

INFORMATION TECHNOLOGY, COMPUTER SCIENCE AND MANAGEMENT ИНФОРМАТИКА, ВЫЧИСЛИТЕЛЬНАЯ ТЕХНИКА И УПРАВЛЕНИЕ



UDC 004.896

Original article

<https://doi.org/10.23947/2687-1653-2023-23-2-169-179>

Predicting the Behavior of Road Users in Rural Areas for Self-Driving Cars

Sergey A. Ivanov  , Bader Rasheed 

Laboratory of Unmanned Technology, Innopolis University, Innopolis, Russian Federation

 se.ivanov@innopolis.ru



Abstract

Introduction. The prediction module generates possible future trajectories of dynamic objects that enables a self-driving vehicle to move safely on public roads. However, all modern prediction methods evaluate their performance only under urban conditions and do not consider their applicability to the domain of rural roads. This work examined the adaptability of existing methods to work under rural unstructured conditions and suggested a new, improved approach.

Materials and Methods. As a solution, we propose to use a neural network that includes the following submodules: a graph-based scene encoder, a multimodal trajectory decoder, and a trajectory filtering module. Another proposed feature is to use an adapted loss function that penalizes the network for generating trajectories that go beyond the drivable area. These elements use standard practices for solving the prediction problem and adapting it to the domain of rural roads.

Results. The presented analysis described the basic features of the prediction module in the rural road domain, showed a comparison of popular models, and discussed its applicability to new conditions. The paper describes the new approach that is more adaptive to the considered domain of study. A simulation of the new domain was performed by modifying existing public datasets.

Discussion and Conclusion. Comparison to other popular methods has shown that the proposed approach provides more accurate results. The disadvantages of the proposed approach were also identified and possible solutions were described.

Keywords: trajectory prediction, behavior prediction, neural networks, self-driving cars, artificial intelligence, autonomous cars

Acknowledgements: the authors appreciate the “Center for Autonomous Technologies”, Innopolis University, for their assistance in conducting the research.

For citation. Ivanov SA, Rasheed B. Predicting the Behavior of Road Users in Rural Areas for Self-Driving Cars. *Advanced Engineering Research (Rostov-on-Don)*. 2023;23(2):169–179. <https://doi.org/10.23947/2687-1653-2023-23-2-169-179>

Прогнозирование поведения участников дорожного движения в условиях проселочных дорог для беспилотных автомобилей

С.А. Иванов , Б. Рашид 

Центр беспилотных технологий университета Иннополис, г. Иннополис, Российская Федерация

✉ se.ivanov@innopolis.ru

Аннотация

Введение. Благодаря модулю прогнозирования траекторий движения динамических объектов беспилотный автомобиль способен безопасно двигаться по дорогам общего пользования. Однако все современные методы прогнозирования оценивают производительность только в городских условиях и не рассматривают свою применимость к домену проселочных дорог. Цель данного исследования заключается в анализе адаптивности существующих методов прогнозирования и разработке подхода, который будет демонстрировать лучшую производительность при работе в новых условиях.

Материалы и методы. В качестве решения предлагается использовать нейронную сеть, включающую в себя следующие подмодули: графовый кодировщик сцены, мультимодальный декодировщик траекторий, модуль фильтрации траекторий. Также предлагается применить адаптированную функцию потерь, которая штрафует сеть за генерацию траекторий, выходящих за границы дорожного полотна. Данные элементы задействуют распространённые практики решения задачи прогнозирования, а также адаптируют её для домена проселочных дорог.

Результаты исследования. Проанализированы основные отличия и условия работы модуля прогнозирования в условиях проселочных дорог. Выполнена симуляция нового домена путем модификации существующих наборов данных. Проведено сравнение популярных методов прогнозирования и оценена их применимость к новым условиям. Представлен новый, более адаптивный к новому домену, подход.

Обсуждение и заключение. Проведенное сравнение с другими популярными методами показывает, что предложенное авторами решение обеспечивает более точные результаты прогнозирования. Также были выявлены недостатки предложенного подхода и описаны возможные пути их устранения.

Ключевые слова: прогнозирование траекторий, прогнозирование поведения, нейронные сети, беспилотные автомобили, искусственный интеллект, автономные автомобили

Благодарности: авторы выражают благодарность центру беспилотных технологий университета Иннополис за помощь в проведении исследования.

