Short-term Passenger Flow Prediction of Urban Rail Transit Based on LSTM-KAN-Stacking Combination Model

Yuhan Du *, Jinbing Ha

Department of Economics and Management, Nanjing University of Science and Technology, Nanjing, China

* Corresponding Author Email: 1109646559@qq.com

Abstract. Urban rail transit passenger flow prediction is crucial for intelligent management and service of rail transit. Existing research has made some progress in improving passenger flow prediction accuracy, but there is still room for improvement in prediction accuracy. In addition, as the time granularity decreases, the complexity and uncertainty of passenger flow data increase, and the prediction accuracy also decreases. In order to further improve the short-term passenger flow prediction accuracy of rail transit, introduced the KAN network into the field of rail transit passenger flow prediction, and proposed a combined LSTM-KAN-Stacking model. The innovations of this combined model are mainly reflected in two aspects: first, KAN network is integrated into LSTM as a fully connected layer, and the input sequence data enters into KAN after extracting features by LSTM, which enhances the ability to capture complex patterns and improves the model prediction accuracy and robustness. Secondly, the deep learning model is combined with the Stacking integrated learning model, and the LSTM-KAN predicted output values are added to the original time series data as new features to provide richer information input for the Stacking model. By constructing the LSTM-KAN-Stacking model, feature stacking after predicting the original time series with LSTM-KAN model, constructing the integrated model Stacking and outputting the prediction after parameter optimization and relevance assessment, and finally conducting experimental comparative analysis based on Hangzhou metro passenger flow data, the validity of the proposed model is verified. The results show that the LSTM-KAN-Stacking model can reduce the MAE by at least 0.5%, 2.4%, and 5.6% compared with the nine base models, LSTM, LSTM-KAN, and Stacking model, respectively, on the 15-min, 30-min, and 60-min granularity datasets. In addition, the proposed model's goodness-of-fit at three temporal granularities can reach 0.972, 0.973, and 0.941, respectively. Visualization analysis verifies its generalization ability in different types of sites and different temporal patterns. Therefore, LSTM-KAN-Stacking has higher prediction accuracy compared with the traditional model, and KAN network has potential for urban rail transit short-term passenger flow prediction, especially at finer granularity, with high stability and applicability.

Keywords: Urban rail transit, Short-term passenger flow prediction, KAN model, Feature stacking, Combined prediction models.

1. Introduction

Urban rail transit is an important part of modern urban public transportation, undertaking a large number of passenger transportation tasks. In the context of accelerated urbanization and increased population density, effective prediction of short-term passenger flow at its stations is of great significance for optimizing operation scheduling and building smart cities [1].

There are numerous methods for short-term passenger flow forecasting. Traditional approaches primarily rely on statistical techniques, such as Moving Average (MA), Autoregressive Integrated Moving Average Model (ARIMA), and Kalman Filtering (KF) [2,3,4,5]. However, these methods struggle to capture the nonlinear characteristics and volatility of passenger flow data [6]. With the advancement of computing capabilities, machine learning models like Support Vector Regression (SVR) and Random Forest (RF) have been widely applied in passenger flow forecasting, leading to improved prediction accuracy [7,8,9]. Nevertheless, these models suffer from high memory consumption and low computational efficiency when processing large-scale data. In recent years, deep learning methods, particularly Recurrent Neural Networks (RNN) and Long Short-Term Memory Networks (LSTM), have garnered extensive attention due to their outstanding performance in handling sequential data [10,11,12]. Additionally, the attention mechanism and Convolutional Neural Networks (CNN) have also been employed for passenger flow forecasting tasks [13,14,15]. The limitations of single-model feature extraction may result in insufficient prediction accuracy; thus, ensemble models have attracted interest as they can integrate the advantages of different individual models [16,17,18].

Although existing studies have adopted various methods and achieved certain progress in prediction accuracy, there is still room for improvement in this aspect. Furthermore, as the time granularity decreases, the complexity and uncertainty of passenger flow data increase, leading to a prominent issue of declining prediction accuracy—an issue that existing research methods struggle to address effectively. The Kolmogorov-Arnold Networks (KAN) model, by virtue of its ability to learn more complex nonlinear relationships, exhibits unique advantages in function fitting and interpretability. This provides a new perspective for passenger flow forecasting and is expected to make up for the shortcomings of existing studies in mining fine-grained data. Based on this, an LSTM-KAN-Stacking model is proposed. The effectiveness of the proposed model is evaluated through experiments conducted on datasets with different time granularities.

2. Related work

2.1. Statistical Model

Research on urban rail transit passenger flow forecasting has a history of several decades, with traditional time series forecasting models being widely adopted in the early stages. For instance, the Moving Average (MA) model has been extensively used due to its simplicity and ease of understanding. Meng Pinchao et al. [2] have proven its advantages in short-term forecasting, such as high prediction accuracy and fast computation speed. However, the MA model has limited capability in forecasting non-stationary time series. Therefore, the Autoregressive Integrated Moving Average (ARIMA) model was introduced. It combines the Autoregressive Model (AR) and MA, and incorporates differencing to handle non-stationary sequences, thereby enabling effective forecasting. Zhang Guoyun et al. [3] and Zhao Xiaoli [19] both took a specific subway station as a case study, improved and optimized the ARIMA model, and verified its effectiveness in passenger flow forecasting. Based on the ARIMA model, Wu Zhenya [20] integrated the grey relational analysis model to forecast and conduct correlation analysis on the differentiated hub passenger flow data during the epidemic, confirming the reliability of the ARIMA model in passenger flow forecasting. Nevertheless, the ARIMA model remains inadequate when dealing with complex time series with obvious trends or seasonality. Williams and Hoel [21] applied the Seasonal Autoregressive Integrated Moving Average (SARIMA) algorithm to traffic forecasting; the seasonal extension makes this model more accurate than ARIMA [49]. In addition, studies in [22,23,24] have applied the SARIMA model to railway passenger flow forecasting, and all experiments have verified the excellent performance of SARIMA.

2.2. Machine Learning Model

In recent years, machine learning models have been applied to passenger flow forecasting, as they make up for the shortcomings of the strong assumptions (i.e., data is stationary or linear) inherent in statistical models [1]. For example, Sun Yuxing et al. [7] introduced the Support Vector Regression (SVR) model and optimized its parameters to improve the prediction accuracy of rail transit passengers' travel time.

To further enhance prediction performance, researchers have attempted to integrate multiple factors into the models. Zhou Jiazhong [9] comprehensively considered regional characteristic variables and station attribute variables, while Han Hao [25] incorporated network features into the time dimension to measure the spatial dimension; both approaches improved prediction accuracy.

