Traffic Speed Prediction with Congestion Data: An ISGA-Optimized Hybrid Deep Learning Model

Changxi Ma, Xiaoyu Huang and Bo Du

Abstract—**Accurately predicting traffic speed can not only optimize travel route guidance and alleviate congestion but also significantly improve the overall safety of urban road traffic systems. To enhance the prediction accuracy of traffic speed, we propose a hybrid model integrating a multi-head self-attention (MHSA) mechanism, a 1D convolutional network (Conv1D), and a gated recurrent unit (GRU). The improved snow goose algorithm (ISGA) is employed to optimize its hyperparameters, constructing the ISGA-MHSA-Conv1D-GRU model. This model exhibits strong local feature extraction and long-term temporal dependency learning capabilities, while leveraging ISGA's dual advantages of accelerated convergence and superior global search capacity, which provide robustness against hyperparameter sensitivity. Two real-world traffic datasets are selected for analysis, incorporating traffic congestion as an input feature during model training. Comparative experiments were conducted against multiple baseline models, including LSTM, GRU, Conv1D, GRU-LSTM, Conv1D-GRU, MHSA-Conv1D-GRU, and SGA-MHSA-Conv1D-GRU. Numerical results demonstrate that the ISGA-MHSA-Conv1D-GRU model outperforms all the baseline models according to the prediction accuracy. Specifically, compared to the baseline models, it reduces MSE by 26.5%–93.9%, decreases MAE by 26.5%–80.1%, and yields an R^2 closer to 1. The promising results indicate that the proposed model excels in both prediction accuracy and metric stability for traffic speed forecasting.**

Index Terms—**traffic speed, prediction accuracy, ISGA, MHSA- Conv1D-GRU, traffic congestion**

I. Introduction

TRAFFIC congestion has become a critical bottleneck restricting urban development, making it imperative to vigorously develop smart transportation and congestion control technologies. Traffic speed prediction on urban roads serves not only as a fundamental component of intelligent transportation information systems, but also provides crucial support for intelligent traffic control and management systems. As a key parameter for evaluating traffic conditions and predicting potential accidents, urban road average speed warrants in-depth analysis. By thoroughly examining its characteristics, understanding its variation patterns, and enhancing the accuracy, timeliness, and applicability of predictions, we can not only optimize travel route guidance and alleviate congestion, but also significantly improve the overall safety of urban road transportation systems.

At present, traffic prediction models can be categorized into two types: statistical models and deep learning models. Statistical models primarily include the historical average method, time series models, and filtering algorithms [1]. These models typically assume that future traffic flow trends follow similar patterns of the historical data. Although the historical average method generates predictions by computing the mean of past observations, its results exhibit high volatility, are sensitive to outliers, and fail to capture the complex spatiotemporal features and underlying patterns in traffic data. In the era of big data in transportation, its applicability is limited [2]. Some scholars treat traffic flow prediction as a time series forecasting problem and employ models such as autoregressive moving average (ARMA) for predictions. These models account for temporal correlations in the data, thereby improving prediction accuracy to some extent. However, when traffic flow data exhibit high variance, stationarity must be achieved through differencing, which increases computational complexity [3]. To address non-stationary traffic flow sequences, some researchers apply filtering algorithms for prediction [4]. Nevertheless, the Kalman filter is limited to linear problems, and its effectiveness in nonlinear prediction remains an area of ongoing research [5].

Deep learning models, on the other hand, rely on neural networks to extract patterns from large-scale historical data. These methods can effectively model nonlinear relationships among various factors while leveraging historical information for prediction. Common deep learning models for traffic prediction include long short-term memory (LSTM) [6-8], Bi-directional long short-term memory (BILSTM) [9-10], gated recurrent units (GRU) [11-12], and convolutional neural networks (CNN) [13-14]. While these individual models can achieve satisfactory performance, further improvements in prediction accuracy often require hybrid architectures that combine their strengths. Nowadays, most research is based on hybrid deep learning models, which have the advantages of their respective models and can

Manuscript received April 23, 2025; revised August 20, 2025.

This research was supported by Gansu Provincial Science and Technology Major Special Project - Enterprise Innovation Consortium Project (No.22ZD6GA010), Industry Support Plan Project from Department of Education of Gansu Province (No.2024CYZC-28), Key Research and Development Project of Gansu Province (No.22YF7GA142), and the Natural Science Foundation of China (No.52062027).

Changxi Ma is a Professor at School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China (e-mail: machangxi@mail.lzjtu.cn).

Xiaoyu Huang is a postgraduate student at School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou 730070, China (Corresponding author, e-mail: 12231106@stu.lzjtu.edu.cn).

Bo Du is a senior lecturer at the Department of Management, Griffith University, Brisbane QLD 4111, Australia (e-mail: bo.du@griffith.edu.cn).

further improve prediction accuracy. For example, Wang et al. [15] proposed a mixed deep learning model combining 1D-CNN, LSTM, and attention mechanisms for traffic prediction, achieving excellent performance. Zhang et al. [16] proposed an attention graph convolutional sequence-to-sequence hybrid model (AGC-Seq2Seq), which effectively solves the multi-step prediction problem. Riaz et al. [17] proposed a bidirectional LSTM with fully convolutional network (FCN) based on an attention mechanism, which explicitly modeled the temporal backward dependency of traffic data and achieved strong performance in both short-range and long-range traffic speed prediction. Wang et al. [18] built a prediction model based on ARIMA, GRU, and Wavelet Transform (WT), and applied WT to decompose the speed time-series data, significantly improving prediction accuracy. Ke et al. [19] proposed a two-stream multi-channel convolutional neural network (TM-CNN) model, which explicitly modeled the correlation between lanes; experimental results demonstrated the effectiveness of their method. Dong et al. [20] developed a graph attention network with convolutional gated recurrent units and conducted experiments on real datasets, verifying that the proposed model outperforms state-of-the-art models. Ma et al. [21] proposed the ResCNN-GRU-Attention prediction model, which effectively processed complex and low-quality data and made the prediction more accurate.

All the above studies employ hybrid deep learning models to process and predict traffic data, each with distinct characteristics. These studies show that different hybrid models yield varying performance outcomes. Selecting high-performance hybrid models and adapting them to specific datasets remains a critical challenge. Furthermore, the predictive performance of these hybrid architectures exhibits strong dependence on hyperparameter configurations. Most existing studies manually configure these parameters, which becomes overly cumbersome for models with numerous parameters. Improper parameter settings may significantly degrade model performance. Additionally, current research has largely overlooked the impact of traffic congestion, focusing primarily on external factors while neglecting intrinsic data characteristics.

To address the aforementioned challenges, this study proposes a novel traffic speed prediction model integrating the improved snow geese Algorithm (ISGA) and MHSA mechanisms with Conv1D-GRU architecture. The ISGA algorithm is employed to optimize key hyperparameters of the MHSA-Conv1D-GRU framework. Furthermore, by incorporating traffic congestion factors and relevant feature affecting traffic speed, we construct an enhanced ISGA-MHSA-Conv1D-GRU composite model, which effectively reduces prediction errors and improves accuracy.

The main contributions of this study are as follows: (1) An MHSA-Conv1D-GRU model is proposed, which demonstrates strong capabilities in extracting both local features and long-term dependencies, while enhancing the extraction of key features. (2) The model integrates both traffic congestion levels and temporal variations across weekdays and weekends, incorporates an in-depth correlation analysis of these factors with traffic speed patterns, and demonstrates enhanced prediction realism through validation on real-world datasets. (3) The adoption of ISGA for model parameter optimization eliminates the limitations associated with manual parameter setting. Benchmarking against conventional SGA confirms the significant performance enhancement brought by this methodological advancement.

