RESEARCH ARTICLE

Long-term Traffic Flow Prediction using Stochastic Configuration Networks for Smart Cities

Yuqi Lin[1,*]

[1] The Grainger College of Engineering, University of Illinois Urbana-Champaign, Urbana 61801, United States

Abstract

Accurate predictions of traffic flow are very meaningful to city managers. With such information, traffic systems can better coordinate traffic signals and reduce congestion. By understanding traffic patterns, navigation systems can provide real-time routing suggestions that avoid traffic jams, save time, and reduce fuel consumption. However, traffic flow will be interfered with by multiple factors such as collection time and place. In this paper, we propose to use stochastic configuration networks (SCNs) to predict the traffic flow. The network is trained through stepwise construction, and the network parameters are effectively optimized based on the approximation theorem and convergence analysis optimization mechanism. The proposed network automatically adjusts its structure according to the complexity of traffic flow to better adapt to the complex non-linearity of traffic flow. We observed that the proposed model achieves better prediction performance overall and greater flexibility in the length of the prediction period compared to the benchmarks using the Guangzhou urban traffic flow dataset. It's worth noting that SCNs consistently outperform other models across different prediction intervals. They yield RMSE improvements of up to 10.73% for 10-minute predictions, 5.02% for 30-minute predictions, and 11.21% for 60-minute predictions compared to the least effective models. The R-value also exhibits steady enhancement, increasing up to 0.78%, 0.65%, and 2.33% for 10-minute, 30-minute, and 60-minute predictions, respectively. These notable advancements, combined with the model's computational efficiency, especially in short-term predictions, underscore the effectiveness and practicality of SCNs in traffic flow prediction tasks.

Keywords: Traffic flow prediction, time-series data prediction, stochastic configuration network(SCNs), deep learning.

Academic Editor:
Quanmin Zhu

Submitted: 26 August 2024
Accepted: 22 September 2024
Published: 27 September 2024

Vol. 1, **No.** 2, 2024.
10.62762/TIS.2024.952592

***Corresponding author:**
Yuqi Lin
yuqilin2@illinois.edu

1 Introduction

In the era of information, the robust data processing capabilities of deep learning have begun to demonstrate their advantages across various fields, notably within transportation systems. Traffic flow prediction (TFP) stands at the heart of modern transportation studies and is pivotal for the evolution of smart cities [1]. Accurate TFP enables residents to optimize their travel plans, thereby conserving time. While short-term TFP is beneficial, medium and long-term TFP also serves important roles in various scenarios, such as implementing long-term traffic controls to alleviate congestion and accidents [2]. Thus, TFP across all time frames is a critical and valuable area of research [3].

Citation
Lin, Y. (2024). Long-term Traffic Flow Prediction using Stochastic Configuration Networks for Smart Cities. *ICCK Transactions on Intelligent Systematics*, 1(2), 79–90.

© 2024 ICCK (Institute of Central Computation and Knowledge)

The number of factors that can influence traffic speeds is immense, encompassing emergencies, accidental events, and weather conditions. Thanks to the ongoing advancements in deep learning, vast amounts of traffic flow data can be analyzed using computer vision, significantly enhancing the precision and efficiency of TFP.

However, traffic time series data are typically irregular and highly nonlinear. During data collection, for example, traffic sensors may introduce uncertainties, such as abnormal driver behaviors, sensor malfunctions, and extreme weather conditions, which can diminish the performance of traditional statistical methods [4]. Additionally, some models may suffer from overfitting, resulting in poor robustness during the training phase [5]. Moreover, classic deep neural networks face challenges in uncovering intrinsic features and achieving high accuracy when predicting periodic data, such as traffic flow speeds [6].

The complexity of long-term prediction could be more manageable. Unlike short-term forecasts, which often rely on immediate and recent data, long-term predictions must account for many variables that can change over time, including urban development, population growth, and evolving traffic patterns. These long-term factors are inherently more unpredictable and can significantly affect traffic flow. The challenge is compounded by the need to maintain model accuracy over extended periods, which requires the models to be adaptable and resilient to a wide range of future scenarios. Consequently, while the potential benefits of accurate long-term TFP are immense, the difficulties in achieving reliable long-term predictions are substantial, making it a particularly challenging area within the broader field of transportation studies.

With the rise of deep learning, neural networks have been pushed to new heights in recent years. Deep learning methods have a solid ability to learn complex non-linear data and thus can produce better predictions [7].

In recent years, Recurrent neural networks (RNNs) have been widely used in recognizing nonlinear relationships in time series data. Nonetheless, when capturing long-term dependencies using traditional RNNs, the gradient disappearance problem easily occurs. On the other hand, through recent investigation and development, long-short-term memory networks (LSTM) and GRU now gain a relatively higher resistance to this problem [8], so it has been extensively used in traffic prediction.

LSTM and GRU networks offer significant advantages in traffic flow prediction due to their superior ability to handle long-term dependencies. Their key strengths include long-term memory capability, allowing them to capture distant temporal relationships, and selective memory through gating mechanisms, which helps in filtering out noise and irrelevant data. These networks mitigate the vanishing gradient problem, ensuring better gradient flow across time steps. Their adaptability to various time scales makes them ideal for capturing both short-term fluctuations and long-term trends in traffic patterns. Additionally, LSTM and GRU excel in modeling complex non-linear relationships, crucial for understanding intricate traffic dynamics. They demonstrate robustness to noise and outliers in real-world traffic data, enhancing their reliability. Furthermore, their scalability allows for integration with other deep learning techniques, enabling the creation of more sophisticated hybrid models. These combined advantages make LSTM and GRU powerful tools for accurate and long-term traffic flow forecasting, significantly advancing the field of intelligent transportation systems.

Despite their impressive performance, classic deep neural networks often suffer from significant drawbacks related to their architecture and efficiency. A primary concern is their propensity for structural and parametric redundancy. As these networks grow deeper and wider in pursuit of better performance, they frequently accumulate unnecessary neurons, layers, and connections. This redundancy not only increases the computational complexity but also leads to overfitting, where the model memorizes training data rather than learning generalizable patterns. Consequently, the actual running networks become excessively large, demanding substantial computational resources and memory. This size inflation poses challenges for deployment in resource-constrained environments, such as mobile devices or real-time systems, where quick inference is crucial. Moreover, these oversized networks often exhibit diminishing returns in performance improvement relative to their increased complexity. The redundancy also makes the models more challenging to interpret and analyze, hindering efforts to understand their decision-making processes. Addressing these issues of bloated architectures and inefficient parameterization has become a critical focus in the field, spurring research into model compression, pruning techniques, and the development of more efficient network architectures.

