Digital Twin Based Trajectory Prediction for Platoons of Connected Intelligent Vehicles

Hao Du[1,2], Supeng Leng[1,2,*], Jianhua He[3], Longyu Zhou[1,2]
[1]School of Information & Communication Engineering
University of Electronic Science and Technology of China (UESTC), Chengdu, China
[2]Shenzhen Institute for Advanced Study, UESTC, Shenzhen, China
[3]School of Computer Science and Electronic Engineering, University of Essex, UK
*Corresponding author email: spleng@uestc.edu.cn

2021 IEEE 29th International Conference on Network Protocols (ICNP) | 978-1-6654-4131-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/ICNP52444.2021.9651970

***Abstract*—Vehicle platooning is one of the advanced driving applications expected to be supported by the 5G vehicle to everything (V2X) communications. It holds great potentials on improving road efficiency, driving safety and fuel efficiency. Apart from the organization and internal communication of the platoons, real-time prediction of surrounding road users (such as vehicles and cyclists) is another critical issue. While artificial intelligence (AI) is receiving increasing interests on its application to trajectory prediction, there is a potential problem that the pre-trained neural network models may not well fit the current driving environment and needs online fine-tuning to maintain an acceptable high prediction accuracy. In this paper, we propose a digital twin based real-time trajectory prediction scheme for platoons of connected intelligent vehicles. In this scheme the head vehicle of a platoon senses the surrounding vehicles. A LSTM neural network is applied for real-time trajectory prediction with the sensing outcomes. The head vehicle controls the offloading of the trajectory data and maintains a digital twin to optimize the update of LSTM model. In the digital twin a Deep-Q Learning (DQN) algorithm is utilized for adaptive fine tuning of the LSTM model, to ensure the prediction accuracy and minimize the consumption of communication and computing resources. A real-world dataset is developed from the KITTI datasets for simulations. The simulation results show that the proposed trajectory prediction scheme can maintain a prediction accuracy for safe platooning and reduce the delay of updating the neural networks by up to 40%.**

***Index Terms*—intelligent platoon, trajectory prediction, digital twin, LSTM neural network updating.**

I. Introduction

In the last several years we have witnessed tremendous advances of autonomous vehicles and connected vehicles. Variety of sensors have been used to provide effective and abundant perception such as camera, radar, Li-dar, GPS/IMU and ect [1]. On the other hand, the vehicle to everything (V2X) communications is advancing at fast pace to provide ultra-reliable and low latency communications for connected vehicle applications [2]. The connected autonomous vehicles (CAV) is widely regarded one of the most promising technologies to tackle the global road transport safety and efficiency problems [3]. Vehicle platooning is one of the advanced CAV applications expected to be supported by the 5G V2X. In the platoons intelligent vehicles with similar driving speed towards the same direction can be organized together in a group, in which platoon members communicate with each other with low latency [4]. Vehicle states and sensing data can be shared in the platoon so that safe and harmonious driving can be maintained [5].

Platooning holds great potentials on improving road efficiency, driving safety and fuel efficiency. Apart from the organization (e.g., formation of platoons and members leaving) and internal communication of the platoons, real-time prediction of surrounding road users (such as vehicles, cyclists and pedestrians) is another critical issue. With effective and accurate prediction of other road users' trajectories, the platoons can have more reaction time and make proper decisions for safe and efficient driving [6]. Artificial intelligence (AI) is receiving increasing interests on its application to trajectory prediction. For example, the ANN and SVM classifiers have been proposed for lane change prediction which are able to predict a lane change action before it happens [7]. In research [8], a modified LSTM models for trajectory prediction is proposed. However, the traditional trajectory prediction is based on the fixed position sensors and few studies consider the real-time trajectory prediction of surrounding vehicles in the moving CAVs. For the vehicle platooning applications, the task of predicting multiple trajectories by the head vehicle could be overwhelming due to limited computing resources at one CAV. In addition, there is a potential problem that the pre-trained neural network models may not well fit the current driving environment and needs online fine-tuning to maintain an acceptable high prediction accuracy.

