

Research on Traffic Flow Prediction Based on ISMA-CNN-GRU Model

Huihui Tian¹, Jun Su¹ and Orest Kochan^{1,2}

¹ Hubei University of Technology, No.28 Street NanLi, Wuhan, 430068, Country

² Lviv Polytechnic National University, 12 S. Bandera Str., Lviv, 79013, Ukraine,

Abstract

In recent years, how to forecast traffic flow quickly and accurately has become a key issue in building an intelligent transportation system. Due to the temporal and spatial correlation of traffic flow data, we propose a prediction model combining convolutional neural network (CNN), gated recurrent unit (GRU) and improved slime mould algorithm (ISMA). The basic idea is to construct the traffic flow data as a two-dimensional matrix containing temporal and spatial information, and use CNN to obtain location-related spatial features and use GRU's memory function to obtain the temporal distribution features. Secondly, for the shortcomings of the slime mould algorithm with low initial population quality, this paper adds Tent chaotic mapping and adaptive inertia weighting strategy to obtain the ISMA algorithm and uses it to find the optimal combination of hyperparameters for the GRU network to construct the ISMA-CNN-GRU prediction model. Finally, simulation experiments are conducted on the traffic flow dataset of Heathrow Airport, UK. The experiments confirm that the ISMA-CNN-GRU exhibits higher prediction accuracy compared to the APSO-GRU model, the unoptimized CNN-GRU model and the SMA-CNN-GRU model.

Keywords

Adaptive Inertia Weights, Convolutional Neural Network, Chaotic Mapping, GRU Neural Network, Slime Mould Algorithm

1. Introduction

In recent years, the phenomenon of vehicle congestion has been occurring more and more frequently, which not only brings trouble to people's travel [1], but also increases the workload of traffic police departments. Intelligent transportation system came into being, and how to predict urban traffic flow quickly and accurately has become a key issue in building an intelligent transportation system. Most of the traditional traffic flow prediction models are highly dependent on experienced experts, lack of independent learning ability and low prediction accuracy, so they have been gradually eliminated. Neural networks are widely used in traffic flow prediction because of their high operational efficiency and independent learning ability [2]. Therefore, this paper starts from the neural network to improve the traffic prediction accuracy as the goal of research.

With the emergence of intelligent optimization algorithms, more and more scholars have improved the prediction ability of network models by combining algorithms with neural networks. In the literature [3], a based LSTM-RF traffic flow prediction model is proposed, which uses Long Short-Term Memory (LSTM) to obtain the temporal characteristics of the target road and combines them with the upstream and downstream data of neighboring road sections to be incorporated into a random forest model to predict traffic flow. The superior memory function of LSTM network solves the problem that Recurrent Neural Network (RNN) is prone to gradient disappearance and gradient explosion, but the training speed is still slow. The literature [4-7] compared GRU with other

COLINS-2023: 7th International Conference on Computational Linguistics and Intelligent Systems, April 20–21, 2023, Kharkiv, Ukraine

EMAIL: tianhuihui@hbput.edu.cn (H. Tian); sujuncs@hbput.edu.cn (J. Su); orest.v.kochan@lpnu.ua (O. Kochan)

ORCID: 0009-0007-2009-4680 (H. Tian); 0000-0002-4290-5049 (J. Su); 0000-0002-3164-3821 (O. Kochan)



© 2023 Copyright for this paper by its authors.
Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

prediction models, such as RNN, LSTM and Auto Regressive Integrated Moving Average (ARIMA) statistical technology models, where GRU showed superior prediction results. In the literature [8], the adaptive nonlinear inertial weight particle swarm optimization (APSO) algorithm was proposed, and the APSO-GRU model shows significant stability.

2. Related work

2.1. GRU Neural Network

To balance the input and forgetting gates in LSTM [9], GRU improves on the LSTM network by adding update gates z to provide memory of past history information [10]. This not only optimizes its internal structure, but also increases its learning speed. The GRU model is defined in the way shown in Equations (1)-(6) [11]. The inputs to the update gate z_t and reset gate r_t in GRU are x_t, h_{t-1} , where x_t is the t^{th} component of the input sequence x and h_{t-1} is the hidden state of the previous time step [12]; W_z, U_z, W_r, U_r, W_h , and U_h are the weight matrices [13]; b_z, b_r and b_h are the bias matrices. σ is the Sigmoid function. $*$ denotes the corresponding elements in the matrix are multiplied. The GRU always calculates a candidate state before the current hidden state. Finally, the network calculates the final state h_t at the current moment.

$$z_t = \sigma(U_z x_t + W_z h_{t-1} + b_z) \quad (1)$$

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r) \quad (2)$$

$$\tilde{h}_t = \tanh(U_h x_t + W_h (r_t * h_{t-1}) + b_h) \quad (3)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t \quad (4)$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (5)$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (6)$$

