IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 11, 2020, accepted August 19, 2020, date of publication August 24, 2020, date of current version September 4, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3019048

A Novel Network Architecture of Decision-Making for Self-Driving Vehicles Based on Long Short-Term Memory and Grasshopper Optimization Algorithm

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

[1]College of Computer Science and Technology, Jilin University, Changchun 130012, China
[2]Key Laboratory of Symbolic Computation and Knowledge Engineering of Ministry of Education, Jilin University, Changchun 130012, China
[3]Space Technology (Jilin) Company Ltd., Jilin 132013, China
[4]College of Software, Jilin University, Changchun 130012, 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 Long short-term memory network is one of the most important network architectures of decision-making for self-driving vehicles. Nevertheless, the decision-making accuracy of long short-term memory network is limited, the information of the surrounding vehicles is not taken into consideration, which is critical for the decision-making of the ego vehicle, and the classification capability of long short-term memory network is poor. In this article, a novel network architecture called improved long short-term memory network with support vector machine classifier optimized by grasshopper optimization algorithm (GOA-ImLSTM) is proposed. Three improvements are presented in GOA-ImLSTM. Firstly, to consider the information of the surrounding vehicles, a new network architecture, used to extract vital features for self-driving vehicles, with three parallel long short-term memory network units and a network unit serial connected according to vehicle location is designed. Secondly, to improve classification accuracy, support vector machine with stronger classification capability than softmax is introduced to accomplish the classification task. Thirdly, to promote the classification capability of support vector machine, grasshopper optimization algorithm is employed to optimize the parameters of support vector machine. Moreover, to balance exploration and exploitation ability of grasshopper optimization algorithm, dynamic weights in position movement formula are defined. The experiments indicate that GOA-ImLSTM improves the accuracy of results compared with other decision-making methods for self-driving vehicles on the Next Generation SIMulation.

INDEX TERMS Grasshopper optimization algorithm, long short-term memory, self-driving decision-making, support vector machine.

I. INTRODUCTION

With the continuous breakthrough of artificial intelligence, autonomous vehicles have moved towards practicality [1]. The self-driving system is a highly autonomous system, including perception module [2], path planning module [3], behavior decision module [4], and adaptive control module [5]. The behavior decision module is the key technology to determine the safety and stability of autonomous vehicles [6]. Recently, research on driving decision-making for autonomous vehicles has made great strides. Main decision-making algorithms are divided into three categories, rule-based methods, reinforcement learning methods [7], and deep learning methods [8].

For rule-based methods, making autonomous driving decisions by self-definition rules on modeling or the features extracted by neural networks as rules. Furda and Vlacic [9] used multiple criteria decision making (MCDM) to select the

The associate editor coordinating the review of this manuscript and approving it for publication was Vivek Kumar Sehgal.

most appropriate driving maneuver from a set of feasible driving maneuvers, which are obtained through Petri net on vehicle environment model. Chong *et al.* [10] proposed a fuzzy rule-based neural network model to obtain driver behavior rules from individual vehicle trajectory data. Barman *et al.* [11] designed a fuzzy inference system to generate fuzzy rules and define a nonlinear mapping of input data. Li *et al.* [12] chose T-S fuzzy neural network (TSFNN) to establish driving decision-making model under emergency situations. Despite the high accuracy of simple scenarios in the rule-based methods, the complexity of the rule-making method limits its further development in complicated environments.

For reinforcement learning methods, obtaining optimal strategies by maximizing long-term future rewards. Ngai and Yung [13] employed a multiple-goal reinforcement learning framework to determine action decisions by considering seven different goals. Gindele *et al.* [14] proposed a novel approach, which is based on a continuous partially observable Markov decision process (POMDP), to obtain decision-making for autonomous driving. Song *et al.* [15] first developed a continuous hidden Markov model to predict the motion intention, then built a POMDP to model the general decision-making framework. You *et al.* [16] modeled the autonomous vehicle as a stochastic Markov decision process (MDP) and considered the driving style of an expert driver as the target to be learned by reinforcement learning and inverse reinforcement. Chen *et al.* [17] proposed a fuzzy Markov prediction model which can estimate the short-term traffic conditions in urban environment. Although the reinforcement learning methods made a safe and efficient driving decision in uncomplicated condition, the complexity of truly driving environments makes the accuracy of decision making is limited.

For deep learning methods, many deep networks are used in driving decision-making, such as convolutional neural network (CNN) [18], recurrent neural network (RNN) [19], bidirectional recurrent neural network (BiRNN) [20], long short-term memory network (LSTM) [21] and gated recurrent unit (GRU) [22]. Gao *et al.* [23] systematically evaluated the performance of deep learning features in view-based 3D model retrieval on four popular datasets. Li *et al.* [24] proposed a CNN model to detect, recognize and abstract the information in the input road scene, then a decision-making system calculates the specific commands to control the vehicles. Nevertheless, CNN extracts features without time correlation. In this way, the features extracted from network for decision-making limit the accuracy rate of driving decision-making. To introduce time correlation, a RNN employed for driving decision making. Chen *et al.* [25] pretended a novel model for self-driving cars which comprises a CNN, a cognitive map and a RNN to learn the driving behavior and decision-making process of a human driver. However, a RNN is unable to resolve long-term dependencies. To address long-term dependencies in decision making, LSTM adapted in autonomous driving. Chen *et al.* [26] developed a deep CNN–LSTM algorithm for self-driving simulation to control the movement of self-driving vehicles in the driving simulation. However, the memory ability of important features and classification capability of LSTM is poor.

Softmax classifier is used in most deep learning methods. However, the poor classification ability of the softmax classifier leads to poor classification results. Support vector machine (SVM) [27] is introduced because of its stronger classification ability. The classification results of SVM are largely affected by the parameters of kernel function. Metaheuristic optimization algorithm is applied to optimize the parameters of kernel function.

For metaheuristic optimization algorithm, Kennedy and Eberhart [28] proposed Particle Swarm Optimization (PSO), which is inspired by the act of searching for food, each member of the swarm continuously changes its search pattern by learning from its own experience and the experience of other members. Boettcher and Percus [29] proposed extremal optimization (EO), its inspiration is self-organized criticality, a concept introduced to describe emergent complexity in physical systems. Hashim *et al.* [30] proposed a novel metaheuristic algorithm named Henry gas solubility optimization (HGSO), which mimics the behavior governed by Henry's law to solve challenging optimization problems. Heidari *et al.* [31] proposed Harris hawks optimization (HHO), its main inspiration is the cooperative behavior and chasing style of Harris' hawks in nature. Li *et al.* [32] proposed slime mould algorithm (SMA) based on the oscillation mode of slime mould in nature. Faramarzi *et al.* [33] proposed marine predators algorithm (MPA), which is inspired by the widespread foraging strategy along with optimal encounter rate policy in biological interaction between predator and prey. Saremi *et al.* [34] proposed grasshopper optimization algorithm (GOA), which is a new nature-inspired algorithm designed by grasshopper swarm characteristics. There are only two parameters of kernel function in SVM that need to be optimized, GOA is chosen because of strong stability and simpler optimization process.

However, exploration and exploitation capabilities of GOA are not coordinated. Luo *et al.* [35] proposed an improved GOA which combines three strategies including Gaussian mutation, Levy-flight strategy and opposition-based learning. Sulaiman [36] updated the GOA with a better initialization strategy to balance the search capability. Zhao *et al.* [37] proposed improved GOA with nonlinear comfort zone parameter, the Levy flight mechanism and random jumping strategy. Raeesi *et al.* [38] improved GOA by adding opposition-based learning and merit function methods to boost its exploration and exploitation abilities.