The remainder of the paper is organized as follows: Section 2 introduces the overall model and its components; Section 3 presents the experiments, including a comparative analysis between the proposed model and multiple baseline models on real-world datasets. Section 4 concludes the paper.

II. Methodology

A. Conv1D

CNN is a type of neural network that involves convolution operations, and has a deep structure. It is widely used for processing time series or image data [22]. It is mainly composed of convolutional layers, pooling layers, and fully connection layers, as illustrated in Fig. 1.

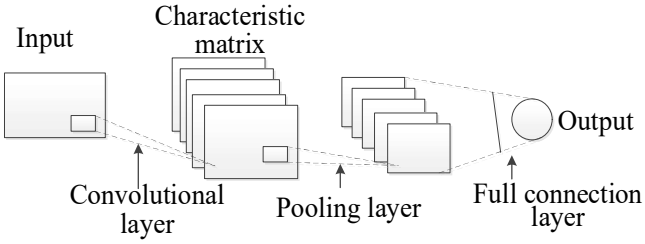


Fig. 1. Overall structure of a CNN model

In a typical CNN structure above, there are two special neural network layers: the convolutional layer and the pooling layer. The convolutional layer performs local feature extraction. The main features and periodic characteristics of the input data are extracted through the convolutional layer. The number of features extracted is mainly determined by the number and size of the convolution kernels. The network extracts features through the convolutional layer, and its mathematical expression is as follows:

$$C_j = \sum W_{ij} * X_j + b_j \tag{1}$$

where * refers to the convolution operation; W_{ij} is the weight of the i-th filter in layer j, $i \in [1, n]$, and n is the number of convolution kernels; X_j is the input data of layer j; b_j is the bias of layer j; C_j is the convolution output of layer j.

The pooling layer can further extract features and down-sample the data, thereby compressing both the data volume and parameters to reduce overfitting. This process combines and reduces dimensions. Pooling operations include average pooling and maximum pooling. This paper adopts maximum pooling. The formula is:

$$PL = \max\left(C_j\right) + b \tag{2}$$

where PL is the output of the pooling layer and b is the deviation.

Therefore, the overall operation of CNN is as follows: Initially, the input feature map passes through the convolution layer containing multiple extracted features. Subsequently, these convolutional features are resampled via the pooling layer's operations. Finally, the pooled neurons are fully connected to a dense layer to generate the final output.

CNN structures can flexibly adapt to different data dimensions. In time series prediction, 1DCNN has significant advantages. It generates a new sequence by sliding a

one-dimensional convolution kernel along the input sequence, performing pointwise multiplication and summation. Although 1DCNN operates along a single dimension, it retains CNN's translation invariance and effectively extracts sequence features. Compared to 2D CNN, it can use larger convolution kernels without significantly increasing computational complexity, thereby expanding the receptive field. The convolutional layer's output is activated by the ReLU function before passing to the pooling layer, where max pooling helps reduce overfitting and improve computational efficiency.

B. GRU

GRU is a type of recurrent neural network (RNN) that addresses the issues of long-term memory and gradient problems during back-propagation in RNNs. It serves a similar purpose to LSTM, but is simpler and easier to train [23-24]. The model structure is shown in Fig. 2.

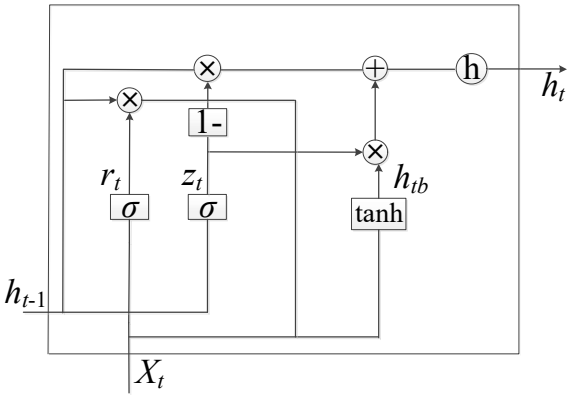


Fig. 2. Structure of a GRU model

GRU employs two gating mechanisms: (1) Reset gate (r_t) controls historical information retention. (2) Update gate (z_t) replaces LSTM's separate forget/input gates. The candidate state (h_{tb}) is generated through reset operations, with final output determined by update gating. The specific formulas are:

$$r_t = \sigma\left(W_r \cdot \left[h_{t-1}, x_t\right]\right) \tag{3}$$

$$z_t = \sigma\left(W_z \cdot \left[h_{t-1}, x_t\right]\right) \tag{4}$$

$$h_{tb} = \tanh\left(W \cdot \left[r_t \odot \left[h_{t-1}, x_t\right]\right]\right) \tag{5}$$

$$h = \left(1 - z_t\right) \odot h_{t-1} + z_t \odot h_{tb} \tag{6}$$

where W is the weight matrix, h is the status, and $\odot$ is the multiplication by element, σ indicates sigmoid activation function.

C. MHSA mechanism

The self-attention mechanism is a strategy used to capture internal dependencies in sequence data. Regardless of the position of the relationship within the sequence, self-attention enables the model to examine different positions in the input sequence to extract meaningful features, thereby capturing the contextual information that may be required in the next step [25-26]. The multi-head self-attention mechanism is an enhancement of the self-attention mechanism. By introducing multiple heads, it can effectively mine the temporal characteristics of the prediction data. Each head can learn different attention representations in parallel across different subspaces and then fuse them, which allows for better capture of data information and more flexible attention to each part of the sequence data [27]. The structure of the multi-head self-attention mechanism is shown in Fig. 3.

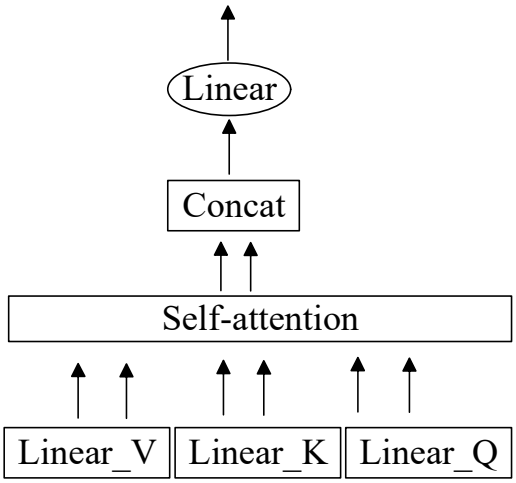


Fig. 3. MHSA mechanism structure

The calculation process of MHSA mechanism is as follows:

(1) Calculate each attention head, and use the operation process of self-attention mechanism for the four heads in the MHSA model to obtain the corresponding results. The calculation formula is as follows:

$$Attention\left(Q, K, V\right) = Soft\max\left(\frac{Q * K^T}{\sqrt{d_k}}\right) * V \tag{7}$$

where Q represents character query vector; K represents character matching vector; V represents character value vector; d_k represents the dimensions of Q and K. The larger the dimension, the larger the corresponding d_k.

(2) Use the *Concat* operation to splice in the channel dimension, and fuse the splicing results through the W^o parameter to get the final result. Its calculation formula is as follows:

$$head_i = Attention\left(QW_i^Q, QW_i^K, QW_i^V\right) \tag{8}$$

$$Cat = Concat\left(head_1, head_2, \cdots, head_{num}\right) \tag{9}$$

$$MultiHead\left(Q, K, V\right) = Cat * W^o \tag{10}$$

where *head* refers to the *i*-th self-attention mechanism of the MHSA module; W_i^Q, W_i^K, W_i^V are independent trainable parameter matrices for each attention head; *Concat* indicates splicing operation; *num* is the number of splicing vectors.