In addition, multi-source data provides richer information dimensions and in-depth insight perspectives for passenger flow forecasting. Xu Xinyue et al. [8] took features such as environmental, demographic, economic, and station-related factors into account, and proposed a fusion model to capture the impact of built environment features on passenger flow. Cheng Y et al. [26] introduced a population migration index and effectively integrated relevant information to achieve improvements in subway passenger flow forecasting. Yao K et al. [27] predicted passenger flow based on multi-source data (e.g., number of employment positions, number of Points of Interest (POIs), etc.) and the Random Forest Regression (RFR) model. Experiments conducted on Qingdao Metro data confirmed the superiority of the proposed model.

2.3. Deep Learning Model

Deep learning possesses the capability of automatic feature extraction, enabling it to learn high-level abstract features from raw data without the need for manual design. This not only improves the efficiency and accuracy of passenger flow forecasting but also reduces the impact of human factors.

In the early stages, due to limitations in computing resources and data volume, deep learning saw limited application in time-series forecasting. As a fundamental deep learning model, neural networks were tentatively applied to simple time-series forecasting tasks [28,29,30,31]. Recurrent Neural Networks (RNN) was developed to handle long-term dependencies in time-series data; however, they suffer from gradient vanishing and explosion issues, which restrict their ability to learn from long sequences. To address this, Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRU) were proposed. LSTM can better handle long-term dependencies through their gating mechanism, while GRU—simplified variants of LSTM—reduce the number of parameters and computational complexity while maintaining predictive performance. Zhang Huizhen et al. [12] compared the short-term passenger flow forecasting capabilities of LSTM and GRU, and identified the time granularity division point at which GRU exhibit superior performance. Studies [10,32] combined LSTM with other models for passenger flow forecasting in different scenarios, leading to improved prediction accuracy.

Convolutional Neural Networks (CNN), originally designed for image recognition, have been extended to time-series forecasting in recent years. CNN excels at capturing local features and extract local patterns and trends from time-series data through convolution operations. Studies [15,33,34] have applied CNN to subway passenger flow forecasting, achieving effective predictions across multiple scenarios. With the advancement of research, the attention mechanism has been introduced into this field. Tang Zhengyi et al. [35] proposed the Adaptive Diffusion Graph Convolution Attention (ADGCA) network to explore the dynamic temporal and spatial correlations between stations. Wang Xueqin et al. [36] utilized Graph Convolutional Networks (GCN) and GRU integrated with the attention mechanism to extract spatiotemporal features of passenger flow. Ma Qian et al. [37] designed a sparse attention mechanism—all of these efforts have optimized subway passenger flow forecasting

2.4. Combined Model

To enhance the accuracy and stability of time-series forecasting, model combination methods have attracted widespread attention.

(1) Combinations of different deep learning models: For example, Zhao Liqiang et al. [34] combined Temporal Convolutional Networks (TCN) with LSTM.

(2) Combinations of machine learning and deep learning models: For instance, Wang Jinshui et al. [17] integrated Random Forests with GRU.

(3) Other types of combinations: Such as Zhao Mingwei et al. [38], who combined an improved particle swarm optimization algorithm with LSTM.

All these studies have confirmed that combined models can leverage the advantages of individual constituent models and improve forecasting performance.

The core idea of ensemble learning models is similar to that of combined models; therefore, many researchers have considered combining ensemble learning with other models, though most optimizations focus solely on the time series itself. For example, Liu Jie [39] applied Adaboost (an ensemble learning method) after performing wavelet decomposition on time series data; Shi Z et al. [40] integrated SVR after reconstructing data based on chaos theory; and Wang X et al. [41] used GRU as base models and integrated them with other machine learning models.

However, existing research on combined models involving ensemble learning has not treated ensemble learning as an independent forecasting unit. Although Wang X et al. [41] combined bagging (an ensemble learning method) with transfer learning, their work lacked parameter optimization for sub-models. Consequently, research on combining deep learning with ensemble learning (as a component of combined models) holds significant importance.

3. Problem Description

Subway inbound passenger flow forecasting can utilize historical time intervals to characterize the complex internal relationships of sequences through model training, thereby predicting the passenger flow volume in the next time interval. Taking the inbound passenger flow sequence as an example, the one-step forecasting method is adopted for passenger flow prediction and analysis.

Suppose at time t, the inbound passenger flow volume in the forecast time interval $[t, t+\Delta t]$ is denoted as $q_{[t,t+\Delta t]}$, where Δt represents the time interval. In this study, Δt is set to 15min、30min、60min[42,43]. Let $q_t = q_{[t,t+\Delta t]}$, then the following holds:

$$q_t = f\left(q_{t-1}, q_{t-2}, q_{t-3}, \ldots, q_{t-k}\right) \tag{1}$$

Where f denotes the mathematical mapping relationship, and k represents the time step. Therefore, the problem can be briefly described as follows: based on k time intervals $\{q_{t-k}, \ldots, q_{t-2}, q_{t-1}\}$ in the historical inbound passenger flow sequence of a specific subway station, a model is used to predict the inbound passenger flow volume q_t in the future time interval.

4. Architecture of the LSTM-KAN-Stacking Model

4.1. The LSTM Model

Proposed by Hochreiter and Schmidhuber in 1997, the Long Short-Term Memory network addresses the gradient vanishing problem of traditional Recurrent Neural Networks through a gating mechanism, making it stable and effective in processing long-sequence data.

The LSTM network consists of an input layer, memory cells, and an output layer. The memory cell is its distinctive structure, comprising a hidden state $\boldsymbol{h}$ (responsible for short-term information) and a cell state $\boldsymbol{C}$ (responsible for long-term memory); both states iterate with the time step t. Inside each memory cell, there are three pathways corresponding to the forget gate $\boldsymbol{f}_t$、input gate $\boldsymbol{i}_t$ and output gate $\boldsymbol{O}_t$ (Fig. 1).

