IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 18, 2021, accepted January 23, 2021, date of publication January 26, 2021, date of current version February 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3054755

A Novel Gated Recurrent Unit Network Based on SVM and Moth-Flame Optimization Algorithm for Behavior Decision-Making of Autonomous Vehicles

TAIQIAO YIN[1,2], YING LI[2,3], JIAHAO FAN[2,3], TAN WANG[4], AND YUNXIA SHI[2,3]

[1]College of Software, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering, Ministry of Education, Jilin University, Changchun 130012, China
[3]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[4]Space Technology (Jilin) Company Ltd., Jilin 132013, China

Corresponding author: Jiahao Fan (jihanfan@hotmail.com)

This work was supported in part by the Department of Science and Technology of Jilin Province under Grant 20190303135SF, and in part by the Development and Reform Commission of Jilin Province under Grant 2019C053-13.

ABSTRACT The behavior decision-making algorithm plays an important role in ensuring the safe driving of autonomous vehicles. However, existing behavior decision-making methods lack the capability to cope with future motion uncertainty in traffic, because the historical state of vehicles are not considered. This article proposes a novel driving behavior decision-making method EnMFO-ImGRU based on Gated Recurrent Unit (GRU) and Moth-Flame Optimization algorithm (MFO). Four improvements are proposed in EnMFO-ImGRU. First, to consider the driving information of the vehicles on the road, ImGRU is designed based on a double-layer GRU. Second, to promote decisions accuracy, Support Vector Machine (SVM), which has good performance in classification problems, replaces the softmax classifier to train the output of the ImGRU. Third, to promote the classification capability of SVM, MFO is introduced to optimize the key parameters that affect the performance of SVM. Finally, to promote the optimization capability of MFO, we propose the Enhanced Moth-Flame Optimization algorithm (EnMFO). A new position updating method is proposed in EnMFO. The experimental results on the NGSIM dataset show that EnMFO-ImGRU brings higher accuracy than existing methods for the behavior decision-making results of autonomous vehicles.

INDEX TERMS Autonomous vehicles, behavior decision-making, gated recurrent unit, support vector machine, moth-flame optimization algorithm.

I. INTRODUCTION

Autonomous vehicles contain environmental awareness module [1], behavior decision-making module [2], and path tracking module [3], of which behavior decision-making module ensures that autonomous vehicle behaves like skilled drivers in traffic flow. Hence, designing and developing behavior decision-making methods with high accuracy and short time-consuming is an important part of autonomous vehicles. Existing behavior decision-making methods are divided into mathematical modeling methods and deep learning methods.

The mathematical modeling methods establish state transition models based on logic rules and build vehicle behavior models by calculating risk indicators. Li *et al*. [4] designed a vehicle dynamics model to calculate the position and speed, and then selected the best trajectory which matched position and speed. Chen *et al*. [5] generated decisions by using Finite State Machine (FSM) to identify the relative positions of surrounding vehicles. Chae *et al*. [6] defined the collision probability by the reachable uncertainty propagation set. Predicted collision time and safety distance were used to monitor the lane changing risk. Kim and Kum [7] calculated the probability distribution model of vehicles' occupation of lanes basing on the lateral position and lateral speed of vehicles. And then, the collision risk was calculated and the collision

The associate editor coordinating the review of this manuscript and approving it for publication was Pavlos I. Lazaridis.

risk map was plotted. Nilsson *et al*. [8] estimated the safety factor of lane change to select appropriate traffic clearance and time instance and solved the loosely coupled convex quadratic program to locate the desired horizontal and vertical position of self-vehicle. The mathematical modeling methods take the potential risks into account, but the proposed models are limited to specific driving styles.

For deep learning methods, neural networks have shown excellent capabilities in the areas of feature extraction [9], object classification [10], and complex scene understanding [11]. Li *et al*. [12] proposed a decision-making network (DMN) using two Convolutional Neural Networks (CNNs) to train speed and corner decisions respectively. Lenz *et al*. [13] proposed a Gaussian Mixture Model which follows the Markov Property to predict vehicle behaviors in specific scenarios. Classification strategies were used for intent prediction, such as SVM [14], Multi-Layer Perception [15], Hidden Markov model [16], [17], and Bayesian Filter [18]. However, time correlations are not considered. Therefore, Recurrent Neural Network (RNN) methods, which have advantages in learning the nonlinear characteristics of time series, are widely used [19]. Shin *et al*. [20] used RNN to learn from the sequential data to predict collision probability. Zou *et al*. [21] put the continuous features extracted by CNN as inputs of RNN and obtained the prediction of lane selection.

Nonetheless, RNN has gradient explosion and gradient disappearance problems, which are solved in Long and Short-Term Memory (LSTM). Xie *et al*. [22] employed Deep belief network (DBN) and LSTM to generate decisions. Altché *et al*. [23] used an LSTM to forecast future vertical and horizontal trajectories of vehicles in the highway scene. Kim *et al*. [24] put vehicles coordinate sequence as the input of the LSTM and generated the future position probability of vehicles on the grid graph of road. Dang *et al*. [25] dealt with the behavior decision-making task as a regression problem rather than a classification problem and used LSTM to predict the time of lane changing. Xin *et al*. [26] used two LSTM modules for driver intent recognition and future trajectory prediction, respectively. Chen *et al*. [27] defined a dynamic sliding window for LSTM. The value of the window determined how much memory information was brought into current decisions. Valiente *et al*. [28] considered the time dependency between the current image frames and future image frames. Learning multiple sets of images through LSTM improved the accuracy of corner controlling. Shi *et al*. [29] proposed a model of three parallel LSTM and one LSTM in series. The model extracted features by different lanes. Zhang *et al*. [30] proposed an LSTM optimized by a Hybrid Retraining Constraint (HRC) training method to extract features. Scheel *et al*. [31] proposed a Bidirectional LSTM (Bi-LSTM), which combined with an Intelligent Driver Model (IDM) to predict future positions.

LSTM solves gradient problems, but the fact of too many parameters leads to high computational complexity. Same as LSTM, Gated Recurrent Unit (GRU) is proposed to solve the gradient problems. GRU performs similarly to LSTM but is computationally cheaper [32]. Benterki *et al*. [33] combined an LSTM with a GRU to analyze past trajectories and generate future trajectories of surrounding vehicles. Fei *et al*. [34] designed a GRU-assisted car-following model with driver time memory (CFDT) to generate decisions. Gu *et al*. [35] designed a fusion deep learning (FDL) model. FDL firstly selected lanes with a high correlation between the lanes to be predicted and then generated driving commands by GRU. GRU guarantees accuracy and training speed and overcomes gradient problems, which means it is suitable for implementing behavior decision-making algorithms.

However, two issues are still not resolved in existing improved GRU methods. First, existing improved GRU methods generally use a separate GRU to extract features of vehicles. Although the separate GRU performs well in feature extracting, the separate GRU trains driving features with constant parameters, network structure, and the number of neurons. Considering that the driving characteristics of surrounding vehicles and the driving characteristics of self-vehicle have different effects on the generated decision, hence, the capability of existing improved GRU methods to cope with future motion uncertainty in traffic is weak. Second, in existing improved GRU methods, the output of GRU is connected to a softmax classifier to generate decisions, resulting in low accuracy. Although SVM performs better than the softmax classifier in classification problems, the performance of SVM depends on the kernel function. Hence, optimizing kernel function parameters with meta-heuristic optimization algorithms is a widely used way to improve SVM performance.

For meta-heuristic optimization algorithms, Jiang *et al*. at [36] was inspired by the foraging behavior of beetle and proposed the Beetle Antennae Search (BAS). The beetle approached the food source according to the comparison of the odor intensity of the two antennae, and the head of the beetle was oriented randomly in each iteration, realizing the three-dimensional optimization. Kennedy and Eberhart [37] proposed Particle Swarm Optimization (PSO). In PSO, one bacteria simulated the foraging process through not only its own experience but also through the experience of other members. Mirjalili [38] simulated the behavior of moths hovering close to the flame and manually controlled the number of flames, and proposed Moth-Flame Optimization algorithm (MFO). Because of excellent robustness and strong exploitation capability, MFO is chosen in our method to combine with SVM. However, the exploration capability of MFO is weak.

Yu *et al*. [39] introduced an emulated annealing strategy into MFO to avoid converging too fast and falling into local optimum solution. However, the fusion of two meta-inspired optimization strategies leads to a lot of calculations. Zhao *et al*. [40] proposed an ameliorated Moth-Flame Optimization (AMFO). AMFO generated flame through Gaussian mutation to make it easy to solve when falling into local optimum solution. However, AMFO did not clearly indicate whether each iteration selects Gaussian mutation or flame number control mechanism of original MFO.

