

Article

A Hybrid Model for Spatiotemporal Air Quality Prediction Based on Interpretable Neural Networks and a Graph Neural Network

Huijuan Ding¹ and Giseop Noh^{2,*}

¹ Department of Computer Information Engineering, Cheongju University, Cheongju 28503, Republic of Korea; 2021020321@office.cju.ac.kr

² Division of Software Convergence, Cheongju University, Cheongju 28503, Republic of Korea

* Correspondence: kafa46@cju.ac.kr

Abstract: To effectively address air pollution and enhance air quality, governments must be able to predict the air quality index with high accuracy and reliability. However, air quality prediction is subject to ambiguity and instability because of the atmosphere's fluidity, making it challenging to identify the temporal and spatial correlations using a single model. Therefore, a new hybrid model is proposed based on an interpretable neural network and a graph neural network (INNGNN), which simulates the temporal and spatial dependence of air quality and achieves accurate multi-step air quality prediction. A time series is first interpreted using interpretable neural networks (INN) to extract the potentially important aspects that are easily overlooked in the data; second, a self-attention mechanism catches the local and global dependencies and associations in the time series. Lastly, a city map is created using a graph neural network (GNN) to determine the relationships between cities in order to extract the spatially dependent features. In the experimental evaluation, the results show that the INNGNN model performs better than comparable algorithms. Therefore, it is confirmed that the INNGNN model can effectively capture the temporal and spatial relationships and better predict air quality.

Keywords: air quality forecasting model; temporal feature interpretation; spatial dependency detection; INN-GNN model; urban air quality modeling



Citation: Ding, H.; Noh, G. A Hybrid Model for Spatiotemporal Air Quality Prediction Based on Interpretable Neural Networks and a Graph Neural Network. *Atmosphere* **2023**, *14*, 1807. <https://doi.org/10.3390/atmos14121807>

Academic Editor: Stephan Havemann

Received: 25 October 2023
Revised: 5 December 2023
Accepted: 7 December 2023
Published: 9 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Human health and ecosystems depend heavily on the air quality, and air pollution is becoming a major global issue [1]. The term “air quality” refers to the quantity and make-up of several pollutants found in the atmosphere, including carbon monoxide, sulfur dioxide, nitrogen dioxide, PM_{2.5}, PM₁₀, and others. It is calculated based on the level of pollutant concentration and the degree of health impact, and is used to measure the air quality index (AQI). In addition to causing respiratory illnesses, air pollution has a negative impact on human health [2]; moreover, it also damages the environment and leads to an imbalance in the ecosystem [3]. The higher the AQI value, the worse the air quality. The air quality index (AQI) is a crucial metric that can help people understand that day's air quality and take the appropriate precautions to safeguard their health. Additionally, pollutants can lower the AQI value and preserve the ecosystem's equilibrium.

Air quality forecasting is a technique for predicting future air quality. This technology uses professional monitoring equipment and technology to collect and analyze atmospheric parameters (such as temperature, humidity, wind speed, air pressure, precipitation, etc.), geographical location, the air quality index, and other data, to predict and warn about air quality in the future [4]. The accurate prediction of AQI is crucial for safeguarding public health and maintaining ecosystem balance. However, air quality prediction (AQP) remains a challenging task due to complex temporal and spatial dependencies:

1. Temporal dependence. Periodicity and trend are the primary ways in which dynamic changes in air quality over time are expressed. Periodicity is the occurrence of similar patterns or regular changes over a certain span of time. As shown in Figure 1a, the air quality index changes periodically over a week (① indicates a period of change). Trend is when a certain data pattern shows a continuous directional development over a certain period of time. Figure 1a shows a downward trend on certain days and for a certain period of time with cyclical changes. As shown in Figure 1b, the air quality index over a day changes with time (② indicates a shift in the trend over a certain amount of time); for instance, air pollutants from a prior or longer period of time have an impact on the current AQI.
2. Complex spatiotemporal correlations. In addition to changing dynamically over time, the spatial location also has an impact on air quality. As illustrated in Figure 2, city B's air quality will be impacted by the atmospheric conditions in city A. Even when taking time into account, the spatial relationship is still challenging.

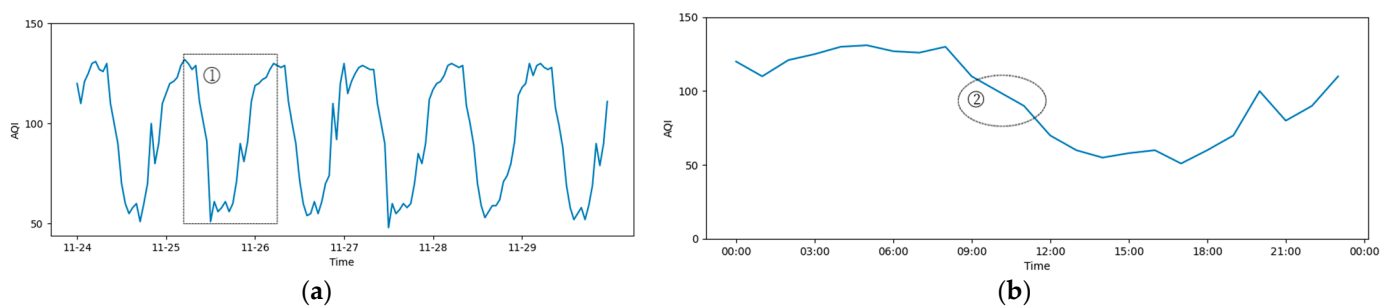


Figure 1. (a) Periodicity: air quality changes periodically over a week. (b) Trend: air quality has a tendency to change over a day. The horizontal axis represents time, and the vertical axis represents AQI value.

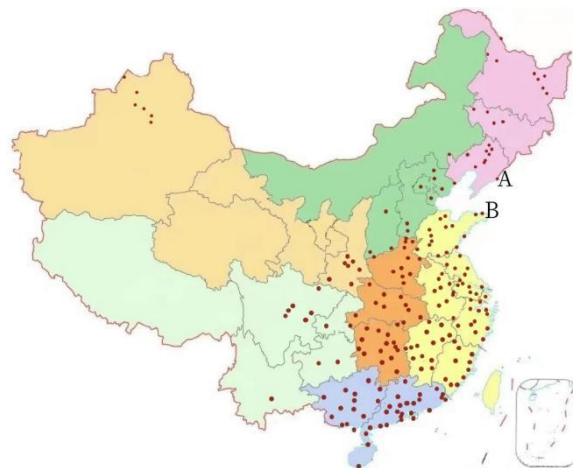


Figure 2. Geographical location of 209 cities.

There are many existing AQP methods, some of which consider the correlation with time, including the autoregressive integrated moving average model [5], the recurrent neural networks model [6], and the long short-term memory model (LSTM) [7]. It is challenging for us to estimate the level of air quality with accuracy, since the aforementioned approaches take into account the dynamic changes in air quality, but do not consider the spatial relationship between air quality and geographic location. Some researchers have used convolution neural networks (CNNs) [8], which are often used in Euclidean space. So, in order to make up for these shortcomings, researchers could use graph convolutional neural networks (GCNs) [9], which can model non-Euclidean spaces well, but are unable to accurately represent the dependency of air quality between cities. Some

researchers have used CNNs to analyze and model air quality in order to express its spatial characteristics. In order to improve the defects of GCNs, the group aware graph neural network (GAGNN) [10] constructs the dependencies between cities. Although these models can represent the spatial relationship of air quality between cities, they ignore the temporal relationship. The graph convolutional neural network–long short-term memory model (GCN-LSTM) [11] incorporates time and space relationships to fully consider time and space dependencies; nonetheless, it still has flaws in handling non-stationary data problems. The ability of researchers to identify patterns and trends across various time ranges is still lacking.

Time series forecasting challenges have been the focus of research when using interpretable neural network (INN) models. A number of strategies and techniques have been applied to enhance the model's interpretability. The proposed NA-BEMD [12] model interprets the model by analyzing the weights of each stage of the time series through an attention mechanism. The N-BEATS [13] model makes the model interpretable through the use of specific functions in order to solve the problem of time series point prediction. In order to improve the estimation accuracy of time series forecasting, the MLP-M [14] model makes the middle layer of the neural network model adjustable and, thus, improves the interpretability of the model. The above model improves the accuracy of time series prediction by enhancing the interpretability of the neural network model, and allowing for a better understanding of the process of feature learning and prediction in the model.

The key to AQP is to effectively capture the patterns and trends in various time frames. A hybrid prediction model for multistage, spatiotemporal air quality prediction is provided based on interpretable neural networks and graph neural networks (INNGNN) in order to address the issues raised by this problem, and to enhance prediction stability and accuracy. The contributions of this paper are summarized as follows:

1. An advanced AQP module is constructed, which introduces the extraction of interpretable trends and periodic time series features. In order to thoroughly extract the properties from the data, the interpretation module uses residual connections in conjunction with the trend and periodicity of the time series, to extract the features that are easily missed and are challenging to extract at random moments.
2. The INNGNN hybrid model is proposed to perform a spatiotemporal prediction of AQP from time and space dimensions. Graph neural networks (GNN) are used to extract the spatial dependency between different cities, and interpretable neural networks (INN) are used to capture the temporal dependence between the observations made on multiple time scales, and allows for self-attention to acquire the local and global dependence of time. The prediction accuracy of the INNGNN model is enhanced, as shown by its more accurate prediction outcomes in the evaluation we conducted on a Chinese urban air quality dataset.