To tackle the above challenges faced by the vehicle platooning, we should consider offloading the computing jobs of predicting multiple trajectories to other CAV vehicles in the platoons. Mobile Edge Computing(MEC) servers and Vehicular Fog Computing(VFC) infrastructures which have computing and communication abilities [9], [10]. In addition, a distributed machine learning paradigm can be applied, where the clients can train the deep neural network (DNN) models locally [11]. The platoon enables the distributed learning for training the neural networks for multi targets trajectories prediction. The trajectories data can be shared with the platoon

This work was supported by the National Key R&D Program of China (No.2018YFE0117500), China Scholarship council (No.202006070036), and the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No 824019 and grant agreement no 101022280.

members and used to simultaneously train and fine tune the trajectory prediction models, which can reduce the training of the DNNs.

As platooning is a safety critical driving application, the trajectory prediction model needs to maintain very high accuracy in order to prevent wrong predictions and cause adverse consequences. Therefore the head vehicle should monitor the driving environment and performance of the trajectory prediction models, and adaptive control the fine tuning of the models as needed. If the models are updated unnecessarily too frequent, excessive communication and computing resources may be consumed and prevent timely training of the models. Therefore proper schemes should be designed to support efficient and safe decisions on the offloading of prediction jobs and distributed training of the models.

In this paper, we propose a digital twin based real-time trajectory prediction scheme for platoons of connected intelligent vehicles. Digital twin system is a promising technology which mirror the physical entities into the virtual models for long-term optimization of decision making [12]. It has been widely studied and found applications in manufacturing and Internet of thing (IoT), such as in task offloading [13] and preventing cyber-attacks for smart grid [14]. In the proposed trajectory prediction scheme the head vehicle of a platoon senses the surrounding vehicles and controls the offloading of the trajectory data. It also maintains a digital twin to optimize the update of LSTM model. A Deep-Q Learning (DQN) algorithm is utilized in the digital twin for adaptive fine tuning of the LSTM models, to ensure the prediction accuracy and minimize the consumption of communication and computing resources. To the best of our knowledge, our work is the first of its kind on the application of digital twin for CAV safe driving and platooning.

Our research contributions can be summarized as follows.

- A real-time trajectory prediction scheme for CAV platooning is proposed in this paper. The tasks of prediction multiple trajectories are offloaded to the CAV platoon members and the LSTM model for trajectory prediction is fine tuned with distributed learning to reduce processing delay.
- We propose a digital twin based method to adaptive fine tune the LSTM neural network model. The digital twin system mirrors the platoon into a virtual model and a DRL agent is applied to assist the decision making of updating the trajectory prediction model.
- A DQN algorithm is proposed for the DRL agent to maintain high prediction accuracy in a safe threshold and reduce the processing delay. A real-world dataset is developed from the KITTI datasets for simulations. The simulation results show that the proposed trajectory prediction scheme can maintain a prediction accuracy for safe platooning and reduce the delay of updating the neural networks by up to 40%.

The remainder of this paper is organized as follow. In Section II, we introduce the digital twin based prediction scheme for platoon driving and the system model of physical entities and digital mirrors. The DQN algorithm empowered digital twin system for trajectory prediction is presented in Section III. Section IV gives the numerical results. In Section V, conclusions and future research are presented.

II. Digital Twin based Trajectory Prediction Scheme

In this section, we first introduce the architecture of digital twin system for real-time trajectory prediction in the platoon. Then, the system model is presented for both physical entities and digital twin system.

A. Architecture of vehicular Digital twin system

As shown in Fig. 1. The architecture consists of two layer which are physical entities layer and digital twin layer.

