IRSTI 28.23.25 https://doi.org/10.26577/jpcsit20253106

Nurbolat Amilbek[1], Beibut Amirgaliyev[1,*], Didar Yedilkhan [1], Zharasbek Baishemirov[1,2]

[1] Astana IT University, Astana, Kazakhstan
[2] Kazakh British Technical University, Almaty, Kazakhstan
*e-mail: beibut.amirgaliyev@astanait.edu.kz

DEVELOPMENT OF A DEEP LEARNING MODEL FOR FORECASTING AND OPTIMIZING RIDE-SHARING ROUTES

Abstract. The study investigates the potential use of machine learning (ML) technologies, including Recurrent Neural Networks (RNNs), in ride-sharing and urban mobility optimization. Advanced deep learning (DL) models can solve growing challenges in urban areas, such as road safety, environmental pollution, and traffic congestion. Three different RNN architectures (SimpleRNN, LSTM, GRU) are compared to predict trips with their pickup and drop-off points. According to the assessment metrics, GRU shows better results in terms of Mean Haversine Distance (6.450 km) than SimpleRNN (7.156 km) and LSTM (6.569 km). Moreover, the GRU model surpasses other models in other indicators, such as MSE (0.0010) and MAE (0.0211). In addition, OSRM API is used to build routes between predicted pickup and drop-off points, as well as to optimize ride-sharing routes using real-time geographic data. The study highlights that ML approaches, in particular DL, can be used to solve problems related to urban mobility by improving transport efficiency and reducing traffic. The study results provide recommendations for developing urban transport systems using data-driven approaches to enhance ride-sharing opportunities.

Key words: Trip Forecasting, Ride-Sharing, Machine Learning, Deep Learning, SimpleRNN, Long Short-Term Memory, Gated Recurrent Unit.

1. Introduction

Turning rural populations into urban ones is known as urbanization, typically seen as a gauge of social and economic progress. It is a crucial sign of modern society and a crucial historical phase in the industrialization of any nation [1]. Urbanization has been a defining global trend for many years, and in the twenty-first century, it is still growing quickly. Approximately 56% of people on the planet now reside in cities, and nearly 67% of the world's population is expected to live in cities by 2050 [2]. Urbanization has also occurred in Kazakhstan. As of 2023, 58.18% of Kazakhstan's population resides in urban regions 2023 [3]. The proportion of Kazakhstan's population living in cities has gradually increased during the last ten years. About 53.2% of people lived in cities in 2009; by 2019, that number had increased to 57.6%. In absolute terms, the number of people living in cities increased from approximately 8.5 million in 2009 to 10.5 million in 2019 [4].

Congestion, traffic safety, air pollution, and greenhouse gas emissions are only a few of the transportation-related issues that have worsened due to urbanization [5]. The Texas Institute of Transportation claims that individual congestion indicators can be measured using the following indicators. The values for 2022 are also shown below: Yearly delay per auto commuter, the additional time that private vehicle drivers and passengers who normally travel during peak hours spend travelling at crowded speeds throughout the year as opposed to free-flowing speeds, and this value is equal to 54 hours; The next indicator is the travel time index, the ratio of travel time during peak hours to travel time in free traffic, and the result for 2022 is 1.21 [6]. It means that a 20-minute free-flow commute takes 26 minutes during peak hours. According to the World Health Organization, 1.19 million people die in car accidents each year, and road safety is the second most important issue [7]. The next challenge is the impact of urbanization on air quality: in 2017, air pollution from transportation caused 3.5 million premature deaths from cancer, diabetes, heart disease, lung infections, obstructions, and respiratory infections. Furthermore, global transport emissions resulted in approximately 8 million years of lost life and USD 1 trillion in health effects in 2015 [8].

According to Guyader et al. (2021), shared mobility through initiatives like car, bike, and ride-sharing can be seen as a strategy to lessen traffic on the roads, transportation infrastructure, carbon dioxide emissions, and the environmental impact of travel. [9]. In addition, some cities have implemented data-driven approaches to improve urban transportation. For example, applications of data-driven technology include automatic traffic signal control based on data gathered on traffic congestion using sensors integrated into traffic lights and transportation service management based on data received by the situation center in London. Global Positioning System (GPS) sensors are used to track the movements of public transport in Barcelona. Additionally, smart traffic light systems are used to automatically give priority to public transport and other modes of transport, including emergency services [10].

The study's goal is to optimize urban mobility and ride-sharing using ML methods, as it can be useful to increase the sustainability, reliability, and efficiency of urban transportation. To evaluate large datasets and find patterns, the usage of ML approaches is advantageous. The research and development process includes data collection, trip and weather data, data preprocessing, cleaning, and transforming into a suitable format for analysis, building, training, and evaluating models, trip prediction, building route, and optimizing ride-sharing route. Traditional transport management is often the cause of urbanization challenges in the transport sector of megacities, as it does not meet the dynamic requirements of large urban areas. ML methods for optimizing ride-sharing routes will reduce traffic, avoid traffic problems in the future, and develop more efficient urban mobility that meets the needs of urban residents.

2. Literature Review

Today a few urban planning strategies exist. For instance, the Park-and-Ride (PnR) system, implemented in Tsukuba, Japan, shows promise in lowering vehicle emissions and urban traffic congestion. According to the results, well-designed PnR systems with the ideal bus capacity and frequency minimize emissions, ease traffic, and enhance travel efficiency. In order to assess how PnR systems affect waiting times, travel times, and vehicle emissions, the study creates a mathematical framework that combines queueing theory and emissions modelling. During the study, a customized model is provided for calculating the total societal cost while considering the mean time of all trips and the mean emissions of all vehicles. The resulting cost in international dollars per capita is the total Social Cost of Emissions and Total Trip Time (SCETT). The PnR system is enhanced with SCETT to function as a tool for the finding of transport policies. The model incorporates SCETT for optimal parameters of interest that describe PnR hubs, such as bus frequency interval, bus capacity, and percentage vehicle usage. Compared to the current situation, deploying the suggested socially optimal PnR transportation strategy in Tsukuba city results in a 30% reduction in social expenses. A queueing model that combines two separate queues – one that records client movement and the other that records vehicle movement on the road – is presented to assess the waiting and travel times of consumers under various PnR situations. To ensure the robustness of the results, the theoretical approximation analysis using the matrix geometric approach and the Monte Carlo simulation are presented for this model. A comparison of the analytical and simulation results confirmed the approach's validity. Moreover, total car emissions and journey times are simulated using empirical data from the 2018 Person journey study, producing results that closely match observed data. For example, for journeys under 60 minutes, the simulation estimated an expected total travel time of 0.4249 h, which was quite similar to the corrected empirical figure of 0.3893 h [11].