To sum up, three issues are not resolved in the above methods based on deep learning. First, current algorithms have simple network structure and not consider the information of the surrounding vehicles, which leads to inadequate factors considered in the decision of the ego vehicle and inaccurate decision results. Second, softmax classifier is used in existing algorithms, which leads to their poor classification accuracy. Third, fixed kernel function parameters lead to poor

classification ability in SVM classifier after replaced softmax classifier. Moreover, in the process of using GOA to optimize parameters of SVM, exploration and exploitation capabilities of GOA are not coordinated.

To solve above issues, we propose a novel network architecture named improved LSTM network with SVM classifier optimized by GOA (GOA-ImLSTM). Firstly, in order to consider information of surrounding vehicles, a novel network architecture called improved LSTM (Im-LSTM), used to extract vital features for self-driving vehicles, with three parallel LSTM network units and a network unit serial connected according to vehicle location is constructed. The first three parallel LSTM network units extract the key features containing vital status information in the left lane, the right lane, and the front vehicle that facilitate the ego vehicle to understand surrounding environment and make correct driving decision-making. Then the extracted features are concatenated with the state of the ego vehicle, and feed it to the last LSTM network unit for obtaining the driving decision-making. Secondly, to enhance the decision-making classification accuracy for self-driving vehicles, SVM with stronger classification capability than softmax classification is employed. Softmax classifier simply maps input vectors to categories, which results in low classification accuracy under complex samples, while SVM classifier with radial basis function kernel maps the samples to a higher dimensional space, which more appropriate to classification under complex samples. In this way, the increasing accuracy of driving decision-making is ensured. Thirdly, since the performance of SVM is affected by the parameters in the kernel function, GOA is used to optimize the parameters of SVM. However, GOA not take into account that the found food source is locally optimal in the early stage of the GOA and the found food source gradually tended to be globally optimal in the later stage of the GOA. This leads to uncoordinated exploration and exploitation capabilities of GOA. Thus, dynamic weights in position movement formula are defined to balance the effects of the interaction of grasshoppers and tendency towards food source on grasshoppers position during the optimization process. A dynamic weight that is increased with the number of iterations for tendency towards food source is defined, and a dynamic weight that is decreased with the number of iterations for interaction of grasshoppers is defined in position movement formula. In this way, in the early stage of the GOA, the position of grasshoppers was more affected by the interaction of grasshoppers. And in the later stage of the GOA, the position of the grasshopper was more affected by the tendency to move towards the source of food, that is more suitable to pattern of grasshoppers motion. Then GOA is easier to find excellent parameters which increase classification accuracy for SVM.

The rest of our paper is organized as follows. In section II, we introduce the related work of our research. In section III, we propose a network architecture called GOA-ImLSTM. In section IV, the whole procedure of the GOA-ImLSTM is introduced. In section V, we evaluate the performance of proposed algorithm with three improvements on Next Generation SIMulation (NGSIM) [39]. In section VI, we give our conclusion.

II. RELATED WORK

In this section, we introduce a LSTM network, a classifier named SVM and a GOA.

A. LONG SHORT-TERM MEMORY NETWORK

Hochreiter *et al.* proposed LSTM network in 1997 [21]. LSTM network is a special type of RNN that can learn long-term dependence information. In the structure of LSTM network, LSTM cell replaced traditional RNN hidden cell to construct LSTM network. LSTM network has been widely used in speech recognition, language modeling, sentiment analysis, and text prediction. The structure of LSTM cell is shown in Fig. 1.

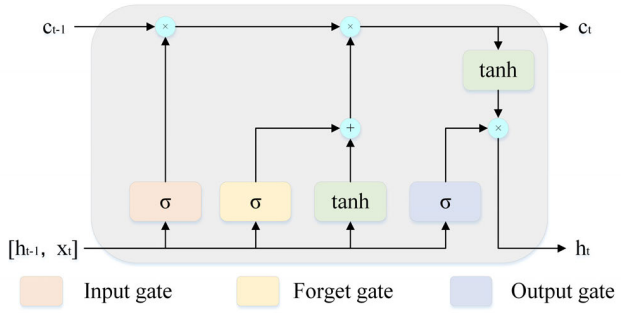

FIGURE 1. The diagram of an LSTM cell at time t.

As shown in Fig. 1, the LSTM cell includes three calculation gates, forget gate, input gate and output gate. Each gate is responsible for different function, the forget gate is responsible for determining to keep the unit status of previous moment to the unit status of current moment, the input gate is responsible for determining to keep the input to the unit status of current time, the output gate is responsible for determining the output of the unit status at the current moment. The computational process in LSTM cell is presented as follows.

$$f_t = \sigma\left(w_f\left[x_t, h_{t-1}\right] + b_f\right) \tag{1}$$
$$i_t = \sigma\left(w_i\left[x_t, h_{t-1}\right] + b_i\right) \tag{2}$$
$$\widetilde{c_t} = \tanh\left(w_c\left[x_t, h_{t-1}\right] + b_c\right) \tag{3}$$
$$c_t = f_t * c_{t-1} + i_t * \widetilde{c_t} \tag{4}$$
$$o_t = \sigma\left(w_o\left[x_t, h_{t-1}\right] + b_o\right) \tag{5}$$
$$h_t = o_t * \tanh\left(c_t\right) \tag{6}$$

where x_t is input data of current moment, h_{t-1} is output data of previous LSTM cell, w_f, w_i, w_c, w_o are weights of input data x_t and previous output data h_{t-1}. Where b_f, b_i, b_c, b_o are the bias vectors. $f_t, i_t, \widetilde{c_t}, o_t$ are the output of forget gate, input gate, generated information, and output gate. c_{t-1} and c_t are the unit status of the previous moment and current moment. h_t is the output data of LSTM cell.

σ and tanh are represented the sigmoid and tanh function as follows.

$$\sigma(x) = \frac{1}{1+e^{-x}} \tag{7}$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \tag{8}$$

B. SUPPORT VECTOR MACHINE

SVM [27] is one of the most robust and accurate methods in all well-known classification algorithms. SVM is a supervised learning method that can be widely used in statistical classification and regression analysis. The basic model of SVM is defined as the linear classifier with the largest interval in the feature space. The purpose of SVM is to find an optimal segmentation surface to maximize the classification interval.

Only considered a two-classification problem, the input point is represented by $x_i \in R^d (i = 1, 2, \cdots, N)$, which is N-dimensional vector, and the category is represented by $y_i \in \{-1, 1\} (i = 1, 2, \cdots, N)$, which represented two different classes. Assume that two classes are linearly separable. The purpose of classifier is to find a hyperplane in N-dimensional data space, which is defined as follows.

$$\omega^T x + b = 0 \tag{9}$$

To find the optimal hyperplane that maximizes the distance between the closest training sample and the hyperplane. By simply rescaling the hyperplane parameters, the distance can be calculated as equal to $1/\|\omega\|$, the problem of maximizes distance transforms the minimization problem of $\|\omega\|$. Furthermore, the minimization problem of $\|\omega\|$ is equivalent to as follows.

$$\begin{cases} minimize: \ \dfrac{1}{2}\omega^2 \\ subject\ to: \ y_i\left(\omega^T x_i + b\right) \geq 1, \quad i = 1, 2, \ldots, N \end{cases} \tag{10}$$

This classic linearly constrained optimization problem can be transformed into the following optimization problem of dual variables by Lagrangian formulation as follows.

$$\begin{cases} maximize: -\dfrac{1}{2}\displaystyle\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j\right) \\ subject\ to: \displaystyle\sum_{i=1}^{N}\alpha_i y_i = 0 \ \ and\ \alpha_i \geq 0, \ i = 1, 2, \ldots, N \end{cases} \tag{11}$$

where α_i is Lagrange multiplier which is estimated using quadratic programming methods in [40].