Для цитирования. Иванов С.А., Рашид Б. Прогнозирование поведения участников дорожного движения в условиях проселочных дорог для беспилотных автомобилей. *Advanced Engineering Research (Rostov-on-Don)*. 2023;23(2):169–179. <https://doi.org/10.23947/2687-1653-2023-23-2-169-179>

Introduction. The latest achievements in the field of artificial intelligence (AI) are being actively implemented in various areas of activity. One of such achievements is autonomous vehicles (AV). The current research was aimed at creating algorithms that allow AV to move safely on public roads. This will significantly reduce the number of road accidents [1].

The scientific community has already identified the basic modules of an autonomous vehicle. One of them is a system for predicting the future behavior of road users (agents) [2]. A clear understanding of how the environment will develop and in which direction dynamic objects (pedestrians, cars, cyclists) will move is urgently needed for AV to search and use a safe and effective trajectory of movement.

Numerous scientific papers are devoted to the problem of predicting such trajectories [3–12]. However, there is currently no active research on the application of existing methods outside of urban conditions. And this is extremely important, since autonomous cars will be used on country roads, too [13]. Urban conditions are highly structured: cars

mostly follow traffic lanes, and pedestrians move through special zones. In this sense, the area of country roads is the complete opposite, which means that it will have additional difficulties during development. In the given paper, attention is focused on these difficulties: the existing predicting methods and their applicability to new, less structured conditions are considered.

The objective of the study involved:

- analysis of the major differences and working conditions of the prediction module under the conditions of country roads;
- simulation of a less structured country road domain by modifying the existing datasets;
- comparison of modern prediction methods, including their applicability to new conditions;
- description of the new approach and proof of its higher accuracy in comparison to other prediction methods.

Materials and Methods. At first glance, it would seem that the domain presented is a simpler version of urban conditions due to the fact that country roads are characterized by less traffic. However, the absence of complex multi-level junctions, special traffic-free zones, a large number of signs, markings, etc., makes the domain of country roads less structured, i.e., fewer rules and specific traffic patterns increases the randomness and reduces the predictability of the behavior of cars and pedestrians.

The following features of the country road domain influence strongly on the selection of the architecture of the prediction module:

- crossroads. Undoubtedly, they are more simple on country roads in comparison to urban ones, but at the same time this fact of simplicity means that the model must take into account multimodality and assess the probability of choosing each possible direction of movement at the crossroad when the agent approaches it;
- country roads do not have lane markings, pedestrian crossings, bike paths, etc. Instead, the HD map will contain only information about the boundaries of the roadway. Therefore, the stage of encoding the scene should take into account this feature to describe the surrounding context more effectively;
- pedestrians and cyclists will move along the same road with conventional and autonomous vehicles. Therefore, the model should be adaptive for predicting the future trajectories of both cars and pedestrians/cyclists.

The prediction module implies the presence of AV recognition, tracking and localization systems and their accurate operation. The authors of the article use the Argoverse dataset, which stores the required records of the operation of all systems in a convenient form [14].

The dataset consists of recordings of road scenes observed on the streets of Miami and Pittsburgh, USA. Each of the entries contains a local part of the terrain map (lane boundaries, roads, pedestrian crossings) and a list of all recognized agents, including the current position and movement history of each of them. Each of the records is divided into two parts: two seconds of the observation history and three subsequent seconds for which prediction is made (prediction horizon). Data on the future movement of objects is also available and used to calculate the accuracy of prediction methods and model training.

Information about agents is presented in a discrete format. The time interval between measurements is fixed, in this work it is equal to 0.1 seconds (10 Hz).

For each moment of time t , the module receives the observation history S_i^p for each detected agent i . The observation history consists of the agent's current and past states, where each of the states s_t is a 2D position in the global coordinate system. The authors make the assumption that the height information is redundant.

The dataset also provides access to an HD map that contains information about road borders and roadway, pedestrian crossings. To simulate the domain of country roads, the dataset was modified in such a way as to exclude all information from road maps, except for the boundaries of the roadway D . This reduces the amount of information about the road context and complicates the task of prediction.

Hence, the context of the scene is represented as

$$c = \{S_1^p, S_2^p, \dots, S_k^p, D\}, \quad (1)$$

where k — the total number of tracked agents on the scene.

This approach implies predicting the trajectory for only one agent per execution, therefore further S_i^p is treated as S^p for simplification. To generalize the model for all recognized agents, it is required to repeat the proposed approach for all k agents on the scene. The agent for which the prediction is currently being made is considered a target agent.

To assess the accuracy of prediction methods, the dataset contains recorded future trajectories S^f for each target agent:

$$S^f = \{s_1, s_2, \dots, s_H\}, \quad (2)$$

where H indicates the number of next time steps. In this case, parameter H will be equal to 30, since the planning horizon is three seconds with a sampling frequency of 10 Hz.