2.2. Convolutional Neural Network

The traffic flow of a site is not only related to the past traffic flow of that site, but also influenced by the traffic flow of other sites, that is, by the spatial characteristics. CNNs use both local connectivity and shared weights to identify spatially relevant features, which can effectively reduce the complexity and errors such as over-fitting during data processing [14]. Therefore, this paper uses CNN for spatial feature mining of traffic flow. As shown in Figure 1. Its structure consists of an input layer, a convolutional layer, a pooling layer, and a fully connected layer [15]. The convolution layer mainly performs convolution operations on the input data to extract local feature information and obtain the feature matrix. After feature extraction in the convolutional layer, the number of features is not reduced, so it needs to go through the pooling layer to reduce the amount of data processing, and avoid too many parameters in the fully connected layer.

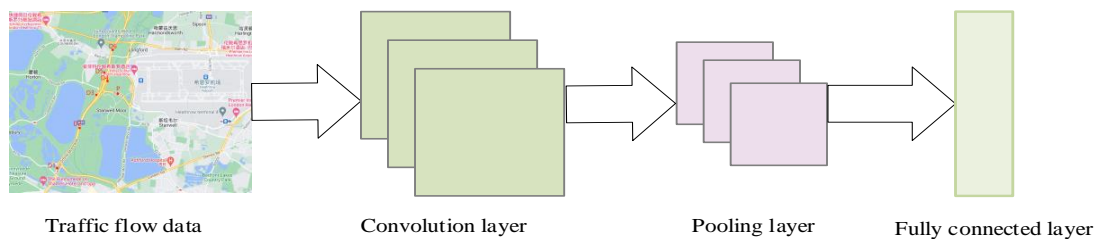


Figure 1: Convolutional neural network flow chart

2.3. Slime Mould Algorithm

In 2020, Li et al [16] proposed the Slime Mould Algorithm (SMA) based on the oscillatory foraging activity of slime mould. The higher the concentration of food odor in the air, resulting in the accumulation of more slime mould in that domain; when the concentration of food in that range is low, slime mould will be lured to forage in other domains. When the slime mould finds a superior quality food source, they will also separate some individuals to explore better food in other areas.

The equation of slime mould population renewal location is shown in (7). t is the number of iterations; $x_b(t)$ is the current best position; $x_A(t)$ and $x_B(t)$ are the positions of two arbitrarily selected individuals. w is the slime mould quality, which denotes the fitness weight. v_b and v_c are the control parameters; v_b lies within the range $[-a, a]$; r and $rand$ are random numbers between $[0,1]$; ub and lb represent the upper and the lower bounds of the exploration domain, respectively; and z is a custom parameter.

$$x(t+1) = \begin{cases} rand \cdot (ub - lb) + lb, rand < z \\ x_b(t) + v_b \cdot (w \cdot x_A(t) - x_B(t)), r < p \\ v_c \cdot x(t), r \geq p \end{cases} \quad (7)$$

The mathematical model to describe the control variable p and parameter a is shown in Equations (8) and (9). $i \in 1, 2, 3, \dots, n$; $s(i)$ is the current individual fitness value; DF is the current best fitness value.

$$p = \tanh |s(i) - DF| \quad (8)$$

$$a = \arctan h \left(1 - \frac{t}{t_{\max}} \right) \quad (9)$$

The weight parameter w is described as follows in Equation (10). b_f denotes the best fitness of the current iteration; w_f denotes the worst; *condition* denotes the top half of individuals in terms of fitness; *else* denotes the remaining individuals; and $S_{index}(i)$ is the fitness ranking and denotes the odor index.

$$w(S_{index}(i)) = \begin{cases} 1 + r \cdot \log \left(\frac{b_f - s(i)}{b_f - w_f} + 1 \right), \text{condition} \\ 1 - r \cdot \log \left(\frac{b_f - s(i)}{b_f - w_f} + 1 \right), \text{else} \\ S_{index}(i) = \text{sort}(s) \end{cases} \quad (10)$$

2.4. Proposed methodology

2.4.1. Improved Slime Mould Algorithm

It was found that the synergistic effect of v_b and v_c causes the slime mould to not only shrink toward the optimal position, but also separate a part of organic matter to explore other fields, and the oscillatory effect of v_b increases the possibility of global exploration [17]. But the oscillatory effect of v_b weakens significantly in the late iterations, which leads to the fact that the algorithm falls into a local optimum easily [18]. Therefore, the improved slime mould algorithm (ISMA) which improved by Tent chaotic mapping and the adaptive inertia weighting strategy makes the search of slime mould more effective.