Assume the solution to the optimization problem of dual variables has been obtained as follows.

$$\left\{\alpha^* = \left(\alpha_1^*, \alpha_2^*, \cdots, \alpha_N^*\right)^T \right. \tag{12}$$

The ω and b of optimal hyperplane are defined as follows.

$$\boldsymbol{\omega}^* = \sum_{i=1}^{N}\alpha_i^* y_i \boldsymbol{x}_i \tag{13}$$

$$b^* = y_j - \sum_{i=1}^{N}\alpha_i^* y_i \left(\boldsymbol{x_i} \cdot \boldsymbol{x}_j\right) \tag{14}$$

The optimal hyperplane is redefined as follows.

$$\sum_{i=1}^{N}\alpha_i^* y_i \left(\boldsymbol{x} \cdot \boldsymbol{x}_i\right) + b^* = 0 \tag{15}$$

The classification decision function is defined as follows.

$$f(\boldsymbol{x}) = sign\left(\sum_{i=1}^{N}\alpha_i^* y_i \left(\boldsymbol{x} \cdot \boldsymbol{x}_i\right) + b^*\right) \tag{16}$$

where *sign* is a symbolic function, which is defined as follows.

$$sign(x) = \begin{cases} 1, & x \geq 0 \\ -1, & x < 0 \end{cases} \tag{17}$$

C. GRASSHOPPER OPTIMIZATION ALGORITHM

Saremi *et al.* proposed GOA in 2017 [34]. GOA is a new nature-inspired algorithm designed by two grasshopper swarm characteristics. First characteristic is slow movement and small steps of the grasshoppers in the larval phase, as well as long-range and abrupt movement in adulthood. Second characteristic is food source seeking of grasshoppers. Two characteristics of grasshoppers performed two tendencies of search process in nature-inspired algorithms, which are exploration and exploitation, as well as target seeking.

To model the swarming behaviors of grasshoppers, the mathematical model is presented as follows.

$$X_i = S_i + G_i + A_i \tag{18}$$

where X_i defines the position of the *i-th* grasshopper, S_i is the social interaction, G_i is the gravity force on the *i-th* grasshopper, and A_i shows the wind advection.

S_i presented the social interaction is defined as follows.

$$S_i = \sum_{\substack{j=1 \\ j\neq i}}^{N} s\left(d_{ij}\right)\widehat{d_{ij}} \tag{19}$$

where d_{ij} is the distance between the *i-th* and the *j-th* grasshoppers, s is a function to define the strength of social forces, is calculated as follows.

$$s(r) = fe^{\frac{-r}{l}} - e^{-r} \tag{20}$$

where f indicates the intensity of attraction and l is the attractive length scale, we have chosen l equals 1.5 and f equals 0.5 in [34]. And $\widehat{d_{ij}}$ is a unit vector from the *i-th* grasshopper to the *j-th* grasshopper as follows.

$$\widehat{d_{ij}} = \frac{x_j - x_i}{d_{ij}} \tag{21}$$

G_i is calculated as follows.

$$G_i = -g\widehat{e_g} \tag{22}$$

where g is the gravitational constant and $\widehat{e_g}$ shows a unity vector towards the center of the earth.

A_i is calculated as follows.

$$A_i = u\widehat{e_w} \tag{23}$$

where u is a constant drift and $\widehat{e_w}$ is a unity vector in the direction of wind.

However, this mathematical model cannot be used directly to solve optimization problems, a modified equation is proposed to solve optimization problems in [34].

$$X_i^d = c\left\{\sum_{\substack{j=1\\ j\neq i}}^{N} c\frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right)\right\} + \widehat{T_d} \tag{24}$$

where ub_d is the upper bound in the Dth dimension, lb_d is the lower bound in the Dth dimension, $\widehat{T_d}$ is the value of the Dth dimension in the target, and c is a decreasing coefficient to shrink the comfort zone, repulsion zone, and attraction zone.

To balance exploration and exploitation in search process, the parameter c is required to be decreased proportionally to the number of iterations. Decreasing coefficient c is defined as follows.

$$c = cmax - l\frac{cmax - cmin}{L} \tag{25}$$

where $cmax$ is the maximum value, $cmin$ is the minimum value, l indicates the current iteration, and L is the maximum number of iterations. We have chosen 1 and 0.00001 for $cmax$ and $cmin$ in [34].

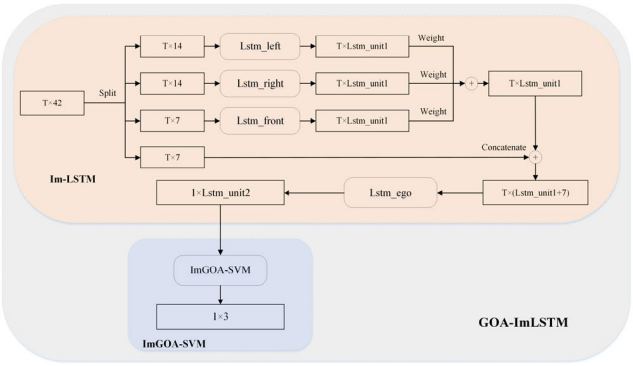

FIGURE 2. Architectural diagram of GOA-ImLSTM. Given the state of vehicles including the past T time steps, GOA-ImLSTM first splits it by vehicle lanes and feeds them into Im-LSTM to extract features. Then ImGOA-SVM is used for driving decision-making classification.

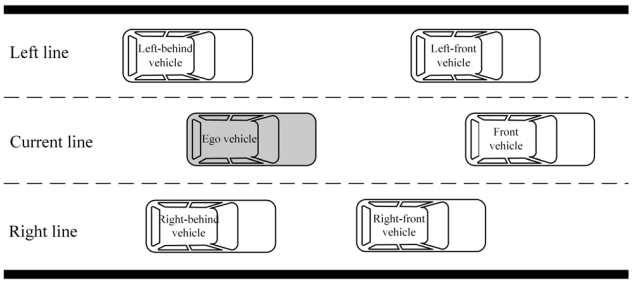

FIGURE 3. The input of Im-LSTM contains the status of six vehicles, including left-behind vehicle, left-front vehicle, front vehicle, right-behind vehicle, right-front vehicle, and ego vehicle.

III. GOA-IMLSTM

GOA-ImLSTM is proposed based on LSTM, SVM, and GOA as shown in Fig. 2. Three improvements are presented in this section. In part A, a novel network architecture with three parallel LSTM units and a LSTM unit serial connected according to vehicle location is designed, termed Im-LSTM. In part B, the commonly used softmax is replaced by SVM to accomplish the classification task. In part C, improved GOA is applied to optimize the parameters of the SVM, termed ImGOA-SVM.

A. IM-LSTM

Im-LSTM consists of four LSTM network units, which are Lstm_left, Lstm_right, Lstm_front and Lstm_ego. The input of Im-LSTM contains the status of six vehicles as shown in Fig. 3, including angle steer, speed, acceleration, lateral distance from ego vehicle, longitudinal distance from ego vehicle, vehicle length, vehicle width. Because driving decisions are time-dependent, the status of each vehicle in the past T time steps is selected as input. To consider the information of the surrounding vehicles, network architecture, used to extract important features for self-driving vehicle, with three parallel LSTM units and a LSTM unit serial connected according to vehicle location is designed.