The combination of shared mobility services (SMSs) and public transport is the next example. In order to reduce traffic and increase travel efficiency, the study focuses on how e-hailing, ridesharing, and carpooling services could improve the current public transit networks. SMSs have the potential to significantly increase mobility, particularly in areas with low demand and low population where public transport cannot offer first and last mile service. The paper investigates that the proposed model can be used in both continuous and integer settings, and it computes the user equilibrium, which states that every commuter attempts to reduce their own expenses, and when no one is prepared to alter their decisions in order to do so, equilibrium is reached, using the Beckmann formulation and the system optimum, which is balance of the system and presumes that the overall cost of the system is kept to a minimum, when it handles mode and path choices and passenger-driver matching simultaneously, leading to a Mixed-Integer Bilinear Programming (MIBLP) formulation. Moreover, the suggested model sheds light on variables influencing how the modes are used in a synthetic multi-modal network. Examining the cost of anarchy in these multimodal systems also contrasts user equilibrium with system optimal solutions. The Sioux Falls urban transportation is also studied to assess commuters' behavior and scale up the suggested model's applicability to realistic settings [12].

The last strategy is traffic signal scheduling using the model predictive control (MPC) approach, with the goal of equitably reducing delays for both cars and pedestrians. A framework for adaptive traffic signal control is introduced, considering a historical database, a machine learning-enabled artificial intelligence prediction model, a macroscopic mixed-flow optimization model, and the VISSIM simulation platform. The experiment results showed that the suggested approach in a network structured like Manhattan may effectively balance the requirements of both walkers and car drivers. This technique is more flexible and adaptable to meet passenger demand because it combines bus dispatching and on-road operation, particularly boarding control, to minimize passenger delay time and operating bus vacancy [13].

According to Sayed et al., using machine learning to urban mobility increases the precision of the city's transit distribution by forecasting traffic flow [14]. Moreover, the ride-sharing demand forecast is useful since 50% of carriers can finish 96% of trips [15–18]. Furthermore, driver-passenger matching decreases waiting times by 10–20 minutes, or from 30% to 43%, for both drivers and passengers [19–21]. The impacts of waiting and travel time optimization have also been the subject of several studies [20, 21]. Route optimization helps to reach more people for excursions by expanding the distribution of transport in the city for flexible pickup and drop-off locations [15, 19, 22]. Other research looks at ways to optimize current ride-sharing systems, such as integrating many of them [23].

According to research, Reinforcement Learning (RL) is the most widely used algorithm for optimizing urban mobility, particularly when combined with other techniques like Graph Neural Networks (GNN), Markov Decision Processes (MDP), Model Predictive Control (MPC), and Shared Autonomous Vehicles (SAV) to cut down on waiting times [16, 18, 21]. Additionally, Deep Q-Network (DQN) is a popular urban planning strategy for optimizing the dispersion of fewer transportation alternatives for many people [17].

There are limitations and disadvantages to the existing corpus of study. Large amounts of data are required for accurate forecasting and transportation system optimization, including traffic, passenger demand, and vehicle characteristics. Deep learning (DL) and reinforcement learning (RL) are two ML approaches that need a lot of processing capacity. Because of this, they can be costly for large transportation networks and are challenging to employ in real time. Furthermore, integrating new methods and models into the existing traffic management systems may prove difficult. This is a result of the need to upgrade infrastructure, update work procedures, and train personnel. Finally, certain models and methodologies developed for one urban area may not be applicable in another due to differences in infrastructure, population, and traffic patterns. To accomplish this, models need to be adjusted to each city's particular situation.

The current research overlooks the significance of meteorological variables in traffic predictions. Much more accurate traffic forecasts would be produced if traffic prediction algorithms included comprehensive meteorological data, such as temperature, wind speed, and precipitation. The forecasting models' applicability to various urban environments is also a significant problem in transport management. By modifying the input parameters to account for local traffic data, infrastructure, population, and other pertinent aspects, traffic forecasting models may be customized for use in a variety of places. Future studies will examine the effects of various weather patterns on traffic patterns and the effectiveness of various machine learning algorithms in using this data to generate forecasts that are more accurate. Additionally, it will offer and evaluate a methodology for modifying current traffic models to fit new urban settings while maintaining high levels of accuracy for ridesharing route construction and traffic forecasting in the face of shifting city-specific features.

3. Materials and Methods

3.1. Dataset

The study employed the New York City Taxi-Trip Distance Matrix dataset to optimize ride-sharing and anticipate routes. The Kaggle platform is the source of the dataset. Latitude, longitude, and pickup and drop-off timings are among the taxi trip parameters included in the dataset.

There are 39,396 rows in the dataset. Figure 1, which shows the number of taxi journeys for each hour of the day, indicates that the average number of taxi trips per day is around 1,700. The most journeys are made between 10 and 11 p.m., while the fewest are made between 5 and 6 a.m. Figure 2 shows the number of travels for each day of the week, with Friday accounting for the bulk of journeys. Additionally, Figure 3 displays the travel density for every hour and day of the week. It is observed that the hours between midnight and five in the morning are not the most popular periods for taxi rides, except for Fridays.

The next dataset is New York City Taxi Trip – Hourly Weather Data, from Kaggle platform. The dataset includes datetime, temperature in Celcius and Fahrenheit, dew point in Celcius and Fahrenheit, humidity, wind speed, wind direction in degrees, visibility in kilometers, pressure, boolean value rain, boolean value snow, and boolean value thunder.

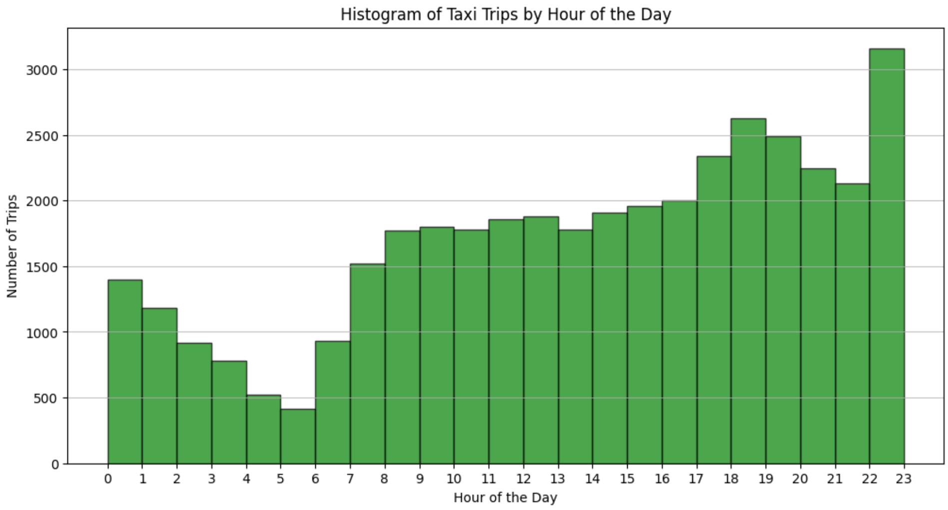

Figure 1 – Histogram of Taxi Trips by Hour of the day.