To sum up, we propose a novel network architecture named EnMFO-ImGRU. First, a novel neural network model called ImGRU is constructed to extract the key features of surrounding vehicles and self-vehicle, including speed, acceleration, lane number, vehicle number, steering angle, horizontal position, and vertical position of each vehicle. Of which, horizontal position and vertical position contain the width and length of vehicle. ImGRU consists of two GRUs calling GRU-Surround and GRU-Self respectively. GRU-Surround extracts the key features of vehicles from the left and right lane and the front vehicle, and GRU-Self extracts the key features of self-vehicle. In this way, ImGRU trains the key features of surrounding vehicles and self-vehicle with different parameters, ensuring that correct and reliable driving decisions are generated. Moreover, at each time step of ImGRU training, the output of GRU-Surround and the output of GRU-Self are fused into an internal hidden state matrix, and then the internal hidden state matrix and the vehicle features vector are fed to ImGRU at the next time step. In this way, ImGRU comprehensively considers the potential actions of surrounding vehicles and the adaptability of self-vehicle, and generates more reasonable and safe decisions. Second, to improve the accuracy of decisions, SVM replaces the softmax classifier to generate decisions by training the output of ImGRU. Third, to improve the classification capability of SVM, a meta-heuristic optimization algorithm is used to optimize key parameters of the SVM kernel function and tend to choose MFO. However, MFO suffers from the degenerating of convergence speed and uncoordinated exploration and exploitation capabilities. Fourthly, EnMFO is proposed to improve the optimization capability of MFO. On the one hand, the position updating formula of MFO is modified in EnMFO to improve the local exploitation capability, and a new position updating formula is added in EnMFO to improve the global exploration capability. In other words, EnMFO uses two position updating formulas. Therefore, a function whose value decreases as the number of iterations is proposed. By comparing the value of the function with a random number, it is determined which position updating formula is selected in the current iteration. On the other hand, Levy flight and adaptive weight are introduced in EnMFO. Levy flight uses random disturbances to make global exploration and local exploitation capabilities achieve balance, and the adaptive weight makes EnMFO converge faster and prevents falling into local optimal solution.

The main contribution of this article is the proposal of an effective behavior decision-making algorithm for autonomous vehicles. It contains four innovations to complete the feature extracting module and decision generating module of the behavior decision-making algorithm:

1) The feature extracting module is designed based on a double-layer GRU. One layer extracts features of surrounding vehicles, and the other extracts features of self-vehicle. In this way, the feasibility of decisions is improved.

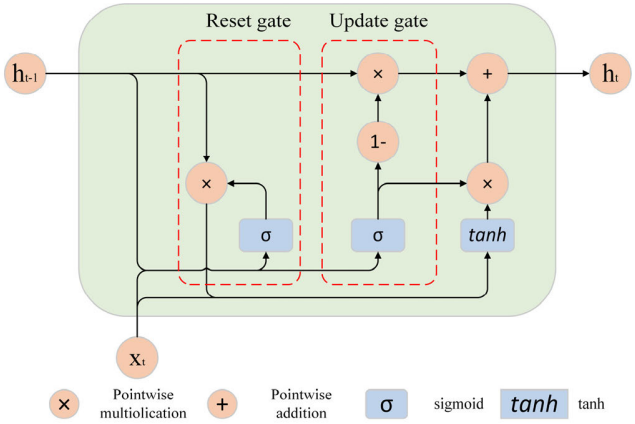

FIGURE 1. The internal structure diagram of a GRU cell at *t*-th time step.

2) The decision generating module is constructed based on SVM instead of a softmax classifier. In this way, the accuracy of decisions is improved.
3) MFO is integrated into the training process of SVM to improve the performance of SVM in handling decision classification problem.
4) EnMFO is proposed to improve the optimization capability of MFO. Firstly, to solve the lack of local exploitation capability of MFO, a new position updating method is proposed. Secondly, to solve the problem that MFO is easy to fall into a local optimal solution, Levy flight and adaptive weight are introduced into the new position updating method.

In the rest of the paper, the basic theories of our research are introduced in section II. Section III describes a novel method named EnMFO-ImGRU in detail. Then, in section IV, figures and tables of simulation experimental results show the performance of EnMFO-ImGRU. Finally, the conclusion and evaluation of EnMFO-ImGRU are shown in section V.

II. RELATED WORK

In this section, we introduce three parts of basic knowledge, namely GRU, machine learning classifier SVM, and meta-heuristic optimization algorithm MFO.

A. GRU

GRU [32] is a deep neural network for problems of sequence data. Different from the hidden unit structure of traditional RNN, GRU uses gated unit structures to deal with long-term dependence problems. GRU retains memory information and selectively forgets unimportant information, modeling long-term sequence data. Moreover, the problem of gradient explosion and gradient disappearance is reduced.

Fig. 1 shows the internal structure and variable flow of a GRU cell.

Same as RNN, GRU uses a parameter called max time step to split the input matrix into batches. For a certain time step t, x_t is the input vector representing the instant data to be processed by GRU, and h_{t-1} is the output of GRU at the last

time step, which contains previous memorizes. A GRU cell has a reset gate and an update gate. Reset gate determines how much previous information to ignore, and update gate controls the degree of bringing previous memories to the current state. Reset gated signal r_t and update gated signal z_t are calculated as follows.

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right) \tag{1}$$

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right) \tag{2}$$

where t means the current time step, W_z means the weight matrices of the update gate, and W_r means the weight matrices of the reset gate. σ is Sigmoid activation function and is defined as follows.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{3}$$

Reset gate and update gate map the input to a set of [0,1] vectors as gated signals. The candidate hidden state h_t^* is calculated as follows.

$$h_t^* = \tanh\left(W \cdot [r_t \times h_{t-1}, x_t]\right) \tag{4}$$

where W means the weight matrix, *tanh* is the activation function and is defined as follows.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{5}$$

h_t is the output and is calculated based on selective retention of h_{t-1} and selective forgetting h_t^* as follows.

$$h_t = (1 - z_t) \times h_{t-1} + z_t \times h_t^* \tag{6}$$

B. SVM

SVM is a linear classifier defined on the feature space, which makes the largest interval between the positive and negative samples [41]. For a set of training input samples as follows.

$$\left\{(x_1, y_1), \left(x_{2,} y_2\right), \ldots, (x_k, y_k)\right\}, \quad x \in R^n, \ y \in R \tag{7}$$

where x indicates the feature of the sample, y indicates the category of the sample, n indicates the dimensions of the feature vector, and k indicates the amounts of samples.

A linear method is described as follows.

$$f(x) = \omega^T x + b \tag{8}$$

where ω is the normal vector, which determines the direction of the hyperplane, and b is the displacement term, the distance between the hyperplane and the sample point.

To ensure that the hyperplane makes the max interval between positive and negative samples, some constraints are imposed to handle abnormal samples. The problem is expressed as a convex quadratic programming problem as follows.

$$\begin{cases} minimize: \dfrac{1}{2}\|\omega\|^2 + C\displaystyle\sum_{i=1}^{k}\left(\xi_i + \xi_i^*\right) > 0 \\ subject\ to \begin{cases} y_i - \omega x_i - b \le \varepsilon + \xi_i \\ \omega x + b - y_i \le \varepsilon + \xi_i^* \end{cases} \quad i = 1, \ldots, n, \ \xi_i \ge 0, \\ \qquad\qquad\qquad\qquad \xi_i^* \ge 0 \end{cases} \tag{9}$$

where ε represents the prediction results deviation from the actual target y_i. ξ_i and ξ_i^* are elastic error variables. C is the penalty coefficient that determines the tolerance for errors.

For linear inseparable problems, kernel function method is introduced into SVM to map the non-linear samples to high-dimensional space as follows.

$$\begin{cases} maximize: W\left(\alpha, \alpha^*\right) = -\dfrac{1}{2}\displaystyle\sum_{ij=1}^{n}\left(\alpha_i - \alpha_i^*\right)\left(\alpha_j - \alpha_j^*\right) \\ \qquad\qquad K\left(x_i x_j\right) \\ subject\ to \begin{cases} \displaystyle\sum_{i=1}^{n}\left(\alpha_i - \alpha_i^*\right) = 0 \quad i = 1, \ldots, n \\ 0 \le \alpha_i, \alpha_i^* \le C \end{cases} \end{cases} \tag{10}$$

where α means a Lagrangian multiplier estimated by quadratic programming methods [42], $K\left(x_i x_j\right)$ is kernel function. The settlement to the optimization task of dual variables is defined as follows.

$$\alpha^* = (\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*)^T \tag{11}$$

The function which calculates the classification decisions results is defined as follows.

$$f(x) = sign\left(\sum_{i=1}^{n}(\alpha_i - \alpha_i^*)K(x_i x_j) + b\right) \tag{12}$$

where *sign* is a linear symbolic function whose value is either 1 or -1. *sign* is defined as follows.

$$sign\,(X) = \begin{cases} 1, & X \ge 0 \\ -1, & X < 0 \end{cases} \tag{13}$$

In this way, the non-linearly separable problem is mapped to a high-dimensional space and transformed into a linearly separable problem.