D. ISGA

The SGA is a new metaheuristic algorithm proposed in 2024 [28]. Its inspiration comes from the migration behavior of snow geese, particularly the unique "herringbone" and "straight-line" flight patterns formed during their migration. By simulating the flying behavior of snow geese, the algorithm achieves efficient search and optimization in the solution space. The SGA is primarily divided into three stages: the initialization stage, the exploration stage (herringbone shapes), and the exploitation stage (straight line shapes).

The initialization stage is the initialization of individual position. The position of snow geese is randomly generated in the search space, and the initial position is determined according to the population size, solution space boundary and dimension. The location update formula is as follows:

$$P = lb + rand \times (ub - lb) \tag{11}$$

where P is the initialized set position, ub and lb are the upper and lower bounds of the solution space respectively, and $rand$ is a random number in the range of [0,1].

The exploration stage simulates the "herringbone" flight formation of snow geese to enhance the population's search capability within the solution space. During migration, factors such as aerodynamic drag and energy expenditure significantly influence their flight dynamics. The geese's energy level initially increases and then gradually declines as they approach the destination. For simplicity, gravitational effects are excluded from the analysis. The flight dynamics are modeled by the following formula:

$$V^{t+1} = cV^t + a \tag{12}$$

$$c = \frac{4t}{Me^{\frac{4t}{M}}} \tag{13}$$

where V is the velocity, c is the weight factor, and a is the acceleration. M is the maximum number of iterations.

During snow geese flight, two fundamental factors must be considered: the aerodynamic drag (air resistance) and their intrinsic flight energy, both governed by Newton's second law. The position update is then calculated using the following formula:

$$P_i^{t+1} = P_i^t + b\left(P_b^t - P_i^t\right) + V_i^{t+1} \tag{14}$$

where P_i^{t+1} is the position of the current optimal individual, and b is the weight coefficient.

During the exploitation stage, simulate the "straight line" flight mode of snow geese and focus on searching to improve the algorithm's development capabilities. When the angle between snow geese exceeds π, they will enter this stage. The location update formula is as follows:

$$P_i^{t+1} = \begin{cases} P_i^t + \left(P_i^t - P_b^t\right) r & r > 0.5 \\ P_b^t + \left(P_i^t - P_b^t\right) r \odot \text{Brownian}(d) & r \le 0.5 \end{cases} \tag{15}$$

where r is a random number within the range of [0,1], if the random number $r > 0.5$, the goose follows experienced and physically strong peers to collectively search for the best destination; When $r \leqslant 0.5$, if trapped in a local solution, geese will exhibit random behavior similar to Brownian motion. ⊙ representing element by element multiplication, and Brownian (d) represents Brownian motion.

ISGA is an improved variant of SGA. ISGA significantly improves the exploration and development ability of the algorithm by introducing three improvement strategies, so as to improve the convergence speed and accuracy of the algorithm [29]. Three improvement strategies are as follows.

The first mechanism is the leader geese rotation, which simulates the migration process of snow geese. When the leader geese become tired, other strong individuals take over their position to maintain flight efficiency and speed. Through a competition mechanism, the individual with the highest fitness value is selected as the new leader, thereby enhancing the algorithm's global exploration capability. The location update formula is as follows.

$$P_i^{t+1} = (1 - rand) P_{k1}^t + raand\left(\frac{P_{k1}^t + P_b^t}{2}\right) + a_1\left(\left(P_b^t\right) - \frac{P_{k2}^t + P_b^t}{2}\right) \tag{16}$$

where, P_{k1} and P_{k2} respectively represent the top three and top five individual positions in the population, and a_1 is the weight factor.

The second mechanism is the call guidance mechanism, which simulates how snow geese communicate through calls to guide their flight direction. By using an attenuation model of sound wave propagation, the position update is adjusted based on the distance between an individual and the leading goose, thereby avoiding a decline in convergence ability due to excessive aggregation or dispersion. The location update formula is as follows.

$$P_i^{t+1} = P_i^t + \left(1 - \frac{L_A\left(l_i^t\right) - L_{low}}{L_{WA} - L_{low}}\right)\left(P_b^t - P_i^t\right) \odot \times \text{Brownian}(d) + a_2\left(P_c^t - P_i^t\right) \tag{17}$$

where L_A is the received sound intensity, L_{WA} is the initial sound intensity of the sound source, L_{low} is the lowest acceptable sound intensity, and a_2 is the weight factor.

The third mechanism is the abnormal boundary strategy, which considers the behavior of snow geese as social birds that avoid isolation. By calculating the difference between an individual's fitness and the group's average fitness, the strategy adjusts the individual's position update to enhance the algorithm's convergence speed and accuracy. The position update formula is as follows.

$$P_i^{t+1} = \begin{cases} P_b^t + a_3\left(P_b^t - P_i^t\right) & f\left(P_i^t\right) > f(P)_{avg} \\ P_i^t - e\dfrac{\left(P_i^t - P_n^t\right)}{\left(f\left(P_n^t\right) - f\left(P_i^t\right)\right)} + levy & f\left(P_i^t\right) \le f(P)_{avg} \end{cases} \tag{18}$$

where $f(P)_{avg}$ is the average fitness value of the population, P_n^t is the position of the individual with the lowest fitness value, e is the weight factors, a_3 is a random number with a range of (0,1), $levy$ is the levy flight strategy, which effectively simulates the irregular movement of snow geese in preventing them from leaving the sheep.

During exploration, ISGA employs leader rotation to enhance global search capability; during exploitation, it utilizes call guidance and abnormal boundary strategies to refine local search accuracy.

E. MHSA-Conv1D-GRU model

In traffic speed prediction problems, speed data exhibits characteristics such as periodicity, trends, outliers, and noise, while also containing implicit temporal dependencies. Effectively capturing these autocorrelations is crucial for prediction performance. This requires the model to capture both short-term and long-term patterns in traffic data while possessing robust time-series modeling capabilities.

To address this, building on the GRU architecture which captures both long- and short-term characteristics of traffic data, this paper integrates Conv1D with its superior global feature extraction capability for optimization, constructing a Conv1D-GRU model. This model effectively captures multi-scale temporal dependencies, extracting rich short-term and long-term feature information to better characterize key historical traffic patterns.

To further enhance the network's ability to represent and model base station traffic data, this paper introduces a MHSA to fuse multi-perspective attention information. This enables the Conv1D-GRU network to adaptively focus on key features across different temporal dimensions, thereby

improving prediction accuracy. The overall architecture of the MHSA-Conv1D-GRU model is shown in Fig. 4.

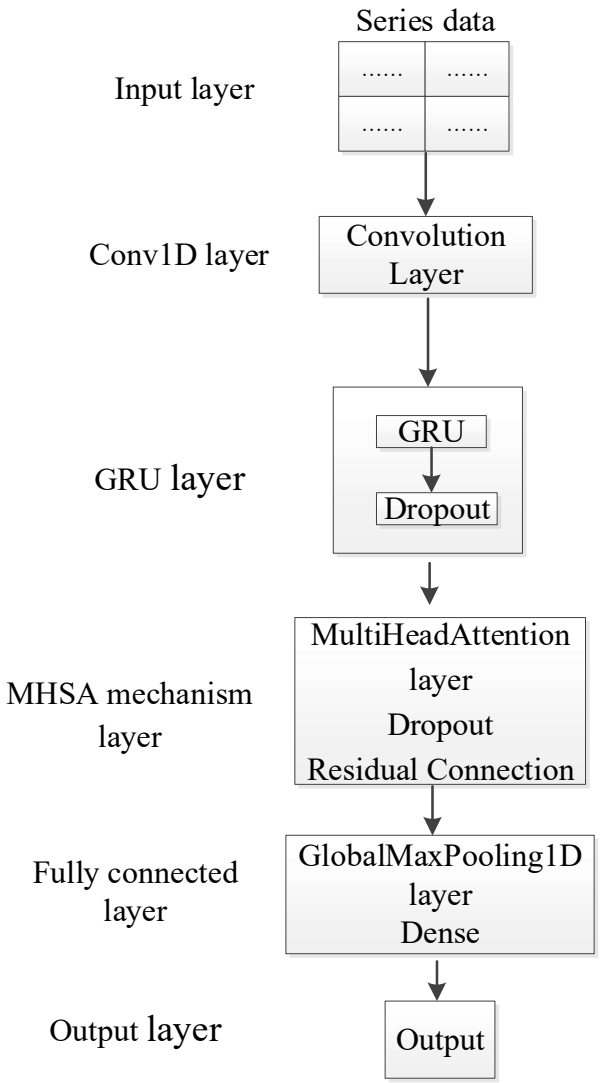


Fig. 4. Structure of a MHSA-Conv1D-GRU model

Input layer: Preprocess historical data and input multi feature time series into the model, represented as.

$$X = [X_1, X_2, ..., X_n]^T, X_t \in R^d \tag{19}$$

where n is the sequence length; d is the feature dimension; X_t is the eigenvector at time step t.