Among the many chaotic mapping functions, Tent chaotic mapping is chosen in this paper because of its uniform distribution of chaotic sequences, better traversal uniformity and higher iteration speed

[19]. The expressions are shown in (11) and (12). i denotes the population size; j denotes the chaotic sequence number; $Y_{i,j}$ is the chaotic sequence of [0,1]; the initial position of the population is obtained using the $X_{i,j}$ inverse mapping; and $[lb, ub]$ denotes the search range of the individual position $X_{i,j}$.

$$Y_{i,j+1} = \begin{cases} \frac{Y_{i,j}}{0.7}, Y_{i,j} < 0.7 \\ \frac{1-Y_{i,j}}{0.3}, Y_{i,j} \geq 0.7 \end{cases} \quad (11)$$

$$X_{i,j} = Y_{i,j} \cdot (ub - lb) + lb \quad (12)$$

It is found that at the beginning of each iteration the slime mould is prone to a large forward state, which leads to a lack of search range and makes the algorithm easily fall into local extrema. We propose the adaptive inertia weight strategy as shown in (13)(14) to help the slime mould jump out of the local extrema. w_1 and w_2 are weight adjustment parameters. The pseudo-code for ISMA is shown in Table 1.

$$x(t+1) = \begin{cases} rand \cdot (ub - lb) + lb, rand < z \\ x_b(t) + m \cdot v_b \cdot (w \cdot x_A(t) - x_B(t)), r < p \\ m \cdot v_c \cdot x(t), r \geq p \end{cases} \quad (13)$$

$$m = w_1 + (w_1 - w_2) \left(\frac{t}{t_{max}} \right)^2 + 2(w_1 - w_2) \left(\frac{t}{t_{max}} \right) \quad (14)$$

Table 1

The pseudo-code for ISMA

Algorithm ISMA
Initialize: Number of Population $popsiz$, maximum number of iterations $maxgen$, dimension of Population dim , objective fitness function $f(x)$;
Tent Chaotic Mapping initialize positions of slime mould x_i ;
Counter number of iterations $t=0$
While $t \leq maxgen$ do
Calculate the fitness of all slime mould;
Update the best fitness x_b
Calculate the w by Equation (10);
For each individual
Update parameters p, v_b, v_c ;
Update position by Equation (13)(14);
End For
$t=t+1$;
End While
Return the best fitness x_b after iteration;

2.4.2. ISMA-CNN-GRU Prediction Model

We propose to combine CNN and GRU to form a model. CNN mines spatial association features by using local connectivity when processing time series data, while GRU can extract deep features before and after time series data; then the ISMA algorithm is used to solve for the best combination of hyperparameters in the network, and an ISMA-CNN-GRU prediction model is established. The input to the model will be a two-dimensional spatio-temporal matrix constructed using the traffic flow by time as the horizontal coordinate of the matrix and the site ID as the vertical coordinate of the matrix. The spatio-temporal matrix is equation (15). a_i, b_i, m_i denote the traffic flow at different sites.

$$H = \begin{bmatrix} a_1 & a_2 & \dots & a_n \\ b_1 & b_2 & \dots & b_n \\ \dots & \dots & \dots & \dots \\ m_1 & m_2 & \dots & m_n \end{bmatrix} \quad (15)$$

The ISMA-CNN-GRU model framework is illustrated in Figure 2.

First, the traffic flow is preprocessed and transformed into a two-dimensional spatial and temporal matrix, and the training set is fed into the convolutional neural network to extract the spatial distribution features of the traffic flow.

After the convolutional operation, the data are transferred to the pooling layer, which reduces the dimensionality of the model to improve the computational speed of the model. The final data are convolved by the convolutional neural network to obtain data samples with spatial characteristics.

The local features extracted by the convolutional neural network are then used as input to the GRU to further extract the temporal features of traffic flow.

Finally, the predicted results are output through the fully connected layer, which is the output layer.

