Traffic & Transportation

A Passenger Flow Prediction Model of Urban Rail Transit Based on ICEEMDAN Decomposition and TCN-LSTM-CAM Module

Haijun LI[1], Yuheng ZHANG[2], Jing CHEN[3], Yan HUANG[4], Dehua WEI[5]

Original Scientific Paper
Submitted: 25 Feb 2025
Accepted: 6 June 2025
Published: 29 June 2026

[1] lihaijun@mail.lzjtu.cn, School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou, China; Key Laboratory of Railway Industry on Plateau Railway Transportation Intelligent Management and Control, Lanzhou Jiaotong University, Lanzhou, China; Wuwei Vocational College, Wuwei, China
[2] Corresponding author, Zhangyuheng_2001@163.com, School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou, China; China Railway Hohhot Bureau Group Co., Ltd., Hohhot, China
[3] chenjing_021002@163.com, School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou, China
[4] yanhuang@mail.lzjtu.cn, School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou, China; Key Laboratory of Railway Industry on Plateau Railway Transportation Intelligent Management and Control, Lanzhou Jiaotong University, Lanzhou, China
[5] codingWw@163.com, School of Traffic and Transportation, Lanzhou Jiaotong University, Lanzhou, China; Key Laboratory of Railway Industry on Plateau Railway Transportation Intelligent Management and Control, Lanzhou Jiaotong University, Lanzhou, China

Publisher:
Faculty of Transport and Traffic Sciences, University of Zagreb

ABSTRACT

With the rapid development of intelligent operation and management of urban rail transit, accurate passenger flow prediction is crucial for management and operation. However, the complex, nonlinear and non-smooth characteristics make detecting passenger flow evolution features challenging. In this regard, a novel decomposition integration model, IC-TLCA, is proposed. The model first decomposes the raw passenger flow data into multiple sublayers with different frequencies using an improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN) algorithm to better capture the intrinsic structure of the data. Then, the temporal convolutional network (TCN), long-short-term memory network (LSTM) and channel attention module (CAM) are combined to predict each sublayer separately. Finally, by combining the prediction results of each sublayer, the final passenger flow prediction is obtained. After comparing with all baseline models and conducting ablation experiments, the IC-TLCA model has been validated to be innovative in theory and superior in practical applications, thereby providing an effective solution for passenger flow prediction.

KEYWORDS

passenger flow prediction; improved complete ensemble EMD; temporal convolutional network; channel attention mechanism; long short-term memory; deep learning.

1. INTRODUCTION

Due to its big capacity, punctuality and convenience, travellers are favouring urban rail transit (URT) more and more. Nevertheless, the demand from traffic is also rising in tandem with the annual increase in the sharing rate. The administration and operation of URTs have come under intense pressure in recent years due to the sharp increase in the demand for passenger transit in many cities [1]. In general, the outcomes of forecasting can assist in supporting the choices made by passengers (route selection) and metro authorities (demand management, moment optimisation) [2]. Specifically, predictive insights allow metro authorities to dynamically adjust train scheduling and energy allocation, thereby improving operational efficiency. Enhanced safety management through proactive crowd control measures can mitigate risks and ensure passenger security.

From a user perspective, predictive analytics facilitate congestion reduction strategies that directly improve commuter experiences. Furthermore, such results drive the synergy of a smart city and support multi-modal transportation linkage and scheduling. Unfortunately, because URTs are typically complex systems that are vulnerable to heterogeneous influences, the time series they produce frequently show a high level of complexity, making it difficult to capture the evolutionary features of passenger movements. As a result, developing a precise method for passenger flow prediction is an urgent and critical problem that requires immediate attention.

To better establish predictive models, researchers employ various methods to identify the intrinsic dynamics of traffic time series [3]-[8]. Autoregressive integral moving average (ARIMA) has been used to model traffic time series as a traditional time-domain analytical model [4],[5]. Chaos theory has been utilised to identify patterns in time evolution [6], complexity [7] and traffic time series [8] to represent the nonlinear nature of dynamics. Traffic time series typically show signs of multiscale volatility, including weekly and daily variations. From a frequency domain perspective, the Fourier transform [9], spectral analysis [10] and wavelet transform [11] were utilised to mine frequency features of traffic time series. The fitting capability of the model plays a significant role in predicting the forecasting error during the passenger flow forecasting procedure. Fortunately, because of their great fitting ability and portability, deep learning models are increasingly widely utilised to tackle tasks such as self-supervised traffic demand forecasting [12], vehicle control [13] and categorisation [14]. Researchers used automatic fare collection (AFC)-extracted single-time granularity passenger flows of URTs as inputs to deep learning models to fit the models and predict future passenger flows [15].

Building upon deep learning models, scholars believe that incorporating the empirical mode decomposition (EMD) method can enhance the accuracy of predictive performance [16], [17]. To predict travelling speed, Hamad et al. [18] suggested a hybrid EMD-BPNN (back-propagation neural network) model that consists of an artificial neural network (ANN) prediction stage, an IMF filtering stage and an EMD decomposition stage. Wei and Chen [19] proposed an EMD-BPNN model to predict passenger flow, which includes an EMD decomposition stage, a component identification stage and a BPNN prediction stage. Based on correlation analysis (CA), the IMF was split into two parts during the component identification stage. To predict high-speed rail demand, Jiang et al. [20] built the EEMD (ensemble empirical mode decomposition)-GSVR (gray support vector machine) model, which is composed of an individual prediction stage, an integrated prediction stage and an EEMD decomposition stage. The GSVR predictor forecasts each IMF separately in the individual forecasting stage and the ensemble forecasting stage. The individual forecasts are then combined to provide the final forecast. Furthermore, two studies by Li et al. [21] and Yang and Chen [22] presented an EEMD-based model with two stages: predicting and EEMD decomposition. For network-wide traffic speed prediction, Zhang et al. [23] presented a hybrid model that combines EEMD and 3D (three-dimensional) CNN (convolutional neural network). The EEMD breakdown of locations, individual prediction, integration and fusion of additional features (external and historical) are the four processes in the EEMD-3DCNN model. A hybrid EMD/EEMD-LSTM model with an IMF processing stage, wherein the noiseless part of the IMF is chosen based on energy analysis, was created by Chen et al. [24].

To effectively improve the accuracy of passenger flow sequence prediction for urban rail transit, this paper presents a new hybrid model, IC-TLCA, for urban rail passenger flow prediction. Initially, we utilise the intrinsic computing expressive empirical mode decomposition with adaptive noise (ICEEMDAN) technique, which breaks through the constraints of conventional EMD/EEMD techniques by breaking down the passenger flow data into several intrinsic modal functions (IMFs). Second, we incorporate the channel attention mechanism into the temporal convolutional network (TCN)-long short-term memory (LSTM) architecture to investigate the precise influence of each reconstruction component more intuitively on the prediction outcomes. By leveraging the advantages of each component to the fullest, the combined model TLCA can extract deep, useful characteristics from the data and preserve both short- and long-term information.

This paper’s primary objectives can be summed up as follows:

1) An innovative decomposition framework is adopted to explore the intrinsic patterns of passenger flows.
2) An improved TCN-LSTM model incorporating a channel attention mechanism is proposed for passenger flow prediction.
3) To validate the effectiveness of the proposed model, comparative studies were conducted using actual datasets.

2. LITERATURE REVIEW

2.1 Passenger flow prediction model

For a long time, researchers have been interested in passenger flow prediction, as it has been a hot topic in recent years and forms the basis for managing and operating urban rail transit. Numerous studies have been conducted on modelling methods, research subjects and travel environments. There is a distinction between parametric and nonparametric modelling approaches. Generally, parametric approaches are easier to interpret but may be less accurate and adaptable. The autoregressive integrated moving average (ARIMA) [25] model represents a traditional parametric technique that accounts for the influence of random fluctuations and time series dependence. Apart from ARIMA, seasonality is also incorporated by the seasonal auto-regressive integrated moving average (SARIMA) [26], [27], which can handle cyclically fluctuating data more effectively. While parametric approaches work well for straightforward systems, they are challenging to apply for the analysis of intricate passenger flow data.

On the other hand, because of their strong modelling capabilities, nonparametric approaches can efficiently model complicated situations. Support vector machine (SVM) [28] and basic Bayes [29] are two of these techniques that are frequently used for modelling because of how easy they are to construct and how little data they need. Furthermore, by resampling and randomly selecting data, integration techniques make use of all the information present in the data to increase prediction accuracy. Decision trees [30] and random forests [31] are two examples of multiple integration techniques that are frequently employed.

With the advancement of deep learning in recent years, researchers have built increasingly complex passenger flow forecast models. Deep learning techniques, such as deep belief networks (DBN) [32] and stacked autoencoders (SAE) [33], are frequently employed to capture the spatio-temporal properties of passenger movement. Recurrent neural network (RNN) and its variations, including gated recurrent unit (GRU) and long short-term memory (LSTM), are popular deep learning models that are used to mine temporal characteristics of passenger/traffic flows [2],[34], [36]. This is because RNN can model sequential data. Furthermore, for traffic series prediction tasks, convolutional neural network (CNN) [37], [38] and graph neural network (GNN) [39],[40] have been extensively employed to collect spatial characteristics. An analysis of the above algorithms shows that they take into account the temporal dependence of the signals; however, it is usually limited to the temporal dependence between neighbouring time steps. Nonetheless, there is typically a long-term degradation tendency indicated by the effect of signal dependence on performance degradation at various time intervals.