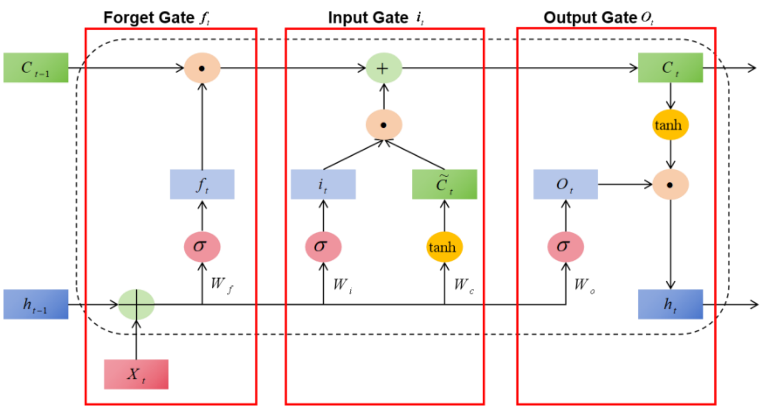

Fig 1. Flow Diagram of the Memory Cell Interior

The forget gate filters out old information by calculating a retention ratio based on the input information $\boldsymbol{X}_t$ at the current time step and the short-term information $\boldsymbol{h}_{t-1}$ from the previous time step. Its calculation formula is:

$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + \boldsymbol{b}_f\right) \tag{2}$$

The input gate specifies which new information is stored in the cell state. It consists of two components: first, the input weight is calculated via the sigmoid layer (Formula 3); second, a new candidate value vector is generated (Formula 4).

$$\boldsymbol{i}_t = \sigma(\boldsymbol{W}_i \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + \boldsymbol{b}_i) \tag{3}$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{W}_c \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + \boldsymbol{b}_c) \tag{4}$$

Subsequently, the cell state is updated based on the results of the forget gate and the input gate (Formula 5).

$$\boldsymbol{C}_t = \boldsymbol{f}_t * \boldsymbol{C}_{t-1} + \boldsymbol{i}_t * \tilde{\boldsymbol{C}}_t \tag{5}$$

The output gate calculates the filtering ratio $\boldsymbol{O}_t$ using contextual information and weights $\boldsymbol{W}_o$ (Formula 6), then filters the new long-term information $\boldsymbol{C}_t$ to extract short-term information $\boldsymbol{h}_t$ suitable for the current time step, which is used for prediction (Formula 7):

$$\boldsymbol{O}_t = \sigma(\boldsymbol{W}_o \cdot [\boldsymbol{h}_{t-1}, \boldsymbol{X}_t] + \boldsymbol{b}_o) \tag{6}$$

$$\boldsymbol{h}_t = \boldsymbol{O}_t * \tanh(\boldsymbol{C}_t) \tag{7}$$

In the above Formulas, σ denotes the sigmoid function; $\tanh$ represents the hyperbolic tangent function; $\boldsymbol{W}$ denotes the weight parameter, and $\boldsymbol{b}$ denotes the bias term.

4.2. Structure and Theoretical Foundation of KAN Networks

Kolmogorov-Arnold Networks (KAN) [44] is a new type of neural network architecture inspired by the Kolmogorov-Arnold representation theorem. This theorem states that a continuous multivariate function can be expressed as a combination of univariate functions.

The core of KAN networks lies in replacing weight parameters with learnable univariate functions (usually parameterized by splines). Each input variable is processed through a set of univariate functions, enabling the network to handle multi-input to multi-output mappings. The architecture of the KAN network is shown in Fig. 2.

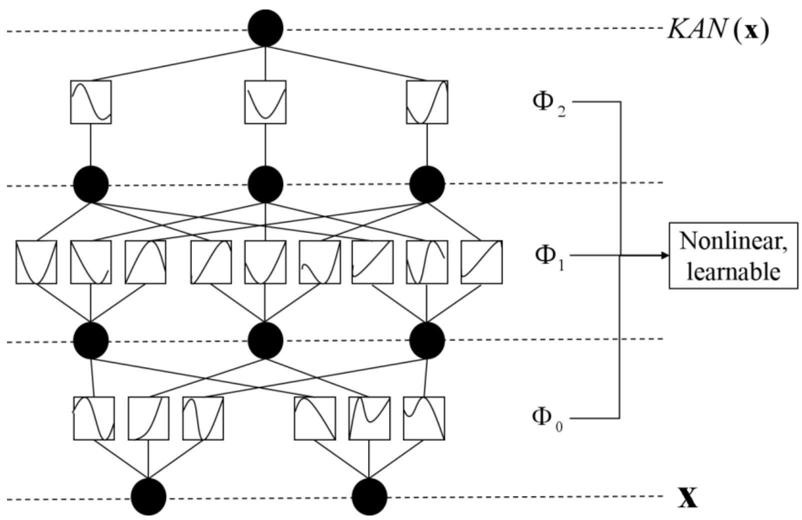

Fig 2. Deep KAN Model Architecture

In the figure, Φ represents a nonlinear, learnable activation function, and $\boldsymbol{x}$ denotes the input variable. For a KAN network with L layers of activation functions, the calculation formula of $KAN(\boldsymbol{x})$ is as follows:

$$KAN(\boldsymbol{x}) = \Phi_{L-1} \circ \Phi_{L-2} \circ \ldots \circ \Phi_1 \circ \Phi_0(\boldsymbol{x}) \quad (8)$$

As can be seen from Fig. 2, KAN places activation functions on the weights. Nodes only simply sum the input signals without applying any nonlinearity. This design makes KAN more flexible in processing complex data—it can automatically adjust its shape according to different data to minimize approximation errors, thereby enhancing the network's ability to learn subtle patterns from high-dimensional datasets.

4.3. The Stacking Model

Ensemble learning demonstrates significant advantages in passenger flow forecasting: it integrates the strengths of different models, captures complex patterns, reduces the risk of overfitting, and balances bias and variance to enhance generalization ability.

The Stacking algorithm was proposed by Wolpert [45] in 1992 and has since been widely applied [46]. As an ensemble learning method, its training process consists of two stages:

(1) Base model training and prediction via K-fold cross-validation: Different algorithms are used as base models, which are trained and validated using K-fold cross-validation. Each base model generates prediction results on the validation subsets.

(2) Meta-model training: The prediction outputs of all base models are combined into a new feature matrix. This matrix, together with the actual labels, forms the training data for the second-layer model (meta-model). The meta-model is then trained on this data to integrate the prediction results of the base models, thereby improving prediction accuracy and generalization ability.

This two-stage training approach can effectively capture the complementary advantages of different base models and enhance the overall performance of the forecasting system [47].

4.4. The LSTM-KAN-Stacking Model

The overall framework of the proposed LSTM-KAN-Stacking model is illustrated in Fig. 3. The model architecture mainly consists of three stages: LSTM-KAN prediction, feature stacking, and Stacking ensemble for prediction output.

In the LSTM-KAN module, the KAN network is integrated into LSTM as a fully connected layer. Input sequence data undergoes feature extraction via LSTM before being fed into KAN. This integration enhances the capability to capture complex patterns, thereby improving the model's prediction accuracy and robustness.

In the feature stacking stage, the prediction outputs of the LSTM-KAN module are treated as new features and appended to the original time-series data. This process provides richer information input for the subsequent Stacking model.

In the Stacking ensemble stage, various machine learning models are trained and evaluated. Base models are combined and assessed according to specific criteria, and the optimal combination is selected for Stacking ensemble to generate the final prediction output.

