IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 25 January 2025, accepted 18 March 2025. Date of publication 00 xxxx 0000, date of current version 00 xxxx 0000.

Digital Object Identifier 10.1109/ACCESS.2025.3553490

A Multi-Factor-Fusion Framework for Efficient Prediction of Pedestrian-Level Wind Environment Based on Deep Learning

ZHEN-ZHONG HU[1,2], YAN-TAO MIN[1], SHUO LENG[3], SUNWEI LI[1], AND JIA-RUI LIN[4]

[1]Institute for Ocean Engineering, Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China
[2]Institute for Ocean Engineering, Tsinghua University, Beijing 100084, China
[3]Guangzhou Metro Construction Management Company Ltd., Guangzhou, Guangdong 510330, China
[4]Department of Civil Engineering, Tsinghua University, Beijing 100084, China

Corresponding author: Zhen-Zhong Hu (huzhenzhong@tsinghua.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2022YFC3801100, in part by the Natural Science Foundation of China under Grant 52378306, and in part by Guangdong Basic and Applied Basic Research Foundation under Grant 2022B1515130006.

ABSTRACT Efficient and accurate assessment of the Pedestrian-Level Wind Environment is essential to maintain a healthy and safe urban living environment. Numerical simulations, such as computational fluid dynamics and multi-scale modeling techniques, are commonly used for wind environment analysis. However, they are computationally intensive and time-consuming, particularly when dealing with the complexities of urban landscapes. This study proposes a novel Multi-Factor-Fusion (MFF) framework that leverages deep learning techniques. This framework integrates Graph Convolutional Networks and Long Short-Term Memory networks to extract and fuse multiple factors and create an end-to-end neural network model capable of directly predicting wind fields. By avoiding the need for grid division and iterative calculations, the framework significantly enhances the efficiency of wind environment analysis. Furthermore, multi-scale simulation data is used to train the model and correct the predictive results, ensuring the accuracy of the final results. This innovative approach has the potential to revolutionize the Pedestrian-Level Wind Environment prediction by achieving a trade-off between efficiency and accuracy.

INDEX TERMS Wind environment, deep learning, numerical simulation, green building.

I. INTRODUCTION

Pedestrian-level wind environment (PLWE) has a significant impact on people's daily lives in terms of human comfort, building ventilation, and pollutant dispersion [1]. With the rapid progress of urbanization in the past few decades, the natural wind environment has been directly affected by the construction of urban buildings [2]. On the one hand, high-rise buildings can induce intense downwash flows and corner streams, leading pedestrians to be exposed to uncomfortable or even dangerous environments [3]. On the other hand, densely packed buildings will block the near-ground air circulation and reduce wind speed, which obstructs pollutant dispersion and exacerbates the heat island effect [4]. As a result, accurate assessment of the PLWE around buildings has become an important issue in modern cities such as Hong Kong [5].

As an effective way to investigate wind flows, the numerical simulation technique has attracted the attention of numerous researchers and has made rapid progress in recent years [6]. In particular, computational fluid dynamics (CFD) is the most popular and successful numerical method to simulate the wind environment [7]. CFD applies numerical analysis algorithms to solve the governing Navier–Stokes equations, to reproduce the distribution of airflows in the computer. The technique is widely applied in the field of PLWE analysis and has proven to be able to provide reliable and satisfactory results [8]. For example, Du et al. [9] applied CFD simulation with high-quality meshes to investigate the PLWE on Hong Kong Polytechnic University campus. The

The associate editor coordinating the review of this manuscript and approving it for publication was Ghufran Ahmed.

simulation results achieved a good agreement with the wind tunnel experimental data. Zhang et al. [10] used CFD to simulate the PLWE around a 400m super-tall building. The results showed a significant increase in the maximum wind speed after the construction of the building. Mittal et al. [11] investigated the effect of corner-modified buildings on the PLWE using different turbulence models. It is observed that the rounding of building corners significantly reduced the wake length and increased wind speeds. Zhong et al. [12] conducted a detailed review of recent CFD modeling studies on PLWE and concluded that the main advantage of CFD simulations is that they can provide whole-flow field data at acceptable costs.

