

Received April 28, 2020, accepted May 17, 2020, date of publication May 21, 2020, date of current version June 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2996203

An Improved LSTM Model for Behavior Recognition of Intelligent Vehicles

HAIPENG XIAO¹, MIGUEL ANGEL SOTELO², (Fellow, IEEE), YULIN MA³, (Member, IEEE),
BO CAO¹, YUNCHENG ZHOU⁴, YOUCUN XU⁵, RENDONG WANG⁵,
AND ZHIXIONG LI⁶, (Senior Member, IEEE)

¹Department of Joint Delivery, Army Military Transportation University, Tianjin 300181, China

²Department of Computer Engineering, University of Alcalá, 28801 Alcalá de Henares, Spain

³Suzhou Automotive Research Institute, Tsinghua University, Suzhou 215134, China

⁴ITS Research Center, China Design Group Company Ltd., Nanjing 210014, China

⁵Institute of Military Transportation, Army Military Transportation University, Tianjin 300181, China

⁶School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong, NSW 2522, Australia

Corresponding authors: Yulin Ma (mayulin@tsari.tsinghua.edu.cn) and Rendong Wang (wr1992@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0100903, in part by NSFC under Grant 51979261, in part by the Jiangsu Industrial Technology Research Institute (JITRI) Suzhou Automotive Research Institute Project under Grant CEC20190404, in part by the Taishan Scholar of Shandong Province of China under Grant tsqn201812025, and in part by the Australia Research Council under Grant DE190100931.

ABSTRACT Long Short-Term Memory (LSTM) neural network has been widely used in many applications, but its application in classification of vehicle movement patterns is still limited. In this paper, LSTM is applied to the vehicle behavior recognition problem to identify the left turn, right turn and straight behavior of the vehicle at the intersection. On the basis of the traditional LSTM classification model, this paper transversely merges the input features and then inputs into a LSTM cell to get an improved model. The improved model can make full use of the input information and reduce unnecessary calculations, and the output of a single LSTM cell model can filter out interference information and retain important information, so it has better classification effect and faster training speed. The experimental results show that the proposed improved LSTM network classification model in this paper has a significant improvement in recognition accuracy and training speed compared with the improved model, the accuracy is increased by 1.6%, and the training time is reduced by 3.96 s. In addition, this paper also applies the improved model to regression problems, emotion classification and handwritten digit recognition and all of them have a good improvement effect, which improves the applicability and stability of LSTM in classification problems and provides a new way to deal with classification problems.

INDEX TERMS Intelligent vehicles, LSTM.

I. INTRODUCTION

In the field of self-driving, the trajectory prediction of dynamic vehicles in surrounding environment is very important for the safety and comfort of the vehicles, and it is the focus of research, and the correct identification of vehicle behavior is the premise of accurate trajectory prediction.

Aiming at the classification of vehicle behavior characteristics, a lot of research has been carried out at home and abroad. Geng *et al.* [1] used the Hidden Markov

Model (HMM) to learn the behavior characteristics of vehicles; then built a knowledge base and store a priori probability of features based on the scene to predict the vehicle behavior. Edelbrunner and Iossifidis [2] compared the effects of the support vector machine (SVM), feedforward neural network (FNN) and recurrent neural network (RNN) in identifying lane-changing behavior. Li *et al.* [3] established a general framework for multi-agent behavior prediction and tracking, and trained with generative adversarial network (GAN) with distributed learning ability. Yoon and Kum [4] use multilayer perceptron (MLP) for behavior recognition. Previous studies demonstrate effectiveness of the application of artificial

The associate editor coordinating the review of this manuscript and approving it for publication was J. D. Zhao.

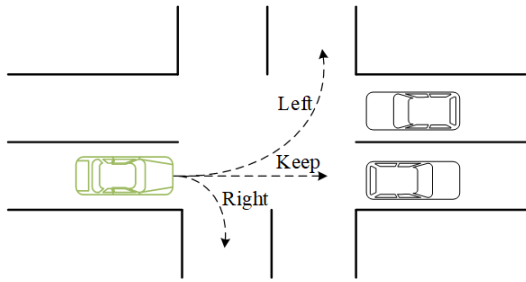


FIGURE 1. Common behaviors of a vehicle at crossroads.

neural networks in vehicle behavior identification. However, very limited tries have been done and the recognition accuracy needs improvement.

In order to identify the left-turn, right-turn and straight-going behaviors of driverless vehicles at the intersection [5]–[9], as shown in Fig. 1, machine learning is adopted in this work. Considering the strong long-term memory function of LSTM, this paper proposed an improved LSTM model to increase the recognition accuracy of vehicle behaviors and compares it with the MLP method in [4]. Experimental results show that LSTM is able to connect different features to achieve better classification performance than MLP.

Since the birth of LSTM, there have been many improvements on this algorithm, including the establishment of encoders and decoders based on LSTM, the use of attention mechanism to enhance learning efficiency and so forth. These improvements further promote the applications of LSTM in translate languages, control robots, image analysis, document summaries, speech recognition, image recognition, handwriting recognition, chatbot control, disease prediction, click-through rates and stocks, synthesize music and many other tasks [10]–[19]. Graves *et al.* [20] use the bi-directional deep recurrent neural network constructed by LSTM unit and has successfully carried out the speech recognition of English essays collection TIMIT. The recognition accuracy is higher than that of HMM and depth feedforward neural network under the same conditions. Sutskever *et al.* [21] used LSTM with end-to-end learning to successfully translate French-English text. In the field of speech synthesis (speech synthesis), Zen and Sak [22] combined multiple bi-directional LSTM to establish a low-delay speech synthesis system, which successfully converted English text into near-real speech output. The convolutional neural network (CNN) has also combined with LSTM. For example, He *et al.* [23] used CNN to extract features from images containing characters in text recognition, and fed the features into a LSTM model for sequence labeling, Pareek and Kesavadas [24] propose a novel LSTM-based robot learning from demonstration (LfD) paradigm to mimic a therapist's assistance behavior.

LSTM has done a very good job in dealing with time series problems, but there is still a lot of room for research on classification. This paper finds that the traditional LSTM network model cannot produce satisfactory classification results in

vehicle behavior recognition. In order to improve the traditional LSTM model, this paper proposes an improved LSTM model for vehicle behavior recognition.

Traditional LSTM model adopts a many-to-one way for classification problem, which abandons many nodes in the output layer. In fact, due to the abandonment of a large number of nodes, useful information may lose. As a result, the traditional LSTM is only suitable to process a few input features. To this end, this paper develops an improved LSTM model by horizontally merging the input vectors, so that the model can get correctly formatted output data without discarding nodes. Because of little loss of information, higher accuracy can be achieved. The main contributions of this paper are as follows:

- (1) The proposed LSTM model is not only suitable for solving classification problems, but also suitable for regression problems.
- (2) Compared with the traditional LSTM classification model, the improved model has wider applicability, more stable and excellent performance.

The remainder of this paper is organized as follows. In Section 2 the proposed method is introduced. Section 3 conducts the experimental tests and analyses the results. Section 4 concludes the findings.

II. METHODS

Long Short-term Memory network (LSTM) is a recurrent neural network (RNN) structure [25]–[27]. LSTM mainly solves the problems of gradient explosion (gradient explosion) and gradient vanishing (gradient vanishing) of RNN [28]–[31]. In the field of deep learning, LSTM belongs to feedback neural network. The difference between LSTM and RNN lies in the addition of three gates: the forget gate, the input gate and the output gate. As shown in figure 3, the input gate determines how much new information will be added to the cell state, the forget gate determines which information to be discarded from the cell, and the output door determines what information needs to be output. LSTM uses three such gate structures to protect and control information.

LSTM realizes the function of long-term memory through long-term memory C, because there is only simple multiplication and addition on the track of long-term memory, and there is no nonlinear operation, so information flows more smoothly at different times, which can effectively restrain the problem of gradient dissipation of long-term memory. In the classification problem, we can establish a relationship between the input parameters (that is, input features), so as to achieve a better classification effect. The following is a further introduction to the structure and principle of LSTM cells. Two functions, sigmoid function and tanh function, are mainly used in LSTM cell structure, such as equation (1) and (2).

