Model Controlled Prediction: A Reciprocal Alternative of Model Predictive Control

Citation for the original published paper (version of record):

Li, S., Liu, Y., Qu, X. (2022). Model Controlled Prediction: A Reciprocal Alternative of Model Predictive Control. IEEE/CAA Journal of Automatica Sinica, 9(6): 1107-1110.
http://dx.doi.org/10.1109/JAS.2022.105611

Letter

Model Controlled Prediction: A Reciprocal Alternative of Model Predictive Control

Shen Li, Yang Liu, and Xiaobo Qu, *Senior Member, IEEE*

Dear editor,

This letter presents a reciprocal alternative to model predictive control (MPC), called model controlled prediction. More specifically, in order to integrate dynamic control signals into the transportation prediction models, a new fundamental theory of machine learning based prediction models is proposed. The model can not only learn potential patterns from historical data, but also make optimal predictions based on dynamic external control signals. The model can be used in two typical scenarios: 1) For low real-time control signals (e.g., subway timetable), we use a transfer learning method, so that the prediction models obtained from training data under the old control strategy can be predicted accurately under the new control strategy. 2) For dynamic control signals with high real-time (e.g., online ride-hailing dispatching instructions), we establish a simulation environment, design a control algorithm based on reinforcement learning (RL), and then let the model learn the mapping relationship among dynamic control signals, data, and output in the simulation environment. The experimental results show that the reasonable modeling of control signals can significantly improve the performance of the traffic prediction model.

Unprecedented urbanization has led to the expansion of urban size and density. In order to meet the challenges of mobility and sustainability, more accurate transportation prediction (e.g., passenger flow prediction in public transport systems, spatio-temporal supply-demand prediction in ride-hailing services) is essential to guide the design, planning, operations, and control of urban transportation systems.

Numerous transportation prediction methods based on artificial intelligence (AI) techniques have emerged, such as long short-term memory network (LSTM) [1], convolution neural network (CNN) [2] and graph convolution neural network (GCN) [3]. Unfortunately, most intelligent transportation systems (ITS) are affected by external control signals (e.g., intersection signal timing, metro timetables, online ride-hailing dispatching instructions), while existing traffic prediction methods only mine potential patterns from historical data without introducing control to form a closed loop. Prediction models based on historical data tend to fail or perform poorly as external control signals change.

In order to resolve this critical issue, substantial efforts are conducted to consolidate control and prediction, which are inextricably connected. Therefore, the well-known model predictive control has developed vigorously in enhancing the performance of optimal control, and it has also been applied in numerous traffic control problems [4], but it can not deal with other aspects of transport problems including operations, design, and planning. In the field of ITS, considering the influence of external control signals, the reciprocal alternative of model predictive control has more important and extensive value and has the potential to be applied to all aspects of ITS research.

Corresponding author: Yang Liu.

Citation: S. Li, Y. Liu, and X. B. Qu, "Model controlled prediction: A reciprocal alternative of model predictive control," *IEEE/CAA J. Autom. Sinica*, 2022, DOI: 10.1109/JAS.2022.105611

S. Li is with the Department of Civil Engineering, Tsinghua University, Beijing 100084, China (e-mail: sli299@tsinghua.edu.cn).

Y. Liu is with the Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg SE-41296, Sweden (e-mail: liuy@chalmers.se).

X. B. Qu is with the School of Vehicle and Mobility, Tsinghua University, Beijing 100084, China (e-mail: xiaobo@tsinghua.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JAS.2022.105611

Related work: Transport engineering is increasingly interdisciplinary with automatic control, AI, and many other emerging areas of information science which form the core of new ITS technology [5]. Traffic control and prediction are two important pillars of ITS research.

A representative example in the field of traffic control is intersection signal control, which aims to minimize vehicle travel time by coordinating vehicle movements at road intersections. Since signalized intersection is the bottleneck of urban traffic, effective signal control will reduce traffic congestion [4]. Another example is the railway timetabling control, which has proved to be an NP-Hard problem [6]. Its offline optimization objectives include train travel time [7], total energy consumption [8], transfer waiting time [9], etc. The existing research mainly focuses on mathematical programming [10]. So far, most real-world traffic control strategies are based on offline data optimization, while online rolling optimization has not been implemented. This is due to the complexity and scale of real-world traffic problems, making it difficult to meet the real-time requirements using mathematical programming or heuristic methods. RL has the potential to address this challenge, and few studies have attempted to solve complex large-scale dynamic optimization problems in the ITS field, such as traffic signal control [11] and online ride-hailing fleet management control [12].