In addition to the common CFD-based simulation methods, multi-scale modeling techniques have emerged and received much attention in recent years [13]. The technique couples the mesoscale meteorological model with the microscale CFD model, which enables the dynamic analysis of PLWE under specific meteorological conditions and urban morphologies [14]. Huang et al. [15] combined the mesoscale Weather Research and Forecasting (WRF) model with CFD to simulate the PLWE in complex urban environments. A three-stage simulation approach was proposed in this study, including 1) WRF was utilized to generate mesoscale meteorology data, 2) mesoscale data was downscaled to provide an inflow wind profile for CFD, and 3) CFD simulation was carried out to obtain the PLWE. Zhao et al. [16] proposed a Numerical Weather Prediction (NWP)-CFD model to simulate the urban wind environments under typhoon weather conditions. With the involvement of the mesoscale NWP model, the prediction errors of wind profiles were greatly reduced. Mortezazadeh et al. [17] integrated WRF with the microscale Large Eddy Simulation (LES) model to simulate the urban microclimate under heatwaves. The study revealed that the multi-scale numerical model can accurately predict not only wind speeds but also temperatures. The multi-scale modeling technique is gaining more and more applications when the wind environment needs to be simulated in real-time or over complex terrain.

However, despite the popularity of numerical simulation in investigating the PLWE, these methods are still facing challenges. One of the main challenges is their limited computational efficiency [18]. Typical numerical models such as microscale CFD and mesoscale WRF use space discretization methods to chop continuous computational domains into discrete grids [19]. To achieve acceptable analysis resolution, pedestrian-level wind environment simulations often need to deal with millions to billions of grids [20], resulting in a large amount of processing time to traverse each grid. On the other hand, heavy iterative calculations are involved in the process of solving the governing equations to decouple the flow field variables [21]. As a result, an enormous amount of computation is required in the current numerical simulation process. A refined simulation of PLWE usually takes hours to dozens of hours [22], which is too long compared with the workflow cycle of building design and management activities. The low efficiency of numerical simulation has become one of the key challenges hindering its applications.

The artificial intelligence (AI) technique has made rapid progress in recent years and offers a fascinating solution to this problem [23]. Shao et al. [24] designed a physics-informed graph neural network model for rapid prediction of the urban wind field. The model is proven to run 1-2 orders of magnitude faster than the CFD model which provides training and test datasets for it. He et al. [25] proposed a hybrid framework composed of CFD simulation, machine learning model, and image processing algorithm to predict the low wind velocity areas around buildings. The authors demonstrated that the framework provided a highly efficient quantitative analysis method of the building wind environment. Mortezazadeh et al. [26] developed a random forest model to predict the wind power potential in urban areas. The model was trained based on the simulation data from the CFD and showed good agreement with the numerical simulation results on the test dataset. Reja et al. [27] reviewed recent studies on AI-based wind resource evaluation methods and commented that the utilization of AI models made wind resource utilization easier and reduced calculation time. However, despite the popularity of the AI technique, current AI models are mainly designed to complete the work of CFD simulation on a single scale. For the field of multi-scale wind environment analysis, the application of AI technique is still not found in the literature. As a result, current multi-scale wind environment analysis still relies on the coupling of numerical models, which is time-consuming and has greatly hindered the application of this method.

The above-mentioned problem motivates this study to explore an original approach for accurate and efficient prediction of PLWE. In detail, a Multi-Factor-Fusion (MFF) framework based on deep learning is proposed in this paper which incorporates Graph Convolutional Networks (GCN) and an attention-based Long Short-Term Memory (ALSTM) network. The deep learning model applies the end-to-end neural network to predict the wind field directly, which avoids the disadvantages of grid division and iterative calculation in numerical simulations and greatly improves the efficiency of wind environment analysis. Meanwhile, the deep learning model not only approximates the training data generated by multi-scale simulation but also corrects the intermediate results on that basis, whose reliability has been validated in the author's previous work [28]. As a result, the proposed approach can be scaled up to rapidly predict the PLWE with acceptable accuracy.

The remainder of this paper is organized as follows. Section II gives an overview of the proposed deep learning framework, and Section III explains its detailed design. Section IV discusses the data processing and training process. Section V presents a case study to validate the feasibility and accuracy of the framework. Finally, discussions and conclusions are given in Section VI.

II. OVERVIEW OF THE MFF FRAMEWORK

The prediction of the pedestrian-level wind environment is a complex task that requires consideration of a variety of influencing factors. For numerical simulation techniques such as CFD, boundary conditions and the distribution of obstacles are the main factors affecting the analysis results [29]. The boundary conditions are further configured based on local meteorological conditions. Besides, patterns of local wind environments can be extracted from the distribution of historical wind environments to assist in the prediction of the future.

As illustrated in Figure 1, an original deep learning framework for pedestrian-level wind environment prediction is constructed in this paper. Three features are selected to participate in the prediction of the wind environment, that is, the framework accepts three categories of input data:

1) Distribution of historical wind environment (historical pedestrian-level wind speed data),

2) Spatial distribution of obstacles (geometric and connectivity characteristics),

3) Local meteorological conditions (meteorology data at prediction time).