2. Related Work

Deep learning models and non-deep learning models are the two broad groups into which existing AQP techniques can be separated. Deterministic models and statistical models are the two types of non-deep learning models. The most representative and latest deterministic models for predicting air quality include the community multi-scale air quality (CMAQ) model [15–17] and the comprehensive air-quality model with extension (CAMx) [18,19]. These models make it simple to see how pollutants that have an impact on air quality are produced, transformed, and transported, but they have significant limitations when it comes to the incomplete previous knowledge of complex chemicals and their diffusion processes. The statistical models highlight the non-linear correlations among all the factors that may affect air quality, thereby overcoming the shortcomings of the deterministic models. Classic statistical methods, such as the auto regression integrated moving average (ARIMA) [5,20–22] and geographically weighted regression (GWR) [23], have been used with statistical models for predicting small datasets and univariate time series of air quality. In addition, traditional machine learning methods along with statistical

models have also been applied to AQP, such as random forest (RF) [24], SVM [22,25–27], improved SVM (LSSVM) [28], LR [29,30], ANNs [31,32], and the improved neural network models BPNN [33], GRNN [32], RBFNN [31], and other models for processing. Based on restricted datasets, the typical machine learning approach can only capture limited non-linear temporal and spatial correlation aspects affecting air quality. It is more appropriate for shallow hidden features, as the results demonstrate. Very complex dynamic nonlinear spatiotemporal connection features are still not amenable for the efficient acquisition of huge datasets.

In recent years, deep learning has been applied extensively to AQP, and this has significantly increased prediction accuracy. Among them, a recurrent neural network (RNN) [34] has been used to capture temporal dependence. RNN's enhanced models—long short-term memory (LSTM) [35–37] and gated recurrent units (GRUs) [38,39]—provide an improved long-term dependency model for AQP. Some researchers have proposed improved LSTM-related models, including read-first LSTM (RLSTM) [40] and vanilla LSTM with multichannel input and multi-route output (IVLSTM-MCMR) [41], which have enhanced the function of the multidimensional feature extraction of air quality. An auto-encoder with Bi-LSTM neural networks (AE-Bi-LSTM) [42] has been used to analyze the correlation between air quality and climate variables. Some hybrid models have been used to improve the temporal dependence; for example, the use of LSTM-LSTM [43] improves the nonlinear capability of air quality data. For AQP, the hybrid model attention-LSTM [44] is used to automatically train and focus on significant time steps in the time series data. While the temporal correlation of AQP can be extracted efficiently using the deep learning RNN correlation model, the patterns and trends for various time scales in the time series remain ineffectively captured. In addition, a number of variables that impact air quality must be taken into account, and that information is intricate. Temporal and spatial linkages must be taken into account while modeling, yet these models still have issues with handling these relationships.

Some academics have suggested using different deep learning algorithms for time and space modeling in order to make up for the drawbacks of the RNN-related models. Strong feature extraction is a capability of convolutional neural networks (CNNs) [45–48], which may also be used to extract the spatial correlations between several sites and regions. When used in AQP, they produce better prediction outcomes than the RNN correlation models. The primary use of CNN is in Euclidean space. AQP's multiregional space is typically non-Euclidean in nature. Graph convolutional neural networks (GCNs) [9,11] are capable of performing undirected graph data creation. The model is a good representation of the non-Euclidean space of the network of air quality monitoring points. In order to better represent the city dependence of AQP, the group-aware graph neural network (GAGNN) [10] has been proposed, and its results are better than those of the GCNs. These models have demonstrated clear advantages for the extraction of spatial features. Several hybrid deep learning models have been developed to address the spatial–temporal dependence of AQP. These models, which include CNN-LSTM [47,49,50], GC-LSTM [11], and CNN-biLSTM [51], are able to extract spatiotemporal correlations from historical air quality data.

Combined with the hybrid model, which combines a residual neural network (ResNet), bidirectional long short-term memory (BiLSTM), and GCN, the adaptive monitoring network topology can adapt to the characteristics of pollutants and improve the prediction accuracy [52]. The GCN and CNN hybrid model, incorporating a spatiotemporal attention mechanism module, enhances the features of critical information collected from across the different dimensions [53]. Even though these models have performed well in AQP, they still struggle to handle non-stationary data issues. The fusion of multiple CNNs and backpropagation neural networks (BPNN) has good denoising capabilities [54]. The fusion of CNN with a spatiotemporal attention mechanism and residual learning enhances the temporal and spatial dimension feature extraction [55]. The fusion of LSTM and wavelet transform can handle unstable time series signals very well [56], but these models still cannot handle the correlation between multiple sites well. The fusion of a time convolutional network and

a graph convolutional network can deal with multi-site dependency and time dependency well, but it ignores the characteristics of repeated time series changes within a certain time interval [57]. The AAGCN model integrates the transformer and GCN, which can capture multivariable dynamic dependencies, but cannot fully capture the time dependencies of time series [58]. Therefore, more research is necessary to identify the patterns and trends across a range of time scales.

3. Methodology

3.1. Problem Definition

In this study, the goal of air quality forecasting is to predict the air quality for a certain period of time based on the AQI, weather data, and geographic location. We will first define some concepts to describe the AQP problem.

Definition 1: cities and city groups. We use a set $S = \{s_i\}_{i=1}^{N_s}$ to represent the collection of cities, where N_s represents the number of cities, and the location of a city is represented by its longitude and latitude, and the location matrix of the city is expressed as $L \in \mathbb{R}^{N_s \times 2}$. In addition, a set $U = \{u_i\}_{i=1}^{N_u}$ is used to represent the set of city groups, where N_u represents the number of city groups.

Definition 2: city group graph. We will consider city groups as cities that have strong dependencies, and construct the city group graph as a fully connected graph. We use an undirected graph $g(v, \epsilon)$ to represent the topological structure of a city group graph, as shown in Figure 3, where $v = \{v_1, v_2, \dots, v_n\}$ represents the set of all city group nodes, and for any node v_i , we use R_i to represent its attribute value, and the attribute values of all nodes in v can be expressed as the matrix R . $\epsilon = \{\epsilon_1, \epsilon_2, \dots, \epsilon_n\}$ represents the set of edges of all city groups. For any edge ϵ_i , we use M_i to represent its attribute value, and the attribute values of all the edges in ϵ can be expressed as the matrix M .

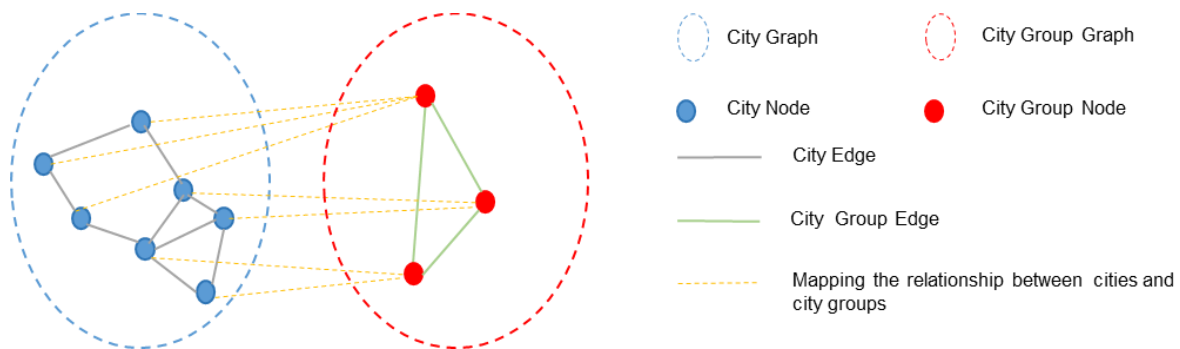


Figure 3. City graph and city group graph.

Definition 3: city graph. We use an undirected graph $G(V, E)$ to represent the topology of a city graph, as shown in Figure 3, where $V = \{v_1, v_2, \dots, v_n\}$ represents the set of all city nodes, and for any node V_i , we use X_i to represent its attribute value, and the attribute values of all nodes in V can be expressed as the matrix X . $E = \{e_1, e_2, \dots, e_n\}$ represents the set of all city edges. For any edge E_i , we use Y_i to represent its attribute value, and the attribute values of all the edges in E can be expressed as the matrix Y . The city graph is constructed as follows:

$$Y_{i,j} = \frac{1}{\sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}}, \tag{1}$$

$$d_{i,j} = \sqrt{(a_i - a_j)^2 + (b_i - b_j)^2}, 0 < d_{i,j} < r_n, \tag{2}$$

where $Y_{i,j}$ represents the edge attribute of the edge between city S_i and city S_j , $[a_i, b_i]$ represents the location of city S_i , and $[a_j, b_j]$ represents the location of city S_j . $d_{i,j}$ represents the Euclidean

distance between city S_i and city S_j , and r_n represents the distance threshold. When the distance between the two cities is very large, the cities are hardly affected by each other. The strategy we chose is that when the distance is less than r_n , city S_i and city S_j can be connected on the city map.

Definition 4: urban AQP. This paper mainly predicts the future AQI of a city by giving the historical observation sequence of the city’s location, time, AQI, and weather data. That is, for city S_i , given the city’s location L , the time vector T^t , and the historical observation sequence $H = (h_i^{t_1}, h_i^{t_2}, \dots, h_i^{t_m})$ of m time steps at time t , we can predict the AQI observation sequence $\hat{H} = (\hat{h}_i^{t_{m+1}}, \hat{h}_i^{t_{m+2}}, \dots, \hat{h}_i^{t_{m+n}})$ of the city’s n time steps.

