DEVELOPMENT OF A METHOD FOR PREDICTING THE FOLLOWER SPEED IN CAR-FOLLOWING MODEL

DOI: 10.36724/2072-8735-2026-20-10-62-68

Mikhail G. Gorodnichev,
Moscow Technical University of Communications and Informatics, Moscow, Russia, ***m.g.gorodnichev@mtuci.ru***

Marina V. Yashina,
Moscow Technical University of Communications and Informatics, Moscow, Russia, ***m.v.yashina@mtuci.ru***

Marina S. Moseva,
Moscow Technical University of Communications and Informatics, Moscow, Russia, ***m.s.moseva@mtuci.ru***

Anastasia D. Lipatova,
Moscow Technical University of Communications and Informatics, Moscow, Russia, ***a.d.lipatova@mtuci.ru***

Manuscript received 05 August 2026;
Accepted 07 October 2026

Keywords: *machine learning, traffic flow, car-following model, big data processing, traffic flow modeling, data-driven models*

Amid rapid growth in car ownership, the spread of car-sharing, and the emergence of highly automated vehicles, the strain on road infrastructure continues to increase, leading to traffic congestion, reduced travel efficiency, and rising harmful emissions. Addressing these issues requires more sophisticated traffic flow management, which is achieved through modeling in particular, using microscopic car-following models that describe the behavior of individual drivers based on the actions of the leading vehicle. To collect real-world data, a client-server application has been developed to implement a data-driven car-following model. The paper describes data collection in urban traffic and on highways, as well as data processing. An LSTM-based model was developed to predict the speed of a following vehicle based on time series data that include motion dynamics, interactions with the lead vehicle, and the surrounding environment. The input data included current speed, distance to the lead vehicle, direction of travel, weather conditions (temperature, precipitation), as well as driver characteristics (age, driving experience, gender) and vehicle characteristics (model, year of manufacture). The results of the speed prediction for the following vehicle were obtained based on data about the road user describing its movement, environmental conditions, and the driver's condition. The results demonstrate the model's potential but highlight the need for further improvements in data quality. Moving forward, we plan to expand the dataset to include a wider variety of driving scenarios, add a new source of distance data, and improve the user experience by implementing recommendations for optimal speed and distance.

Information about authors:

Mikhail G. Gorodnichev, *Ph.D. of Engineering Sciences, Associate Professor, Head of the Department, Moscow Technical University of Communications and Informatics, Moscow, Russia. ORCID: 0000-0003-1739-9831*

Marina V. Yashina, *Ph.D. of Engineering Sciences, Professor, Professor of the Department, Moscow Technical University of Communications and Informatics, Moscow, Russia. ORCID: 0000-0001-8810-918X*

Marina S. Moseva, *Ph.D. of Engineering Sciences, Associate Professor of the Department, Moscow Technical University of Communications and Informatics, Moscow, Russia. ORCID: 0000-0002-9778-124X*

Anastasia D. Lipatova, *senior lecturer of the Department, Moscow Technical University of Communications and Informatics, Moscow, Russia. ORCID: 0009-0005-5966-4367*

Для цитирования:
Городничев М.Г., Яшина М.В., Мосева М.С., Липатова А.Д. Разработка метода прогнозирования скорости следования за автомобилем в модели следования за лидером // T-Comm: Телекоммуникации и транспорт. 2026. Том 20. №10. С. 62-68.

For citation:
M.G. Gorodnichev., Yashina M.V., Moseva M.S., Lipatova A.D., "Development of a method for predicting the follower speed in car-following model," *T-Comm*, 2026, vol. 20, no. 10, pp. 62-68.

Intoduction

Today, as car ownership continues to grow at a rapid pace and the number of vehicles on the roads increases year after year, the demand for travel on urban and intercity road networks is also on the rise [1].

In addition to ongoing urbanization, the growth in the number of cars is also influenced by the widespread adoption of car-sharing. Furthermore, highly automated vehicles are emerging [2].

As a result of the growing strain on road infrastructure which often cannot expand at the same rate as the increase in car ownership many cities are facing worsening traffic congestion, a decrease in effective travel time, and higher levels of harmful emissions into the atmosphere, which have a detrimental effect on public health [3].

This growing strain on street and highway networks calls for more effective traffic flow management. Traditional approaches, such as widening roads or optimizing traffic light operations, are proving insufficient. To ensure sustainable and adaptive traffic management, there is a growing reliance on strategies based on data collected in real-world conditions. Thanks to such strategies, it is easier to respond to traffic conditions in real time, anticipate traffic congestion, and ensure safer and more efficient vehicle movement [4].