C. MFO

Mirjalili proposed the biological heuristic optimization algorithm MFO in 2015 [38]. In MFO, each moth updates position around the flame searching for a better solution. A logarithmic spiral curve is introduced to define the position updating method as follows.

$$S\left(M_i, F_j\right) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + F_j \tag{14}$$

where D_i is the distance between the position coordinates of j-th flame and the position coordinates of i-th moth, b is a constant which is used to define the shape of a logarithmic spiral, and t is a random number in $[-1,1]$. F_j represents the position coordinates of j-th flame.

D_i is calculated as follows.

$$D_i = \left|F_j - M_i\right| \tag{15}$$

where M_i represents the position coordinates of i-th moth.

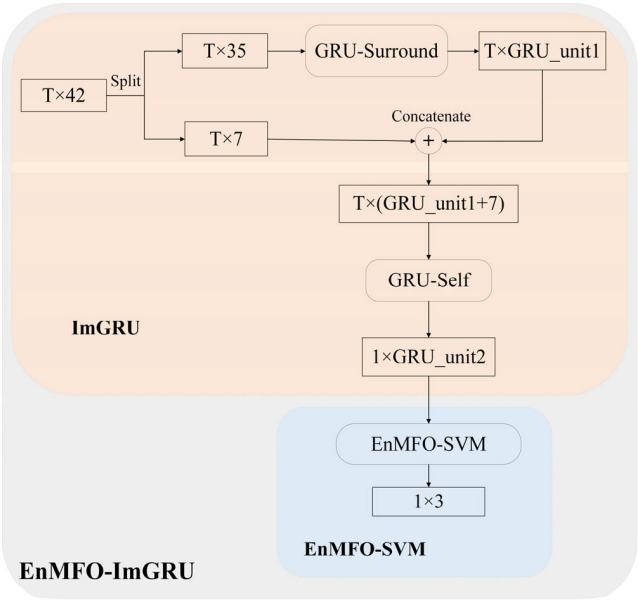

FIGURE 2. Architectural diagram of EnMFO-ImGRU. The input is firstly split into two parts, respectively representing the state of surrounding vehicles and self-vehicle in the past T time steps. ImGRU extracts features and then EnMFO-SVM generates decisions.

To balance global exploration and local exploitation capability, MFO utilizes a flame reduction mechanism. The mechanism is defined as follows.

$$flame\_no = round\left(N - t \times \frac{N-1}{T}\right) \tag{16}$$

where t indicates the current number of iterations, T indicates the upper limit of the number of iterations, and N indicates the amounts of flames.

III. METHODOLOGY

Fig. 2 shows the structure of EnMFO-ImGRU, which is proposed based on GRU, SVM, and MFO. In part A, ImGRU is designed basing on GRU to extract key features of vehicles in the traffic flow. In part B, SVM with radial basis function (RBF) kernel is combined with ImGRU to generate driving decisions. In part C, the enhanced MFO (EnMFO) is introduced into SVM training to optimize parameters of RBF kernel function.

A. ImGRU

Fig. 3 shows the situation of multiple lanes and multiple traffic participants on the road. Two factors affect the driving decisions of self-vehicle. On the one hand, decisions of self-vehicle are greatly influenced by the behavior of surrounding vehicles in five positions, including front and rear vehicles from left and right lanes and the front vehicle. On the other hand, the driving state of self-vehicle in the past short period of time determines the adaptability of self-vehicle to decisions generated by EnMFO-ImGRU. Therefore, ImGRU is designed based on GRU-Surround and GRU-Self, which are two parallel GRU cells. ImGRU uses different parameters and

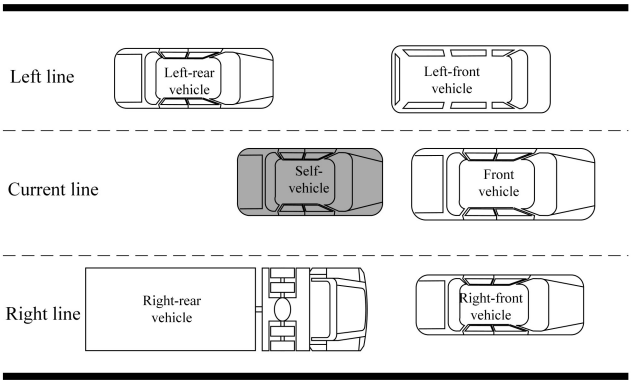

FIGURE 3. Surrounding vehicles on the road that affect decisions of self-vehicle. The input of ImGRU contains the state of the front and rear vehicles from left and right lanes, and the front vehicle and self-vehicle.

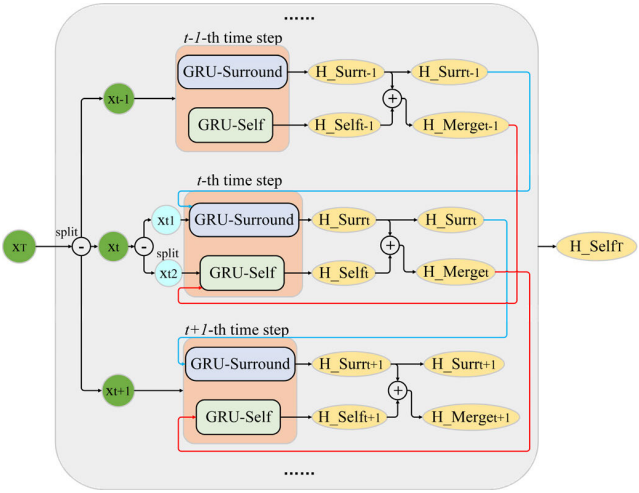

FIGURE 4. The diagram of ImGRU expanded upon T time steps. The input of ImGRU represents the state of vehicles, the output of ImGRU represents real-time feature extracting results.

weight matrices to extract features of surrounding vehicles and self-vehicle.

Fig. 4 shows the structure of ImGRU expanded by time step. X_T is the input of ImGRU and H_Self_T is the output of ImGRU. T indicates the max time step of ImGRU. The max time step is a key parameter of ImGRU, which determines how much previous inputs are related to the current input. X_T represents the driving state matrix of surrounding vehicles in the past T time and is split into T parts in time-sequential order. The driving state includes vehicle ID, lane number, speed, acceleration, lateral position, longitudinal position, and steering angle. Width and length of vehicles are considered into lateral and longitudinal position. At the t-th time step, ImGRU performs feature extraction on X_t. H_Merge_{t-1} is a hidden state matrix, which represents the feature extraction result of the ImGRU cell at the t-1-th time step.

In ImGRU cell, X_t is firstly split into X_{t1} and X_{t2}, and then X_{t1} and X_{t2} are fed to GRU-Surround and GRU-Self, respectively. At the t-th time step, H_Surr_t is the feature extraction

result of GRU-Surround and H_Self_t is the feature extracting result of GRU-Self. As the iteration progresses, the parameters and weight matrices of GRU-Self and GRU-Surround are continuously adjusted to preserve long-term memories.

Moreover, GRU-Surround and GRU-Self are not only two parallel unrelated GRU cells. Considering that the state of surrounding vehicles influences driving decisions of the self-vehicle, H_Merge_t is formed by concatenating H_Surr_t to H_Self_t and is fed to GRU-Self as input at the $t+1$-th time step.

To sum up, ImGRU contains two parts GRU-Self and GRU-Surround for feature extracting. GRU-Self extracts feature by long-term memories of GRU-Self and GRU-Surround. GRU-Surround assists GRU-Self to generate accurate and reasonable feature extracting results. In this way, the generated driving decisions ensure self-vehicle avoids surrounding vehicles and smoothly deals with uncertainty in traffic.

B. SVM WITH RBF KERNEL FUNCTION

SVM with RBF kernel function is introduced into EnMFO-ImGRU as the decision generating module. C and γ are two key parameters in SVM training process.

The penalty coefficient C indicates the tolerance to errors and determines the generalization ability of SVM. An inappropriate value of C leads to overfitting or underfitting.