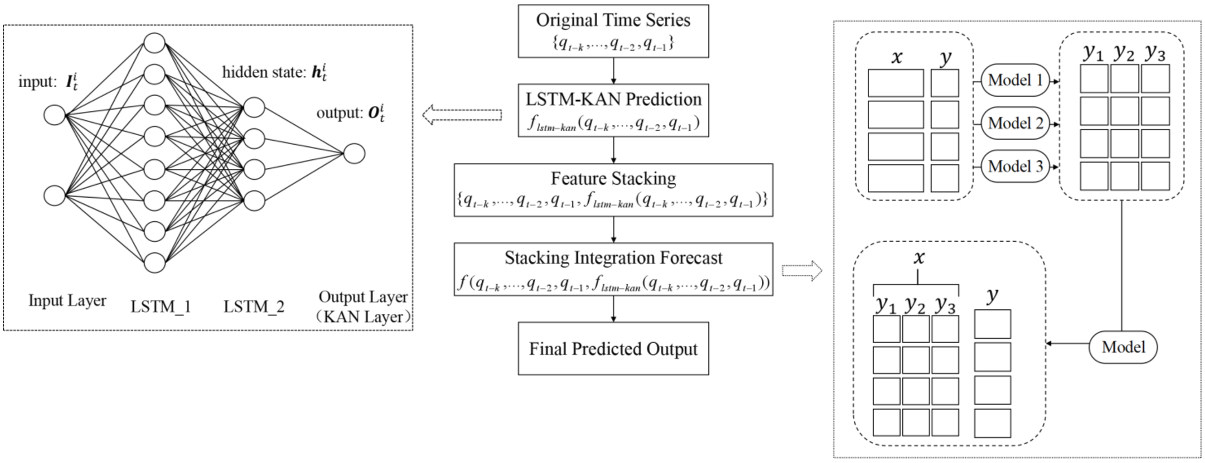

Fig 3. Overall Architecture of the LSTM-KAN-Stacking Model

5. Experiments

5.1. Dataset

To verify the performance of the LSTM-KAN-Stacking model in subway passenger flow forecasting, experiments were conducted using Hangzhou Metro passenger flow data. The data covers the inbound and outbound passenger information of some subway stations and lines in Hangzhou from January 1 to 25, 2019. Taking Qianjiang Road Station as an example, a short-term inbound passenger flow forecasting model was constructed, and the inbound passenger flow data of this station was extracted.

Considering the characteristics of passenger flow, holiday passenger flow data were removed. The operating time range (6:00–24:00) was determined, and data outside this time range were excluded. The final data format is shown in Table 1, where "period_start" represents the start time of each time interval, "in_count" denotes the inbound passenger flow volume during that interval, and "week" indicates the day of the week.

Table 1. Sample Data of Inbound Passenger Flow at Qingdao North Station _15min

period_start	in_count	week
2022-01-02 06:00:00	6	Wednesday
2022-01-02 06:15:00	27	Wednesday
2022-01-02 06:30:00	61	Wednesday
2022-01-02 06:45:00	83	Wednesday
2022-01-02 07:00:00	126	Wednesday

In this study, the sliding window method was used to generate sample data. Specifically, the model utilizes T_h steps of historical observations to predict the data of the next T_p steps [48]. For the selection of time steps, the appropriate number of time steps was determined using the Autocorrelation Function (ACF) and Partial Autocorrelation Function (PACF). Considering that a larger number of time steps may prevent the model from capturing important short-term changes, the smallest time step among all possible selections was identified as the final time step. Eventually, $T_h^{15}=6, T_h^{30}=4, T_h^{60}=2, T_p=1$.

The model dataset was split into a training set and a test set at an 8:2 ratio. The traffic flow values were preprocessed using the StandardScaler normalization method.

5.2. Evaluation Metrics

Multi-metric assessment is commonly used to measure the performance of passenger flow forecasting models. Three metrics—Mean Absolute Error E_{ma} 、 Root Mean Squared Error E_{rms}

and Coefficient of Determination R^2 —were selected to evaluate the performance of the short-term passenger flow forecasting model. The calculation formulas for these three metrics are as follows:

$$E_{ma} = \frac{1}{N}\sum_{i=1}^{N}\left|y_i - y_i\right| \tag{9}$$

$$E_{rms} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_i\right)^2} \tag{10}$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2} \tag{11}$$

Among the formulas, y represents the actual value, $\hat{y}$ represents the predicted value, $\bar{y}$ is the average of all actual values, and N is the total number of samples.

The first two metrics are used to measure the error magnitude between predicted values and actual values; smaller values indicate better performance. The coefficient of determination R^2 evaluates the model’s goodness of fit and ability to explain data variability; a larger value indicates better performance.

5.3. Experimental Setup

In the experiment, the model was implemented in Python 3.8.18 using the PyTorch framework, and the Adam optimizer was adopted for training.

5.3.1 Construction of the Stacking Model

For a Stacking ensemble model, the base models should be strong regression models with distinct characteristics to meet the requirements of "diversity in numbers" and "diversity in quality"—this ensures the ensemble model achieves superior performance [49,50]. Drawing on the design methods for Stacking models in the literature [51,52,53,54], the following machine learning models were selected as candidate base models: Support Vector Regression (SVR), K-Nearest Neighbors (KNN), Decision Tree (DT), Lasso Regression (LR), Random Forest (RF), Gradient Boosting Decision Tree (GBDT), Extreme Gradient Boosting (XGBoost), Light Gradient Boosting Machine (LightGBM), and Multilayer Perceptron (MLP). The optimal hyperparameters for each model were determined using 5-fold cross-validation combined with grid search, with RMSE as the evaluation metric.

For the dataset with 15-minute temporal granularity, the optimal hyperparameters of each candidate model are listed in Table 2. A threshold of $R^2>0.96$ was set for base classifiers to exclude "poor-performing models," resulting in the elimination of DT and Lasso Regression. The diversity of base models was analyzed by evaluating their correlation using the Pearson correlation coefficient, with the correlation heatmap shown in Fig. 4.

A correlation coefficient threshold of less than 0.997 was adopted as the selection criterion for base classifiers to meet the "diversity" requirement, and the number of base classifiers was set to be no less than 3 to satisfy the "sufficiency" requirement. Eventually, 17 Stacking base model combinations were obtained. In the second layer of Stacking, Linear Regression (LR) was selected as the meta-model to avoid overfitting. The 17 combinations were evaluated, with results presented in Table 3, and the best-performing combination was chosen as the final Stacking model.