The domain of the prediction module is multimodal, i.e., the future behavior of agents may differ significantly in absolutely identical traffic situations. Let us say, a car approaching a crossroad may continue straight ahead or make a turn. To take this into account, it is required to generate M possible future trajectories and M probabilities of each of them at the model output.

Therefore, the purpose of the prediction module is to create function f , that takes the context of the scene c as input and generates M pairs of possible future trajectories and their probabilities:

$$f(c) = \{S_1^f, S_2^f, \dots, S_M^f, p_1, p_2, \dots, p_M\}. \quad (3)$$

Here, at least one generated trajectory S^f should be as close as possible to the real trajectory S^f , and the probability of its execution p should be close to unity.

Model architecture. The proposed approach involves the use of a neural network consisting of submodules of scene encoding, decoding and filtering trajectories. The architecture of the system is shown in Figure 1.

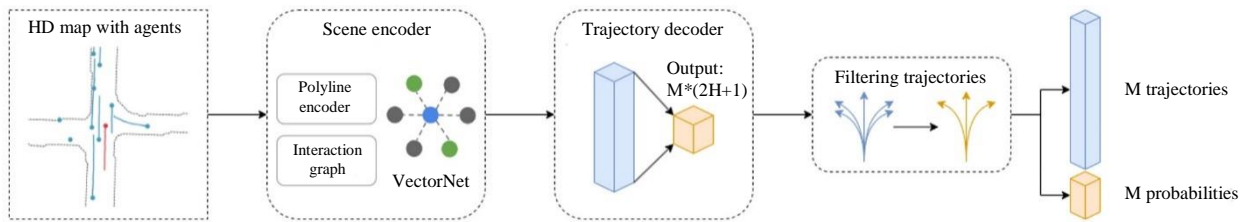


Fig. 1. System architecture

A neural network adapted to the new conditions, based on a vector representation, is responsible for encoding the scene. This selection is due to the fact that on country roads, the HD map will contain a limited amount of information (only the roadway boundaries and the history of observations of dynamic objects). Popular methods represent the context of a road scene c in an image format and process it using convolutional neural networks. However, vector coding avoids the overhead associated with image generation [4–5].

The presented encoder is based on the VectorNet model, but its input data format has been modified to receive information only about the boundaries of the roadway and the state of the agents. [3]. This encoder represents the boundaries of the road and the state of agents using polylines, which are further processed by a graph neural network. This provides encoding the interaction between polylines. Details of the implementation are described in paper [3].

A trajectory decoder is a task of regressing several possible trajectories and generating a set of probabilities. To solve this problem, a multilayer perceptron model is used. The decoder implementation is inspired by the MTP model [4], however, the authors of the article propose a different formula for calculating the best trajectory m^* from the set of M trajectories. It is also proposed to use an additional mechanism that penalizes the model for predictions that go beyond the area of movement.

The authors of the original MTP model propose to train a multilayer perceptron using the loss function that represents the sum L_{reg} and L_{class} , where:

$$L_{reg} = L(S^f, S_m^f) \quad (1)$$

$$L_{class} = -\sum_{m=1}^M I_{m=m^*} \log p_m \quad (5)$$

In this case, L_{reg} — mean-square error between the real trajectory S^f and the best trajectory m^* of M generated.

$$L(S^f, S_{m^*}^f) = \frac{1}{H} \sum_{i=1}^H (s_i - s_i^*)^2, \quad (6)$$

where s_i — the agent's actual future position at time i , and \tilde{s}_i^* — the predicted future state of the best trajectory m^* .

L_{class} — loss function based on cross entropy, which increases the probability of executing the best of the predicted trajectories m^* to 1 and reduces the probability of other trajectories to 0.

I_c is a binary indicator equal to 1, if condition c is true, and 0 — otherwise.

In the original article, the best of the predicted trajectories m^* is defined as the one that has the minimum value of the root-mean-square error in comparison to the real trajectory:

$$m = \underset{m}{\operatorname{argmin}} L(S^f, S_m^f). \quad (7)$$

The authors of the article suggest using the following modification:

$$m = \underset{m \in \Delta}{\operatorname{argmin}} L(S^f, S_m^f), \quad (8)$$

where Δ — subset of generated trajectories that has a similar final direction to the real trajectory S^f .

The idea is to remove from consideration trajectories in which the final direction of the agent differs significantly from the direction in the real trajectory when calculating the best trajectory m^* . If the difference in directions features less than certain threshold γ , then the generated trajectory is considered correct, i.e., $m \in \Delta$. In the case under consideration $\gamma=30^\circ$. Therefore, the best trajectory m^* should have a similar final direction and the lowest value of the loss function.