As shown in Fig. 2, Im-LSTM first splits the information of the surrounding vehicles by vehicle location and feeds them into Lstm_left, Lstm_right, Lstm_front, and Lstm_ego. The input of Lstm_left contains the status of left-behind vehicle and left-front vehicle. The output of Lstm_left are pivotal features of left lane. The input of Lstm_right contains the status of right-behind vehicle and right-front vehicle. The output of Lstm_right are pivotal features of right lane. The input of Lstm_front contains the status of front vehicle. The output of Lstm_front are pivotal features of front vehicle.

The number of neural units in Lstm_left, Lstm_right, and Lstm_front is Lstm_unit1. The pivotal features of the left lane, the right lane, and the front vehicle are weighted separately and added, then concatenated with the status of ego vehicle, the concatenated vector is input to Lstm_ego. The number of neural units in Lstm_ego is Lstm_unit2. The output of Lstm_ego is the key feature for driving decision-making eventually.

The weights of the outputs of the Lstm_left, Lstm_right, and Lstm_front are α, β, and γ, separately. They represent the importance of the left lane, right lane and front vehicle for driving decision-making. When ego vehicle makes lane keeping decision, it is more affected by the front vehicle, and when ego vehicle makes lane changing decision, it is more affected by the left or right lane. Since the probability of collision is higher when making lane changing decision, the weight of the left lane and right lane is greater than the

weight of front vehicle. And the sum of three weights is 1, the α, β, and γ are set as 0.4, 0.4 and 0.2.

B. SVM CLASSIFIER

Softmax classifier is used in most methods based on LSTM for decision-making. However, the classification ability of softmax classifier is poor, which leads to inaccurate classification results. To solve this issue, SVM classifier is employed for classifying the driving decision-making in GOA-ImLSTM.

The input of SVM classifier is the key feature for driving decisions from Im-LSTM, and the output of SVM classifier is the final decision-making of ego vehicle. SVM classifier in GOA-ImLSTM as shown in Fig. 4.

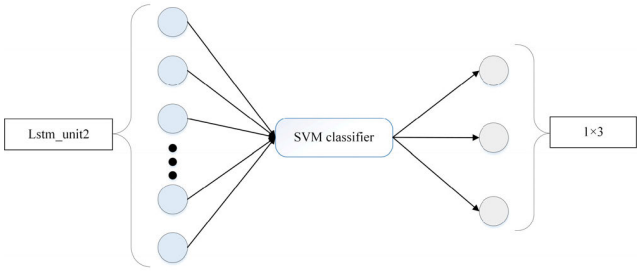

FIGURE 4. SVM classifier in GOA-ImLSTM. Input key features for driving decisions from Im-LSTM and classify the final decision-making by using SVM classifier.

The kernel function of SVM classifier chose radial basis function (RBF) kernel. RBF kernel function is chosen because the fitting ability of RBF kernel function is better than logistic regression classifier by mapping samples to a higher dimensional space. The classification ability of SVM classifier with RBF kernel function depends heavily on the choice of c and γ, the parameters of RBF kernel function. Therefore, the main purpose of the next part is to optimize c and γ of RBF kernel function.

C. IMGOA-SVM

To achieve excellent classification results, the parameters c and γ of RBF kernel function obtained through optimization algorithm. The parameter c is the penalty coefficient, which is the tolerance of the error. The larger c indicates a lower tolerance for errors, which easily leads to overfitting. The smaller c is more easily to cause underfitting. The parameters γ implicitly determine the distribution of the data after it is mapped to the new feature space. Larger γ results in fewer support vectors, and smaller γ results in more support vectors. The number of support vectors affects the speed of the training and prediction process in classification. Consequently, the values of parameters c and γ are very important to the accuracy of classification results.

In order to promote the classification ability of SVM, the GOA method is introduced to optimize the parameters c and γ of SVM, termed GOA-SVM. Optimizing parameters of SVM by using GOA includes three aspects. First, each grasshopper means a set of solutions for c and γ. Thus, the movement of grasshopper represents the changes in parameters c and γ, and the searching of the best grasshopper is to find the best c and γ. The performance of each grasshopper is evaluated by the fitness function. The parameters c and γ are updated based on the fitness values.

Furthermore, the movement equation of grasshoppers in grasshopper optimization algorithm is defined in (24). The first term of movement equation considers the position of other grasshoppers and implements the interaction of grasshoppers in nature. The second term of movement equation simulates their tendency to move towards the source of food. Nevertheless, in the early stage of the grasshopper optimization algorithm, the found food source was locally optimal, thus, the position of grasshoppers was more affected by the interaction of grasshoppers. In the later stage of the grasshopper optimization algorithm, the found food source gradually tended to be globally optimal, so the position of the grasshopper was more affected by the tendency to move towards the source of food. To balance interaction of grasshoppers and tendency towards food source, dynamic weights in position movement formula are defined.

At the beginning of the optimization process, the interaction of the grasshoppers has a great influence on the position of the grasshoppers. As the number of iterations increases, the positions of the grasshoppers gradually move to the source of food, which is less affected by the interaction of the grasshoppers. Thus, a gradually decreasing dynamic weight is defined for the interaction of the grasshoppers. And at the beginning of the optimization process, the tendency to move towards the source of food has little effect on the grasshoppers position, since the current source of food is locally optimal. As the number of iterations increases, the influence of the tendency to move towards the source of food becomes greater, since the current source of food gradually tends to global optimization. Thus, a gradually increasing dynamic weight is defined for the tendency to move towards the source of food.

A modified equation is proposed as follows.

$$X_i^d = w_1 \left\{ \sum_{\substack{j=1 \\ j \neq i}}^{N} c \frac{ub_d - lb_d}{2} s\left(\left|x_j^d - x_i^d\right|\right) \right\} + w_2 \widehat{T}_d \quad (26)$$

where w_1 is the weight of interaction of grasshoppers, w_2 is the weight of tendency towards food source.

The linearly decreasing and increasing weights leads unbalanced exploration and exploitation capabilities of GOA, to match these two capabilities and enhance the search ability of the algorithm, the exponential and logarithmic functions with good performance are introduced to nonlinear adjust parameters.

w_1 and w_2 are defined as follows.

$$w_1 = 2 - e^{\frac{l * lg2}{L}} \quad (27)$$

$$w_2 = \log_l [\frac{l\,(l-1)}{L} + 1] \quad (28)$$

where l indicates the current iteration, and L is the maximum number of iterations.

The graph of weights w_1 and w_2 is presented in Fig. 5 and Fig. 6.

As shown in Fig. 5, w_1 decreases as the number of iterations increases, when the number of iterations is extremely small, w_1 is close to 1, at this time the position of grasshoppers was more affected by the interaction of grasshoppers.

In contrast, as shown in Fig. 6, w_2 increases as the number of iterations increases, when the number of iterations is extremely high, w_2 is close to 1, at this time the position of grasshoppers was more affected by the tendency towards the source of food.

Improved GOA with new position formula is employed to optimize parameters of SVM, termed ImGOA-SVM.