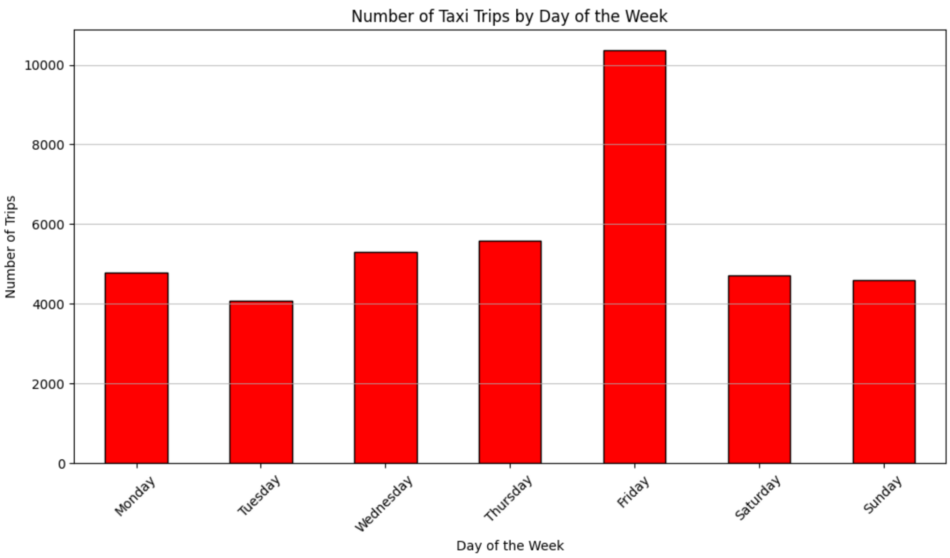

Figure 2 – Number of Taxi Trips by Day of the Week.

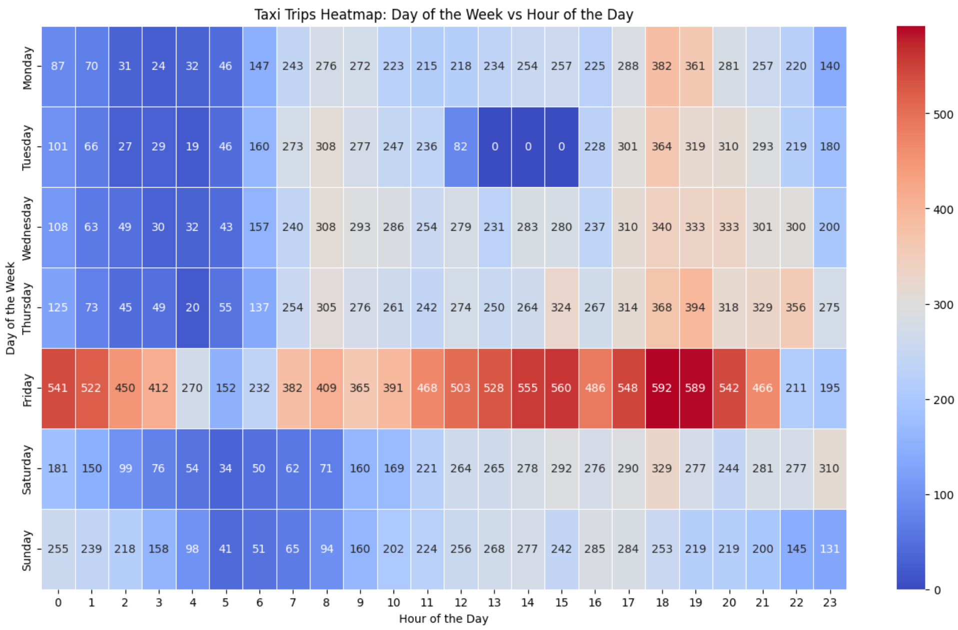

Figure 3 – Taxi Trips Heatmap: Day of the Week vs Hour of the Day.

New York City Taxi Trip – Hourly Weather Data dataset has the weather data for the full 2016 year (Figure 4). However, Figure 5 shows that the New York City Taxi Trip-Distance Matrix dataset contains data from January 1, 2016, to January 8, 2016, which is why weather data for this period is extracted.

The filtered dataset has 232 rows for the time frame from January 1 to January 8. Figure 6 displays the average temperature for each hour of the day. The average hourly temperature climbs substantially in the afternoon after falling in the morning, peaking between 15:00 and 16:00. The temperature peaks in the evening and night and then progressively drops until midnight, when it is at its lowest. The average daily temperature is shown in Figure 7. On January 1, the temperature is at its lowest. Over the next two days, it rises significantly, and on January 3, it slightly decreases. There is a discernible peak on January 4 and a steep drop on January 5. The most significant increase occurs on January 6 and reaches the greatest temperature in the record, followed by a small stability over the next three days.

```
First pickup time: 2015-12-31 00:15:00
Last pickup time: 2016-12-31 23:51:00
```

Figure 4 – First and last weather datetime.

```
Minimum pickup_datetime: 2016-01-01 00:00:00
Maximum pickup_datetime: 2016-01-08 21:56:00
Minimum dropoff_datetime: 2016-01-01 00:03:00
Maximum dropoff_datetime: 2016-01-31 01:01:00
```

Figure 5 – First and last taxi pickup and drop-off datetime.

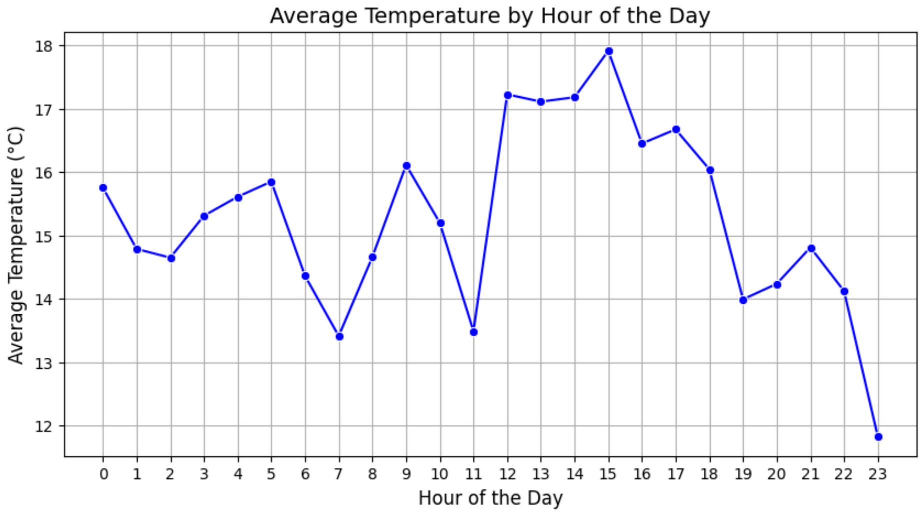

Figure 6 – Average Temperature by Hour of the Day.