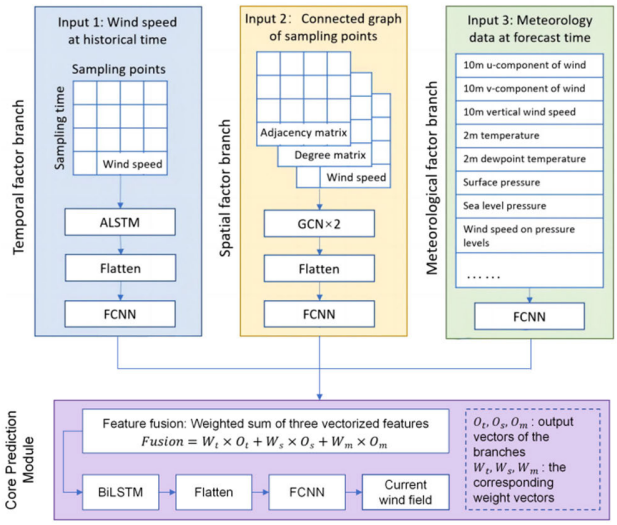

FIGURE 1. The MFF deep learning framework.

Correspondingly, three branches are designed in the framework to process the input data with a series of deep learning techniques. The temporal feature is time series data composed of historical wind speeds at different moments. The proposed approach utilizes the Long Short-Term Memory (LSTM) network to process the feature. And the attention mechanism is further introduced to improve the performance of LSTM. The spatial feature is organized in the form of graphs in this study, and the Graph Convolutional Network (GCN) is selected to deal with these graph data. The meteorological feature is represented by a vector of meteorological parameters provided by weather forecasts. The fully connected (FC) layer can handle the processing of this feature. The feature vectors extracted by the three branches are transferred to the core prediction module to jointly participate in the prediction process. Finally, pedestrian-level wind speed data at prediction time are output.

The core idea of the MFF framework is to transform the problem of dynamic wind environment forecasting into the task of wind speed time series prediction on sampling points. Therefore, before the framework is put into application, multiple sampling points need to be set in the target prediction area. The input and output wind speed data will be both stored on sampling points, and the spatial distribution of obstacles will also be represented by the connected graph of sampling points. After the prediction is completed, the overall wind speed distribution in the target area can be estimated through the predicted wind speed data on the sampling points.

The key model parameters of the proposed framework are listed in Table 1, and a detailed introduction to the framework structure will be given in the following subsections.

TABLE 1. Key model parameters in the MFF framework.

Parameter name	Parameter meaning
Sample_Times	The number of historical moments in the input historical wind environment data
Sample_Points	The number of wind environment sampling points
Time_Lag	The time interval between the output moment and the nearest input moment
LSTM_Dim	Output data size of the ALSTM network
Meteorology_Params	The number of meteorological parameters
BiLSTM_Dim	Output data size of the BiLSTM network

In addition, it is necessary to consider the issue of error accumulation arising as the prediction result propagates forward in long-term prediction tasks. Therefore, this study uses multi-scale simulation results at certain times for calibration to prevent divergence of results. The AI model is also trained on the multi-scale simulation.

III. DETAILED DESIGN OF THE MFF FRAMEWORK

A. TEMPORAL FACTOR BRANCH

The temporal factor branch is designed to extract the distribution patterns of the historical wind environment in the target area to assist the prediction process. This branch receives normalized historical pedestrian-level wind speed data as input and outputs a feature vector to the core prediction module. As shown in Figure 2, the input data of the branch is a two-dimensional matrix of size $Sample\_Times * Sample\_Points$. The matrix represents the wind speed data at sampling points during a period of historical moments. To be detailed, each column of the matrix is a wind speed vector that consists of the wind speed data at each sampling point at a certain moment. The length of the wind speed vector is the number of sampling points, i.e. *Sample_Points*. A total of *Sample_Times* wind speed vectors are input to

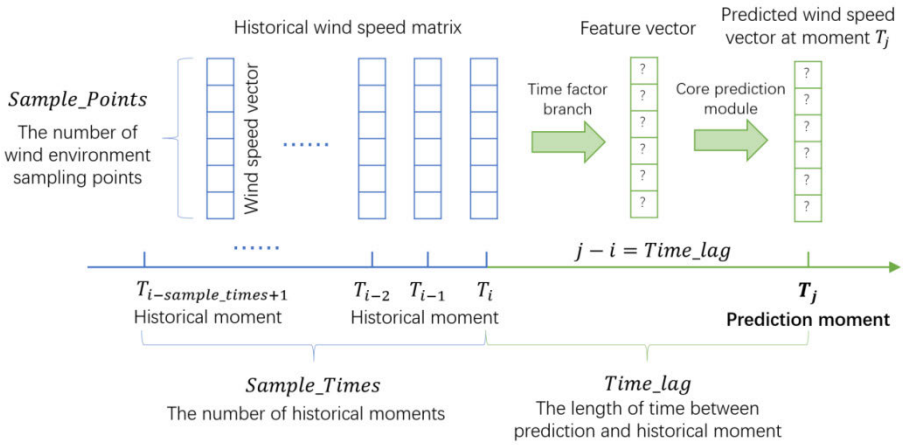

FIGURE 2. Model parameters of the temporal factor branch.