3.2. Framework

Figure 4 displays the INNGNN framework structure. The temporal dependency module and the spatial dependency module make up the INNGNN. The temporal dependency module extracts the local and global time series features using self-attention and the interpretable trend, and the periodic time series features using INN. The geographical dependence of the geographic location is extracted with the spatial dependency module using GNN. First, the historical observation sequence feature matrix of the 209 cities is input, and the time series features of the different cities are extracted through multiple INNs. Then, the features extracted from the different cities are spliced together. Self-attention is added to the time series in order to capture the local and global dependencies of the various time steps. The input for self-attention is the feature matrix output obtained from splicing several city characteristics that were retrieved using multiple INNs. Natural language processing frequently uses encode–decode to manage mapping relationships and to map one sequence onto another more effectively [59,60]. The GNN module receives its input from the self-attention output. The GNN module uses the encode–decode structure and, to implement the message transmission mechanism, two layers of GNN are layered in the encode–decode, respectively. The decoder uses the output from the encoder to update the city and implement the mapping relationship between the individual cities and city groups. Ultimately, the fully connected layer outputs the prediction findings.

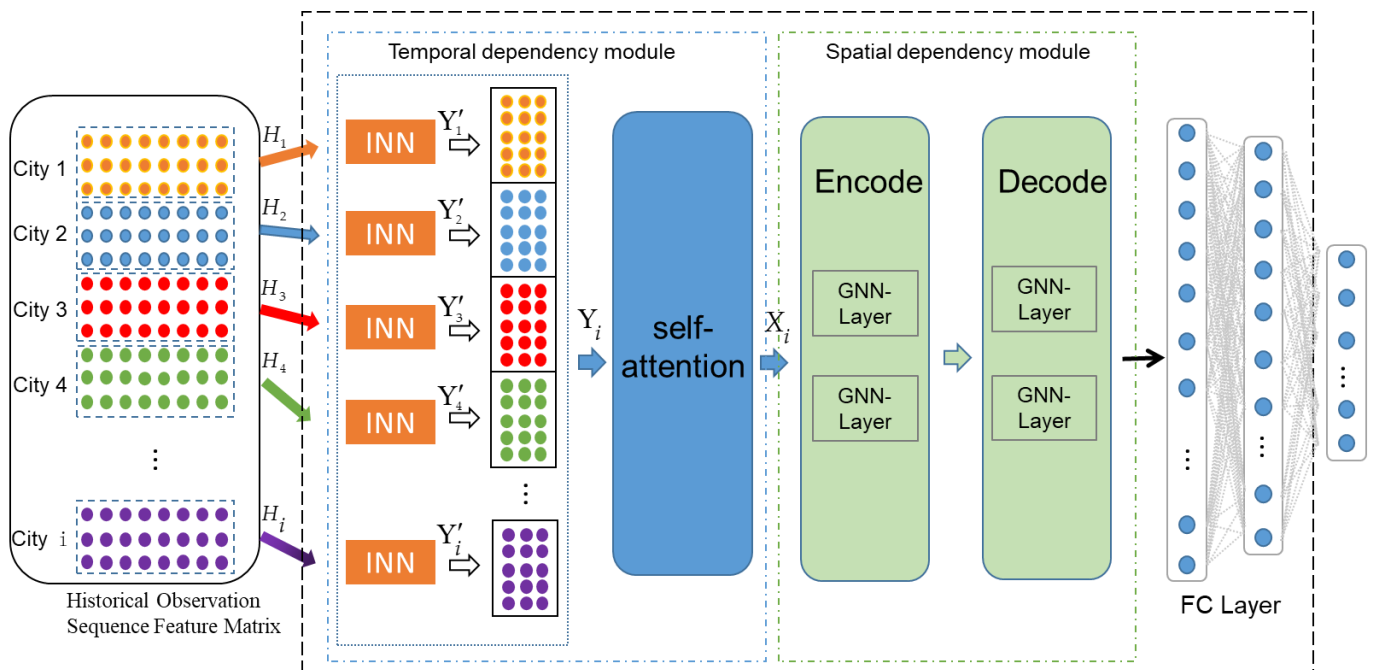


Figure 4. INNGNN framework.

3.3. Temporal Dependency Modeling

3.3.1. Interpretable Neural Networks

Given its periodic and trend properties, we suggest using interpretable neural networks (INN) to explain the periodicity and trend of the AQI time series. The two stacks comprising the interpretable neural networks, as depicted in Figure 5, are utilized to elucidate the trend and periodicity of the time series. Each stack consists of three blocks, and each block in a stack comprises two parts: the first portion comprises four layers of fully connected layers, and the second component is used to generate the expansion coefficients for the trend and periodic functions. A residual connection is introduced, meaning that each stack completes its own information aggregation procedure, and the input of the current block, less the output of the current block, is used as the input of the next block in order to capture the features of different levels in the input data. Input the historical observation sequence H_i of the i -th city, and the first part is described by Formula (3) as follows:

$$h_{l,1} = FC_{l,1}(x_l), h_{l,2} = FC_{l,2}(h_{l,1}), h_{l,3} = FC_{l,3}(h_{l,2}), h_{l,4} = FC_{l,4}(h_{l,3}), \tag{3}$$

where $h_{l,n}$ represents the n th layer of the l th block in each stack, x_l represents the residual of the previous block as the input of the current block, and FC represents the standard fully connected layer of the RELU nonlinear function, that is, $h_{l,1} = \text{RELU}(W_{l,1}H_l + b_{l,1})$. The second part is determined using Formula (4), and is described as follows:

$$\begin{aligned} \phi_l^a &= \text{LINEAR}_l^a(h_{l,4}), \phi_l^b = \text{LINEAR}_l^b(h_{l,4}) \\ \hat{y}_l &= f_l^b(\phi_l^b), \hat{x}_l = f_l^a(\phi_l^a) \end{aligned}, \tag{4}$$

where LINEAR is a simple linear projection layer, such as $\phi_l^b = W_l^b h_{l,4}$. \hat{y}_l and \hat{x}_l represent the output function of the current block of each stack and the residual calculation function of the next block, and ϕ_l^a and ϕ_l^b are the expansion coefficients of \hat{y}_l and \hat{x}_l , respectively. The periodicity and trend of the time series' interpretability are specifically described as follows.

Trend: The time series of air quality has a certain trend, as shown in Figure 1b. One of the trend characteristics is that it exhibits an upward or downward trend over time. Either a slowly changing function or a monotone function can simulate the trend. A slowly changing function is used here. The formula is described as follows:

$$\hat{y}_l^{\text{trend}} = T\phi_l^b, \hat{x}_l^{\text{trend}} = T\phi_l^a, \tag{5}$$

where $T = [1, t, \dots, t^p]$ is a power exponent matrix of t . When the p value is small, such as 3 or 4, it will force $\hat{y}_l^{\text{trend}}, \hat{x}_l^{\text{trend}}$ to simulate the trend. t is a discrete grid from 0 to $(N - 1)/N$, and t is expressed as $[0, 1, 2, \dots, N - 2, N - 1]^T / N$, N is the number of steps, the first stack is used as the trend model, the historical sequence H_i is used as the input, and the process formulae for the input and output are described as follows:

$$\begin{aligned} \hat{x}_1 &= H_i - \hat{x}_1^{\text{trend}}, \hat{x}_2 = \hat{x}_1 - \hat{x}_2^{\text{trend}}, \hat{x}_3 = \hat{x}_2 - \hat{x}_3^{\text{trend}} \\ y &= \sum_l \hat{y}_l^{\text{trend}} + \hat{x}_3 \end{aligned}, \tag{6}$$

where $\hat{x}_1, \hat{x}_2, \hat{x}_3$ represent the residuals of each block in the first stack and y represents the final output of the first stack, which is used as the input for the periodicity of the next stack.

Periodicity: As seen in Figure 1, there is a certain periodicity to the air quality time series (a). Periodic functions can be chosen to mimic the periodicity. Periodicity is defined as recurring or cyclical patterns within a specific time span. The Fourier series is chosen here, and the formula is described as follows:

$$\hat{y}_l^{\text{per}} = P\phi_l^b, \hat{x}_l^{\text{per}} = P\phi_l^a, \tag{7}$$

where $P = [1, \cos(2\pi t), \dots, \cos(2\pi \lfloor N/2 - 1 \rfloor t), \sin(2\pi t), \dots, \sin(2\pi \lfloor N/2 - 1 \rfloor t)]$ is the matrix of the sine and cosine waveforms, and the meanings of t and N are the same as those

for the trend. The second stack has periodicity characteristics, and the process formulae for the input and output are described as follows:

$$\begin{aligned} \tilde{x}_1 &= y - \hat{x}_1^{per}, \tilde{x}_2 = \tilde{x}_1 - \hat{x}_2^{per}, \tilde{x}_3 = \tilde{x}_2 - \hat{x}_3^{per} \\ Y'_i &= \sum_l \hat{y}_l^{per} + \tilde{x}_3 \end{aligned} \quad (8)$$

where $\tilde{x}_1, \tilde{x}_2, \tilde{x}_3$ represent the residuals of each block in the second stack, respectively, and Y'_i represents the final output of the second stack, which is the feature extraction of the i -th city. Then, the features of each city are merged and completed using the stacking approach. The formula is described as follows:

$$Y_i = [Y'_1; Y'_2; Y'_3; \dots; Y'_4], \quad (9)$$

where Y_i represents the feature merging of all the cities as the input for the next part of the model.