Compared to CNN and RNN, a temporal convolutional network (TCN) is far more efficient at extracting information on performance degradation over extended time scales. In particular, the TCN-LSTM framework [41] demonstrates its superiority when dealing with time-series prediction problems. Furthermore, the TCN framework has been effectively utilised to identify the temporal and spatial dependencies of prediction targets through the application of spatial attention mechanisms. For instance, Lin et al. [42] created a deep learning model based on the channel attenuation mechanism to anticipate the remaining service life of aircraft engines and confirmed the channel attention mechanism's efficacy. However, previous studies have not realised to capture both long-term and short-term dependencies of passenger flow data. Therefore, the TCN-LSTM-CAM framework is proposed to fill the research gap.

2.2 Empirical mode decomposition-based model

In order to handle nonlinear and non-smooth time series, empirical modal decomposition (EMD) and its enhanced form, ensemble EMD (EEMD), can split the original data into several IMFs and one residual. EMD has been applied to investigate multiscale oscillations in transportation time series because of its benefits [43]-[45].

A hybrid model for short-term traffic time series forecasting was created using the data analysis methodology mentioned above as a basis [46]. Decomposing the traffic time series into meaningful components is advantageous to increase forecasting performance, especially given the complexity of the traffic time series. Consequently, researchers have made use of decomposition-based hybrid models; for instance, wavelet transformations are taken into consideration in hybrid models for passenger flow prediction [47],[48].

Furthermore, time series analysis and prediction are still difficult because of their instability and randomness. Hamad et al. have demonstrated that prediction accuracy can be increased and the series' characteristics can be adequately captured by frequency decomposition techniques [18],[19], [49]-[51].

Specific methods include empirical mode decomposition (EMD) [52], ensemble empirical mode decomposition (EEMD) [53], complete ensemble empirical mode decomposition (CEEMD) [54], [55] and complete integrated empirical mode decomposition with adaptive noise (CEEMDAN), which can efficiently decompose any type of time series [56], [57]. Nevertheless, EMD faces the issue of "mode mixing". EEMD reduces mode mixing by adding Gaussian white noise to the original signal; however, it also generates spurious components that interfere with later signal processing. Although CEEMD significantly reduces the reconstruction issue by adding white noise in pairs to the original data, it is unable to verify completeness. Adaptive noise is used by CEEMDAN to perform fully empirical mode decomposition, resulting in nearly zero reconstruction errors. It also resolves the issue of introducing noise to various signals to produce varying numbers of modes. Still, there is some residual noise in its modes. The improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN, IC-EMD) algorithm [58] addresses the shortcomings of the original CEEMDAN. It can significantly reduce the residual noise in intrinsic mode functions (IMFs) and also resolve the issue of averaging across different numbers of IMFs. Therefore, the prediction model proposed in this paper is based on IC-EMD, and its effectiveness has been demonstrated through experiments.

Inspired by previous studies, this study proposes a novel hybrid model aimed at improving the accuracy of metro passenger flow prediction. The model is based on the IC-EMD decomposition and combines the parallel integration of the TCN-LSTM module and the channel attenuation module. In this way, the model is able to effectively capture the spatial and temporal dependencies in the prediction process, which is expected to provide more accurate prediction results.

3. METHODOLOGY

To reduce liquidity and volatility and improve the prediction accuracy of the passenger flow series, this paper proposed a passenger flow prediction model, which is called IC-TLCA. In this section, we analyse the models' components and methods of learning or application first. Next, the combination algorithm's basic idea is presented. *Figure 1* shows the structure of the IC-TLCA model.

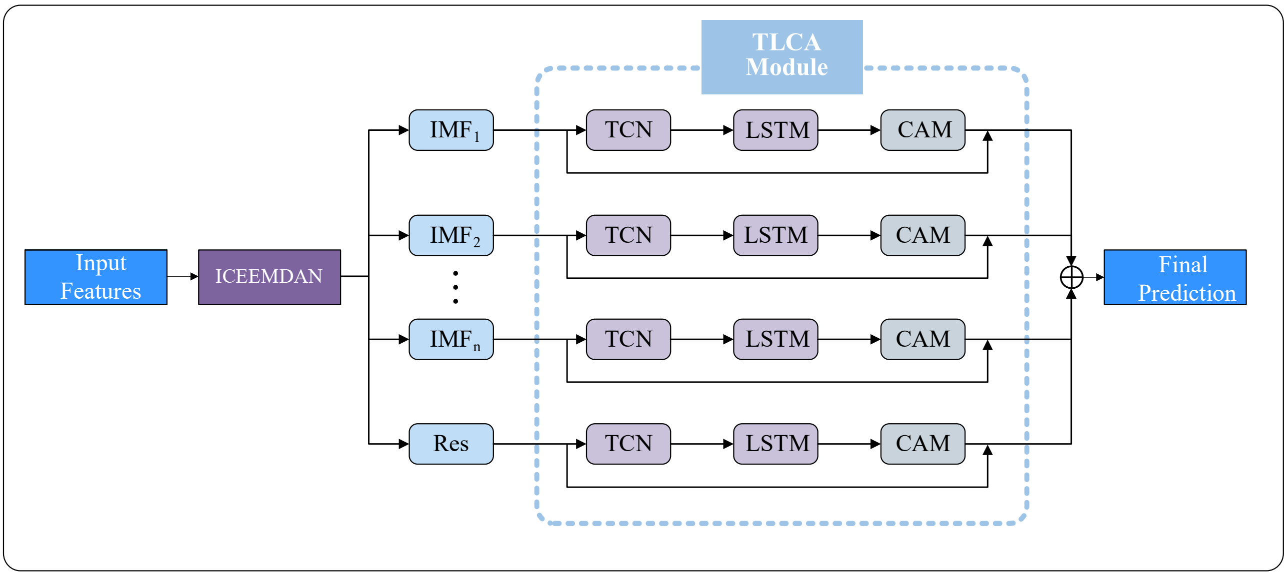


Figure 1 – The structure of the IC-TLCA

3.1 Problem definition

In this study, a time series forecasting method based on covariate forecasting is used for urban rail passenger flow forecasting. The method is based on the principle of multidimensional time series synergy analysis and realises multi-step forward forecasting by modelling the dynamic coupling relationship between the covariates and the target series. Given a collection of time series $\{(X(t),(Y(t))\}_{t=1}^{T}$, where $X(t)=[x_1(t),x_2(t),\dots,x_p(t)]$ represents a vector of p-dimensional covariates, Y(t) is the target passenger flow sequence, t represents a time step (t=1, 2, ..., T), and T denotes the observation window length. For time step t, past covariate information and target sequence information are utilised to predict future values.

The prediction task can be formally defined as constructing a mapping function F to predict the passenger flow data for the next k steps at any time step t_0, based on the historical data $X(1{:}t_0), Y(1{:}t_0)$, which is defined as follows:

$$\hat{Y}(t_0 + k) = F(X(t_0 - \tau{:}t_0), Y(t_0 - \tau{:}t_0)) \tag{1}$$

where: τ is the time window length and k is the prediction step. When k=1, the output sequence is denoted as $\hat{y}(t_0 + 1)$, which at this moment only predicts the target value of one time step in the future, belonging to single-step prediction. When t＞1, the output sequence is denoted as $\{\hat{y}(t_0 + 1), \hat{y}(t_0 + 2), \dots, \hat{y}(t_0 + k)\}$, which means predicting the target value of multiple future time steps, belonging to multi-step prediction. In this study, single-step prediction and multi-step prediction are used together to realise urban rail passenger flow prediction.

3.2 Decomposition: ICEEMDAN

Time-domain, frequency-domain and occasionally time-frequency domain processing techniques are frequently used in signal processing.

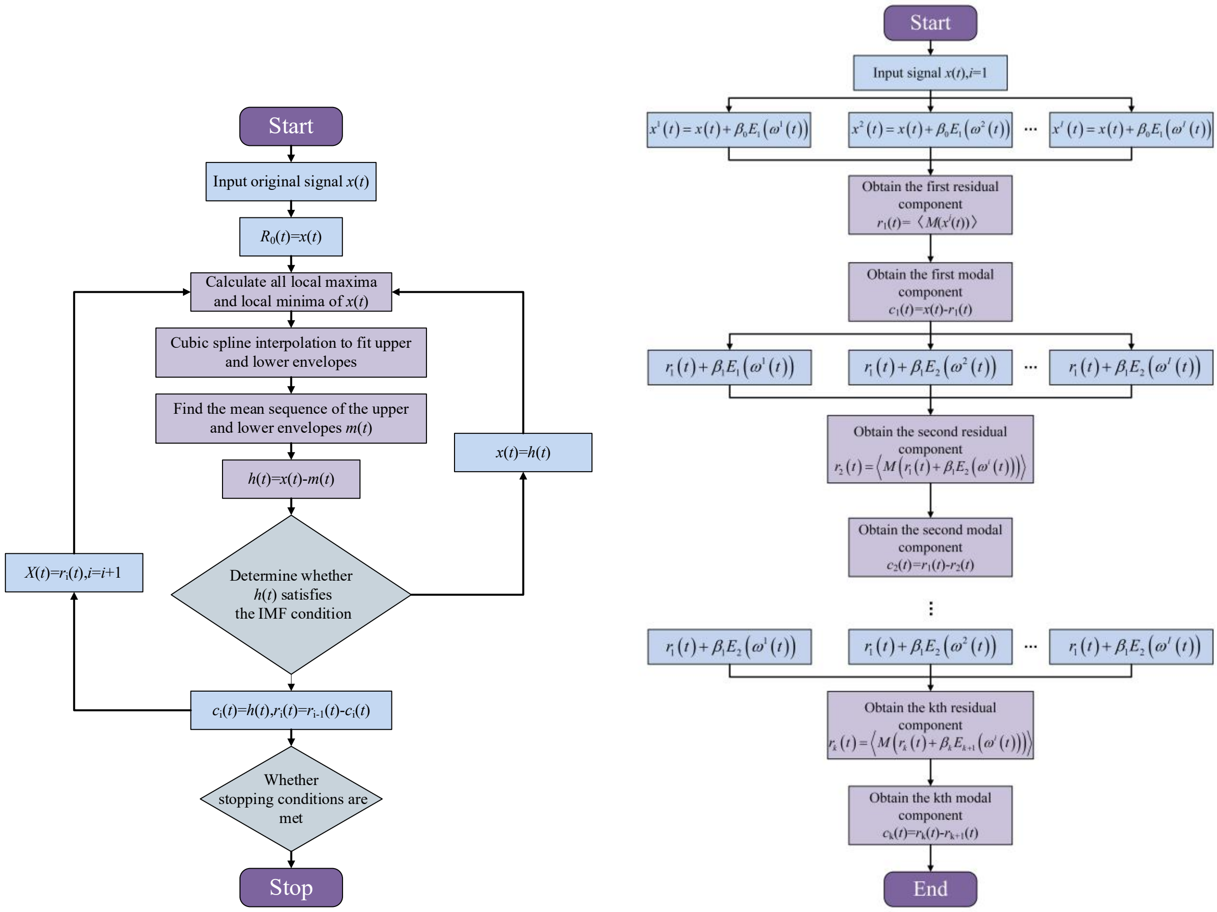
Time domain: the dynamic signal is a function that characterises the value of the signal at various times; the link between amplitude and time is shown by the time domain graph. The independent variable is time; the dependent variable is the signal's change.