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (1)$$

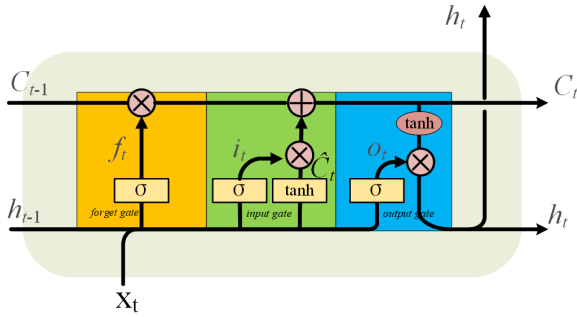


FIGURE 2. Structure of LSTM.

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

The cell structure of LSTM is shown in Figure 2. From left to right, there is a forget gate, an input gate, and an output gate. The working process of the three gates is as follows. Forget gate is shown in Eq. (3).

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (3)$$

In Fig. 2, h_{t-1} represents the short-term memory state transmitted by the previous LSTM unit; x_t represents the state of the input; f_t represents the forgetting factor; σ denotes the sigmoid function. The output of the forget gate determines the retention of information. The input gates are expressed as

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \quad (4)$$

$$\hat{C}_t = \tanh(W_C[h_{t-1}, x_t] + b_C) \quad (5)$$

$$C_t = f_t * C_{t-1} + i_t * \hat{C}_t \quad (6)$$

where, \hat{C}_t represents the alternative vector required for the update, and C_t represents the resulting new long-term memory. The output gate is

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \quad (7)$$

$$h_t = o_t * \tanh(C_t) \quad (8)$$

where, o_t represents the switch that output gate at the current moment, and h_t represents the final output of short-term memory status information. It is through this gate structure that LSTM realizes the function of long-term and short-term memory.

A. TRADITIONAL LSTM NETWORK CLASSIFICATION MODEL

The structure of the traditional LSTM network classification model is shown in Figure 3. From the above analysis of LSTM cell structure principle, we can know that the output of single LSTM cell structure is h_t , as shown in Eq. (8). For the traditional classification model, the traditional model divides the input features into $x_1, x_2, \dots, x_{\text{seq_length}}$ and input them into seq_length LSTM cells respectively; the output of the

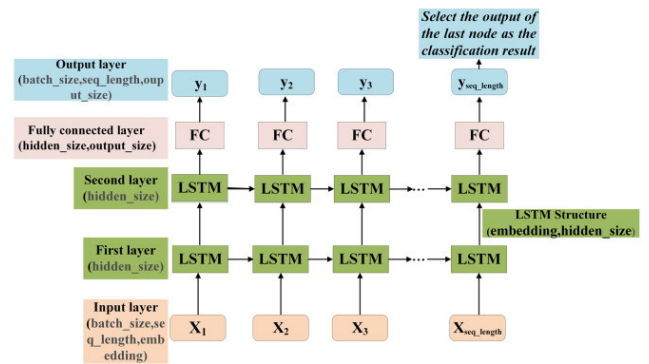


FIGURE 3. Traditional LSTM classification model.

Algorithm 1 Traditional Model

1. input: x
2. output: out
3. $(batch_size, seq_length, embedding) \leftarrow x.shape$;
4. $LSTM(embedding, hidden_size, 2) \leftarrow LSTM$;
5. $((2, batch_size, hidden_size), (2, batch_size, hidden_size)) \leftarrow hidden$
6. $lstm_out, hidden \leftarrow LSTM(x, hidden)$;
7. $Linear(hidden_size, output_size) \leftarrow FC$;
8. $fc_out \leftarrow lstm_out * FC$;
9. $out \leftarrow fc_out[-1]$;
10. end;
11. **return** out ;

traditional LSTM network classification model is as Eq. (9).

$$lstm_out = \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_{\text{seq_length}} \end{bmatrix} \quad (9)$$

The basic steps of the traditional model are as follows:

Where, Seq_length is the number of features of the input LSTM network, that is, the number of LSTM units. Embedding is the length of the vector corresponding to the feature input LSTM unit, that is, the length of the word vector in dealing with the text problem. For the non-text data such as the vehicle behavior data set, $embedding = 1$. The data is loaded by batch training, and the $batch_size$ is the loading size. The $hidden_size$ is the number of LSTM hidden layer nodes. The $output_size$ is the output category size. The n_layers is the number of hidden layers of LSTM. Finally, a full connection layer FC is used to achieve the final classification. In this model, only the last node is taken as the output, that is, only the last node $y_{\text{seq_length}}$ is taken as the classification result. All the other nodes are discarded. The output layer does not use activation functions. Use Adam to update weights, and $learning_rate$ is the learning rate. The loss function is crossentropyloss function L (CrossEntropyLoss Funcyion).

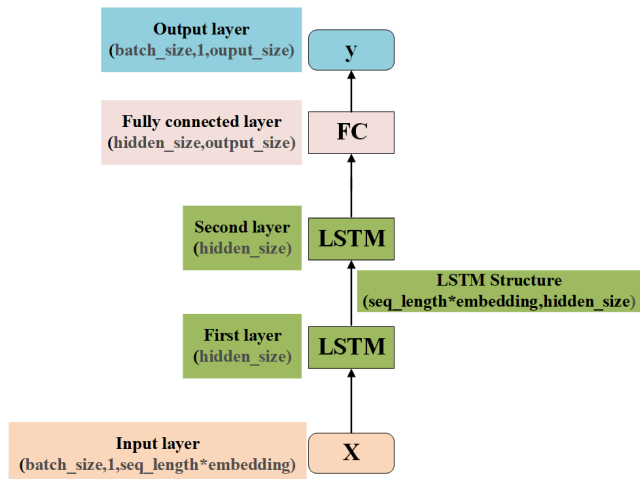


FIGURE 4. Improved LSTM classification model.

As shown in formula (10).

$$L = - \sum_i^N y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \quad (10)$$

where, $y^{(i)}$ is the actual value, $\hat{y}^{(i)}$ is the predicted value.

B. IMPROVED LSTM NETWORK CLASSIFICATION MODEL

From the traditional LSTM classification model in Figure 3, we can see that the output node of the traditional model is equal to the number of input features, but in the end, only the last node is taken and the rest of the nodes are lost. On the one hand, discarding nodes will lose information, resulting in poor training effect; on the other hand, discarded nodes actually participate in the computing process, which will consume computing resources and slow down the training speed. As shown in Eq. (11), it can be seen that the amount of retained information is inversely proportional to the number of input features. It can be known that the more the number of input features, the worse the training effect of the traditional model will be, and the slower the training speed will be. Therefore, this paper makes some improvements to the traditional model to solve the above problems. The structure of the model is improved as shown in Figure 4.

$$information_reserve = \frac{1}{seq_length} * information \quad (11)$$

After merging the input features, the input features are input into 1 LSTM cell, and the merging vector is like formula (12).

$$X = [x_1 \ x_2 \ \dots \ x_{seq_length}] \quad (12)$$

Reference formula (8), then the output of the improved LSTM network classification model is like formula (13).

$$lstm_out = [h_t] \quad (13)$$

The basic steps to improve the model are expressed as **Algorithm 2**. On the one hand, the improved model does

not discard nodes, and complete information can be obtained, which ensures that the improved model has a better classification effect than the traditional LSTM classification model; on the other hand, it can be seen from figures 4 and 5 that the improved model has fewer nodes, which can ensure that the model has faster training speed. The improved model has only one LSTM cell, so it can filter the interference information and save the important information, so that the classification effect of the model can be further improved.