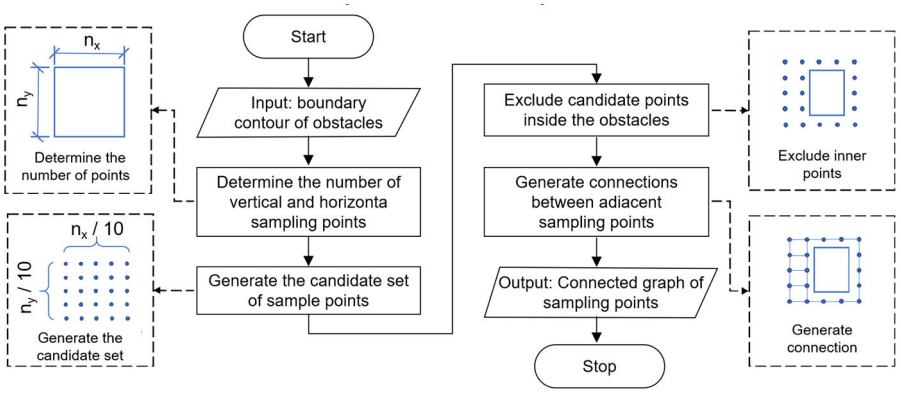

FIGURE 3. Algorithm for generating a connected graph based on obstacle contours.

the branch, representing that equivalent historical moments are considered. The framework is designed to predict the pedestrian-level wind speed distribution at time T_j in the future. Assuming that the most recent historical moment is T_i, the parameter *Time_Lag* is used to describe the number of intermediate moments between the two moments, i.e. $Time\_Lag = j - i$. It is noted that the interval time between two adjacent moments is 10 minutes in this study.

LSTM is a type of Recurrent Neural Network (RNN) that is designed to process long sequence data and capture the association relationship between contexts [30]. Compared with general RNN neurons, the LSTM unit introduces a special structure called gate to control the transmission and forgetting of features, which enables LSTM to deal with long-distance dependencies in data sequences. In the temporal factor branch, the historical wind speed matrix composed of wind speed vectors at historical moments is first input into an ALSTM [31] network layer. The network is designed to extract the temporal correlation of historical data based on the ability of the LSTM network to process time series data.

At each historical moment, wind speed data is fed into an LSTM unit. The length of its output vector can be set manually, which is denoted as *LSTM _Dim*. As a result, the network outputs a matrix of size $Sample\_Times * LSTM\_Dim$. A flatten layer is added subsequent to the LSTM network to convert the matrix into a vector. The vector is then input into an FC layer, which is used to map the data vector to a vector of a specified length for the convenience of data fusion with other branches. The length of the FC Layer output vector is set to *Sample_Points*. Finally, this vector represents the result of the feature extraction process of the temporal factor branch and participates in the subsequent network calculation process.

B. SPATIAL FACTOR BRANCH

The spatial factor branch is designed to consider the impact of the spatial distribution of obstacles, mainly buildings, on the pedestrian-level wind environment. In the proposed approach, the distribution of obstacles will be presented in the form of a connected graph of sampling points. Figure 3 presents an original algorithm for generating graphs based on obstacle contours.

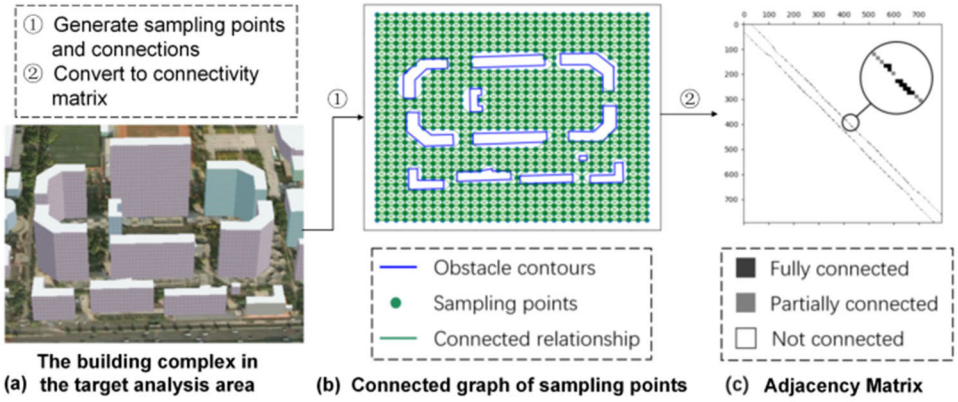

FIGURE 4. Example of generating a connected graph.