Self-Constructing Networks (SCNs) offer several advantages over traditional deep learning models. Firstly, SCNs determine the output weights of hidden layer nodes using a pseudo-inverse operation, eliminating the need for iterative updates through back-propagation. This approach enhances the learning rate and effectively avoids issues such as local optima, vanishing gradients, or gradient explosions. Secondly, SCNs employ incremental modeling techniques and monitoring mechanisms to configure the input weights and biases of hidden layer nodes within a preset parameter range, ensuring a strong general approximation without predefined parameters. In contrast, traditional randomized neural networks rely on the number of hidden layer nodes and the random parameter range, which can lead to overfitting or underfitting if not properly set, reducing the likelihood of accurately approximating the objective function. Consequently, SCNs require less human intervention, avoid time-consuming network structure adjustments, and exhibit superior generalization performance. Compared to other models, SCNs incorporate more automation during the training process, significantly improving overall efficiency.

The rest of the paper is arranged as follows. Section 2 introduces recent research results on time series data prediction, especially in traffic flow prediction. Since this research mainly focuses on classical deep networks, we will introduce these deep learning networks. In particular, we will focus on the baseline model of this paper's experiments. Section 3 describes the general algorithm and construction process of the models. Section 4 exhibits the experimental results and analysis. It shows that compared with other baseline methods, the proposed model has good prediction performance and has a better adaptability in all terms of prediction. The model's validity is proven by conducting practical demonstrations on the Guangzhou urban traffic flow data set. Ultimately, we conclude the article with future research directions and objects in Section 4.

2 Related Work

In time series prediction, various models have been employed to capture temporal dependencies and make accurate forecasts. Linear models [9] offer simplicity and interpretability, suitable for basic trends but limited in capturing complex patterns. Recurrent Neural Networks (RNNs) introduced the ability to process sequential data, retaining information from previous time steps. However, RNNs often struggle with long-term dependencies due to vanishing gradients. Long Short-Term Memory (LSTM) networks [10] further refined this approach with additional gates, providing even greater control over information flow and memory retention. Meng [11] append dynamic time-warping model into LSTM to outperform the traditional LSTM in traffic prediction. Chen [12] introduced a hybrid traffic flow prediction model based on LSTM and Sparse Auto-Encoder, which significantly reduces the computation complexity in TFP by achieving a high compression ratio for high-dimensional traffic data. Theoretically, due to issues in long-term predictions, the performance of the above deep neural networks would have better performance for the short term.

Gated Recurrent Units (GRUs) [13] addressed this limitation by incorporating gating mechanisms, allowing for better long-term memory and more efficient training. Both GRU and LSTM have shown remarkable success in capturing intricate temporal patterns and long-term dependencies in various time series prediction tasks, including traffic flow forecasting, financial market analysis, and natural language processing applications.

While traditional RNN series models fix the size of input and output sequences, the encoder-decoder network breaks through this limitation. The encoder-decoder is a sequence-to-sequence structure using deep neural networks. It is effective in features extraction for time series data [14], but when the size of the input information increases, later information overlays the earlier one, which leads to the missing of earlier information during long-term prediction, causing the decline of prediction accuracy [15]. The attention mechanism [16] is introduced to resolve the issue by assigning different attention weights to all time steps to enhance more important time frames while suppressing others. Currently, attention mechanisms have been widely used in time series prediction. Jin et al. [17] combined bidirectional LSTM networks with wavelet decomposition and attention mechanisms to perform a better prediction of the temperature and humidity of a smart greenhouse. Qiu et al. [18] designed an event-aware graph attention fusion network to effectively capture the spatiotemporal dependencies for TFP, including event impacts, in road networks, which improves the accuracy of TFP by a lot.

Kao et al. [19] explored a Long-Short-Term Memory (LSTM)- based encoder-decoder framework for multi-step-ahead flood forecasting. This approach demonstrated the capability to capture complex temporal dependencies, making it particularly suitable for long-term predictions in hydrological systems. However, the model's reliance on extensive training data and computational resources, coupled with potential instability in extreme events or anomalous patterns, presents challenges for widespread implementation.

Du et al. [20] proposed an attention-based encoder-decoder framework for multivariate time series forecasting. By incorporating attention mechanisms, their model enhances the ability to focus on relevant input features, potentially improving prediction accuracy. While this approach shows promise in handling complex multivariate data, the added complexity of the attention mechanism may increase model training difficulty and the risk of over-fitting in simpler scenarios, necessitating careful consideration of the trade-off between model sophistication and practical applicability.

In the realm of smart agriculture, Kong et al. [21] developed BMAE-Net, a data-driven weather prediction network. This model integrates various deep-learning techniques to enhance forecasting precision for agricultural applications. While BMAE-Net showcases the potential of hybrid models in specialized domains, it may suffer from over-parameterization, leading to reduced interpretability. Furthermore, the generalizability of such complex models across diverse geographical locations and climate conditions remains a concern, highlighting the need for extensive validation in varied environments.

From the above literature, we can conclude that the classical network is still the mainstream network structure for time series prediction. As we know, classic deep neural networks, while powerful, often suffer from structural and parametric redundancy as they grow larger. This redundancy increases computational complexity, leads to over-fitting, and results in excessively large models that are challenging to deploy in resource-constrained environments. The oversized networks show diminishing returns in performance relative to their increased complexity and are difficult to interpret.

This paper proposes a predictor based on stochastic configuration networks(SCNs), which omits the backpropagation process, to perform traffic speed prediction. It aims to overcome gradient-related problems and improve the efficiency and accuracy of the prediction.

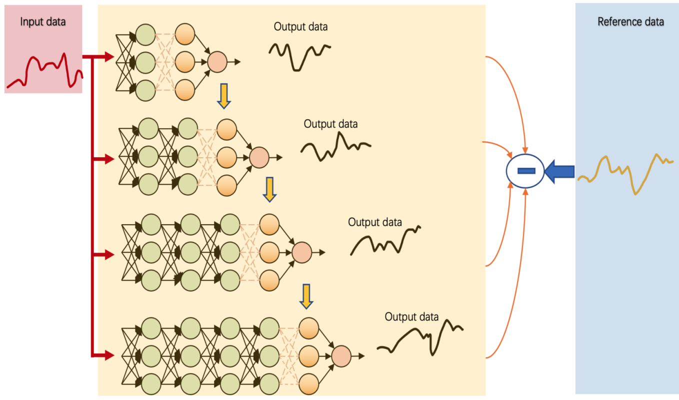

Figure 1. Basic Structure of SCNs.

