A Lane-Changing Decision-Making Model of Bus Entering considering Bus Priority Based on GRU Neural Network

Wanjun Lv, Yongbo Lv, Jianwei Guo, and Jihui Ma

School of Traffic and Transportation, Beijing Jiaotong University, Beijing 100044, China

Correspondence should be addressed to Yongbo Lv; yblv@bjtu.edu.cn

Received 26 July 2022; Revised 30 August 2022; Accepted 9 September 2022; Published 24 September 2022

Academic Editor: Ahmedin M. Ahmed

Copyright © 2022 Wanjun Lv et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A mandatory lane change occurs when buses are ready to enter the station, which will easily cause a reduction of urban road capacity and induce traffic congestion. Using deep learning methods to make lane-changing decisions has become one of the research hotspots in the field of public transportation, especially with the development of the Cooperative Vehicle-Infrastructure System. Aiming at the exploration of the bus lane-changing rules and decisions during entering, we built a GRU neural network model considering bus priority by using the first real-world V2X (vehicle to everything) dataset. Firstly, we illustrated the image and point cloud data processing by coordinate transformation. Secondly, the Kalman filtering algorithm was used to evaluate the vehicle state. Combined with the bus priority rules, we propose a flexible right-of-way lane in front of the bus stop. And then, we obtain the feature variables as inputs to the model. The XGBoost algorithm was chosen to train the GRU model. Results show that the model has higher identification accuracy for lane-changing maneuvers by comparison with other models. It plays a very important role in providing a decision basis for more refined bus operation management.

1. Introduction

Lane-changing behavior plays an important role in traffic flow theory. There are two types of lane-changing operations, namely, mandatory lane-changing (MLC) and discretionary lane-changing (DLC) [1]. Recently, researchers have shown an increased interest in the MLC maneuvers but paid less attention to the DLC scenes [2, 3]. MLC means that drivers are forced to leave the current lane because external factors have disturbed them or they are planning to reach their destination. A bus changing lanes and entering a station is one of the typical scenarios of MLC. Within the vicinity of the station, the bus must complete the lane change and prepare to stop at the station. Therefore, drivers are likely to adopt a radical lane change strategy, causing the surrounding vehicles to slow down, change lanes, and even queue, resulting in traffic congestion. More importantly, due to the large size of the bus itself, it will block the sight of the surrounding car drivers when changing lanes, which will easily lead to traffic accidents. Therefore, it is very necessary to study lane-changing behavior during the bus entering process.

One of the most significant current discussions in lane-changing is decision-making. Early lane change decision methods were mainly rule-based methods, which determined whether a vehicle met certain lane change rules by building a set of decision rules [4, 5]. Furthermore, models based on discrete choices are also used in lane change decisions [6–8]. This kind of model can take driver characteristics into account and at the same time has a simpler structure. However, the influence of different factors on the outcome of lane change is linear in these models, whereas lane change behavior is a process of coupled influence of multiple factors, and the relationship among the factors is often nonlinear. In recent years, in order to represent the nonlinear relationship between the input and output data more accurately, machine learning algorithms are increasingly being applied to lane-changing decision-making [9], such as support vector machine (SVM) [10], Bayesian filtering (BF) [11], and so on.
For MLC scenarios, many previous studies have focused on lane-changing behaviors in freeway on-ramp merging areas. There are fewer methods that have tried to solve the important issues of the lane-changing maneuvers that occur on buses entering. With the development of autonomous driving (AD) technology and the cooperative vehicle infrastructure system (CVIS), it is easier to collect enough bus-related data for machine learning algorithm training.

The purpose of this research is to model the bus lane-changing decision-making process using the first real-world V2X dataset for CVIS, with attention to the flexible bus priority along the bus entering route. The main contributions of this study are as follows:

(1) This paper focuses on the process of bus lane-changing decision-making, one of the MLC behaviors, which can seriously affect traffic conditions and is rarely explored by previous research studies.

(2) Based on real traffic data, a GRU neural network is used to solve the nonlinear bus lane-changing decision-making problem during entering. Combining the V2X technology, we take the bus priority into account, which is helpful to update real-time feature variables.