In the physical entities, the platoon consists of a number of intelligent vehicles and they can communicate with each other. In order to keep safe driving, an intelligent vehicle equipped with high accuracy sensors is responsible for monitoring the traffic situations. The platoon tracks and predicts the trajectories of surrounding social vehicles which are not members of the platoon. LSTM neural network is adopted for predicting the trajectories of social vehicles and a LSTM neural network is used for each detected social vehicle. Due to the limited computation resources of the head vehicle, a distributed LSTM neural network is utilized in the platoon. Therefore, the sensing data are offloaded to the platoon members and the group members train the LSTM neural networks simultaneously to reduce the processing delay. However, with the traffic conditions changed, the performance of pre-trained neural network deteriorates. Consequently, the neural networks should be updated with proper decision making strategies to keep the prediction accuracy in a safe threshold and reduce the processing delay in a long-term driving of the platoon.

We deploy the digital twin system at the head vehicle to control the update (fine tuning) of the LSTM neural network. The digital twin system analyses the information and mirrors the intelligent vehicles in the virtual model. Meanwhile, a DRL agent is set on this digital twin system to learn a optimal updating strategies through the state information processed by the digital mirror. The DRL agent aims at minimizing the processing delay with a reliable prediction accuracy. Finally, the optimal updating strategies processed in the platoon.

The operation of the digital twin based prediction scheme can be described as follows.

1) The head vehicle collects the sensing data in real time and process the trajectory prediction with distributed LSTM neural networks. If a updating decision is made, the head vehicle transmits the sensing data to the corresponding intelligent vehicles in the platoon to train locally.
2) The digital twin system deployed on the head vehicle clean and fuse the information for mirroring the physical entities as the virtual models.
3) The digital twin system analyses the state information of virtual models. Then, the pre-processed state information is sent to the DRL agent for optimization.

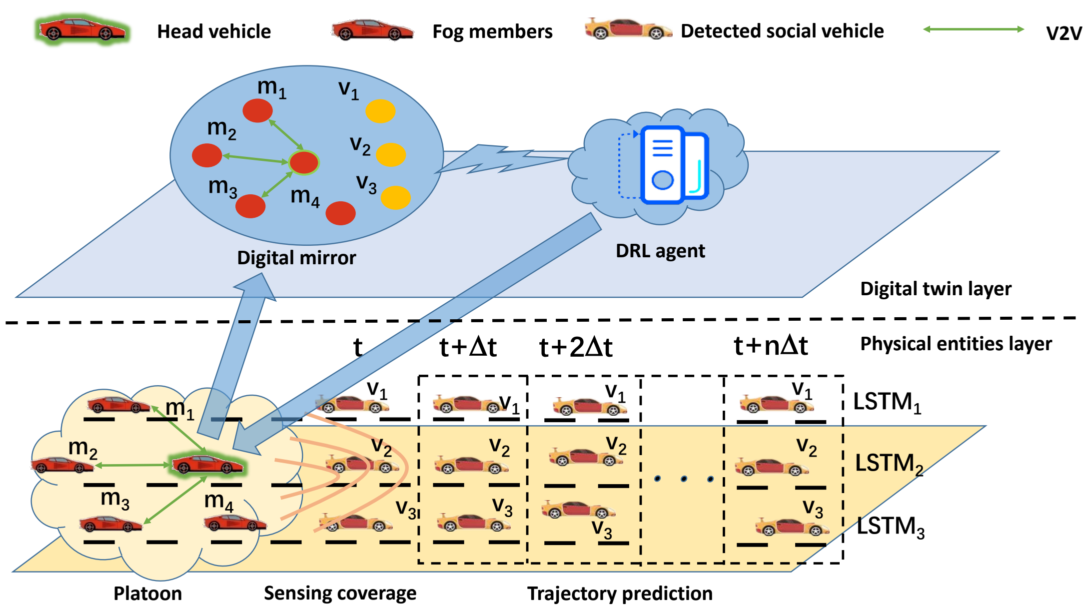

Fig. 1. Digital twin system for Platoon

4) The DRL agent calculates the optimal updating strategies for the LSTM neural network of trajectory prediction. The optimization is aiming at keeping the prediction accuracy in a safe threshold and minimizing the processing delay in a long-term driving. Consequently, the optimal strategies are sent back to the physical entities.
5) The platoon executes the updating strategies optimized by DRL agent. Meanwhile, the trajectory prediction still based on the present neural network trained in the last run. When finishing the updating, the new LSTM neural networks send back to the head vehicle immediately for trajectory prediction.