The strategies described above are implemented through modeling. Among the various types of traffic models, microscopic models that simulate the behavior of individual vehicles are of particular interest. One class of such models consists of leader-follower models [5]. They describe how a driver behaves in various situations depending on the actions of the lead vehicle. These models are essential for understanding and simulating driver behavior on the road.

At present, the taxonomy of leader-following models is still in its early stages of development. In [6], two broad groups are identified: traditional (also known as mathematical) models and data-driven models. This paper will use the classification based on these two broad groups.

Traditional car-following models have been studied since the dawn of automobile traffic, beginning in the 1930s. In turn, data-driven models have become widespread with the development of computing power and methods for big data analysis and machine learning. Each class has certain advantages and limitations, determined by its underlying assumptions, methodological framework, and intended application.

The strength of mathematical models lies in their interpretability. The model parameters have clear physical meanings such as reaction time or the desired safe distance which significantly enhances their applicability to theoretical research [7]. These models do not require significant computational power and are relatively simple to implement, which contributes to their widespread use in traffic simulation programs and real-time control systems.

On the other hand, many traditional models are based on idealized assumptions, such as homogeneous traffic flows, constant driver behavior, and single-lane traffic without interference. In real-world traffic conditions, driver behavior is not so easy to predict and is context-dependent; it is influenced by psychological, weather-related, and situational factors that cannot always be captured using a fixed set of equations [8]. As a result, traditional models are often unable to account for varying road conditions or geographical contexts. Furthermore, the results of such models may depend on calibration.

Among data-driven models, approaches such as artificial neural networks (ANN), long short-term memory (LSTM) networks, convolutional neural networks (CNN), and hybrid CNN-LSTM architectures can be identified.

The main advantage of data-driven models lies in their ability to capture subtle patterns in driver behavior, including reaction delays, inconsistent responses, and adaptation to the surrounding environment. This makes them well-suited for more flexible, scalable, and generalizable modeling [9].

However, these models also have certain limitations. The main problem is their reliance on extensive, high-quality data, which is needed to ensure that the models are trained effectively. Furthermore, deep learning-based models often function as “black boxes,” offering only limited interpretability, and it can be difficult to understand the causal mechanisms underlying their predictions. This can pose challenges when developing applications that require safety guarantees [10].

The choice between traditional and data-driven car-following models depends on specific requirements. Traditional models are characterized by reliability, simplicity, and analytical clarity, making them suitable for fundamental research and traffic engineering. Data-driven models, in turn, can provide flexibility and high forecasting accuracy by taking into account a wider variety of features; however, this makes them suitable for applications requiring detailed modeling of driver behavior only when sufficient data is available. Furthermore, research is ongoing aimed at combining these two classes of models for example, into a hybrid model, as done by the authors of [11].

In real-world traffic, driver behavior is heterogeneous and context-dependent. Reactions to changes in the lead vehicle’s distance and speed vary over time, and environmental factors (weather, road type) and individual driver characteristics lead to nonlinearities and reaction delays. These effects are poorly captured by the fixed equations of traditional models, which reduces their transferability across locations and scenarios.

Therefore, this paper proposes using a Long Short-Term Memory (LSTM) recurrent neural network as the primary method for solving the problem of predicting the speed of a following vehicle. This choice is based on the LSTM’s ability to effectively model long-term temporal dependencies, which are key to understanding the dynamics of a driver’s behavior when following a lead vehicle. The novelty of this work lies in its comprehensive approach: from the development of a custom client-server application for data collection to the application of the LSTM architecture to build a model based on real-world urban traffic conditions.

One of the main challenges of data-driven car-following models is the need to account for temporal dependencies in driver behavior. The decisions a driver makes in a given situation are based not only on the current state of the environment but also depend on previous actions, such as recent changes in speed or the distance to the vehicle ahead. Traditional models, which often rely on snapshot data, cannot fully capture this dynamic context.

To address this problem, the LSTM architecture is typically used a type of recurrent neural network designed to process sequential data. LSTM networks contain memory cells and block-

ing mechanisms that allow the necessary information to be retained at different time steps. This structure makes them well-suited for modeling time series, such as vehicle motion data. Similarly, [11] highlighted the effectiveness of LSTM-based models in capturing the dynamics of a vehicle following a lead vehicle and noted their widespread use in the research community.

However, it is important to note that LSTM is not the only effective solution in this area of modeling. A wide range of alternative architectures, including hybrid CNN-RNN approaches, have shown promising results in the tasks under study. In this work, LSTM is used as a starting point [12].