Algorithm 2 Improved Model

1. input: x
2. output: out
3. (batch_size, 1, seq_length * embedding) ← x.shape;
4. LSTM(seq_length * embedding, hidden_size, 2) ← LSTM
5. ((2, batch_size, hidden_size), (2, batch_size, hidden_size)) ← hidden
6. lstm_out, hidden ← LSTM(x, hidden);
7. Linear(hidden_size, output_size) ← FC;
8. fc_out ← lstm_out * FC;
9. out ← fc_out;
10. end;
11. **return** out;

In order to ensure the rigor of the comparison of the experimental results, the parameters of the improved model are consistent with the traditional model.

III. EXPERIMENT

The minimum verification loss of training, the loss of test set, the accuracy of test set and the training time are taken as the indexes to judge the effect of the model.

A. EXPERIMENT PLATFORM

In this paper, the experiment platform is carried out on the ubuntu16.04 system, the GPU is GTX 1050Ti and GTX 1650, and the network model is built on jupyter notebook based on pytorch.

B. DATA SOURCES AND PREPROCESSING

Training is mainly carried out on four data sets. One is the vehicle behavior data set provided by Udacity which is the idea source of the improved model; one is the NGSIM vehicle trajectory data set; the other is the movie review emotion classification data set provided by Udacity and the last is the Mnist handwritten digital data set. In order to better observe the training process and test the model, this paper divides the test set into two, half as the test set and half as the verification set. In the training process, the verification set is used to monitor the training process in real time. After the training, the test set is used to test the effect of the model. The data set is loaded by batch training uniformly, and the batch size is set to batch_size.

1) PREPROCESSING OF VEHICLE BEHAVIOR DATA SET

The vehicle behavior data set is provided by the self-driving engineer course of Udacity, which records the straight, left and right turns of the vehicle at the crossroads, and collects the corresponding horizontal and longitudinal coordinates and horizontal and longitudinal speeds at the same time. Its data structure is (x, y, v_x, v_y) .

It mainly deals with the label of the data set and transforms the label into the number that the model can train. Use 0, 1, and 2 to represent the left-turn (left), straight-turn (keep), and right-turn (right) of the vehicle, respectively. There are a total of 1000 data, including training set, verification set and test set, accounting for 75%, 12.5% and 12.5%, respectively.

2) PREPROCESSING OF NGSIM VEHICLE TRACK DATA SET

The data set, which is based on the Next Generation Simulation (NGSIM) program initiated by the Federal Highway Administration, has a sampling frequency of 10 Hz and records information including vehicle coordinates, speed, acceleration, vehicle type and lane number NGSIM data set is a CSV format file, each column represents a feature, a total of 25 columns, 25 features. Each line represents the data of a car, there are a total of 1676606 sampling points, and the number of vehicles collected is 1,545. Because there are some errors and noise in the original data, especially the obvious jitter of the data signal, the symmetrical exponential moving average filtering algorithm is used to filter the original data. Then normalize the four columns of data of Location, V_class, Direction and Movement. for the two columns of Preceding and Following, mark each data that is not zero as 1, and the data that is 0 as 0. Finally, the data of each column is standardized, as shown in Eq. (14).

$$x = \frac{\hat{x} - \text{mean}}{\text{max} - \text{min}} \quad (14)$$

\hat{x} indicates that a single value in each column, x indicates that the changed value, mean represents the average value of each column, max represents the maximum value, min represents the minimum value.

[Location_X, Location_Y, v_length, v_width, v_class_1, v_Class_2, v_Class_3, v_Vel, v_Acc, Preceding, Following, Direction_1, Direction_2, Direction_3, Direction_4, Movement_1, Movement_2, Movement_3] is the selected feature as the input to the dataset, and the [Location_X, Location_Y] of the next input data as the tag. The purpose of the training is that the model can infer the coordinates of the next point based on an input value. Training set, verification set and test set account for 80%, 10% and 10%, respectively.

3) EMOTION CLASSIFICATION DATA SET PREPROCESSING

For text datasets, this dataset is mainly divided into two parts: features and labels. Features dataset is a collection of people's comments on a movie, and its tags correspond to positive and negative, in the label file to express the emotions expressed by the comments. This paper deals with the data set as follows:

- (1) First unify the words of the comments to lowercase, and then remove the punctuation marks.
- (2) Put all the comment words together, count the frequency of each word, mark all the words according to the frequency from small to large, form a vocabulary, mark them from 1 instead of 0, that is, the word with the highest frequency corresponds to the number 1, and so on, so that each word corresponds to a different number.
- (3) Use glossary to assign values to words in comment.
- (4) Unify the length of each comment. Given a length seq_length, if the number of comment words in a paragraph is less than seq_length, use 0 to fill the missing part; if the number of comment words in a paragraph is greater than seq_length, cut off the redundant part.
- (5) For label datasets, the main thing is to convert the contents into numbers, using 0 and 1 to express the negative (positive) and positive (positive) of comments, respectively.
- (6) There are a total of 25000 comments in the data set, with training set, verification set and test set accounting for 80%, 10% and 10%, respectively.

4) MNIST HANDWRITTEN DATA SET PREPROCESSING

The handwritten digital data set is a public data set included in pytorch, in which there are 60000 gray images in the training set and 10000 gray images in the test set [32]–[40], corresponding to a label of 0 to 9. Training set, verification set and test set account for 85%, 7.5%, 7.5%, respectively.

C. VEHICLE BEHAVIOR DATA SET EXPERIMENT

1) EXPERIMENTAL METHOD

First of all, this paper uses the naive Bayesian method (Naive Bayesian, NB) in traditional machine learning to identify vehicle behavior, such as formula (15) and (16).

The conditional probability of a feature S_i of the input data S under a tag B_k is calculated based on the formula (15). B_k is the k -th behavior of label B . and S_i is the i -th feature of input data S . μ , σ is the average value and standard variance of a feature S_i set of input data S under B_k , respectively.

$$P(x = S_i | B_k) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(S_i - \mu)^2}{2\sigma^2}\right) \quad (15)$$

The vehicle behavior corresponding to the input data S is calculated based on the formula (16). B_k is the k -th behavior of tag B , $p(B_k)$ is the prior probability of B_k , and S_i is the i -th feature of the input data S .

$$y = \arg \max(p(B_k) \prod_{i=1}^n p(x_i = S_i | B_k))_{k=0,1,2,\dots} \quad (16)$$

This paper also uses traditional machine learning methods such as support vector machine ((Support Vector Machine, SVM) and decision tree (Decision Tree, DT), as well as the k-means algorithm in unsupervised learning (Unsupervised Learning, UL).

TABLE 1. Experimental results of traditional machine learning.

Method	SVM (poly)	SVM (linear)	SVM (rbf)	SVM (sigmoid)
Accuracy	0.956	0.888	0.772	0.396
Method	DT	NB	UL	
Accuracy	0.928	0.844	0.356	

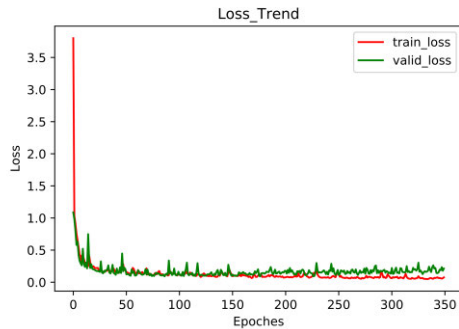


FIGURE 5. Variation of Training loss of MLP experiment.

Then this paper uses MLP network and LSTM network modeling to solve the problem of vehicle behavior recognition, and compares each model, and selects the optimal model as the final model.

After referring to the MLP method used by Yoon and Kum [4], this paper uses MLP to carry out experiments. The MLP model contains three hidden layers, the input layer contains four nodes, the number of nodes in the first hidden layer is 256×4 , the number of hidden nodes in the second layer is 256×6 , the number of nodes in the third layer is 256×4 , and the number of output nodes is output category 3. The method of dropout is used to prevent overfitting. Set the dropout = 0.2, hidden layer to use the ReLu activation function. The output layer does not use activation functions. The loss function is crossentropyloss function L (CrossEntropyLoss Funcyion). Use Adam to update the weight, and set the learning rate to 0.00128, the number of iterations to 350, using batch training to load the data set, batch_size for batch training size, set batch_size = 64.