B. System model

In the physical entities layer, a head vehicle v_s acts as a head vehicle in the platoon which is responsible for sensing the social vehicles and making trajectory prediction. We assume that there are n intelligent vehicles in the platoon which are expressed as follow:

$$V = \{v_1, v_2, ..., v_n\}. \tag{1}$$

Meanwhile, the locational information of social vehicles which are not member of the platoon are detected by the head vehicle. We assume that the head vehicle is the origin of coordinates and the location of the m social vehicles are expressed as follow:

$$V^{Social} = \{v_1^{social}, v_2^{social}, ..., v_m^{social}\}, \tag{2}$$

$$X^t = \{x_1^t, x_2^t, ..., x_m^t\}, \tag{3}$$

$$Y^t = \{y_1^t, y_2^t, ..., y_m^t\}, \tag{4}$$

where x_i^t and y_i^t are the longitudinal and lateral position coordinates of the social vehicle v_i^{social} at time t.

When the digital twin system announces the vehicles to perform the update of neural networks. The sensing data are sent to the corresponding members in the platoon to train locally. The communication delay is depicted as follow:

$$D_{s_{v_i}} = \frac{B_i}{R_{s,i}}, \tag{5}$$

$$R_{s,i} = \lambda_{s,i} \log_2 \left(1 + \frac{P_i}{\sigma_0^2}\right), \tag{6}$$

where $D_{s_{v_i}}$ is the transiting delay of platoon member v_i. B_i is the volume of trajectory data transmitted to v_i. $R_{s,j}$ is the transmission rate of V2V channel between platoon member v_i and head vehicle v_s. $\lambda_{s,j}$ is the channel bandwidth. P_i is the transmission power. σ_0^2 is the noise power.

Then, the trajectory data is fed into the neural networks of each platoon member for local training. The delay of local training is computed as follow.

$$D_{c_{v_i}} = \frac{B_i \zeta}{f_i}, \tag{7}$$

where ζ is the number of CPU cycles per unit of data volume for training the LSTM network. f_i is the CPU frequency of platoon member v_i. Therefore, if the number of social vehicles are lower or equal to the platoon members. All the local training could process simultaneously. Otherwise, the local training should process the second or more round which depends on how many updates are going on simultaneously.

In order to minimize the processing delay, the offloading strategies is depict as follow:

$$\arg\min_{B_i} \max(D_{c_{v_i}} + D_{s_{v_i}}), \tag{8}$$

Then, the trajectory data are sent to the corresponding vehicles and the offload policy obeys the equation (8). In addition, the end to end input and output of the distributed LSTM neural networks are defined as:

$$I_i^t = [(x_i^{t-t_h}, y_i^{t-t_h}), (x_i^{t-t_h+T_s}, y_i^{t-t_h+T_s}), \\ ..., (x_i^t, y_i^t)] \tag{9}$$

$$O_i^t = [x_i^{t+\Delta t}, y_i^{t+\Delta t}] \tag{10}$$

where T_s is the sampling time slot of head vehicle and t_h determines the data length of input. Δt is the prediction time horizon. After training, the platoon members v_i updates the local neural networks and send back to the head vehicle for real time prediction. The back-haul delay is dismissed due to the insignificant data volume of parameters of neural networks.