This work also involves prior knowledge of the domain to achieve greater convergence of the model [15]. Since only information about the roadway boundaries is available from the HD map when driving in the domain of country roads, an additional variable is introduced — L_{da} into the loss function. Thanks to it, the model will penalize the predicted trajectories that go beyond the road in cases where at least one state is $s_i \notin D$. The model penalizes only the best trajectory, since only in this case, it is possible to determine the direction of error reduction by approximating the best of the generated trajectories m^* to the real trajectory S^f .

Thus, L_{da} is defined as:

$$L_{da} = \frac{1}{H} \sum_{i=1}^H I_c \cdot (s_i - s_i^*)^2, \quad (9)$$

where I_c is equal to 1, if $s_i \notin D$, and 0 — otherwise.

The final loss function is defined as

$$L = L_{class} + \alpha \cdot L_{reg} + \beta \cdot L_{da}, \quad (10)$$

where α and β — neural network hyperparameters used for training. In this case, both of these parameters are equal to 0.5.

To filter similar and duplicate trajectories, the proposed approach uses the filtering of a finite set of trajectories M at the final stage. This module is required because in some cases, the number of possible agent trajectories may be less than M , e.g., when a car is moving along a straight road at a constant speed, the model can generate only one trajectory: the car continues to move straight. However, the need to generate exactly M trajectories will result in the situation when all predictions are similar to each other.

The proposed filtering is based on the final direction and positions of states s_i : if the direction and the sum of the deviations between states s_i of the real and generated trajectories are less than the threshold value σ , then the trajectories are considered similar. The authors average each state of the trajectories and sum up the probabilities of the trajectories p_i .

This approach was implemented in the Python programming language on the PyTorch deep learning framework. The model was trained on GeForce RTX 2080 Ti graphics card for 40 epochs, the training took four hours.

Research Results. To assess the accuracy of prediction models, this section applies widely used metrics for the trajectory prediction problem: average displacement error, ADE, final displacement error, FDE [6], MissRate (MR), and Offroad rate (OR).

For multimodal cases with the generation of several trajectories, ADE and FDE are taken as the minimum ADE and FDE among M trajectories (the trajectory with the lowest metric value) [5].

The prediction is considered “missed” if ADE metric of the generated trajectory is more than two meters. OR metric is calculated as the percentage of trajectories in which at least one state s_i goes beyond the range of motion D .

To visualize the context of scene c , as well as the real and predicted future trajectories S^f and S^f a script in the Python programming language was implemented using the Matplotlib library.

This section compared the operation of several different methods in the case of an unstructured domain. The following methods were used in comparison:

- Kalman filter;
- Single trajectory output – the proposed scene encoder with the generation of a single trajectory;
- Fixed set classification – the proposed scene encoder with the reduction of the task to classification among predefined trajectories: by sets of 64 and 415 predefined trajectories;
- Proposed approach.

Table 1 presents comparison of the accuracy of the methods when working under unstructured conditions. Several methods are compared, including the proposed approach.

Table 1

Comparison of models in the unstructured domain of work

Method	Modes	ADE ₁	FDE ₁	ADE ₆	FDE ₆	MR ₂ ₁	MR ₂ ₆	OR
Kalman filter	1	3.78	8.05	3.78	8.05	0.89	0.89	5.89
Single trajectory output	1	3.12	6.75	3.12	6.75	0.89	0.89	3.26
Fixed set classification	415	3.27	7.00	1.74	3.57	0.84	0.52	3.61
Fixed set classification	64	2.6	5.63	1.52	2.91	0.82	0.49	2.58
Proposed approach	6	2.36	5.29	1.32	2.55	0.78	0.38	1.84

Kalman filter. The simplest way to predict behavior is to obtain the current state of the object (current lane, speed, direction, etc.) and extend this state to future steps based on some assumptions, e.g., that the car will continue to follow its lane or will have a constant speed and/or acceleration. Another popular method for such tasks is to use the Kalman filter [12].

According to Table 1, the Kalman filter works worse than all the presented methods based on neural networks.

Figure 2 shows two cases. In the first case, the Kalman filter successfully performs prediction because the vehicle is moving straight, without any turns or speed variation. In the second case, the Kalman filter mispredicts due to lack of knowledge about the context of the traffic situation.