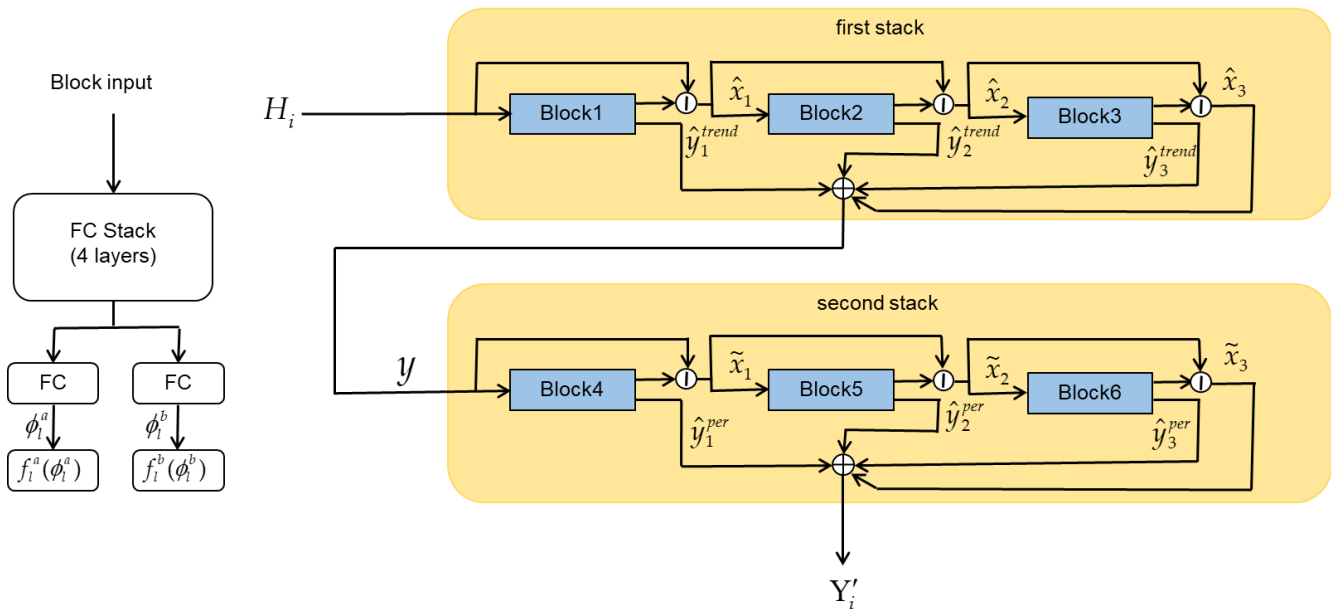


Figure 5. Architecture of interpretable neural networks (INN). \ominus : represents the input minus the output of each block; \oplus : represents the addition of the output information from each block in every stack with the residual information from the final block, for information aggregation.

3.3.2. Time Step Local and Global Dependency Capturing

A self-attention method is introduced in order to be able to capture the correlation between the distinct time steps of the time series, as well as the global dependent features and local dependent features. As shown in Figure 6, the self-attention mechanism is designed by referring to transformer [61] architecture, in which a multi-head attention can consider local and global dependencies at the same time. Each head can pay attention to the dependencies between the various time steps by computing the correlation weights between the items. The computation procedure is represented by the following formula:

$$Q = f(Y_i, W_Q), K = f(Y_i, W_K), V = f(Y_i, W_V), \quad (10)$$

$$Attention(Q, K, V) = soft \max \left(\frac{QK^T}{\sqrt{d_K}} \right) V, \quad (11)$$

where Q is the query matrix, K is the key matrix, and V is the value matrix. W_Q, W_K, W_V are all learnable parameters, K^T represents the transpose of K , and d_K represents the dimension of the K vector. Through matrix multiplication, the degree of association between each

element’s query vector and key vector in the matrix is realized, a weight calculation is performed, a normalization operation is realized through *softmax* and, then, a weighted summation is performed. Finally, all the dimensions are flattened using skip connection. The output is obtained through a fully connected (FC) layer.

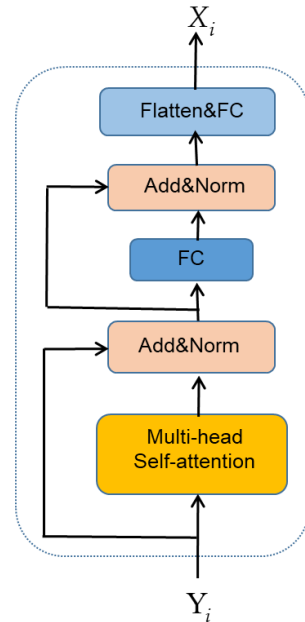


Figure 6. Self-attention network mechanism.

3.4. Spatial Dependency Modeling

The spatial relationship of the location data for each city can be shown as a graph. The spatial interdependence between cities must be taken into account in order to forecast the air quality of each city. A graph neural network (GNN) is used to construct the dependencies between cities. A detailed description of the composition process is as follows.

City grouping: The cities with strong dependencies are assigned to a city group, and each city is mapped onto a city group one by one; this grouping method allows us to identify any potential spatial dependencies between the cities. We use the $\Omega \in \mathbb{R}^{N_s \times N_u}$ matrix to represent the mapping relationship between the cities and city groups. Cities can be assigned to multiple city groups. In order to illustrate the correlation between cities and city groups, Ω is randomly initialized during training and optimized at the same time. For the case given, and shown in Figure 7, there are 10 cities divided into 3 city groups, among which the probability of city s_6 being assigned to city group u_1 is 0.1, the probability it of being assigned to city group u_2 is 0.8, and the probability it of being assigned to city group u_3 is 0.1. This shows that city s_6 has a stronger correlation with city group u_2 . In order to capture the spatial dependence between the cities, the geographic location L of the city is added, and the process definition for city grouping is as follows:

$$\hat{X}_i = FC(g_v(X_i, L_i)), \tag{12}$$

$$R_j = \sum_{i=1}^{N_s} \Omega_{j,i}^T \hat{X}_i, \tag{13}$$

where X_i is the output of the temporal dependent module, L_i represents the geographic coordinates of city s_i , \hat{X}_i represents city s_i and contains location information, $g_v(\cdot)$ represents the function implemented by using a FC layer, and $\Omega_{j,i}$ represents city s_i assigned to the city probabilities for groups u_j . We then have $\sum_{j=1}^{N_u} \Omega_{i,j} = 1$, where R_j represents the

city group representation of the cities assigned to the city group, and Ω^T represents the transpose of the matrix Ω .

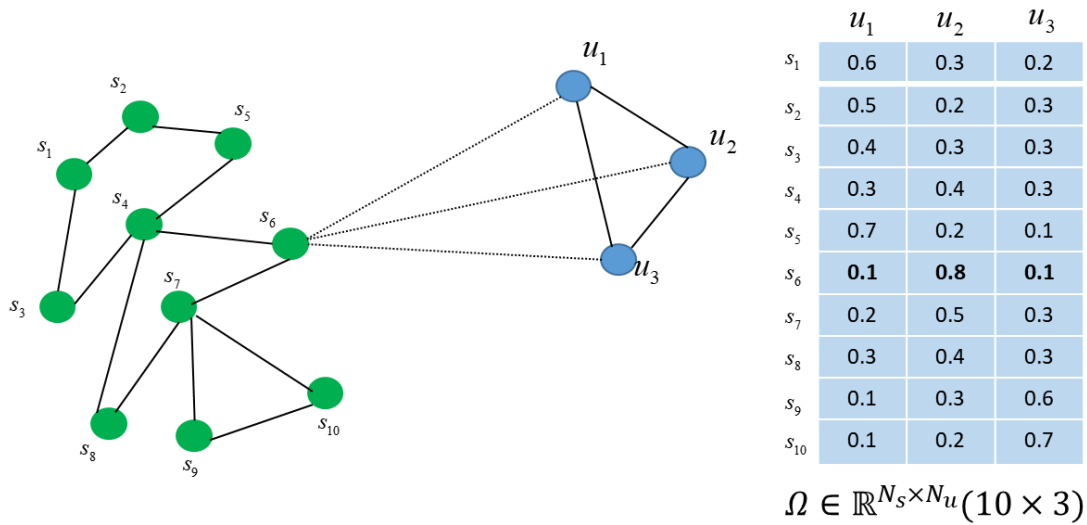


Figure 7. Relationship between cities and city groups.

Dependencies between city groups: In a city group, the nodes of each city group are fully connected to generate a fully connected undirected graph g , and the dependency relationship between the city groups is modeled through the mechanism of message passing. The modeling process is as follows:

$$M_{i,j} = f_g(R_i, R_j, T^t), \tag{14}$$

$$M_i = \{(R_i, R_j, M_{i,j})\}_{i \neq j} \tag{15}$$

$$\hat{M}_i \leftarrow \mu_g(M_i), \hat{R}_i \leftarrow \varphi_g(\hat{M}_i, R_i)'$$

where $M_{i,j}$ represents the city group u_i , the city group u_j contains the edge attributes of the time vector T^t , and the time vector T^t is composed of months, weeks, and days. R_i and R_j are the attribute values of the city groups u_i and u_j , respectively, M_i is all the information collected and is passed on to city group u_i through message passing, and is converted into vector \hat{M}_i after the message passing is completed. \hat{R}_i is the city group representation after the city group u_i is updated. f_g, μ_g , and φ_g are all implemented using a FC layer.

Dependencies between cities: During city representation, the cities are updated through the process of assigning cities to city groups, and the dependency relationship between the cities is completed through the transmission of messages in city graph G , and the message transmission mechanism is similar to that of the city groups. The difference is that the cities incorporate time series features from the temporal dependency module output, geographic location information, and city group information assigned to the city. The specific process is as follows:

$$\tilde{X}_i = \sum_{j=1}^{N_u} \Omega_{i,j} \hat{R}_j, \tag{16}$$

$$X'_i = \text{concat}(X_i, \tilde{X}_i), \tag{17}$$

$$Y_i = \left\{ (\tilde{X}_i, \tilde{X}_k, Y_{k,i}) \right\}_{k \in k(i)}, \tag{18}$$

$$\hat{Y}_i \leftarrow \mu(Y_i), X''_i \leftarrow \varphi(X'_i, \hat{Y}_i)$$

where \hat{R}_j represents the updated city group u_j ; \tilde{X}_i represents the city group containing city s_i ; X'_i represents city s_i and integrates information from the time series features and city

group information, which contains geographic location information; and Y_i represents all the information collected through message transmissions that were passed to city s_i . The neighbor information is aggregated, and is then converted into vector \hat{Y}_i, X_i'' , which is the updated city representation of city s_i . The connection functions $concat, \mu$, and φ are also implemented using a FC layer.