The algorithm mainly consists of 4 steps, including:

1) Determine the number of vertical and horizontal sampling points based on the size of the target area. After testing, a point spacing of around 10m is appropriate considering the calculation efficiency and sampling precision.

2) Generate the candidate set of sample points based on the determined number of points.

3) Exclude points inside the obstacles according to their contours from the candidate set to generate sampling points. The classic ray casting algorithm [32] is applied to judge whether a point is inside an obstacle. Thereafter the size of the sample point set is referred to as *Sample_Points*.

4) Generate connected relationships. In this study, the connection relationship of sampling points represents the degree of correlation between the wind speeds at the two points. For vertical/horizontal adjacent sampling points, they are considered to be fully connected, and the weight of connection in the adjacency matrix is set to 1. For obliquely adjacent sampling points, they are considered to be partially connected, and the weight of the adjacency matrix is set to 0.5. For other combinations of sampling points, the weight is set to 0.

Figure 4 illustrates an example of generating a graph based on obstacle contours. Figure 4(a) shows the distribution of obstacles in the target analysis area, and Figure 4(b) shows the generated sampling points and the graph. It is evident that the algorithm adapts well to the actual scenario and the adjacency matrix of the graph can be calculated, whose size is $Sample\_Points * Sample\_Points$ as shown in Figure 4(c). The adjacency and degree matrices, together with the historical wind speed matrix, are input to the spatial factor branch. GCN is a neural network model that applies convolution operation to graphs [33], extracting features by inspecting neighboring nodes in a graph. In this study, the convolutional part of the GCN network is applied to extract the spatial features of the pedestrian-level wind environment. In the convolution layer, the wind speed matrix is regarded as the graph signals to be convolved, while the adjacency and degree matrices are used to construct convolution kernels. The size of the output data is set to be the same as the input graph signals, i.e. $Sample\_Times * Sample\_Points$.

The operation after graph convolution is similar to the time factor branch. The signal matrix is converted into a one-dimensional vector by a flatten layer, and subsequently input into a FC Layer layer. The output vector length of FC Layer is set to be *Sample_Points*, so as to perform the feature fusion operation in the core prediction module.

C. METEOROLOGICAL FACTOR BRANCH

The meteorological factor branch is used to consider the influence of regional meteorology conditions at the prediction moment, approximating the setting of boundary conditions in multi-scale simulations. The meteorology data used in this study are collected from a global-scale meteorology research and reanalysis database — ECMWF Reanalysis v5 (ERA5) [34]. The ERA5 dataset provides gridded hourly meteorological data, which is commonly used to initiate the mesoscale meteorology simulations.

The selection and processing of data from ERA5 involves consideration of two aspects:

1) Due to the low resolution (of the ERA5 data, it is difficult to obtain detailed meteorology information at each sampling point. Consequently, this study uses meteorological data at the center of the target area to characterize the meteorological characteristics of the entire region. Specifically, the ERA5 data at the four grid points nearest to the central point are employed for trilinear interpolation in three dimensions, i.e. longitude, latitude, and time, to determine the meteorological parameters at the central point of the target area at a designated time.
2) This branch is designed to approximate the role of large-scale meteorological data in the multi-scale simulation, which is used to configure the boundary conditions of the mesoscale weather model Weather Research and Forecasting Model (WRF) and drive the model. Consequently, this study chooses 15 parameters

TABLE 2. Meteorological parameters input to the model.

Category		Parameter	Unit
Land surface data		10m u-component of wind	m/s
		10m v-component of wind	m/s
		2m temperature	K
		2m dewpoint temperature	K
		Skin temperature	K
		Sea-level pressure	hPa
		Surface pressure	hPa
		Relative humidity	%
Pressure level data	975 hPa	Temperature	K
		U-component of wind	m/s
		V-component of wind	m/s
		Relative humidity	%
	1000 hPa	Temperature	K
		U-component of wind	m/s
		V-component of wind	m/s

from ERA5 as the branch input (Table 2), which is consistent with the configuration of boundary conditions in WRF. These parameters are generally considered to have a significant impact on the pedestrian-level wind environment. The length of the branch input is denoted as *Meteorology_Params*. Since a total of 15 meteorological parameters are selected, the value of *Meteorology_Params* is set to 15.