Conv1D layer: A one-dimensional convolution operation is applied to extract local temporal features, and the output of the Conv1D layer is shown below.

$$C_t = ReLU\left(W_{ij} * X_t + b_j\right), \quad C_t \in R^{d_c} \tag{20}$$

where d_c is the feature dimension output by convolution.

GRU layer: Long-term dependencies are extracted using two GRU layers with dropout, and the processed data $H^{(2)}$ is shown below.

$$H_t^{(1)} = GRU_1(C_t, H_{t-1}^{(1)}), \quad H_t^{(1)} \in R^{h_1} \tag{21}$$

$$\tilde{H}_t^{(1)} = Dropout(H_t^{(1)}) \tag{22}$$

$$H^{(1)} = [\tilde{H}_1^{(1)}, \tilde{H}_2^{(1)}, \ldots, \tilde{H}_n^{(1)}]^T, \tilde{H}_t^{(1)} \in R^{h_1} \tag{23}$$

where $H_t^{(1)}$ is the hidden state of GRU at the t-th time step, with a hidden dimension of $h1$. $\tilde{H}_t^{(1)}$ is the output after dropout processing. $H^{(1)}$ is the output of the layer GRU.

MHSA mechanism layer: Global features are extracted from the complete time series to capture long-range dependencies, with the resulting output shown below.

$$Q == W_Q H^{(1)}, K = W_K H^{(1)}, V = W_V H^{(1)} \tag{24}$$

$$A = softmax(\frac{QK^T}{\sqrt{h_a}}) \tag{25}$$

$$O = AV, \quad O \in R^{n \times h_1} \tag{26}$$

$$\tilde{O} = Dropout(O) \tag{27}$$

$$Z = LayerNorm(H^{(1)} + \tilde{O}) \tag{28}$$

where W, Q, K are queries, keys, and values; W_Q, W_K, and W_V are linear transformation parameters with a shape of $R^{h1 \times ha}$; A is the attention weight; O is attention output; $\tilde{O}$ is the output of the Dropout layer; Z is the output after residual connection.

Fully connected layer: Through additional processing, the network extracts salient features, with the final output presented below.

$$P = GlobalMaxPooling(Z), \quad P \in R^{h_2} \tag{29}$$

$$Y = W_o P + b_o \tag{30}$$

where P is a fixed dimensional vector; W_o is the fully connected layer weight; b_o is the bias term.

Output layer: The output layer applies transformations to the preceding dense layer's activations, computed as:

$$\hat{Y} = \sigma(Y) \tag{31}$$

where $\hat{Y}$ is the predicted output value; σ is the sigmoid activation function.

F. ISGA-MHSA-Conv1D-GRU model

The ISGA-MHSA-Conv1D-GRU traffic speed prediction hybrid model is established, which primarily takes the number of neurons in the GRU hidden layers, the number of heads in the MHSA layer, and the learning rate as optimization objectives for the ISGA algorithm in the MHSA-Conv1D-GRU framework. Through global search optimization, the optimal parameter combination is assigned to the MHSA-Conv1D-GRU model to enhance prediction accuracy. The prediction process of the ISGA-MHSA Conv1D-GRU model is illustrated in Fig. 5.

Step 1: The model initialization phase establishes critical parameters for both the optimization algorithm and neural network architecture. For the ISGA, we define the population size representing potential solutions and set the maximum iteration limit to control optimization duration. Concurrently, we configure the MHSA-Conv1D-GRU model by specifying its architectural components: the number and dimensions of convolutional layers, GRU layer configurations, and the mean squared error (MSE) loss function for performance evaluation. This dual initialization ensures proper coordination between the optimization process and model training.

Step 2: The optimization process commences by mapping the initial population positions to neural network

hyperparameters, including learning rates, kernel sizes, and layer dimensions. Each candidate solution undergoes training on the designated dataset, with the resulting MSE serving as the fitness score for evolutionary selection. This evaluation phase provides the necessary performance metrics to guide subsequent optimization steps while maintaining computational efficiency through parallel processing of candidate solutions.

Step 3: Population evolution follows a hierarchical update strategy governed by mathematical formulations. The entire population's positions undergo transformation through formula (11), while individual velocities are adjusted according to formula (12). Elite members (top 20%) receive specialized updates via the leader rotation mechanism in formula (16), promoting exploration of promising solution spaces. The majority cohort (next 60%) follows the call guidance protocol from formula (17), balancing exploitation of current best solutions. Formula (18) enforces boundary constraints to maintain parameter validity throughout these updates, ensuring all hyperparameters remain within their defined operational ranges.

Step 4: The optimization cycle iteratively repeats, with each generation's refined parameters feeding back into the training process. This continues until either reaching the maximum iteration count or meeting convergence criteria. The final output selects the parameter set demonstrating optimal fitness (minimum MSE), which configures the complete ISGA-MHSA-Conv1D-GRU model. This optimized architecture then processes traffic speed data to generate final predictions, completing the implementation pipeline while maintaining all intermediate validation checks and performance benchmarks.

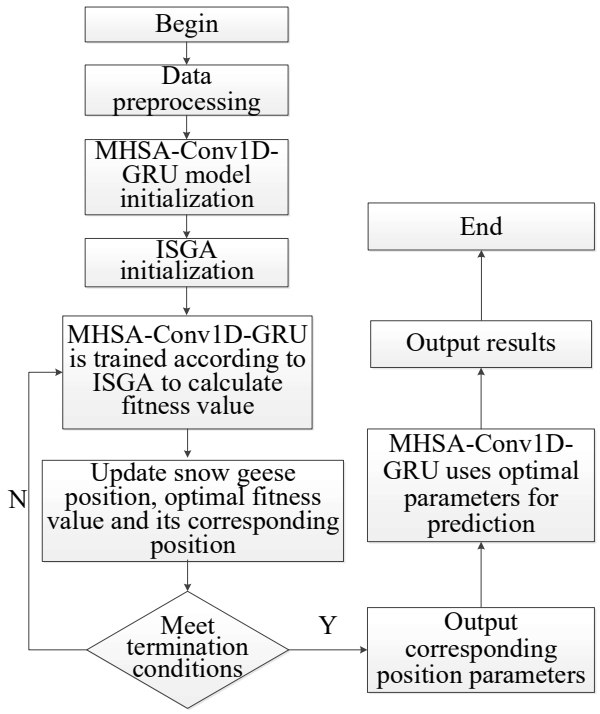


Fig.5. Prediction process of the ISGA-MHSA-Conv1D-GRU model

III. Experimental Results and Analysis

A. Data Description and Analysis

In this study, traffic data of the road networks in Chengdu and Wuhan were collected from 4 March to 24 March, 2025, using the API of Baidu Maps. A total of 5,760 traffic data points were obtained for each city. The processed data samples are presented in Table I, and 0 represents working days and 1 represents weekends.