Since the emergence of AI and the development of data collection techniques, the application of AI in transportation prediction has affected all aspects of ITS [13]. For example, accurate passenger flow prediction not only helps passengers make better decisions by adjusting their travel routes and departure times, but also helps transit operators optimize train timetables and save operating costs [14]. Spatio-temporal data prediction is another core issue, accurately predicting future spatio-temporal supply and demand can help improve traffic conditions, fleet organization, utilization rate, and social welfare. A large number of spatio-temporal data prediction methods based on artificial intelligence techniques have been proposed and applied. Existing state-of-art research is to transform the traffic prediction problem into a regression problem in machine learning. However, these typical traffic prediction problems are affected by the above control signals, but so far, none of these algorithms consider dynamic external control signals. Therefore, it is necessary to develop a new fundamental theory of AI-driven prediction model considering dynamic control.

The fundamental theory of model controlled prediction: As mentioned in the related work, there is no research on integrating dynamic control signals into traffic prediction models. In order to fill the research gap, we will solve three basic scientific research questions.

Q1: Why is Model Predictive Control not applicable to many ITS studies?

Q2: What are the flaws of existing traffic prediction methods compared with model predictive control?

Q3: How can we deal with the flaws in Q2?

Fig. 1 is the illustration of model predictive control. Through Fig. 1, we can analyze and answer Q1 systematically.

The main reasons limiting the application of model predictive control in ITS are:

- The measurement step in ITS has not been completely solved. It is a challenging task to obtain the travel data of millions of residents in a megacity. In the era of big data and high resolution, the ITS field has only solved very preliminary data acquisition problems. For example, in the bus system, swipe cards in most cities only record the pick-up station, missing the drop-off station. As a result, the measurement step has not yet been completely solved.
- The computational cost of implementing online rolling opti-

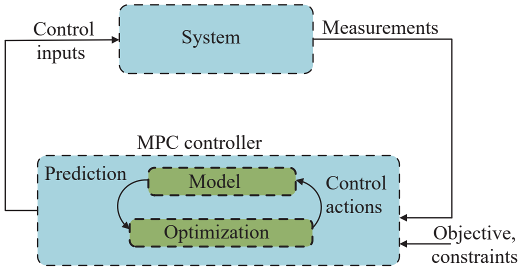

Fig. 1. Illustration of model predictive control.

mization in ITS is high. Most ITS studies are large-scale and complex (such as optimizing the timetable of the entire city subway line), which are computationally expensive. As a result, these problems are usually optimized offline, and they are difficult to optimize online on a rolling basis.

- As discussed earlier, ITS systems have numerous applications not only in control, but also in operating, designing, and planning. Compared with transportation prediction, model predictive control has not been able to fully satisfy the diverse requirements of ITS systems, which further limits its wide implementation in ITS.

For Q2, it should be noted that in the existing data-driven traffic prediction models, only data (e.g., dividing training set/test set, data preprocessing, feature engineering), models, and tasks are considered in the modeling process (as shown in Fig. 2), without proper consideration and reflection of the system and optimization.

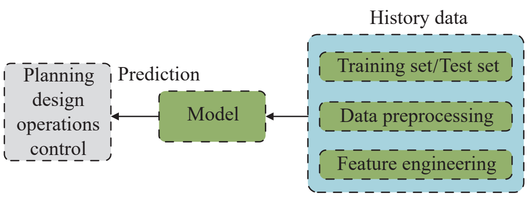

Fig. 2. Illustration of the modeling process of the existing data-driven traffic prediction model.

Referring to the illustration of model predictive control, we complete Fig. 2 by adding components such as system and optimization. To distinguish from Figs. 1 and 2, the structure in Fig. 3 is called "control-prediction". Note that Fig. 3 is a presentation of existing method in the form of model predictive control illustration, where the existing method is flawed. In Fig. 3 , although the data-driven model can implicitly learn weak information about external control signals from a large amount of historical data, it is far from sufficient because of the model's fragility and the inability to respond quickly when external signals change if the model fails to explicitly learn external signals.