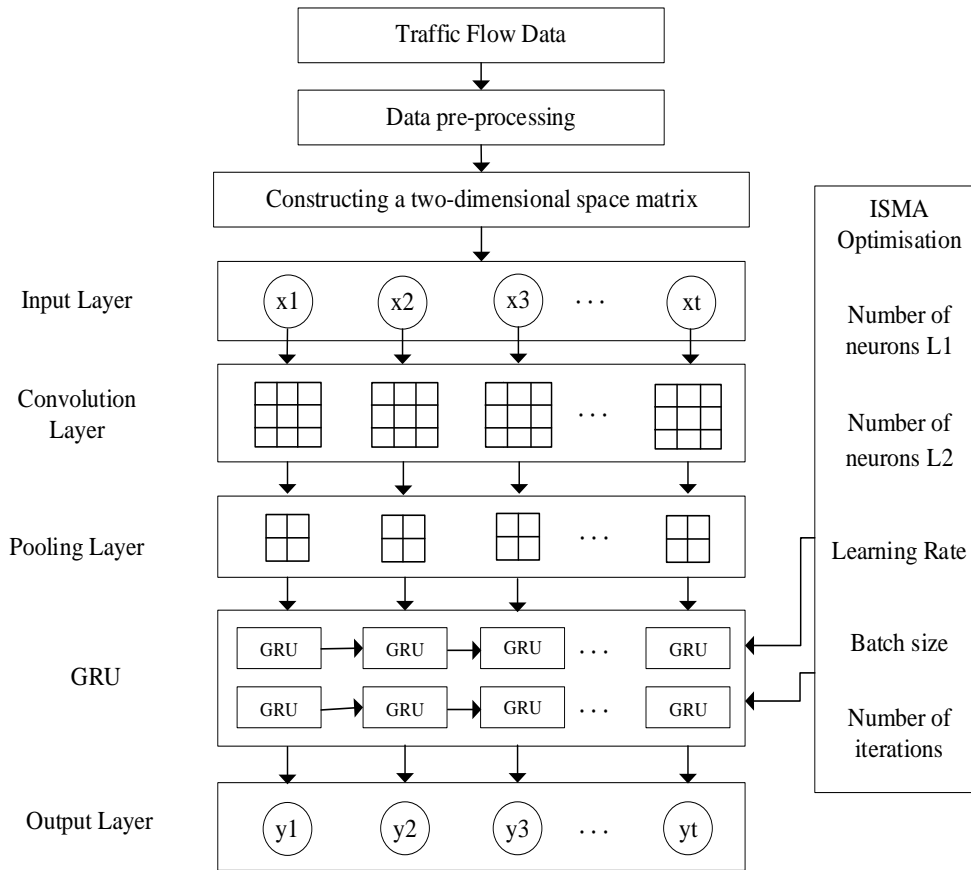


Figure 2: ISMA-CNN-GRU model framework diagram

The specific steps are as follows.

Step 1: pre-processing of traffic flow data and partitioning of the data set.

Step 2: initialize the structure of the CNN-GRU neural network and initialize the basic parameters.

Step 3: initialize the slime by the Tent chaos mapping.

Step 4: construct a spatio-temporal matrix of the traffic flow based on site and time traffic flows.

Step 5: input the matrix data into the CNN to extract the spatial features of the traffic flow.

Step 6: input the CNN-processed data to the GRU layer.

Step 7: calculate the fitness of the slime mould using the fitness function, retain the best value.

Step 8: calculate the weight parameter w according to equation (10) and generate adaptive inertia weight factors by equation (13) (14) to update the position of the slime mould.

Step 9: calculate and re-ranking the slime adaptation, update the best position and the fitness value.

Step 10: determine whether the algorithm has reached the maximum number of iterations, and if so, end and output the optimal solution, otherwise return to step 6 to continue execution.

Step 11: the hyperparameters optimized by the slime mould algorithm are used as parameters for the CNN-GRU network, and the optimized model is used to train and predict traffic flow.

We choose the convolutional kernel size of 3×3 , the number of convolutional layers is 1, the number of convolutional kernels is 32, and the number of pooling layers is 1, using mean pooling. The GRU contains two layers. The activation function is the Relu. The loss function is MSE. Setting the slime population size to 30 and the maximum number of iterations to 10. The adaptive inertia weights $w_1=0.9$, $w_2=0.5$. Using MSE as the fitness function, the formula is shown in (16), where n is the number of predicted samples, Y_i is the sample output value and y_i is the actual output value.

$$f(i) = MSE = \sum_{i=1}^n \frac{(Y_i - y_i)^2}{n} \quad (16)$$

3. ISMA-CNN-GRU Based Traffic Flow Prediction

3.1. Experiment Dataset

The experimental running environment in this section is a 64-bit Windows 10 operating system with an Intel Core i5-7200U CPU @ 2.50GHz and a host installation with 4+8G of RAM. The simulation software used for testing is MATLAB version R2021b. The traffic flow data was taken from the British Motorways Dataset website. As shown in Figure 3, the selected dataset contains information from seven sites on the M25 freeway near Heathrow Airport, UK. The seven sites in the dataset are used as the vertical coordinates of the spatio-temporal matrix, and the horizontal coordinates are the data of each site divided by 15 minutes. In order to observe the prediction effect of the ISMA-CNN-GRU model more intuitively, this subsection takes the flow of site P as an example for prediction. The traffic volumes from August 1 to 25 in summer and September 1 to 25 in autumn were set as the training set, and August 25 to 30 and September 25 to 30 were set as the test set.