Frequency domain: the amplitude of the signal at that frequency is the dependent variable, while frequency is the independent variable. The frequency domain plot displays the peak amplitude vs frequency without displaying the amplitude change over time.

The term "time-frequency domain" refers to a technique for analysing non-stationary, time-varying signals that integrate data from the frequency and time dimensions to clearly describe the relationship between signal frequency and time. By creating a combined function of time and frequency, this approach describes the energy density or intensity of a signal at various times and frequencies. These time-frequency distributions allow for the analysis of the signal's instantaneous frequency and amplitude, as well as the execution of time-frequency filtering and time-varying signal studies.

The fundamental advantage of empirical mode decomposition (EMD) as a time-frequency domain processing technique is that it solves the issue of the basis function's lack of adaptivity; for a segment of an unknown signal, the decomposition can begin immediately without the need for earlier research and analysis. Without the need for human settings or interaction, this system will automatically follow a few solid modes hierarchically. Based on the local properties of the data itself, such as local maxima, local minima, and zero-crossings, EMD can break down the original data into a set of intrinsic mode functions (IMFs) and residuals. *Figure 2a* shows the flowchart of EMD decomposition.

As a variant of EMD, improved complete ensemble empirical mode decomposition with adaptive noise (ICEEMDAN, IC-EMD) decomposes complex signals into finite IMFs and residuals [58]. Every IMF can further minimise the noise and contain the signal's local features. First, the modal estimation is substituted with the local mean. Second, the k-order modes are extracted using the local mean value of the signal rather than white noise directly. IC-EMD automatically determines the number of IMFs in decomposition. This makes it more flexible and precise in handling complex signals. During decomposition, it mainly stops when the residue shows monotonicity or has too few extremes to form a new IMF, ensuring reasonable and effective decomposition results. *Figure 2b* shows the flowchart of IC-EMD decomposition.



(a) *(b)*

Figure 2 – The flowchart of signal decomposition: a) The flowchart of EMD decomposition; b) The flowchart of ICEEMDAN decomposition

3.3 Temporal convolutional network (TCN)

CNN processes an image by treating it as a matrix of m·n, or a two-dimensional "block". The series can be thought of as a one-dimensional object (1·n vector) after converting to a time series. A sufficiently large receptive field can be achieved with a multi-layer network configuration. Because of the benefit of massively parallel processing, the CNN may be processed in parallel, regardless of its depth, which saves a significant amount of time. This approach makes CNN extremely deep.

For the first time, temporal convolutional networks for video-based action segmentation were introduced in the groundbreaking work of Lea et al. [59]. There are two steps in this conventional process. Initially, a CNN is utilised to calculate low-level features, which typically encode spatiotemporal data. Subsequently, these low-level features are input into a classifier, which employs an RNN to obtain high-level temporal data. The primary disadvantage of this method is that it necessitates the use of two different models. Models based on the CNN architecture that are specifically designed to process time series data are called temporal convolutional networks (TCN). Convolutional operations are used by TCN to efficiently compute in parallel, capture local dependencies in sequence data, and maintain stable gradient propagation during training to prevent vanishing or ballooning gradients.

At the moment, extensive studies have demonstrated the adaptability of TCNs across various sectors, including speech recognition, machine translation, traffic forecasting and human activity identification. Nonetheless, the utilisation of TCNs for predicting passenger flow is still not very frequent. To improve future passenger flow prediction, we use a TCN network in this study to extract time-step-specific features from short-duration passenger flow time-series data. We then use feature and trend learning through training.

TCN possesses three core components: causal convolution, dilated convolution and residual connections. These fundamental characteristics offer a consistent method for hierarchically capturing all two layers of information.

Two characteristics of the causal convolution's structure are as follows.

It has a unidirectional architecture, is non-bidirectional, and ignores information that may become available in the future. Only the previously observed sequence ($x_1, \cdots, x_t$) can be employed when given an input sequence ($x_i, \cdots, x_t$) and a prediction ($y_1, \cdots, y_t$). There are more hidden layers in information the further back in time it is tracked. When the first hidden layer is used as the output, its final node is linked to nine input nodes, such as x_{t-8}, x_{t-7}, $\cdots$, x_{t-1},x_t; if the first hidden layer is used as the output, its final node is linked to three input nodes, i.e. x_{t-2},x_{t-1},x_t.

Suppose the TCN filter is F =(f_1,f_2,f_3,$\cdots$,f_k), and the causal convolution of sequence X = (x_1,x_2,$\cdots$,x_t) at x_t is:

$$(F \cdot X)_{x_t} = \sum_{k=1}^{K} f_k \, x_{t-K+k} \tag{2}$$

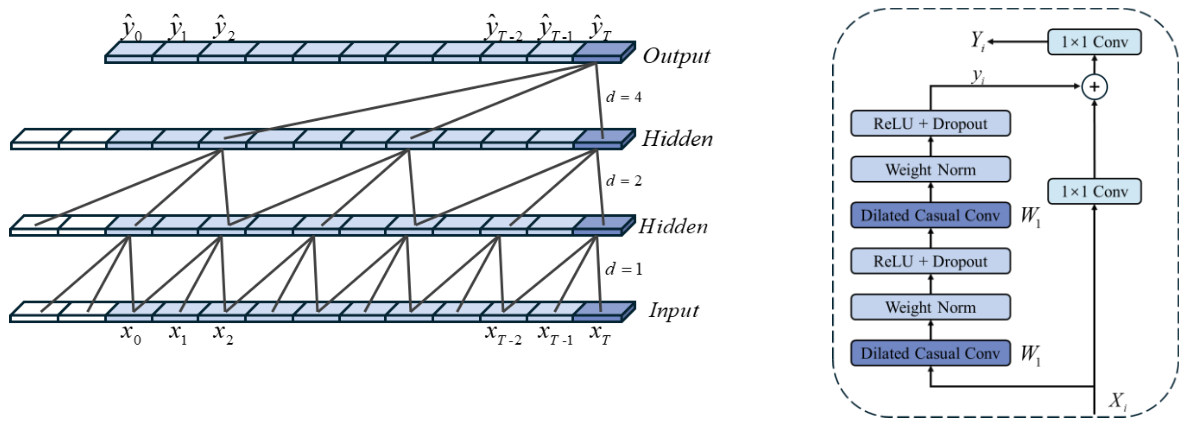
The issue with typical convolutional neural networks persists in simple causal convolution as well: the number of layers that must be linearly stacked in order to capture longer dependencies, and the modelling length is restricted by the size of the convolutional kernel. CNN can get a larger sense of the field by incorporating more pooling layers; nevertheless, information loss following pooling remains a concern. This problem is extremely nicely solved using dilated convolution. Dilated convolution allows for intermittent sampling of the input throughout the convolution process, in contrast to standard convolution. As a result, the convolutional network requires fewer layers to create a bigger receptive field. This method has the benefit of not requiring the addition of a pooling layer, which could result in information loss, and enabling each convoluted output to include a wider variety of data. *Equation 3* provides the definition of dilated convolution at x_t.

$$F(s) = \left(x \cdot {}_d f\right)(s) = \sum_{i=0}^{k-1} f(i) \cdot x_{s-d \cdot i} \tag{3}$$

where d is the dilation rate, which generally grows exponentially by 2 as the number of network layers increases, *s-d·i* is the past direction, *k* is the size of the filter. *Figure 3a* depicts the architecture of a dilated causal convolution. The efficacy of residual connections in deep network training has been shown, allowing cross-layer information transfer. A residual block has a branch that goes to a set of transformations F, the outputs of which are added to the block's input X:

$$o = Activation\left(X + F(X)\right) \tag{4}$$

TCN uses the rectified linear unit (ReLU) to represent the nonlinearity and dilated causal convolution layers within a residual block, as illustrated in *Figure 3b*. The convolutional filters are normalised by applying weight normalisation. Moreover, each layer incorporates a spatial dropout for regularisation, which results in the zeroing out of the entire channel at every training step, following dilated convolution.