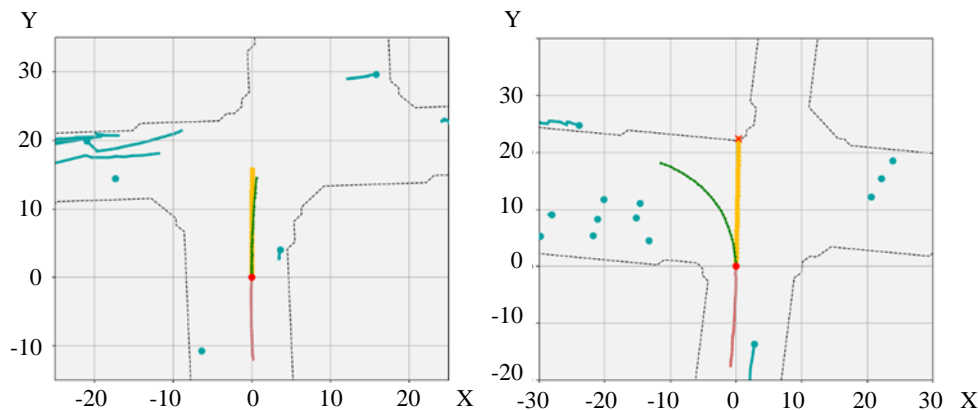


Fig. 2. Example of predictions using the Kalman filter. Dotted lines — roadway boundaries, red lines — target agent with history of observations, blue — other agents, green — real trajectory, yellow — predicted trajectory, red crosses indicate predicted states outside the roadway

Single trajectory output. This method involves the use of a graph scene encoder, which is identical to the one used in the proposed approach. The output of the network implies the generation of only one trajectory. This model is trained using the root-mean-square loss function.

As shown in Table 1, the neural network, even with the generation of a single trajectory, demonstrates better results in comparison to the Kalman filter.

Figure 3 shows the visualization of this prediction method operation. The image on the left shows that the model can successfully predict the agent's turn. The image on the right shows that generating one trajectory is not enough. The neural network tries to imagine both possible outcomes: going straight and turning right. As a result, the model outputs the average of the two outcomes.

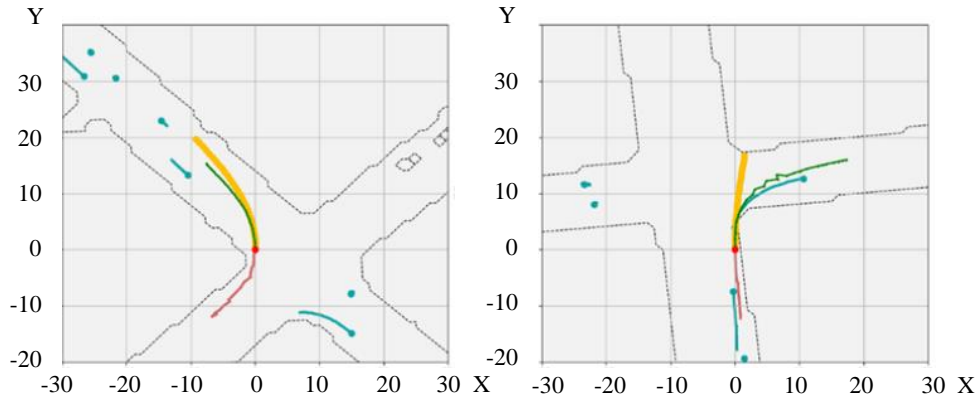


Fig. 3. Example of generating a single trajectory. Red line shows the target agent with history of observations, green — real trajectory of movement, yellow — predicted trajectory

Fixed set classification. The implementation was inspired by the CoverNet prediction method [5]. This model consists of a proposed vector scene encoder, followed by a different trajectory decoder. The decoder is a classification task based on a predefined set of trajectories consisting of physically realizable vehicle trajectories with sufficient coverage. Two sets were created for experiments: of 415 and 64 possible trajectories. The second set has the same coverage as the first, but provides a lower density of trajectories. Detailed information about the sets of trajectories is contained in paper [5].

The visualization of the work is shown in Figure 4. The classification model successfully copes with multimodality at crossroads, but in some cases, the lack of sufficient coverage by a set of trajectories negatively affects the results.