4. Experiments

4.1. Data Description

The INNGNN model was evaluated using the AQI dataset of some cities in China. With an hourly sampling frequency from 1 January 2017 to 30 April 2019, 209 cities were gathered as the dataset. The dataset includes geographic location, weather, and AQI data. The details of the data are as follows:

1. AQI dataset: The data come from the national urban air quality real-time release platform, downloaded from the public platform https://drive.google.com/file/d/1L_vpbLJhOJpNh-TpLdSWsaG3xCpzMVSQ/ (accessed on 15 June 2023).
2. Weather data: Weather data include the humidity, wind direction, rainfall, wind speed, air pressure, temperature, and visibility. The data come from the open platform for environmental big data <http://www.envicloud.cn/> (accessed on 15 June 2023).
3. Geographic location data: The geographic location of each city is shown in Figure 2. The geographic locations of all the cities are marked on the map with red dots.

The time step in this study is 1 h, the time step for a day is 24, and samples with a step of one were generated using a sliding window. We sorted all of the generated samples into training, test, and validation sets, and arranged them chronologically.

4.2. Experimental Settings

Experiments on the GPU were conducted, and all model constructions were completed within the open-source pytorch framework. The training epoch for all the models was set to 300, the batch size was set to 64, and the learning rate was set to 0.001. In addition, the power index p of matrix T was set to 4, the size of the city group was 15, the hidden unit of the GNN was 32, the distance threshold r_n was 250 km, and the GNN was set to 2 layers. We used the Adam [62] optimizer to optimize the training of the model.

In addition, AQP is a multiple linear regression problem. In order to effectively evaluate the model, the deviation between the predicted value and the observed value was calculated, using the mean absolute error (MAE) and the root mean square error (RMSE) as the indicators for the model's evaluation. The calculation process is as follows:

$$MAE = \frac{1}{N_S \times \tau} \sum_{i=1}^{N_S} \sum_{t=1}^{\tau} |O_i^t - \tilde{O}_i^t|, \quad (19)$$

$$RMSE = \sqrt{\frac{1}{N_S \times \tau} \sum_{i=1}^{N_S} \sum_{t=1}^{\tau} |O_i^t - \tilde{O}_i^t|^2}, \quad (20)$$

where τ represents the number of time samples, and O_i^j and \tilde{O}_i^j represent the observed value and predicted value for city i at time t , respectively. The same experimental parameters were used for five verifications for each of our tests, and the average result was used to determine the final outcome.

4.3. Experimental Results

4.3.1. Comparative Prediction Results

In this section, the INNGNN model is compared to the following baseline models:

DeeperGCN [63]: An extended form of GCN, DeeperGCN uses a deeper network structure that enhances its capacity to model complicated graph data. Eight layers were used in the experimental comparison.

LSTM [35]: LSTM (long short-term memory) is a variant of a recurrent neural network (RNN), which solves the problems of traditional RNNs regarding gradient disappearance and gradient explosion. The experiment's hidden unit was set to 32, the predicted value of the AQI was the output, and the features from the historical observation sequence were the input.

GC-LSTM [11]: The GC-LSTM model combines a graph convolutional neural network (GCN) with a long short-term memory network (LSTM). In the experimental settings, the optimal setting of LSTM hidden units was 64.

GAGNN [10]: GAGNN is a neural network model designed to analyze graph data with intra- and inter-group links. It is specifically designed for graph-structured data. The hidden layer unit was set to 32 in the experiment.

SHARE [64]: With numerous hierarchical recursive layers and graph attention layers, SHARE is a semi-supervised hierarchical recursive graph neural network model for graph-structured data. The shared semi-supervised portion of the experimental settings was removed, while the other parameters stayed the same.

ST-UNet [65]: ST-UNet is a neural network model for processing graph-structured time series data. To improve the time series modeling ability of the graph-structured time series data, the model presented an extended GRU. The hidden unit of the expanded GRU was 32 in the experiment.

XGBoost [66]: XGBoost is based on the decision tree model and introduces feature parallel computing. For parameter optimization of the experiments, network search methods were used.

HighAir [67]: A neural network model, HighAir uses graph structural information to optimize the graphs. It creates connections between the graph data by simulating the dynamic elements and creating the graphs from a hierarchical viewpoint.

The prediction task results for the baseline model and the INNGNN model, which were run on the dataset for one to six hours per day, are displayed in Table 1. The INNGNN model's assessment metrics, MAE and RMSE, showed the best prediction results across all the prediction ranges, demonstrating the model's efficacy in the task of spatiotemporal air quality prediction.

Table 1. Prediction results of the INNGNN model and baseline model.

Model	Metric	1 h	2 h	3 h	4 h	5 h	6 h
INNGNN	MAE	5.48	8.49	10.67	12.34	13.72	14.91
	RMSE	10.70	16.03	19.66	22.29	24.37	26.11
DeeperGCN	MAE	6.54	9.74	11.77	13.40	15.29	16.41
	RMSE	13.67	18.93	21.14	23.83	26.25	28.02
LSTM	MAE	6.50	10.26	13.18	15.52	17.40	18.91
	RMSE	13.85	19.26	23.52	26.83	29.46	31.55
GC-LSTM	MAE	5.95	9.16	11.58	13.46	15.00	16.31
	RMSE	11.91	16.98	20.82	23.69	25.97	27.82
GAGNN	MAE	5.56	8.59	10.80	12.52	13.91	15.10
	RMSE	10.81	16.17	19.84	22.51	24.63	26.37
SHARE	MAE	5.84	9.07	11.49	13.35	14.74	15.79
	RMSE	11.27	16.84	20.77	23.60	25.80	27.38
ST-UNet	MAE	5.95	9.30	11.58	13.38	14.82	16.02
	RMSE	11.74	18.01	21.34	23.90	25.94	27.64
XGBoost	MAE	6.85	10.89	13.99	16.27	18.14	19.56
	RMSE	14.25	19.80	24.72	28.14	30.63	33.44
HighAir	MAE	5.50	8.52	10.81	12.50	14.00	15.09
	RMSE	10.80	16.10	19.85	22.70	24.91	26.40

In terms of performance, the hybrid INNGNN model is still superior to the graph neural network. As shown in Table 1, for the 1 h and 6 h prediction tasks, the MAE error of the INNGNN model was reduced by 19% and 10%, respectively, and the RMSE error

was reduced by 27% and 7%, compared to the DeeperGCN model. In the 6 h prediction task, the RMSE error of the INNGNN model was reduced by 4% and 5.8%, compared to the SHARE model and the ST-UNet model, respectively. When compared to the enhanced graph neural network models, GAGNN and HighAir, the INNGNN model improved the MAE and RMSE errors for short-term, medium-term, and long-term predictions during the 1 h to 6 h prediction tasks. For the 1 h prediction task, the MAE error was reduced by 6.6% and 8.6%, respectively. Our model's MAE and RMSE errors are still significantly better than those of the HighAir model, which has a better prediction effect. From this thorough comparison, the INNGNN model improves upon the graph neural network foundation by incorporating the extraction of temporal data. This allows for a more accurate capture of the temporal and spatial connections, particularly for long-term predictions, demonstrating its superior benefits.

Regarding the time series feature extraction process, the INNGNN model adds trend and periodic feature extraction, as compared to the GC-LSTM hybrid model. This greatly enhances the feature capture of burst points. In order to better extract the spatiotemporal aspects of air quality prediction, the feature extraction for the correlation between cities is integrated into the spatial feature extraction process concurrently. As demonstrated in Table 1, the INNGNN model's prediction accuracy outperformed that of the GC-LSTM model over the 1–6 h prediction test.

4.3.2. Comparative Analysis: Individual Module vs. Hybrid Model

A few of the model's modules were investigated to confirm that the suggested hybrid INNGNN model is accurate in its forecasting performance. Figure 8 illustrates how we compared the RMSE and MAE values of the prediction outcomes of the GNN module, INNGNN model, and INN module from one hour to six hours, without including self-attention. The results displayed in the figure demonstrate the greater predictive performance of the hybrid model, with the INNGNN model outperforming each separate module.

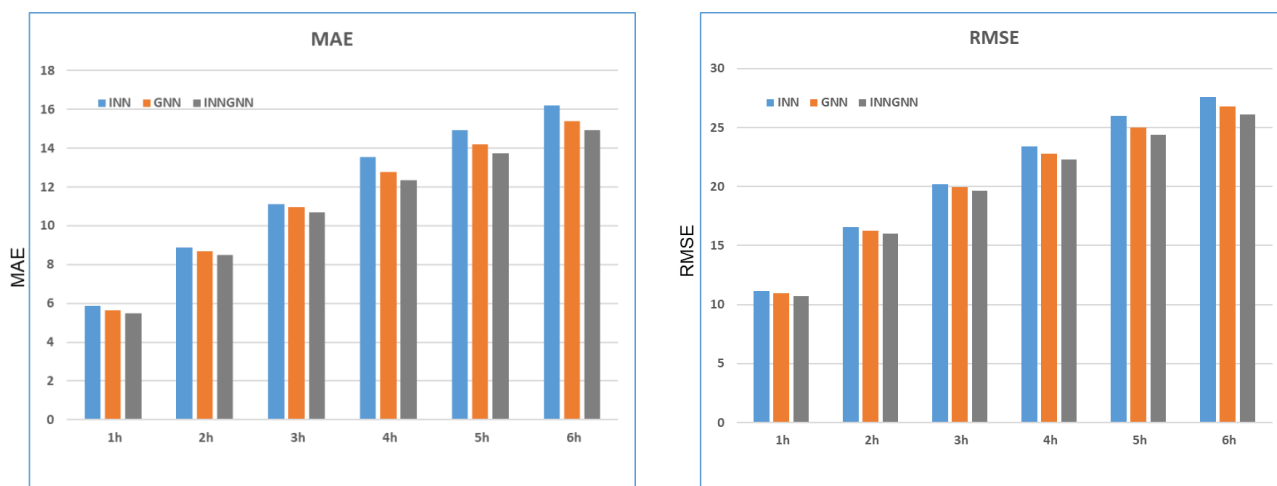


Figure 8. Comparison of MAE value and RMSE value of prediction results of the INN module, GNN module, and INNGNN model.