In order to compare with the MLP model, the batch training size, learning rate and iteration times of the traditional LSTM classification model and the improved LSTM classification model are consistent with the MLP model.

2) TRADITIONAL MACHINE LEARNING METHODS

First of all, the traditional machine learning method is used for classification. The experimental results are shown in Table 1.

From the experimental results, it is obvious that SVM (kernel = poly) is the best, achieving a correct rate of 95.6%, but still falling short of the expected goal.

3) MLP EXPERIMENT

After using the machine learning method, use MLP for training. The training process is shown in Figures 5 and 6.

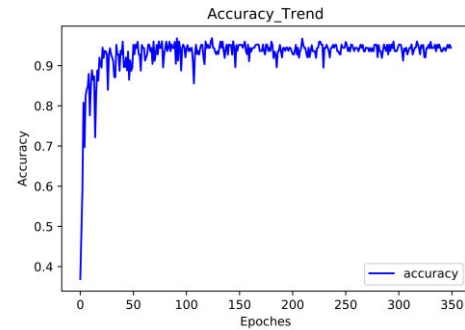


FIGURE 6. Variation of training accuracy of MLP experiment.

TABLE 2. Experimental results of MLP.

	Left	Keep	Right	ALL	Test_Loss
Correct_n/ALL	41/42	46/49	34/34	121/125	0.1152
Accuracy	0.9762	0.9387	1	0.968	

Finally, the minimum verification loss is 0.07991, and the training time is 37.472s. From the perspective of the training process, the verification loss of training began to rise at about 170 times, but not decreased, indicating that the model began to fit and finally used the test set to test the trained model, the results are shown in Table 2.

From the test results, the correct rate of MLP is much higher than that of SVM, but from the test results, the classification effect of MLP on data such as keep is not good, considering that it may be because the data jitter occurs in the x direction when the vehicle keeps going straight, which leads to the model misclassifying keep into Left or Right, keeping straight in the actual situation, so it is particularly important to accurately identify Keep. Therefore, it is decided to consider a model that can connect different features, that is, the traditional LSTM network classification model.

4) TRADITIONAL LSTM CLASSIFICATION MODEL

Set the number of batch_size = 64, embedding = 1, hidden_size = 64, n_layers = 2, output_size = 3, dropout = 0, learning_rate = 0.00128, iterations to 350. The output layer does not use activation functions. Save only the model with minimal validation loss. The training process is shown in Figures 7 and 8.

Finally, the minimum verification loss is 0.07724 and the training time is 25.178s. From the perspective of the training process, the verification loss of training began to rise at about 150 times, but not decreased, indicating that the model began to be over-fitted. Finally, the test set is used to test the trained model, and the results are shown in Table 3.

From the test results, compared with the MLP model, the verification loss and test loss of the LSTM model are reduced, and the accuracy is improved. In particular, the classification accuracy of Keep has been improved by 2.04%. The training time was reduced by 12.294 s.

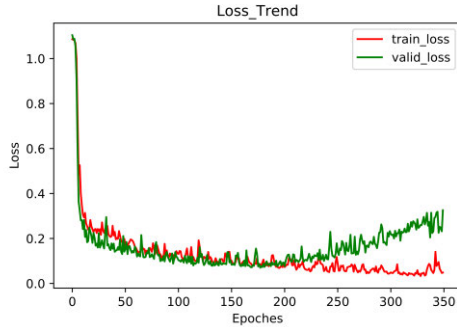


FIGURE 7. Variation of training loss of traditional LSTM experiment.

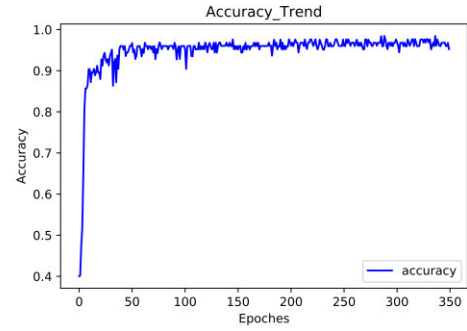


FIGURE 10. Variation of training accuracy of improved LSTM experiment.

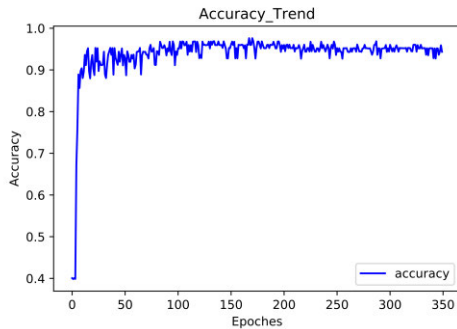


FIGURE 8. Variation of training accuracy of traditional LSTM experiment.

TABLE 3. Experimental results of traditional LSTM.

	Left	Keep	Right	ALL	Test_Loss
Correct_n/ALL	42/42	47/49	33/34	122/125	0.077
Accuracy	1	0.9592	0.9706	0.976	

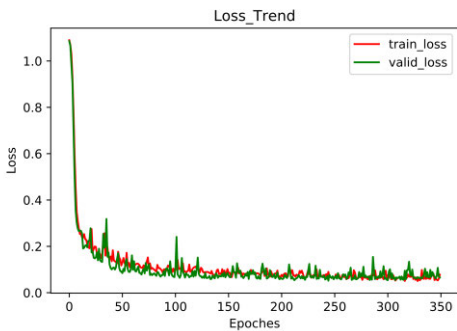


FIGURE 9. Variation of training loss of improved LSTM experiment.

5) IMPROVED LSTM CLASSIFICATION MODEL

The following experiment is carried out to improve the LSTM classification model, setting the number of iterations to 350. Save only the model with minimal validation loss. The training process is shown in Figures 9 and 10.

Finally, the minimum verification loss is 0.05537 and the training time is 21.223s. From the point of view of the training process, there is basically no fitting in the improved model, and the model has been learned all the time. Use the

TABLE 4. Experimental results of improved LSTM.

	Left	Keep	Right	ALL	Test_Loss
Correct_n/ALL	42/42	48/49	34/34	124/125	0.036
Accuracy	1	0.9796	1	0.992	

TABLE 5. Experimental results of traditional LSTM.

Min valid loss	Test loss	Training time
0.00066	0.001	9180.643s

test set to test the trained model, and the results are shown in Table 4.

From the test results, the overall accuracy of the improved LSTM classification model is improved by 1.6%. The accuracy of keep is improved by 2.04%, and the accuracy of the other two is up to 100%. The minimum verification loss and test loss are reduced, and the training time is reduced by 3.955 s. The experimental results show that the improved method can improve the traditional LSTM classification model.

D. NGSIM DATASET VERIFICATION EXPERIMENT

1) TRADITIONAL LSTM CLASSIFICATION MODEL

Set the number of batch_size = 512, embedding = 1, hidden_size = 256, n_layers = 2, output_size = 2, dropout = 0, learning_rate = 0.0001, iterations to 30. The mean square error loss function (MSELoss), is used here as shown in equation (17).

$$MSELoss = \frac{1}{2n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \tag{17}$$

where, n represents the total number of variables, y_i represents the actual value, and \hat{y}_i represents the predicted value.

Only models with minimum validation losses are saved. The experimental results are shown in Table 5.

From the perspective of the training process, the effect of training is not ideal, especially the training speed is very slow.

2) IMPROVED LSTM CLASSIFICATION MODEL

Only models with minimum validation losses are saved. The training results are shown in Table 6.

TABLE 6. Experimental results of improved LSTM.

Min valid loss	Test loss	Training time
0.0006898	0.001	3126.329s



FIGURE 11. Variation of training loss of traditional LSTM experiment.

Comparing the two results, it is found that when the training effect is similar, the time of the traditional LSTM classification model is about 3 times longer than that of the improved LSTM classification model, the more time is 6054.314s, and the average time of each iteration is 201.81 s. When the effect is similar, the model with faster training obviously has more advantages. It shows that the improved model is more suitable for dealing with this kind of regression problem.