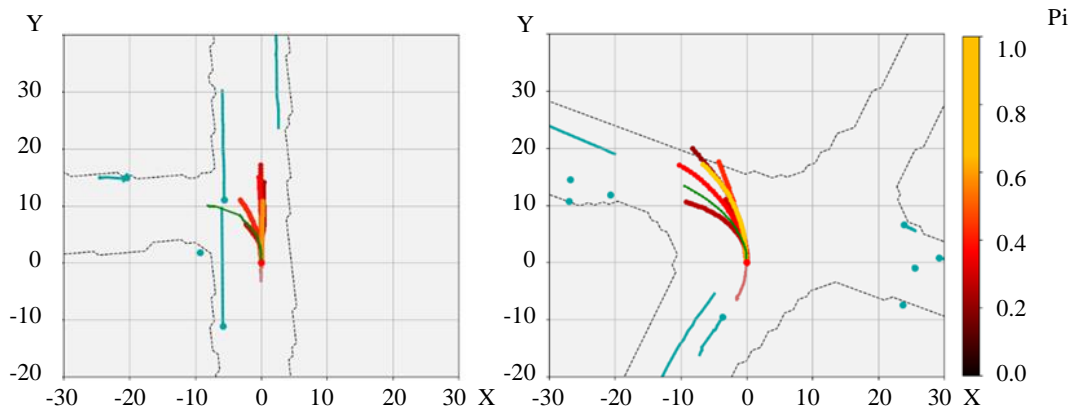


Fig. 4. Example of prediction using a classification model. Red lines represent the target agent with history of observations, green lines — real trajectory. M predicted trajectories with different probability of execution p_i are presented using red-yellow hues

As shown in Table 1, this method works more accurately than generating a single trajectory, but worse than the proposed approach. In addition, increasing the density of the set of trajectories by using a set of 415 trajectories did not improve the results. The authors attribute this to the presence of noise in the dataset, which comes from the tracking system used in the data collection.

Proposed approach. The proposed approach eliminates the disadvantages of all the methods described above. This is a multimodal forecasting method that does not suffer from the limitations of a predefined set of trajectories.

Moreover, according to Table 1, the proposed approach surpasses all other methods in all indicators. As shown in Figure 5, the method successfully captures two possible outcomes at the crossroad: driving straight or making a turn.

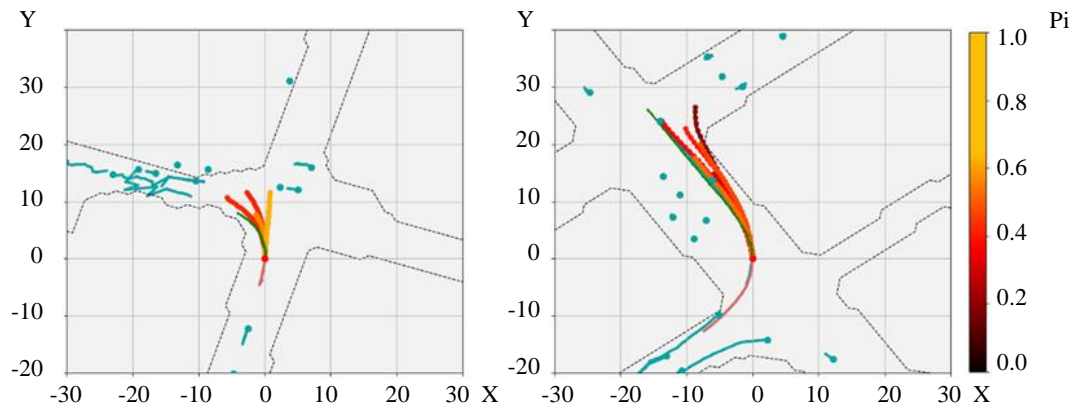


Fig. 5. Example of prediction using a classification model. Red lines represent the target agent with history of observations, green lines — real trajectory. M predicted trajectories with different probability of execution p_i are presented using red-yellow hues

Figure 6 shows an example of filtering similar trajectories in the case of a single possible outcome. The probability that the agent will complete the initiated turn is close to 1, since he is already in the process of turning. Therefore, in this case, the probability of other outcomes is close to 0. The proposed module successfully filters similar trajectories.