(a) *(b)*

Figure 3 – The architecture of TCN: a) The architecture of a dilated causal convolution; b) The structure of TCN

3.4 Long short-term memory (LSTM)

The LSTM model, which was introduced by Hochreiter and Schmidhub [60], is an enhanced version of the RNN that addresses the problems associated with gradient disappearance and explosion that are inherent in conventional neural networks. Constructed to address long-term dependency issues, the LSTM model primarily consists of memory cells, input gates, output gates and forget gates. The sigmoid function is utilised by the three gates to activate them. The forgetting gate stores previously learned information in reserve in the neuron at a prior time, the input gate controls the neuron's input information at that moment, and the output gate controls the neuron's output information at that same moment [61]. The LSTM network structure is depicted in *Figure 4*. Among them, the activation functions σ and tanh are *Equation (5)* and *Equation (6)* which increase the non-linearity of the neural network.

$$\sigma = \frac{1}{1+e^{-x}} \tag{5}$$

$$tanh(x) = \frac{1-e^{-2x}}{1+e^{-2x}} \tag{6}$$

The information from the preceding cell memory is filtered using the forget and input gates, as shown in Figure 4, to identify which old data should be deleted and which new data should be inserted. This procedure can be expressed using *Equations (7)–(10)*.

$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t \tag{7}$$

$$f_t = \sigma\left(W_f \cdot [y_{t-1}, x_t] + b_f\right) \tag{8}$$

$$i_t = \sigma(W_i \cdot [y_{t-1}, x_t] + b_i) \tag{9}$$

$$\tilde{C}_t = tanh(W_C \cdot [y_{t-1}, x_t] + b_C) \tag{10}$$

In the above formula: W is the weight matrix, $[y_{t-1}, x_t]$ is the new vector created by concatenating vectors y_{t-1} and x_t, and b is the bias of each unit. $f_t \cdot C_{t-1}$ represents the information that remains after the cell discards some of the previously remembered information. The newly entered data that the cell has to remember is represented by $i_t \cdot C_t$. The cell memory C_t and the current input x_t are then fused using the output gate to produce the output y_t, which can be computed using *Equation (11)* and *Equation (12)*.

$$o_t = \sigma(W_o \cdot [y_{t-1}, x_t] + b_o) \tag{11}$$

$$y_t = o_t\, tanh(C_t) \tag{12}$$

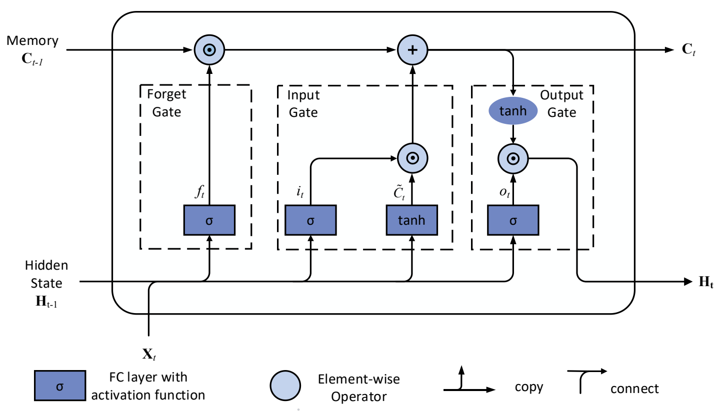


Figure 4 – The structure of LSTM

3.5 Channel attention module (CAM)

The attention mechanism's core idea is to selectively focus on the desired target region to obtain more insightful data. It was first studied in the field of the human vision. This is also valid for time series, which are typically quite lengthy. However, only a small portion of the data is important or relevant for the conclusion, therefore, using attention mechanisms can frequently enhance the network's performance. For passenger flow data series, efficient feature extraction – that is, high accuracy and high speed is required. To fulfil this requirement, this paper uses the channel attention module of the convolutional block attention module to improve the model's perceptual ability and performance without increasing the network complexity. The structure of the channel attention module is shown in *Figure 5*.

By giving effective channels a higher weight and less weight to less valid channels, channel attention concentrates on the most significant aspects. The steps to implement the channel attention module are as follows.

Initially, the process begins with the application of global max and global average pooling to the input feature maps across each channel. This step extracts the maximum and average feature values for each channel, creating two distinct vectors that encapsulate the global extrema and mean characteristics of the channels. Subsequently, these vectors are fed into a shared dense layer, which is positioned after the pooling operations. The purpose of this dense layer is to ascertain the attention weights for the channels, enabling the model to discern and emphasise the significance of each channel relative to the task at hand. The channel-wise attention weights are derived by merging the global max and average feature vectors, followed by the application of a sigmoid function. This function scales the weights to a 0-to-1 range, which are then integrated into the original feature maps on a per-channel basis. Ultimately, the attention weights are doubled, augmenting the feature maps with a focus on task-relevant channels while downplaying the less pertinent ones.

The above is expressed by:

$$M_c(F) = \sigma\left(W_1\left(W_0\left(F_{avg}^c\right)\right) + W_1\left(W_0\left(F_{max}^c\right)\right)\right) \tag{13}$$

where σ is the sigmoid activation function, W_0 and W_1 are weights, and F is the feature.

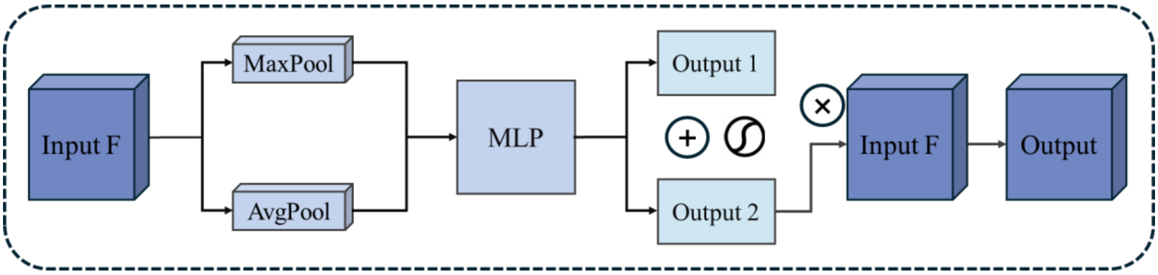


Figure 5 – The structure of CAM

3.6 Prediction: TLCA

As shown in *Figure 1*, the TLCA model incorporates the TCN module and the LSTM module with the CAM module. The unique one-dimensional causal convolutional structure of TCN ensures the time-series properties of the data, while the dilation convolution ensures the extraction of all data features, and the residual connectivity unit speeds up the network's convergence. Furthermore, as a variation of the classical RNN, the LSTM model effectively manages sequential data by fitting entities in a nonlinear manner. It captures long-term dependencies within time series through its use of gating mechanisms and memory cells. The prediction of passenger flow is a nonlinear time series prediction problem. Passenger flow data has two characteristics: the diversity of features and the volatility of passenger flow. Inspired by the breakthrough of the ST-LSTM model [41] in network traffic prediction, we used TCN-LSTM, the model's core component, and adjusted its structure. In the ST-LSTM model, TCN is composed of two residual blocks. The first block comprises two layers of causal dilated convolution, each with a kernel size of 5, a dilation factor of 1, and 10 filters. The second expansion factor is 2, and the remaining parameters are the same as the first one. In the TLCA model, we increased the number of layers. Specifically, the model utilised dilation factors of 8, 16, 32 and 64, along with a kernel size of 3; all other parameters remained consistent with the original model. In this study, we merge the TCN and LSTM models to create a TCN-LSTM module that accounts for the nonlinear characteristics of passenger flow and the influence of numerous parameters to forecast the number of passengers. Furthermore, we implement the CAM in TCN-LSTM to determine the relative weights of passenger flow data features.

In the TLCA model, higher-dimensional features are created by the TCN-LSTM module through data-driven learning. Subsequently, the channel attention mechanism is employed to prioritise various features based on their relative value, with a greater emphasis placed on the characteristics that have a larger reaction to passenger flow. The training dataset serves as the input for the TCN module, which uses the input data to extract features. After that, the data's impurities will be significantly decreased, exposing the features more clearly and helping the LSTM model learn. The LSTM model receives the features extracted by the TCN module and uses the forget gate, input gate and output gate to control the network's output. That enables the model to handle long-term serial data and produce accurate predictions by forgetting irrelevant information and remembering important information that must be remembered for a long time.

3.7 Hybrid model IC-TLCA

The decomposition and prediction phases make up the IC-TLCA model. First, the original dataset is decomposed using IC-EMD. Then, each IMF is predicted separately. The ultimate result is obtained by adding the sum of the individual predictions. *Figure 1* shows the structure of IC-TLCA.

IC-EMD first decomposes passenger flow data into residuals and IMFs. They are arranged from high to low frequency. The IMFs are then processed by the TCN model first, and then the LSTM and CAM models receive them to produce the prediction results for each IMF. The residual structure is then used to concatenate the TLCA model's predictions with the real IMF data during the training phase, which improves stability and speeds up convergence. Ultimately, the total of each forecast is calculated to determine the outcome.

The proposed IC-TLCA model comprises 143,360 trainable parameters, with the LSTM module accounting for the largest proportion (68.6%, 98,304), followed by the TCN module (25.7%, 36,864), and the CAM contributing the smallest share (5.7%, 8,192). In terms of computational complexity, the decomposition phase exhibits a time complexity of $O(L \cdot N^2)$, where the decomposition layer count L=8 and the input sequence length N=1095. The prediction phase (TLCA) requires 3.8 GFLOPs in total, with the LSTM module dominating 70% of the computational load, while the TCN and CAM modules contribute the remaining 30%. Based on the above, IC-TLCA offers practical applicability for urban rail transit passenger flow prediction tasks.