E. EMOTION CLASSIFICATION DATASET VERIFICATION EXPERIMENT

In order to prove that the improved model is more universal, the experiment is changed from GPU to GTX 1650, and the rest remains unchanged.

The loss function is binary cross-entropy loss function (BCEloss), sets output_size to 1, and uses an sigmoid activation function in the output layer to keep the output value between 0 and 1. When the output value is less than 0.5, the predicted value is 0, and when the output value is greater than 0.5, the predicted value is 1. This kind of text class problem needs to go through an embedded layer to process the data before entering the LSTM, and its function is to assign a fixed-length (embedding) vector to each input word.

1) TRADITIONAL LSTM CLASSIFICATION MODEL

Set the number of batch_size = 50, embedding = 256, hidden_size = 128, n_layers = 1, output_size = 1, dropout = 0, learning_rate = 0.01, seq_length = 400, iterations to 4. The training process is shown in Figures 11 and 12.

Finally, the minimum verification loss is 0.4612 and the training time is 91.007s. From the perspective of the training process, the verification loss of training began to rise at about 6, but not decreased, indicating that the model began to be over-fitted. Use the test set to test the trained model, and the results are shown in Table 7.

From the training results, the results are not ideal.

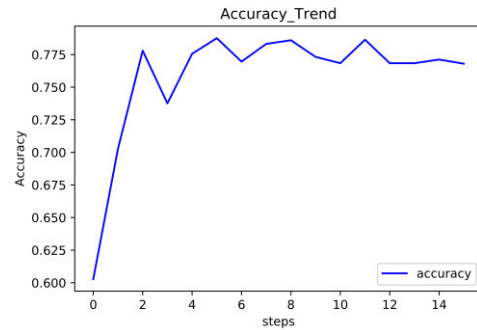


FIGURE 12. Variation of training accuracy of traditional LSTM experiment.

TABLE 7. Experimental results of traditional LSTM.

	Negative	Positive	All	Test_Loss
Correct_n/ALL	1012/1250	967/1250	1979/2500	0.463
Accuracy	0.8096	0.7736	0.7916	

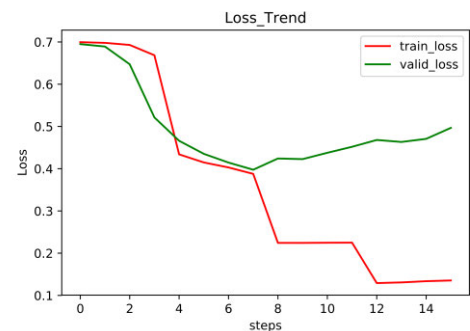


FIGURE 13. Variation of training loss of improved LSTM experiment.

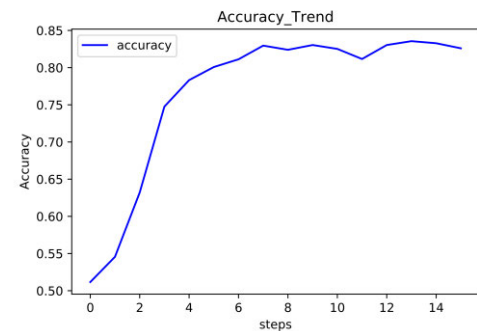


FIGURE 14. Variation of training accuracy of improved LSTM experiment.

2) IMPROVED LSTM CLASSIFICATION MODEL

After repeated experiments, it is found that the training effect of embedding = 1 is the best under the improved model. The rest is the same as the traditional LSTM model. The training process is shown in Figures 13 and 14.

Finally, the minimum verification loss is 0.4051 and the training time is 5.051 s. From the perspective of the training process, the verification loss of training began to rise at about 11, but not decreased, indicating that the model began to be

TABLE 8. Experimental results of improved LSTM.

	Negative	Positive	All	Test_Loss
Correct_n/ALL	1099/1250	992/1250	2091/2500	0.406
Accuracy	0.8792	0.7936	0.8364	

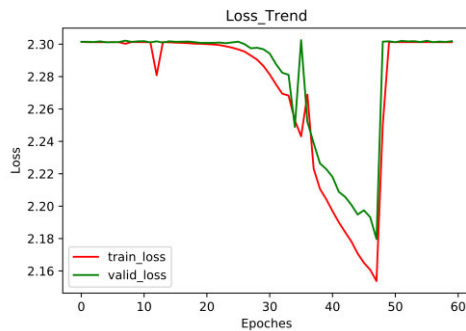


FIGURE 15. Variation of training loss of traditionalLSTM experiment.

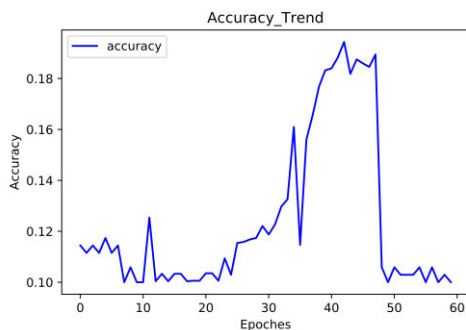


FIGURE 16. Variation of training accuracy of traditional LSTM experiment.

over-fitted. Use the test set to test the trained model, and the results are shown in Table 8.

Comparing the two results, it is found that both the minimum verification loss and the test loss are reduced, and the accuracy is improved by about 5%, especially the training time. The improved model only needs 5.051s, which is 85.949s less. The experimental results show that the improved model has a certain improvement effect on text classification.

F. MNIST HANDWRITTEN DIGITAL DATA SET VERIFICATION EXPERIMENT

When using the traditional LSTM classification model, this paper refers to the practice of MLP, that is, the image of 1*28*28 is flattened into the format of 1*784 (28*28 = 784), and then the data is input into the model training.

1) TRADITIONAL LSTM CLASSIFICATION MODEL

Set the number of n_layers = 2, batch_size = 128, hidden_size = 128, embedding = 1, output_size = 10, dropout = 0, learning_rate = 0.0001, seq_length = 784, iterations to 60. The training process is shown in Figures 15 and 16.

Finally, the minimum verification loss is 2.056 and the training time is 5230.121s. From the perspective of the

TABLE 9. Experimental results of traditional LSTM.

	0	1	2	3
Correct_n/ALL	0/520	458/564	0/502	0/510
Accuracy	0	0.8120	0	0
	4	5	6	7
Correct_n/ALL	0/482	0/436	0/496	514/516
Accuracy	0	0	0	0.9961
	8	9	ALL	Test loss
Correct_n/ALL	0/485	0/489	972/5000	2.0716
Accuracy	0	0	0.194	

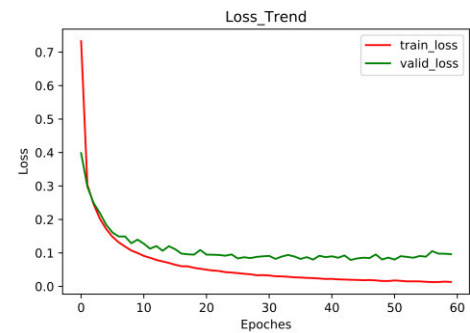


FIGURE 17. Variation of training loss of improved LSTM experiment.

training process, the verification loss of the training has been relatively large, the accuracy of the verification set has been low, and the training speed is very slow. Use the test set to test the trained model, and the results are shown in Table 9.

The experimental results are not ideal, and the accuracy can only reach about 20%, because this traditional many-to-one model takes only the last node when taking the final result, while the result of the handwritten image actually depends on 784 pixels. Compared with the previous vehicle behavior data set, the reason why the effect of the two experiments on the traditional LSTM network classification model is huge is that there is a huge difference in the number of feature points between the two. The vehicle behavior data set only needs to abandon 3 nodes, while the latter needs to abandon 783 nodes, resulting in nothing to be learned from the model. From the test results, the model only has a relatively good learning effect for the relatively simple numbers 1 and 7, while for other more complex numbers, the model basically learns nothing.

2) IMPROVED LSTM CLASSIFICATION MODEL

The improved LSTM network classification model is used to carry out the experiment. The training process is shown in Figures 17 and 18.