Section 2 reviews the literature on lane-changing methods. Section 3 is concerned with the methodology used for this study. The training results and discussion are shown in Section 4. Finally, Section 5 gives a brief summary and areas for further research.

2. Related Work

Lane-changing is one of the common driving behaviors in traffic flow. Gipps [4] is the first scholar who built the vehicle lane-changing decision model to test the factors that affect drivers’ lane-changing decisions in urban traffic. He proposed a logical frame for the lane-changing decision process. To better understand the mechanisms of the lane-changing model, other authors expanded the investigations by examining the diversified aspects of expressways and urban roads, including the lane-changing probability [12], drivers’ aggressive and courtesy behaviors [13], and the triggered negative effects [14]. On the other hand, some authors also tried to develop Gipps’s model. Kesting et al. first demonstrated the minimizing overall braking induced by Lane Change (MOBIL) model and considered the vehicle acceleration changes on the basis of the previous model [5]. As connected vehicles appeared, there was a large volume of published studies describing the role of cooperative merging strategies [15] and optimization algorithms to reveal the detailed mandatory lane-changing behaviors on freeways [16].

Besides, a considerable amount of literature uses utility theory to investigate lane-changing behaviors. The study of utility theory was first carried out by Ahmed et al. to examine the process of lane-changing decision-making on freeways. Toledo et al. integrated DLC and MLC behaviors by using the utility model [17]. Sun and Elefteriadou provided an in-depth analysis of the influences of drivers’ characteristics on lane-changing behaviors by observing drivers’ profiles and corresponding vehicle trajectory data [18, 19].

Recently, a number of machine learning-based lane-changing decision models have also emerged. Tang et al. utilized an adaptive learning approach to develop a fuzzy neural network-based prediction model for lane change behavior [20]. In response to the problem that it is difficult to model the multiparameter nonlinear features in the lane change decision process, the authors’ team developed a support vector machine-based lane change decision model [21]. A neural network-based lane change decision model was developed in the paper [22]. Multilayer perceptron (MLP) and convolutional neural network (CNN) architectures were combined to demonstrate how the lane change acceptance of a specific driver can be learned from lane change intention and the surrounding environment. The work [23] proposed a deep reinforcement learning (DRL)-based motion planning strategy for autonomous driving tasks in highway scenarios where an autonomous vehicle merges into two-lane road traffic flow and realizes the lane-changing maneuvers.

These existing LC models were mostly developed for freeways and are not likely to be applied to buses that are entering. In the above analysis, we introduce the GRU neural network to improve the MLC decision process during bus entering.

3. Materials and Methods

3.1. Data Processing. The DAIR-V2X dataset is the first large-scale, multimodal, multiview dataset captured from real scenarios for CVIS and contains data captured from the Vehicle-Infrastructure Cooperative view [24]. It contains 71,254 LiDAR frames and 71,254 camera frames captured in intersection scenes where a well-equipped vehicle passes through intersections with infrastructure sensors deployed. 40% of the frames are captured from infrastructure sensors and 60% of the frames are captured from vehicle sensors. All of them are precisely labeled by expert annotators. The dataset covers 10 km of city roads, 10 km of highways, 28 intersections, and 38 km² of driving regions with diverse weather and lighting variations. The subdatasets are presented in Table 1.

In this paper, the DAIR-V2X–C dataset containing 38,845 frames of image data and 38845 frames of point cloud data were selected to train the model. In order to obtain information about the buses and their surrounding vehicles, the 3-dimensional (3D) LiDAR point cloud data and the 2-dimensional (2D) camera image set need to be correlated through spatial coordinate transformation. There are three coordinate systems used in the dataset, including the world coordinate, the camera coordinate, and the LiDAR coordinate. The schematic diagram of coordinate transformation is shown in Figure 1.

The Lidar-to-Camera transformation can be obtained by simply multiplying the Lidar-to-World and World-to-Camera transformations. In order to generate the depth map $D_G$ of the ground plane with the same size as the image,
we use the ground plane equation $G(\alpha, \beta, \gamma, d)$ and camera intrinsic $K_{3 \times 3}$ as shown in the following equation:

$$
\begin{align*}
Z[x, y, 1]^T &= K^{3 \times 3} [X, Y, Z]^T, \\
G^{4 \times 4} [X, Y, Z, 1]^T &= 0
\end{align*}
$$

where $[x, y]$ is the pixel in the image coordinates and $[X, Y, Z]$ is the corresponding 3D point in the camera coordinate that lies on the ground plane. Thus, the depth $Z$ can be derived with the known 2D image points and the ground plane equation $G$.