4.3.3. Ablation Studies

Ablation investigations were carried out to confirm the prediction performance of our proposed INNGNN model. The model for prediction used in the experiment was renamed INNGNN-INN after the INN part was removed, and INNGNN-GNN after the GNN part was removed for making predictions. This model compares the linear correlation between the INNGNN model and its corresponding predicted values. Among these three models, it can be observed from Figure 9 that the INNGNN model has the best fitting effect on the predicted value and the observed value, and the correlation coefficient R^2 is 0.91, which is higher than that of the INNGNN-INN model (the correlation coefficient

R^2 is 0.88), indicating that the temporal correlation results in a significant improvement of the prediction results. At the same time, it is higher than the INNGNN-GNN model (the correlation coefficient R^2 is 0.88), indicating that the spatial correlation significantly improved the prediction results.

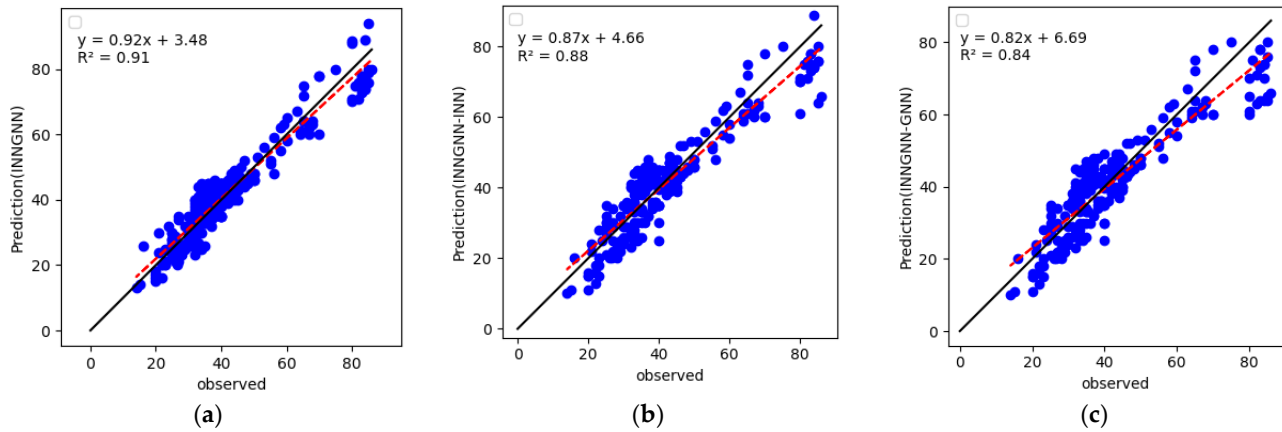


Figure 9. Correlations between observations and predictions on the test dataset for different components of the model: (a) INNGNN model; (b) INNGNN-INN model; and (c) INNGNN-GNN model. The red dashed line and black solid line are the regression lines, and $y = x$ is the reference line, respectively.

4.3.4. Display and Analysis

This section shows the AQP results for two cities from 1 April to 8 April using the INNGNN model. In Figure 10, the blue line represents the real value, and the red line represents the predicted value. Figure 10a represents the prediction effect for cities with low-scale fluctuations in air quality, and Figure 10b represents the prediction effect for cities with high-scale fluctuations in air quality. Our INNGNN model reacted promptly to time series fluctuations in both scenarios. The AQI increased significantly between 2 April and 3 April, as shown in Figure 10a, but the INNGNN model was still able to react promptly to the time series changes, demonstrating the model’s strong accuracy and efficient performance.

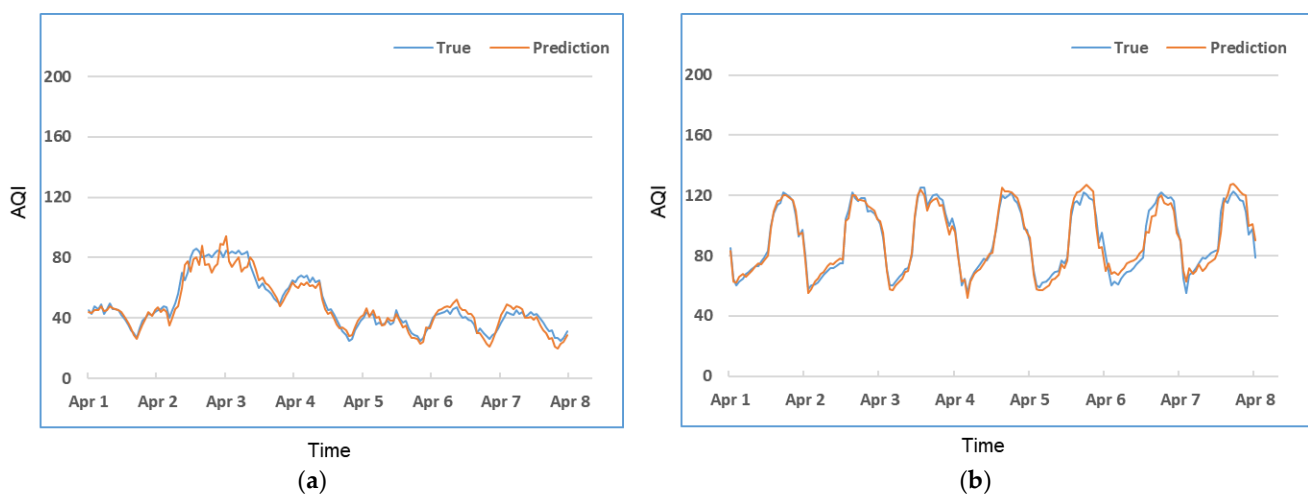


Figure 10. Final predictions for city 1 and city 2: (a) city 1 is a city with low-scale fluctuations, and (b) city 2 is a city with high-scale fluctuations. The AQI value was calculated by mapping the concentration values of different pollutants onto indices, calculating the sub-index of each pollutant, and taking the highest value as the AQI value [68].

5. Conclusions

This study proposes a hybrid model, INNGNN, to forecast air quality by combining GNN and self-attention. The INN module is used to fully extract the time series characteristics in order to obtain the temporal dependence, and to give the periodicity and trend the interpretability needed for forecasting. These issues with periodicity and trend are often ignored in time series forecasting. Furthermore, the self-attention mechanism is incorporated in order to capture the time series' local and global dependencies, as well as the varying relevance of each input feature at different times. In order to achieve the expected spatial dependencies, the combination of GNN models can determine the connection relationship between cities in order to capture the dependencies between them. When the INNGNN model was compared to previous models and assessed using real air quality datasets, it performed better over a range of prediction ranges. To further validate its performance, ablation experiments were conducted by comparing different modules of the model, and the INNGNN consistently exhibited superior performance.

In conclusion, the spatiotemporal aspects of urban air quality data are effectively captured by the INNGNN model, and this model can be expanded and used for other multivariate timeseries spatiotemporal applications. Environmental factors or specific dataset characteristics that were not considered in this study may influence the model's performance. Additionally, the possibility of further investigating and improving the comprehensibility of the model's forecasts exists. In subsequent studies, we want to investigate further how graph neural networks could be integrated with other time series models in order to increase the predictability and versatility of these models. We will also concentrate on enhancing the interpretability of the model and resolving any biases or limitations present in the dataset.