Finally, the minimum verification loss is 0.08147 and the training time is 595.94s. From the perspective of the training process, the verification loss of training has been declining, and tends to be flat at about 35. Compared with the traditional training model, the training of the improved model is much smoother, which shows that the improvement effect of the improved model is obvious. Use the test set to test the trained model, and the results are shown in Table 10.

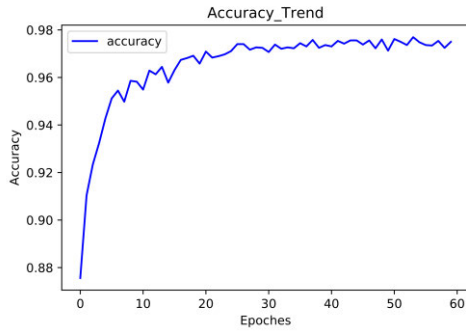


FIGURE 18. Variation of training accuracy of improved LSTM experiment.

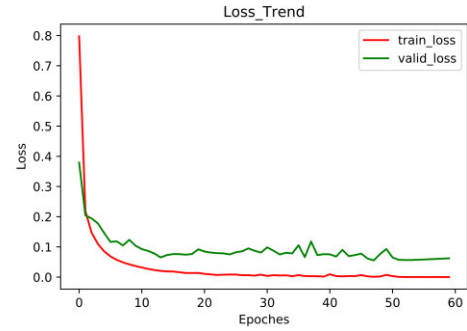


FIGURE 19. Variation of training loss of Non-flattening Method.

TABLE 10. Experimental results of improved LSTM.

	0	1	2	3
Correct_n/ALL	516/520	560/564	497/502	503/510
Accuracy	0.9923	0.9929	0.9900	0.9862
	4	5	6	7
Correct_n/ALL	476/482	432/436	495/496	511/516
Accuracy	0.9875	0.9908	0.9979	0.9903
	8	9	ALL	Test_loss
Correct_n/ALL	480/485	484/489	4954/5000	0.032
Accuracy	0.9896	0.9897	0.9910	

Compared with the traditional classification model, the effect of the model proposed in this paper is much better. In this experiment, the shortcomings of the traditional LSTM network classification model are magnified, although the long-term memory function of LSTM can be used to connect each feature, but the more feature points of classification, the more nodes it finally abandons, resulting in worse classification results. The improved model proposed in this paper has good applicability, regardless of whether there are many or few features, the performance is better than MLP and traditional LSTM classification model, because it can not only use each feature like MLP, but also automatically filter out interference information and retain important information.

3) EXPERIMENT OF NON-FLATTENING METHOD

After referring to Graves’s paper on using RNN to recognize Mnist handwritten digits [23] and some LSTM-based Mnist handwritten digit recognition codes on github, this paper uses the non-flattening method to carry out experiments, which can be used as a reference for improving the model in this paper, from which we can see both advantages and disadvantages. The main methods are as follows:

- (1) Traditional LSTM classification model is used.
- (2) The input of handwritten digital graphics is not flattened, and the input format of the data is kept as 1*28*28, that is, seq_length = 28, embedding = 28, which retains the two-dimensional plane information of the image. The other parameters are the same as the 4.6.2 experiment.

The training process is shown in Figures 19 and 20.

Finally, the minimum verification loss is 0.0578 and the time-consuming is 1854.27s. After using this data processing

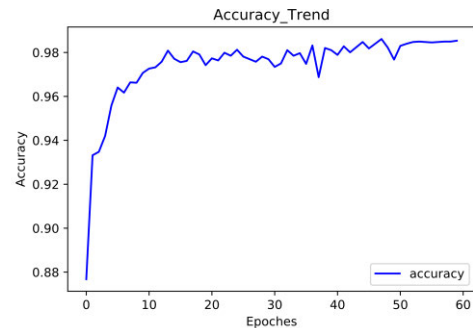


FIGURE 20. Variation of training accuracy of Non-flattening Method.

TABLE 11. Experimental results of Non-flattening Method.

	0	1	2	3
Correct_n/ALL	517/520	563/564	501/502	509/510
Accuracy	0.9942	0.9982	0.998	0.998
	4	5	6	7
Correct_n/ALL	477/482	431/436	494/496	512/516
Accuracy	0.9896	0.9885	0.9959	0.9922
	8	9	ALL	Test_loss
Correct_n/ALL	480/485	487/489	4971/5000	0.026
Accuracy	0.9896	0.9959	0.994	

method, the effect of training is very good. Use the test set to test the trained model, as shown in Table 11.

From the test results, each item of data is slightly better than the model proposed in this paper, the main reason is that this data processing method not only retains the two-dimensional spatial characteristics of the image (equivalent to putting the picture into training directly, similar to cnn), but also reduces the number of discarded nodes. However, But its training time is still more than three times longer. The improved model can also have a better training effect when the training effect of the traditional model is very poor, which shows that the performance of the improved model is stable.

G. DISCUSSION

In order to facilitate the statistical experimental results, the traditional LSTM classification model, the improved LSTM classification model and the non-flattening method are abbreviated as TLMC, ILCM and NFM respectively. The vehicle behavior data set, NGSIM vehicle track data set, movie

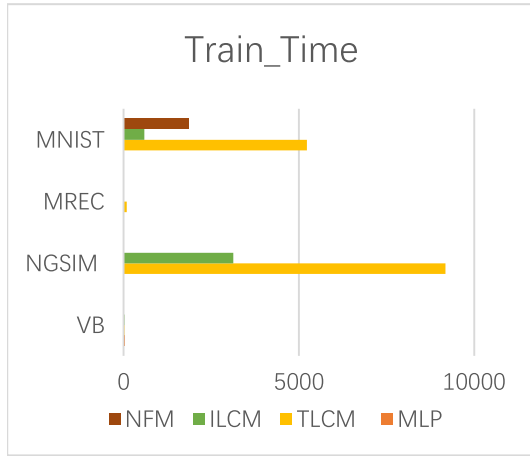


FIGURE 21. Overall training time.

TABLE 12. Overall training time.

Data set	VB	NGSIM	MREC	MNIST
NFM				1854.27
ILCM	21.22	3126.33	5.05	595.94
TLMC	25.18	9180.64	91.00	5230.12
MLP	37.47			

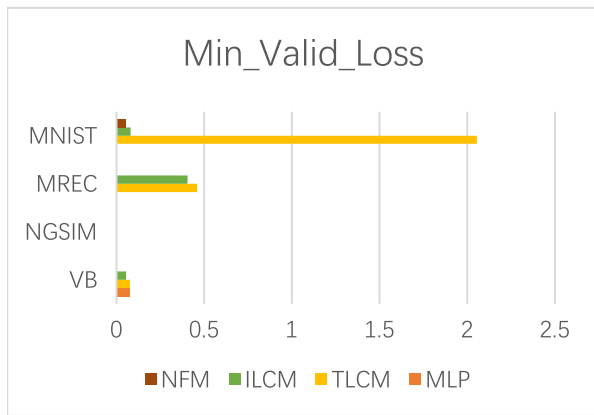


FIGURE 22. Overall training Min_valid_loss.

review emotion classification data set and Mnist handwritten digital data set are abbreviated as VB, NGSIM, MREC and MNIST, respectively.

1) TRAINING SPEED

The comparison results are shown in Figure 21 and Table 12.

It can be seen from the table that the training time of the improved LSTM classification model is the least under all data sets, which shows that the improved model improves the training speed.

2) MINIMUM VERIFICATION LOSS

The comparison results are shown in Figure 22 and Table 14.

It can be seen from the table that, except for NGSIM data sets, the minimum verification loss of the improved LSTM classification model is smaller than that of the traditional LSTM classification model.

TABLE 13. Overall training Min_valid_loss.