Kernel function performs high-dimensional mapping on low-dimensional samples, and γ determines the number of support vectors in the new feature high-dimensional space, and affects training speed. The RBF kernel function is showed as follows.

$$K\left(x_i, x_j\right) = \exp\left(-\gamma \left\|x_i - x_j\right\|^2\right), \quad \gamma > 0,\ i \neq j \tag{17}$$

where x_i and x_j are arbitrary different samples point in feature space.

C. EnMFO-SVM

Traditional methods find the values of C and γ with the grid search method. However, it is difficult to define the search step size. A small search step leads to a local optimum solution, and a large search step brings poor development capability. Hence, the enhanced MFO (EnMFO) is applied to optimize C and γ. Each moth means a set of solutions to C and γ.

MFO assumes a superellipse in all directions of each flame. It ensures that the next position one moth may reach is inside the space. (14) shows that the optimization process of MFO depends on the flame position and spiral trajectory, and (16) reduces the number of flames with iteration. Thus, earlier in the iteration, moths update position more affected by the spiral trajectory. Later in the iteration, as the number of flames decreases, moths update position largely depends on the flame position.

To balance the influence of spiral track and number of flames on position updating of moths, an adaptive weight ω is added to the logarithmic spiral curve. ω decreases as the moth approaching the optimal solution, leading to enhance ability to search the space near the optimal solution.

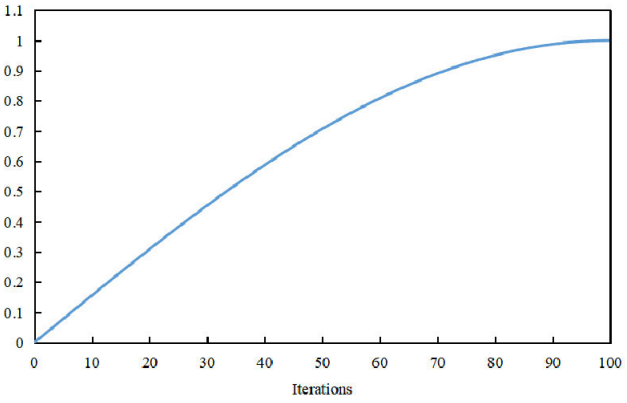

FIGURE 5. When L is set to 100, the trend chart of ω.

After introducing the adaptive weight, the position updating method is defined as follows.

$$S(M_i, F_j) = D_i \cdot e^{bt} \cdot \cos(2\pi t) + \omega \cdot F_j \tag{18}$$

where ω is defined as follows.

$$\omega = \sin(\frac{\pi \cdot l}{2 \cdot L}) \tag{19}$$

where l is the current iteration number, and L is the upper limit of the number of iterations.

The graph of ω is presented in Fig. 5. The ordinate represents the value of ω and the abscissa represents the number of iterations.

Combining (19) and Fig. 5, it can be seen that ω increases as l increases. When l is equal to L, ω is close to 1. Position of the flame becomes more important for the moth updating position with iteration.

In order to avoid falling into local optimum solution, searching agents follow Levy flight during the position updating process of EnMFO. Levy flight is a non-Gaussian stochastic process with steps following Levy distribution, which is defined as follows.

$$Levy(s) \sim |s|^{-1-\beta}, \quad 0 < \beta \leq 2 \tag{20}$$

where β is an index and s is the step size of Levy flight. s is calculated as follows.

$$s = \frac{\mu}{|\upsilon|^{\frac{1}{\beta}}}, \quad \mu \sim N(0, \sigma_\mu^2),\ \upsilon \sim N(0, \sigma_\upsilon^2) \tag{21}$$

where u and v follow a normal distribution as follows.

$$\sigma_\mu = \left\{ \frac{\Gamma(1+\beta) \cdot \sin(\frac{\pi\beta}{2})}{\beta \cdot \Gamma(\frac{1+\beta}{2}) \cdot 2^{\frac{\beta-1}{2}}} \right\}^{\frac{1}{\beta}}, \quad \sigma_\upsilon = 1 \tag{22}$$

where Γ is the standard gamma function. According to (18) and (20), the position updating method of moths $S\left(M_i, F_j\right)$ is modified as follows.

$$S\left(M_i, F_j\right) = D_i \cdot e^{bt} \cdot \cos\left(2\pi t\right) \cdot Levy\left(s\right) + \omega \cdot F_j \tag{23}$$

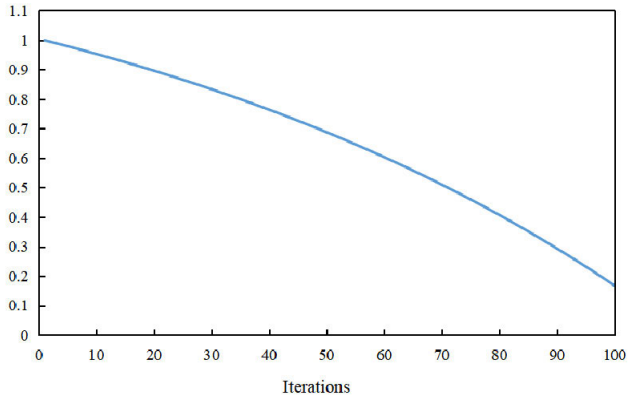

FIGURE 6. When L is set to 100, the trend chart of $\lambda(l)$.

However, MFO possesses the capability of global exploration but lacks local exploitation capabilities for developing search space. Thus, another position updating method is defined as follows.

$$S\left(M_i, F_j\right) = D_i \cdot \varepsilon \cdot Levy\left(s\right) + \omega \cdot F_j \tag{24}$$

where ε is a random number between 0 and 1. (24) makes moths move in the shortest distance to the best position along a certain direction.

To balance the capability of global exploration and local exploitation capability, $\lambda\ (l)$ is designed to determine whether (23) or (24) is selected during each iteration. $\lambda\ (l)$ decreases with iteration and is defined as follows.

$$\lambda(l) = \frac{3 - e^{\frac{l}{L+1}}}{2} \tag{25}$$

where l is the current iteration number, and L is the upper limit of the number of iterations.

The graph of $\lambda\ (l)$ is presented in Fig. 6.

As shown in Fig. 6, $\lambda\ (l)$ decreases steadily as the iteration progresses. In the early stage, EnMFO performs more global optimization, and local optimization is gradually increased as with iteration.

The final method of position updating of EnMFO is summarized as follows.

$$S(M_i, F_j) = \begin{cases} D_i \cdot e^{bt} \cdot \cos(2\pi t) \cdot Levy(s) + \omega \cdot F_j, \\ \qquad\qquad\qquad\qquad random \leq \lambda(l) \\ D_i \cdot \varepsilon \cdot Levy(s) + \omega \cdot F_j, \quad random > \lambda(l) \end{cases} \tag{26}$$

Accuracy of the classification results of SVM is used as the value of fitness function and is defined as follows.

$$T = \frac{right}{total} \times 100\% \tag{27}$$

where *right* represents the number of samples with correct classification results, and *total* represents the amount of all samples.

Fig. 7 shows the flowchart of EnMFO.

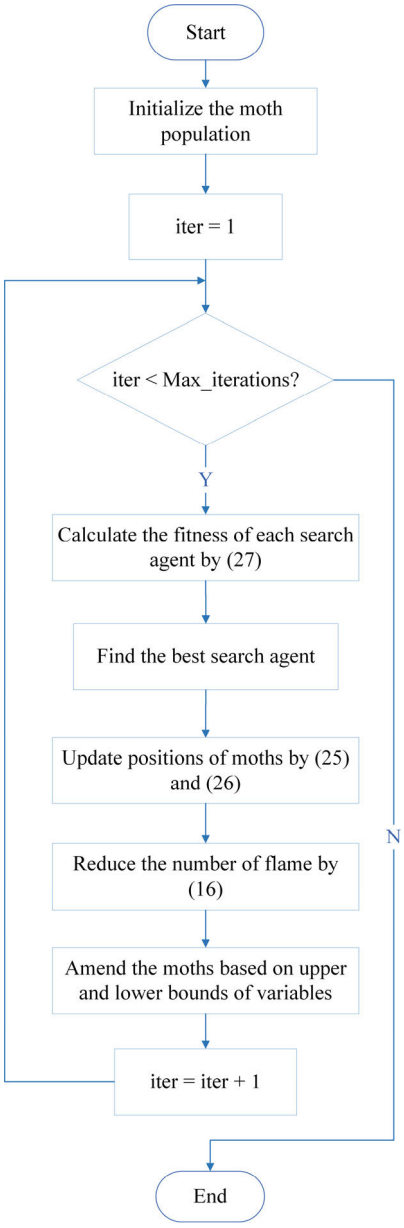

FIGURE 7. The flowchart of EnMFO.