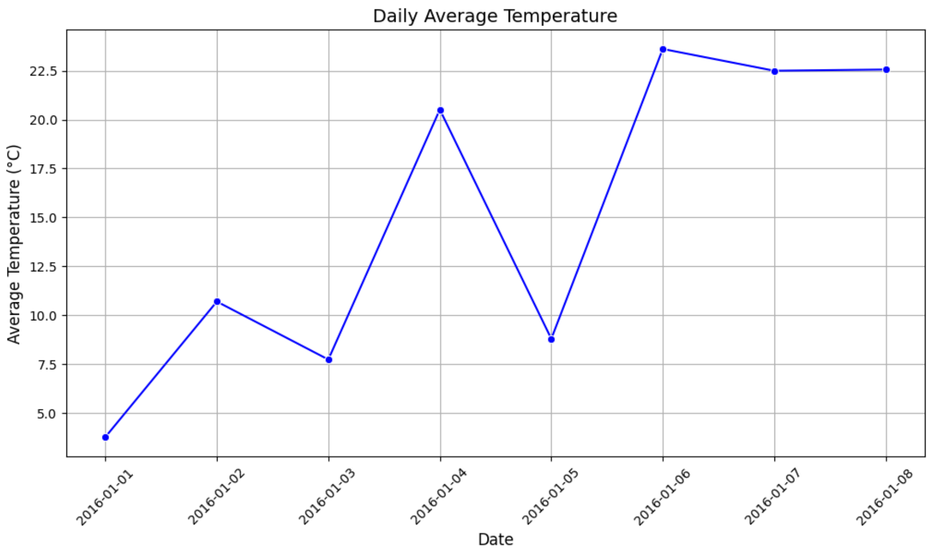

Figure 7 – Daily Average Temperature.

3.2. Model Development

A DL model pipeline with three main stages – Data Preprocessing, Model Development, and Simulation and Prediction – is shown in Figure 8. Five processes make up the Model Development process: feature extraction, data scaling, DL model construction, model training, and model evaluation using various metrics. Among these measurements are Mean Haversine Distance, Mean Squared Error (MSE), Mean Absolute Error (MAE), and Coefficient of Determination (R^2).

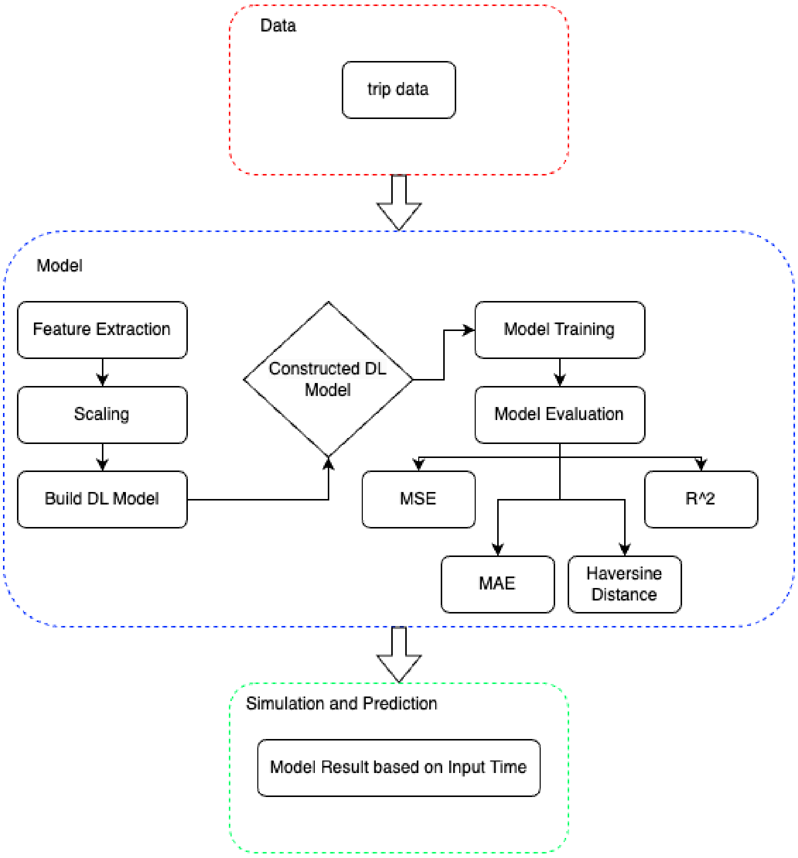

Figure 8 – Deep Learning Model Pipeline.

3.3. Data Preprocessing

The data preprocessing pipeline prepares the taxi trip dataset for the modelling. Firstly, before extracting pertinent time-based data, including the hour, minute, and day of the week, the pickup timestamp is transformed into a datetime format to record trip trends over time. The latitude and longitude coordinates are the target variables. Some datetime parameters, including the year and month, are not the features since the data only covers one year (2016) and one month (January). Figure 9 illustrates the significance of the input parameters in relation to the Long Short-Term Memory (LSTM) model. It is demonstrated that the LSTM model is most affected by the hour of the day, visibility based on the weather, pressure, and the presence of rain, but the day of the week, temperature, dew point, humidity, and wind speed have a somewhat smaller effect. The following parameters have little impact: fog, wind direction, and minute. The model is unaffected by the year or month. Furthermore, certain weather conditions – like the presence of snow, hail, thunder, and tornadoes – are superfluous.

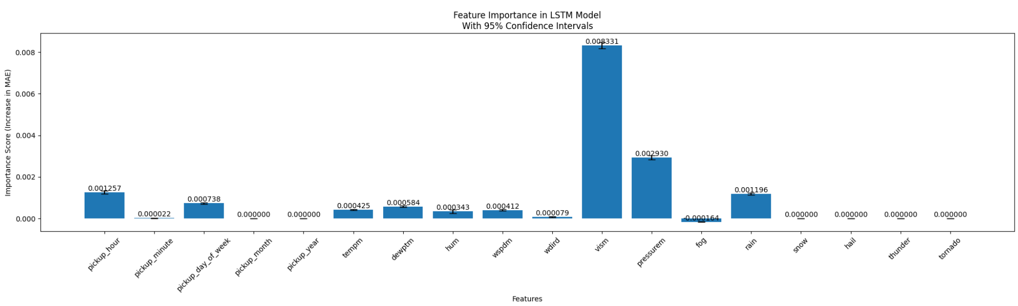

Figure 9 – Feature importance in LSTM Model.

3.4. SimpleRNN

The first architecture is a SimpleRNN. A SimpleRNN with 256 units and a hyperbolic tangent activation function makes up the first layer (Figure 10). A tangent function can model non-linear interactions, and a larger number of neurons enables the network to learn more complicated characteristics. Deeper representations are also possible because the initial RNN layer provides a parameter that guarantees the full sequence is transmitted across the network.

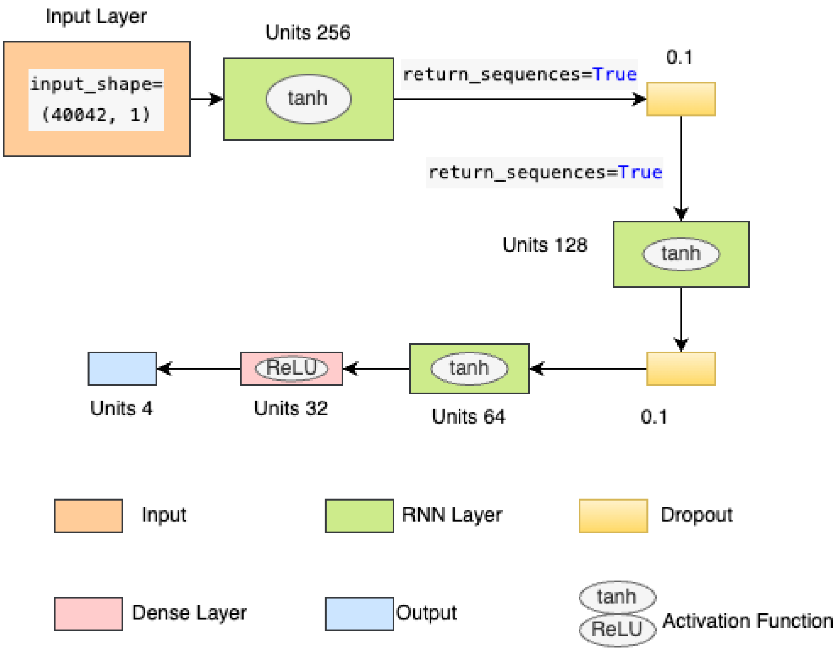

Figure 10 – SimpleRNN Architecture.