The contribution of this paper is:

1. Application of Stochastic Configuration Networks (SCNs) to traffic flow prediction: The paper introduces a novel approach to predicting traffic flow using SCNs, which is a departure from traditional methods in this field.

2. Adaptive network structure based on traffic flow complexity: The proposed network has the capability to automatically adjust its structure according to the complexity of the traffic flow data. This adaptive feature allows the model to better handle the non-linear nature of traffic flow patterns.

3. Improved performance and flexibility in prediction length: The model demonstrates superior overall prediction performance compared to benchmark methods. Additionally, it offers greater flexibility in terms of the length of the prediction period, which is a significant advantage in traffic flow forecasting applications.

The following paper presents an experiment comparing SCN with the above models. They are selected because they have been used in traffic flow prediction multiple times, so the information about their structure (code) is easily approachable, and their prediction results are reliable due to constant verification. Thus, the above models creates a good baseline for comparison in testing the performance of SCN.

3 Methodology

This paper proposes stochastic configuration networks (SCNs), an innovative deep learning network that determines the parameters of hidden layer nodes by monitoring a mechanism to maintain fast convergence. It has a relatively faster learning speed and requires less human intervention, which fits the purpose of traffic flow prediction well. The detailed structure of SCNs will be introduced in the following paragraphs.

3.1 Basic Structure

The SCN's network structure is similar to that of a traditional single hidden layer feedforward neural network (SFLN), as shown in Figure 1, which includes an input layer, a hidden layer, and an input layer [22].

The network construction process follows a progressive, data-dependent supervision approach, beginning with a single hidden layer node. Initially, the input weight and bias of this node are randomly initialized, providing a starting point for the network's learning by breaking symmetry and enabling exploration of the parameter space. As the network develops, additional hidden layer nodes are incrementally introduced, allowing the network's complexity to adapt to the nuances of the input data. Each new node's addition is informed by the data-dependent supervision mechanism, ensuring meaningful contributions to the network's predictive performance. The input weights and biases of these newly added nodes are also randomly initialized, preserving the network's ability to capture diverse patterns and avoid local optima throughout its construction.

Crucially, after each addition of a new hidden layer node, the output weights of the entire network are recalculated using the least squares method. This step is vital for optimizing the network's performance. The least squares method ensures that the output of the network minimizes the sum of the squared differences between the predicted and actual values in the training data.

This iterative process of adding nodes and recalculating weights continues until a predefined stopping criterion is met. This criterion could be based on factors such as the network's performance, its size, or computational constraints.

By employing this stepwise construction method, the network can effectively adapt its structure to the complexity of the problem at hand, potentially leading to improved performance and efficiency compared to networks with fixed architectures.

3.2 Algorithm in Network Construction

The main part of constructing SCNs involves the algorithms of adding hidden layer nodes. The following calculation illustrates the process of creating a single new node in the hidden layer to reduce the error of prediction. This process will repeat as long as the number of nodes does not exceed the maximum node amount (the hyper-parameters).

For instance, a training data set $\{X, Y\}$ has features $X = \{x_1, x_2, \ldots, x_N\}$, $x_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\} \in \mathbb{R}^d$ and tag data $Y = \{y_1, y_2, \ldots, y_N\}$, $y_i = \{y_{i,1}, y_{i,2}, \ldots, y_{i,d}\} \in \mathbb{R}^m$, $i = 1, 2, \ldots, N$. We suppose now the hidden layer of SCNs has $L-1$ nodes.

First, the output of the network can be calculated by:

$$f_{L-1}(X) = \sum_{j=1}^{L-1} \beta_j g_j(w_j^T X + b_j), \quad L = 1, 2, \ldots, L_{\text{max}} \quad f_0 = 0 \tag{1}$$

where f_{L-1} denotes the output function of $L-1$ node; L_{max} denotes the upper limit of the number of nodes in the hidden layer; β_j denotes the output weight of hidden layer node j; g_j denotes the activation function, w_j and b_j denote the input weight and the bias of hidden layer node j.

Second, we calculate the network residual vector e_{L-1} by:

$$e_{L-1} = f - f_{L-1}(X) = \begin{bmatrix} e_{L-1,1}(X) \\ e_{L-1,2}(X) \\ \vdots \\ e_{L-1,m}(X) \end{bmatrix} \in \mathbb{R}^{N\times m} \tag{2}$$

If $\|e_{L-1}\|^2$ does not meet the expected error ε and the number of hidden layer node L does not reachL_{max}, we can add the L^{th} hidden layer node to reduce the error. Hidden layer node L's input weight h_L and bias $\xi_{L,q}$ are related to following equations:

$$h_L = \begin{bmatrix} g_L(w_L^T X_1 + b_L) \\ g_L(w_L^T X_2 + b_L) \\ \vdots \\ g_L(w_L^T X_N + b_L) \end{bmatrix} \in \mathbb{R}^N \tag{3}$$

$$\xi_{L,q} = \frac{\left(e_{L-1,q}^T \cdot h_L\right)^2}{h_L^T h_L} - (1 - r - \mu_L)\|e_{L-1}\|^2, \quad q = 1, 2, \ldots, m \tag{4}$$

where h_L denotes the output of hidden layer node L, w_L and b_L are the candidates parameters, $r \in (0,1)$, $\{\mu_L\}$ is a non-negative real number sequence that satisfies $\mu_L \leq 1-r$, $\lim_{L\to+\infty}\mu_L = 0$. We take the candidates parameters w_L, b_L that satisfies $\sum_{q=1}^{m}\xi_{L,q} \geq 0$ and maximize ξ_L as the parameters for hidden layer node L.

Next, the hidden layer node L's output weight,β_L , can be determined by:

$$\beta_L = arg\min_{\beta} \|H\beta - Y\|^2 = H^+Y \tag{5}$$

where $H = [h_1, h_2, \ldots, h_L]$, and H^+ denotes the Moore-Penrose generalized inverse of.

Finally, we calculated the output f of the network :

$$f = H\beta \tag{6}$$

Through the above process, a new hidden layer node is added to the network. This process is repeated again and again until the number of hidden layer node reaches L_{max} or the error is smaller than the expected error ε.

3.3 Approach Ability Proof

Wang [23] provided the proof of the approach ability of SCNs model. Suppose $\Gamma = \{g_1, g_2, g_3, \ldots\}$ is a set of real-valued functions. Span(Γ) is the function space of Γ. Span(Γ) is denser than L_2 and $\forall g \in \Gamma, 0 < \|g\| < b_g, b_g \in R^+$.