4. EXPERIMENTS

4.1 Dataset

In this study, we use operational data and smart card data from Xi'an Metro to validate the proposed IC-TLCA model. Xi'an Railway Transportation Group Co. provides these real datasets and covers the operation records from 1 January 2017 to 31 December 2019. It includes one metro line and 26 metro stations. We used 1,095 days of passenger flow data during the target period. To achieve an accurate prediction of drastic changes in passenger flow, we specifically tagged the holiday data. A weekly cyclical dataset is obtained directly from the calendar, where Monday to Sunday is a repeating pattern for the week. In China, double holidays (Saturday and Sunday) are considered holidays. In addition, the dates of special holidays can be determined by querying the 'Chinese calendar' package in Python. We use 0-1 variables to mark the day of the week and whether it is a holiday. As a result, the input dataset has a dimension of 1095×10. We created three subsets of data: 70% for training, 15% for testing, and 15% for validation to ensure the robustness of the model. Before model training, we normalised the data to standardise variables of different magnitudes to improve the generalisation of the model.

Employing the IC-EMD algorithm, we decomposed the passenger flow data into several independent intrinsic mode functions (IMFs) and one residual component. *Figure 6* presents the IC-EMD decomposition results for the metro passenger flow data from 2017 to 2019, revealing a total of 8 IMFs and 1 residual. These decomposed IMFs are ordered from the highest to the lowest frequency. The high-frequency IMFs represent the rapid fluctuations within the original data, while the low-frequency IMFs reflect the more gradual changes. The residual component, which is the final output of the decomposition process, often encapsulates the underlying trend of the original data. However, its interpretation may vary depending on the specific characteristics of the dataset.

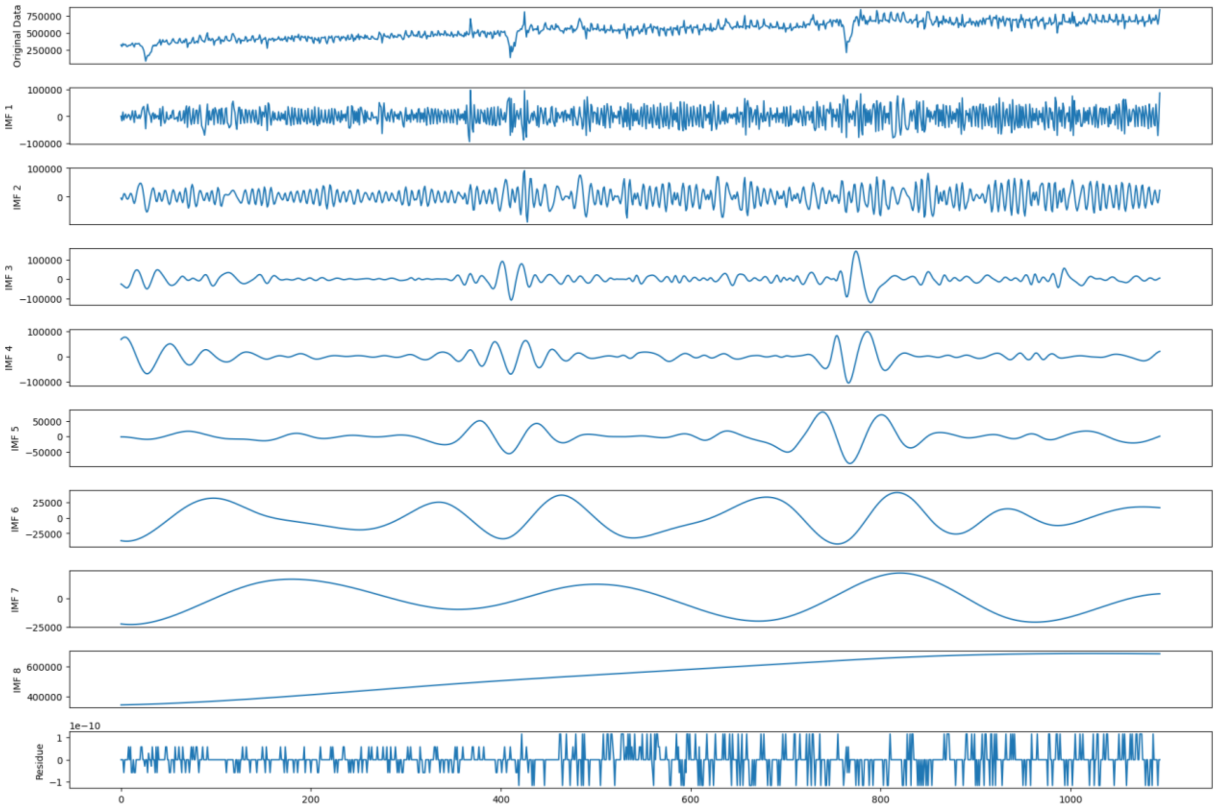


Figure 6 – The result of ICEEMDAN

To accommodate the diverse needs of metro operations, we use forecasting methods with different time granularities, including predicting passenger flows for the upcoming 1 day, 3 days and 6 days. Specifically, the 1-day and 3-day forecasts are categorised as short-term predictions, enabling operators to swiftly reallocate resources in response to immediate operational challenges, such as abrupt passenger flow variations caused by unforeseen events or meteorological changes. Meanwhile, the 6-day forecast, which falls into the medium-term category, equips operators with advanced insights into passenger flow trends over an extended horizon. This facilitates the strategic planning of vehicle deployment, workforce scheduling and infrastructure maintenance. This multi-granularity forecasting approach enables metro operators to obtain richer and more detailed information for decision-making, thus ensuring operational efficiency while improving service quality and passenger experience.

4.2 Model configurations and evaluation metrics

In this experiment, the hyperparameters of the model include the number of hidden units, batch size and learning rate. During the training phase, the number of hidden units was set to 64, chosen from the set {16, 32, 64, 128} after preliminary experiments indicated optimal performance at this value. The batch size was determined to be 16, selected from the set {16, 32, 64, 128} based on computational efficiency and convergence speed considerations. The learning rate was set to 0.08, which was identified as the most effective value through a series of hyperparameter tuning experiments. The training process spanned 200 epochs to ensure thorough learning. To enhance the model’s generalisation capabilities, a dropout rate of 0.05 was incorporated during training. The model parameters were optimised using the ADAM optimiser, chosen for its robustness in handling issues related to vanishing and exploding gradients. The training objective was to minimise the mean square error (MSE), a standard metric for evaluating the accuracy of model predictions. This was achieved through the backpropagation algorithm, which facilitated the adjustment of model weights to reduce prediction errors.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{14}$$

In this paper, root mean square error (RMSE), mean absolute error (MAE) and mean absolute percentage error (MAPE) are chosen as the assessment criteria to evaluate the performance of the proposed hybrid model:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{15}$$

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \tag{16}$$

$$MAPE = \frac{1}{N}\sum_{i=1}^{N}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\% \tag{17}$$

y_i is the real value, $\hat{y}_i$ is the predicted value and N is the number of data points.

4.3 Baseline models

To thoroughly assess the predictive performance of our proposed model, we compared it against seven established baseline models in our experiments. Furthermore, to ensure the rigour of our study, we have provided a comprehensive description of the programming software's running environment in *Table 1*.

Table 1 – Hardware and compiling environment

Categories	Version
Central processing unit (CPU)	13th Gen Intel(R) Core(TM) i7-13620H 2.40 GHz
Operating system (OS)	Windows 11 Professional x64
Random access memory (RAM)	16.0 GB
Graphics processing unit (GPU)	NVIDIA GeForce GTX 4060 8 GB
Programming language	Python 3.9.18
Development environment	VSCode 1.84.1
	CUDA 12.1
Main Python modules	PyTorch 2.3.0
	EMD-signal 1.4.0
	NumPy 1.26.4
	Pandas 1.5.3
	Matplotlib 3.8.3

1) ARIMA: In this model, ARIMA, a widely utilised linear model for forecasting passenger flow, has its parameters configured as follows: the lag order is set to 2, the degree of difference is 1, and the order of moving averages is also 1.
2) LSTM: LSTM is also used for time series prediction on its own because it can model time dependencies. By adding three gating mechanisms and an internal state, it preserves long-term relationships while allowing the network to infer long-term information from sequence data. In this model, there is one fully connected layer accompanied by two hidden layers. The hidden states are configured with a dimension of 128, while the fully connected layer comprises 64 neurons.

3) GRU: Similar to LSTM, GRU is made to address the long-term dependency issue that arises when working with sequential data in classic recurrent neural networks. To decrease the number of parameters and computational complexity of the model, it is a simplified form of LSTM that merges the input and forgetting gates into a single update gate and does not have a separate cell state. In this model, there is one fully connected layer accompanied by two hidden layers. The hidden states are configured with a dimension of 128, while the fully connected layer comprises 64 neurons.
4) LSTM-GRU [62]: GRU possesses fewer structural parameters compared to LSTM neural networks, which in turn have more gates than GRU. Combining LSTM and GRU can produce powerful learning capabilities with fewer network parameters. Thus, the model is used as a baseline. This model architecture consists of an LSTM layer, a GRU layer and a fully connected layer. The LSTM and GRU layers utilise hidden states with a dimension of 128. The fully connected layer, which serves as the output layer, comprises 64 neurons. These parameter settings are consistent with those of the aforementioned LSTM and GRU models.
5) Seq2SeqA: Seq2Seq is a powerful deep learning method with a wide range of applications in time series forecasting. Its capacity to handle long-term dependencies and nonlinear interactions in time series data is its main strength. It generates predictions for future values by using a paired encoder and decoder structure to understand the intricate links between sequence data. In this paper, the Seq2Seq model with attention (Seq2SeqA) is utilised as a baseline model. This model's encoder comprises an LSTM layer, while the decoder consists of an LSTM layer followed by a Bandana attention mechanism. The output of the decoder is then processed through two linear layers to generate the final predictions. Specifically, the LSTM layer utilises 64-dimensional hidden states, and the fully connected layer consists of 128 neurons.
6) Transformer [63]: The transformer is a groundbreaking architecture in the field of deep learning, particularly in natural language processing. However, its application has been extended to time series forecasting due to its ability to capture complex temporal relationships. Unlike RNN and CNN, the transformer uses self-attention mechanisms to weigh the importance of different parts of the input sequence, allowing it to focus on relevant temporal patterns. This makes the transformer highly effective in modelling long-term dependencies and effectively handling large-scale time series data. The model's encoder-decoder structure is particularly useful for tasks requiring understanding the sequence's context to make accurate predictions. In this model, the conventional transformer is composed of three encoder layers and three decoder layers, each equipped with eight multi-head attention mechanisms. The dimensions for d_k and d_v are both configured to be 64. Following the transformer's output, it is fed into two fully connected layers, the first with 128 neurons and the second with 64 neurons.
7) Informer [64]: Informer is an innovative variant of the transformer model, specifically designed for long sequence time series prediction tasks. It introduces a probabilistic sparse self-attention mechanism, which significantly reduces the computational complexity from $O(L^2)$ to $O(L\,log\,L)$, effectively minimising both computation and memory usage during the processing of extended sequences. Furthermore, the informer employs a self-attention distillation technique that optimises allocating computational resources and memory consumption for long sequences by emphasising the most salient attentional patterns. This approach contrasts with traditional step-by-step decoding methods, as the informer's generative decoder can predict entire sequences in a single pass, thereby markedly enhancing inference speed. These enhancements enable Informer to perform better than existing methods across multiple large-scale datasets. The parameters of the informer are the same as the transformer.