In the digital twin layer, the state information of platoon is collected and mirrored as the virtual models. We utilize a DRL agent to optimize the updating strategies of LSTM neural networks. It is known that DRL problem can be modelled as a Markov decision process (MDP). The update strategies of neural networks aim at keeping high precision of trajectory prediction and minimizing the processing delay. It is obviously that with higher update rate, the trajectory prediction is more accurate. Therefore, we define the system state as S. The system state at time t can be depict as follow:

$$S_t = [P(t), D(t)], S_t \in S, \tag{11}$$

$$P(t) = \sqrt{\sum_{i=1}^{m} \frac{1}{m}[(x_i^t - \overline{x}_i^t)^2 + (y_i^t - \overline{y}_i^t)^2]}, \tag{12}$$

where $P(t)$ is the Root Mean Squared Error(RMSE) between the real location and prediction location of target vehicles. $(\overline{x}_i^t, \overline{y}_i^t)$ is the prediction location of social vehicle i. $D(t)$ is the system delay of completing the update of neural networks.

Then, the digital twin system analyses the state information, the state information is sent to the DRL agent for optimizing the update strategies of neural networks for trajectory prediction. The action is defined as A and the neural networks update strategies at time t is depict as:

$$A_t = [a_t^1, a_t^2, ..., a_t^m], a_t \in A \tag{13}$$

where $a_t^i \in (0, 1)$ is the updating strategies of LSTM neural networks for social vehicle v_i^{social}. $a_t^i = 1$ means that the neural networks update with the newly acquired trajectory data. $a_t^i = 0$ means that the head vehicle uses the current neural network to process trajectory prediction applications. Therefore, the processing delay $D(t)$ can be written as:

$$D(t) = max\{D_{c_{v_i}} + D_{s_{v_i}}\} * \lceil \frac{\sum_{i=1}^m a_t^i}{n} \rceil \tag{14}$$

Once the action is decided by the DRL agent, the digital twin system gets an immediate reward. The reward function is depict as:

$$R_t = \alpha C(t) - \beta D(t) \tag{15}$$

$$C(t) = \begin{cases} \frac{1}{P(t)} & P(t) \leq Threshold_{RMSE} \\ -P(t) & Threshold_{RMSE} < P(t) \end{cases} \tag{16}$$

where α and β are the parameters of prediction RMSE and processing delay. And the $Threshold_{RMSE}$ is the maximum acceptable RMSE of trajectories prediction. With lower RMSE and processing delay the digital twin system will get more reward. If the $P(t)$ is bigger than the $Threshold_{RMSE}$, the reward will be punished with negative value to avoid this situation happen.

However, it is obvious that we cannot get all rewards to calculate the return of each state. Therefore, the Q-learn based Action-Value function is proposed to evaluate the reward after execute the action:

$$Q^{\pi}(s, a) = E[R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + ... \\ |(s, a)] \tag{17}$$

where γ is the discount factor and it determines the length of MDP's forecast horizon for the future, with 1 indicating that all rewards are treated equally, and 0 indicating that the affect of future rewards is ignored. We set the time slot between two observation statement as T. To ensure that the $P(t)$ would not bigger than the $Threshold_{RMSE}$, the processing delay of distributed training must be lower than the time slot T. Therefore, we set the observation time slot as:

$$T = max\{D_{s_{v_i}} + D_{c_{v_i}}\} \tag{18}$$

III. DQN EMPOWERED DIGITAL TWIN SYSTEM FOR TRAJECTORY PREDICTION

In physical entities, the LSTM neural network are used for the application of trajectory prediction. In digital twin layer, a DQN algorithm is applied for optimizing the update strategies of the LSTM neural network. The trade-off between prediction precision and processing delay is well settled. As mentioned in the last section, we formulate a problem to find the optimal strategy of updating the neural networks. We utilize DQN algorithm to optimize the action A. Therefore, the formula (17) can be written as:

$$Q(s, a, \theta) = E_{s'}[R + Q^{\pi}(s', a', \theta)|(s, a, \theta)] \tag{19}$$

where θ is the parameter of approximation of value function Q. s' and a' are the next state on executing action A in state S. Meanwhile, the loss function is depict as follow:

$$L(\omega) = E_{s'}[R + max_{s'}(s', a', \theta) - Q(s, a, \theta)^2] \tag{20}$$