Data set	VB	NGSIM	MREC	MNIST
NFM				0.05780
ILCM	0.05537	0.00069	0.4051	0.08147
TLMC	0.07724	0.00066	0.4612	2.0560
MLP	0.07991			

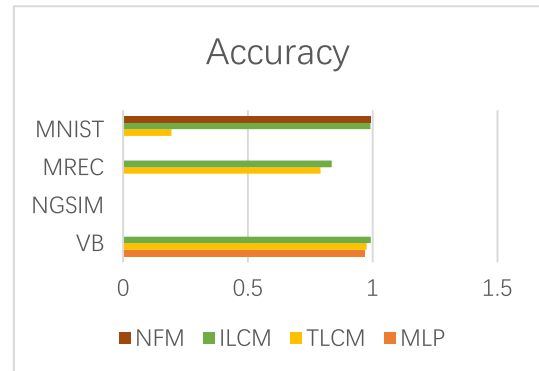


FIGURE 23. Overall training accuracy.

TABLE 14. Overall training accuracy.

Data set	NB	NGSIM	MREC	MNIST
NFM				0.994
ILCM	0.992		0.8364	0.991
TLMC	0.976		0.7916	0.194
MLP	0.968			

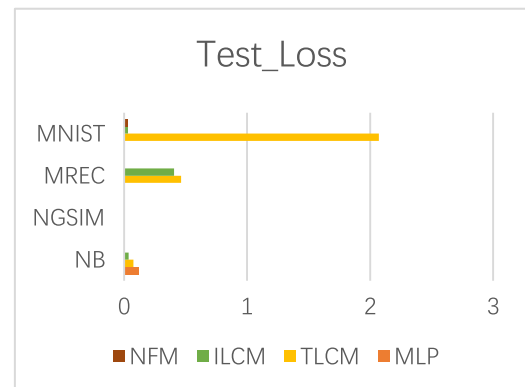


FIGURE 24. Overall training test_loss.

3) ACCURACY

The comparison results are shown in Figure 23 and Table 14.

It can be seen from the table that the accuracy of the improved LSTM classification model is higher than that of the traditional LSTM classification model under all data sets, indicating that the improved model improves the classification accuracy.

4) TEST LOSS

The comparison results are shown in Figure 24 and Table 15.

It can be seen from the table that the test loss of the improved LSTM classification model is smaller than that of the traditional LSTM classification model under all data sets.

TABLE 15. Overall training test_loss.

Data set	VB	NGSIM	MREC	MNIST
NFM				0.026
ILCM	0.036	0.001	0.406	0.032
TLCM	0.077	0.001	0.463	2.0716
MLP	0.1152			

From the comparison of the experimental data, it can be found that compared with the traditional LSTM model, the training speed, accuracy and test loss of the improved model are improved, especially the training speed is obviously faster. These experimental results show that when the traditional model can not be used, the performance of the improved model is still excellent, indicating that it has better stability and adaptability, and its experimental effect is better than the traditional classification model.

IV. CONCLUSION

In this paper, we mainly merge the input vectors of the traditional LSTM model horizontally and then input them into a LSTM cell. Compared with the traditional LSTM classification model, the improved model does not need to discard information, and the improved model can automatically filter out interference information and retain important information, thus better training results can be achieved.

First, experiments are carried out with vehicle behavior data and good experimental results are obtained. Compared with the traditional classification model, the performance has been greatly improved. In order to verify the universality of the improved model, NGSIM data set, emotion classification data set and Mnist handwritten digital data set are used for training respectively. Compared with the traditional LSTM classification model, the improved model has achieved better experimental results. In particular, the last Mnist handwritten digit recognition experiment magnifies and shows the shortcomings of the traditional LSTM network classification model, and further proves that the traditional LSTM classification model is not suitable for multi-feature input data classification, while the improved model still performs well, indicating that the improved model has good stability

Summing up the above experimental results, we can see that the improved model proposed in this paper is more applicable in most fields, and the performance is more excellent and stable. In the experiment, it is also found that the effect of the improved model on text classification is not good enough, mainly because the input size of its embedded layer (embedding) cannot be too large, but the effect is the best when taking 1, which leads to little improvement, and in some cases it is not as good as the traditional classification model. At the same time, the parameters of LSTM cells will be greatly increased if the embedded vector size (embedding) is not 1, which causes GPU memory to overflow easily during training.

In the next step of work, we will continue to optimize this model. In order to solve the problem that the input of the embedded layer is too large, which leads to the overflow of

GPU memory, this paper will consider training on the server with more GPU memory so that the experimental data can be fully collected for improvement. In this paper, we will use more data sets to verify the model, and find and solve the problems, such as replacing the full connection layer in the convolution neural network to deal with some image classification problems.

REFERENCES

- [1] X. Geng, H. Liang, B. Yu, P. Zhao, L. He, and R. Huang, "A scenario-adaptive driving behavior prediction approach to urban autonomous driving," *Appl. Sci.*, vol. 7, no. 4, p. 426, 2017.
- [2] U. Dogan, J. Edelbrunner, and I. Iossifidis, "Autonomous driving: A comparison of machine learning techniques by means of the prediction of lane change behavior," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Dec. 2011, pp. 1837–1843.
- [3] J. Li, H. Ma, and M. Tomizuka, "Interaction-aware multi-agent tracking and probabilistic behavior prediction via adversarial learning," 2019, *arXiv:1904.02390*. [Online]. Available: <http://arxiv.org/abs/1904.02390>
- [4] S. Yoon and D. Kum, "The multilayer perceptron approach to lateral motion prediction of surrounding vehicles for autonomous vehicles," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2016, pp. 1307–1312.
- [5] K. I. Ahmed, "Modeling driver's acceleration and lane changing behaviors," Ph.D. dissertation, Dept. Civil Environ. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1999.
- [6] P. G. H. Bham, "Estimating driver mandatory lane change behavior on a multi lane freeway," in *Proc. Transp. Res. Board 88th Annu. Meeting*, 2009, pp. 1–22.
- [7] C. F. Choudhury and M. E. Ben-Akiva, "Lane selection model for urban intersections," *Transp. Res. Rec., J. Transp. Res. Board*, vol. 2088, no. 1, pp. 167–176, Jan. 2008.
- [8] P. G. Gipps, "A model for the structure of lane-changing decisions," *Transp. Res. B, Methodol.*, vol. 20, no. 5, pp. 403–414, Oct. 1986.
- [9] Y. Hou, P. Edara, and C. Sun, "Modeling mandatory lane changing using bayes classifier and decision trees," *IEEE Trans. Intell. Transp. Syst.*, vol. 15, no. 2, pp. 647–655, Apr. 2014.
- [10] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling" in *Proc. INTERSPEECH 13th Annu. Conf. Int. Speech Commun. Assoc.*, Portland, OR, USA, Sep. 2012.
- [11] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system," in *Proc. SSW*, Sunnyvale, USA, 2016, pp. 1–6.
- [12] A.-L. Bianne-Bernard, F. Menasri, R. A.-H. Mohamad, C. Mokbel, C. Kermorvant, and L. Likforman-Sulem, "Dynamic and contextual information in HMM modeling for handwritten word recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 10, pp. 2066–2080, Oct. 2011.
- [13] T. Mokolov, M. Karafiát, L. Burget, J. Černocký, and S. Khudanpur, "Recurrent neural network based language model," in *Proc. 11th Annu. Conf. Int. Speech Commun. Assoc.*, 2010, pp. 1–24.
- [14] H. Chen, J. Chen, R. Hu, C. Chen, and Z. Wang, "Action recognition with temporal scale-invariant deep learning framework," *China Commun.*, vol. 14, no. 2, pp. 163–172, Feb. 2017.
- [15] M. Zhang, Y. Zhang, and D. T. Vo, "Gated neural networks for targeted sentiment analysis," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3087–3093.
- [16] Y. Kim, Y. Jernite, D. Sontag, and A. M. Rush, "Character-aware neural language models," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2741–2749.
- [17] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: <http://arxiv.org/abs/1412.3555>
- [18] N. Kalchbrenner and P. Blunsom, "Recurrent continuous translation models," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2013, pp. 1700–1709.
- [19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," 2014, *arXiv:1409.0473*. [Online]. Available: <http://arxiv.org/abs/1409.0473>
- [20] A. Graves, A.-R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, May 2013, pp. 6645–6649.
- [21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