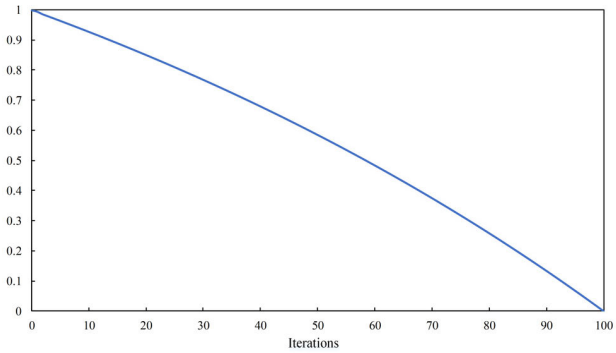

FIGURE 5. The graph of w_1 when L equals 100.

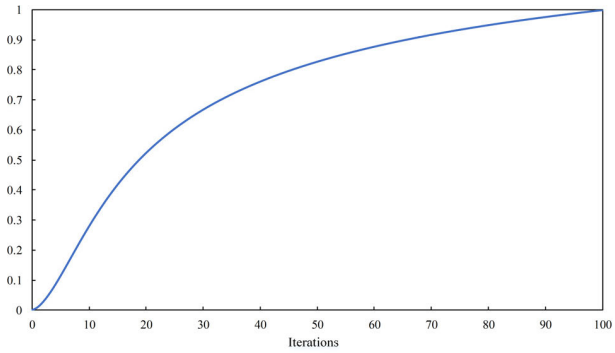

FIGURE 6. The graph of w_2 when L equals 100.

IV. THE OVERALL PROCESS OF GOA-IMLSTM

The flowchart of GOA-ImLSTM is illustrated in Fig. 7. First, Im-LSTM trains the input samples and obtain the key feature for driving decision-making. Second, the positions of N grasshoppers, which presented parameters of SVM, are randomly initialized at the beginning of ImGOA-SVM. Then, the fitness values of grasshoppers are calculated by obtaining the accuracy of SVM classification results with RBF kernel of parameters c and γ. Next, update the position of all grasshoppers in the population according to the defined movement equation in (26). Furthermore, if the maximum iteration of GOA is reached, the optimization process is finished. As a result, we can obtain the optimal parameters c and γ. The searching range for parameters c and γ is [0.01, 10] and [0.0001, 5] respectively.

Algorithm 1 GOA-ImLSTM

Input: S: status of surrounding vehicles and ego vehicle
Output: D: decision-making of ego vehicle
split S to S_l, S_r, S_f, S_e by vehicle location
obtain pivotal features of left lane F_l by feeding S_l to Lstm_left
obtain pivotal features of right lane F_r by feeding S_r to Lstm_right
obtain pivotal features of front lane F_f by feeding S_f to Lstm_front
F ← weight F_l, F_r, $F_f$separately and add
F ← F concatenated with S_e
obtain pivotal features F_e by feeding F to Lstm_ego
iter ← 1
while iter < Max_iterations **do**
  Initialize the positions of the grasshoppers
  Calculate the fitness of grasshoppers by SVM
  Update the food source
  iter ← iter + 1
  Update the positions of grasshoppers by (26)
D ← classification by SVM with parameters by optimal position

V. EXPERIMENT AND ANALYSIS

In order to evaluate the performance of the GOA-ImLSTM, we have applied the algorithm on NGSIM. The data processing platform is windows10 on laptop PC 2.3GHz processor and 4GB memory.

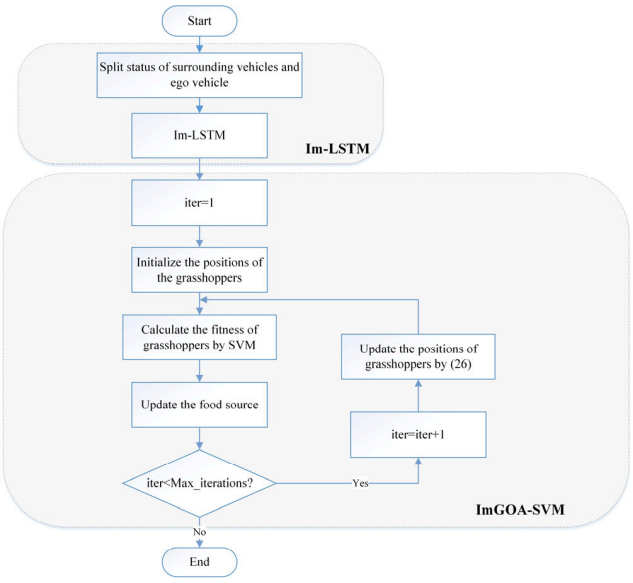

FIGURE 7. The flowchart of GOA-ImLSTM.

A. DATASET AND PROCESSING

Researchers at the NGSIM project collected detailed vehicle trajectory data using data from US 101 Southbound Road from Los Angeles, California [39]. Vehicle trajectory data provides the precise position of each vehicle in the study area every tenth of a second to obtain detailed lane positions and positions relative to other vehicles. We select representative data for training and testing from dataset.

First, find the turning angle of each vehicle at the current moment by formula as follows.

$$\theta = \arctan \frac{x_{t+t_0} - x_t}{y_{t+t_0} - y_t} \tag{29}$$

where x_t is the horizontal position of vehicle at time t, x_{t+t_0} is the horizontal position of vehicle at time $t + t_0$, y_t is the vertical position of vehicle at time t, y_{t+t_0} is the vertical position of vehicle at time $t + t_0$.

A lane-changing trajectory of vehicle selected from the dataset is shown in Fig. 8. The time interval for acquiring track points is 0.1 second, the figure shows the track within 10 seconds. The blue trajectory indicates that the vehicle makes a lane keeping decision, and the orange trajectory indicates that the vehicle makes a lane changing decision. The turning angle of vehicle at each point in the trajectory is calculated by (29) as shown in Fig. 9 when t_0 equals 1,2,3,4, and 5.

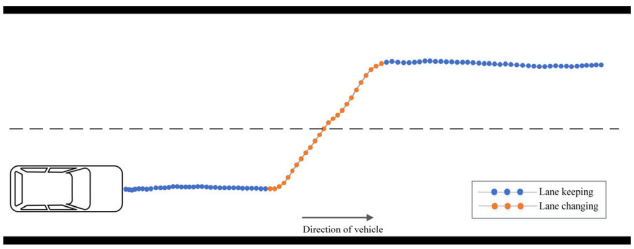

FIGURE 8. A lane-changing trajectory of vehicle.

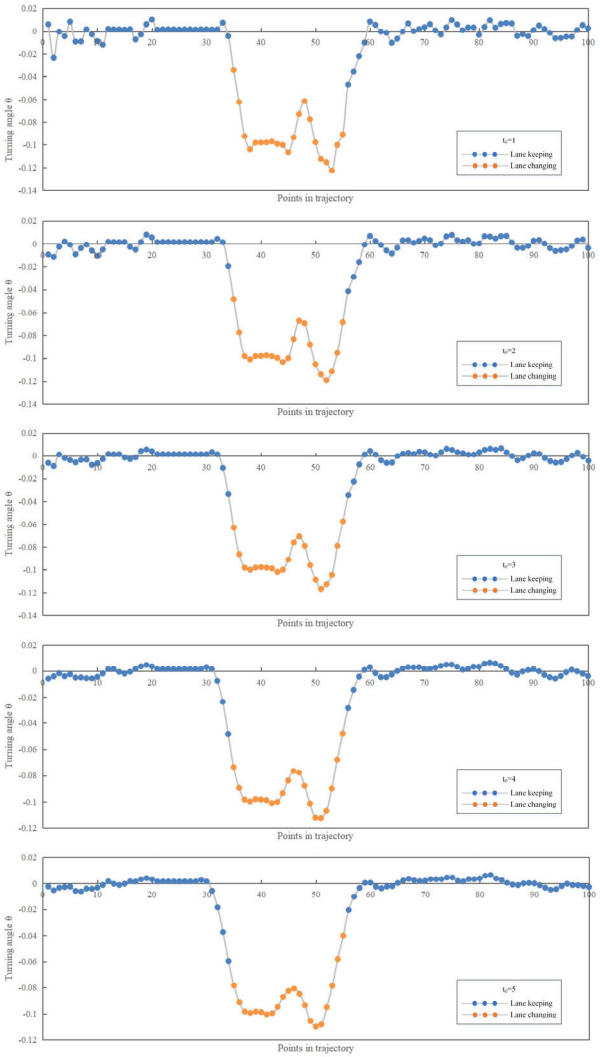

FIGURE 9. Turning angle of each point in trajectory when t_0 equals 1,2,3,4, and 5.