4.4 Prediction performance

To assess the performance of the proposed TLCA and IC-TLCA models, we conducted comparative experiments against established benchmarks, including ARIMA, LSTM, GRU, LSTM-GRU, Seq2SeqA, transformer and informer, for the task of passenger flow prediction. *Table 2* presents the MAE, RMSE and MAPE values for 1-day, 3-day and 6-day forecasting horizons, with the best-performing values highlighted in bold. The TLCA and IC-TLCA models consistently outperformed all baseline models across all prediction intervals. Specifically, TLCA demonstrated an average reduction in error metrics of 5.3% for MAE, 4.3% for RMSE, and 5.6% for MAPE compared to the top-performing baseline model. The IC-TLCA model showed even more significant improvements, with average reductions of 9.6% for MAE, 7.8% for RMSE and 9.7% for MAPE. These enhancements were consistent across different prediction time frames.

Table 2 – Performance comparison of different models

Model	**1-day**			**3-day**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
ARIMA	8.549	15.783	7.447	9.872	18.242	8.855
LSTM	5.364	9.917	4.672	5.989	9.504	4.909
GRU	5.237	9.328	4.391	6.032	9.854	5.077
LSTM-GRU	5.028	9.756	4.475	5.957	9.447	5.071
Seq2SeqA	4.907	9.272	4.136	5.817	9.316	5.237
Transformer	4.895	8.882	4.294	5.469	9.294	4.916
Informer	5.085	9.693	4.413	5.352	9.705	4.823
TLCA	4.634	8.273	4.028	5.089	9.126	4.505
IC-TLCA	**4.373**	**7.858**	**3.973**	**4.795**	**8.809**	**4.147**
Model	**6-day**			**Average**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
ARIMA	8.549	15.783	7.447	9.872	18.242	8.855
LSTM	5.364	9.917	4.672	5.989	9.504	4.909
GRU	5.237	9.328	4.391	6.032	9.854	5.077
LSTM-GRU	5.028	9.756	4.475	5.957	9.447	5.071
Seq2SeqA	4.907	9.272	4.136	5.817	9.316	5.237
Transformer	4.895	8.882	4.294	5.469	9.294	4.916
Informer	5.085	9.693	4.413	5.352	9.705	4.823
TLCA	4.634	8.273	4.028	5.089	9.126	4.505
IC-TLCA	**4.373**	**7.858**	**3.973**	**4.795**	**8.809**	**4.147**

Through experimental testing, we observed that TLCA outperforms all baseline models across all assessment metrics. Specifically, for one-day prediction, TLCA achieved excellent performance with an MAE of 4.634, RMSE of 8.273 and MAPE of 4.028. For three-day and six-day predictions, TLCA continued to demonstrate superior performance, with MAE, RMSE and MAPE values of 5.089, 9.126, 4.505; and 6.361, 11.425 and 5.383, respectively. Collectively, these results substantiate the effectiveness of our proposed method. Furthermore, IC-TLCA, enhanced by the IC-EMD algorithm, achieved even better prediction results. On one-day prediction, IC-TLCA obtained an MAE of 4.373, RMSE of 7.858 and MAPE of 3.973. For three-day and six-day predictions, IC-TLCA exhibited outstanding performance, with MAE, RMSE and MAPE values of 4.795, 8.809, 4.147; and 6.171, 11.087 and 5.163, respectively. The results indicate that the model incorporating the IC-EMD decomposition outperforms the model without this feature. The predictive accuracy is substantially enhanced by the inclusion of IC-EMD decomposition. Consequently, we conclude that IC-EMD decomposition is crucial for improving the accuracy of metro passenger flow predictions.

Compared with other models, the ARIMA model exhibits inferior performance, which suggests that this conventional linear model fails to capture the evolutionary characteristics of passenger flow. On the other hand, both LSTM and GRU are capable of effectively modelling the temporal dependencies inherent in passenger flow. As a variation of the LSTM, the GRU displays a comparable level of capability, and in some cases, even surpasses LSTM. The LSTM-GRU method harnesses the advantages of both: the robust memory retention of LSTM and the significant computational efficiency of GRU, collaboratively enhancing the model's performance and training velocity without compromising the capture of long-term dependency information. In situations where computational resources are limited, this combined approach can yield exceptional outcomes in certain complex sequence modelling tasks. Moreover, the benefits of its encoder-decoder design and the attention mechanism, the Seq2SeqA model outperforms the LSTM-GRU structure in time series prediction tasks, proving especially valuable for capturing the complex interconnections between sequences. In all scenarios, the proposed TLCA model demonstrates the best performance, indicating that it is an effective model for forecasting passenger flow. Furthermore, the transformer architecture leverages multi-head self-attention to capture multiple temporal dependencies, thereby enhancing prediction accuracy. However, its suitability for long-term serial sequence processing is limited due to the memory constraints imposed by the stacked layers when processing lengthy inputs. To address this limitation, the informer model introduces a novel probsparse attention mechanism, which aims to preserve high predictive performance while significantly reducing computational and memory demands. Nevertheless, both the transformer and informer models may exhibit sensitivity to minor variations in input data, potentially leading to significant fluctuations in output and, consequently, affecting the stability of prediction outcomes in certain scenarios. Furthermore, we observed a decrease in prediction accuracy as the prediction horizon extended from 1 day to 3 days and then to 6 days for all models. This trend is likely due to the presence of a strong temporal correlation in passenger flow data within shorter prediction horizons, such as 1 day. This correlation enables the models to more effectively capture the nonlinear dynamics within the data, leading to higher prediction accuracy. Conversely, as the prediction horizon extends, the temporal correlation diminishes, complicating the model's ability to discern long-term patterns, which in turn impacts the accuracy of the predictions.

4.5 Ablation experiments

The IC-TLCA model's structure is centred around the TCN-LSTM framework and the channel attention mechanism. To validate their effectiveness in our approach, we conducted a series of ablation experiments. Initially, we utilised TCN alone for predictions. Subsequently, we integrated LSTM, employing a TCN-LSTM combination for the predictions. We then incorporated the channel attention mechanism, using it with TCN for further predictions. Following this, we experimented with a fusion of TCN-LSTM and CAM for predictive tasks. Next, we employed the IC-EMD method to decompose the data and use a TCN containing two hidden layers, as part of the TLCA model, to perform the prediction task. We also explored the integration of TCN-LSTM with CAM. In each of these tests, residual connections were not utilised. Otherwise, we also built the IC-TLCA model without a residual connection as a control and contrasted it with our suggested model that had a residual connection.

Table 3 quantitatively shows the MAE, RMSE and MAPE of the seven models on the dataset. Based on the experimental results, we find that adding an LSTM after TCN improves the model performance, which suggests that the LSTM model can efficiently receive and utilise the features extracted by the TCN module (see Experiments {1, 2}).

Subsequent test reveals that channel attention can capture both short- and long-term dependent traits, and that models that use it have higher prediction performance. Refer to the {1, 3, 4} experiments.

Furthermore, the model's prediction accuracy is much enhanced by the use of a decomposition technique (see Experiments {4, 7}). Experiments {5, 7} demonstrate that enhancing the number of TCN layers to four can enhance prediction accuracy by efficiently capturing the aspects of passenger flow data.

Finally, the introduction of residual connectivity can reduce the evaluation metrics and further improve the prediction performance of the model (see Experiments {6, 7}).

Combining these improvements, our proposed model demonstrates excellent performance in all evaluation metrics, outperforming other comparative models.

Table 3 – Differences between models in the ablation experiment

Model	1-day			3-day		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
TCN	4.948	8.972	4.313	5.732	10.391	4.982
TCN-LSTM	4.913	8..909	4.281	5.693	10.519	5.012
TCN-CA	4.893	8.879	4.392	5.636	10.225	4.732

Model	1-day			3-day		
	MAE	RMSE	MAPE	MAE	RMSE	MAPE
TLCA	4.634	8.273	4.028	5.089	9.126	4.505
IC-TLCA·	4.505	8.219	4.053	5.108	9.151	4.271
IC-TLCA··	4.596	8.112	4.047	4.926	8.983	4.363
IC-TLCA	**4.373**	**7.858**	**3.973**	**4.795**	**8.809**	**4.147**
Model	**6-day**			**Average**		
	MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
TCN	6.684	12.412	6.292	5.788	10.592	5.196
TCN-LSTM	6.458	12.396	5.822	5.687	10.608	5.038
TCN-CA	6.574	12.243	5.698	5.701	10.449	4.946
TLCA	6.361	11.425	5.383	5.361	9.608	4.639
IC-TLCA·	6.453	11.209	5.362	5.355	9.526	4.562
IC-TLCA··	6.361	12.059	5.238	5.294	9.718	4.549
IC-TLCA	**6.171**	**11.087**	**5.163**	**5.113**	**9.251**	**4.427**

4.6 Comparison of ICCEMDAN with EMD

We performed comparative experiments in this section using EMD, EEMD and CEEMDAN as benchmark models to validate the efficacy of the IC-EMD decomposition employed. Following the previously outlined methodology, we also employed three evaluation metrics to validate the performance of these models.