For datasets with time granularities of 30 minutes and 60 minutes, the same experimental procedure as applied to the 15-minute granularity dataset was followed: determining the optimal parameters for each base model, identifying all possible model combinations that meet the requirements of "good performance", "diversity" and "sufficiency" for base model combinations, and selecting the optimal model through comparison.

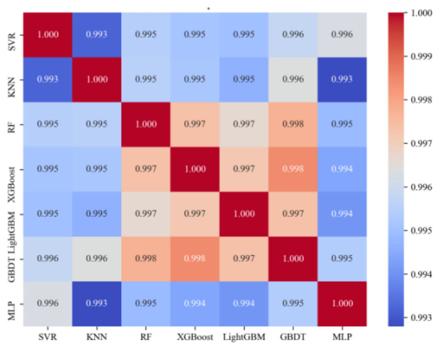

Fig 4. Similarity Analysis Between Different Base Classifiers

Table 2. Optimal Hyperparameter Settings for the Base Classifier

base classifier	optimal hyperparameter value
SVR	C=10, epsilon=0.1, gamma= 'auto', kernel='rbf'
KNN	n_neighbors=7
DT	criterion='friedman_mse',max_depth=10, max_features='sqrt', min_samples_leaf=1, min_samples_split=10
Lasso	alpha=0.001, max_iter=1000
RF	max_depth=7, min_samples_leaf=1, min_samples_split=2, n_estimators=1000
GBDT	learning_rate=0.01, max_depth=5, max_features='sqrt', min_samples_leaf=1, min_samples_split=10, n_estimators=500
XGBoost	colsample_bytree=0.8, learning_rate=0.01, max_depth=5, n_estimators=1000, subsample=0.8
LightGBM	learning_rate=0.1, max_depth=3, min_data_in_leaf=20, n_estimators=500, num_leaves=31, reg_alpha=0.5, reg_lambda=0.5
MLP	alpha=0.0001, hidden_layer_sizes=(100,)

Table 3. Comparison of Prediction Effects of Different Stacking Models

Model Name	Combination of Base Classifiers	Assessment of Indicators		
		R^2	E_{ma}	E_{rms}
ML1	SVR+KNN+RF	0.97	29.07	38.19
ML2	SVR+KNN+XGBoost	0.97	28.79	37.97
ML3	SVR+KNN+LightGBM	0.97	29.01	38.31
ML4	SVR+KNN+GBDT	0.97	28.87	37.96
ML5	SVR+KNN+MLP	0.97	29.29	38.3
ML6	SVR+RF+MLP	0.97	30.5	39.67
ML7	SVR+XGBoost+MLP	0.97	29.31	38.98
ML8	SVR+LightGBM+MLP	0.97	30.4	40.09
ML9	SVR+GBDT+MLP	0.97	29.61	38.96
ML10	KNN+RF+MLP	0.97	28.82	37.81
ML11	**KNN+XGBoost+MLP**	**0.97**	**28.64**	**37.71**
ML12	KNN+LightGBM+MLP	0.97	28.86	38.1
ML13	KNN+GBDT+MLP	0.97	28.75	37.79
ML14	SVR+KNN+RF+MLP	0.97	29.13	38.33
ML15	SVR+KNN+XGBoost+MLP	0.97	28.84	38.12
ML16	SVR+KNN+LightGBM+MLP	0.97	29.1	38.5
ML17	SVR+KNN+GBDT+MLP	0.97	28.97	38.21

As can be seen from Table 3, the optimal Stacking model is ML11, which has the smallest E_{rms}, and also performs best in terms of E_{ma} and R^2 metrics. Therefore, KNN, XGBoost, and MLP were selected as base models, and linear regression was chosen as the meta-model to construct the Stacking model.

5.3.2 Parameter Setting of the LSTM-KAN Model

Before implementing the LSTM-KAN-Stacking combination, the optimal hyperparameters of the LSTM-KAN model were determined, and the details are presented in Table 4.

Table 4. Optimal Hyperparameters of LSTM-KAN

Hyperparameter	batch_size	epochs	hidden_size	learning_rate	Num_layers
Optimum value	16	100	128	0.01	3

5.4. Performance Evaluation of the LSTM-KAN-Stacking Model

5.4.1 Comparative Analysis of Different Time Granularities

Under different time granularities, the performance of the LSTM-KAN-Stacking combined model was compared with that of all base models, LSTM, LSTM-KAN, and Stacking. The results are shown in Table 5.

Table 5. Performance Comparison Results of the LSTM-KAN-Stacking Model

Model Name	Indicators _15min			Indicators _30min			Indicators _60min		
	R^2	E_{ma}	E_{rms}	R^2	E_{ma}	E_{rms}	R^2	E_{ma}	E_{rms}
SVR	0.963	33.910	42.698	0.968	55.440	78.288	0.927	133.384	221.927
KNN	0.969	29.401	39.155	0.967	53.415	79.388	0.932	129.064	214.357
DT	0.938	37.289	55.242	0.887	88.831	146.730	0.832	205.567	336.890
Lasso	0.934	44.811	57.015	0.861	120.071	162.414	0.460	450.931	604.154
RF	0.967	30.418	40.124	0.950	64.619	97.715	0.914	155.505	240.666
GBDT	0.970	29.601	38.618	0.960	65.507	87.417	0.896	160.094	264.912
XGBoost	0.968	29.293	39.404	0.958	63.581	89.291	0.906	157.798	251.527
LightGBM	0.964	31.153	42.177	0.952	69.516	95.471	0.906	160.572	252.335
MLP	0.968	31.190	39.502	0.958	68.756	89.229	0.912	183.011	243.601
LSTM	0.967	30.470	40.130	0.968	57.926	77.959	0.933	145.440	212.911
LSTM-KAN	0.969	29.770	38.920	0.969	58.027	76.780	0.938	130.654	**199.375**
Stacking	0.971	28.639	37.714	0.969	54.865	76.513	0.928	133.371	204.875
LSTM-KAN-Stacking	**0.972**	**28.490**	**37.226**	**0.973**	**52.126**	**71.715**	**0.941**	**121.779**	220.764

In summary, the LSTM-KAN-Stacking model has obvious advantages in passenger flow forecasting, and the KAN network shows great potential in short-term passenger flow forecasting for rail transit, especially for passenger flow with finer granularities.