For $L = 1,2,3,\ldots$, we define $\delta_L = \sum_{q=1}^{m}\delta_{L,q}$, $\delta_{L,q} = (1-\gamma-\mu_L)\|e_{L-1,q}\|^2$, where $\gamma \in (0,1)$, $\mu_L 1-\gamma, \lim_{L\to+\infty}\mu_L = 0$.

If function g_L satisfies:

$$\langle e_{L-1,q}, g_L\rangle^2 \geq b_g^2\delta_{L,q}, q = 1,2,\ldots,m \tag{7}$$

and the output weight β satisfies:

$$\beta = [\beta_1, \beta_2, \ldots, \beta_L] = argmin_{\beta}\left\| f - \sum_{j=1}^{L}\beta_j g_j \right\| \tag{8}$$

Then,

$$\lim_{L\to+\infty}\|f - f_L\| = 0.$$

When we constructing the SCNs network in the previous part, all requirements listed above are satisfied, which means as we gradually increase the size of L, the residual will gradually reduce. The technique of SCN is thus proven to be valid.

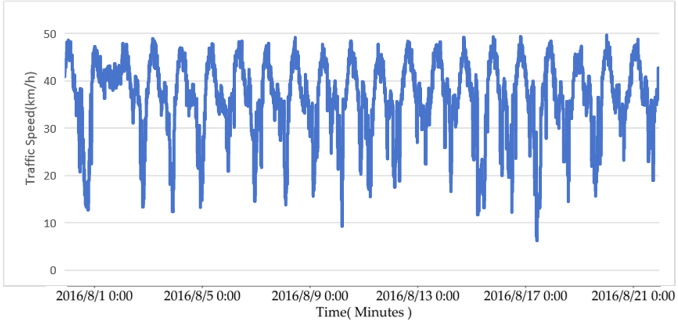

Figure 2. Partial traffic flow time series data.

4 Experimental And Result Analysis

4.1 Data set

We use the Guangzhou urban traffic flow data set(https://zenodo.org/record/1205229) [24] due to its well-processed integrated data and availability. As shown in Figure 2, the data set includes the traffic speeds on 121 different urban roads from August 1st to September 30th in 2016, with a miss rate of 1.29%. To avoid the influence of missing data on the prediction, four roads without any missing data are selected to train and test the models. Data from the first 48 days were used as training samples for the model, and data from the next 13 days were used as testing samples.

4.2 Evaluation Metrics

The experiments used root mean squared error (RMSE), symmetric mean absolute percentage error (SMAPE), and Pearson's correlation coefficient R as the indexes and indicators for evaluating the model. RMSE quantifies the difference between the predicted traffic speed of a model and the actual observed traffic speed in the data. A smaller RMSE value represents a smaller deviation, showing that the model performs better prediction. R measures the correlation between the prediction and actual values. The closer R is to 1, the maximum, the better the regression line fits the actual values, and the closer the patterns from the prediction and the actual values are. SMAPE is a statistical measure used to assess the accuracy of predictions in time series forecasting. The formulas of these indicators are listed as follows.

$$\text{RMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2} \tag{9}$$

$$\text{SMAPE} = \frac{100\%}{m}\sum_{i=1}^{m}\frac{|\hat{y}_i - y_i|}{\left(\frac{|\hat{y}_i|+|y_i|}{2}\right)} \tag{10}$$

Table 1. Evaluation Of Different Models.

Model	10 mins				30 mins				60 mins			
	RMSE	SMAPE	R	Time(s)	RMSE	SMAPE	R	Time(s)	RMSE	SMAPE	R	Time(s)
Linear[9]	2.171	5.223	0.9686	7.439	3.149	7.022	0.9332	7.360	4.152	9.096	0.8805	7.010
RNN[13]	2.159	5.222	0.9690	27.536	3.012	6.829	0.9391	34.773	4.071	9.027	0.8855	33.936
GRU[25]	2.348	5.830	0.9632	90.971	3.020	6.917	0.9386	86.998	3.958	8.876	0.8921	86.807
LSTM[10]	2.237	5.520	0.9667	24.699	3.133	6.992	0.9337	24.272	3.899	8.966	0.8944	24.402
En-Decoder[19]	2.128	5.157	0.9699	37.493	3.012	6.946	0.9389	37.131	3.922	8.756	0.8942	38.684
Attention[20]	2.254	5.511	0.9662	56.538	3.027	6.918	0.9383	56.745	3.998	8.959	0.8898	55.685
Mhatt[21]	2.358	5.833	0.9629	86.340	3.115	7.186	0.9345	84.237	4.388	9.729	0.8742	83.014
Proposed SCNs	**2.105**	**5.145**	**0.9704**	**12.858**	**2.991**	**6.932**	**0.9393**	**31.304**	**3.896**	**8.965**	**0.8946**	**58.681**

$$R = \frac{\sum_{t=1}^{T}(\hat{y}_t - \bar{\hat{y}}_t)(y_t - \bar{y}_t)}{\sqrt{\sum_{t=1}^{T}(\hat{y}_t - \bar{\hat{y}}_t)^2 \sum_{t=1}^{T}(y_t - \bar{y}_t)^2}} \tag{11}$$

where y_i is the actual traffic speed, $\hat{y}_i$ is the prediction, m represents the number of samples, $\bar{y}_i$ is the average of the actual traffic speed, and $\bar{\hat{y}}_i$ is the average of the predicted speed.

Furthermore, the time each model took to complete the training and testing process is recorded. By doing that, we can better compare the efficiency of different models and create a baseline to compare the complexity of SCNs, which have a unique structure due to no back-propagation steps, with other models. With all calculations done on the same device, we expect that the time consumed is directly related to the energy consumption of the model.

Also, in the training process of some models, data is normalized, which leads to inconsistent results. Therefore, after training the training process, data de-normalized before the calculation of the above three indexes is calculated.

4.3 Comparison and Analysis

We compared the proposed model with Linear [9], RNN [13], GRU [25], LSTM [10], En-decoder [19], Attention [20], Multi-head attention (Mhatt) [21].

The training parameters of the model are set as follows: the number of iterations is 100, the optimizer is Adam, and the learning rate is 0.001. The number of the neural network layer is 2, and the units per layer are 64. The batch size is 12, the number of encoder and decoder layers is 1, and the number of attention heads is 2. For SCNs, the maximum number of hidden layer nodes is 100.

All models were written in a Python 3.8 environment based on the Pytorch deep learning framework. All experiments were done on a server with the following parameters: Ubuntu 20.04 bit-64 operating system; Intel Core i7-13620H processor CPU 2.4GHz; NVIDIA GTX1080Ti 11G. Each model was repeated 5 times independently and the result from the best trial(high R, low RMSE, low SMAPE, relatively) is recorded. The evaluation of model prediction performance is performed using the evaluation metrics mentioned in Section 3.2.