Table 4 presents the empirical results. Compared with other decomposition techniques, MAE, RMSE and MAPE values for IC-TLCA are lower. The outcomes demonstrate how much the decomposition approach may increase prediction accuracy. Specifically, IC-EMD outperforms other techniques across three evaluation metrics for 1-day, 3-day and 6-day forecasting horizons. The superiority of IC-EMD is significant, highlighting its utility in reducing prediction error and its critical role in forecasting models.

Table 4 – Prediction based on TLCA model

Model	IMFs	1-day			3-day		
		MAE	RMSE	MAPE	MAE	RMSE	MAPE
EMD	7	4.617	8.152	4.024	5.076	9.013	4.496
EEMD	9	4.609	8.017	4.003	5.049	8.996	4.483
CEEMDAN	7	4.582	7.994	3.991	4.918	8.913	4.262
ICCEMDAN	8	**4.373**	**7.858**	**3.973**	**4.795**	**8.809**	**4.147**
Model	**IMFs**	**6-day**			**Average**		
		MAE	**RMSE**	**MAPE**	**MAE**	**RMSE**	**MAPE**
EMD	7	6.232	11.475	5.379	5.308	9.547	4.633
EEMD	9	6.213	11.392	5.253	5.29	9.468	4.579
CEEMDAN	7	6.197	11.372	5.267	5.232	9.426	4.506
ICCEMDAN	8	**6.171**	**11.087**	**5.163**	**5.113**	**9.251**	**4.427**

5. CONCLUSION

This paper proposes the IC-TLCA, a hybrid deep learning model for urban rail transit passenger flow prediction with a decomposition-prediction technique. The model is bifurcated into two distinct stages: the decomposition phase, wherein the ICEEMDAN method is employed to decompose the passenger flow data into intrinsic mode functions (IMFs) and a residual; and the prediction phase, the TCN extracts local temporal features through multi-scale dilated convolutions, efficiently capturing short-term fluctuations and periodic patterns in passenger flow data. The LSTM models long-term dependencies via its gating mechanism, retaining critical historical events while adapting flexibly to complex nonlinear variations and external disturbances. The multi-scale local features extracted by the TCN are fed into the LSTM to synthesise global temporal patterns, which are further integrated with CAM-based attention weighting to balance accuracy and computational efficiency. This hybrid architecture provides reliable support for real-time scheduling and emergency management in urban rail transit systems. Comparing the prediction results with those of the other seven models, it is evident that the IC-TLCA achieves the highest prediction accuracy. This advancement provides a comprehensive and reliable solution for metro operators, enabling precise identification of travel patterns and dynamic demand fluctuations. The model's implementation facilitates optimised resource allocation in terms of vehicle scheduling, personnel deployment and facility configuration, thereby enhancing passenger travel experience through improved operational efficiency, cost reduction and service quality enhancement. These capabilities not only strengthen the competitiveness of urban rail systems but also contribute significantly to urban sustainable development and the realisation of efficient public transportation networks for residents.

In future work, an efficient method for processing passenger flow data series will be investigated. For instance, to reduce computational costs and increase the efficiency of data analysis, the sliding window technique can be used to segment the complete passenger flow series into a series of equally sized and consecutive subsequences.

ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China (No. 52462045), the Gansu Provincial Natural Science Foundation Key Project (No. 24JRRA221) and the Central guidance for local scientific and technological development funding project (No. 22ZY1QA005).

REFERENCES

[1] Bao J, Kang J, Yang Z, Chen X. Forecasting network-wide multi-step metro ridership with an attention-weighted multi-view graph to sequence learning approach. *Expert Systems with Applications*. 2022;210:18475. DOI: 10.1016/j.eswa.2022.118475.

[2] Liu Y, Liu Z, Jia R. Deep PF: A deep learning based architecture for metro passenger flow prediction. *Transportation Research Part C: Emerging Technologies.* 2019;101:18-34. DOI: 10.1016/j.trc.2019.01.027.

[3] Zhang Y, Zhang Y, Haghani A. A hybrid short-term traffic flow forecasting method based on spectral analysis and statistical volatility model. *Transportation Research Part C: Emerging Technologies.* 2014;43:65-78. DOI: 10.1016/j.trc.2013.11.011.

[4] Ahmed MS, Cook AR. Analysis of freeway traffic time-series data by using Box-Jenkins techniques. 1979.

[5] Vlahogianni EI, Karlaftis MG, Kopelias P. Modeling freeway travel speed across lanes: A vector autoregressive approach. *13th International IEEE Conference on Intelligent Transportation Systems. Funchal, Portugal*, 2010;569-574. DOI: 10.1109/ITSC.2010.5625059.

[6] Vlahogianni EI, Karlaftis MG, Golias JC. Temporal evolution of short-term urban traffic flow: A nonlinear dynamics approach. *Computer-Aided Civil and Infrastructure Engineering*. 2008;23:536-548. DOI: 10.1111/j.1467-8667.2008.00554.x.

[7] Vlahogianni EI, Karlaftis MG. Comparing traffic flow time-series under fine and adverse weather conditions using recurrence-based complexity measures. *Nonlinear Dynamics*. 2012;69:1949-1963. DOI: 10.1007/s11071-012-0399-x.

[8] Wu JX, Zhou XB, Peng Y, Zhao XJ. Recurrence analysis of urban traffic congestion index on multi-scale. *Physica A: Statistical Mechanics and its Applications*. 2022;585:126439. ISSN 0378-4371. DOI: 10.1016/j.physa.2021.126439.

[9] Li XP, Peng F, Ouyang YF. Measurement and estimation of traffic oscillation properties. *Transportation Research Part B: Methodological*. 2010;44:1-14. ISSN 0191-2615. DOI: 10.1016/j.trb.2009.05.003.

[10] Zhao X, Lord D, Peng Y. Examining network segmentation for traffic safety analysis with data-driven spectral analysis. *IEEE Access*. 2019;7:120744-120757. DOI: 10.1109/ACCESS.2019.2937001.

[11] Zheng ZD, Ahn S, Chen DJ, Laval J. Applications of wavelet transform for analysis of freeway traffic: Bottlenecks, transient traffic, and traffic oscillations, *Transportation Research Part B: Methodological*, Volume 45, Issue 2,2011, Pages 372-384, ISSN 0191-2615. DOI: 10.1016/j.trb.2010.08.002.

[12] Yang H, Cheng Y, Li G. A new traffic flow prediction model based on cosine similarity variational mode decomposition, extreme learning machine and iterative error compensation strategy. *Engineering Applications of Artificial Intelligence*. 2022;115:105234. DOI: 10.1016/j.engappai.2022.105234.

[13] Woźniak M, Zielonka A, Sikora A. Diving support by type-2 fuzzy logic control model. *Expert Systems with Applications*. 2022;207:117798. ISSN 0957-4174. DOI: 10.1016/j.eswa.2022.117798.

[14] Ke Q, et al. Deep neural network heuristic hierarchization for cooperative intelligent transportation fleet management. *IEEE Transactions on Intelligent Transportation Systems*. 2022;23:16752-16762. DOI: 10.1109/TITS.2022.3195605.

[15] Han Y, et al. Predicting station-level short-term passenger flow in a citywide metro network using spatiotemporal graph convolutional neural networks. *ISPRS International Journal of Geo-Information*. 2019;8(6):243. DOI: 10.3390/ijgi8060243.

[16] Chen L, Chi Y, Guan Y, Fan J. A hybrid attention-based EMD-LSTM model for financial time series prediction. *2019 2nd International Conference on Artificial Intelligence and Big Data (ICAIBD). Chengdu, China*, 2019: 113-118. DOI: 10.1109/ICAIBD.2019.8837038.

[17] Niu H, Xu K, Wang W. A hybrid stock price index forecasting model based on variational mode decomposition and LSTM network. *Applied Intelligence*. 2020;50:4296-4309. DOI: 10.1007/s10489-020-01814-0.

[18] Hamad K, Shourijeh MT, Lee E, Faghri A. Near-term travel speed prediction utilizing Hilbert–Huang transform. *Computer-Aided Civil and Infrastructure Engineering*. 2009;24:551-576. DOI: 10.1111/j.1467-8667.2009.00620.x.

[19] Wei Y, Chen MC. Forecasting the short-term metro passenger flow with empirical mode decomposition and neural networks. *Transportation Research Part C: Emerging Technologies.* 2012;21:148-162. DOI: 10.1016/j.trc.2011.06.009.

[20] Jiang XS, Zhang L, Chen XQ. Short-term forecasting of high-speed rail demand: A hybrid approach combining ensemble empirical mode decomposition and gray support vector machine with real-world applications in China. *Transportation Research Part C: Emerging Technologies.* 2014;44:110-127. DOI: 10.1016/j.trc.2014.03.016.

[21] Li LC, et al. Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network. *Applied Soft Computing*. 2018;73:921-932. DOI: 10.1016/j.asoc.2018.09.023.

[22] Yang HF, Chen YP. Hybrid deep learning and empirical mode decomposition model for time series applications. *Expert Systems with Applications*. 2019;120:128-138. DOI: 10.1016/j.eswa.2018.11.019.