IV. THE OVERALL PROCESS OF THE EnMFO-ImGRU

Fig. 8 shows the flowchart of EnMFO-ImGRU. First, EnMFO-ImGRU preprocesses the collected data and splits state matrices of surrounding vehicles and self-vehicle. Second, ImGRU performs feature extracting on input data. Next, EnMFO-SVM randomly initializes the positions of moths, which represent a set of SVM parameters C and γ. Then, during the iteration of SVM, the accuracy of classification results is used as the fitness function of EnMFO to optimize C and γ. In this way, optimal parameters C and γ are obtained. Finally, EnMFO-ImGRU generates behavior decision-making results in the form of one-hot vector.

V. EXPERIMENTS AND ANALYSIS

In this section, the NGSIM dataset is used to validate the performance of EnMFO-ImGRU.

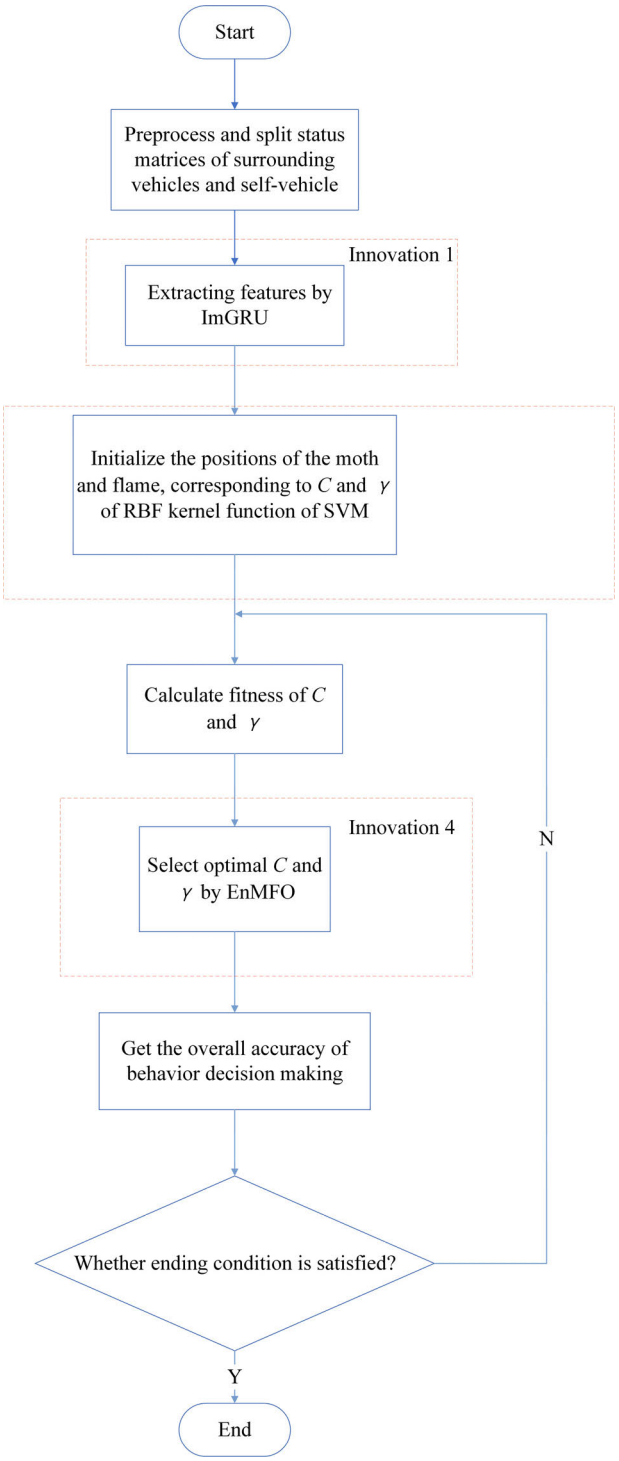

FIGURE 8. The flowchart of EnMFO-ImGRU.

The task of data set preprocessing and task of EnMFO-ImGRU training are performed by PyCharm Community Edition 2019.3 and MATLAB R2016a in the environment of windows10 on laptop PC Intel Core i7-7700HQ CPU, NVIDIA GeForce GTX 1050 Ti GPU, 2.8GHz processor, and 16GB memory.

TABLE 1. Data attributes of the preprocessed NGSIM.

Data name	Data description	Unit
Vehicle ID	Unique vehicle identification number in the area.	-
Sampling interval	Data collection interval.	*0.1s*
Lane ID	The current lane position of the vehicle. Lane 1 is the left-most lane and lane 5 is the right-most lane.	-
Global X	X-coordinate of vehicle front center based on CA State Plane III in NAD83.	*feet*
Global Y	Y-coordinate of vehicle front center based on CA State Plane III in NAD83.	*feet*
Vehicle length	Length of the vehicle.	*feet*
Vehicle width	Width of the vehicle.	*feet*
Speed	Instantaneous speed of the vehicle.	*feet / s*
Acceleration	Instantaneous acceleration of the vehicle.	$feet/s^2$
Steering angle	The steering angle of the vehicle in 0.3s.	*degree*

A. DATASET AND PROCESSING

The NGSIM dataset includes trajectories of vehicles traveling on the freeway between the Ventura Avenue and the Cavanga Avenue exit ramp south of the US 101 freeway in Los Angeles, California [43]. This data set has three data subsets, all of which are collected within 15 minutes, and the sampling interval is 0.1s.

Instead of using a real vehicle model, the public traffic flow dataset NGSIM is used to train EnMFO-ImGRU on the simulation model to verify effectiveness and reliability of EnMFO-ImGRU.

NGSIM does not contain vehicle steering angle information. Horizontal and vertical coordinates of three consecutive moments of the vehicle are used to calculate the steering angle [29]. The steering angle is calculated as follows.

$$\alpha = arctan\left(\frac{x_t - x_{t-3}}{y_t - y_{t-3}}\right) \times \frac{180}{\pi} \tag{28}$$

where x and y are horizontal and vertical positions of the calculated vehicle at an instant, and t is the real-time sampling time point. The sampling interval is set to 3.

The attributes of the preprocessed NGSIM are shown in Table. 1, all of which have an impact on driving decisions.

B. PARAMETERS SETTING

NGSIM dataset provides continuous instantaneous data at 0.1s intervals. Hence, the max time step of ImGRU is set to 20, which means that the memories in the past 2 seconds are referenced to generate decisions.

According to NGSIM, the behavior of vehicles is divided into three realities: left-turn lane changing, right-turn lane changing, and lane-keeping. In particular, some slight oscillations in the graph indicate that the vehicle adjusts driving style

TABLE 2. Parameter steeings.

Algorithms	Parameter	Description	Value
ImGRU	*batch size*	Batches per training	500
	hidden units of GRU-Self	Number of hidden neural units of GRU-Self	70
	hidden units of GRU-Surround	Number of hidden neural units of GRU-Surround	100
	learning rate	Learning rate	0.0001
	optimizer	Minimize loss	Adam optimizer
	loss function	Instruct GRU to update weight parameters	categorical_ crossentropy
	max time step	GRU blocks expanded by time	20
SVM	*kernel function*	Implicit mapping of feature space	RBF
	Cmax	Maximum number of C	100
	Cmin	Minimum number of C	0.01
	Gmax	Maximum number of γ	0.1
	Gmin	Minimum number of γ	0.001
EnMFO	*Num*	Number of search agents	30
	ub	Upper bound of searching space	100
	lb	Lower bound of searching space	-100

according to the road condition and keep driving straight in the original lane rather than changing lanes.

EnMFO-ImGRU was completed in the environment of TensorFlow 1.1.3 and Keras 1.3.0. Table. 2 introduces the parameter settings.

C. COMPARISON OF RESULTS

In this section, two tasks are completed. First, the performance of EnMFO-ImGRU on the NGSIM dataset is evaluated. Then, the accuracy of the results of EnMFO-ImGRU is compared with other methods.

For EnMFO-ImGRU, a training set of 48000 pieces of data and a testing set of 3600 pieces of data are provided. Among them, the data ratio of vehicles performing left-turn lane changing, right-turn lane changing, and lane-keeping is 1:1:1. To evaluate the performance of generated decisions, accuracy of decision result is used as an indicator.