If t_0 is selected too small, the absolute value of the turning angle calculated when the vehicle starts to make the lane changing decision is very small, and the absolute value of the turning angle calculated when the vehicle has finished the lane changing decision and starts the lane keeping decision is very large. On the contrary, if t_0 is selected too large, the absolute value of the turning angle calculated when the vehicle starts to make the lane changing decision is very large, and the absolute value of the turning angle calculated when the vehicle has finished the lane changing decision and starts the lane keeping decision is very small. We need the absolute value of calculated turning angle as large as possible when making the lane changing decision, and at the same time the absolute value of calculated turning angle as small as possible when making the lane changing decision, in this way, the decisions made by vehicles be more accurate. To sum up, we set t_0 equals 3.

Second, the dataset is grouped by time to complete the status information of each vehicle, including the left-front vehicle, the left-behind vehicle, the right-front vehicle, the right-behind vehicle, the front vehicle, and the ego vehicle. The rotation angle, speed, acceleration, lateral distance, longitudinal distance, vehicle length, and vehicle width of each vehicle constitute the training dataset. Third, get the corresponding vehicle decision for each vehicle including lane-keeping, turn left lane, and turn right lane by ego vehicle status.

B. PARAMETERS SETTING

The parameters setting in GOA-ImLSTM is introduced in table. 1. All the codes are implemented in Tensorflow 1.12.0 and Keras 1.3.0., and the network was trained by the Adam optimizer.

C. EXPERIMENT RESULTS AND ANALYSIS

In this section, we evaluate the performance of the proposed network architecture on NGSIM and compare the results with other methods. The network has 48,000 training samples and

TABLE 1. Parameters setting.

Parameter	Description	Value
T	Time step in Lstm_left, Lstm_right, Lstm_front and Lstm_ego	20
Lstm_unit1	Number of neural units in Lstm_left, Lstm_right and Lstm_front	35
Lstm_unit2	Number of neural units in Lstm_ego	100
learning_rate	Learning rate	0.0001
NUM	Number of grasshoppers	20
cMax	Maximum number of Coefficient	1
cMin	Minimum number of Coefficient	0.00004
Cmax	Maximum number of C in SVM	10
Cmin	Minimum number of C in SVM	0.01
Gmax	Maximum number of gamma in SVM	5
Gmin	Minimum number of gamma in SVM	0.0001

3,600 testing samples, and each decision has the same number of samples.

As shown in Fig. 10, the accuracy of driving decision-making increases with the number of iterations. The maximum number of iterations of GOA-ImLSTM is 200. When the number of iterations is between 1 and 50, the decision accuracy increases with the number of iterations. If speed is required but accuracy is not required, the number of iterations is selected to be 20. If speed and accuracy are both required, the number of iterations is selected to be 50. When the number of iterations is between 50 and 100, the rate of decision accuracy increases decreases. If speed is required but accuracy is more required, the number of iterations is selected to be 100. When the number of iterations is between 100 and 200, the training time is very long but the increase in accuracy slows down significantly, so we not consider the case of 200 iterations.

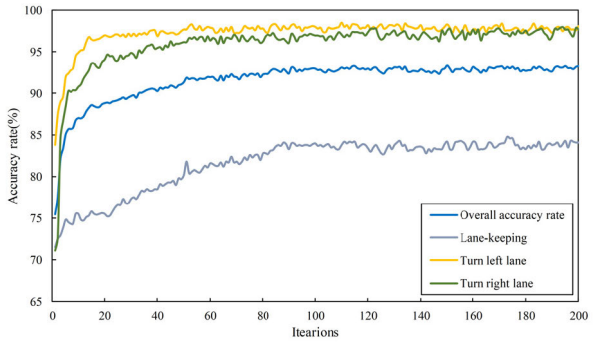

FIGURE 10. Accuracy rate of GOA-ImLSTM corresponding to different iterations.

We compared the driving decision-making task against six methods: RNN, LSTM, GRU, Im-LSTM, Im-LSTM with SVM and Im-LSTM with GOA-SVM, which network architectures as shown in table. 2.

Table. 3, 4, and 5 shows the quantitative results of GOA-ImLSTM on the NGSIM dataset. The criterion is the accuracy rate. The results show that three innovations gradually improve the accuracy of decision results on the NGSIM dataset.

TABLE 2. The network architectures of six algorithms.

Algorithms	Network architecture	References
RNN	100 RNN units with softmax	[19]
GRU	100 GRU units with softmax	[22]
LSTM	100 LSTM units with softmax	[21]
Im-LSTM	Im-LSTM with softmax	—
Im-LSTM with SVM	Im-LSTM and SVM as classifier replaced softmax	—
Im-LSTM with GOA-SVM	Im-LSTM and GOA-SVM as classifier replaced softmax	—

TABLE 3. The accuracy rate (%) in 20 iterations.

Algorithms	Lane-keeping	Turn left lane	Turn right lane	Overall
RNN	44.2%	72.3%	48.9%	55.1%
GRU	30.9%	72.3%	50.6%	51.3%
LSTM	33.0%	58.9%	61.3%	51.1%
Im-LSTM	64.5%	89.5%	90.3%	81.4%
Im-LSTM with SVM	72.0%	95.0%	89.0%	85.3%
Im-LSTM with GOA-SVM	72.0%	95.5%	90.9%	86.1%
GOA-ImLSTM	**75.5%**	**97.0%**	**94.1%**	**88.7%**

As shown in table. 3, 4, and 5, the performances of RNN, GRU, and LSTM are compared in first, second and third lines. LSTM has the highest overall accuracy rate in the case of 20 iterations, 50 iterations and 100 iterations. That proves LSTM network structure is promising in driving decision-making. The advantage of LSTM compared with RNN is time dependence, and compared with GRU, LSTM has more accurate decision-making in this case of large data volume because of its more parameters. And as the number of iterations increases, the accuracy rate of decision-making increases.

The comparison of LSTM and Im-LSTM is shown in third and fourth lines in table. 3, 4, and 5. For overall accuracy rate, Im-LSTM is 30.3% better than LSTM in the case of 20 iterations, 23.3% better in the case of 50 iterations and 23.1% better in the case of 100 iterations. That indicates the novel network architecture of Im-LSTM indeed promotes the accuracy rate of self-driving decision-making because Im-LSTM extracts more accurately characteristics of the situation at that time to make more accurate decisions.