Figure 3: 7 sites on the M25 near Heathrow Airport, UK

3.2. Data Pre-processing

In order to facilitate data processing, the commonly used normalization method was chosen to normalize the historical data before training the network, scaling the data to the range of $[0,1]$. The calculation formula is shown in (17), where x is the original data; x' is the normalized data; x_{max} is the maximum values of the input quantities; and x_{min} is the minimum.

$$x' = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \quad (17)$$

3.3. Prediction Evaluation Indicators

In order to facilitate us to compare the prediction accuracy of each model more clearly, we choose the commonly used error evaluation metrics of mean absolute error (MAE) [20, 21], root mean square error (RMSE) [22], mean absolute percentage error (MAPE) [23] and coefficient of determination (R^2) [24, 25] to evaluate the prediction effectiveness and goodness of fit of the models. The formulas are shown in (18)-(21). n indicates the number of predicted data; \hat{x}_i indicates the predicted data of the i^{th} ; and x_i indicates the true data of the corresponding.

$$MAE = \frac{1}{n} \sum_{i=1}^n |x_i - \hat{x}_i| \quad (18)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2} \quad (19)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{x_i - \hat{x}_i}{\hat{x}_i} \right| \times 100\% \quad (20)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{x}_i - x_i)^2}{\sum_{i=1}^n (\bar{x}_i - x_i)^2} \quad (21)$$

The metrics used to evaluate the prediction intervals are selected as the prediction intervals coverage probability (PICP) [26], prediction interval normalized average width (PINAW) [27] and coverage width criterion (CWC) [28]. n denotes the total number of samples; c_i is the reliability metric of the upper interval U_i and the lower interval L_i against the true value \hat{y}_i ; μ is the confidence level and η is the penalty parameter [29]. Here $\mu = 0.95$, $\eta = 20$. The formulas are shown in (22)-(26).

$$PICP = \frac{1}{n} \sum_{i=1}^n c_i \times 100\% \quad (22)$$

$$c_i = \begin{cases} 1, & \hat{y}_i \in [L_i, U_i] \\ 0, & \hat{y}_i \notin [L_i, U_i] \end{cases} \quad (23)$$

$$PINAW = \frac{1}{n} \sum_{i=1}^n \frac{U_i - L_i}{\hat{y}_i} \times 100\% \quad (24)$$

$$CWC = PINAW (1 + \gamma(PICP) e^{-\eta(PICP - \mu)}) \quad (25)$$

$$\gamma = \begin{cases} 0, & PICP \geq \mu \\ 1, & PICP < \mu \end{cases} \quad (26)$$

3.4. Prediction Results of ISMA-CNN-GRU

ISMA searches for hyperparameter combinations of CNN-GRU networks, using MSE as the judging criterion. The trend of the fitness value of the ISMA iteration process is shown in Figure 4. The fitness value decreases rapidly with the increasing number of iterations and then remains stable, and reaches a steady state with the minimum error at the maximum number of iterations.

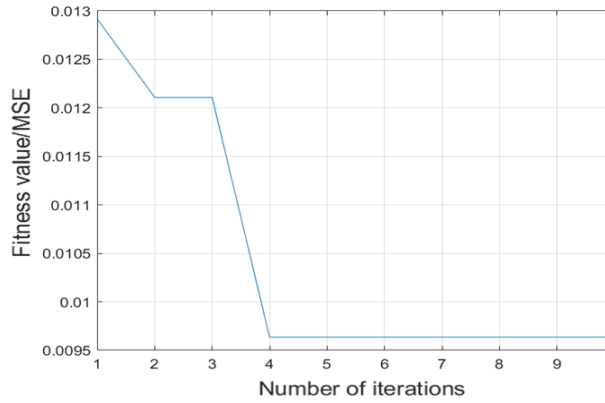


Figure 4: The fitness curve of ISMA

The variation of RMSE of the ISMA-CNN-GRU model, is shown in Figure 5. It can be seen that the RMSE metric of the model decreases rapidly in the first 15 iterations of training. When the number of training iterations reached 48, the model was basically stable. Finally, the number of neurons in the first hidden layer is $L1=88$, the number of neurons in the second hidden layer is $L2=78$, the batch size is $\text{Batchsize}=27$, the learning rate is $\text{lr}=0.0055$, and the number of iterations is $K=48$.