After completing the matching of point cloud information and image information through coordinate conversion, we use the Kalman filtering algorithm to evaluate the vehicle state, including the 2 steps, namely, predicting and updating.

The first step is predicting, as shown in the following equation:

$$
\begin{align*}
x_{t+1} &= A\hat{x}_t + Bu_t + w_t, \\
P_{t+1} &= A\hat{P}_t A^T + Q
\end{align*}
$$

where $x_{t+1}$ is the predicted value of the system state vector at moment $t+1$. $\hat{x}_t$ and $u_t$ are the optimal estimated value and input control vector of the state vector at moment $t$, respectively. Matrices $A$ and $B$ denote the state transfer matrix and control matrix, respectively. $w_t$ is the noise of the system. $P_{t+1}$ represents the predicted value of the state estimation covariance matrix at moment $t+1$. $\hat{P}_t$ means the optimal estimation of the state estimation covariance matrix at moment $t$. $Q$ is the process noise covariance matrix.

The second step is updating, as shown in the following equation:

$$
\begin{align*}
K_{t+1} &= P_{t+1} H^T (H P_{t+1} H^T + R)^{-1} \\
\hat{x}_{t+1} &= x_{t+1} + K_{t+1} (z_{t+1} - H \hat{x}_{t+1}) , \\
\hat{P}_{t+1} &= (I - K_{t+1} H) P_{t+1}
\end{align*}
$$

where $K_{t+1}$ denotes the Kalman gain coefficient. $H$ is the measurement matrix. $R$ is the measurement noise covariance matrix. $\hat{x}_{t+1}$ denotes the optimal estimate of the state vector after the update at moment $t+1$. $z_{t+1}$ is the measurement value at moment $t+1$. $\hat{P}_{t+1}$ is the optimal estimation value of the state evaluation covariance matrix at moment $t+1$.

Because the bus lane-changing process does not involve external inputs to the system, both $B$ and $u_t$ in (2) are not taken into account. In this paper, we assume that the vehicle is treated as a uniform acceleration motion in each sampling period. Suppose the state vector is $[x, v, a]^T$, which means position, speed, and acceleration. $A_k$ is shown in (4). $H$ is the unit matrix.

$$
A_k = \begin{bmatrix}
1 & t & t^2 \\
0 & 1 & t \\
0 & 0 & 1 
\end{bmatrix}
$$

The state estimation of the target vehicle by applying the Kalman filtering algorithm jointly to the 3D LiDAR point cloud data and image data can obtain the position, velocity, and acceleration state information of the target vehicle.

3.2 Extraction of Feature Variables. Targeting buses’ apparently purposeful and time-sensitive lane-changing
behavior, it is necessary to guarantee the priority of right-of-way during their approach to the station. The process is illustrated in Figure 2.

The green area is the flexible right-of-way lane upstream of the bus stop, the length of which is determined by the traffic and roadway network status. FT is the upstream vehicle in the adjacent lane that needs to slow down or change lanes to avoid collision during the bus switches into this lane. PT is located downstream in the flexible right-of-way lane and will not turn right in the intersection, so it is requested to leave this lane; PT is located downstream in the flexible right-of-way lane and needs to turn right in the intersection, so it is not necessary to leave the current lane; PC is the vehicle located downstream in the lane where the bus is located. It is not allowed to change in the green area. The FC is the vehicle located upstream in the lane where the bus is located. Vehicles in other lanes do not directly affect the bus and are not analyzed.

According to Gipps' safe distance rules [4], \(D_{PC}\) and \(D_{PT}\) indicate the safe distance between the subject vehicle (the bus) and the preceding vehicle in the current lane and the target lane, respectively.

\[
D^i_{PC} = v_b^i \tau + \frac{(v_b^i)^2}{2d_b} - \frac{(v_{PC}^i)^2}{2d_{PC}},
\]

\[
D^i_{PT} = v_b^i \tau + \frac{(v_b^i)^2}{2d_b} - \frac{(v_{PT}^i)^2}{2d_{PT}},
\]

where \(\tau\) indicates the reaction time of bus driver. \(v_b, v_{PC},\) and \(v_{PT}\) are the initial velocity of the bus, PC, and PT in each sampling period, respectively. \(d_b, d_{PC}\), and \(d_{PT}\) are the maximum deceleration of the corresponding vehicle.