Fourth and fifth lines in table. 3, 4, and 5, the performances of Im-LSTM and Im-LSTM with SVM are compared. In the case of 20 iterations, Im-LSTM has higher accuracy rate in turn right lane decision-making, but Im-LSTM with SVM has higher lane-keeping decision-making, turn left lane decision-making and overall accuracy rate. Although the accuracy rate of Im-LSTM and Im-LSTM with SVM network in some specific decision-makings are equal, Im-LSTM with SVM

TABLE 4. The accuracy rate (%) in 50 iterations.

Algorithms	Lane-keeping	Turn left lane	Turn right lane	Overall
RNN	50.8%	79.2%	50.0%	60.0%
GRU	35.8%	75.9%	53.3%	55.0%
LSTM	43.5%	81.9%	61.3%	62.2%
Im-LSTM	67.8%	96.5%	92.3%	85.5%
Im-LSTM with SVM	72.1%	96.8%	92.3%	87.2%
Im-LSTM with GOA-SVM	77.8%	96.8%	93.4%	89.3%
GOA-ImLSTM	**80.2%**	**97.7%**	**96.2%**	**91.4%**

TABLE 5. The accuracy rate (%) in 100 iterations.

Algorithms	Lane-keeping	Turn left lane	Turn right lane	Overall
RNN	53.8%	80.8%	58.0%	64.2%
GRU	36.1%	80.8%	61.3%	59.4%
LSTM	49.9%	83.6%	62.8%	65.4%
Im-LSTM	75.3%	97.2%	93.0%	88.5%
Im-LSTM with SVM	75.3%	97.7%	93.0%	88.6%
Im-LSTM with GOA-SVM	79.3%	97.9%	95.1%	90.7%
GOA-ImLSTM	**84.0%**	**98.0%**	**97.0%**	**93.0%**

has higher overall accuracy rate in the cases of 50 and 100 iterations. According to this comparison, the SVM classifier indeed increases decision precision because its classification ability is stronger than softmax classifier.

The comparison of Im-LSTM with SVM and Im-LSTM with GOA-SVM is shown in fifth and sixth lines in table. 3, 4, and 5. Although these two methods have equal accuracy rate in lane-keeping in the case of 20 iterations, Im-LSTM with GOA-SVM has higher accuracy rate in turn left lane, turn right lane decision-making and overall. For overall accuracy rate, Im-LSTM with GOA-SVM is 0.8% better than Im-LSTM with SVM in the case of 20 iterations, 2.1% better in the case of 50 iterations and 2.1% better in the case of 100 iterations. The comparison shows that GOA finds more suitable parameters for the classification of driving decision-making to improve the classification ability of SVM, so Im-LSTM with GOA-SVM obtained a higher accuracy rate.

Last two lines in table. 3, 4, and 5, the performances of Im-LSTM with GOA-SVM and GOA-ImLSTM are compared. The overall accuracy rate of GOA-ImLSTM has 2.6% improvement over LSTM with GOA-SVM in the case of 20 iterations, 2.1% improvement in the case of 50 iterations and 2.3% improvement in the case of 100 iterations. That demonstrates the improved GOA gets more excellent parameters of SVM due to the balance of exploration and exploitation ability, thereby further improving the performance of SVM classifier.

TABLE 6. The accuracy rate (%) of other algorithms.

Algorithms	Overall accuracy rate
GOA-ImLSTM	**93.0%**
Lane change intent prediction method	91.4%
Game-theoretical decision-making model	86.0%
Bidirectional Recurrent Neural Network	80.6%
Bayesian Network	71.6%
J48-Decision Tree	75.4%
Free lane-changing model	86.9%
A novel approach using a combination of spectral graph analysis and deep learning	91.2%

Table. 6 shows the comparison of GOA-ImLSTM and the previous state-of-the-art methods, including lane change intent prediction method [41], game-theoretical decision-making model [42], bidirectional recurrent neural network [43], bayesian network [44], J48-Decision tree [44], free lane-changing model [44], and a novel approach using a combination of spectral graph analysis and deep learning [45]. It shows that the overall accuracy boost to 93.0%, which has 1.6% improvement over lane change intent prediction method, 7.0% improvement over game-theoretical decision-making model, 12.4% improvement over bidirectional recurrent neural network, 21.4% improvement over bayesian network, 17.6% improvement over J48-Decision tree, 6.1% improvement over free lane-changing model, and 1.8% improvement over a novel approach using a combination of spectral graph analysis and deep learning. The results clearly show that GOA-ImLSTM achieves the best performance in terms of overall accuracy rate.

D. DISCUSSION

According to above observations, it can be concluded that the performance of GOA-ImLSTM is better than other methods in autonomous driving decision-making. The reason is that each innovation contributes to the improvement of decision-making accuracy. The runtime of GOA-ImLSTM is 520μs in driving decision-making.

Although GOA-ImLSTM has obtained good experimental results, it still has certain limitations. First, the computational complexity of the algorithm is relatively high, which is $O(n^2)$. Second, the final decision classification result still largely depends on the parameters of the SVM classifier. Despite we use an improved GOA to optimize the parameters in SVM, since the computational complexity cannot be too high and the number of optimization iterations is limited, there is no guarantee that the parameters after a fixed number of optimizations will be the best results.

VI. CONCLUSION

Because the information of the surrounding vehicles is not considered and poor classification capabilities of LSTM, the classification accuracy rate of decision-making by using LSTM is unsatisfactory. To solve this issue, we presented an efficient network architecture for driving decision-making

which named GOA-ImLSTM. First, we designed a novel network architecture with three parallel LSTM units and a LSTM unit serial connected according to vehicle location to consider the information of the surrounding vehicles for making decisions. Second, we introduced SVM classifier instead of universally used softmax to accomplish the classification task because of its stronger classification capability. Third, we employed GOA to optimize the parameters in SVM. Furthermore, we defined dynamic weights in position movement formula. In this way, the optimization effect was further enhanced.

The results verify that the GOA-ImLSTM achieves better performance on the NGSIM dataset compared with other methods. In the future, we will apply the proposed network architecture to other tasks and continue to improve.

REFERENCES

[1] J. Kolodko and L. Vlacic, "Cooperative autonomous driving at the intelligent control systems laboratory," *IEEE Intell. Syst.*, vol. 18, no. 4, pp. 8–11, Jul./Aug. 2003.

[2] M. Munz, M. Mahlisch, and K. Dietmayer, "Generic centralized multi sensor data fusion based on probabilistic sensor and environment models for driver assistance systems," *IEEE Intell. Transp. Syst. Mag.*, vol. 2, no. 1, pp. 6–17, Aug. 2010.

[3] Y. Rasekhipour, A. Khajepour, S.-K. Chen, and B. Litkouhi, "A potential field-based model predictive path-planning controller for autonomous road vehicles," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1255–1267, May 2017.

[4] S. M. Veres, L. Molnar, N. K. Lincoln, and C. P. Morice, "Autonomous vehicle control systems—A review of decision making," *Proc. Inst. Mech. Eng. I, J. Syst. Control Eng.*, vol. 225, no. 2, pp. 155–195, Mar. 2011.

[5] M. H. Lee, H. G. Park, S. H. Lee, K. S. Yoon, and K. S. Lee, "An adaptive cruise control system for autonomous vehicles," *Int. J. Precis. Eng. Manuf.*, vol. 14, no. 3, pp. 373–380, Mar. 2013.

[6] J. Nie, J. Zhang, W. Ding, X. Wan, X. Chen, and B. Ran, "Decentralized cooperative lane-changing decision-making for connected autonomous vehicles," *IEEE Access*, vol. 4, pp. 9413–9420, 2016.