1) Comparing Different Prediction Intervals:

We use the proposed SCNs to perform short-, medium-, and long-term (10mins/30mins/60mins) traffic speed predictions. Different models' performances are compared in Table 1. According to the result, SCNs outperform baseline models in each index most of the time. Relatively low RMSE and SMAPE values indicate SCNs have a small difference between actual and predicted values, and high R indicates the SCNs' predicted value and the actual value have a high goodness-of-fit. Specifically, in Figure 3, in the short-term (10-minute) prediction, SCNs outperform every other model by having the lowest RMSE of 2.105, the lowest SMAPE of 5.145, and the highest R of 0.9704. It also consumed the second shortest time, 12.858s, to complete the prediction. In the 30-minute prediction, SCNs have the lowest RMSE of 2.991 and the highest R of 0.9393. SCNs have a SMAPE of 6.932. The proposed SCNs demonstrate significant advantages over other models across various prediction time intervals. In short-term predictions (10 minutes), SCNs achieve the lowest RMSE of 2.105, outperforming the next-best model (En-Decoder) by 1.08% and the worst-performing model (Mhatt) by 10.73%. It also shows the highest R-value of 0.9704, marginally surpassing the En-Decoder model by 0.05% and significantly exceeding the Mhatt model by 0.78%. The SMAPE value of 5.145 for SCNs is only slightly higher than the Linear model but lower than all other models, representing a 1.49% improvement over the Linear model. For medium-term predictions (30 minutes),

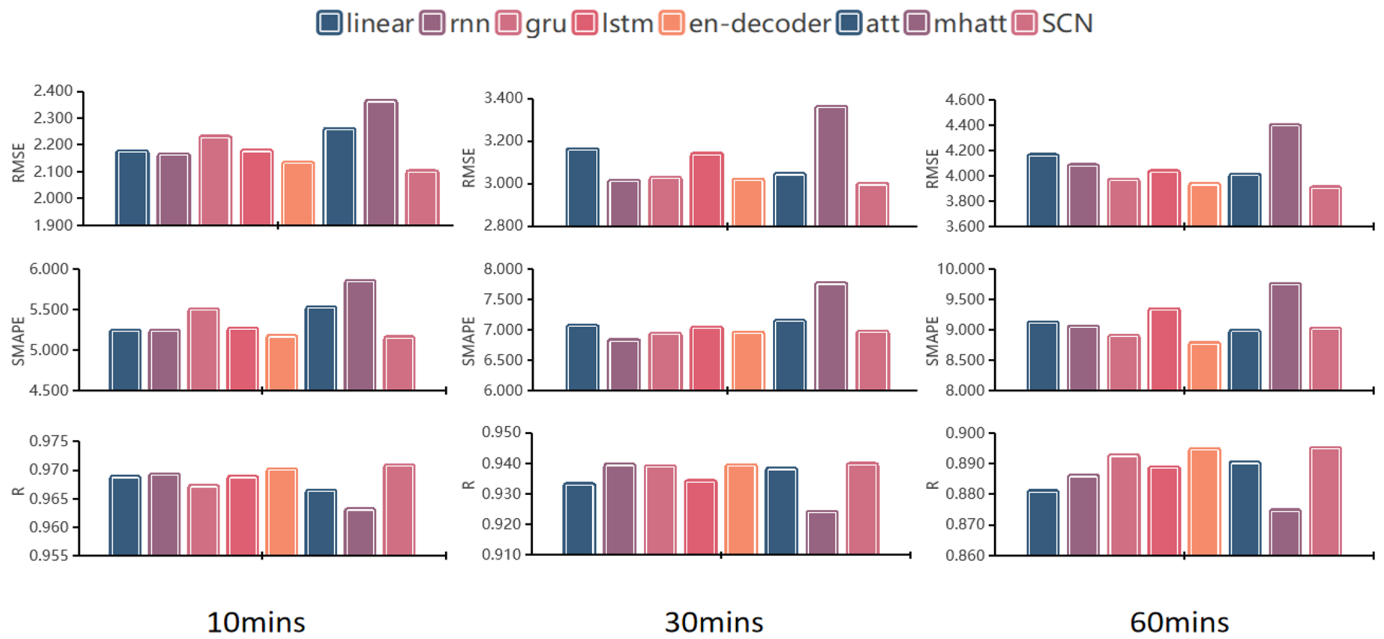

Figure 3. Traffic speed prediction evaluation index in different indicators.

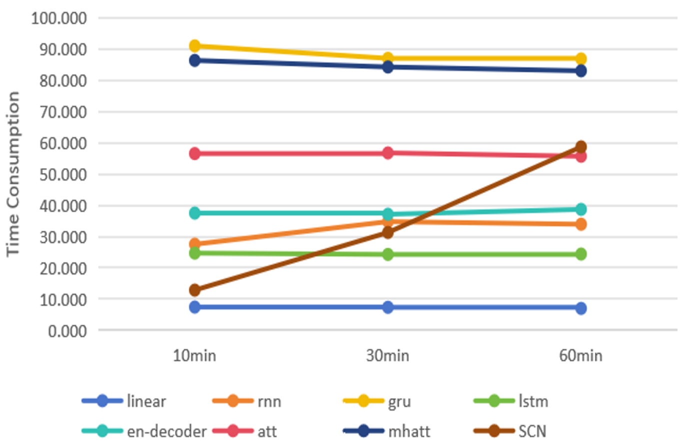

Figure 4. Time consumed(complexity) of different models in different terms of forecasting.

SCN maintains its strong performance. It shares the lowest RMSE of 2.991 with the RNN model, showing a 0.96% improvement over the next best model (GRU) and a 5.02% improvement over the worst-performing model (Linear). SCNs also achieve the highest R-value of 0.9393, slightly outperforming the RNN model by 0.02% and showing a more substantial 0.65% improvement over the Linear model.

In long-term predictions (60 minutes), SCNs demonstrate their superior predictive capabilities. It achieves the lowest RMSE of 3.896, marginally better than the LSTM model by 0.08% but significantly outperforming the worst model (Mhatt) by 11.21%. The R-value of 0.8946 is the highest among all models, showing a slight improvement of 0.02% over the LSTM model and a more substantial 2.33% improvement over the Mhatt model.