Among researchers, there are several popular datasets that have proven effective when using machine learning in data-driven models. These include NGSIM [13], highD [14], Waymo [15], and Woven Planet (formerly the Toyota Research Institute – Advanced Development) [16].

Nevertheless, even the datasets listed above have a number of limitations that can negatively affect the predictive power of the trained models. This is discussed, for example, by the researchers in [17]. Here are some of the key points regarding these limitations that the authors highlight in their publication:

• vehicle positioning errors due to insufficient video accuracy when objects overlap, as well as low camera resolution (640x480 pixels). These errors result in unrealistic speed and acceleration values, including sudden spikes that distort the dynamics of motion.

• the trajectories show negative distances between vehicles, which is physically impossible.

• NGSIM covers only short road segments (up to 640 m) and short time intervals (15 to 45 minutes), which limits the generalizability of the results.

In general, datasets are often limited by specific geographic, infrastructural, or weather conditions. They may not provide sufficient diversity in driver behavior. In addition, privacy and access restrictions may prevent their use in certain applications or research. Consequently, the task of independently collecting data under real-world road conditions is of particular importance.

1 Data collection method

As part of this project, a client-server application was developed for data collection. Devices running the Android operating system act as clients. The application's architecture includes a client component installed on the user's smartphone and a server component located on a remote server. The client-side collects data from the device's sensors (coordinates, speed, direction), while the server-side receives this data, calculates the distance between vehicles, stores the information in a database, and provides an API for accessing the data. The architecture of the developed application is shown in Figure 1.

The app's usage scenario involves creating a user profile. Before the trip, you must enter information about yourself; this information will remain anonymous, as you are not required (and there is no option to do so) to provide your name, phone number, or email address. Enter your vehicle's details: model and year [18-22].

The user is then taken to the main screen, where they can control the app's main functions: start data transmission, stop data transmission, and change the server. When the "Start Data Transmission" button is pressed, the screen displays the current coordinates, speed, and status (leader or follower).

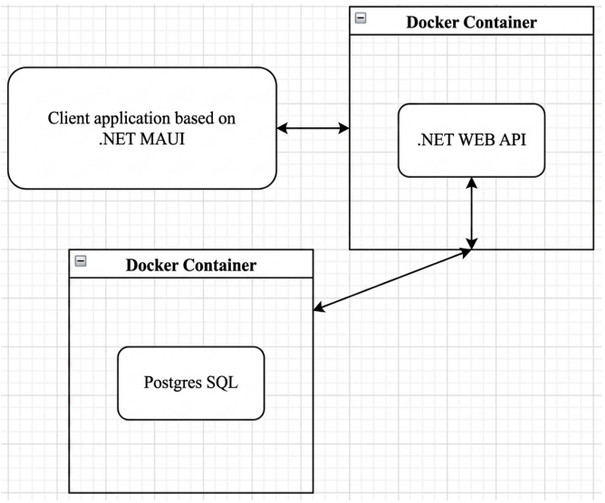


Fig. 1. Architecture of a client-server application

The app begins transmitting the car's position data using the Plugin.Geolocator library. The OnPositionChanged event handler retrieves the current coordinates, speed, and heading angle. Additionally, when new values are received, the information is serialized into a JSON object and sent to the server; this occurs once per second. The generated JSON object is sent to the server via an HTTP POST request. The process for sending weather data works in a similar way, but this occurs once an hour. When the location data for the two cars is received by the server, the distance between them is calculated and the result is recorded in a separate table in the database.

On the server, several controllers have been created, each of which contains several endpoints for writing information to the database or sending it to the client-side. Among these endpoints is /api/MainData, which is responsible for processing POST requests received from the client. In addition, this controller is responsible for automatically determining the leader. Other controllers on the server contain methods that check whether cars are on the same road and whether they are moving in the same direction.

The "Finish Data Transfer" button completes the data transfer process and clears the screen of information displayed from the previous session. In the profile menu, the user can add several more vehicles if the data is being collected by the same person but from a different vehicle.

All information received from the client is recorded via server-side logic into a relational database, where several tables have been created to store information about vehicles, users, general data, and weather data, as well as a table containing the calculated distance.

Data was collected through experimental drives on various roads in the Moscow Region, including both highways and city streets. The dataset included both periods of free-flowing traffic, with vehicles maintaining roughly the same distance from one another and traffic jams. In total, the trips take about 160 minutes.

The processing of the collected data involved several steps. Python libraries such as pandas and numpy were used for this purpose.