As can be seen from the table, across datasets of all granularities, the LSTM-KAN-Stacking model achieves the highest R^2 values compared to other models, which are 0.972, 0.973, and 0.941 respectively. The E_{ma} is the lowest, with reductions of 0.5%, 2.4%, and 5.6% respectively. For E_{rms}, the model also achieves the lowest values on the datasets of the first two time granularities, with reductions of 1.3% and 6.3% respectively. However, on the 60-minute granularity dataset, its E_{rms} is relatively high (220.764). The reason for this lies in the calculation mechanism of RMSE: after squaring the errors and then taking the square root, RMSE amplifies the impact of large errors. The model performs well in terms of overall fitting (with an R^2 of 0.929, which is at the highest level) and average deviation (with the lowest E_{ma}). However, there may be individual large error values, which prevent the E_{rms} from reaching the lowest. Nevertheless, from the perspective of overall evaluation metrics, the model still has certain advantages over other models, confirming its effectiveness and stability.

From the performance comparison between the LSTM and LSTM-KAN models, the LSTM-KAN model outperforms the LSTM model in almost all evaluation metrics. This highlights the positive impact of integrating the KAN network into LSTM and proves the potential of the KAN network in short-term passenger flow forecasting for rail transit systems.

From the perspective of time granularity division, the proposed LSTM-KAN-Stacking model performs better on datasets with finer granularities (15 minutes and 30 minutes). Specifically, its R^2 on these finer-grained datasets is significantly higher than that on the 60-minute granularity dataset. This result proves that the integration of the KAN network enhances the model's ability to capture complex non-linear relationships in passenger flow data. It enables the model to more accurately capture subtle changes and short-term fluctuation characteristics of subway passenger flow, thereby better adapting to the complex patterns inherent in such data.

In summary, the LSTM-KAN-Stacking model demonstrates distinct advantages in passenger flow forecasting. Additionally, the KAN network shows considerable potential for short-term passenger flow forecasting in rail transit systems, particularly for passenger flow data with finer temporal granularities.

5.4.2 Comparative Analysis of Different Types of Stations

Passenger flow patterns vary across different types of subway stations. Using a clustering method, 80 subway stations in Hangzhou were categorized into three types: core commuter hub stations, suburban commuter stations, and comprehensive regional service stations.

To evaluate the prediction performance of the LSTM-KAN-Stacking model across these station types, one representative station was selected from each category (denoted as Station A, Station B, and Station C respectively) for experimental analysis. Figure 5 shows the fitting results between the predicted and actual passenger flow values of the model at Stations A, B, and C under both weekday and weekend conditions.

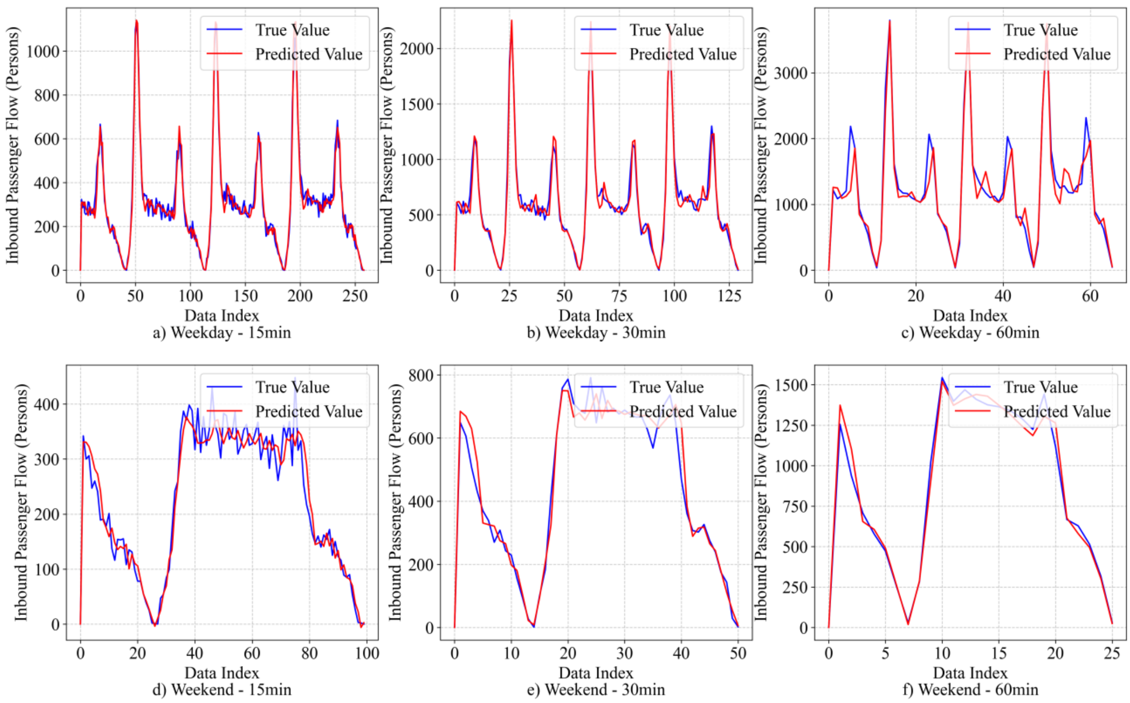

Fig 5. Prediction Fitting Plots of the LSTM-KAN-Stacking Model in Weekday and Weekend Scenarios

As can be seen from the fitting plots, the LSTM-KAN-Stacking model exhibits excellent fitting performance across multiple station types. This indicates that the model can effectively adapt to the passenger flow characteristics of different stations, demonstrating high stability and applicability.

Meanwhile, the model performs well in terms of fitting both on weekdays and weekends, which further verifies its generalization ability under different temporal patterns. Such consistent performance in heterogeneous scenarios (varying station types and temporal patterns) proves that the LSTM-KAN-Stacking model can effectively cope with the complexity and diversity of passenger flow data, providing a reliable model option for passenger flow forecasting in practical applications.

6. Conclusion and Outlook

1) To address the issue of prediction accuracy for short-term passenger flow in urban rail transit, an LSTM-KAN-Stacking hybrid model is proposed. Different from general hybrid models, this model combines deep learning and ensemble learning while integrating the Kolmogorov-Arnold Network (KAN), thereby improving prediction accuracy.

2) The proposed model integrates the KAN network into LSTM as a fully connected layer. Specifically, the input sequence data first undergoes feature extraction by LSTM, and the extracted features are then fed into the KAN—this enhances the model's ability to capture complex patterns. Furthermore, the predicted output of the LSTM-KAN sub-model is treated as a new feature and added to the original time-series data, providing richer information input for the subsequent Stacking model.

3) Experiments were conducted using subway passenger flow data from Hangzhou, with time granularities set to 15 minutes, 30 minutes, and 60 minutes. Comparative experiments with multiple baseline models show that the proposed model achieves higher prediction accuracy than traditional models; meanwhile, the KAN network demonstrates significant potential in short-term subway passenger flow forecasting, especially for forecasts with finer time granularities. Visualization analysis further confirms that the proposed model can effectively handle the complexity and diversity of passenger flow data: it maintains high prediction accuracy across different station types and passenger flow patterns, thereby providing a reliable model option for passenger flow forecasting in practical applications.