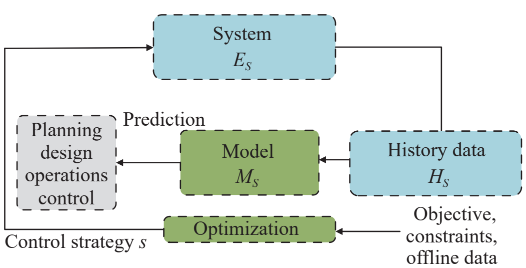

Fig. 3. Illustration of "control-prediction" structure in ITS.

The focus of this paper is on explicitly learning external control signals, and below we briefly analyze the impact of external signals on existing data-driven methods.

The control strategy in Fig. 3 is a time sequence composed of several control signals.

$$s = \{c_1, c_2, \dots, c_k\} \tag{1}$$

where c_k represents the control signal at time k, e.g., control a subway departure at 6 a.m. from the starting station, or send a control command at 8 a.m. to dispatch the vehicle from grid a to grid b.

As shown in Fig. 3 , assuming that we have obtained a control strategy s by optimization based on the current environment. Under the control strategy s, the system (denoted as E_s) will continuously accumulate historical data (denoted as H_s). Based on H_s, we can train a supervised machine learning model (denoted as M_s) to predict future states of the system E_s. The prediction results can be used for several key tasks (e.g., operations, design, planning, etc.). Compared with model predictive control, since optimization and model are considered separately, once the conditions change (e.g., objective functions, constraints, or offline data changes), a new control signal t will be obtained. At this point, the system turns into E_t, and the data distribution will also change, so using the model M_s to predict the system E_t will not be optimal.

For Q3, the shortcomings of existing methods can be overcome inspired by ideas of model predictive control. As the optimization process in Fig. 3 is a large-scale offline optimization, the control strategy does not change particularly frequently (i.e., low real-time). Therefore, we can use the transfer learning methods for model transfer with the help of control strategy. The model M_s can accurately predict the system E_t even if the training data does not contain data from the system E_t. This situation corresponds to the Scenario 1 below. Note that in Fig. 3 there is no direct link between the optimization component and the model component, while in Fig. 4, we add a direct link inspired by the ideas of model predictive control, i.e., the effect of the control strategy is explicitly considered in the machine learning.

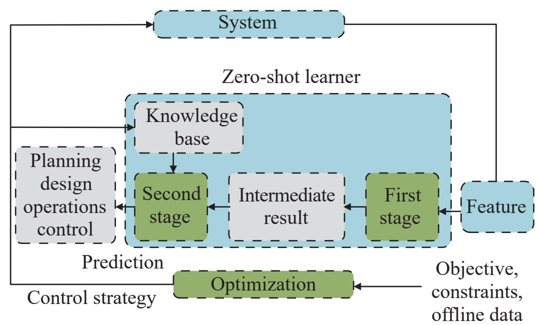

Fig. 4. Flowchart of the zero-shot transfer learner.

The essence of model transfer is to predict another system using the experience learned from the previous system. However, the online car-hailing dispatching algorithm may issue dozens of dispatching instructions per second, which will lead to dynamic changes of the system. Therefore, for these high real-time dynamic control signals, the model transfer approach is not suitable. To address this challenge, we establish a simulation environment, design a control algorithm based on RL, and subsequently let the model learn mapping relationships among dynamic control signals, data, and output in the developed simulation environment. The details of the solution in this situation will be elaborated in Scenario 2.

Scenario 1: This scenario deals with low real-time control signals which do not change very frequently, such as subway timetables. We use the set S to represent the set of control strategies that have been adopted, which have correspondingly available historical data. In general, for any low real-time control strategy $s \in S$, the current system is E_s, we can establish a supervised machine learning model, denoted by M_s, to learn the mapping from the input features to the output. When the control strategy changes to t and $t \notin S$, the system is denoted by E_t at this time. The model M_s cannot predict the system E_t directly because there is no data of the system E_t in the training data. This kind of problem is called the zero-shot learning problem in the field of transfer learning [15].

Inspired by [15] and other work on transfer learning [16], [17], a two-stage zero-shot model is built to predict traffic information under a partial observation control strategy. The steps for building the two-stage zero-shot model in this scenario are as follows:

Step 1: The data in the system E_s are taken as the training data set (input features and labels are known), and the data in the system E_t are the test data (input features are known and labels are unknown). First, the training data (features and labels) and the test data (features) should be standardized to the same scale (e.g., in the form of a ratio of all-day traffic volume). We let X^m denote the input features, where m is the dimension of the features. The zero-shot learner first maps from X^m to a scaling label $\hat{Y}$, which is an intermediate result. Since the features in both the training and test sets are in the same m-dimensional input space, the traffic patterns learned by the model in the training set can be used for prediction in the test set. Note that prediction results are relative and need to be rescaled to absolute values.