Data was standardized to a single format. Each record contained a timestamp in milliseconds. The dataset was then manu-

ally segmented into individual trips, identified by new "trip id" fields. These segments correspond to different data collection sessions, each of which has its own movement patterns.

Several filtering methods were used to eliminate noise in the velocity and position data caused by connection failures. Among these methods was the removal of velocity spikes. Unrealistic velocity spikes were removed using a threshold-based moving average filter.

The beginning and end of trips during which vehicles were either stationary or moving toward a parking lot or gas station were excluded. These time intervals are not relevant for modeling active vehicle behavior and could have introduced noise into the training process.

Some data points contained unrealistic zero-speed values; for example, the car was traveling at 0 km/h at the 55th second, while a speed of 78 km/h was recorded at the 54th and 56th seconds. In such cases, the zero value was considered an error and was replaced with the average of the adjacent values. This type of interpolation helped correct anomalies in the data caused by GPS malfunctions and ensure physical plausibility.

Speed values that were more difficult to identify have been removed: for example, at the 28-second mark, there are 3 entries with a speed of zero for the user and 2 entries with a speed of 57 km/h. In such cases, a non-zero speed was considered the correct value.

There were also instances where, due to an unstable internet connection on mobile devices, the system was unable to calculate the distance between the leader and the follower at certain points in time. These gaps were filled using linear interpolation via the pandas.interpolate() method. This ensured the continuity of the time series and enabled the model to process the data.

The dataset was divided into training, test, and validation sets in a 70/20/10 ratio. The training set was used to train the model and tune its hyperparameters, while the test set was used to evaluate its performance on previously unseen data. The validation set was used for interim evaluation of the model during training. The total dataset consisted of 9,600 records.

As part of the data processing, categorical variables were also coded. User gender was coded as a binary variable: 0 for male, 1 for female. This allowed the model to potentially account for behavioral differences in driving styles.

Driving experience is represented as a numerical value indicating the number of years since obtaining a driver's license, which implies the following: generally speaking, the longer the driving experience, the more experienced the driver. The year the vehicle was manufactured was also included, based on the assumption that newer vehicles may be more responsive to steering.

The car model was encoded using one-hot encoding, which converts car makes and models into binary vectors. This allowed the model to account for the influence of a specific car on driver behavior.

The LSTM-based model was trained to predict the speed of the following vehicle based on a time series of contextual characteristics that reflect both the vehicle's own dynamics and its interaction with the environment and the lead vehicle.

The model consists of a single LSTM layer with 256 hidden neurons, followed by a fully connected layer for predicting the target variable the follower's speed. The model was trained using the Adam optimizer, and the mean squared error (MSE) was used as the loss function. Early stopping was also applied after 5 epochs without improvement, and training was conducted with a batch size of 64 for 20 epochs using a GPU.

To evaluate the effectiveness of the selected LSTM architecture and justify its advantages over other approaches, a comparative analysis was conducted as part of this study. Three main configurations were examined:

1. A ngle-layer LSTM with 256 hidden neurons (base model).
2. A two-layer LSTM with 128 hidden neurons in each layer.
3. A hybrid CNN-LSTM architecture, in which a convolutional layer preprocesses the input sequences to extract local features, and an LSTM layer then models temporal dependencies.

A grid search procedure was used to tune the key hyperparameters for each of these configurations (number of neurons, batch size, learning rate). The grid search space was as follows:

1. Window length W ∈ {5, 10, 20}
2. Number of hidden units H ∈ {64, 128, 256}
3. Dropout ∈ {0.0, 0.2, 0.5}
4. Learning rate l ∈ {1e-2, 5e-3, 1e-3}
5. Ba h size ∈ {32, 64, 128}

The search was carried out using a grid of values, and the RMSE metric on the validation set (extracted from the training set) was used as the performance criterion for selecting the optimal configuration. This made it possible not only to select the best parameters for the base model, but also to objectively compare the performance of different architectures. The final configuration was chosen as a compromise between training accuracy and stability: W=[20], H=[256], dropout=[0.2], lr=[1e-2], batch=[64].

The following were used as input data for the model: current speed, distance to the vehicle in front and direction of travel, as well as environmental conditions such as temperature and precipitation in millimetres. To account for variations in behaviour, the model also incorporated driver characteristics (age, driving experience, gender) and vehicle information (model and year of manufacture). Continuous features were scaled using the MinMaxScaler. A series of input windows, each comprising 10 steps (approximately 10 seconds), were constructed to enable the LSTM to learn temporal dependencies.