Fig. 9 shows the processing results of RNN, LSTM, GRU, and ImGRU behavior decision-making models on NGSIM. The maximum number of iterations of the three methods is set to 120. When methods are iterated between 1 and 20, the accuracy of the results of four models increase as iterations progress, but still low. When methods are iterated between 21 and 40, accuracy of ImGRU method increases slowly and reaches the highest value of the four methods. Accuracy of other three methods still increase fast. When methods are iterated between 41 and 90, the accuracy of RNN

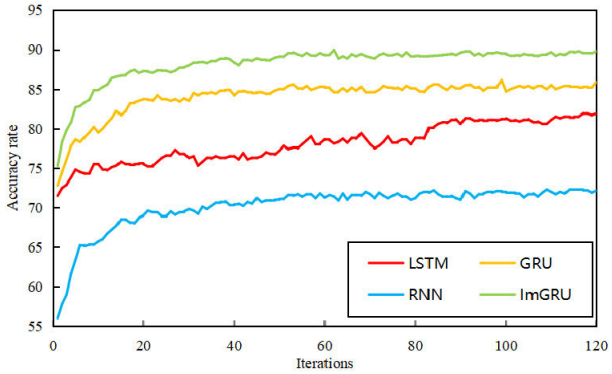

FIGURE 9. RNN, LSTM, GRU and ImGRU behavior decision-making accuracy curve graph within 120 iterations.

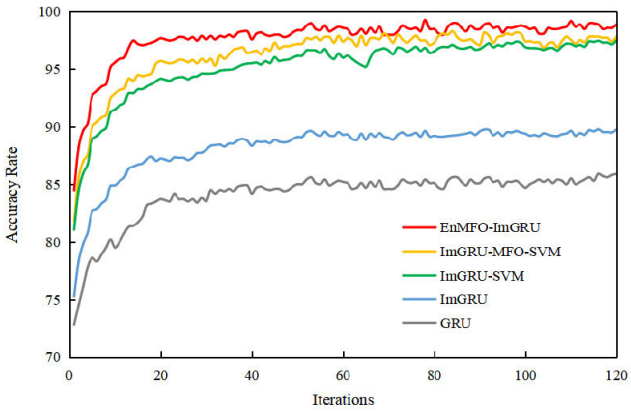

FIGURE 10. GRU, ImGRU, ImGRU-SVM, ImGRU-MFO-SVM and EnMFO-ImGRU behavior decision-making accuracy curve graph within 120 iterations.

and GRU are relatively stable. For LSTM, the fact of too many parameters causes the result accuracy to increase faster than RNN method and GRU method. For ImGRU, in the later stage of iterations, the accuracy of result fluctuates little and tend to be stable. When methods are iterated between 91 and 120, the accuracy of the results of the three network models stable at their maximum value. When methods are iterations equals 120, the RNN network obtains an accuracy of 72.16%, the LSTM network obtains an accuracy of 81.94%, the GRU network obtains an accuracy of 85.89%, and the ImGRU network obtains an accuracy of 89.78. Simulation experimental results show the advantage of our proposed EnMFO-ImGRU in building models based on GRU rather than LSTM and RNN. GRU brings a higher accuracy than RNN and LSTM, and the result accuracy of GRU reaches and dynamically maintains the maximum value faster than RNN and LSTM. In addition, accuracy of ImGRU increases faster in the earlier stage of iteration and is higher than other methods during the whole iteration process, which confirms the excellent performance of ImGRU as the feature extraction module.

Fig. 10 shows the processing results of GRU, ImGRU, ImGRU-SVM, ImGRU-MFO-SVM, and EnMFO-ImGRU within 120 iterations on NGSIM. When methods are iterated between 1 and 20, the results accuracy of the five network

models increases rapidly, but the overall accuracy is low. Compared with GRU and ImGRU, the accuracy of the other three networks with SVM increased faster, indicating that SVM performs better than softmax in terms of training speed and result accuracy. When methods are iterated between 21 and 60, result accuracy of GRU, ImGRU, ImGRU-SVM, and ImGRU-MFO-SVM are gradually approaching their maximum value, but the speed of accuracy improvement is slow. Softmax classifier in GRU and ImGRU is a fully connected layer with softmax activation function. Therefore, later in the iteration, the accuracy of GRU and ImGRU reach the maximum but are still lower than models with SVM. For ImGRU-SVM and ImGRU-MFO-SVM, the training time in each iteration is longer than the training time of GRU and ImGRU because the process of training SVM and adjusting parameters are more complicated. Hence, ImGRU-SVM and ImGRU-MFO-SVM do not reach their maximum value of accuracy when the number of iterations equals 60. The accuracy of EnMFO-ImGRU results fluctuates steadily at a high level, which shows that EnMFO-ImGRU performs well in both the accuracy of generating results and the speed of accuracy increasing. When methods are iterated between 61 and 120, the accuracy of the five network models fluctuates steadily near the highest value. When the number of iterations equals 120, the accuracy of GRU reaches 85.89%, the accuracy of ImGRU reaches 89.78%, the accuracy of ImGRU-SVM reaches 95.19%, the accuracy of ImGRU-MFO-SVM reaches 96.03%, and the accuracy of EnMFO-ImGRU reaches 97.79%.

When comparing the groups of comparative simulation experiments in Fig. 10, the superiority of EnMFO-ImGRU improvement points is verified. First, the growth trends of ImGRU network and GRU network are similar, but the accuracy of ImGRU is higher than that of GRU during the entire iteration. When the number of iterations equals 120, the accuracy of ImGRU is 3.89% higher than that of GRU. The comparative experiment of GRU and ImGRU proves the superiority of using ImGRU as the feature extracting module. Second, the accuracy of ImGRU-SVM is 5.41% higher than that of ImGRU when the number of iterations equals 120. The comparative experiment of ImGRU and ImGRU-SVM proves the superiority of using SVM instead of softmax classifier. Third, the accuracy of ImGRU-MFO-SVM is 2.04% higher than that of ImGRU-SVM when the number of iterations equals 120. And in the early stage of the iteration, the accuracy increased faster. The comparative experiment of ImGRU-SVM and ImGRU-MFO-SVM proves the superiority of the improvements that using MFO to optimize SVM parameters C and γ. Fourth, the accuracy of EnMFO-ImGRU increases fastest among all models. Accuracy of EnMFO-ImGRU quickly reaches close to the maximum value earlier in the iteration, and the accuracy is 0.66% higher than ImGRU-MFO-SVM when the number of iterations equals 120. Although the improvement in accuracy is not obvious, 97.89% is a sufficiently high accuracy value. In general, five GRU based methods bring low accuracy and high speed of accuracy increasing in the initial stage of the iteration. In the middle of the iteration, the accuracy of results is improved with iterations, but the speed of improvement turns slow. In the later stage of the iteration, the accuracy of results no longer increase with iterations, but dynamically stabilize at the maximum value. Accuracy of results of five methods stabilized at their maximum value. In a word, EnMFO-ImGRU obtains higher accuracy in fewer iterations.

TABLE 3. The number of iterations for several methods to achieve the highest accuracy.

Algorithms	Number of iterations of at the highest accuracy	Accuracy
RNN	109	72.34%
LSTM	117	81.97%
GRU	119	85.89%
ImGRU	116	89.78%
ImGRU-SVM	120	97.50%
ImGRU-MFO-SVM	84	98.29%
EnMFO-ImGRU	**78**	**99.27%**

GRU network solves the gradient problem of RNN. Moreover, GRU solves the problem of the huge amount of calculation caused by too many parameters of LSTM, improving the speed of feature extracting. Result accuracy of GRU-based network increases faster than that of RNN-based network and LSTM-based network. In addition, compared with softmax classifier, methods that use SVM for decision generating models perform better in terms of result accuracy. SVM reduces the number of iterations and improves accuracy. EnMFO-ImGRU brings high result accuracy and speed of accuracy increasing on dealing with behavior decision-making methods.

Table. 3 shows the accuracy of the results of various networks on the NGSIM dataset and the number of iterations required to obtain the highest accuracy. Within 120 iterations, RNN obtained an accuracy of 72.34% at the 109th iteration, LSTM obtained an accuracy of 81.97% at the 117th iteration, GRU obtained an accuracy of 85.89% at the 119th iteration, ImGRU obtained an accuracy of 89.78% at the 116th iteration, ImGRU-SVM obtained an accuracy of 97.50% at the 120th iteration, ImGRU-MFO-SVM obtained an accuracy of 98.29% at the 84th iteration, EnMFO-ImGRU obtained an accuracy of 99.27% at the 78th iteration. The results in the first, second, and third lines confirm the superiority of our modeling based on GRU instead of LSTM and RNN. The results in the third and fourth lines confirm that ImGRU improves the accuracy of the decisions. The results in the fourth and fifth lines confirm that SVM brings higher accuracy than softmax classifier. The results in the fifth and sixth lines confirm that the introduction of MFO improves the performance of SVM. Moreover, using MFO to provide optimal parameters for SVM also reduce the number of iterations of ImGRU and maintain a high accuracy. The results in the last two lines confirm the superiority of the proposed EnMFO.