Step 2: Palaucci *et al*. [15] inferred new image classes in the case of zero-data, in which auxiliary information was added in the form of a semantic knowledge base. The semantic knowledge base contains many complex images attribute descriptions. Therefore, we also established a control strategy knowledge base to obtain additional information for the zero-data traffic information prediction problem. Let K denote the control strategy knowledge base, which is a collection of control strategy, attributes, correction factor.

$$K = \{control\ strategy, attributes, correction\ factor\}. \tag{2}$$

In the case of no training data, we can still define the control strategy knowledge base according to the inherent attributes and calculation formula of ITS and use a look-up table to correct the results during prediction. With the assistance of the knowledge base, the above intermediate results are mapped to the final predicted values Y.

The proposed two-stage process is reflected by two functions $\mathcal{F}(\cdot)$ and $\mathcal{L}(\cdot)$, the model $\mathcal{H}(\cdot)$ is generated by the combination of the two functions

$$\mathcal{H}(\cdot) = \mathcal{L}(\mathcal{F}(\cdot)) \tag{3}$$

where

$$\mathcal{F} : X^m \rightarrow \hat{Y} \tag{4}$$

$$\mathcal{L} : \hat{Y} \rightarrow Y. \tag{5}$$

Fig. 4 shows the flowchart of the zero-shot transfer learner. Based on the developed zero-shot model, the traffic information can be predicted accurately when the control strategy is unknown.

Scenario 2: This scenario deals with high real-time control signals in ITS, which will change in real-time with the change of the system. For example, dozens of online car-hailing dispatching instructions are issued every second. To address this challenge, we establish a simulation environment, design control algorithms based on RL, and then let the model learn the mapping relationships among dynamic control signals, data, and output in the simulation environment, for improving the accuracy of spatio-temporal prediction.

In fact, in general RL, the agent only inputs the current state of the simulator without considering the influence of previous control action on prediction. Whereas in traffic problems, previous control actions can also have a significant impact on prediction results.

For example, the driver's execution of the dispatching instruction issued by the online car-hailing platform will have a direct impact on the future supply and demand, resulting in a dramatic decrease in the performance of the prediction model. Therefore, we design a RL model with a "recurrent" structure. The term "recurrent" means that the output of the model depends not only on the current computation but also on previous computations, which is similar to recurrent neural networks (RNN) [18].

Fig. 5 shows a simple RNN architecture. x_t is the input of time step t and P_t is the hidden state of the time step t. P_t is calculated according to the previous hidden state and the input of the current step as follows:

$$P_t = f(Ux_t + WP_{t-1} + b) \tag{6}$$

where function $f(\cdot)$ is usually a nonlinear function. U and W are weight matrices.

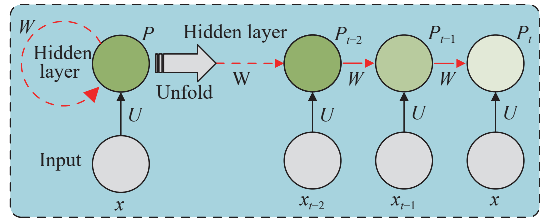

Fig. 5. Illustration of one input unit and one recurrent hidden unit.

In our method (as shown in Fig. 6), the output of the agent depends not only on the current state of the simulator, but also on the previous control actions. We consider the influence of dynamic control signals on the output in the form of "recurrent".

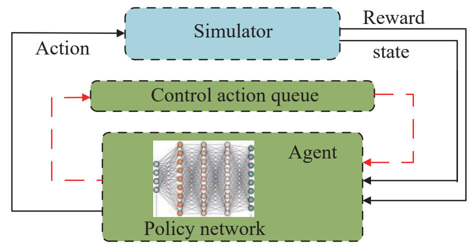

Fig. 6. Illustration of an RL model with "recurrent" structure.

Experiments: Taking the classical passenger flow prediction problem as an example, we conducted a preliminary experiment to verify the hypothesis of this letter, i.e., whether control signals (e.g., metro timetable) will play a key role in traffic prediction. The data were collected from the Nanjing metro system, including travel records of weekdays from March 18 to April 30 and from August 1 to November 9, 2016. A dataset containing 103 days of records was obtained by denoising, in which the last 33 days of data are the test set, while the rest of the samples were used as the training set. In this case study, the length of the time slice is set to 10 minutes, which means our task is to predict the number of card swipes in the next ten minutes.