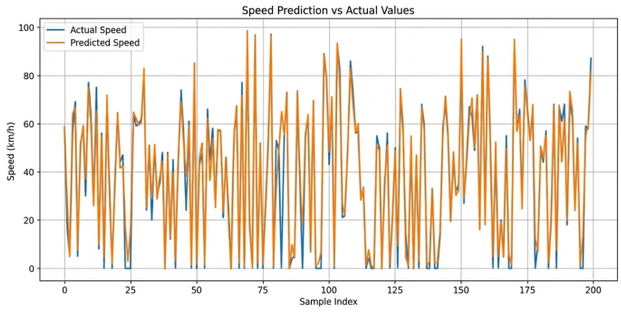


Fig. 2. Learning outcomes

The RMSE metric was used to evaluate the model's performance; based on the training results, the average speed error in the baseline model (a single-layer LSTM with 256 hidden neurons) was 5 km/h. Meanwhile, a two-layer LSTM with 128 hidden neurons in each layer showed an average error of 8.2 km/h. The hybrid CNN-LSTM architecture showed an average error of 6.8 km/h. Thus, the comparison shows that the base model delivered the best results for this specific task. The results of the study are presented graphically in Figure 2.

The figure shows anomalous zero values for the actual velocities, which are attributable to data processing. The collected data require further investigation and the application of additional data processing methods.

A comparison of the results of the proposed solution with other known methods is presented in Table 1.

Table 1

Comparison of results

Method	RMSE
Linear regression	24.82
Polynomial regression	24.41
Gradient boosting	18.74
CNN	19.19
CNN-LSTM	17.01
LSTM (baseline)	11.34
LSTM (our)	**9.12**

Conclusion

The increasing complexity of modern transport systems and the limitations of existing datasets highlight the need to develop specialised data collection methods that ensure the data is up to date for use in modelling.

A client-server application for data collection was proposed and implemented; this enables the retrieval of data on position, speed and direction of movement without the need for specialised equipment, thereby facilitating the reproducibility of the experiment. Three trips were organised to collect data using the developed application, during which a dataset was compiled.

LSTM neural network architecture was selected as the base data-driven model due to its ability to model temporal dependencies and handle non-linearity and volatility. Pre-processing involved synchronising and cleaning the raw data, which enabled the model to be trained.

The results show that the trained model is capable of reproducing the dynamics of a vehicle's behaviour whilst following a lead vehicle with sufficient accuracy. A comparative analysis has shown that the proposed single-layer LSTM architecture is suitable for this dataset, demonstrating the best balance between accuracy and computational efficiency. However, LSTM is not the only model suitable for the task under consideration; the research can be expanded and supplemented by applying other models or combinations of models.

In future work, we plan to expand the dataset to include a wider variety of driving scenarios over longer periods, incorporate an additional source of data on the distance between vehicles, and enhance the user experience of the app by recommending the optimal speed or distance to the vehicle ahead under the given conditions.

Reference

[1] Federal State Statistics Service (Rosstat), "Official website," [Online]. Available: https://rosstat.gov.ru (accessed: 13.05.2025).

[2] R. V. Dushkin, "Mnogoagentnye sistemy dlya kooperativnykh ITS," *Trendy i Upravlenie*, no. 1, pp. 42–50, 2021. DOI: 10.7256/2454-0730.2021.1.34169 (in Russian).

[3] I. A. Bakhirev et al., "Issledovanie peregruzhennosti ulichno-dorozhnoy seti Moskvy s analizom prichin i vliyaniya otlazhennogo sprosa," *Izvestiya Vuzov. Investitsii, Stroitel'stvo, Nedvizhimost'*, vol. 10, no. 4, pp. 634–643, 2020, doi: 10.21285/2227-2917-2020-4-634-643 (in Russian).

[4] A. B. Feroz Khan and I. Perl, "Integrating machine learning and deep learning in smart cities for enhanced traffic congestion management: An empirical review," *Journal of Urban Development and Management*, vol. 2, no. 4, pp. 211–221, 2023, doi: 10.56578/judm020404.

[5] M. G. Gorodnichev, "Informatsionnye i matematicheskie aspekty modeli sledovaniya za liderom," Candidate of Technical Sciences dissertation, Moscow, Russia, 2015 (in Russian).

[6] X. Chen, M. Zhu, K. Chen, P. Wang, H. Lu, H. Zhong, X. Han, and Y. Wang, "Followmet: A comprehensive benchmark for car-following behavior modeling," *Scientific Data*, vol. 10, no. 1, p. 828, 2023, doi: 10.1038/s41597-023-02718-7.