Passenger flow is influenced by factors in the built environment, such as land use and network attributes. Therefore, future work can incorporate additional features (e.g., geographic location) to enable passenger flow forecasting at a higher dimensional level.

Reference

[1] CHUWANG D D, CHEN Weiya, ZHONG Ming. Short-term Urban Rail Transit Passenger Flow Forecasting Based on Fusion Model Methods Using Univariate Time Series[J]. Applied Soft Computing, 2023, 147: 110740.

[2] ZHU Haining. N days average volume based ARIMA forecasting model for Shanghai metro passenger flow[C]//2010 International Conference on Artificial Intelligence and Education (ICAIE). IEEE, 2010: 631-635.

[3] ZHANG Guoyun, JIN Hui. Research on Short-term Passenger Flow Prediction of Urban Rail Transit Based on Improved ARIMA Model[J]. Computer Applications and Software,2022,39(01):339-344.

[4] OKUTANI I, STEPHANEDES Y J. Dynamic prediction of traffic volume through Kalman filtering theory[J]. Transportation Research Part B: Methodological, 1984, 18 (1): 1−11.

[5] DA SILVA D, SHALABY A. Forecasting short-term subway passenger flow using Wi-Fi data: comparative analysis of advanced time-series methods[J]. Journal of Intelligent Transportation Systems, 2024: 1-21.

[6] LI Shuqing, LI Wei, LIU Yaohong, et al. Short-term Passenger Flow Prediction Model of Rail Transit Based on Combined Deep Learning[J]. Journal of Chongqing Jiaotong University (Natural Science Edition),2024,43(02):92-99.

[7] LI Chuan, WANG Xiaodan, CHENG Zhiwei, et al. Forecasting bus passenger flows by using a clustering-based support vector regression approach[J]. Ieee Access, 2020, 8: 19717-19725.

[8] XU Xinyue, KONG Qingxue, LI Jianmin, et al. Analysis of the Spatiotemporal Heterogeneity of the Built Environment on the Passenger Flow of Rail Transit[J]. Journal of Transportation Systems Engineering and Information,2023,23(04):194-202+281.

[9] SHI Zhi, ZHANG Na, SCHONFELD P M, et al. Short-term metro passenger flow forecasting using ensemble-chaos support vector regression[J]. Transportmetrica A: Transport Science, 2020, 16(2): 194-212.

[10] CHEN Quanchao, WEN Di, LI Xuqiang, et al. Empirical mode decomposition based long short-term memory neural network forecasting model for the short-term metro passenger flow[J]. PloS one, 2019, 14(9): e0222365.

[11] ZHANG Guoyun, JIN Hui. Passenger Flow Prediction of Urban Rail Transit Station Based on Improved PSO-LSTM Model[J]. Journal of Computer Applications and Software,2021,38(12):110-114+134.

[12] ZHANG Huizhen, GAO Zhengkai, LI Jianqiang, et al. Short-term Passenger Flow Prediction of Urban Rail Transit Based on Recurrent Neural Network[J]. Journal of Jilin University (Engineering Science),2023,53(02):430-438.

[13] LI Xin, ZHANG Xu, YU Le'an, et al. Research on Short-term Passenger Flow Prediction of Scenic Area Based on Improved Transformer Model[J]. Chinese Management Science, 2025,33(02):105-117.

[14] ZHANG Xinyu, SHI Tianyun, LI Haoguang, et al. Collaborative prediction of passenger flow at multiple entry gates of a large passenger station based on CLTAttention[J]. Journal of Railway Science and Engineering, 1-13.

[15] TANG Zhengyi, HUANG Jiahuan, WANG Jinshui, et al. Passenger flow prediction in subway based on adaptive diffusion graph convolutional attention network[J]. Journal of Railway Science and Engineering, 2024, 21 (12): 4910-4923.

[16] TANG Jiqiang, YANG Luqi, YANG Wu. Prediction of Short-term Passenger Flow Starts and Ends of Rail Transit Based on Long Short-Term Memory Network and Graph Convolutional Network[J]. Journal of Chongqing University,2022,45(11):91-99.

[17] WANG Jinshui, OU Xuewen, CHEN Junyan, et al. Classification and Prediction of Short-term Passenger Flow of Urban Rail Stations Based on Combined Model[J]. Journal of Railway Science and Engineering,2023,20(06):2004-2012.

[18] YANG Jing, ZHU Jingwei, LIU Bo, et al. Short-term Passenger Flow Prediction of Urban Rail Transit Based on Combined Model[J]. Journal of Transportation Systems Engineering and Information Technology,2019,19(03):119-125.

[19] Zhao Xiaoli. Research on the application of ARIMA model in short-term passenger flow prediction of urban rail transit[J]. Modern Urban Rail Transit,2023, (08):77-82.

[20] Wu Zhenya. Data Analysis and Prediction of Passenger Flow of Shanghai Hongqiao Hub Based on ARIMA Model[J]. Traffic & Transportation,2022,35(S1):315-320)

[21] Williams B M, Hoel L A. Modeling and Forecasting Vehicular Traffic Flow as a Seasonal ARIMA Process: Theoretical Basis and Empirical Results[J]. Journal of Transportation Engineering, 2003, 129(6): 664-672.

[22] Li Jie,Peng Qiyuan,Yang Yuxiang. Passenger Flow Prediction of Guangzhou-Zhuhai Intercity Railway Based on SARIMA Model[J]. Journal of Southwest Jiaotong University,2020,55(01):41-51.

[23] Duan Ran,Pang Jianhua,Zhang Liangjun. Research on Passenger Flow Prediction of Railway Station Based on SARIMA Model[J]. Practice and Understanding of Mathematics,2019,49(09):1-10.

[24] Milenković M, Švadlenka L, Melichar V, et al.SARIMA Modelling Approach for Railway Passenger Flow Forecasting[J]. Transport,2015,1-8.

[25] Han Hao, Xu Sheng'an, Zhao Meng. Short - Term Passenger Flow Prediction Model of Rail Transit Based on LightGBM Considering Network Structure[J]. Railway Transport and Economy, 2021, 43(10): 109 - 117.

[26] Cheng Y, Li H, Sun S, et al.Short-term Subway Passenger Flow Forecasting Approach Based on Multi-source Data Fusion[J]. Information Sciences,2024,679121109-121109.

[27] Yao K, Gao G, Liu Y, et al. A Stable Passenger Flow Forecast Approach for Newly Opened Metro Stations Based on Multi-Source Data and Random Forest Regression Model[C]//2022 3rd International Conference on Intelligent Design (ICID). IEEE, 2022: 249-254.