We analyze the impact of workdays and congestion index on traffic speed, highlighting the importance of considering these two factors in the model. In Fig. 6, the traffic speed curves on weekdays and weekends are shown using Wuhan dataset as an example.

Fig. 6 illustrate the fluctuations in traffic speed throughout the day on weekdays and weekends. Additionally, there are significant differences in travel demand between weekdays and weekends at different times. For example, on weekdays, the speed during peak hours drops sharply due to a large number of residents commuting by car. During off-peak hours, the speed increases significantly. In contrast, on non-working days, the decline in speed during peak hours is more gradual, and the lowest speed remains higher than that on working days. Between 10:00 and 17:00, the traffic speed on non-working days is lower than that on working days. This may be because residents are at work and are less likely to travel on weekdays, whereas on weekends, increased travel activities lead to reduced speeds. Statistical analysis reveals significant weekday-speed associations.

TABLE I
Sample Data in Datasets Chengdu and Wuhan

	Date time	Speed (km/h)	Congestion index	Weekend
Dataset Chengdu	2025-3-8 17:30	27.849	1.822	1
	2025-3-8 17:35	27.470	1.847	1
	2025-3-8 17:40	26.587	1.909	1
	2025-3-8 17:45	26.278	1.931	1
	2025-3-8 17:50	26.491	1.916	1
Dataset Wuhan	2025-3-5 7:00	38.965	1.152	0
	2025-3-5 7:05	37.240	1.206	0
	2025-3-5 7:10	35.257	1.273	0
	2025-3-5 7:15	32.858	1.366	0
	2025-3-5 7:20	30.700	1.462	0

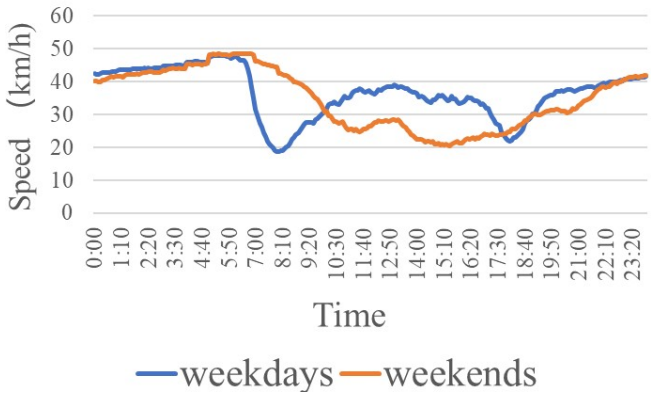


Fig.6. Continuous daily speed on weekdays and weekends

This study employs the Pearson correlation test to analyze the relationship between the congestion index and traffic speed. The Pearson correlation test formula is as follows.

$$\rho = \frac{\sum\left(A_i - \overline{A}\right)\left(B_i - \overline{B}\right)}{\sqrt{\sum\left(A_i - \overline{A}\right)^2} \cdot \sqrt{\sum\left(B_i - \overline{B}\right)^2}} \tag{32}$$

where ρ is the Pearson correlation coefficient, with a value range of [-1,1]. A_i and B_i are the sample values of variables A and B, respectively. $\overline{A}$ and $\overline{B}$ are the means of A and B, respectively. If ρ is close to 1, it indicates a high positive correlation; If ρ is close to -1, it indicates a high negative correlation.

According to formula (32), the ρ value of Dataset Chengdu is -0.9546, and that of Dataset Wuhan is -0.9521. The results show that both datasets yield values close to -1, indicating a strong negative correlation between congestion and traffic speed.

Based on the above analysis, traffic speed, road congestion index, and whether it is a working day are used as input features for the model. The data input is normalized via min–max normalization, and all datasets are divided at a ratio of 8:2.

B. Baseline Models

This study employs three comparison experiments and one ablation experiment. The first comparative experiment evaluates LSTM model, Conv1D model, and GRU model; The second comparative experiment compares the GRU-LSTM, Conv1D-GRU models; The third comparative experiment contrasts ISGA-MHSA-Conv1D-GRU model with SGA-MHSA-Conv1D-GRU model. The ablation experiment assesses Conv1D model, GRU model, Conv1D-GRU model, MHSA-Conv1D-GRU, and ISGA-MHSA-Conv1D-GRU model. Together, these experiments validate the predictive performance of the ISGA-MHSA-Conv1D-GRU model. The baseline models are described as follows.

LSTM: One of the classic variants of recurrent neural networks [30].

Conv1D: One dimensional convolutional layer extracts local features through local receptive fields [31].

GRU: GRU simplifies RNN architecture with fewer parameters than LSTM while maintaining comparable performance [32].

GRU-LSTM: The model integrates the strengths of GRU and LSTM, exhibiting superior performance in time series prediction [33].

Conv1D-GRU: The model leverages convolutional operations for local pattern detection and GRU networks for sequential dependency modeling.

MHSA-Conv1D-GRU: The model incorporates Conv1D, GRU, and MHSA, further enhancing its capability to extract both local features and long-term dependencies.

SGA-MHSA-Conv1D-GRU: The model is based on SGA and MHSA-Conv1D-GRU. SGA is used to optimize the hyperparameters of the MHSA-Conv1D-GRU model.

C. Parameter Settings

The ISGA-MHSA-Conv1D-GRU model utilizes the Adam optimizer with hyperparameters optimized through the ISGA, with initial configurations detailed in Table II. The architecture comprises a single one-dimensional convolutional layer with 24 filters, a kernel size of 3, and ReLU activation, followed by one GRU layer employing ReLU activation and hidden layer neurons ranging between 1 and 100 as determined by ISGA. The MHSA mechanism's head count is similarly optimized by ISGA, with the dimensionality of Key, Query, and Value vectors matching the neuron count in the GRU layer. A uniform dropout rate of 0.2 is applied throughout the network, while ISGA concurrently optimizes three critical parameters: the number of MHSA attention heads, the learning rate, and GRU hidden layer neuron counts.

TABLE II
ISGA PARAMETER SETTINGS

Parameter	Value
Population size	20
L_{WA}	65
L_{low}	20
Maximum number of iterations	10

D. Evaluating Indicator

The evaluation indicators selected for the article include mean square error (MSE), mean absolute error (MAE), and coefficient of determination (R^2). The formulas are defined as follows.

$$\mathrm{MSE} = \frac{1}{m}\sum_{i=1}^{m}\left(\hat{y}_i - y_i\right)^2 \tag{33}$$

$$\mathrm{MAE} = \frac{1}{m}\sum_{i=1}^{m}\left|\hat{y}_i - y_i\right| \tag{34}$$

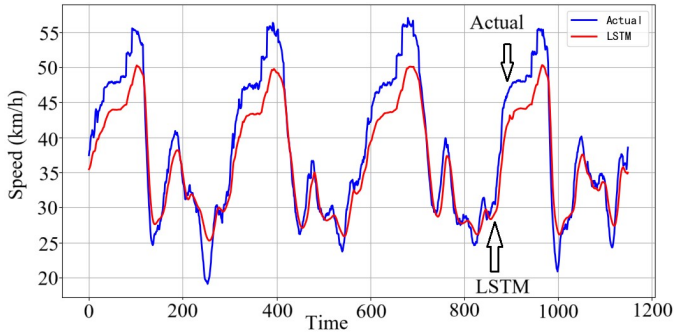
$$R^2 = \frac{\sum_{i=1}^{m}\left(\hat{y}_i - \overline{y_i}\right)^2}{\sum_{i=1}^{m}\left(y_i - \overline{y_i}\right)^2} \tag{35}$$

where y_i is the real value, $\widehat{y_i}$ is the predicted value, $\overline{y_i}$ is the average value, and m is the number of samples. The smaller the MSE and MAE, the smaller the model's error in fitting the true values, and the higher its prediction accuracy; Models with R^2 values nearer to 1 exhibit better fitting performance.