According to DQN algorithm, ϵ greedy algorithm is widely used in action selection. The threshold ϵ and random number ϕ

are set for the ϵ greedy algorithm in advance and ϕ is generated each time. The value of Φ determines the exploration and exploitation of ϵ greedy algorithm. Especially, if $\phi > \epsilon$ then the agent selects the action by exploitation. Otherwise, the agent selects the action according to exploration. Then, Stochastic Gradient Descent(SGD) is utilized to update the ω to achieve end-to-end optimization goals. The DQN algorithm for updating the neural networks is shown in **Algorithm 1**. Then, the digital twin system gets the optimal update strategies and transmits these strategies to the physical entities. The physical entities execute the trajectory prediction application with the update strategies.

Algorithm 1 DQN algorithm for updating the neural networks

Initialize the replay memory D to capacity N
Initialize action-value Q with random weights

for episode = 1, M do **do**
  Initialize sequence $s_1 = [P(1), D(1)]$ and preprocessed sequenced $\phi_1 = \phi(s_1)$
  for $t = 1$,T do **do**
    With probability ϵ select a random action a_t
    Otherwise select $a_t = max_a Q^*(\phi(s(t)), a; \theta)$
    Execution action a_t in digital twin system and observe reward R_t and image $[P(t+1), D(t+1)]$
    Set $s_{t+1} = s_t, a_t, [P(t+1), D(t+1)]$ and preprocess $\phi_{t+1} = \phi(s_{t+1})$
    Store transition $(\phi_t, a_t, R_t, \phi_{t+1})$ in D
    Sample random minibatch of transition $(\phi_t, a_t, R_t, \phi_{t+1})$ from D

$$y_j = \begin{cases} R_j & \text{for terminal } \phi_{t+1} \\ R_j + \gamma max'_a Q^*(\phi(s(t+1)), a'; \theta) \\ \text{for non-terminal } \phi_{t+1} \end{cases}$$

    Perform a gradient descent step on $(y_j - Q(\phi_j, a_j, \theta))^2$ according to equation (19)
  end for
end for

IV. Numerical results

For updating the LSTM neural networks, the real-world dataset KITTI is utilized for our simulation. The KITTI dataset uses a variety of sensors to record traffic scenes. In order to extract the path information in the real environment, we need to transform the object file in KITTI. We transform the classification and location information from the first person perspective of KITTI dataset into the third person perspective of fixed scale coordinate system. To convert between object dataset formats we used Python 3.6 to convert data from KITTI format to Cartesian coordinate format. In the conversion process, we first converted the GPS/IMU data of the test platform, and we used Mercator projection to get the Cartesian coordinates of the test platform:

$$x_i^t = s \cdot r \cdot \frac{\pi \cdot lon_i^t}{180}. \tag{21}$$

$$y_i^t = s \cdot r \cdot \log \tan \frac{\pi \cdot (90 + lat_i^t)}{180}. \tag{22}$$

where, $s = \cos \frac{\pi \cdot lon_{v_i}^0}{180}$ is a scale. $lon_{v_i}^t$ $lat_{v_i}^t$ are the longitude and latitude of vehicle v_i at time t. $lon_{v_i}^0$is the latitude of the starting point of v_i.

We extract 4 vehicles trajectories of 100 seconds for simulation. For comparison we set two benchmarks, namely fixed period update and random update. Meanwhile, the parameters are shown in Table 1. The processing delay of the maximum distributed training is 2 seconds with a 10 seconds history data input and 2 seconds prediction of trajectory. Consequently, we set the observation time slot $T = 2s$.

TABLE I
Summary of the simulation parameters.