Author Contributions: Conceptualization, H.D. and G.N.; methodology, H.D. and G.N.; software, H.D. and G.N.; validation, H.D. and G.N.; formal analysis, H.D. and G.N.; investigation, H.D. and G.N.; resources, H.D. and G.N.; data curation, H.D. and G.N.; writing—original draft preparation, H.D. and G.N.; writing—review and editing, H.D. and G.N.; visualization, H.D. and G.N.; supervision, H.D. and G.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this study are openly available in https://drive.google.com/file/d/1I_vpbLJhOjPnh-TpLdSWsaG3xCpzMVSQ/. The flared volume data are available from 1 January 2017 to 30 April 2019 (data <http://www.envicloud.cn/> (accessed on 15 June 2023)).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lin, Y.-C.; Chi, W.-J.; Lin, Y.-Q. The Improvement of Spatial-Temporal Resolution of PM_{2.5} Estimation Based on Micro-Air Quality Sensors by Using Data Fusion Technique. *Environ. Int.* **2020**, *134*, 105305. [[CrossRef](#)] [[PubMed](#)]
2. Wijnands, J.S.; Nice, K.A.; Seneviratne, S.; Thompson, J.; Stevenson, M. The Impact of the COVID-19 Pandemic on Air Pollution: A Global Assessment Using Machine Learning Techniques. *Atmos. Pollut. Res.* **2022**, *13*, 101438. [[CrossRef](#)] [[PubMed](#)]
3. Brett, G.J.; Whitt, D.B.; Long, M.C.; Bryan, F.O.; Feloy, K.; Richards, K.J. Submesoscale Effects on Changes to Export Production Under Global Warming. *Glob. Biogeochem. Cycles* **2023**, *37*, e2022GB007619. [[CrossRef](#)]
4. Pruthi, D.; Liu, Y. Low-Cost Nature-Inspired Deep Learning System for PM_{2.5} Forecast over Delhi, India. *Environ. Int.* **2022**, *166*, 107373. [[CrossRef](#)] [[PubMed](#)]
5. Balachandran, S.; Chang, H.H.; Pachon, J.E.; Holmes, H.A.; Mulholland, J.A.; Russell, A.G. Bayesian-Based Ensemble Source Apportionment of PM_{2.5}. *Environ. Sci. Technol.* **2013**, *47*, 13511–13518. [[CrossRef](#)] [[PubMed](#)]
6. Singh, K.P.; Gupta, S.; Kumar, A.; Shukla, S.P. Linear and Nonlinear Modeling Approaches for Urban Air Quality Prediction. *Sci. Total Environ.* **2012**, *426*, 244–255. [[CrossRef](#)] [[PubMed](#)]
7. Navares, R.; Aznarte, J.L. Predicting Air Quality with Deep Learning LSTM: Towards Comprehensive Models. *Ecol. Inform.* **2020**, *55*, 101019. [[CrossRef](#)]

8. Zhang, C.; Yan, J.; Li, C.; Rui, X.; Liu, L.; Bie, R. On Estimating Air Pollution from Photos Using Convolutional Neural Network. In Proceedings of the 24th ACM International Conference on Multimedia; Association for Computing Machinery, New York, NY, USA, 15–19 October 2016; Springer: Berlin/Heidelberg, Germany, 2016; pp. 297–301.
9. Bruna, J.; Zaremba, W.; Szlam, A.; LeCun, Y. Spectral Networks and Locally Connected Networks on Graphs 2014. *arXiv* **2014**, arXiv:1312.6203.
10. Chen, L.; Xu, J.; Wu, B.; Qian, Y.; Du, Z.; Li, Y.; Zhang, Y. Group-Aware Graph Neural Network for Nationwide City Air Quality Forecasting 2021. *arXiv* **2021**, arXiv:2108.12238.
11. Qi, Y.; Li, Q.; Karimian, H.; Liu, D. A Hybrid Model for Spatiotemporal Forecasting of PM_{2.5} Based on Graph Convolutional Neural Network and Long Short-Term Memory. *Sci. Total Environ.* **2019**, *664*, 1–10. [[CrossRef](#)]
12. Baek, J.; Lee, C.; Yu, H.; Baek, S.; Lee, S.; Park, C. Automatic Sleep Scoring Using Intrinsic Mode Based on Interpretable Deep Neural Networks. *IEEE Access* **2022**, *10*, 36895–36906. [[CrossRef](#)]
13. Oreshkin, B.N.; Carpov, D.; Chapados, N.; Bengio, Y. N-BEATS: Neural Basis Expansion Analysis for Interpretable Time Series Forecasting 2020. *arXiv* **2020**, arXiv:1905.10437.
14. Huang, D.; Tang, S.; Zhou, D.; Hao, J. Nox Emission Estimation in Gas Turbines via Interpretable Neural Network Observer with Adjustable Intermediate Layer Considering Ambient and Boundary Conditions. *Measurement* **2022**, *189*, 110429. [[CrossRef](#)]
15. Byun, D.W.; Ching, J.K.S. *Science Algorithms OF THE Epa Models-3 Community Multiscale Air Quality (CMAQ) Modeling System*; Office of Research and Development, US Environmental Protection Agency: Washington, DC, USA, 1999.
16. Byun, D.; Schere, K.L. Review of the Governing Equations, Computational Algorithms, and Other Components of the Models-3 Community Multiscale Air Quality (CMAQ) Modeling System. *Appl. Mech. Rev.* **2006**, *59*, 51–77. [[CrossRef](#)]
17. Mueller, S.F.; Mallard, J.W. Contributions of Natural Emissions to Ozone and PM_{2.5} as Simulated by the Community Multiscale Air Quality (CMAQ) Model. Available online: <https://pubs.acs.org/doi/pdf/10.1021/es103645m> (accessed on 25 June 2023).
18. Environ. User's Guide to the Comprehensive Air Quality Model with Extensions (CAMx) 2014. Available online: <http://www.camx.com> (accessed on 11 July 2023).
19. Koo, Y.-S.; Choi, D.-R.; Kwon, H.-Y.; Jang, Y.-K.; Han, J.-S. Improvement of PM₁₀ Prediction in East Asia Using Inverse Modeling. *Atmos. Environ.* **2015**, *106*, 318–328. [[CrossRef](#)]
20. Ni, X.Y.; Huang, H.; Du, W.P. Relevance Analysis and Short-Term Prediction of PM_{2.5} Concentrations in Beijing Based on Multi-Source Data. *Atmos. Environ.* **2017**, *150*, 146–161. [[CrossRef](#)]
21. Zhang, L.; Lin, J.; Qiu, R.; Hu, X.; Zhang, H.; Chen, Q.; Tan, H.; Lin, D.; Wang, J. Trend Analysis and Forecast of PM_{2.5} in Fuzhou, China Using the ARIMA Model. *Ecol. Indic.* **2018**, *95*, 702–710. [[CrossRef](#)]
22. García Nieto, P.J.; Sánchez Lasheras, F.; García-Gonzalo, E.; de Cos Juez, F.J. PM₁₀ Concentration Forecasting in the Metropolitan Area of Oviedo (Northern Spain) Using Models Based on SVM, MLP, VARMA and ARIMA: A Case Study. *Sci. Total Environ.* **2018**, *621*, 753–761. [[CrossRef](#)]
23. Ma, Z.; Hu, X.; Huang, L.; Bi, J.; Liu, Y. Estimating Ground-Level PM_{2.5} in China Using Satellite Remote Sensing. *Environ. Sci. Technol.* **2014**, *48*, 7436–7444. [[CrossRef](#)]
24. Masmoudi, S.; Elghazel, H.; Taieb, D.; Yazar, O.; Kallel, A. A Machine-Learning Framework for Predicting Multiple Air Pollutants' Concentrations via Multi-Target Regression and Feature Selection. *Sci. Total Environ.* **2020**, *715*, 136991. [[CrossRef](#)]
25. Leong, W.C.; Kelani, R.O.; Ahmad, Z. Prediction of Air Pollution Index (API) Using Support Vector Machine (SVM). *J. Environ. Chem. Eng.* **2020**, *8*, 103208. [[CrossRef](#)]
26. Zhou, Y.; Chang, F.-J.; Chang, L.-C.; Kao, I.-F.; Wang, Y.-S.; Kang, C.-C. Multi-Output Support Vector Machine for Regional Multi-Step-Ahead PM_{2.5} Forecasting. *Sci. Total Environ.* **2019**, *651*, 230–240. [[CrossRef](#)] [[PubMed](#)]
27. He, H.; Li, M.; Wang, W.; Wang, Z.; Xue, Y. Prediction of PM_{2.5} Concentration Based on the Similarity in Air Quality Monitoring Network. *Build. Environ.* **2018**, *137*, 11–17. [[CrossRef](#)]
28. Li, Z.; Yang, J. PM-25 Forecasting Use Reconstruct Phase Space LS-SVM. In Proceedings of the 2010 The 2nd Conference on Environmental Science and Information Application Technology, Wuhan, China, 17–18 July 2010; Volume 1, pp. 143–146.
29. Beckerman, B.S.; Jerrett, M.; Martin, R.V.; van Donkelaar, A.; Ross, Z.; Burnett, R.T. Application of the Deletion/Substitution/Addition Algorithm to Selecting Land Use Regression Models for Interpolating Air Pollution Measurements in California. *Atmos. Environ.* **2013**, *77*, 172–177. [[CrossRef](#)]
30. He, Z.; Liu, P.; Zhao, X.; He, X.; Liu, J.; Mu, Y. Responses of Surface O₃ and PM_{2.5} Trends to Changes of Anthropogenic Emissions in Summer over Beijing during 2014–2019: A Study Based on Multiple Linear Regression and WRF-Chem. *Sci. Total Environ.* **2022**, *807*, 150792. [[CrossRef](#)] [[PubMed](#)]
31. Wahid, H.; Ha, Q.P.; Duc, H.N. Computational Intelligence Estimation of Natural Background Ozone Level and Its Distribution for Air Quality Modelling and Emission Control. In Proceedings of the 28th International Symposium on Automation and Robotics in Construction (ISARC 2011), Seoul, Republic of Korea, 29 June–2 July 2011.
32. Antanasijević, D.Z.; Pocajt, V.V.; Povrenović, D.S.; Ristić, M.Đ.; Perić-Grujić, A.A. PM₁₀ Emission Forecasting Using Artificial Neural Networks and Genetic Algorithm Input Variable Optimization. *Sci. Total Environ.* **2013**, *443*, 511–519. [[CrossRef](#)] [[PubMed](#)]
33. Kamal, M.M.; Jailani, R.; Shauri, R.L.A. Prediction of Ambient Air Quality Based on Neural Network Technique. In Proceedings of the 2006 4th Student Conference on Research and Development, Shah Alam, Malaysia, 27–28 June 2006; pp. 115–119.