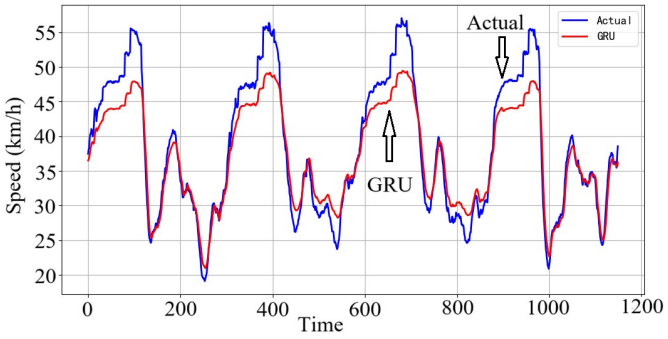
E. Experimental Results and Analysis

The experimental environment is configured with Windows 11 OS, utilizing Python for implementation and TensorFlow 2.6 for deep learning computations.

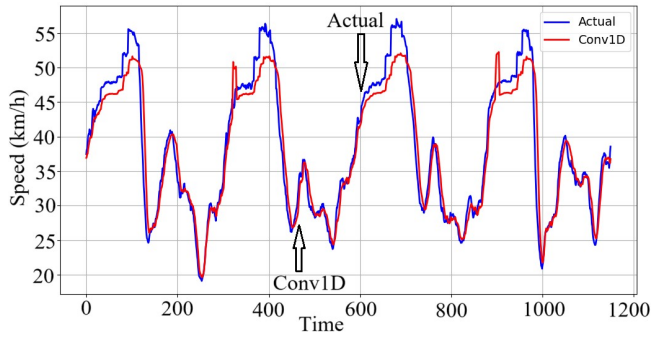
We firstly conduct comparative experiments between the LSTM, Conv1D, and GRU models on Dataset Chengdu. Both LSTM and GRU are designed with one hidden layer, each containing 12 neurons and a ReLU activation function. The Conv1D parameters follow the settings described in the previous section. All three models are optimized using the Adam optimizer. The results are shown in Fig. 7. The fitting performance of these three single-model predictors is relatively low, suggesting that combining them might improve prediction accuracy. The performance metrics of the three single models are presented in Table III.



(a) LSTM model



(b) GRU model



(c) Conv1D model

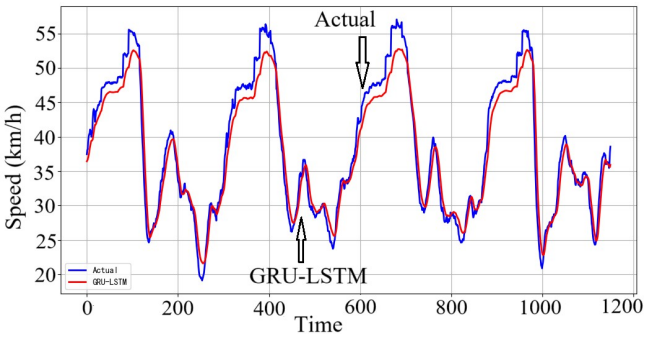
Fig. 7. Comparison on prediction results of the three single models

TABLE III
EVALUATION METRICS ACROSS THE THREE MODELS

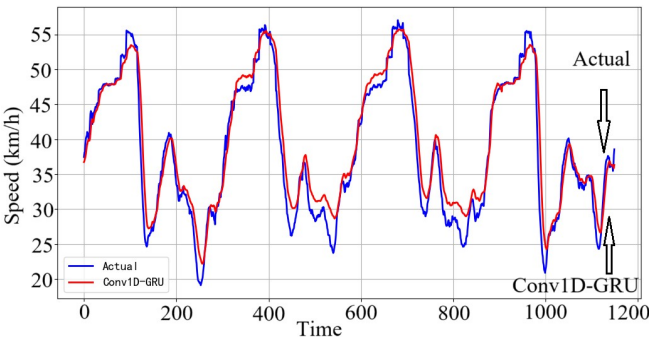
Model	MSE	MAE	R^2
LSTM	11.67	2.79	0.87
GRU	10.04	2.48	0.89
Conv1D	7.04	1.94	0.92

Table III shows that LSTM performs the worst among the three models. Compared to GRU and Conv1D, its MSE increases by 16.2%–65.7%, MAE increases by 12.5%–43.8%, and R^2 decreases by 0.02-0.05. GRU outperforms LSTM but remains inferior to Conv1D, which achieves the highest prediction accuracy. Therefore, combining GRU and Conv1D, which exhibit stronger predictive performance, may further enhance model accuracy.

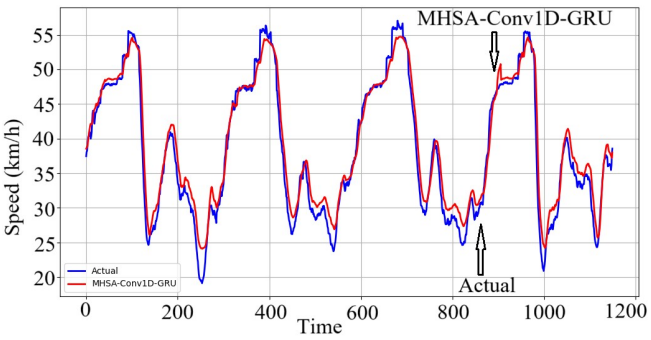
Subsequently, we conduct ablation studies by combining Conv1D and GRU to construct a hybrid Conv1D-GRU model, which is compared with GRU-LSTM model. To further investigate its potential, we introduce the MHSA mechanism and evaluate whether it could enhance predictive performance. The GRU-LSTM model consists of one GRU layer and one LSTM layer with parameters consistent with previous section, followed by a Dropout layer (rate=0.5). The Conv1D-GRU model maintains identical Conv1D and GRU configurations as previously described, while the MHSA component uses 3 attention heads with all other parameters unchanged from earlier implementations. The prediction results are presented in Fig. 8. Table IV shows the evaluation metrics of the three models.



(a) GRU-LSTM model



(b) Conv1D-GRU model



(c) MHSA-Conv1D-GRU model

Fig.8. Comparison on the prediction results of the three models

TABLE IV
EVALUATION METRICS OF THE THREE MODELS

Model	MSE	MAE	R^2
GRU-LSTM	6.39	1.80	0.92
Conv1D-GRU	5.97	1.78	0.93
MHSA-Conv1D-GRU	4.02	1.56	0.95

Tables III and IV demonstrate that hybrid architectures consistently outperform standalone models in prediction accuracy. The GRU-LSTM model achieves superior performance compared to individual GRU or LSTM models, with 42.5% lower MSE, 31.4% reduced MAE, and 0.03-0.05 higher R^2 values. Similarly, the Conv1D-GRU hybrid model exhibits significant improvements over single Conv1D or GRU models, showing 17.9-68.1% MSE reduction, 8.9-39.3% MAE decrease, and 0.01-0.04 R^2 enhancement. Notably, the proposed model further surpasses GRU-LSTM

performance with additional 6.5% MSE reduction, 1.1% MAE improvement, and 0.01 R^2 increase, attributable to Conv1D's enhanced local feature extraction capability compared to LSTM. The integration of MHSA mechanisms further boosts the model's predictive performance, with the MHSA-Conv1D-GRU variant attaining an MSE of 4.02 (48.5% reduction versus the baseline hybrid), MAE of 1.56 (14.1% decrease), and R^2 of 0.95 (0.02 increase). It is important to note that these results are obtained using manually configured parameters without exhaustive optimization, suggesting considerable potential for further accuracy enhancement.