Table. 4 shows the results of EnMFO-ImGRU and several behavior decision-making methods on the NGSIM dataset. EnMFO-ImGRU obtains 97.79% accuracy at 120 iterations,

TABLE 4. The overall accuracy (%) of other algorithms.

Algorithms	Accuracy	References
EnMFO-ImGRU	**97.79%**	**-**
MLP	74.90%	[15]
Clustering-based MLP	86.90%	[15]
HMM	91.23%	[17]
Bayes-BPNN	71.60%	[18]
DBN	95.16%	[22]
GOA-ImLSTM	93.00%	[29]
HRC-LSTM	96.00%	[30]
Bi-LSTM	88.19%	[31]

which has 4.79% improvement over GOA-ImLSTM method, 26.19% improvement over Bayes-BPNN method, 2.63% improvement over DBN method, 22.89% improvement over MLP method, 10.89% improvement over Clustering-based MLP method, 6.56% improvement over HMM method, and 9.60% improvement over Bi-LSTM method. Comparative simulation experimental results show that the accuracy of the generated decisions results of EnMFO-ImGRU has excellent performance among the listed methods in Table. 4.

D. DISCUSSION

In this section, all improvement point of EnMFO-ImGRU are compared and finally compared EnMFO-ImGRU with existing behavior decision-making methods. The overall accuracy of generated behavior decision-making is used as an indicator. Simulation experimental results showed that EnMFO-ImGRU outperforms existing behavior decision-making methods.

The work of EnMFO-ImGRU is divided into a feature extracting module and a decision generating module. Four innovations were introduced into EnMFO-ImGRU.

Firstly, ImGRU, a double-layer GRU, is proposed and used as the feature extracting module. One layer of GRU extracts features of self-vehicle, and the other extracts features of surrounding vehicles. In each iteration, feature extraction results of the two GRUs are combined. In this way, ImGRU generates reasonable feature extraction results. Hence, it could be seen from Fig. 9 that building a feature extracting module based on GRU is better than RNN and LSTM.

Secondly, SVM is used instead of softmax as the decision generating module. Because the performance of SVM is better than softmax on classification tasks, the accuracy of decisions is improved. The comparative experiment of ImGRU and ImGRU-SVM in Fig. 10 shows that a decision generating module based on SVM brings higher accuracy than that based on softmax.

Thirdly, MFO is integrated into the training process of SVM. Because the performance of SVM largely depends on the setting of parameters, the performance of SVM is improved in this way. The comparative experiment of ImGRU-SVM and ImGRU-MFO-SVM in Fig. 10 and Table. 3 shows that MFO reduces the number of iterations of SVM and improves accuracy.

Fourthly, EnMFO is proposed based on MFO. A new position updating method is defined to promote the optimization capability of MFO. The new position updating method follows adapted weight and Levy flight to avoid EnMFO falling into local optimal solution. As shown in Fig. 10, because EnMFO is not easy to fall into local optimum solution and EnMFO converges faster than MFO, the accuracy of EnMFO-ImGRU dynamically stable at maximum when the number of iterations is less than 20, while the accuracy of ImGRU-MFO-SVM dynamically stable at maximum when the number of iterations is over 40. Accuracy of EnMFO-ImGRU is higher than accuracy of ImGRU-MFO-SVM during the whole iteration, which confirms that EnMFO performs better than MFO.

Despite the good performance, EnMFO-ImGRU still has limitations. First, compared with softmax classifier, the training process of SVM has high time complexity. Second, later in the iteration, the accuracy of EnMFO-ImGRU fluctuated around 98%, and the accuracy showed dynamic stability. In other words, with iterations, the accuracy may be lower than the maximum value reached before. In addition, advanced information such as driving habits and vehicle model is not contained in the feature extracting results.

VI. CONCLUSION

Generating safe and feasible decisions for autonomous vehicles is a challenging task due to future motion uncertainty in traffic. To confront that challenge, in this article, we presented an efficient network architecture named EnMFO-ImGRU. First, ImGRU was designed and used for separately training surrounding vehicles and self-vehicles as a feature extracting module. Second, an model based on SVM with RBF kernel function was used as decision generating module instead of softmax classifier. Third, EnMFO was proposed to optimize key parameters during the SVM training process, which increased classification accuracy and reduced the number of iterations. Excellent global exploration capability and local exploitation capability of EnMFO made the process of optimizing SVM parameters converge quickly without falling into the local optimal solution. Different from many state-of-the-art works, our method used the time correlation of GRU to solve behavior decision-making problems in traffic, increasing the accuracy of results while reducing training time.

Compared with state-of-the-art behavior decision-making methods, EnMFO-ImGRU achieves better performance on the NGSIM dataset. In future work, the type of vehicles, the length of the vehicles, and the driving habits of different types of vehicles may be added to the feature extracting module.

REFERENCES

[1] L. J. Karam, J. Katupitiya, V. Milanes, I. Pitas, and J. Ye, "Autonomous driving: Part 1-sensing and perception [From the guest editors]," *IEEE Signal Process. Mag.*, vol. 37, no. 4, pp. 11–13, Jul. 2020.

[2] W. Yang, L. Zheng, Y. Li, Y. Ren, and Z. Xiong, "Automated highway driving decision considering driver characteristics," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 6, pp. 2350–2359, Jun. 2020, doi: 10.1109/TITS.2019.2918117.

[3] R. Wang, Y. Li, J. Fan, T. Wang, and X. Chen, "A novel pure pursuit algorithm for autonomous vehicles based on salp swarm algorithm and velocity controller," *IEEE Access*, vol. 8, pp. 166525–166540, 2020, doi: 10.1109/ACCESS.2020.3023071.

[4] B. Li, H. Du, W. Li, and B. Zhang, "Dynamically integrated spatiotemporal-based trajectory planning and control for autonomous vehicles," *IET Intell. Transp. Syst.*, vol. 12, no. 10, pp. 1271–1282, Dec. 2018.

[5] K. Chen, B. Yang, X. Pei, and X. Guo, "Hierarchical control strategy towards safe driving of autonomous vehicles," *J. Intell. Fuzzy Syst.*, vol. 34, no. 4, pp. 2197–2212, Apr. 2018, doi: 10.3233/jifs-171186.

[6] H. Chae, M. Lee, and K. Yi, "Probabilistic prediction based automated driving motion planning algorithm for lane change," in *Proc. 17th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2017, pp. 1640–1645, doi: 10.23919/ICCAS.2017.8204250.

[7] J. Kim and D. Kum, "Collision risk assessment algorithm via lane-based probabilistic motion prediction of surrounding vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 9, pp. 2965–2976, Sep. 2018, doi: 10.1109/TITS.2017.2768318.

[8] J. Nilsson, M. Brannstrom, E. Coelingh, and J. Fredriksson, "Lane change maneuvers for automated vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1087–1096, May 2017, doi: 10.1109/TITS.2016.2597966.

[9] Z. Li, Q. Wu, Y. Xiao, M. Jin, and H. Lu, "Deep matching network for handwritten Chinese character recognition," *Pattern Recognit.*, vol. 107, Nov. 2020, Art. no. 107471, doi: 10.1016/j.patcog.2020.107471.

[10] S. Fan, J. Li, Y. Zhang, X. Tian, Q. Wang, X. He, C. Zhang, and W. Huang, "On line detection of defective apples using computer vision system combined with deep learning methods," *J. Food Eng.*, vol. 286, Dec. 2020, Art. no. 110102.

[11] C. Ji, Y. Li, J. Fan, and S. Lan, "A novel simplification method for 3D geometric point cloud based on the importance of point," *IEEE Access*, vol. 7, pp. 129029–129042, 2019, doi: 10.1109/ACCESS.2019.2939684.

[12] L. Li, K. Ota, and M. Dong, "Humanlike driving: Empirical decision-making system for autonomous vehicles," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 6814–6823, Aug. 2018, doi: 10.1109/TVT.2018.2822762.

[13] D. Lenz, F. Diehl, M. T. Le, and A. Knoll, "Deep neural networks for Markovian interactive scene prediction in highway scenarios," in *Proc. 4th IEEE Intell. Vehicles Symp.*, Los Angeles, CA, USA, Jun. 2017, pp. 685–692, doi: 10.1109/IVS.2017.7995797.