It's worth noting that while SCNs consistently perform well in terms of RMSE and R-value, their SMAPE values are generally in the mid-range compared to other models. However, this is offset by its outstanding performance in other metrics. Additionally, SCNs demonstrate good computational efficiency, especially in short-term predictions, where they process data much faster than most complex models. This combination of accuracy and efficiency makes SCNs a highly practical and effective model for traffic flow prediction across various time intervals.

More importantly, SCNs have overall better flexibility in different lengths of prediction periods. To be specific, in the short-term (10 min) prediction, the linear model performs better than most models, except SCNs. However, in long-term (60 min) prediction, it becomes one of the worst models. On the other hand, the GRU model seems to be the worst model in short-term prediction, but it performs well in the long term. One explanation is the complexity of models. GRU's high complexity can be proven by long training and testing time. Such complex models may overfit when the data are relatively simple(short-term prediction has 12 input variables and only 1 output variable). Also, simple models like linear models cannot fit complex conditions, such as 60-minute long-term prediction(12 input variables and 6 output variables).

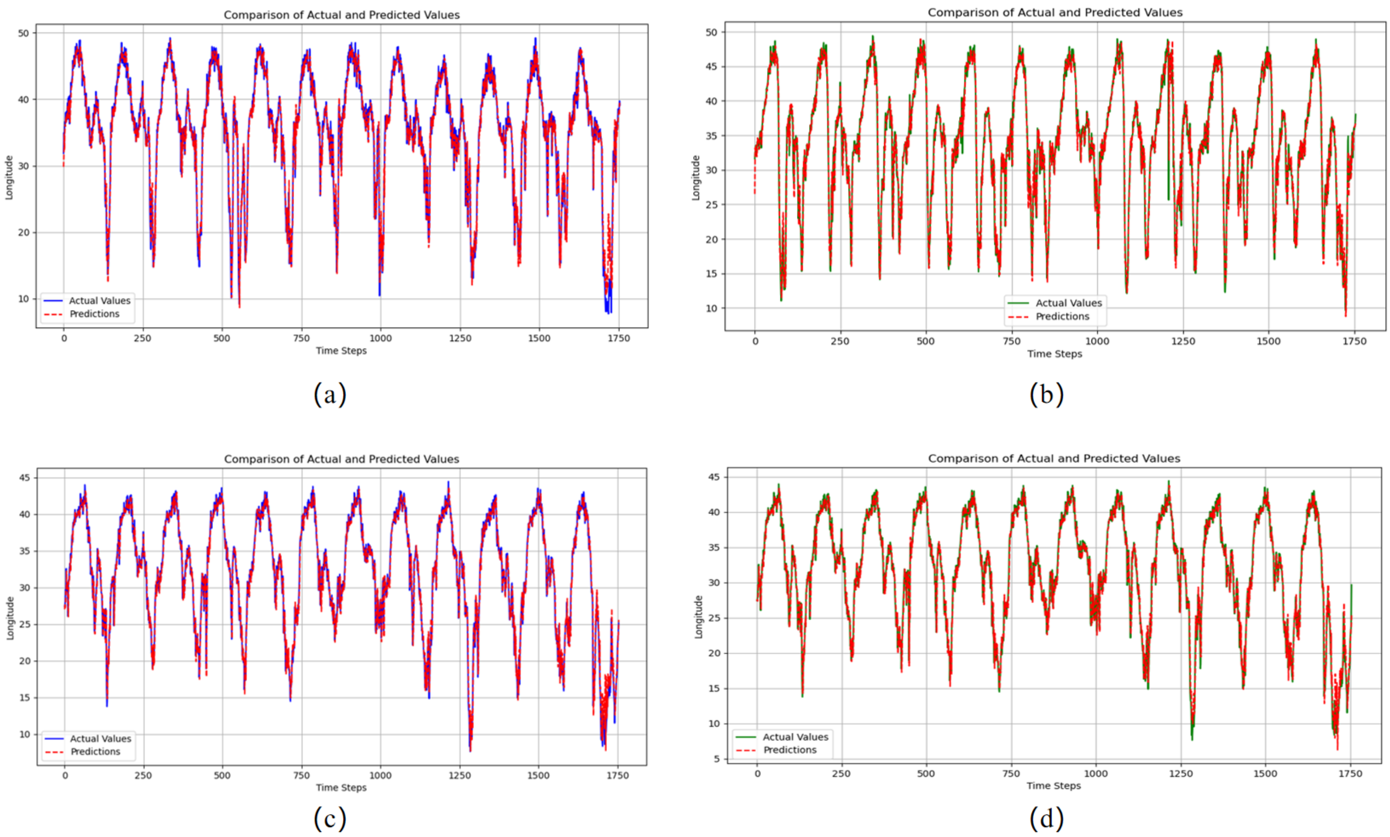

Figure 5. Prediction result of traffic flow in different Roads for 10mins prediction. (a) Road 1, (b) Road 2, (c) Road 3, (d) Road 4.

From Figure 4, it can be seen that the time each model consumed in different terms of prediction is similar, meaning different numbers of output values do not change the models, complexity a lot. However, SCNs took different lengths in different terms, so they will not waste sources in simple conditions and can also fit complex conditions. While other models can only achieve this by re-adjusting hyper-parameters, SCNs are highly automated, and at the same time, highly accurate.

2) Robust Performance Under Different Roads:

In our research, we concentrated on predicting traffic flow for four specific roads in Guangzhou, a major city in southern China renowned for its intricate urban traffic patterns. The selection of these four roads was deliberate, aiming to encompass a wide range of traffic conditions and road characteristics typical of a bustling metropolitan area.

Table 2 in our research paper provides a comprehensive breakdown of the prediction results for each of the four selected roads, segmented by different time periods or terms. The evaluation indexes are derived from the average of 5 separate trials. This detailed presentation offers a nuanced understanding of the model's performance across various temporal scales and road-specific conditions. Upon analyzing these results, we observed that the RMSE indicators consistently remained at low levels across all four roads and different prediction terms. Additionally, we noted that the correlation coefficients (R indexes) were uniformly high across all scenarios, complementing the RMSE findings. This high level of accuracy and correlation remains consistent across different roads, demonstrating the model's robustness and adaptability to varying road conditions and traffic patterns.

Table 2. Prediction Performance for Different Roads.