We use four evaluation metrics, namely, symmetric mean absolute percent error (SMAPE), root mean square error (RMSE), mean absolute error (MAE), and mean relative error (MRE), to evaluate the performance of the model separately.

$$SMAPE = \frac{2}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i + \hat{y}_i} \times 100\% \tag{7}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{8}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{9}$$

$$MRE = \frac{1}{N}\sum_{i=1}^{N}\frac{|y_i - \hat{y}_i|}{y_i} \times 100\% \tag{10}$$

where y_i is the actual passenger flow at the i-th time slice and $\hat{y}_i$ is the predicted result. N is the number of samples to be predicted.

The proposed model controlled prediction method is compared with the autoregressive integrated moving average (ARIMA) model and the LSTM model. The parameters p (AR term), d (difference order), and q (MA term) of the ARIMA model are set to 7, 1, and 1 respectively. In the LSTM model, we use the information from the previous four time slices to predict the passenger flow in the next time slice, stacking three LSTM layers to enable the model to learn higher-level temporal representation.

In the model controlled prediction model, based on the LSTM model, we further encode the metro arrival information (i.e., metro timetable) within i-th time slice as a 10-dimensional feature vector.

Multiple fully connected layers are used to learn the relationship between the metro timetable and the metro passenger flow. We use M_i to represent metro arrival information at i-th time slice

$$M_i = \left[m_i^1, m_i^2, \ldots, m_i^{10}\right] \tag{11}$$

where m_i^j=1 means that the metro will arrive at the current station at j-th minute in the i-th time slice.

This study selects three representative metro stations, namely: transfer station, regular station, and regular station with low passenger flow. The results are presented in Tables 1−3. Compared with the metrics of the benchmark model, the model controlled prediction method reduces the error of all three types of metro stations. The experimental results show that the hypothesis of this paper is correct, and the reasonable modeling of control signals can significantly improve the performance of the traffic prediction model. In future work, we will conduct in-depth research for Scenarios 1 and 2 proposed in this paper.

Table 1. Comparison of Different Models (Transfer Station)

	ARIMA	LSTM	Model controlled prediction
SMAPE	25.2%	16.7%	16.5%
RMSE	96.2	65.4	60.4
MAE	57.8	45.6	42.0
MRE	35.1%	17.2%	16.6%

Table 2. Comparison of Different Models (Regular Station)

	ARIMA	LSTM	Model controlled prediction
SMAPE	22.8%	18.1%	17.5%
RMSE	55.5	37.9	35.7
MAE	31.0	25.1	23.0
MRE	29.4%	19.9%	19.0%

Table 3. Comparison of Different Models (Regular Station With Low Passenger Flow)

	ARIMA	LSTM	Model controlled prediction
SMAPE	41.6%	26.8%	26.1%
RMSE	11.5	10.6	10.0
MAE	7.6	7.4	7.1
MRE	52.1%	26.3%	22.9%

Conclusions: The accurate transportation prediction is the foundation for all aspects of ITS, including control, operations, design, and planning. However, most prediction models in ITS do not consider the influence of external control signals (e.g., subway timetables), which compromise the performance, applicability, and transferability of these models. So far, only model predictive control has integrated predictions with external control signals. However, these models are only used for control and not for other aspects of ITS. This research is the first attempt to deal with the most fundamental issue of traffic prediction, considering external control signals, and provide a foundation for ITS applications at all levels. Although the model is developed for ITS, the fundamental theory developed will be sufficiently general to be applicable to other disciplines and systems, provided that the predictions are heavily influenced by external control signals.

In the short term, the research provides a theoretical basis for consolidating predictions and external control signals, thus promoting the scientific development in this area. The theory can also be used in many key use cases, such as the early warning of sudden passenger flow in public transport systems, and the supply-demand balancing in ride-hailing services. In the long run, this research will be even more important in the coming era of connected, automated, and electric vehicles, where the transportation systems, communication systems, and electricity grid are coupled together. This research provides a possible solution for the interactions among different sub-systems in the future urban transportation systems.

Acknowledgments: This study is part of a project that has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie (101025896).

References

[1] Y. Liu, Z. Liu, and R. Jia, "DeepPF: A deep learning based architecture for metro passenger flow prediction," *Transportation Research Part C*: *Emerging Technologies*, vol. 101, pp. 18–34, 2019.