Dropout regularization is used after the first RNN layers with a dropout rate of 0.1 to counteract the possible overfitting from deep architectures. This drives the network to create redundant representations that better generalize to unseen data by randomly setting 10% of the units to zero at each update cycle during training.

Additionally, the model layers a second 128-unit SimpleRNN layer (which also has tangent activation and a parameter to return the final output), a 10% Dropout layer, and a third 64-unit SimpleRNN that does not return sequences.

After the recurrent layers, the network switches to the Dense layer, which is a 32-unit layer that uses the rectified linear unit (ReLU) activation to provide non-linear adjustments that make it easier to capture complex relationships between the compressed attributes and the target variables.

The four objective outputs (pickup latitude, pickup longitude, drop-off latitude, and drop-off longitude) are represented by the four units of the final Dense layer.

The Adam optimizer, which has a learning rate of 0.001, is used to optimize and train the model. The loss function employed is the MSE, which is standard for regression tasks, and the MAE is tracked as an additional performance metric to provide more interpretable error magnitudes.

Early stopping is implemented as a safeguard against overfitting by monitoring the validation loss. With a patience parameter set to 10 epochs, the training process halts if the validation loss does not improve over 10 consecutive epochs, and it restores the best model weights observed during training.

The model is trained on a dataset of scaled input features and corresponding target variables with 50 epochs and a batch size of 32, while validation is performed on the test dataset.

3.5. LSTM

The next implemented DL model is LSTM. The architecture commences with an LSTM layer containing 64 hidden units and utilizing ReLU activation function (Figure 11). Moreover, the parameter, that returns sequence, is pivotal, as it ensures that the complete sequence of outputs is relayed to the subsequent LSTM layer, thereby preserving the temporal structure of the input data.

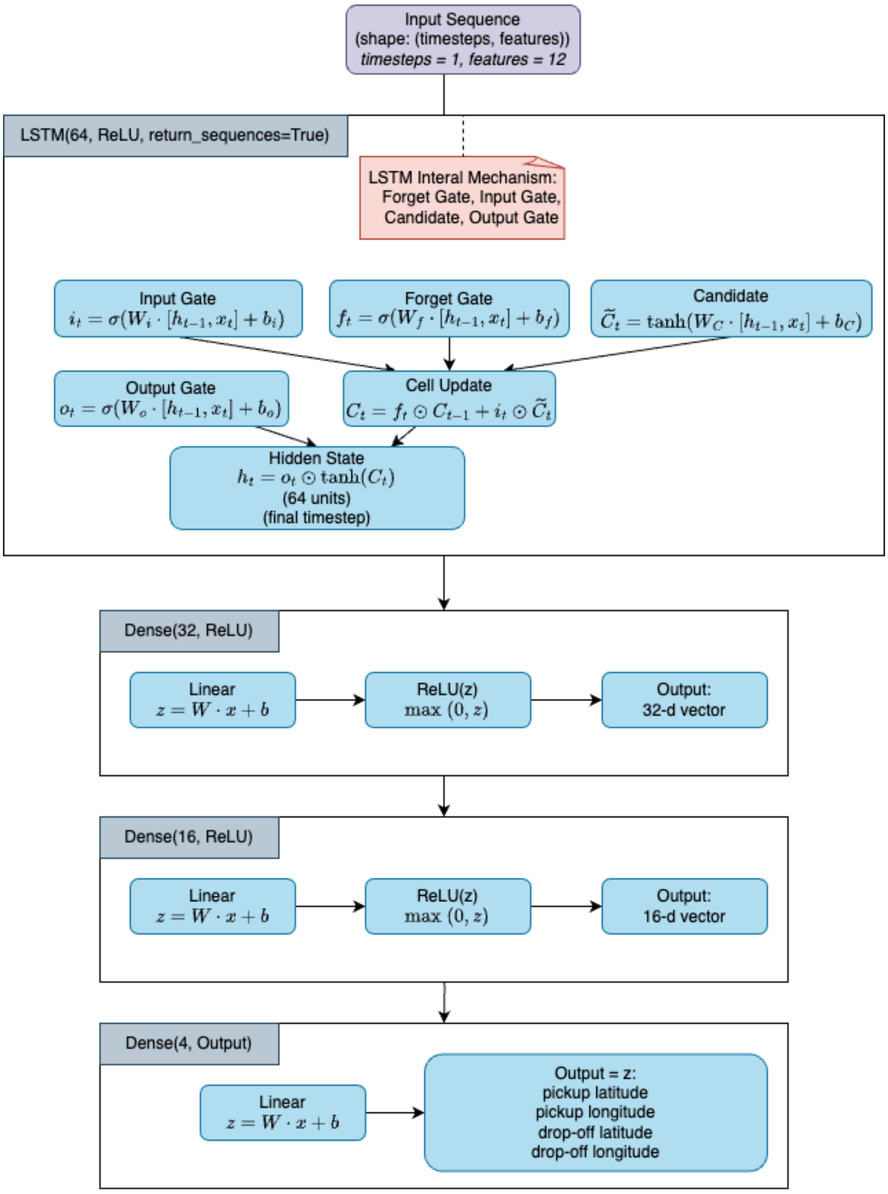

Figure 11 – LSTM Architecture.

In order to transfer the high-level characteristics to the final regression outputs, the temporal features that were recovered by the LSTM layers are then processed via a number of Dense layers. The first Dense layer, which has 32 units and a ReLU activation, introduces non-linearity to enhance the feature representation further. The following step is a second 16-unit dense layer that continues the feature condensation process while retaining the critical information required for accurate prediction. The last Dense layer, which comprises four units, directly outputs the predicted values for the four target variables.

The model is optimized using the Adam optimizer, which has a learning rate 0.001. The model is trained according to the MSE loss function, and its performance is assessed using the MAE measure. The likelihood of overfitting is decreased by employing an early stopping technique. After 10 consecutive epochs, it monitors the validation loss and terminates training if no improvement is observed. Training is conducted with a batch size of 32 over 50 epochs to allow for stable learning dynamics.

3.6. GRU

The Gated Recurrent Unit (GRU) architecture implements the final model. Using the tangent activation function, the first layer, the GRU layer with 128 units (Figure 12), effectively learns temporal connections while addressing the vanishing gradient issue. By configuring the parameter in the first layer, more temporal patterns can be captured by propagating the entire series of concealed states to the subsequent layer.

