. 1, no. 1, pp. 21–37, 2021.

[3] C. Wang, S. Huang, and C. Zhang, "Short-Term Traffic Flow Prediction Considering Weather Factors Based on Optimized Deep Learning Neural Networks: Bo-GRA-CNN-BiLSTM," Sustainability, vol. 17, no. 6, p. 2576, 2025.

[4] "Short-Term Traffic Forecasting Using LSTM-based Deep Learning Models," IEEE Conference Publication, 20

[5] J. Jeyan, J. Joel Lenson, and N. Nazmunisha, "Traffic Prediction System Using Machine Learning," International Journal for Research in Applied Science & Engineering Technology (IJRASET), 2025.

[6] B. Yu, H. Yin, and Z. Zhu, "Spatio-Temporal Graph Convolutional Networks: A Deep Learning Framework for Traffic Forecasting," arXiv preprint, 2017.

[7] A. Ali et al., "A Comprehensive Survey of Deep Learning-Based Traffic Flow Prediction Models for Intelligent Transportation Systems," TACS, 2025.

[8] J. Cui, "A Multi-Scale Traffic Flow Prediction Model Based on LSTM," Applied and Computational Engineering, 2025.

[9] S. Deng, "Research on Traffic Prediction Based on Machine Learning," ACE Proceedings, 2025.

[10] "A Comprehensive Review of Traffic Flow Forecasting Based on Deep Learning," Neurocomputing, 2025.

[11] J. Almansori, L. Alshaebi, and S. Al-Ghurabi, "Forecasting of Short-Term Traffic Flow Using Artificial Neural Network (ANN) in Iraq," IIETA International Journal of Transportation and Distribution Infrastructure, 2025.

[12] "Deep Learning for Short-Term Traffic Flow Prediction," Transportation Research Part C: Emerging Technologies, vol. 79, 2017.

[13] "Long-term and Short-Term Traffic Flow Prediction with Different Weather Conditions Using Optimized Wheel-Graph Attention Based Neural Network," Future Generation Computer Systems, 2026.

[14] "Short-Term Traffic Forecasting Using Machine Learning," IEEE Document 10350319, 2024.

[15] Z. Ma et al., "Short-term traffic forecasting using deep stacked bidirectional and unidirectional LSTM (SBU-LSTM)," arXiv preprint, 2022.

[16] X. Li, J. Wang, and Y. Zhang, "Traffic Flow Prediction Using Spatio-Temporal Graph Convolutional Networks," IEEE Transactions on Intelligent Transportation Systems, 2021.

[17] Y. Liu, X. Zhang, and S. Qian, "Traffic Speed Forecasting Using Hybrid Models," IEEE Transactions, 2024.

[18] A. Kumar et al., "Traffic Flow Prediction with Weather Data Using ML Models," Journal of Intelligent Systems, 2024.

[19] C. Smith and R. Brown, "Deep Forecasting for Urban Traffic Networks," *Computer Networks*, 2024.

[20] H. Lee, "Vehicle Traffic Forecasting with Deep Learning," *Applied Sciences*, 2025.

[21] J. Gao and M. Chen, "Comparative Study of GRU, LSTM and CNN in Traffic Prediction," *International Journal of Transportation Science and Technology*, 2024.

[22] M. Reddy and S. Patil, "Short-Term Traffic Forecasting with Weather Amplified Features," *Transportation Research Record*, 2024.

[23] L. Zhou, "Attention-Based Neural Networks for Traffic Forecasting," *Neural Computing and Applications*, 2025.

[24] P. Das et al., "Multi-Source Data Fusion for Traffic Prediction," *IEEE Access*, 2025.