Similarly, the safe distance between the bus and the following vehicle before and after the bus changes the lane (enters the green area in Figure 2) can be formulated as follows:

\[
D'_{PC} = v_{FC}^i \tau + \frac{(v_{PC}^i)^2}{2d_{PC}} - \frac{(v_b^i)^2}{2d_b},
\]

\[
D'_{PT} = v_{FT}^i \tau + \frac{(v_{PT}^i)^2}{2d_{PT}} - \frac{(v_b^i)^2}{2d_b},
\]

where \(v_{FC}\) and \(v_{FT}\) denote the initial velocity of the FC and FT in each sampling period, respectively. \(d_{PC}\) and \(d_{PT}\) represent the maximum deceleration of the corresponding vehicle.

In a real-world lane-changing scenario, the bus tends to finish the lane-changing process as soon as possible. Some studies adopt the time ratio \(t^i_b/t_{max}\) to represent this characteristic, where \(t^i_b\) denotes the time used to finish the lane-changing process and \(t_{max}\) is the longest allowable time for lane-changing maneuver. The traffic efficiency description in this paper is developed by the above-mentioned method, and the index can be calculated as follows:

\[
\frac{t^i_b}{t_{max}} = \frac{\sum_{i}^{N} x^i_{b} - x^0_{b} / v_b^i}{t_{max}},
\]

where \(x^i_{b}\) and \(x^0_{b}\) represent the final and initial position of the \(i\)th sampling period, respectively. \(t^i_b\) can be approximately calculated by \(\sum_{i}^{N} x^i_{b} - x^0_{b} / v_b^i\).

After considering the safety and efficiency, we can obtain the feature variables as shown in Table 2.

Thanks to the advantages of CVIS, using V2X communication seems to be the logical solution. Other vehicles on the road can check whether there are approaching buses from behind periodically within a specific sensing distance. These feature parameters can be transmitted by vehicle-to-vehicle technology according to the expected wireless communication range of V2X systems. In case a vehicle detects an approaching bus that is entering the station in its adjacent or current lane, the vehicle will determine whether it affects the ideal bus operation to guarantee its priority. Of course, this paper assumes that all vehicles have real-time communication capabilities and does not reject the bus lane-changing requirement.

3.3. Lane-Changing Decision-Making Model. The gated recurrent unit (GRU) network is one of the variants of long-short-term memory (LSTM) networks in recurrent neural network (RNN), which mainly replaces the two gating units of LSTM (forgetting gate and input gate) with one gating unit (update gate). The GRU model is more streamlined than the standard LSTM model, with only two gating units in the hidden layer neuron structure and fewer network parameters, and its model structure is shown in Figure 3. It has been well developed and applied in the field of time series prediction.

The GRU neural network works similarly to the LSTM in which it has two gates, namely, a reset gate and an update gate. Both the reset gate and the update gate receive the hidden state \(h_{t-1}\) at the previous moment and the input information \(x_t\) at the current moment and the output gating signals are denoted as \(r_t\) and \(z_t\), respectively. The gating unit also consists of a sigmoid function and a dot product operation. The GRU works as follows.

Short time series dependencies are obtained via the reset gate as follows:

\[
r_t = \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r).
\]

Calculate update gate as follows:

\[
z_t = \sigma(W_{xz} x_t + W_{hz} h_{t-1} + b_z).
\]

The candidate vector is obtained after updating as follows:

\[
\tilde{h}_t = \tanh\left(W_{xh} x_t + W_{hh} (r_t h_{t-1}) + b_h\right).
\]
Update memory to get hidden layer output results as follows:

\[
ht = \sigma(1 - z_t) \cdot \ht_{t-1} + z_t \cdot \bar{ht},
\]

where \(\sigma\) is the sigmoid function and \(\text{tanh}\) is the \(\text{tanh}\) function. \(W_{xz}, W_{xr},\) and \(W_{xc}\) are the weight matrices from \(x_t\) to the update gate, reset gate, and candidate hidden states, respectively; \(W_{hzt}, W_{hr},\) and \(W_{hb}\) are the weight matrices from \(h_{t-1}\) to the update gate, reset gate, and candidate hidden states, respectively; \(b_x, b_r,\) and \(b_c\) are the bias.