[7] L. Ma, S. Qu, L. Song, Z. Zhang, and J. Ren, "A physics-informed generative car-following model for connected autonomous vehicles," *Entropy,* vol. 25, no. 7, p. 1050, 2023, doi: 10.3390/e25071050.

[8] A. Adewale and C. Lee, "Prediction of car-following behavior of autonomous vehicle and human-driven vehicle based on drivers' memory and cooperation with lead vehicle," *Transportation Research Record*, vol. 2678, no. 6, pp. 248–266, 2024, doi: 10.1177/03611981231195051.

[9] K. Shi, Y. Wu, H. Shi, and B. Ran, "An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network," *Physica A: Statistical Mechanics and its Applications*, vol. 599, p. 127303, 2022, doi: 10.1016/j.physa.2022.127303.

[10] X. Ma et al., "Assessing the impact of car-following driving style on traffic conflict risk using asymmetric behavior model and explainable machine learning," *Accident Analysis & Prevention*, vol. 211, p. 107904, 2025, doi: 10.1016/j.aap.2024.107904. [Online] DOI: 10.1016/j.aap.2024.107904. Available: https://pubmed.ncbi.nlm.nih.gov/39721206/

[11] T. Zhang, P. J. Jin, S. T. McQuade, A. Bayen, and B. Piccoli, "Car-following models: A multidisciplinary review," *IEEE Transactions on Intelligent Vehicles*, vol. 10, no. 1, pp. 92–116, 2024, doi: 10.1109/TIV.2024.3409468.

[12] P. Qin, H. Li, Z. Li, W. Guan, and Y. He, "A CNN-LSTM car-following model considering generalization ability," *Sensors*, vol. 23, no. 2, p. 660, 2023, doi: 10.3390/s23020660.

[13] N. Williams, "NGSIM vehicle trajectory data (US-101)," Kaggle. [Online]. Available: https://www.kaggle.com/datasets/nigelwilliams/ngsim-vehicle-trajectory-data-us-101 (accessed: 13.05.2025).

[14] "HighD: A dataset of naturalistic vehicle trajectories on German highways," *LevelX Data*. [Online]. Available: https://levelxdata.com/highd-dataset/ (accessed: 13.05.2025).

[15] Waymo open dataset: An autonomous driving dataset," Waymo. [Online]. Available: https://waymo.com/open/ (date of access: 13.05.2025).

[16] Toyota Woven Planet, "Woven prediction dataset." [Online]. Available: https://woven.toyota/en/prediction-dataset/ (date of access: 13.05.2025).

[17] Z. He, "Research based on high-fidelity NGSIM vehicle trajectory datasets: A review," *ResearchGate*, pp. 1–33, 2017, doi: 10.13140/RG.2.2.11429.60643/1.

[18] A. Joshi, K. Adugyamfi, J. Merickel, P. Gunaratne, and A. Sharma, "Driver age and its effect on key driving metrics: Insights from dynamic vehicle data," arXiv preprint arXiv:2501.06918, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2501.06918

[19] C. T. Huang, M.-L. Huang, H.-H. Chiang, and C.-H. Lee, "Toward personalized car-following behaviors from driver data using learned and hybrid control strategies," in *Proc. Int. Conf. Syst. Sci. Eng. (ICSSE)*, Hsinchu, Taiwan, 2024, pp. 1–6, doi: 10.1109/ICSSE61472.2024.10608932.

[20] M. S. Moseva, "About methods for collecting and analyzing traffic flow characteristics," *T-Comm*, 2022, vol. 16, no.2, pp. 29-38.

[21] A.S. Bugaev, A.G. Tatashev, M.V. Yashina, O.S. Lavrov, E.A. Nosov, "Reconstruction of traffic flow dynamics based on deterministicstochastic model and data obtained from intelligent transport systems," *T-Comm*, 2019, vol. 13, no.10, pp. 35-44.

[22] A.P. Buslaev, D.A. Kuchelev, M.V. Yashina, "Dynamical systems and mathematical models of information traffic," *T-Comm*, 2018, vol. 12, no.3, pp. 22-38.