To fully exploit this potential, we implement the ISGA for systematic hyperparameter optimization of critical parameters including attention head count, learning rate, and the number of hidden layer neurons.

Furthermore, a complementary experiment is conducted to compare the ISGA with the SGA in order to validate the algorithm's performance improvement. After SGA optimization, the model used 2 MHSA attention heads, 60 GRU neurons, and a learning rate of 0.004. With ISGA optimization, the model configuration was 5 attention heads, 99 GRU neurons, and a learning rate of 0.001. The prediction results are presented in Fig. 9. Table V shows the evaluation metrics of the two models.

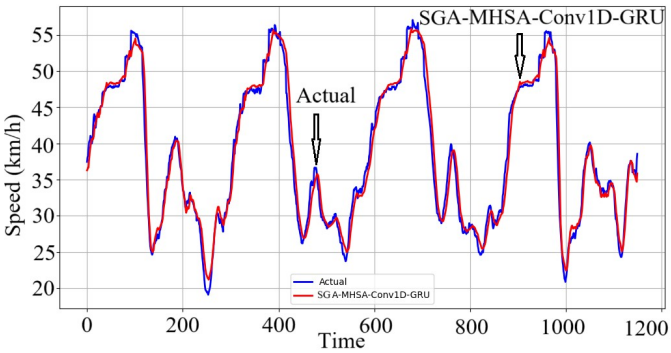
TABLE V
EVALUATION INDICATORS FOR TWO OPTIMIZED MODELS

Model	MSE	MAE	R^2
SGA- MHSA-Conv1D-GRU	2.58	1.15	0.97
ISGA- MHSA-Conv1D-GRU	1.74	1.12	0.98

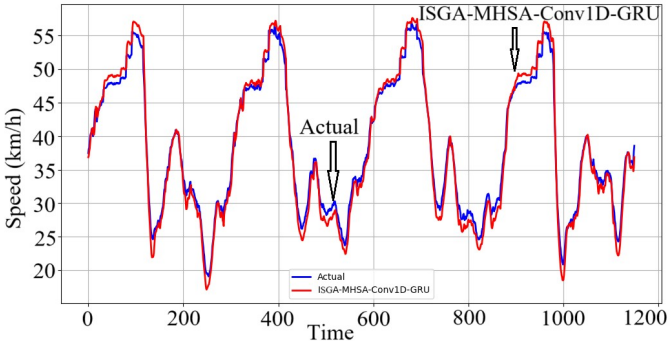
Tables IV and V demonstrate that the SGA-optimized model achieves superior prediction accuracy compared to the unoptimized baseline. Furthermore, ISGA exhibits enhanced optimization effectiveness relative to SGA. Specifically, the SGA-MHSA-Conv1D-GRU model shows better metrics with an MSE of 2.58 (a 35.8% reduction versus the unoptimized baseline), MAE of 1.15 (a 26.3% reduction), and R^2 of 0.97 (a 0.02 increase). When benchmarked against SGA optimization, the ISGA achieves significant performance improvements: 32.8% in MSE reduction, 2.6% in MAE reduction, and 0.01 in R^2 increase. These results confirm the enhanced optimization capability of ISGA over traditional SGA methods, demonstrating the outstanding prediction accuracy of the proposed model.

To further verify the performance of the proposed model and enhance the experimental results. The same eight models, with parameters unchanged from Dataset Chengdu testing, are subsequently applied to Dataset Wuhan. Fig. 10 shows the prediction indicators of the eight models.

In consistent with the findings from Dataset Chengdu, the hybrid architecture demonstrates superior predictive capability compared to individual model configurations. Furthermore, the optimized hybrid variant exhibits enhanced accuracy relative to its non-optimized counterpart. The ISGA-MHSA-Conv1D-GRU exhibits remarkable forecasting capability. Fig. 10(a) shows the column diagram of MSE. The MSE value of the ISGA-MHSA-Conv1D-GRU model is the lowest, which is 93.9%-26.5% lower than those of other models, indicating more accurate predictions that better match real data. As evidenced in Fig. 10(b), our approach achieves substantially reduced MAE value, which is 26.5% to 80.1% lower than those of other models, demonstrating significantly better performance than baseline approaches. Fig. 10(c) confirms that the R^2 of the proposed model is 0.99, increased by 0.01-0.1, indicating good fitting performance. Therefore, the ISGA-MHSA-Conv1D-GRU model has good prediction performance.

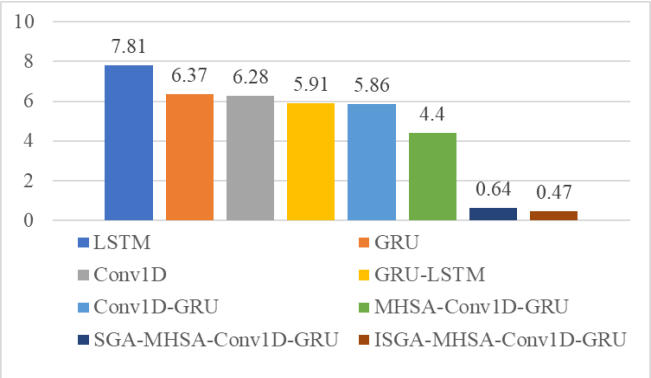


(a) SGA- MHSA-Conv1D-GRU model

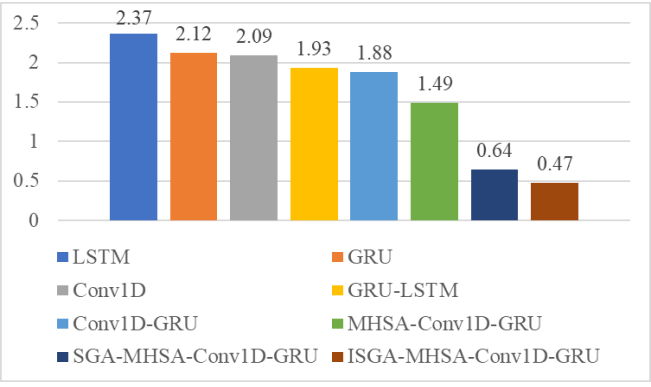


(b) ISGA- MHSA-Conv1D-GRU model

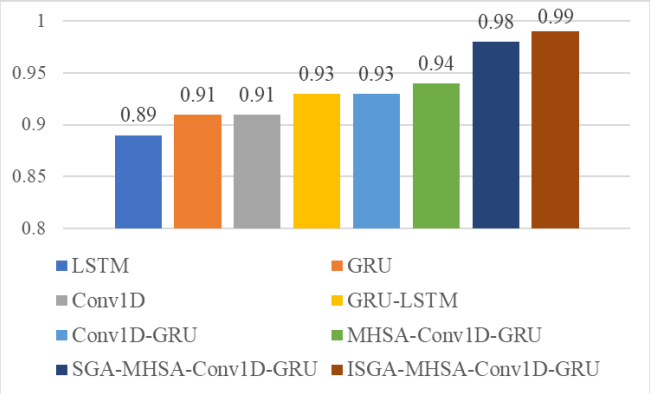
Fig.9. Comparison on the prediction results of the two models



(a) MSE



(b) MAE



(c) R^2

Fig.10. Evaluation indicators for eight models

IV. Conclusions

To improve the accuracy of traffic speed prediction, historical speed data is analyzed to examine the impact of traffic congestion and weekends on speed correlation. Given the strict temporal nature of the data, an ISGA-MHSA-Conv1D-GRU prediction model is developed using deep learning methods. The proposed model achieved significant improvements over both the baseline MHSA-Conv1D-GRU without ISGA optimization (showing 56.7% lower MSE, 28.2% lower MAE, and 0.03% higher R^2) and the SGA-optimized model (demonstrating 32.8% MSE reduction, 2.6% MAE reduction, and 0.01 R^2 increase), confirming ISGA's superior optimization capability compared to both unoptimized and SGA-optimized approaches. These results provide a forward-looking theoretical foundation for mitigating traffic congestion through data-driven forecasting. However, this study focuses solely on temporal feature analysis, leaving spatial features for future research.