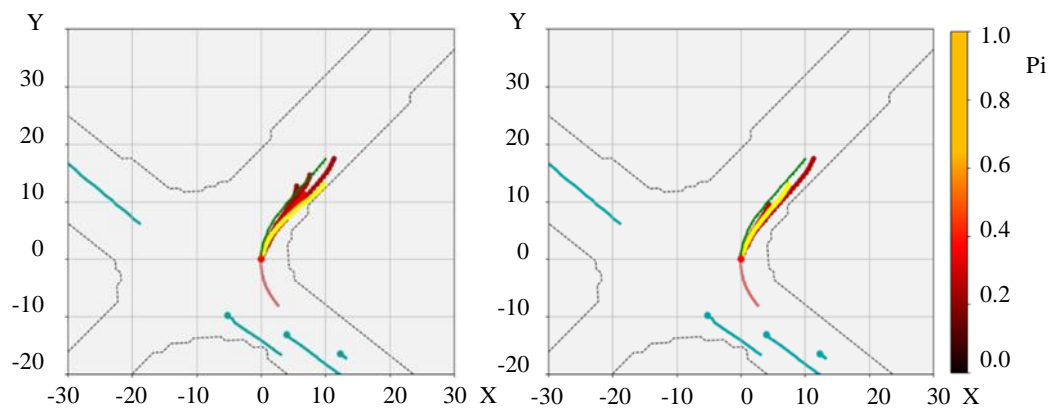


Fig. 6. Filtering effect. The entire set of predictions is shown on the left, and only filtered set — on the right

Limitations. Although the authors of the original article on the MTP model [4] indicate that their method solves the problem of mode collapse, the experiments conducted by the authors of this article do not confirm this. The problem still occurs in some cases. It is assumed to be due to the following features: the loss function does not penalize the neural network for generating all possible trajectories that the target agent can execute, as long as the best of them is as close as possible to the real trajectory. But also, the model does not encourage the network in any way to predict a variety of possible trajectories. Therefore, it is advantageous for the network to make several similar predictions in one direction, in which it is more confident than to make one prediction for each possible trajectory.

One of the possible solutions to this problem may be the use of a trajectory decoder presented in the TnT, DenseTnT models [10–11], which imply the generation of final goals at the first stages of work. In these models, all possible final goals for the agent are generated first, and then trajectories that describe the movement from the starting position to each of the goals, are generated. This provides filtering out similar final goals in the early stages, and preventing the mode collapse.

Discussion and Conclusion. In the work performed, modern methods of solving the trajectory prediction problem are investigated. The adaptability of the methods to unstructured road conditions — country roads, is considered. Insufficient accuracy of the methods is established, and a new approach to predicting is proposed.

The proposed approach is based on the VectorNet and MTP models, but has been adapted for the country road domain. In addition, a trajectory filtering module and an additional mechanism for the loss function, which penalizes trajectories for going beyond the movement zone, are proposed.

The presented comparison shows that the proposed approach is superior to other popular methods.

Limitations of the MTP approach have been identified: the output data still tends to mode collapse. The suggestion for further modifications is to use methods that generate the final goal at the early stages of prediction and thus are less susceptible to regime collapse.

References

1. Qing Rao, Jelena Frtunikj. *Deep Learning for Self-Driving Cars: Chances and Challenges*. In: Proc. 1st International Workshop on Software Engineering for AI in Autonomous Systems. New York, NY: Association for Computing Machinery; 2018. P. 35–38. <https://doi.org/10.1145/3194085.3194087>
2. Shaoshan Liu, Liyun Li, Jie Tang, et al. *Creating Autonomous Vehicle Systems*. San Rafael, CA: Morgan & Claypool; 2020. 216 p.
3. Jiyang Gao, Chen Sun, Hang Zhao, et al. *VectorNet: Encoding HD Maps and Agent Dynamics from Vectorized Representation*. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA: IEEE; 2020. P. 11525–11533. <https://doi.org/10.48550/arXiv.2005.04259>
4. Henggang Cui, Vladan Radosavljevic, Fang-Chieh Chou, et al. *Multimodal Trajectory Predictions for Autonomous Driving Using Deep Convolutional Networks*. In: Proc. IEEE International Conference on Robotics and Automation (ICRA). Montreal, BC: IEEE; 2019. P. 2090–2096. <https://doi.org/10.48550/arXiv.1809.10732>
5. Tung Phan-Minh, Elena Corina Grigore, Freddy A. Boulton, et al. *CoverNet: Multimodal Behavior Prediction Using Trajectory Sets*. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA: IEEE; 2020. P. 14074–14083. <https://doi.org/10.48550/arXiv.1911.10298>
6. Abdullah Mohamed, Kun Qian, Mohamed Elhoseiny, et al. *Social-STGCNN: A Social Spatio-Temporal Graph Convolutional Neural Network for Human Trajectory Prediction*. In: Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition. Seattle, WA: IEEE; 2020. P. 14424–14432. <https://doi.org/10.48550/arXiv.2002.11927>
7. Biktairov Yu, Stebelev M, Rudenko I, et al. *PRANK: Motion Prediction Based on RANKing*. In: Neural Information Processing Systems. Vancouver: Virtual Conference; 2020. P. 2553–2563. <https://doi.org/10.48550/arXiv.2010.12007>
8. Yuning Chai, Benjamin Sapp, Mayank Bansal, et al. MultiPath: Multiple Probabilistic Anchor Trajectory Hypotheses for Behavior Prediction. *Proceedings of the Conference on Robot Learning*. 2020;100:86–99. <https://doi.org/10.48550/arXiv.1910.05449>
9. Ajay Jain, Sergio Casas, Renjie Liao, et al. Discrete Residual Flow for Probabilistic Pedestrian Behavior Prediction. In: Proc. 3rd Conference on Robot Learning, Osaka, Japan, 2019. *Proceedings of Machine Learning Research*. 2019;100:407–419. <https://doi.org/10.48550/arXiv.1910.08041>
10. Hang Zhao, Jiyang Gao, Tian Lan, et al. *TNT: Target-driveN Trajectory Prediction*. In: Conference on Robot Learning. Cambridge, MA: Virtual Conference; 2020. P. 895–904. <https://doi.org/10.48550/arXiv.2008.08294>
11. Junru Gu, Chen Sun, Hang Zhao. *Dense TNT: End-to-end Trajectory Prediction from Dense Goal Sets*. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. Montreal, BC: IEEE; 2021. P. 15303–15312. <https://doi.org/10.48550/arXiv.2108.09640>