The meteorology data are organized as a one-dimensional vector to be input to the model. The meteorological factor branch directly connects the input data to a FC layer to extract the meteorological feature. The output vector length of FC Layer is set to *Sample_Points*, which is equal to the other two branches.

D. CORE PREDICTION MODULE

The core prediction module is designed to fuse the feature vectors extracted by the three branches and export the final results. The method of feature fusion is represented as follows:

$$\boldsymbol{Fusion} = \boldsymbol{W_t} \times \boldsymbol{O_t} + \boldsymbol{W_s} \times \boldsymbol{O_s} + \boldsymbol{W_m} \times \boldsymbol{O_m} \tag{1}$$

In the equation, $\boldsymbol{O_t}$, $\boldsymbol{O_s}$, and $\boldsymbol{O_m}$ stand for the output vectors of temporal, spatial, and meteorological factor branches respectively, and $\boldsymbol{W_t}$, $\boldsymbol{W_s}$, and $\boldsymbol{W_m}$ are the corresponding weight vectors. The symbol $\times$ means to multiply vectors element by element. The principle of feature fusion is essentially the same as that of FC Layer, where $\boldsymbol{W_t}$, $\boldsymbol{W_s}$, and $\boldsymbol{W_m}$ are similar to the network weights in FC Layer, which are randomly initialized at the beginning of training and adjusted with the error backpropagation. The only difference is that the fusion vector is output directly without being processed by an activation function. The length of the output vector remains to be *Sample_Points*.

The fused feature is then input to a Bi-directional Long Short-Term Memory (BiLSTM) network to further capture the spatial-temporal dependencies of the pedestrian-level wind environment distribution. The bidirectional network is selected because wind speeds at adjacent sampling points have a two-way dependence relationship. The BiLSTM network can well capture the associations in both directions of the data series. The length of the output vector of the BiLSTM layer is determined manually, which is denoted as the parameter *BiLSTM _Dim*.

The BiLSTM network is subsequently connected with a FC layer, which is designed to decode the feature vectors into final wind environment prediction results. The output vector length of the FC layer is set to *Sample_Points*, where each element in the vector corresponds to a sampling point. After denormalization, the value of the element is applied as the predicted wind speed at the sampling point. As a result, the vector represents the final prediction results of the proposed framework. In the post-processing process, the overall distribution of the pedestrian-level wind environment can be derived based on the wind speeds at the sampling points.

E. INITIALIZATION AND CALIBRATION

The aforementioned deep learning framework can predict the wind field at a certain moment. To predict the wind field over a period of time containing N moments, it is necessary to prepare input data separately for these N predictions. Generally, outputs from previous moments can be used as input for these predictions, except for the first prediction at the initial moment. Besides, relying solely on the framework's predictive results from previous moments as input for subsequent moments can lead to the issue of error accumulation, especially for long-term prediction tasks. To address these two issues, the multi-scale wind environment simulation technique is used to calculate the wind environment at specific moments, providing the input "initial values" and "calibration" for the deep learning framework.

As shown in Figure 5, for the long-term prediction tasks, the multi-scale wind environment numerical simulation is performed every 2 hours. The wind field distribution at the moments between these 2-hour intervals is then predicted by the deep learning framework. Meanwhile, the results of numerical simulations are used to correct predictions made by the deep learning framework, as described by the following formula:

$$\boldsymbol{y_c} = \delta \boldsymbol{y_o} + (1 - \delta)\, \boldsymbol{y_r} \tag{2}$$

$$\boldsymbol{y_r} = (120 - t)/120 \times \boldsymbol{Y_0} + t/120 \times \boldsymbol{Y_{120}} \tag{3}$$

where t is the current time (in minutes), $\boldsymbol{Y_0}$, $\boldsymbol{Y_{120}}$ are the results of numerical simulation at 0 and 120 minutes respectively, and $\boldsymbol{y_r}$ is the reference wind speed derived through linear interpolation with time. $\boldsymbol{y_o}$ represents the predicted results of the deep learning model, and $\boldsymbol{y_c}$ refers to the corrected wind speed. δ is the coefficient weighing $\boldsymbol{y_o}$ and $\boldsymbol{y_c}$, which is set to 0.1.