The lane-changing decision-making model of the bus entering is illustrated in Figure 4. Firstly, image and point cloud data are matched by coordinate transformation. Then, using the Kalman filtering method and considering the bus priority rules during the entering process, we can obtain the state of corresponding vehicles and extract feature variables, namely, the inputs of the GRU neural network.

Table 2: Bus entering lane change decision feature variables.

<table>
<thead>
<tr>
<th>Feature variable</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>(v_b)</td>
<td>The velocity of the bus</td>
</tr>
<tr>
<td>(x_b)</td>
<td>The position of the bus</td>
</tr>
<tr>
<td>(d_b)</td>
<td>The maximum deceleration of the bus</td>
</tr>
<tr>
<td>(v_{n,n} \in {PC, PT, FC, FT})</td>
<td>The velocity of (PC, PT, FC, FT)</td>
</tr>
<tr>
<td>(x_{n,n} \in {PC, PT, FC, FT})</td>
<td>The position of (PC, PT, FC, FT)</td>
</tr>
<tr>
<td>(D_{n,n} \in {PC, PT, FC, FT})</td>
<td>The safe distance between the bus and (PC, PT, FC, FT)</td>
</tr>
<tr>
<td>(d_{n,n} \in {PC, PT, FC, FT})</td>
<td>The maximum deceleration of (PC, PT, FC, FT)</td>
</tr>
<tr>
<td>(T)</td>
<td>The reaction time of bus driver</td>
</tr>
<tr>
<td>(t_f)</td>
<td>The time used to finish the lane-changing process</td>
</tr>
<tr>
<td>(t_{max})</td>
<td>The longest allowable time for lane-changing maneuver</td>
</tr>
</tbody>
</table>

Figure 2: The bus priority of right-of-way when approaching the station.

Figure 3: GRU structure and calculation process. (a) Step 1 of GRU. (b) Step 2 of GRU. (c) Step 3 of GRU. (d) Step 4 of GRU.
4. Results and Discussion

4.1. Testing Data and Evaluation Metrics. We adopted the first large-scale DAIR-V2X dataset, which provides high-quality traffic data for research. The captured images obtained by cameras and the point clouds scanned by the LiDAR are calibrated and aligned between different sensors. The transformations among the world, the camera, and LiDAR are obtained, as well as the ground plane equation. The 2D-3D joint annotation is carried out by projecting the point clouds onto the images and adjusting the 3D bounding boxes manually to fit the 2D instance. As shown in Figure 5, there are a total of 9 kinds of traffic elements in the dataset. The objective of this paper is to show that, the corresponding number of samples is more than 60,000.

We used the scaled exponential linear unit (SeLU) as the activation function in the hidden layers. Compared with the traditional sigmoid and the popular rectified linear unit (ReLU), SeLU can converge towards zero mean and unit variance automatically. Therefore, it has the same effect as batch normalization, which means that exploding and vanishing gradients can be avoided. The SeLU activation function is shown as follows:

\[
\text{SeLU}(x) = \lambda \begin{cases} 
  x & \text{if } x > 0 \\
  \alpha e^x - \alpha & \text{if } x \leq 0 
\end{cases}
\]

(12)

where \(\lambda\) and \(\alpha\) are two parameters with typical values, namely, 1.0507 and around 1.6733, respectively.

According to our previous research [25], the operating time of the XGBoost algorithm is much faster than that of other programs on a single device. It can scale up to billions of examples in the case of distributed or limited memory. We chose XGBoost algorithm to train the GRU neural network in this paper.