12. Prévost CG, Desbiens A, Gagnon E. *Extended Kalman Filter for State Estimation and Trajectory Prediction of a Moving Object Detected by an Unmanned Aerial Vehicle*. In: Proc. American Control Conference. New York, NY: IEEE; 2007. P. 1805–1810. <https://doi.org/10.1109/ACC.2007.4282823>
13. Zeyu Zhu, Nan Li, Ruoyu Sun, et al. *Off-road Autonomous Vehicles Traversability Analysis and Trajectory Planning Based on Deep Inverse Reinforcement Learning*. In: IEEE Intelligent Vehicles Symposium (IV). Las Vegas, NV: IEEE; 2020. P. 971–977. <https://doi.org/10.1109/IV47402.2020.9304721>
14. Mig-Fang Chang, John Lambert, Patsorn Sangkloy, et al. *Argoverse: 3D Tracking and Forecasting with Rich Maps*. In: Proc. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA: IEEE; 2019. P. 8748–8757. <https://doi.org/10.1109/CVPR.2019.00895>
15. Casas S, Gulino C, Suo S, et al. *The Importance of Prior Knowledge in Precise Multimodal Prediction*. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Las Vegas, NV: IEEE; 2020. P. 2295–2302. <https://doi.org/10.48550/arXiv.2006.02636>

Received 09.04.2023

Revised 28.04.2023

Accepted 10.05.2023

About the Authors:

Sergey A. Ivanov, Senior Engineer, Center for Autonomous Technologies, Innopolis University (1, Universitetskaya St., Innopolis, 420500, RF), [ORCID](#), se.ivanov@innopolis.ru

Bader Rasheed, Head of the Recognition Systems Development Department, Center for Autonomous Technologies, Innopolis University (1, Universitetskaya St., Innopolis, 420500, RF), [ResearcherID](#), [ScopusID](#), [ORCID](#), b.rasheed@innopolis.university

Claimed contributorship:

SA Ivanov: basic concept formulation; research objective and tasks; computational analysis; text preparation; formulation of conclusions.

B Rasheed: academic advising; analysis of the research results; hypotheses advancement; revision of the text.

Conflict of interest statement: the authors do not have any conflict of interest.

All authors have read and approved the final manuscript.

Поступила в редакцию 09.04.2023

Поступила после рецензирования 28.04.2023

Принята к публикации 10.05.2023

Об авторах:

Сергей Александрович Иванов, старший инженер центра беспилотных технологий университета Иннополис (420500, РФ, г. Иннополис, ул. Университетская, 1), [ORCID](#), se.ivanov@innopolis.ru

Бадер Рашид, руководитель отдела разработки систем распознавания центра беспилотных технологий университета Иннополис (420500, РФ, г. Иннополис, ул. Университетская, 1), [ResearcherID](#), [ScopusID](#), [ORCID](#), b.rasheed@innopolis.university

Заявленный вклад соавторов:

С.А. Иванов — формирование основной концепции, цели и задачи исследования, проведение расчетов, подготовка текста, формирование выводов.

Б. Рашид — научное руководство, анализ результатов исследований, выдвижение возможных гипотез, доработка текста.

Конфликт интересов: авторы заявляют об отсутствии конфликта интересов.

Все авторы прочитали и одобрили окончательный вариант рукописи.