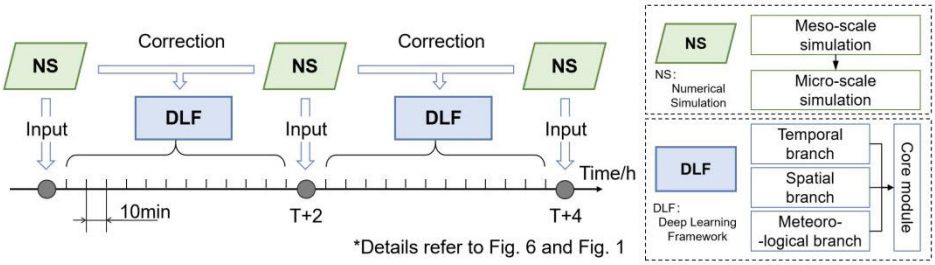

FIGURE 5. Prediction tasks executed with two-hour intervals.

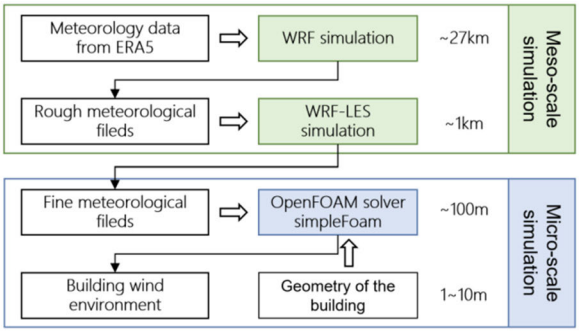

FIGURE 6. Multi-scale numerical simulation process.

IV. TRAINING OF THE MFF FRAMEWORK

A. DATA PREPARATION

The proposed framework involves three categories of data including pedestrian-level wind speeds, obstacle distribution, and meteorological conditions. These data need to be pre-processed for model training. This section will introduce the preparation methods of these data.

1) PEDESTRIAN-LEVEL WIND SPEED DATA

The pedestrian-level wind speed data used in the training process come from a multi-scale numerical simulation framework which is constructed in the author's previous work [28]. The framework is designed to perform dynamic pedestrian-level wind environment simulations under actual meteorological conditions. As shown in Figure 6, the numerical simulation process involves two models of different spatial scales. The mesoscale WRF simulation is first executed for meteorology analysis, while the microscale CFD simulation is subsequently launched for wind environment prediction. The mesoscale WRF model will transfer its simulation results to the microscale CFD model, which enables pedestrian-level wind environment simulation to run under realistic meteorological conditions.

WRF is a state-of-the-art mesoscale meteorology prediction model that is developed for both atmospheric research and weather forecasting tasks. In the proposed approach, it is used to perform mesoscale meteorological simulations to obtain local meteorology conditions which will be later provided to CFD simulations. In detail, two rounds of WRF simulations are carried out successively in this process. The first round of simulation is initialized with macroscale meteorological data from the ERA5 dataset whose data are organized in a 0.25-degree latitude-longitude grid. WRF assimilates the ERA5 data into its analysis model and outputs a mesoscale meteorological field around the target analysis area with a resolution of 1km. The second round of simulation is then initialized with the output of the first round. In this round, the Large Eddy Simulation (LES) mode in WRF is activated, which enables WRF to run with a finer resolution at the expense of a higher computational cost. Finally, the simulation process exports meteorological data with a resolution of around 100m, which is fine enough to be introduced into the microscale pedestrian-level wind environment simulation.

At the microscale, the proposed approach builds CFD models with the OpenFOAM software to perform pedestrian-level wind environment analysis. OpenFOAM is a widely applied CFD software that provides a customized programming environment for developing CFD applications. In this study, a steady-state CFD model is constructed with the OpenFOAM solver SimpleFoam. The OpenFOAM utility GmshToFoam is utilized to build the computational domain based on the building geometry data in Gmsh format. The mesoscale meteorological data are extracted from WRF at regular intervals to configure boundary and initial conditions, ensuring that CFD simulations can be performed under realistic meteorological conditions. The simulation finally outputs wind environment distribution results with a resolution of 1 to 10m, depending on the fitness of the grid. Wind environment data of the finest part of the computational domain, which is at the pedestrian level, is exported as the final result of the numerical simulation.

The proposed approach will run this framework on the same target analysis area for two different periods to generate training and test datasets separately. Despite the inefficiency of numerical simulations, the feasibility and reliability of the framework have been validated [27]. Therefore, the quality of model training and test data in this study is guaranteed.