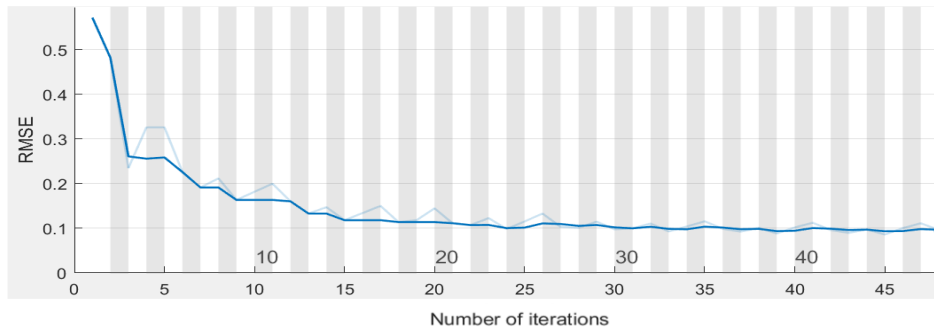


Figure 5: Variation of RMSE of ISMA-CNN-GRU model with the number of iterations

In order to observe the prediction effect of the ISMA-CNN-GRU model more intuitively, this subsection takes the flow of site P as an example for prediction, with a sample size of 960 for the traffic flow from the test set. This experiment compares the prediction results of the ISMA-CNN-GRU model with the APSO-GRU [8], the CNN-GRU and SMA-CNN-GRU models, as shown in Figure 6. The ISMA-CNN-GRU model and SMA-CNN-GRU model perform better than the CNN-GRU model and the APSO-GRU model in prediction performance, and the ISMA-CNN-GRU model can fit the true value curve better. It can be seen that although the prediction results do not show a significant difference between summer and autumn, and the seasonality is not obvious, the ISMA-CNN-GRU still shows high prediction accuracy.

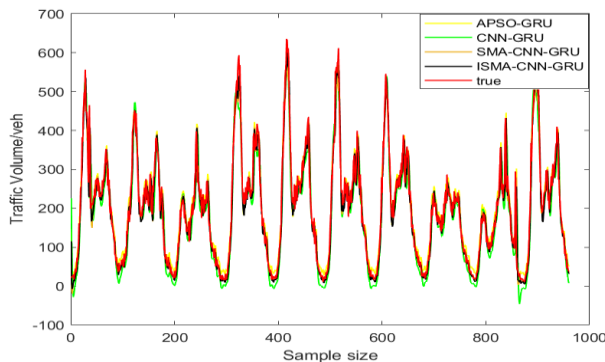


Figure 6: Comparison curves of prediction results of four models on P-site

The comparison results based on the error estimation evaluation metrics are shown in Table 2. The prediction accuracy of the ISMA-CNN-GRU model is 98.4787%, that of the SMA-CNN-GRU model is 98.16326%, and that of the CNN-GRU model is 97.6305%. The ISMA-CNN-GRU model is higher than the SMA-CNN-GRU and CNN-GRU by 0.3161% and 0.8482%, respectively. The prediction error of the APSO-GRU model is relatively large and the fit is poor. In addition, the highest R^2 of ISMA-CNN-GRU on this dataset is 0.9661, which is closer to 1 compared with several other models, which reflects that this model can learn the changing pattern of traffic flow well and has high prediction accuracy and precision, and verifies its feasibility in traffic flow prediction.

Table 2

Comparison of prediction error evaluation indicators results of four models

Predictive Models	Error Indicators			
	MAE	RMSE	MAPE%	R^2
APSO-GRU	0.1371	0.1832	3.985	0.9346
CNN-GRU	0.12496	0.15886	2.3695	0.95094
SMA-CNN-GRU	0.11807	0.13735	1.83674	0.9628
ISMA-CNN-GRU	0.11598	0.12802	1.5213	0.9661

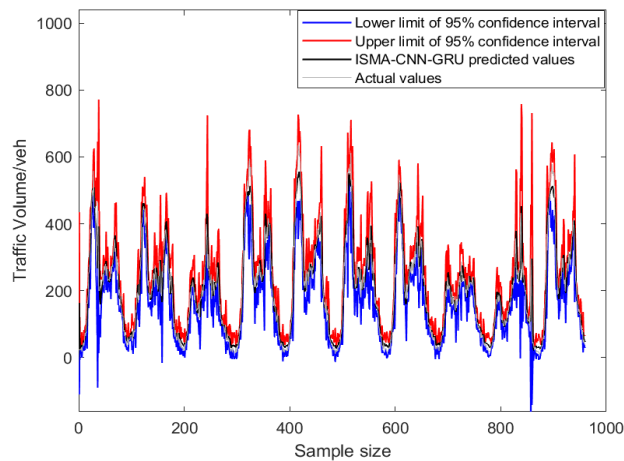


Figure 7: The prediction results of the ISMA-CNN-GRU model at the 95% confidence interval

The experiment was also done based on the interval prediction evaluation index. Figure 7 shows the prediction results of the ISMA-CNN-GRU model at 95% confidence interval. As can be seen from Table 3, the true values of all models fall within the interval when the confidence level is 95%. Because $PICP=1$ indicates that the true value falls within the constructed interval. The minimum value of CWC for the interval composite evaluation index of ISMA-CNN-GRU model is 0.628. Compared with APSO-GRU, CNN-GRU and SMA-CNN-GRU models, both PINAW and CWC of ISMA-CNN-GRU model have been reduced, and these two indicators fully illustrate the superiority of the ISMA-CNN-GRU model, because when the $PICP$ is determined, the smaller the PINAW and the smaller the value of CWC, the narrower the prediction interval, indicating better prediction.