An appropriate evaluation framework can help to estimate performance. Basically, samples in classification problems are divided into four categories, including true positives, true negatives, false positives, and false negatives. We introduce this evaluation framework for the model in Table 3.
TP and FN are the probabilities of actual lane-changing events correctly and wrongly classified as lane-changing and lane keeping events, respectively. FP and TN are the numbers of actual lane keeping events wrongly and correctly classified as lane-changing and lane keeping events, respectively. The following evaluation indicators (13) and (14) calculated by these four parameters are chosen in this paper:

\[
\text{accuracy (ACC)} = \frac{TP + TN}{TP + TN + FP + FN}, \quad (13)
\]

\[
\text{true positive rate (TPR)} = \frac{TP}{TP + FN}. \quad (14)
\]

Furthermore, the ROC curve can describe the relationship between TP and FP. The ROC score which calculates the area under the ROC curve is also a judging basis for the effect of algorithm.

4.2. The Optimal GRU Structure. In the paper [26], it was stated that overfitting in training neural networks was serious and should be given more attention. When overfitting happens, the error on the testing dataset is still high although the error of the model on the training dataset could be ignored. To avoid this phenomenon, various network structures, from simple to complicated, are tested in this paper. The different neural network structures are illustrated in Table 4 (where 0 means that there is no layer in the corresponding structure).

We use the cross-validation method to avoid overfitting, and 30% of the training samples are chosen as the validation set. Further statistical tests revealed that the model works efficiently with the help of the cross-validation method. We separate all the samples into 3 parts: 30% as the training dataset, 30% as the validation set, and the remaining 40% as the test dataset. For all the neural network schemes in Table 4, combined with the flexible right-of-way lane in front of the bus stop, generally at least 4,000 to 14,000 samples (10% to 40% of the total sample) are required. The results are shown in Figure 6.

Results show that structure 5 (3 hidden layers that contain 30, 10, and 0 neurons, respectively), trained by 10,000 samples, obtained the best performance. Furthermore, we selected 10,000 training samples to test different lengths of the historical time interval. Results show that considering the 8-second historical input is always the best choice. Figure 7 gives an example of ACC of the model.

To examine the accuracy of the bus lane-changing model in this paper, we compared the performance of the stochastic gradient descent (SGD) algorithm, the Random Forest model, and the SVM model with our model in the dataset. The indexes of different models are presented in Table 5. It can be seen that the GRU model in this paper performs better than other models. The ACC value of our model is up to 92.45%, much higher than that of others. It is worth noting that the TPR can guarantee the safety during bus lane-changing, which is vital for the driver to operate the lane-changing successfully. Hence, it could conceivably be noticed that the higher the value of TPR is, the safer the lane-changing will be. At the same time, the false negative would not have a negative impact on driving safety because it only leads to the loss of an opportunity to change lanes. Therefore, it is necessary to pay more attention to the accuracy of lane-changing maneuvers. The results of the model indicate that it is a safe and effective bus lane-changing decision-making system.

Figure 8 compares ROC curves among SGD, Random Forest, SVM, and GRU. It is apparent that the performance of the proposed GRU neural network is better than that of other binary classifiers, which means that the same conclusion can be drawn as what we get in Table 5. The gated
Figure 6: ACC values of different GRU network structures.

Figure 7: ACC values of different length of the historical time interval.
branch plays an important role in correlation analysis, which explicitly extracts the characteristics of the feature variables, which is helpful to the lane-changing decision.

5. Conclusion

Based on the DAIR-V2X dataset, this study builds a lane-changing decision-making model of a bus entering based on the GRU neural network. Image and point cloud data are matched by coordinate transformation to obtain the number of bus vehicles. To reduce the noise interference, the Kalman filtering algorithm is used to evaluate the buses’ position, speed, and acceleration. Meanwhile, the proposed method takes the bus priority rules into consideration when it enters the station, thus getting novel feature variables as input vectors. After calculating the experimental data, we choose the network structure “3 hidden layers that contain 30, 10, and 0 neurons, respectively,” as the optimal stricture and 8 seconds as the length of the historical time interval because the outcome performs best. By comparison and verification, it was proved that the proposed model has higher accuracy. The ACC value of our model is up to 92.45%.

Further research regarding the role of lane-changing would be worthwhile. This study considers the rules to ensure the priority of buses entering with the precondition that there is a 100% penetration rate of V2X technology in all vehicles. In a realistic road environment, the existence of disobedience to the lane-changing rules needs to be considered.

Data Availability

The labeled dataset used to support the findings of this study is available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (61872036).

References