[23] Zhang S, et al. Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition. *Computer-Aided Civil and Infrastructure Engineering*. 2020;35:1132–1147. DOI: 10.1111/mice.12575.

[24] Chen XQ, et al. Traffic flow prediction by an ensemble framework with data denoising and deep learning model. *Physica A: Statistical Mechanics and its Applications*. 2021;565:125574. DOI: 10.1016/j.physa.2020.125574.

[25] Shahriari S, et al. Ensemble of ARIMA: Combining parametric and bootstrapping technique for traffic flow prediction. *Transportmetrica A: Transport Science*. 2020;16(3):1552-1573. DOI: 10.1080/23249935.2020.1764662.

[26] Milenković M, et al. SARIMA modelling approach for railway passenger flow forecasting. *Transport*. 2018;33(5):1113-1120. DOI: 10.3846/16484142.2016.1139623.

[27] Kumar SV, Vanajakshi L. Short-term traffic flow prediction using seasonal ARIMA model with limited input data. *European Transport Research Review*. 2015;7:21. DOI: 10.1007/s12544-015-0170-8.

[28] Huang WC, et al. Railway dangerous goods transportation system risk identification: Comparisons among SVM, PSO-SVM, GA-SVM and GS-SVM. *Applied Soft Computing*. 2021;109:107541. DOI: 10.1016/j.asoc.2021.107541.

[29] Zhao Y, Ma Z. Naïve Bayes-based transition model for short-term metro passenger flow prediction under planned events. *Transportation Research Record*. 2022;2676(9):309-324. DOI: 10.1177/03611981221086645.

[30] Jing Y, et al. Short-term prediction of urban rail transit passenger flow in external passenger transport hub based on LSTM-LGB-DRS. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(7):4611-4621. DOI: 10.1109/TITS.2020.3017109.

[31] Li L, et al. Prediction modeling of railway short-term passenger flow based on random forest regression. *In: Green Intelligent Transportation Systems: Proceedings of the 8th International Conference on Green Intelligent Transportation Systems and Safety*. Springer Singapore. 2019;867–875. DOI: 10.1007/978-981-13-0302-9_84.

[32] Huang W, et al. Deep architecture for traffic flow prediction: deep belief networks with multitask learning. *IEEE Transactions on Intelligent Transportation Systems*. 2014;15(5):2191-2201. DOI: 10.1109/TITS.2014.2311123.

[33] Lv Y, et al. Traffic flow prediction with big data: A deep learning approach. *IEEE Transactions on Intelligent Transportation Systems*. 2015;16(2):865-873. DOI: 10.1109/TITS.2014.2345663.

[34] Ma XL, et al. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. *Transportation Research Part C: Emerging Technologies*. 2015;54:187-197. DOI: 10.1016/j.trc.2015.03.014.

[35] Sun P, Boukerche A, Tao YJ. SSGRU: A novel hybrid stacked GRU-based traffic volume prediction approach in a road network. *Computer Communications*. 2020;160:502-511. DOI: 10.1016/j.comcom.2020.06.028.

[36] Zhang D, Kabuka M R. Combining weather condition data to predict traffic flow: A GRU-based deep learning approach. *IET Intelligent Transport Systems*. 2018;12:578-585. DOI: 10.1049/iet-its.2017.0313.

[37] Ma D, Song X, Li P. Daily traffic flow forecasting through a contextual convolutional recurrent neural network modeling inter- and intra-day traffic patterns. *IEEE Transactions on Intelligent Transportation Systems*. 2021;22(5):2627-2636. DOI: 10.1109/TITS.2020.2973279.

[38] Ma XL, et al. Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*. 2017;17:818. DOI: 10.3390/s17040818.

[39] Ke JT, et al. Predicting origin-destination ride-sourcing demand with a spatio-temporal encoder-decoder residual multi-graph convolutional network. *Transportation Research Part C: Emerging Technologies*. 2021;122:102858. DOI: 10.1016/j.trc.2020.102858.

[40] Zhang DP, et al. DNEAT: A novel dynamic node-edge attention network for origin-destination demand prediction. *Transportation Research Part C: Emerging Technologies*. 2021;122:102851. DOI: 10.1016/j.trc.2020.102851.

[41] Bi J, et al. A hybrid prediction method for realistic network traffic with temporal convolutional network and LSTM. *IEEE Transactions on Automation Science and Engineering*. 2022;19(3):1869-1879. DOI: 10.1109/TASE.2021.3077537.

[42] Lin L, et al. Channel attention & temporal attention based temporal convolutional network: A dual attention framework for remaining useful life prediction of the aircraft engines. *Advanced Engineering Informatics*. 2024;60:102372. DOI: 10.1016/j.aei.2024.102372.

[43] Chen MC, Chen LS, Wei Y. Apply ensemble empirical mode decomposition to discover time variants of metro station passenger flow. *In: 4th International Conference on Industrial Engineering and Applications (ICIEA). Nagoya, Japan*, 2017: 239–243. DOI: 10.1109/IEA.2017.7939214.

[44] Chen MC, Wei Y. Exploring time variants for short-term passenger flow. *Journal of Transport Geography*. 2011;19(4):488-498. DOI: 10.1016/j.jtrangeo.2010.04.003.

[45] Kim EJ, et al. Spatiotemporal filtering method for detecting kinematic waves in a connected environment. *PLoS ONE*. 2020;15(12):e0244329. DOI: 10.1371/journal.pone.0244329.

[46] Ding C, et al. Using an ARIMA-GARCH Modeling Approach to Improve Subway Short-Term Ridership Forecasting Accounting for Dynamic Volatility. *IEEE Transactions on Intelligent Transportation Systems*. 2018;19(4):1054-1064. DOI: 10.1109/TITS.2017.2711046.

[47] Kashi SO, Akbarzadeh M. A framework for short-term traffic flow forecasting using the combination of wavelet transformation and artificial neural networks. *Journal of Intelligent Transportation Systems*. 2019;23(1):60–71. DOI: 10.1080/15472450.2018.1493929.

[48] Yang X, et al. A novel prediction model for the inbound passenger flow of urban rail transit. *Information Sciences*. 2021;566:347-363. DOI: 0.1016/j.ins.2021.02.036.

[49] Li LC, et al. Travel time prediction for highway network based on the ensemble empirical mode decomposition and random vector functional link network. *Applied Soft Computing*. 2018;73:921-932. DOI: 10.1016/j.asoc.2018.09.023.

[50] Yang HF, Chen Y. Hybrid deep learning and empirical mode decomposition model for time series applications. *Expert Systems with Applications*. 2019;120:128-138. DOI: 10.1016/j.eswa.2018.11.019.

[51] Zhang S, et al. Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition. *Computer-Aided Civil and Infrastructure Engineering*. 2020;35:1132–1147. DOI: 10.1111/mice.12575.

[52] Huang NE, et al. The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis. *Proceedings of the Royal Society of London A: Mathematical*, Physical and Engineering Sciences. 1998;454(903):903–995. DOI: 10.1098/rspa.1998.0193.

[53] Wu Z, Huang NE. Ensemble empirical mode decomposition: a noise-assisted data analysis method. *Advances in Adaptive Data Analysis*. 2009;1(01):1–41. DOI: 10.1142/S1793536909000047.

[54] Yeh JR, Shieh JS, Huang NE. Complementary ensemble empirical mode decomposition: A novel noise enhanced data analysis method. *Advances in Adaptive Data Analysis*. 2010;2(02):135–156. DOI: 10.1142/S1793536910000422.

[55] Zhou F, Huang ZH, Zhang CH. Carbon price forecasting based on CEEMDAN and LSTM. *Applied Energy*. 2022;311:118601. DOI: 10.1016/j.apenergy.2022.118601.

[56] Ali M, et al. Complete ensemble empirical mode decomposition hybridized with random forest and kernel ridge regression model for monthly rainfall forecasts. *Journal of Hydrology*. 2020;584:124517. DOI: 10.1016/j.jhydrol.2020.124647.

[57] Lu X, et al. Factor analysis of financial time series using EEMD-ICA based approach. *Sustainable Futures*. 2020;2:100003. DOI: 10.1016/j.sftr.2019.100003.

[58] Colominas MA, Schlotthauer G, Torres ME. Improved complete ensemble EMD: A suitable tool for biomedical signal processing. *Biomedical Signal Processing and Control*. 2014;14:19–29. DOI: 10.1016/j.bspc.2014.06.009.

[59] Lea C, Vidal R, Reiter A, Hager GD. Temporal convolutional networks: A unified approach to action segmentation. *In: Hua G, Jégou H (eds). Computer Vision – ECCV 2016 Workshops. ECCV 2016. Lecture Notes in Computer Science*, vol 9915. Springer, Cham, 2016. DOI: 10.1007/978-3-319-49409-8_7.

[60] Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Computation*. 1997;9(8):1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[61] Siami-Namini S, Tavakoli N, Namin AS. The performance of LSTM and BiLSTM in forecasting time series. *In: IEEE International Conference on Big Data (Big Data)*. Los Angeles, CA, USA, 2019:3285–3292.DOI: 10.1162/neco.1997.9.8.1735. DOI: 10.1109/BigData47090.2019.9005997.

[62] Hong JC, et al. Multi-forward-step state of charge prediction for real-world electric vehicles battery systems using a novel LSTM-GRU hybrid neural network. *eTransportation*. 2024;20:10032. DOI: 10.1016/j.etran.2024.100322.

[63] Vaswani A. Attention is all you need. *Advances in Neural Information Processing Systems*. 2017. DOI: 10.48550/arxiv.1706.03762.

[64] Zhou H, et al. Informer: Beyond efficient transformer for long sequence time-series forecasting. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. 2021;35(12):11106–11115. DOI: 10.1609/aaai.v35i12.17325.