References

[1] Y. Xie, K. Zhao, Y. Sun, D. Chen, "Gaussian processes for short-term traffic volume forecasting," *Journal of the Transportation Research Board*, vol.2165, no.1, pp69-78, 2010.

[2] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. -Y. Wang, "Traffic Flow Prediction With Big Data: A Deep Learning Approach," *IEEE Transactions on Intelligent Transportation Systems*, vol.16, no.2, pp865-873, 2015.

[3] S. Shahriari, M. Ghasri, S. A. Sisson, and T Rashidi, "Ensemble of ARIMA: combining parametric and bootstrapping technique for traffic flow prediction," *Transportmetrica A: Transport Science*, vol.16, no.3, pp1552-1573, 2020.

[4] L. Cai, Z. Zhang, J. Yang, Y. Yu, T. Zhou, and J. Qin, "A noise-immune Kalman filter for short-term traffic flow forecasting," *Physica A: Statistical Mechanics and its Applications*, vol.536, p. 122601, 2019.

[5] X. Ma, Z. Dai, Z. He, J. Ma, Y. Wang, and Y.P. Wang, "Learning Traffic as Images: A Deep Convolutional Neural Network for Large-Scale Transportation Network Speed Prediction," *Sensors*, vol. 17, no.4, p.818, 2017.

[6] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp187-197, 2015.

[7] Z. Zhao, W. Chen, X. Wu, P.C.Y. Chen, and J. Liu, "LSTM network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol.11, pp68-75, 2017.

[8] Z. Gong, B. Du, Z. Liu, W. Zeng, P. Perez and K. Wu, "SD-seq2seq: A Deep Learning Model for Bus Bunching Prediction Based on Smart Card Data," in *29th International Conference on Computer Communications and Networks (ICCCN)*, 2020, Honolulu, HI, USA, pp1-9.

[9] Yongqing Wu, Yao Jin, Peng Sun, and Zhichen Ding, "HBDTA: Hierarchical Bi-LSTM Networks for Drug-target Binding Affinity Prediction," *Engineering Letters*, vol. 32, no. 2, pp284-295, 2024.

[10] Bo Zhang, Xinfeng Yang, Yongqing Zhang, and Dongzhi Li, "Short-Term Inbound Passenger Flow Prediction of Urban Rail Transit Based on RF-BiLSTM," *Engineering Letters*, vol. 31, no.2, pp665-673, 2023.

[11] W. Shu, K. Cai and N. N. Xiong, "A Short-Term Traffic Flow Prediction Model Based on an Improved Gate Recurrent Unit Neural Network," *IEEE Transactions on Intelligent Transportation Systems*, vol.23, no.9, pp16654-16665, 2022.

[12] Xin Gao, Chen Xue, Wenqiang Jiang, and Bao Liu, "Attention-GRU Based Intelligent Prediction of NOx Emissions for the Thermal Power Plants," *IAENG International Journal of Computer Science*, vol. 51, no. 8, pp1171-1181, 2024.

[13] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, "Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning," *Transportmetrica A: Transport Science*, vol.15, no. 2, pp1688-1711, 2019.

[14] N. Formosa, M. Quddus, S. Ison, M. Abdel-Aty, and J. Yuan. "Predicting real-time traffic conflicts using deep learning," *Accident Analysis & Prevention*, vol.136, p.105429, 2020.

[15] K. Wang, C. Ma, Y. Qiao, X. Lu, W. Hao, and S. Dong, "A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction," *Physica A: Statistical Mechanics and its Applications*, vol.583, p.126293, 2021.

[16] Z. Zhang, M. Li, X. Lin, Y. Wang, and F. He, "Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies," *Transportation Research Part C: Emerging Technologies*, vol.105, pp297-322, 2019.

[17] A. Riaz, H. Rahman, M.A. Arshad, M. Nabeel, A. Yasin, M.H. Al-Adhaileh, E.T. Eldin, and N.A. Ghamry, "Augmentation of Deep Learning Models for Multistep Traffic Speed Prediction," *Applied Sciences*, vol.12, p.9723, 2022.

[18] K. Wang, C. Ma, and X. Huang, "Research on traffic speed prediction based on wavelet transform and ARIMA-GRU hybrid model," *International Journal of Modern Physics C*, vol.34, no.10, p.2350127, 2023.

[19] R. Ke, W. Li, Z. Cui, and Y. Wang, "Two-Stream Multi-Channel Convolutional Neural Network (TM-CNN) for Multi-Lane Traffic Speed Prediction Considering Traffic Volume Impact," *Transportation Research Record*, vol.2674, no.4, pp725-735, 2022.

[20] C. Dong, K. Zhang, X. Wei, Y. Wang, Y. Yang, "Spatiotemporal Graph Attention Network modeling for multi-step passenger demand prediction at multi-zone level," *Physica A: Statistical Mechanics and its Applications*, vol.603, p.127789, 2022.

[21] C. Ma, B. Zhang, S. Li, and Y. Lu, "Urban rail transit passenger flow prediction with ResCNN-GRU based on self-attention mechanism," *Physica A: Statistical Mechanics and its Applications*, vol.638, p. 129619, 2024.

[22] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, "Recent advances in convolutional neural networks," *Pattern Recognition*, vol.77, pp354-377, 2018.

[23] C. Ma, Y. Zhao, G. Dai, X. Xu, and Sze-Chun Wong, "A Novel STFSA CNN-GRU Hybrid Model for Short-Term Traffic Speed Prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol.24, no.4 pp3728-3737, 2023.

[24] J. Zhou, Y. Qin, D. Chen, F. Liu, and Q. Qian, "Remaining Useful Life Prediction of Bearings by a New Reinforced Memory GRU Network," *Advanced Engineering Informatics*, vol.53, p.101628, 2022.

[25] X. Ran, Z. Shan, Y. Fang, and C. Lin, "An LSTM-based method with attention mechanism for travel time prediction," *Sensors*, vol.19, no. 4, p.861, 2019.

[26] H. Zang, R. Xu, L. Cheng, T. Ding, L. Liu, Z. Wei, and G. Sun, "Residential load forecasting based on LSTM fusing self-attention mechanism with pooling," *Energy*, vol.229, p.120682, 2021.

[27] Y. Hong, Y. Zhang, K. Schindler, and M. Raubal, "Context-aware multi-head self-attentional neural network model for next location prediction," *Transportation Research Part C: Emerging Technologies*, vol.156, p.104315, 2023.

[28] A. Tian, F. Liu, and H. Lv, "Snow Geese Algorithm: A novel migration-inspired meta-heuristic algorithm for constrained engineering optimization problems," *Applied Mathematical Modelling* vol.126, pp327-347, 2024.

[29] H. Bian, C. Li, Y. Liu, Y. Tong, S. Bing, J. Chen, Q. Ren, and Z. Zhang, "Improved snow geese algorithm for engineering applications and clustering optimization," *Scientific Reports*, vol.5, p.4506, 2025.

[30] S. Hochreiter, J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol.9, no.8, pp1735-1780, 1997.

[31] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, Daniel J. Inman, "1D convolutional neural networks and applications: A survey," *Mechanical Systems and Signal Processing*, vol.151, p. 107398, 2021.
[32] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv*: 1412.3555, 2014.
[33] C. Ma, X. Huang, Y. Zhao, T. Wang, B. Du, "GRU-LSTM model based on the SSA for short-term traffic flow prediction," *Journal of Intelligent and Connected Vehicles*, vol.8, no.1, p.9210051, 2025.