[7] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[8] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[9] A. Furda and L. Vlacic, "Enabling safe autonomous driving in real-world city traffic using multiple criteria decision making," *IEEE Intell. Transp. Syst. Mag.*, vol. 3, no. 1, pp. 4–17, Apr. 2011.

[10] L. Chong, M. M. Abbas, A. Medina Flintsch, and B. Higgs, "A rule-based neural network approach to model driver naturalistic behavior in traffic," *Transp. Res. C, Emerg. Technol.*, vol. 32, pp. 207–223, Jul. 2013.

[11] B. Barman, R. Kanjilal, and A. Mukhopadhyay, "Neuro-fuzzy controller design to navigate unmanned vehicle with construction of traffic rules to avoid obstacles," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 24, no. 3, pp. 433–449, Jun. 2016.

[12] S. Li, J. Zhang, S. Wang, P. Li, and Y. Liao, "Ethical and legal dilemma of autonomous vehicles: Study on driving decision-making model under the emergency situations of red light-running behaviors," *Electronics*, vol. 7, no. 10, p. 264, Oct. 2018.

[13] D. C. K. Ngai and N. H. C. Yung, "A multiple-goal reinforcement learning method for complex vehicle overtaking maneuvers," *IEEE Trans. Intell. Transp. Syst.*, vol. 12, no. 2, pp. 509–522, Jun. 2011.

[14] T. Gindele, S. Brechtel, and R. Dillmann, "Learning driver behavior models from traffic observations for decision making and planning," *IEEE Intell. Transp. Syst. Mag.*, vol. 7, no. 1, pp. 69–79, Jan. 2015.

[15] W. Song, G. Xiong, and H. Chen, "Intention-aware autonomous driving decision-making in an uncontrolled intersection," *Math. Problems Eng.*, vol. 2016, pp. 1–15, Mar. 2016.

[16] C. You, J. Lu, D. Filev, and P. Tsiotras, "Advanced planning for autonomous vehicles using reinforcement learning and deep inverse reinforcement learning," *Robot. Auto. Syst.*, vol. 114, pp. 1–18, Apr. 2019.

[17] C. Chen, X. Liu, T. Qiu, and A. K. Sangaiah, "A short-term traffic prediction model in the vehicular cyber–physical systems," *Future Gener. Comput. Syst.*, vol. 105, pp. 894–903, Apr. 2020.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[19] J. L. Elman, "Finding structure in time," *Cognit. Sci.*, vol. 14, no. 2, pp. 179–211, Mar. 1990.

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

[21] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[22] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder–decoder approaches," in *Proc. 8th Workshop Syntax, Semantics Struct. Stat. Transl.*, 2014, p. 103.

[23] Z. Gao, Y. Li, and S. Wan, "Exploring deep learning for view-based 3D model retrieval," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 16, no. 1, pp. 1–21, Feb. 2020.

[24] 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.

[25] S. Chen, S. Zhang, J. Shang, B. Chen, and N. Zheng, "Brain-inspired cognitive model with attention for self-driving cars," *IEEE Trans. Cognit. Develop. Syst.*, vol. 11, no. 1, pp. 13–25, Mar. 2019.

[26] 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, Sep. 2019.

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

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

[29] S. Boettcher and A. G. Percus, "Optimization with extremal dynamics," *Phys. Rev. Lett.*, vol. 86, no. 23, pp. 5211–5214, Jun. 2001.

[30] F. A. Hashim, E. H. Houssein, M. S. Mabrouk, W. Al-Atabany, and S. Mirjalili, "Henry gas solubility optimization: A novel physics-based algorithm," *Future Gener. Comput. Syst.*, vol. 101, pp. 646–667, Dec. 2019.

[31] A. A. Heidari, S. Mirjalili, H. Faris, I. Aljarah, M. Mafarja, and H. Chen, "Harris hawks optimization: Algorithm and applications," *Future Gener. Comput. Syst.*, vol. 97, pp. 849–872, Aug. 2019.

[32] S. Li, H. Chen, M. Wang, A. A. Heidari, and S. Mirjalili, "Slime mould algorithm: A new method for stochastic optimization," *Future Gener. Comput. Syst.*, vol. 111, pp. 300–323, Oct. 2020.

[33] A. Faramarzi, M. Heidarinejad, S. Mirjalili, and A. H. Gandomi, "Marine predators algorithm: A nature-inspired metaheuristic," *Expert Syst. Appl.*, vol. 152, Aug. 2020, Art. no. 113377.

[34] S. Saremi, S. Mirjalili, and A. Lewis, "Grasshopper optimisation algorithm: Theory and application," *Adv. Eng. Softw.*, vol. 105, pp. 30–47, Mar. 2017.

[35] J. Luo, H. Chen, Q. Zhang, Y. Xu, H. Huang, and X. Zhao, "An improved grasshopper optimization algorithm with application to financial stress prediction," *Appl. Math. Model.*, vol. 64, pp. 654–668, Dec. 2018.

[36] M. Sulaiman, "Implementation of improved grasshopper optimization algorithm to solve economic load dispatch problems," *Hacettepe J. Math. Statist.*, vol. 48, no. 5, pp. 1570–1589, 2019.

[37] R. Zhao, H. Ni, and H. Feng, "An improved grasshopper optimization algorithm for task scheduling problems," *Int. J. Innov. Comput. Inf. Control.*, vol. 15, no. 5, pp. 1967–1987, Oct. 2019.

[38] F. Raeesi, B. F. Azar, H. Veladi, and S. Talatahari, "An inverse TSK model of MR damper for vibration control of nonlinear structures using an improved grasshopper optimization algorithm," *Structures*, vol. 26, pp. 406–416, Aug. 2020.

[39] *Next Generation Simulation (NGSIM) Vehicle Trajectories and Supporting Data*. Accessed: Oct. 23, 2019. [Online]. Available: https://data.transportation.gov/Automobiles/NextGeneration-Simulation-NGSIM-Vehicle-Trajector/8ect-6jqj

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

[41] L. Li, M. Zhang, and R. Liu, "The application of Bayesian filter and neural networks in lane changing prediction," in *Proc. ICCET*, Guangzhou, China, 2015, pp. 2004–2007.

[42] K. Kang and H. A. Rakha, "A repeated game freeway lane changing model," *Sensors*, vol. 20, no. 6, p. 1554, Mar. 2020.

[43] O. Scheel, L. Schwarz, N. Navab, and F. Tombari, "Situation assessment for planning lane changes: Combining recurrent models and prediction," in *Proc. ICRA*, Brisbane, QLD, Australia, May 2018, pp. 2082–2088.
[44] 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.
[45] R. Chandra, T. Guan, S. Panuganti, T. Mittal, U. Bhattacharya, A. Bera, and D. Manocha, "Forecasting trajectory and behavior of road-agents using spectral clustering in graph-LSTMs," *IEEE Robot. Autom. Lett.*, vol. 5, no. 3, pp. 4882–4890, Jul. 2020.

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.

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 of computer application technology with Jilin University. She has published over 60 papers in journals and international conference. Her research interests include big data, 3D visual modeling, 3D 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. From 2015 to 2017, he was a Graduate Student with Jilin University. His research interests include swarm intelligence algorithm, machine learning, image processing, data mining, and 3D 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 at Space Technology (Jilin) Company Ltd. Her research interests include media technology ethics, network ecological, and feature selection.

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.