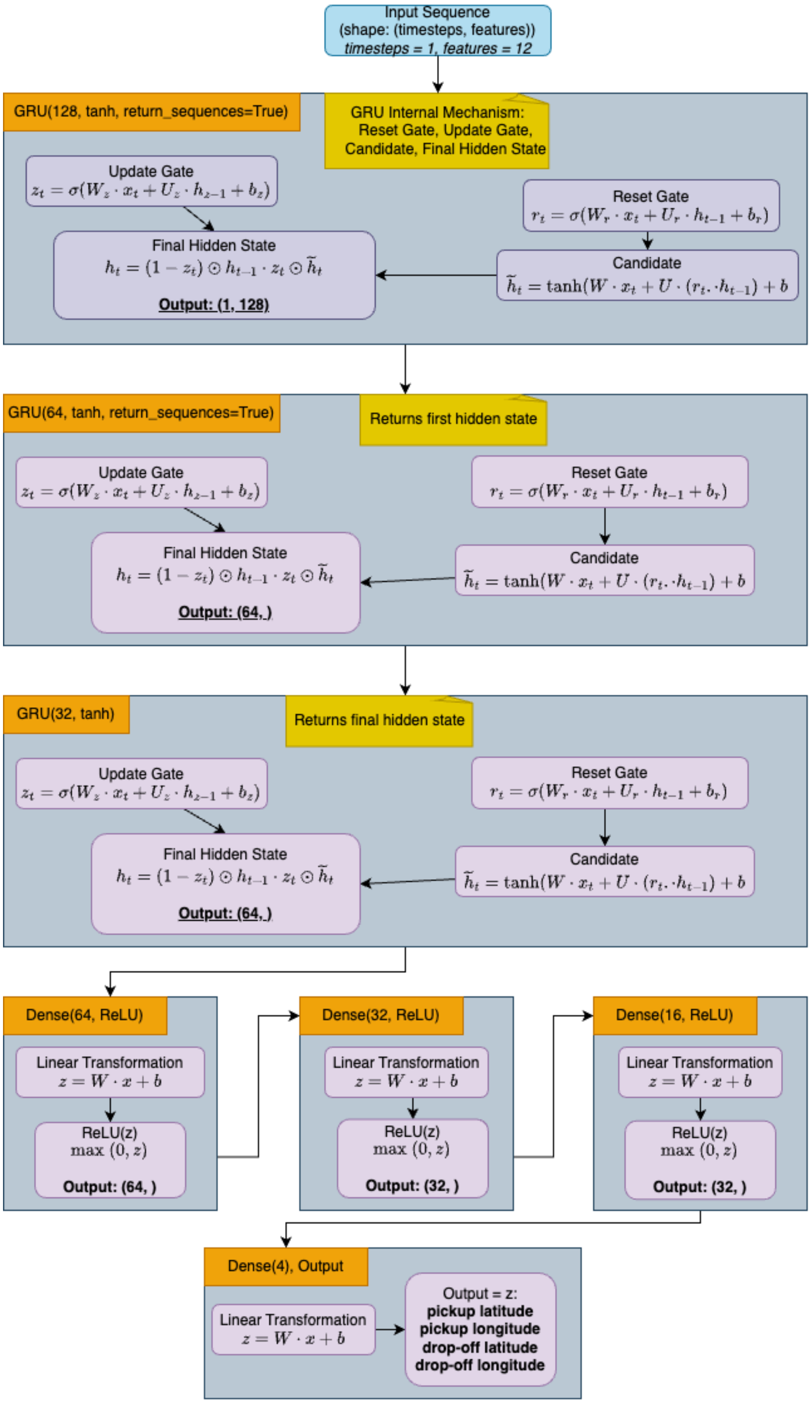

Figure 12 – GRU Architecture.

After the first level, the second GRU level is implemented, consisting of 64 blocks, and then 32 blocks. To gradually decrease the time information and reduce the complexity of the model, the number of blocks has been purposefully minimized. The tangent function is also utilized in these layers. However, there is no parameter to return sequences in the last layer with 32 neurons.

The model uses a sequence of Dense layers to transform the temporal representations into the final output space, departing from the recurrent design. The purpose of the first Dense layer's 64 units, which have a ReLU activation function, is to integrate nonlinear transformations that help close the gap between sequential feature extraction and the regression task. The feature space is further refined by a second Dense layer of 32 units, which enables the model to recognize intricate patterns relevant to the prediction task. The last Dense layer, which has four units, has a strong correlation with the goal outputs.

The model is trained with the MSE loss function and optimized with the Adam optimizer (learning rate = 0.001), with MAE as an evaluation metric. Early stopping is used to enhance generalization and avoid overfitting, monitoring validation loss and restoring the best model if no improvement is seen over 10 epochs. A batch size of 32 is used for training over 50 epochs, ensuring a balance between computing efficiency and learning stability.

3.7. Trip Prediction

The first step of the prediction pipeline involves entering meteorological and time information and then extracting important time variables, including the hour, minute, and day of the week, from the designated pickup time. Following this, the features are normalized using the same scaler employed during the training phase to guarantee consistency in the data transformation procedure.

Additionally, a time range can be added to the pipeline, allowing one to predict which trips will take place before and after the chosen time window. The model then predicts the locations for pickup and drop-off based on the scaled inputs. Using the output scaler that was set up during training, the inverse transformation is then applied to return the predicted values to their initial scale.

3.8. Ride-Sharing Route Building

The OSRM API is used to determine the best driving routes between designated pickup and drop-off locations. Using OpenStreetMap data, the OSRM service offers road network routing options that are both quick and versatile. The Folium program is then used to depict each route on an actual road map. Figure 13 shows a flowchart for a trip forecasting and routing system that includes trip forecasting, route design with OSRM API connectivity, and map visualization.

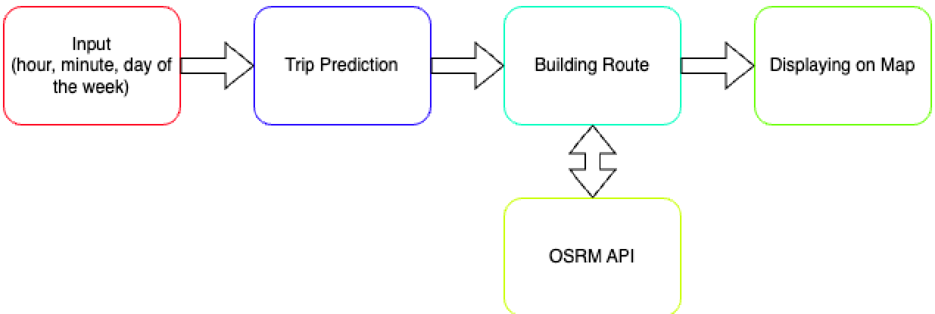

Figure 13 – Trip Forecasting and Routing System Stages.

4. Results and Discussions

Table 1 shows the metrics of the three DL models: SimpleRNN, LSTM, and GRU.

The GRU model is chosen as the main predictive model for ride-sharing route optimization and pickup and drop-off location estimation. Compared to other models, its performance on various evaluation metrics supports this choice. A lower prediction error and more accuracy in estimating location coordinates are indicated by the GRU model's lowest MAE and MSE. It also achieves the second-highest R^2 value, indicating better explanatory power across trip location variability. Most significantly, the GRU model's Mean Haversine Distance is the lowest of all the models evaluated, demonstrating its improved spatial accuracy in trip trajectory prediction. These results affirm that the GRU model offers a balanced trade-off between predictive accuracy and generalization capability, making it the most suitable choice for this application.