РАЗРАБОТКА МЕТОДА ПРОГНОЗИРОВАНИЯ СКОРОСТИ СЛЕДОВАНИЯ ЗА АВТОМОБИЛЕМ В МОДЕЛИ СЛЕДОВАНИЯ ЗА ЛИДЕРОМ

Городничев Михаил Геннадьевич, *Московский технический университет связи и информатики, Москва, Россия,* ***m.g.gorodnichev@mtuci.ru***

Яшина Марина Викторовна, *Московский технический университет связи и информатики, Москва, Россия,* ***m.v.yashina@mtuci.ru***

Мосева Марина Сергеевна, *Московский технический университет связи и информатики, Москва, Россия,* ***m.s.moseva@mtuci.ru***

Липатова Анастасия Дмитриевна, *Московский технический университет связи и информатики, Москва, Россия,* ***a.d.lipatova@mtuci.ru***

Аннотация

В условиях стремительного роста автомобилизации, распространения каршеринга и появления высокоавтоматизированных транспортных средств нагрузка на дорожную инфраструктуру продолжает увеличиваться, приводя к пробкам, снижению эффективности перемещений и росту вредных выбросов. Для решения этих проблем требуется более совершенное управление транспортными потоками, что достигается за счет моделирования, в частности, с использованием микроскопических моделей следования за лидером, описывающих поведение отдельных водителей в зависимости от действий лидирующего транспортного средства. На сегодняшний день развиваются модели на основе данных, способные улавливать сложные закономерности поведения водителей, которые требуют больших объемов данных. Для сбора реальных данных разработано клиент-серверное приложение с целью реализовать модель следования за лидером, основанную на данных. Описан сбор данных в условиях городского движения и на трассах, а также обработка данных. Модель на основе LSTM была разработана для предсказания скорости ведомого автомобиля на основе временных последовательностей, включающих динамику движения, взаимодействие с лидером и окружающей средой. Входные данные содержали текущую скорость, дистанцию до лидера, направление движения, погодные условия (температура, осадки), а также характеристики водителя (возраст, стаж, пол) и автомобиля (модель, год выпуска). Получены результаты предсказания скорости ведомого на основе данных об участнике транспортного движения, описывающих его движение, состояние окружающей среды и описания состояния водителя. Результаты демонстрируют потенциал модели, но указывают на необходимость дальнейшего улучшения качества данных. В дальнейшем планируется расширить данные за счёт более разнообразных сценариев вождения, добавить новый источник данных о дистанции и улучшить пользовательский опыт, внедрив рекомендации по оптимальной скорости и дистанции.

Ключевые слова: *машинное обучение, транспортный поток, модель "следования за лидером", обработка больших данных, моделирование транспортного потока, модели, основанные на данных*

Литература

1. Росстат: Федеральная служба государственной статистики: официальный сайт // URL: https://rosstat.gov.ru (дата обращения: 13.05.2025).

2. *Душкин Р. В.* Многоагентные системы для кооперативных ИТС // Тренды и управление. 2021. №. 1. С. 42-50. DOI: 10.7256/2454-0730.2021.1.34169.

3. *Бахирев И. А.* и др. Исследование перегруженности улично-дорожной сети Москвы с анализом причин и влияния отложенного спроса // Известия вузов. Инвестиции. Строительство. Недвижимость. 2020. Т. 10. №. 4. С. 634-643. DOI: 10.21285/2227-2917-2020-4-634-643.

4. *Feroz Khan A. B., Perl I.* Integrating machine learning and deep learning in smart cities for enhanced traffic congestion management: an empirical review // Journal of Urban Development and Management. 2023. Vol. 2. №. 4. Pp. 211-221. DOI: 10.56578/judm020404.

5. *Городничев М.Г.* Информационные и математические аспекты модели следования за лидером: дис. кандидат технических наук: 13.05.17. Москва, 2015. 120 с. (на русском языке).

6. *Chen X., Zhu.M., Chen K., Wang P., Lu H., Zhong H., Han X., Wang Y.* Follownet: a comprehensive benchmark for car-following behavior modeling // Scientific data. 2023. Vol. 10. №. 1. Pp. 828. DOI: 10.1038/s41597-023-02718-7.

7. *Ma L., Qu S., Song L., Zhang Z., Ren J.* A Physics-Informed Generative Car-Following Model for Connected Autonomous Vehicles // Entropy. 2023. Vol. 25. №. 7. 1050. DOI: 10.3390/e25071050.

8. *Adewale A., Lee C.* Prediction of Car-Following Behavior of Autonomous Vehicle and Human-Driven Vehicle Based on Drivers' Memory and Cooperation With Lead Vehicle // Transportation Research Record Journal of the Transportation Research Board. 2024. Vol. 2678. №. 6. Pp. 248-266. DOI: 10.1177/03611981231195051.