- [22] H. Zen and H. Sak, "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Apr. 2015, pp. 4470–4474.
- [23] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. 13th AAAI Conf. Artif. Intell.*, vol. 16, 2016, pp. 3501–3508.
- [24] S. Pareek and T. Kesavadas, "iART: Learning from demonstration for assisted robotic therapy using LSTM," *IEEE Robot. Automat. Lett.*, vol. 5, no. 2, pp. 477–484, Apr. 2020, doi: [10.1109/LRA.2019.2961845](https://doi.org/10.1109/LRA.2019.2961845).
- [25] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [26] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: <https://ai.google>
- [27] J. Zhao, Y. Gao, Z. Bai, H. Wang, and S. Lu, "Traffic speed prediction under non-recurrent congestion: Based on LSTM method and BeiDou navigation satellite system data," *IEEE Intell. Transp. Syst. Mag.*, vol. 11, no. 2, pp. 70–81, Summer 2019.
- [28] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen," Netzen. Diploma thesis, Institut für Informatik, Lehrstuhl Prof. Brauer, Technische Universität München, Munich, Germany, 1991. [Online]. Available: <http://www7.informatik.tu-muenchen.de/Ehochreit>
- [29] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *Int. J. Uncertainty, Fuzziness Knowl.-Based Syst.*, vol. 6, no. 2, pp. 107–116, 1998.
- [30] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proc. 13th Int. Conf. Artif. Intell. Statist.*, 2010, pp. 249–256.
- [31] H. R. Maier and G. C. Dandy, "The effect of internal parameters and geometry on the performance of back-propagation neural networks: An empirical study," *Environ. Model. Softw.*, vol. 13, no. 2, pp. 193–209, Apr. 1998.
- [32] U. R. Babu, Y. Venkateswarlu, and A. K. Chintha, "Handwritten digit recognition using K-nearest neighbour classifier," in *Proc. World Congr. Comput. Commun. Technol.*, Trichirappalli, India, Feb./Mar. 2014, pp. 60–65.
- [33] S. Jain and R. Chauhan, "Recognition of handwritten digits using DNN, CNN, and RNN," in *Advances in Computing and Data Sciences (Communications in Computer and Information Science)*, vol. 905, M. Singh, P. Gupta, V. Tyagi, J. Flusser, and T. Ören Eds. Singapore: Springer, 2018.
- [34] Y. Chherawala, P. P. Roy, and M. Cherié, "Feature set evaluation for offline handwriting recognition systems: Application to the recurrent neural network model," *IEEE Trans. Cybern.*, vol. 46, no. 12, pp. 2825–2836, Dec. 2016.
- [35] A. Graves and J. Schmidhuber, "Frame-wise phoneme classification with bidirectional LSTM and other neural network architectures," *Neural Netw.*, vol. 18, nos. 5–6, pp. 602–610, Jul. 2005.
- [36] D. C. Cireşan, U. Meier, L. M. Gambardella, and J. Schmidhuber, "Deep, big, simple neural nets for handwritten digit recognition," *Neural Comput.*, vol. 22, no. 12, pp. 3207–3220, 2010.
- [37] A. Graves and J. Schmidhuber, "Offline handwriting recognition with multidimensional recurrent neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 21, 2008, pp. 545–552.
- [38] M. M. Abu Ghosh and A. Y. Maghari, "A comparative study on handwriting digit recognition using neural networks," in *Proc. Int. Conf. Promising Electron. Technol. (ICPET)*, Oct. 2017, pp. 77–81.
- [39] J. Yi, Z. Wen, J. Tao, H. Ni, and B. Liu, "CTC regularized model adaptation for improving LSTM RNN based multi-accent mandarin speech recognition," *J. Signal Process. Syst.*, vol. 90, no. 7, pp. 985–997, Jul. 2018.
- [40] W. Swaileh, Y. Soullard, and T. Paquet, "A unified multilingual handwriting recognition system using multigrams sub-lexical units," *Pattern Recognit. Lett.*, vol. 121, pp. 68–76, Apr. 2019.



HAIPENG XIAO is currently pursuing the degree with the Army Military Transportation University, China. His research interests include deep learning and trajectory prediction of unmanned vehicles.



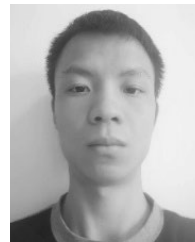
he was the Editor-in-Chief of the *IEEE Intelligent Transportation Systems Magazine*. From 2008 to 2015, he was an Associate Editor of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEM.

MIGUEL ANGEL SOTELO (Fellow, IEEE) received the Ph.D. degree in electrical engineering from the University of Alcalá (UAH), Alcalá de Henares, Madrid, Spain, in 2001. He is currently the Head and a Full Professor with the INVETT Research Group. He is also the Vice-President for International Relations with the University of Alcalá. Since January 2017, he has been the President-Elect of the IEEE Intelligent Transportation Systems Society. From 2014 to 2016,



include intelligent vehicles and intelligent transportation systems. He is also a Reviewer of the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and the *International Journal of Intelligent Transportation Systems Research*.

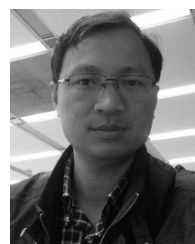
YULIN MA (Member, IEEE) received the Ph.D. degree in transportation engineering from the Wuhan University of Technology, China. He did his Postdoctoral Research at the Academy of Military Transportation, China. He joined the Research Institute of Highway, Ministry of Transport, where he was an Associate Professor. He is currently the Director of the Institute of Automated Vehicle Test, Suzhou Automotive Research Institute, Tsinghua University, China. His research interests



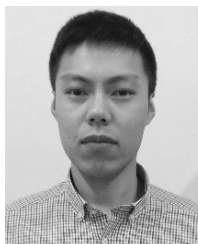
BO CAO is currently a Research Assistant with the Department Vehicle Engineering, Army Military Transportation University, China. His research interests include the construction and application of HD Map and motion planning of intelligent vehicle. From 2017 to 2019, his research was applied to the 2nd World Intelligent Driving Challenge, Crossing 2018 land Unmanned System Challenge, the 2018 and 2019 Intelligent Vehicle Future Challenge (IVFC), and committed to award five first prize for his team.



YUNCHENG ZHOU received the master's degree in measurement techniques and measuring instruments from the Nanjing University of Posts and Telecommunications, Nanjing, China. He is currently a Senior Staff Engineer/Laboratory Manager with the ITS Research Center, China Design Group Company Ltd., Nanjing. His research interests include intelligent transportation and intelligent parking systems.



YOUCHUN XU received the Ph.D. degree in vehicle application engineering from Jilin University, China, in 2001. From 2002 to 2004, he was a Postdoctoral Researcher with Tsinghua University, China. Since 2015, he has been a Visiting Scholar with UC Berkeley, USA. He is currently a Professor with the Institute of Military Transportation, Army Military Transportation University, China. His current research interests include key technologies of autonomous vehicle, computer vision, and deep learning.



RENDONG WANG received the Ph.D. degree in mechanical engineering from Army Military Transportation University, China, in 2019. He is currently an Engineer with the Institute of Military Transportation, Army Military Transportation University. His research interests include scene understanding and localization and mapping using multi-sensor fusion for autonomous vehicle.



ZHIXIONG LI (Senior Member, IEEE) received the Ph.D. degree in transportation engineering from the Wuhan University of Technology, in 2013, China. He is currently an ARC DECRA Fellow with the School of Mechanical, Materials, Mechatronic and Biomedical Engineering, University of Wollongong, Wollongong, NSW, Australia. He is also an Adjunct Professor with the Ocean University of China. His research interests include mechanical system modeling and control. He is an

Associate Editor of *Measurement* (Elsevier) and a Column Editor of the *IEEE Intelligent Transportation Systems Magazine*.

• • •