Table 1 – Deep Learning Metrics.

#	Model	MSE	MAE	R^2	Mean Haversine Distance
1	SimpleRNN	0.0011	0.02303	0.0239	7.156
2	LSTM	0.0010	0.0215	0.1098	6.569
3	GRU	0.0010	0.0211	0.1018	6.450

10:00 a.m. on a Wednesday, considering a 5-minute time window (9:55 – 10:05 a.m.), was used to predict pickup and drop-off locations for New York City. Furthermore, the following weather conditions have been added as input parameters: temperature (12 degrees Celsius), dew point (-4 degrees Celsius), humidity 50%, wind speed (7 km/h), wind direction (0 degrees), visibility (15 km), pressure (1017.9 mBar), rain, and no fog. The anticipated pickup and drop-off sites are shown in Figure 14 as follows: red lines connecting the pickup and drop-off spots are the best roadways, as determined by the OSRM API; blue dots represent pickup locations, and green points represent drop-off locations.

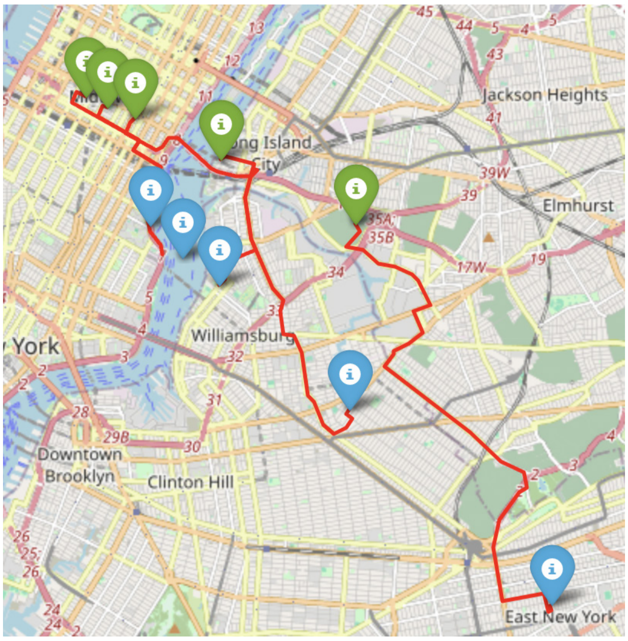

Figure 14 – GRU Model Predicted Trips.

To share rides, one can utilize a ride-detecting algorithm. To find a trip where the pickup location is closer than the drop-off location, at a guaranteed maximum distance (1 km), and in the same direction, the algorithm first randomly selects the pickup location on the trip. It then calculates the distance between the pickup location and the drop-off location, as well as between the pickup location and another drop-off location. The drop-off site will be chosen once each of these requirements has been satisfied. The last drop-off location is linked to the route, and it is contingent upon which drop-off location is closest to the second pickup location.

Figure 15 shows the selected trips, combined pickup and drop-off sites, and corresponding routes. The system chose trips 32603 and 32604 to be shared. These goods satisfy the previously mentioned requirements: Less than 1 mile separates the boarding and disembarkation points, and both trips are taken in the same direction.

(a) (b) (c) (d)

Figure 15 – Selected Trips for Ride-sharing: (a) Pickup Point 32603; (b) Dropoff Point 32603; (c) Pickup Point 32604; (d) Dropoff Point 32604.

The ability to share travels by specifying trips is seen in Figure 16. At 10:00 a.m., the joint journey starts at the first pickup site (Figure 16(a)) and travels to the second pickup location at 10:00 a.m. (Figure 16(b)), continues to the first drop-off point (Figure 16(c)) and ends at the second drop-off point (Figure 16(d)).

Figure 16 – Ride-sharing route: (a) First Pickup Point; (b) Second Pickup Point; (c) First Drop-off Point; (d) Second Drop-off Point.

5. Conclusions

This study tackles the problems with urban mobility exacerbated by increasing urbanisation by offering a DL-based approach for anticipating and optimizing ride-sharing routes. The study demonstrates the effectiveness of recurrent neural networks, particularly the GRU model, by combining temporal and meteorological data to improve the accuracy of trip location forecasts. The model's performance across several evaluation parameters, including Mean Haversine Distance, R^2, MAE, and MSE, guarantees high spatial accuracy in trip trajectory calculations, supporting the model selection.

Additionally, a ride-sharing system that optimizes the most effective pickup and drop-off sequences was created by integrating the GRU model with geolocation data. This method improved practical applicability by ensuring that the chosen routes matched existing road networks through route optimization utilizing the OSRM API. In addition to reducing unnecessary journey kilometers, the methodical approach will increase overall transportation efficiency, reduce urban traffic, and improve passenger convenience.

ML-based ride-sharing optimization solutions, such as trip prediction and route planning, increase transportation efficiency by reducing traffic and travel delays. The results show that data-driven strategies can successfully support urban planning efforts targeted at minimizing environmental impact and enhancing commuter experiences by aiding in the creation of intelligent and sustainable transportation networks.

Funding

This research has been funded by the Committee of Science of the Ministry of Science and Higher Education of the Republic of Kazakhstan (Grant No.BR24992852 "Intelligent models and methods of Smart City digital ecosystem for sustainable development and the citizens" quality of life improvement").

Author Contributions

Conceptualization, N.A., B.A., Z.B. and D.Y.; Methodology, B.A. and N.A.; Software, N.A. and B.A.; Validation, N.A., Z.B. and D.Y.; Formal Analysis, Z.B.; Investigation, D.Y.; Resources, B.A., and D.Y.; Data Curation, N.A. and Z.B.; Writing – Original Draft Preparation, N.A. and D.Y.; Writing – Review & Editing, B.A. and Z.B..; Visualization, N.A.; Supervision, B.A.; Project Administration D.Y.

Conflicts of Interest

The authors declare no conflict of interest.

References

1. Zhang, Z., Zhao, M., Zhang, Y., & Feng, Y. (2023). How does urbanization affect public health? new evidence from 175 countries worldwide. *Frontiers in Public Health, 10*. https://doi.org/10.3389/fpubh.2022.1096964.

2. Hoffmann, E. M., Schareika, N., Dittrich, C., Schlecht, E., Sauer, D., & Buerkert, A. (2023). Rurbanity: A concept for the interdisciplinary study of rural–urban transformation. *Sustainability Science*, *18*(4), 1739–1753. https://doi.org/10.1007/s11625-023-01331-2.

3. LLC, G. (n.d.). *Kazakhstan percent urban population – data, Chart*. TheGlobalEconomy.com. https://www.theglobaleconomy.com/Kazakhstan/Percent_urban_population/.

4. Kabdesov, K. (2020). Urbanization trends and processes in Kazakhstan. *Eurasian Journal of Economic and Business Studies*, *1*(55). https://doi.org/10.47703/ejebs.v1i55.11.