Parameter	Value
Platoon members	4
noise power: σ^2(dbm)	-70
Transmission power: p_i(dbm)	33
V2V bandwidth: $\lambda_{i,j}$(Mb/s)	40
Length of input data volume:t_h(s)	10
Prediction horizon:Δt(s)	2
Data sampling rate:T_s(s)	0.1
hidden layers of LSTM	3
neurons of LSTM hidden layer	40
hidden layers of DQN	2
neurons of DQN hidden layer	50

Fig. 2 shows the convergence trend of the reward function of digital twin system for updating the neural networks. A small reward means that the processing delay is large. Otherwise, the processing delay is small. After 60 epochs the reward tends to converge. This shows that the DQN algorithm is adapted for the optimization model and the DRL agent can reduce the processing delay and ensure the RMSE below the threshold.

Fig. 3 shows the average RMSE of target vehicles of 2 seconds prediction in a 100 seconds time window. Without updating the neural networks. The performance of neural networks deteriorates rapidly this is very dangerous for driving safety. Based on our proposed scheme, the digital twin system minimizes the frequency of updates while ensuring the trajectories prediction RMSE is below the threshold. Meanwhile, the digital twin system adjusts the updating strategies dynamically. The huge infatuation of the random update is due to it cannot update the neural network at the proper time and cannot ensure the RMSE under the threshold which could be dangerous. Although the regular updating of 10 seconds can ensure the RMSE under the threshold, the too frequent updating wastes lots of computation resources and the processing delay is large. This could affect the processing of other applications of driving safety.

Fig. 4 shows the processing delay of different updating methods. Our proposed DQN algorithm has lower system cost than those two benchmarks. The proposed digital twin system

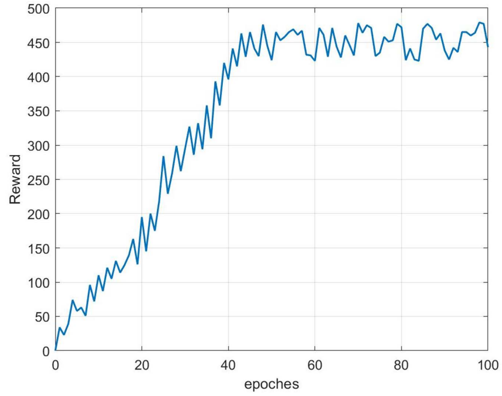

Fig. 2. The reward of digital twin system

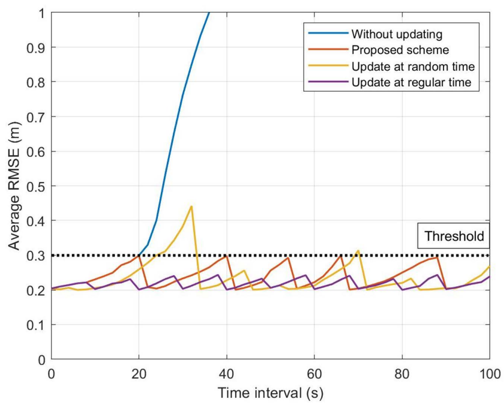

Fig. 3. The average RMSE of social vehicles

reduces the system cost up to 40%. This means that our digital twin system has an excellent updating strategies for dealing with the trade-off between prediction accuracy, computation consumption and processing delay. Moreover, our digital twin system converges quickly 60 epochs and can continuously adjust the update strategy based on the feedback of physical entities.

V. Conclusion

We proposed a digital twin based real-time trajectory prediction scheme for platoons of connected intelligent vehicles. In this scheme the head vehicle of a platoon senses the surrounding vehicles. A LSTM neural network is applied for real-time trajectory prediction with the sensing outcomes. The head vehicle maintains a digital twin to control the offloading of the trajectory prediction and the update of LSTM model. A Deep-Q Learning (DQN) algorithm was proposed for the digital twin to adaptively fine tuning of the LSTM model, with the objectives of ensuring the prediction accuracy and minimizing the processing delay. A real-world dataset is developed from the KITTI datasets for simulations. The simulation results show that the proposed trajectory prediction scheme can maintain a prediction accuracy for safe platooning and reduce the delay of updating the prediction model by up to 40%.