Table 3

Comparison of the prediction results of the four models at 95% confidence intervals

Predictive Models	Interval Forecast Indicators		
	$PICP$	PINAW	CWC
APSO-GRU	1	1.3661	1.3661
CNN-GRU	1	1.249	1.249
SMA-CNN-GRU	1	0.936	0.936
ISMA-CNN-GRU	1	0.628	0.628

4. Conclusion

In recent years, how to forecast traffic flow quickly and accurately has become a key issue in building an intelligent transportation system. We adopt a combination of CNN and GRU network and ISMA algorithm to form a model. CNN is used to obtain the traffic flow distribution characteristics between sites, and then the memory functions of GRU are used to obtain the temporal distribution characteristics of traffic flow. Secondly, for the shortcomings of the slime mould algorithm with low initial population quality, this paper adopts the Tent chaos mapping and adaptive inertia weighting strategy to improve the slime mould algorithm. The hyperparameters of the GRU model are optimized using the proposed ISMA algorithm. The simulation results show that the ISMA-CNN-GRU model exhibits higher prediction accuracy compared with the APSO-GRU, the unoptimized CNN-GRU and the SMA-CNN-GRU models. The accurate prediction of traffic flow by the model in this paper contributes to the operational capacity of the transportation system as well as its operational efficiency, and is of practical value to citizens, traffic management, road operations and infrastructure participants. Future research will be extended in the following. The computational efficiency of the ISMA algorithm in training out the optimal hyperparameters need to be improved. The proposed ISMA-CNN-GRU model only selects the optimal parameter combinations, future research can design controlled experiments to study the degree of influence of each parameter on the overall model performance.

5. References

- [1] Dasgupta, Susmita, Somik Lall, and David Wheeler. "Spatiotemporal analysis of traffic congestion, air pollution, and exposure vulnerability in Tanzania." *Science of The Total Environment* 778 (2021): 147114.
- [2] B. L. Smith, M. J. Demetsky, "Short-Term Traffic Flow Prediction: Neural Network Approach" *Transportation Research Record* 1453 (1994): 98-104.
- [3] Z. Shu-xu, Z. Bao-hua, "Traffic flow prediction of urban road network based on LSTM-RF model." *Journal of Measurement Science & Instrumentation Journal of Measurement Science & Instrumentation* 11 (2020) 135-142.
- [4] N. Buslim, I. L. Rahmatullah, B. A. Setyawan, A. Alamsyah, Comparing bitcoin's prediction model using GRU, RNN, and LSTM by hyperparameter optimization grid search and random search, in: *Proceedings of 2021 9th International Conference on Cyber and IT Service Management, CITSM 2021*, pp. 1-6.
- [5] V. Kumar, V. M. Nookesh, B. S. Saketh, S. Syama, J. Ramprabhakar, Wind speed prediction using deep learning-LSTM and GRU, in: *Proceedings of 2021 2nd International Conference on Smart Electronics and Communication ICOSEC 2021*, pp. 602-607.
- [6] B. Hussain, M. K. Afzal, S. Ahmad, A. M. Mostafa, Intelligent traffic flow prediction using optimized GRU model, *IEEE Access* 9 (2021) 100736-100746.
- [7] K. E. ArunKumar, D.V. Kalaga, C. M. S. Kumar, M. Kawaji, T. M. Brenza, Comparative analysis of gated recurrent units (GRU), long short-term memory (LSTM) cells, autoregressive integrated moving average (ARIMA), seasonal autoregressive integrated moving average (SARIMA) for forecasting COVID-19 trends, *Alexandria engineering journal* 61 (2022) 7585-7603.
- [8] D. Han, X. Yang, G. Li, S. Wang, Z. Wang, J. Zhao, Highway traffic speed prediction in rainy environment based on APSO-GRU, *Journal of advanced transportation* 2021 (2021) 1-11.
- [9] L. Sun, H. Qin, K. Przystupa, M. Majka, O. Kochan, Individualized Short-Term Electric Load Forecasting Using Data-Driven Meta-Heuristic Method Based on LSTM Network, *Sensors*, 22 (2022) 7900. doi: 10.3390/s22207900
- [10] D. Bahdanau, K. Cho, Neural Machine Translation by Jointly Learning to Align and Translate, *Computer Science* 69 (2014) 2437-2448.
- [11] S. M. Abdullah, M. Periyasamy, N. A. Kamaludeen, Optimizing traffic flow in smart cities: soft GRU-based recurrent neural networks for enhanced congestion prediction using deep learning, *Sustainability* 15 (2023). doi: 10.3390/su15075949.