Model	10 mins		30 mins		60 mins	
	RMSE	R	RMSE	R	RMSE	R
Road 1	2.105	0.9704	2.991	0.9393	3.896	0.8946
Road 2	1.969	0.9739	2.809	0.9462	3.701	0.9047
Road 3	1.626	0.9779	2.170	0.9603	2.835	0.9313
Road 4	1.546	0.9801	2.172	0.9603	2.808	0.9326

To provide a more intuitive understanding of our results, we have included visual representations in Figures 5. Each of these figures corresponds to one of the four selected roads, offering a graphical

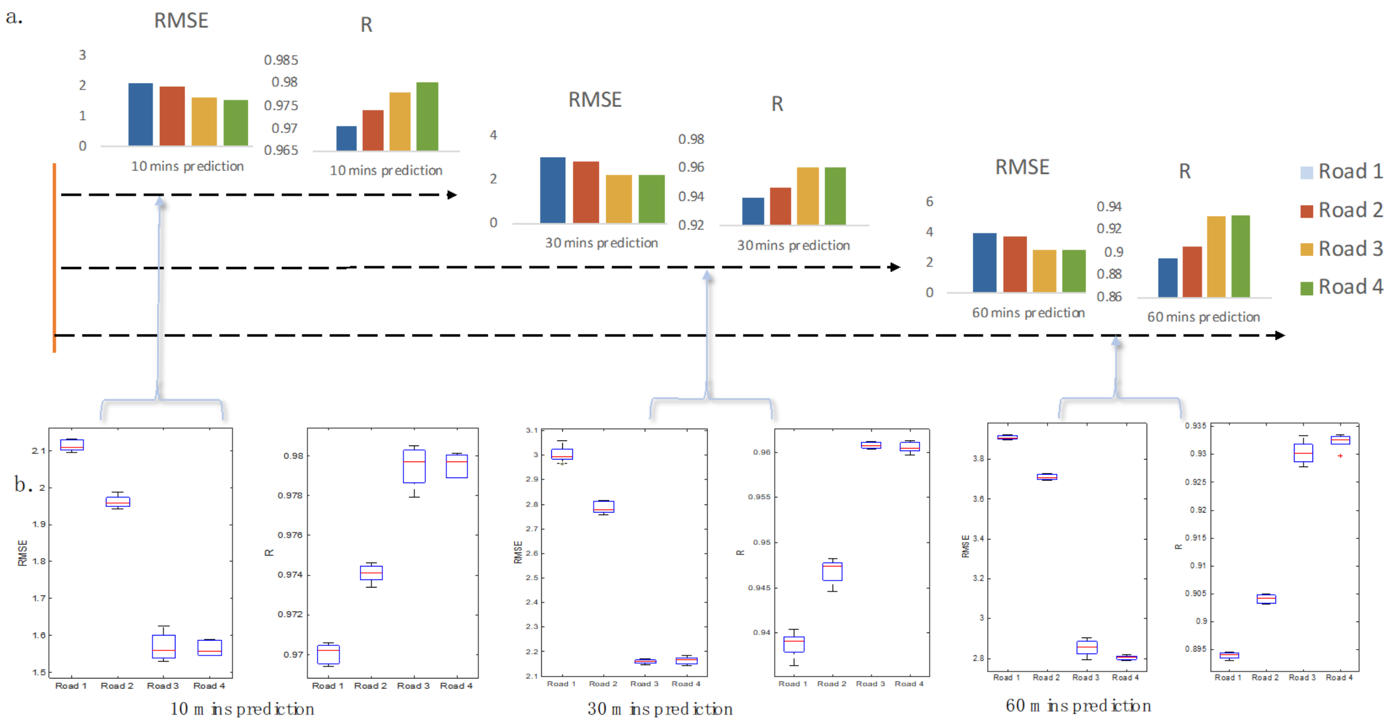

Figure 6. Performance for 10, 30, and 60 mins prediction for different roads: a. the RMSE and R, b. Box plots.

comparison between the predicted and actual traffic flow curves. Upon examination of these figures, it's evident that the predicted curves align closely with the actual traffic flow curves for the majority of the time periods. This visual confirmation reinforces our statistical findings and allows for easy interpretation of the model's performance.

The box plots in Figure 6 demonstrate the strong performance and applicability of our method across different roads and prediction time intervals. Here's a summary of the findings:

For short-term predictions (10 minutes), all roads show low RMSE values (ranging from 1.529 to 2.132) and R values very close to 1 (between 0.9694 and 0.9805), indicating high prediction accuracy. Notably, Road3 and Road4 perform exceptionally well, with average RMSE values of 1.5688 and 1.5644 and average R values of 0.97944 and 0.97952, respectively.

For medium-term predictions (30 minutes), RMSE values increase slightly but remain within acceptable limits (ranging from 2.144 to 3.060), and R values remain high (between 0.9364 and 0.9613). Road3 and Road4 continue to perform well, with average RMSE values of 2.1594 and 2.1642 and average R values of 0.96076 and 0.96054, respectively.

Even in long-term predictions (60 minutes), our method maintains good prediction performance. Although RMSE values increase further (ranging from 2.791 to 3.923), R values remain at a good level (between 0.8931 and 0.9335). Road3 and Road4 stand out again, with average RMSE values of 2.8538 and 2.8048 and average R values of 0.9303 and 0.9323, respectively.

In summary, our proposed method demonstrates excellent applicability and stability across all tested roads, from short-term to long-term predictions. It maintains a high level of prediction accuracy, particularly in the more challenging long-term prediction tasks, proving its effectiveness and robustness.

5 Conclusion

In this paper, the stochastic configuration network, an innovative deep learning network that determines the parameters of hidden layer nodes by monitoring mechanism, is proposed to predict traffic speed. The model is tested on the Guangzhou urban traffic flow data set. Compared to baseline models, it has better prediction performance and more automation. Under short, medium, and long-term predictions, it achieves outstanding results with the R evaluation indexes of 0.9704, 0.9393, and 0.8946 and low RMSE of 2.105, 2.991, and 3.896, respectively. Also, the flexibility that allows the variation in models' complexity in different conditions can avoid wasting computing source and save the time of adjusting hyper-parameters.

However, the current SCN model used has limited capability in feature extraction, which may affect the learning and generalization performance of the model. Also, as a new deep learning model, SCN lacks a theoretical basis, so the construction of the network does not focus much on the rate of converging network residuals. This may influence the learning rate and have potential issues of over-fitting.

In future work, besides basic SCNs, more developed SCN models, such as DeepSCNs, 2D SCNs, or robust SCNs, could be applied to the same field. These advanced SCNs have better computing power and can handle more complicated traffic data set for longer-term predictions. It is also applicable to involve the attention mechanism in SCN so it gains better feature extraction ability. While keeping the flexibility of SCNs, these models may achieve better accuracy, which further assists traffic flow prediction and smart city construction. Besides traffic flow predictions, many studies show that SCNs also have great potential in image classification, face recognition, and medical data analysis due to their unique benefits. With future development, SCNs may be of great help to the above fields.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Funding

This work was supported without any funding.