[14] E. Salari and D. Ouyang, "Camera-based forward Collision and lane departure warning systems using SVM," in *Proc. IEEE 56th Int. Midwest Symp. Circuits Syst. (MWSCAS)*, Columbus, OH, USA, 2013, pp. 1278–1281, doi: 10.1109/MWSCAS.2013.6674.888.

[15] G. Ren, Y. Zhang, H. Liu, K. Zhang, and Y. Hu, "A new lane-changing model with consideration of driving style," *Int. J. Intell. Transp. Syst. Res.*, vol. 17, no. 3, pp. 181–189, Sep. 2019.

[16] T. Streubel and K. H. Hoffmann, "Prediction of driver intended path at intersections," in *Proc. IEEE Intell. Vehicles Symp.*, Dearborn, MI, USA, Jun. 2014, pp. 134–139, doi: 10.1109/IVS.2014.6856508.

[17] S. Li, L. Zhang, D. Yu, and Y. Tian, "Driving lane change intent recognition based on vehicle exterior environment variables," in *Proc. 37th Chin. Control Conf. (CCC)*, Wuhan, China, Jul. 2018, pp. 9315–9319, doi: 10.23919/ChiCC.2018.8483804.

[18] L. Li, M. Zhang, and R. Liu, "The application of Bayesian filter and neural networks in lane changing prediction," in *Proc. 5th Int. Conf. Civil Eng. Transp.*, 2015, pp. 2004–2007.

[19] K. Khnissi, C. Ben Jabeur, and H. Seddik, "A smart mobile robot commands predictor using recursive neural network," *Robot. Auto. Syst.*, vol. 131, Sep. 2020, Art. no. 103593.

[20] D. Shin, H.-G. Kim, K.-M. Park, and K. Yi, "Development of deep learning based human-centered threat assessment for application to automated driving vehicle," *Appl. Sci.*, vol. 10, no. 1, p. 253, Dec. 2019.

[21] Q. Zou, H. Jiang, Q. Dai, Y. Yue, L. Chen, and Q. Wang, "Robust lane detection from continuous driving scenes using deep neural networks," *IEEE Trans. Veh. Technol.*, vol. 69, no. 1, pp. 41–54, Jan. 2020, doi: 10.1109/TVT.2019.2949603.

[22] D.-F. Xie, Z.-Z. Fang, B. Jia, and Z. He, "A data-driven lane-changing model based on deep learning," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 41–60, Sep. 2019.

[23] F. Altche and A. de La Fortelle, "An LSTM network for highway trajectory prediction," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2017, pp. 353–359, doi: 10.1109/ITSC.2017.8317913.

[24] B. Kim, C. M. Kang, J. Kim, S. H. Lee, C. C. Chung, and J. W. Choi, "Probabilistic vehicle trajectory prediction over occupancy grid map via recurrent neural network," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 399–404, doi: 10.1109/ITSC.2017.8317943.

[25] H. Q. Dang, J. Furnkranz, A. Biedermann, and M. Hoepfl, "Time-to-lane-change prediction with deep learning," in *Proc. IEEE 20th Int. Conf. Intell. Transp. Syst. (ITSC)*, Yokohama, Japan, Oct. 2017, pp. 1–7, doi: 10.1109/ITSC.2017.8317674.

[26] L. Xin, P. Wang, C.-Y. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention-aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst. (ITSC)*, Maui, HI, USA, Nov. 2018, pp. 1441–1446, doi: 10.1109/ITSC.2018.8569595.

[27] S. Chen, Y. Leng, and S. Labi, "A deep learning algorithm for simulating autonomous driving considering prior knowledge and temporal information," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 35, no. 4, pp. 305–321, Apr. 2020.

[28] R. Valiente, M. Zaman, S. Ozer, and Y. P. Fallah, "Controlling steering angle for cooperative self-driving vehicles utilizing CNN and LSTM-based deep networks," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Paris, France, Jun. 2019, pp. 2423–2428, doi: 10.1109/IVS.2019.8814260.

[29] Y. Shi, Y. Li, J. Fan, T. Wang, and T. Yin, "A novel network architecture of decision-making for self-driving vehicles based on long short-term memory and grasshopper optimization algorithm," *IEEE Access*, vol. 8, pp. 155429–155440, 2020, doi: 10.1109/ACCESS.2020.3019048.

[30] X. Zhang, J. Sun, X. Qi, and J. Sun, "Simultaneous modeling of car-following and lane-changing behaviors using deep learning," *Transp. Res. C, Emerg. Technol.*, vol. 104, pp. 287–304, Jul. 2019.

[31] O. Scheel, L. Schwarz, N. Navab, and F. Tombari, "Situation assessment for planning lane changes: Combining recurrent models and prediction," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Brisbane, QLD, Australia, May 2018, pp. 2082–2088, doi: 10.1109/ICRA.2018.8462838.

[32] K. Cho, V. M. Bart, G. Caglar, B. Dzmitry, B. Fethi, S. Holger, and B. Yoshua, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, Doha, Qatar, Oct. 2014, pp. 1724–1734, doi: 10.3115/v1/D14-1179.

[33] A. Benterki, V. Judalet, M. Choubeila, and M. Boukhnifer, "Long-term prediction of vehicle trajectory using recurrent neural networks," in *Proc. 45th Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Lisbon, Portugal, Oct. 2019, pp. 3817–3822, doi: 10.1109/IECON.2019.8927604.

[34] R. Fei, S. Li, X. Hei, Q. Xu, F. Liu, and B. Hu, "The driver time memory car-following model simulating in apollo platform with GRU and real road traffic data," *Math. Problems Eng.*, vol. 2020, pp. 1–18, Mar. 2020.

[35] Y. Gu, W. Lu, L. Qin, M. Li, and Z. Shao, "Short-term prediction of lane-level traffic speeds: A fusion deep learning model," *Transp. Res. C, Emerg. Technol.*, vol. 106, pp. 1–16, Sep. 2019.

[36] X. Jiang and S. Li, "BAS: Beetle antennae search algorithm for optimization problems," *Int. J. Robot. Control*, vol. 1, no. 1, p. 1, Apr. 2018.

[37] J. Kennedy and R. C. Eberhart, *Swarm Intelligence*. San Mateo, CA, USA: Morgan Kaufmann, 2001.

[38] S. Mirjalili, "Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm," *Knowl.-Based Syst.*, vol. 89, pp. 228–249, Nov. 2015.

[39] C. Yu, A. A. Heidari, and H. Chen, "A quantum-behaved simulated annealing algorithm-based moth-flame optimization method," *Appl. Math. Model.*, vol. 87, pp. 1–19, Nov. 2020.

[40] X. Zhao, Y. Fang, L. Liu, M. Xu, and P. Zhang, "Ameliorated moth-flame algorithm and its application for modeling of silicon content in liquid iron of blast furnace based fast learning network," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106418.

[41] C. Cortes and V. Vapnik, "Support vector machine," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, 1995.

[42] V. N. Vapnik, "An overview of statistical learning theory," *IEEE Trans. Neural Netw.*, vol. 10, no. 5, pp. 988–999, Sep. 1999.

[43] *Next Generating Simulation (NGSIM) Vehicle Trajectories and Supporting Data*. Accessed: Jan. 2020. [Online]. Available: https://data.transportation.gov/Automobiles/Nextgenerating-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj

TAIQIAO YIN received the B.S. degree from the Software Institute, Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree. His research interests include artificial intelligence, self-driving vehicles, and driving decision-making.

YING LI received the B.S., M.S., and Ph.D. degrees from Jilin University. From 2000 to 2006, she was an Associate Professor with the Department of Space Information Processing, Jilin University. Since 2006, she has been a Professor in computer application technology with Jilin University. She has published more than 60 papers in journals and international conference. Her research interests include big data, 3-D visual modeling, 3-D image processing, machine vision, and machine learning. She is currently a Fellow of the China Computer Federation.

JIAHAO FAN received the B.S. degree from the Computer Science and Technology College, Jilin University, Changchun, China, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include swarm intelligence algorithm, machine learning, image processing, data mining, and 3-D data processing.

TAN WANG received the B.S. degree in journalism and communication from the Jilin University of Finance and Economics, Changchun, China, in 2017, and the M.S. degree from Northeast Normal University, in 2020. She is currently working (part-time) with Space Technology (Jilin) Company Ltd. Her research interests include media technology ethics, network ecological, and feature selection.

YUNXIA SHI received the B.S. degree from the Computer Science and Technology Institute, Dalian Minzu University, Dalian, China, in 2018. She is currently pursuing the master's degree with Jilin University, Changchun, China. Her research interests include artificial intelligence, self-driving vehicles, and driving decision-making.