- [12] U. Gupta, V. Bhattacharjee, P. S. Bishnu, StockNet—GRU based stock index prediction, *Expert Systems with Applications* 207 (2022). 117986. doi: 10.1016/j.eswa.2022.117986.
- [13] L. Munkhdalai, M. Li, N. Theera-Umpon, S. Auephanwiriyakul, K. HoRyu, VAR-GRU: A hybrid model for multivariate financial time series prediction, in: *Proceedings of 2020 12th Asian Conference on Intelligent Information and Database Systems, ACIIDS 2020*, pp. 322-332.
- [14] S. Ghimire, Z. M. Yaseen, A. A. Farooque, R. C. Deo, J. Zhang, X. Tao, Streamflow prediction using an integrated methodology based on convolutional neural network and long short-term memory networks, *Scientific Reports* 11 (2021). doi: 10.1038/s41598-021-96751-4.
- [15] Z. Li, F. Liu, W. Yang, S. Peng, J. Zhou, A survey of convolutional neural networks: analysis, applications, and prospects, *IEEE transactions on neural networks and learning systems* 3 (2021) 6999-7019. doi: 10.1109/TNNLS.2021.3084827.
- [16] .S. Li, H. Chen, M. Wang, A. A. Heidari, S. Mirjalili, Slime mould algorithm: A new method for stochastic optimization, *Future Generation Computer Systems* 111 (2020) 300-323. doi: 10.1016/j.future.2020.03.055.
- [17] E. H. Houssein, M. A. Mahdy, M. J. Blondin, D. Shebl, W. M. Mohamed, Hybrid slime mould algorithm with adaptive guided differential evolution algorithm for combinatorial and global optimization problems, *Expert Systems with Applications* 174 (2021): 114689. <https://doi.org/10.1016/j.eswa.2021.114689>.
- [18] A. A. Ewees, L. Abualigah, D. Yousri, Z. Y. Algamal, M. A. Al-Qaness, R. A. Ibrahim, M. Abd Elaziz, Improved slime mould algorithm based on firefly algorithm for feature selection: a case study on QSAR model, *Engineering with Computers* 38, (2021) 2407–2421. doi: 10.1007/s00366-021-01342-6.
- [19] A. M. Ahmed, T. A. Rashid, S. A. M. Saeed, Cat swarm optimization algorithm: a survey and performance evaluation, *Computational Intelligence and Neuroscience* 2020 (2020) 4854895. doi: 10.1155/2020/4854895.
- [20] J. Chen, J. Su, O. Kochan, M. Levkiv, Metrological software test for simulating the method of determining the thermocouple error in situ during operation, *Measurement Science Review*, 18 (2018) 52–58. doi: 10.1515/msr-2018-0008
- [21] B. Medina-Salgado, E. Sanchez-DelaCruz, P. Pozos-Parra, J. E. Sierra, Urban traffic flow prediction techniques: a review, *Sustainable Computing: Informatics and Systems* 35 (2022) 100739. doi: 10.1016/j.suscom.2022.100739.
- [22] J. Becerra-Rico, M. A. Aceves-Fernández, K. Esquivel-Escalante, J. C. Pedraza-Ortega, Airborne particle pollution predictive model using Gated Recurrent Unit (GRU) deep neural networks, *Earth Science Informatics*, 13 (2020) 821-834. doi: 10.1007/s12145-020-00462-9.
- [23] Navarro-Espinoza, Alfonso, et al. "Traffic flow prediction for smart traffic lights using machine learning algorithms." *Technologies* 10.1 (2022): 5.
- [24] M. Sabri, M. E. Hassouni, A Novel deep learning approach for short term photovoltaic power forecasting based on GRU-CNN model, in: *Proceedings of 2022 International Conference on Energy and Green Computing*, 336 (2022): 00064.
- [25] V. Yeromenko, O. Kochan, The conditional least squares method for thermocouples error modeling, in: *Proceedings of the 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems, IDAACS 2013*, vol. 1, pp. 157–162. doi: 10.1109/IDAACS.2013.6662661
- [26] B. Bommidi, V. Kosana, K. Teeparthi, S. Madasthu, Hybrid attention-based temporal convolutional bidirectional LSTM approach for wind speed interval prediction, *Environ Sci Pollut Res* 30, (2023) 40018–40030. doi: 10.1007/s11356-022-24641-x.
- [27] X. Serrano-Guerrero, M. Briceño-León, J. M. Clairand, G. Escrivá-Escrivá, A new interval prediction methodology for short-term electric load forecasting based on pattern recognition, *Applied Energy* 297 (2021) 117173. doi: 10.1016/j.apenergy.2021.117173.
- [28] A. Saeed, C. Li, M. Danish, S. Rubaiee, G. Tang, et al. Hybrid Bidirectional LSTM Model for Short-Term Wind Speed Interval Prediction, *IEEE Access* 9 (2020) 182283-182294. doi: 10.1109/ACCESS.2020.3027977.
- A. Banik, C. Behera, T. V. Sarathkumar, A. K. Goswami, Uncertain wind power forecasting using LSTM-based prediction interval, *IET Renewable Power Generation* 14 (2020) 2657-2667. doi: 10.1049/iet-rpg.2019.1238.