References

[1] Ma, C., Zhao, Y., Dai, G., Xu, X., & Wong, S. C. (2022). A novel STFSA-CNN-GRU hybrid model for short-term traffic speed prediction. *IEEE Transactions on Intelligent Transportation Systems*, 24(4), 3728-3737. [CrossRef]

[2] Zhang, J., Cui, Y., & Ren, J. (2022). Dynamic mission planning algorithm for UAV formation in battlefield environment. *IEEE Transactions on Aerospace and Electronic Systems*, 59(4), 3750-3765. [CrossRef]

[3] Ma, C., Dai, G., & Zhou, J. (2021). Short-term traffic flow prediction for urban road sections based on time series analysis and LSTM-BILSTM method. *IEEE Transactions on Intelligent Transportation Systems*, 23(6), 5615-5624. [CrossRef]

[4] Kong, J., Yang, C., Wang, J., Wang, X., Zuo, M., Jin, X., & Lin, S. (2021). Deep-stacking network approach by multisource data mining for hazardous risk identification in IoT-based intelligent food management systems. *Computational Intelligence and Neuroscience*, 2021(1), 1194565. [CrossRef]

[5] Jiang, R., Han, S., Yu, Y., & Ding, W. (2023). An access control model for medical big data based on clustering and risk. *Information Sciences*, 621, 691-707. [CrossRef]

[6] Tian, Y., & Pan, L. (2015, December). Predicting short-term traffic flow by long short-term memory recurrent neural network. In *2015 IEEE international conference on smart city/SocialCom/SustainCom (SmartCity)* (pp. 153-158). IEEE. [CrossRef]

[7] Jin, X. B., Wang, Z. Y., Kong, J. L., Bai, Y. T., Su, T. L., Ma, H. J., & Chakrabarti, P. (2023). Deep spatio-temporal graph network with self-optimization for air quality prediction. *Entropy*, 25(2), 247. [CrossRef]

[8] Jin, X., Zhang, J., Kong, J., Su, T., & Bai, Y. (2022). A reversible automatic selection normalization (RASN) deep network for predicting in the smart agriculture system. *Agronomy*, 12(3), 591. [CrossRef]

[9] Lydia, M., Kumar, S. S., Selvakumar, A. I., & Kumar, G. E. P. (2016). Linear and non-linear autoregressive models for short-term wind speed forecasting. *Energy conversion and management*, 112, 115-124. [CrossRef]

[10] Yang, B., Sun, S., Li, J., Lin, X., & Tian, Y. (2019). Traffic flow prediction using LSTM with feature enhancement. *Neurocomputing*, 332, 320-327. [CrossRef]

[11] Meng, X., Fu, H., Peng, L., Liu, G., Yu, Y., Wang, Z., & Chen, E. (2020). D-LSTM: Short-term road traffic speed prediction model based on GPS positioning data. *IEEE Transactions on Intelligent Transportation Systems*, 23(3), 2021-2030. [CrossRef]

[12] Chen, C., Liu, Z., Wan, S., Luan, J., & Pei, Q. (2020). Traffic flow prediction based on deep learning in internet of vehicles. *IEEE transactions on intelligent transportation systems*, 22(6), 3776-3789. [CrossRef]

[13] Connor, J. T., Martin, R. D., & Atlas, L. E. (1994). Recurrent neural networks and robust time series prediction. *IEEE transactions on neural networks*, 5(2), 240-254. [CrossRef]

[14] Jaitly, N., Le, Q. V., Vinyals, O., Sutskever, I., Sussillo, D., & Bengio, S. (2016). An online sequence-to-sequence model using partial conditioning. *Advances in neural information processing systems*, 29. [CrossRef]

[15] Chen, Z., Chen, L., Shen, W., & Xu, K. (2021). Remaining useful life prediction of lithium-ion battery via a sequence decomposition and deep learning integrated approach. *IEEE Transactions on Vehicular Technology*, 71(2), 1466-1479. [CrossRef]

[16] Niu, Z., Zhong, G., & Yu, H. (2021). A review on the attention mechanism of deep learning. *Neurocomputing*, 452, 48-62. [CrossRef]

[17] Jin, X. B., Wang, Z. Y., Gong, W. T., Kong, J. L., Bai, Y. T., Su, T. L., ... & Chakrabarti, P. (2023). Variational

bayesian network with information interpretability filtering for air quality forecasting. *Mathematics*, 11(4), 837. [CrossRef]

[18] Qiu, Z., Zhu, T., Jin, Y., Sun, L., & Du, B. (2023). A graph attention fusion network for event-driven traffic speed prediction. *Information Sciences*, 622, 405-423. [CrossRef]

[19] Kao, I. F., Zhou, Y., Chang, L. C., & Chang, F. J. (2020). Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. *Journal of Hydrology*, 583, 124631. [CrossRef]

[20] Du, S., Li, T., Yang, Y., & Horng, S. J. (2020). Multivariate time series forecasting via attention-based encoder–decoder framework. *Neurocomputing*, 388, 269-279. [CrossRef]

[21] Kong, J. L., Fan, X. M., Jin, X. B., Su, T. L., Bai, Y. T., Ma, H. J., & Zuo, M. (2023). BMAE-Net: A data-driven weather prediction network for smart agriculture. *Agronomy*, 13(3), 625. [CrossRef]

[22] Shi, B., Ou, Y., Wang, D., & Zhao, G. (2024). Self-Organizing Hierarchical Incremental Learning Framework and Universal Approximation Analysis Based on Stochastic Configuration Mechanism. *Information Sciences*, 121402. [CrossRef]

[23] Wang, D., & Li, M. (2017). Stochastic configuration networks: Fundamentals and algorithms. *IEEE transactions on cybernetics*, 47(10), 3466-3479. [CrossRef]

[24] Chen, X., Chen, Y., and He, Z.(2021). Urban Traffic Speed Dataset of Guangzhou, China. Zenodo. [CrossRef]

[25] Wang, Y., Liao, W., & Chang, Y. (2018). Gated recurrent unit network-based short-term photovoltaic forecasting. *Energies*, 11(8), 2163. [CrossRef]

Yuqi Lin is currently enrolled in the College of Engineering at the University of Illinois at Urbana-Champaign (UIUC). His main areas of research focus on deep learning networks and time series prediction, in which he has demonstrated significant academic and practical expertise. Lin's goal is to advance the use of artificial intelligence in addressing complex temporal data challenges. With a solid grasp of both theory and practical application, he contributes to research that connects innovative machine learning methods with real-world use cases.