9. *Shi K., Wu Y., Shi H., Ran B.* An integrated car-following and lane changing vehicle trajectory prediction algorithm based on a deep neural network // Physica A: Statistical Mechanics and its Applications. 2022. Vol. 599 (5):127303. DOI: 10.1016/j.physa.2022.127303.

10. *Ma X.* et al. Assessing the impact of car-following driving style on traffic conflict risk using asymmetric behavior model and explainable machine learning // Accident Analysis & Prevention. 2025. Vol. 211(8):107904. DOI: 10.1016/j.aap.2024.107904. URL: https://pubmed.ncbi.nlm.nih.gov/39721206/

11. *Zhang T., Jin P.J., McQuade S.T., Bayen A., Piccoli B.* Car-Following Models: A Multidisciplinary Review // IEEE Transactions on Intelligent Vehicles. 2024. Vol. 10 (1). Pp. 92-116. DOI: 10.1109/TIV.2024.3409468.

12. *Qin P., Li H., Li Z., Guan W., He Y.* A CNN-LSTM Car-Following Model Considering Generalization Ability // Sensors. 2023. Vol. 23(2), 660. DOI: 10.3390/s23020660.

13. *Williams N*. NGSIM Vehicle Trajectory Data (US-101) [Электронный ресурс] // Kaggle. URL: https://www.kaggle.com/datasets/nigelwilliams/ngsim-vehicle-trajectory-data-us-101 (дата обращения: 13.05.2025).

14. HighD - A Dataset of Naturalistic Vehicle Trajectories on German Highways [Электронный ресурс] // LevelX Data. URL: https://levelxdata.com/highd-dataset/ (дата обращения: 13.05.2025).

15. Open Dataset [Waymo Open Dataset: An autonomous driving dataset [Электронный ресурс] // Waymo. URL: https://waymo.com/open/ (дата обращения: 13.05.2025).

16. Toyota Woven Planet. Woven Prediction Dataset [Электронный ресурс]. URL: https://woven.toyota/en/prediction-dataset/ (дата обращения: 13.05.2025).

17. *He Z*. Research based on high-fidelity NGSIM vehicle trajectory datasets: A review // Research Gate. 2017. Pp. 1-33. DOI: 10.13140/RG.2.2.11429.60643/1.

18. *Joshi A., Adugyamfi K., Merickel J., Gunaratne P., Sharma A.* Driver Age and Its Effect on Key Driving Metrics: Insights from Dynamic Vehicle Data // arXiv preprint arXiv:2501.06918. 2025 URL: https://doi.org/10.48550/arXiv.2501.06918

19. *Huang C. T. Huang M. -L., Chiang H. -H., Lee C. -H*. Toward Personalized Car-Following Behaviors From Driver Data Using Learned and Hybrid Control Strategies // 2024 International Conference on System Science and Engineering (ICSSE). Hsinchu, Taiwan, 2024. Pp. 1-6. DOI: 10.1109/ICSSE61472.2024.10608932.

20. *Мосева М.С.* О методах сбора и анализа основных характеристик транспортного потока // T-Comm: Телекоммуникации и транспорт. 2022. Том 16. №2. С. 29-38.

21. *Бугаев А.С., Таташев А.Г., Яшина М.В., Лавров О.С., Носов Е.А.* Восстановление динамики транспортного потока на основе детерминированно-стохастической модели и данных с интеллектуально транспортных систем // T-Comm: Телекоммуникации и транспорт. 2019. Том 13. №10. С. 35-44.

22. *Буслаев А.П., Кучелев Д.А., Яшина М.В.* Динамические системы и математические модели трафика информации // T-Comm: Телекоммуникации и транспорт. 2018. Том 12. №3. С. 22-38.

Информация об авторах:

Городничев Михаил Геннадьевич, *кандидат технических наук, доцент, заведующий кафедрой "Математическая кибернетика и информационные технологии", Московский технический университет связи и информатики, Москва, Россия. 0000-0003-1739-9831*

Яшина Марина Викторовна, *доктор технических наук, профессор, профессор кафедры "Математическая кибернетика и информационные технологии", Московский технический университет связи и информатики, Москва, Россия. 0000-0002-9778-124X*

Мосева Марина Сергеевна, *кандидат технических наук, доцент кафедры "Программная инженерия", Московский технический университет связи и информатики, Москва, Россия. 0000-0002-9778-124X*

Липатова Анастасия Дмитриевна, *Старший преподаватель кафедры "Программная инженерия", Московский технический университет связи и информатики, Москва, Россия. 0009-0005-5966-4367*