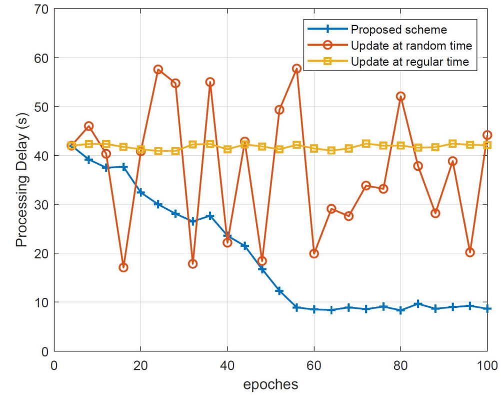

Fig. 4. The processing delay of physical entities

References

[1] E. Marti, M. A. de Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, pp. 94–108, 2019.

[2] J. He, H.-h. Chen, T. M. Chen, and W. Cheng, "Adaptive congestion control for dsrc vehicle networks," *IEEE Communications Letters*, vol. 14, no. 2, pp. 127–129, 2010.

[3] J. He, Z. Tang, Z. Fan, and J. Zhang, "Enhanced collision avoidance for distributed lte vehicle to vehicle broadcast communications," *IEEE Communications Letters*, vol. 22, no. 3, pp. 630–633, 2018.

[4] X. Chen, S. Leng, Z. Tang, K. Xiong, and G. Qiao, "A millimeter wave-based sensor data broadcasting scheme for vehicular communications," *IEEE Access*, vol. 7, pp. 149 387–149 397, 2019.

[5] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 263–284, Firstquarter 2016.

[6] J. Wei, J. He, Y. Zhou, K. Chen, Z. Tang, and Z. Xiong, "Enhanced object detection with deep convolutional neural networks for advanced driving assistance," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 4, pp. 1572–1583, 2020.

[7] Y. Dou, F. Yan, and D. Feng, "Lane changing prediction at highway lane drops using support vector machine and artificial neural network classifiers," pp. 901–906, 2016.

[8] S. Dai, L. Li, and Z. Li, "Modeling vehicle interactions via modified lstm models for trajectory prediction," *IEEE Access*, vol. 7, pp. 38 287–38 296, 2019.

[9] X. Zhu, Y. Luo, A. Liu, M. Bhuiyan, and S. Zhang, "Multi-agent deep reinforcement learning for vehicular computation offloading in iot," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 11 2020.

[10] J. He, J. Wei, K. Chen, Z. Tang, Y. Zhou, and Y. Zhang, "Multitier fog computing with large-scale iot data analytics for smart cities," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 677–686, 2018.

[11] W. Y. B. Lim, N. C. Luong, D. T. Hoang, Y. Jiao, Y.-C. Liang, Q. Yang, D. Niyato, and C. Miao, "Federated learning in mobile edge networks: A comprehensive survey," *IEEE Communications Surveys Tutorials*, vol. 22, no. 3, pp. 2031–2063, 2020.

[12] K. M. Alam and A. El Saddik, "C2ps: A digital twin architecture reference model for the cloud-based cyber-physical systems," *IEEE Access*, vol. 5, pp. 2050–2062, 2017.

[13] W. Sun, H. Zhang, R. Wang, and Y. Zhang, "Reducing offloading latency for digital twin edge networks in 6g," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 10, pp. 12 240–12 251, 2020.

[14] A. Saad, S. Faddel, T. Youssef, and O. A. Mohammed, "On the implementation of iot-based digital twin for networked microgrids resiliency against cyber attacks," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 5138–5150, 2020.