[2] Y. Liu, C. Lyu, Y. Zhang, Z. Liu, W. Yu, and X. Qu, "DeepTSP: Deep traffic state prediction model based on large-scale empirical data," *Communications in Transportation Research*, vol. 1, p. 100012, 2021.

[3] L. Zhao, Y. Song, C. Zhang, Y. Liu, P. Wang, T. Lin, M. Deng, and H. Li, " T-GCN: A temporal graph convolutional network for traffic prediction," *IEEE Trans. Intelligent Transportation Systems*, vol. 21, no. 9, pp. 3848–3858, 2020.

[4] C. Li, Y. Xie, G. Wang, X. Zeng, and H. Jing, " Lateral stability regulation of intelligent electric vehicle based on model predictive control," *J. Intelligent and Connected Vehicles*, vol. 4, no. 3, pp. 104–114, 2021.

[5] F.-Y. Wang, N.-N. Zheng, D. Cao, C. M. Martinez, L. Li, and T. Liu, "Parallel driving in CPSS: A unified approach for transport automation and vehicle intelligence," *IEEE/CAA J. Autom. Sinica*, vol. 4, no. 4, pp. 577–587, 2017.

[6] A. Caprara, M. Fischetti, and Toth, " Modeling and Solving the Train Timetabling Problem," *Operations Research*, vol. 50, no. 5, pp. 851–861, 2002.

[7] J. Högdahl, M. Bohlin, and O. Fröidh, " A combined simulation-optimization approach for minimizing travel time and delays in railway timetables," *Transportation Research Part B*: *Methodological*, vol. 126, pp. 192–212, 2019.

[8] Z. Hou, H. Dong, S. Gao, G. Nicholson, L. Chen, and C. Roberts, "Energy-saving metro train timetable rescheduling model considering ATO profiles and dynamic passenger flow," *IEEE Trans. Intelligent Transportation Systems*, vol. 20, no. 7, pp. 2774–2785, 2019.

[9] M. Abdolmaleki, N. Masoud, and Y. Yin, " Transit timetable synchr-onization for transfer time minimization," *Transportation Research Part B*: *Methodological*, vol. 131, pp. 143–159, 2020.

[10] J. Wang, Y. Sun, Z. Zhang, and S. Gao, "Solving multitrip pickup and delivery problem with time windows and manpower planning using multiobjective algorithms," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 4, pp. 1134–1153, 2020.

[11] L. Li, Y. Lv, and F.-Y. Wang, " Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Autom. Sinica*, vol. 3, no. 3, pp. 247–254, 2016.

[12] Y. Liu, F. Wu, C. Lyu, S. Li, J. Ye, and X. Qu, "Deep dispatching: A deep reinforcement learning approach for vehicle dispatching on online ride-hailing platform," *Transportation Research Part E: Logistics and Transportation Review*, vol. 161, p. 102694, 2022.

[13] X. Li, Y. Liu, K. Wang, and F.-Y. Wang, " A recurrent attention and interaction model for pedestrian trajectory prediction," *IEEE/CAA J. Autom. Sinica*, vol. 7, no. 5, pp. 1361–1370, 2020.

[14] Y. Liu, C. Lyu, X. Liu, and Z. Liu, "Automatic feature engineering for bus passenger flow prediction based on modular convolutional neural network," *IEEE Trans. Intelligent Transportation Syst.*, vol. 22, no. 4, pp. 2349–2358, 2021.

[15] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell, "Zero-shot learning with semantic output codes.," in *Proc. 22nd Int. Conf. Neural Information Processing Systems*, pp. 1410–1418, 2009.

[16] E. F. Ohata, G. M. Bezerra, J. V. S. das Chagas, A. V. L. Neto, A. B. Albuquerque, V. H. C. de Albuquerque, and P. P. Reboucas Filho, "Automatic detection of COVID-19 infection using chest X-ray images through transfer learning," *IEEE/CAA J. Autom. Sinica*, vol. 8, no. 1, pp. 239–248, 2021.

[17] Z. Liu, Y. Liu, Q. Meng, and Q. Cheng, "A tailored machine learning approach for urban transport network flow estimation," *Transportation Research Part C*: *Emerging Technologies*, vol. 108, pp. 130–150, 2019.

[18] M. Schuster and K. K. Paliwal, " Bidirectional recurrent neural networks," *IEEE Trans. Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.