34. Loy-Benitez, J.; Vilela, P.; Li, Q.; Yoo, C. Sequential Prediction of Quantitative Health Risk Assessment for the Fine Particulate Matter in an Underground Facility Using Deep Recurrent Neural Networks. *Ecotoxicol. Environ. Saf.* **2019**, *169*, 316–324. [[CrossRef](#)] [[PubMed](#)]
35. Lu, X.; Sha, Y.H.; Li, Z.; Huang, Y.; Chen, W.; Chen, D.; Shen, J.; Chen, Y.; Fung, J.C.H. Development and Application of a Hybrid Long-Short Term Memory—Three Dimensional Variational Technique for the Improvement of PM_{2.5} Forecasting. *Sci. Total Environ.* **2021**, *770*, 144221. [[CrossRef](#)]
36. Ulpiani, G.; Duhirwe, P.N.; Yun, G.Y.; Lipson, M.J. Meteorological Influence on Forecasting Urban Pollutants: Long-Term Predictability versus Extreme Events in a Spatially Heterogeneous Urban Ecosystem. *Sci. Total Environ.* **2022**, *814*, 152537. [[CrossRef](#)]
37. Wu, C.; He, H.; Song, R.; Peng, Z. Prediction of Air Pollutants on Roadside of the Elevated Roads with Combination of Pollutants Periodicity and Deep Learning Method. *Build. Environ.* **2022**, *207*, 108436. [[CrossRef](#)]
38. Huang, G.; Li, X.; Zhang, B.; Ren, J. PM_{2.5} Concentration Forecasting at Surface Monitoring Sites Using GRU Neural Network Based on Empirical Mode Decomposition. *Sci. Total Environ.* **2021**, *768*, 144516. [[CrossRef](#)]
39. Xu, R.; Deng, X.; Wan, H.; Cai, Y.; Pan, X. A Deep Learning Method to Repair Atmospheric Environmental Quality Data Based on Gaussian Diffusion. *J. Clean. Prod.* **2021**, *308*, 127446. [[CrossRef](#)]
40. Zhang, B.; Zou, G.; Qin, D.; Lu, Y.; Jin, Y.; Wang, H. A Novel Encoder-Decoder Model Based on Read-First LSTM for Air Pollutant Prediction. *Sci. Total Environ.* **2021**, *765*, 144507. [[CrossRef](#)] [[PubMed](#)]
41. Fang, W.; Zhu, R.; Lin, J.C.-W. An Air Quality Prediction Model Based on Improved Vanilla LSTM with Multichannel Input and Multiroute Output. *Expert Syst. Appl.* **2023**, *211*, 118422. [[CrossRef](#)]
42. Zhang, B.; Zhang, H.; Zhao, G.; Lian, J. Constructing a PM_{2.5} Concentration Prediction Model by Combining Auto-Encoder with Bi-LSTM Neural Networks. *Environ. Model. Softw.* **2019**, *124*, 104600. [[CrossRef](#)]
43. Eren, B.; Aksangür, İ.; Erden, C. Predicting next Hour Fine Particulate Matter (PM_{2.5}) in the Istanbul Metropolitan City Using Deep Learning Algorithms with Time Windowing Strategy. *Urban Clim.* **2023**, *48*, 101418. [[CrossRef](#)]
44. Zhu, J.; Deng, F.; Zhao, J.; Zheng, H. Attention-Based Parallel Networks (APNet) for PM_{2.5} Spatiotemporal Prediction—Science-Direct. *Sci. Total Environ.* **2021**, *769*, 145082. [[CrossRef](#)]
45. Rijal, N.; Gutta, R.T.; Cao, T.; Lin, J.; Bo, Q.; Zhang, J. Ensemble of Deep Neural Networks for Estimating Particulate Matter from Images. In Proceedings of the 2018 IEEE 3rd International Conference on Image, Vision and Computing (ICIVC), Chongqing, China, 27–29 June 2018; pp. 733–738.
46. Pak, U.; Ma, J.; Ryu, U.; Ryom, K.; Juhyok, U.; Pak, K.; Pak, C. Deep Learning-Based PM_{2.5} Prediction Considering the Spatiotemporal Correlations: A Case Study of Beijing, China. *Sci. Total Environ.* **2020**, *699*, 133561. [[CrossRef](#)]
47. Yan, R.; Liao, J.; Yang, J.; Sun, W.; Nong, M.; Li, F. Multi-Hour and Multi-Site Air Quality Index Forecasting in Beijing Using CNN, LSTM, CNN-LSTM, and Spatiotemporal Clustering. *Expert Syst. Appl.* **2021**, *169*, 114513. [[CrossRef](#)]
48. Zhang, B.; Geng, Z.; Zhang, H.; Pan, J. Densely Connected Convolutional Networks with Attention Long Short-Term Memory for Estimating PM_{2.5} Values from Images. *J. Clean. Prod.* **2022**, *333*, 130101. [[CrossRef](#)]
49. Wen, C.; Liu, S.; Yao, X.; Peng, L.; Li, X.; Hu, Y.; Chi, T. A Novel Spatiotemporal Convolutional Long Short-Term Neural Network for Air Pollution Prediction. *Sci. Total Environ.* **2019**, *654*, 1091–1099. [[CrossRef](#)] [[PubMed](#)]
50. Zhu, M.; Xie, J. Investigation of Nearby Monitoring Station for Hourly PM_{2.5} Forecasting Using Parallel Multi-Input 1D-CNN-biLSTM. *Expert Syst. Appl.* **2023**, *211*, 118707. [[CrossRef](#)]
51. Gilik, A.; Ogrenci, A.S.; Ozmen, A. Air Quality Prediction Using CNN+LSTM-Based Hybrid Deep Learning Architecture. *Environ. Sci. Pollut. Res.* **2022**, *29*, 11920–11938. [[CrossRef](#)] [[PubMed](#)]
52. Wu, C.; He, H.; Song, R.; Zhu, X.; Peng, Z.; Fu, Q.; Pan, J. A Hybrid Deep Learning Model for Regional O₃ and NO₂ Concentrations Prediction Based on Spatiotemporal Dependencies in Air Quality Monitoring Network. *Environ. Pollut.* **2023**, *320*, 121075. [[CrossRef](#)] [[PubMed](#)]
53. Liu, B.; Wang, M.; Guesgen, H.W. A Hybrid Model for Spatial–Temporal Prediction of PM_{2.5} Based on a Time Division Method. *Int. J. Environ. Sci. Technol.* **2023**, *20*, 12195–12206. [[CrossRef](#)]
54. Kow, P.-Y.; Chang, L.-C.; Lin, C.-Y.; Chou, C.C.-K.; Chang, F.-J. Deep Neural Networks for Spatiotemporal PM_{2.5} Forecasts Based on Atmospheric Chemical Transport Model Output and Monitoring Data. *Environ. Pollut.* **2022**, *306*, 119348. [[CrossRef](#)] [[PubMed](#)]
55. Zhang, K.; Yang, X.; Cao, H.; Thé, J.; Tan, Z.; Yu, H. Multi-Step Forecast of PM_{2.5} and PM₁₀ Concentrations Using Convolutional Neural Network Integrated with Spatial–Temporal Attention and Residual Learning. *Environ. Int.* **2023**, *171*, 107691. [[CrossRef](#)]
56. Zeng, Y.; Chen, J.; Jin, N.; Jin, X.; Du, Y. Air Quality Forecasting with Hybrid LSTM and Extended Stationary Wavelet Transform. *Build. Environ.* **2022**, *213*, 108822. [[CrossRef](#)]
57. Choudhury, A.; Middya, A.I.; Roy, S. Attention Enhanced Hybrid Model for Spatiotemporal Short-Term Forecasting of Particulate Matter Concentrations. *Sustain. Cities Soc.* **2022**, *86*, 104112. [[CrossRef](#)]
58. Wang, X.; Wang, Y.; Peng, J.; Zhang, Z.; Tang, X. A Hybrid Framework for Multivariate Long-Sequence Time Series Forecasting. *Appl. Intell.* **2023**, *53*, 13549–13568. [[CrossRef](#)]
59. Cui, Z.; Zhang, J.; Noh, G.; Park, H.J. MFDGCN: Multi-Stage Spatio-Temporal Fusion Diffusion Graph Convolutional Network for Traffic Prediction. *Appl. Sci.* **2022**, *12*, 2688. [[CrossRef](#)]
60. Song, C.; Lin, Y.; Guo, S.; Wan, H. Spatial-Temporal Synchronous Graph Convolutional Networks: A New Framework for Spatial-Temporal Network Data Forecasting. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 914–921. [[CrossRef](#)]

61. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need 2023. *arXiv* **2023**, arXiv:1706.03762.
62. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization 2017. *arXiv* **2017**, arXiv:1412.6980.
63. Li, G.; Xiong, C.; Thabet, A.; Ghanem, B. DeeperGCN: All You Need to Train Deeper GCNs. *arXiv* **2020**, arXiv:2006.07739. [[CrossRef](#)]
64. Zhang, W.; Liu, H.; Liu, Y.; Zhou, J.; Xiong, H. Semi-Supervised Hierarchical Recurrent Graph Neural Network for City-Wide Parking Availability Prediction. In Proceedings of the National Conference on Artificial Intelligence, New York, NY, USA, 7–12 February 2020.
65. Yu, B.; Yin, H.; Zhu, Z. ST-UNet: A Spatio-Temporal U-Network for Graph-Structured Time Series Modeling 2021. *arXiv* **2021**, arXiv:1903.05631.
66. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
67. Xu, J.; Chen, L.; Lv, M.; Zhan, C.; Chen, S.; Chang, J. HighAir: A Hierarchical Graph Neural Network-Based Air Quality Forecasting Method. *arXiv* **2021**, arXiv:2101.04264.
68. Hsu, A.; Reuben, A.; Shindell, D.; De Sherbinin, A.; Levy, M. Toward the next Generation of Air Quality Monitoring Indicators. *Atmos. Environ.* **2013**, *80*, 561–570. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.