[28] Yang Xinmiao, Wang Wei, Gu Weiping, et al. Fuzzy Neural Network Prediction Model for Passenger Flow of Bus Lines[J]. Journal of Highway and Transportation Research and Development, 2000, (04):38 - 40 + 46.

[29] Jiang Ping, Shi Qin, Chen Wuwei, et al. Bus Passenger Flow Prediction Based on Elman - type Recurrent Neural Network[J]. Journal of Hefei University of Technology (Natural Science), 2008, (03):340 - 342 + 369.

[30] Zou Dong, Liu Qiong, Huang Zirong. Research on Passenger Flow Prediction of Urban Rail Transit Network Considering Spatio-temporal Correlation[J]. Urban Mass Transit, 2016, 19(03):32 - 37.

[31] Li Chunxiao, Li Haiying, Jiang Xi, et al. Short-Term Station Inbound Passenger Flow Prediction Based on Generalized Dynamic Fuzzy Neural Network[J]. Urban Rapid Rail Transit, 2015, 28(04):57 - 61.

[32] Wang Jinfeng, Sun Lianying, Zhang Tian, et al. Short-Term Passenger Flow Prediction of Urban Rail Transit Based on K-LSTM-ecm Model[J]. Manufacturing Automation, 2022, 44(05):103 - 107 + 133.

[33] Yang Jing, Dai Shengxu, Zhang Hongliang, et al. Short-Term Inbound Passenger Flow Prediction of Subway Station During the End of Large-Scale Events[J]. Science Technology and Engineering, 2021, 21(05):2042 - 2048.

[34] Zhao Liqiang, Li Ruisen, Tang Shuixiong, et al. Short-Term Passenger Flow Prediction of Urban Rail Transit Based on TCN-LSTM with Multidimensional Predictable Features[J]. Journal of Beijing University of Chemical Technology (Natural Science Edition), 2024, 51(05):86 - 96.

[35] Tang Zhengyi, Huang Jiahuan, Wang Jinshui, et al. Subway Passenger Flow Prediction Based on Adaptive Diffusion Graph Convolutional Attention Network[J]. Journal of Railway Science and Engineering, 1 - 13.

[36] Wang Xueqin, Xu Xinyue, Wu Yuankai, et al. Short-Term Passenger Flow Prediction of Urban Rail Transit Based on Hybrid Deep Learning Model[J]. Journal of Railway Science and Engineering, 2022, 19(12):3557 - 3568.

[37] Ma Qian, Liang Yi, Duan Yi, et al. Metro Passenger Flow Prediction Model Based on Dynamic Sparse Attention[J]. Urban Mass Transit, 2022, 25(04):22 - 26.

[38] Zhao Mingwei, Zhang Wensheng. Short - term Passenger Flow Prediction of Urban Rail Transit Based on IPSO - LSTM Combined Model[J]. Railway Transport and Economy, 2022, 44(02):123 - 130.

[39] Liu Jie. Research on Short-Term Prediction Model of Railway Passenger Flow Based on Ensemble Algorithm[J]. Journal of Chongqing Jiaotong University (Natural Science), 2022, 41(05):20 - 25.

[40] Shi Z, Zhang N, Schonfeld P M, et al. Short-term Metro Passenger Flow Forecasting Using Ensemble-chaos Support Vector Regression[J]. Transportmetrica A: Transport Science, 2020, 16(2): 194-212.

[41] Wang X, Zhu C, Jiang J. A Deep Learning and Ensemble Learning Based Architecture for Metro Passenger Flow Forecast[J]. IET Intelligent Transport Systems, 2023, 17(3): 487-502.

[42] DING Xinyu, SHI Yan. Prediction of Short-term Traffic Flow in Road Network Based on Deep Spatiotemporal Residual Network[J]. Computer Engineering and Design,2022,43(02):316-322.

[43] ZENG Lu, JIANG Zixuan, PENG Dongliang, et al. Passenger flow prediction of urban rail transit stations based on adaptive spatio-temporal hierarchical network[J]. Journal of Railway Science and Engineering, 1-13.

[44] LIU Ziming, WANG Yixuan, VAIDYA S, et al. Kan: Kolmogorov-arnold networks[J]. arXiv preprint arXiv:2404.19756, 2024.

[45] WOLPERT D H. Stacked Generalization[J]. Neural networks, 1992, 5(2): 241-259.

[46] YANG Anan, HAN Xingyu, TIAN Kuang, et al. Research on Short-Term Passenger Flow Prediction of Airport Line Based on Stacking Ensemble Learning[J]. Shandong Science, 2024, 37(04):112 - 120.

[47] CHEN Peng, FU Xin, WANG Xiang. A Graph Convolutional Stacked Bidirectional Unidirectional-LSTM Neural Network for Metro Ridership Prediction[J]. IEEE Transactions on Intelligent Transportation Systems, 2021, 23(7): 6950-6962

[48] WANG Zitong, LI Chenliang. Traffic Flow Prediction Based on Location-Aware Spatio-Temporal Graph Convolutional Network[J]. Data Analysis and Knowledge Discovery, 2025,9(02):94-105.

[49] LI Beibei, CHEN Chunxiao, ZHENG Liaomo, et al. Tool Wear Prediction Based on FIR-Stacking[J]. Modular Machine Tool & Automatic Manufacturing Technique, 2024(4):87 - 91.

[50] ZHAO Youlin, PANG Hangyuan, SHI Yanqing. Construction of Health Information Portrait and Identification of False Health Information: Integrating Social Sensing Data and Publisher's Prior Knowledge[J]. Documentation, Information & Knowledge, 2024,41(06):141-154+165.

[51] XIE Qianwei, XUE Fengchang, CHEN Jianfei. Integrated Learning Prediction Model for Rape Flowering Period Combined with Virtual Sample Generation[J]. Transactions of the Chinese Society of Agricultural Engineering, 2024,40(19):159-167.

[52] CHEN Fei, CHEN Zhenjie, LI Feixue, et al. Spatio-temporal Prediction Stacking-GWR Model of Urban Land Price[J]. Geography and Geo-Information Science, 2024, 40(05):1-10.

[53] GUO Zhiqiang, ZHANG Botao, ZENG Yunliu. Research on Kiwifruit Sugar Content Detection by Near-Infrared Spectroscopy Combined with Stacking Ensemble Learning[J]. Spectroscopy and Spectral Analysis, 2024, 44(10):2932-2940.

[54] WANG Xiaoming, XU Bin, YIN Yuanya, et al. Wind Power Prediction Method Based on Spatial Correlation and Stacking Ensemble Learning[J]. Electric Power Engineering Technology, 2024, 43(05):224-232.