5. *M1: Growing problems in urban transport*. M1 | Growing Problems in Urban Transport. (n.d.). https://thedocs.worldbank.org/en/doc/6e7165ca96955c40eda96ed29e0a4834-0090062024/m1-growing-problems-in-urban-transport.

6. *2023 urban mobility report and appendices – mobility division*. Mobility Division – Texas A&M Transportation Institute. (2024, June 11). https://mobility.tamu.edu/umr/report/.

7. World Health Organization. (n.d.). *Global status report on road safety 2023*. World Health Organization. https://www.who.int/teams/social-determinants-of-health/safety-and-mobility/global-status-report-on-road-safety-2023.

8. Piracha, A., & Chaudhary, M. T. (2022). Urban Air Pollution, urban heat island and human health: A review of the literature. *Sustainability*, *14*(15), 9234. https://doi.org/10.3390/su14159234.

9. Guyader, H., Friman, M., & Olsson, L. E. (2021). Shared mobility: Evolving practices for sustainability. *Sustainability, 13*(21), 12148. https://doi.org/10.3390/su132112148.

10. Bibri, S. E., & Krogstie, J. (2020). The emerging data–driven smart city and its innovative applied solutions for sustainability: The cases of London and Barcelona . *Energy Informatics, 3*(1). https://doi.org/10.1186/s42162-020-00108-6.

11. Nakamura, A., Ferracina, F., Sakata, N., Noguchi, T., & Ando, H. (2025). Reducing total trip time and vehicle emission through Park-and-ride – methods and case-study. *Journal of Cleaner Production*, *493*, 144860. https://doi.org/10.1016/j.jclepro.2025.144860.

12. Kadem, K., Ameli, M., Zargayouna, M., & Oukhellou, L. (2024). Multimodal urban transportation network equilibrium including intermodality and shared mobility services. *arXiv.Org*. https://doi.org/https://doi.org/10.48550/arXiv.2402.00735.

13. Zhang, Y. (2023). Optimization and scheduling for a large scale urban transportation system in a fast-changing world. *Systems and Control*. https://doi.org/https://doi.org/10.48550/arXiv.2301.12087.

14. Sayed, S. A., Abdel-Hamid, Y., & Hefny, H. A. (2023). Artificial intelligence-based traffic flow prediction: A comprehensive review. *Journal of Electrical Systems and Information Technology, 1*0(1). https://doi.org/10.1186/s43067-023-00081-6.

15. Yuan, E., & Van Hentenryck, P. (2021). Learning model-based vehicle-relocation decisions for real-time ride-sharing: Hybridizing learning and optimization. *arXiv.org*. https://doi.org/10.48550/arXiv.2105.13461.

16. Wen, D., Li, Y., & Lau, F. C. (2024). A survey of machine learning-based ride-hailing planning. *IEEE Transactions on Intelligent Transportation Systems, 25*(6), 4734–4753. https://doi.org/10.1109/tits.2023.3345174.

17. Al-Abbasi, A. O., Ghosh, A., & Aggarwal, V. (2019). DeepPool: Distributed model-free algorithm for ride-sharing using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems, 2*0(12), 4714–4727. https://doi.org/10.1109/tits.2019.2931830.

18. Li, Z. (2023). Leveraging explainable artificial intelligence and big trip data to understand factors influencing willingness to ridesharing. *Travel Behaviour and Society, 31*, 284–294. https://doi.org/10.1016/j.tbs.2022.12.006.

19. Seng, K. P., Ang, L.-M., Ngharamike, E., & Peter, E. (2023). Ridesharing and crowdsourcing for smart cities: Technologies, paradigms and use cases. *IEEE Access, 11*, 18038–18081. https://doi.org/10.1109/access.2023.3243264.

20. Dessouky, M., & Mahtab, Z. (2022). The ridesharing routing problem with flexible pickup and drop-off points. *eScholarship, University of California*. https://doi.org/10.7922/G2M32T3Z.

21. Meneses-Cime, K., Aksun Guvenc, B., & Guvenc, L. (2022). Optimization of on-demand shared autonomous vehicle deployments utilizing Reinforcement Learning. *Sensors, 22*(21), 8317. https://doi.org/10.3390/s22218317.

22. Haliem, M., Mani, G., Aggarwal, V., & Bhargava, B. (2021). A distributed model-free ride-sharing approach for joint matching, pricing, and dispatching using Deep Reinforcement Learning. *IEEE Transactions on Intelligent Transportation Systems, 22*(12), 7931–7942. https://doi.org/10.1109/tits.2021.3096537.

23. Zheng, Y., Wang, Q., Zhuang, D., Wang, S., & Zhao, J. (2023). Fairness-enhancing deep learning for ride-hailing demand prediction. *IEEE Open Journal of Intelligent Transportation Systems, 4*, 551–569. https://doi.org/10.1109/ojits.2023.3297517.

Information About Authors:

Nurbolat Amilbek is a promising junior scientist and graduate student at Astana IT University (Astana, Kazakhstan, 231986@astanait.edu.kz), specializing in data science, machine learning, and artificial intelligence. As a dedicated researcher, Nurbolat is actively involved in various projects that apply advanced technologies to solve real-world challenges. He is currently pursuing his graduate studies, where his academic focus includes intelligent systems, predictive modeling, and data-driven decision-making. Despite being early in his academic career, Nurbolat has already demonstrated a strong commitment to advancing research in smart technologies and their application to urban development. Through his work at Astana IT University, he aspires to contribute to the future of smart cities and innovative solutions for urban infrastructure.

Beibut Amirgaliyev is a distinguished researcher at Astana IT University (Astana, Kazakhstan, beibut.amirgaliyev@astanait.edu.kz) and is recognized for his contributions to academia and industry. He holds a PhD in Computer Science and serves as a Professor at Astana IT University, focusing on research areas such as machine learning and computer vision. Dr. Amirgaliyev has published numerous papers on automatic number plate recognition and solar collector systems, with his work cited by over 200 researchers. ORCID ID: 0000-0003-0355-5856

Didar Yedilkhan is a distinguished researcher at Astana IT University (Astana, Kazakhstan, d.yedilkhan@astanait.edu.kz), recognized for his extensive experience in industry, research, and higher education. He serves as the Director of the Smart City Research Center and is a Senior Researcher at Astana IT University, focusing on data science, machine learning, and deep learning. Dr. Yedilkhan has a robust academic background, holding degrees from institutions such as the Kazakh National University named after al-Farabi and the University College London. His professional roles include being a Lead Researcher and Project Manager at Astana IT University, where he leads projects on intelligent IT systems for urban infrastructure. His projects aim to enhance city safety and convenience through smart technologies. ORCID ID: 0000-0002-6343-5277

Zharasbek Baishemirov is a lead researcher at Astana IT University (Astana, Kazakhstan, zh.baishemirov@astanait.edu.kz) and Kazakh British Technical University, recognized for his extensive experience in the fields of industry, research, and higher education. He holds a key role as a Lead Researcher at Astana IT University, where his work primarily focuses on mathematical modeling, mathematics, machine learning, and artificial intelligence. Dr. Baishemirov has a strong academic foundation, having completed his studies at leading institutions such as the Abai Kazakh National Pedagogical University and other renowned universities. ORCID ID: 0000-0002-4812-4104

CONTENTS