2) OBSTACLE DISTRIBUTION DATA

The obstacles in this study mainly refer to buildings and structures in the target analysis area. The distribution of these objects is collected from the Baidu Map [35]. Baidu Map is an

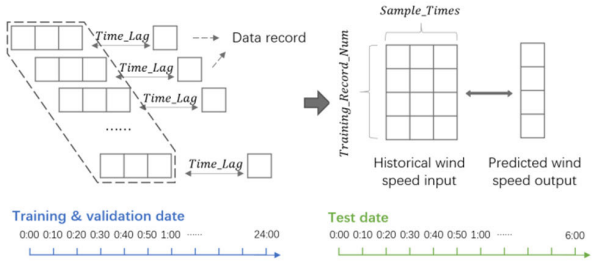

FIGURE 7. Processing of wind environment data in the dataset construction process.

online electronic map that provides geographic information and navigation services for more than 400 cities (mainly in China). In particular, it integrates detailed information on the distribution of buildings and structures in most of the cities. These data are available to the public through the application programming interfaces (APIs) [36]. A crawler program is constructed in this study to automatically generate network requests to the APIs and download obstacle distribution data in the target area. The raw collected data are organized in the form of vector formats which represent a building with its contour polygons and height attribute. These data will be first processed by the connected graph generating algorithm introduced in the previous section, and then participate in the training process.

3) METEOROLOGY DATA

The meteorology data are obtained from the ERA5 database provided by the European Centre for Medium-Range Weather Forecasts (ECMWF) [34]. ERA5 is a collection of global-scale meteorology datasets that provides historical and near-future meteorology analysis data. Specifically, two datasets named ''ERA5-land hourly data'' and ''ERA5 hourly data on single levels'' are involved in this study. The two datasets are used to provide meteorological data on the land surface and pressure levels respectively. The acquired parameters mainly include wind speed, pressure, and temperature, which are detailed in Table 2 in the previous section. ECMWF also exposes the API of the ERA5 database to the public. Following the prescribed API specification, meteorology data in the specified time and space range can be efficiently obtained.

B. TRAINING DATASET GENERATION

Based on the original data, the training, validation, and test datasets are further constructed. In the proposed framework, the pedestrian-level wind environment data are both involved in the model input and output. Therefore, the wind environment data needs to be processed carefully to avoid dataset contamination. The processing of wind environment data in the dataset construction process is shown in Figure 7. Each rectangle in the figure represents a vector of length *Sample_Points*, containing the wind speed data at each sample point at a certain moment.

In the proposed approach, the multi-scale numerical simulation is run on two different dates to generate wind environment data for training and testing separately. In this way, the testing process can avoid the interference of training data and evaluate the model performance independently. In the training data, a training dataset and a validation dataset are further generated randomly at a ratio of 8:2. The validation dataset is used to evaluate the model performance under different hyperparameters to determine the optimal configuration for the model.

The wind environment data for training, validation, and testing are organized in data records. Each data record consists of two parts including the input data and the expected output data. The input data contains wind speeds at consecutive *Sample_Times* moments and is considered known in the current record. The output data contains wind speed data at one moment which is treated as unknown. The time interval between the output moment and the nearest input moment is *Time_Lag*. Assuming the number of generated records is *Record_Num*, and the number of simulated moments is *Moment_Num*, there is a relationship among these parameters:

$$Record\_Num = Moment\_Num - Sample\_Times - Time\_Lag \quad (4)$$

The input parts of each data record are spliced together to form an overall input tensor for training/testing. Since the wind environment data at each moment is a vector of length *Sample_Points*, the overall wind environment input data is a three-dimensional tensor of size $Record\_Num * Sample\_Times * Sample\_Points$. Similarly, the expected output data is a two-dimensional matrix of size $Record\_Num * Sample\_Points$.

For obstacle data and meteorological data, the generation of training and test datasets is much easier. Since the distribution of obstacles will not change over time, obstacle data can participate in the training and test process as constant variables. The obtained ERA5 meteorological data are labeled with time. These data will be matched with wind environment data in the temporal dimension to generate the training and test dataset. The overall meteorology input data is a matrix of size $Record\_Num * Meteorology\_Params$.

C. MODEL TRAINING AND PARAMETER TUNING

During the training process, the Adam algorithm [37] is applied as the model weight optimization method. Adam is a stochastic gradient descent algorithm with adaptive learning rates and is known for its smooth weight optimization process and high convergence efficiency. The Mean Squared Error (MSE) is selected as the loss function, and the Root MSE (RMSE) and Mean Absolute Error (MAE) are applied as indicators to test the model performance. The calculation of these indicators is shown in the following equation:

$$MSE = 1/n \times \sum_{i=1}^{n} \left(y_i - \hat{y}_i\right)^2, RMSE = \sqrt{MSE} \quad (5)$$

$$MAE = 1/n \times \left|y_i - \hat{y}_i\right| \quad (6)$$

where n is the number of data records contained in the current batch, y_i is the expected wind speed data for the *i-th* record, and $\hat{y}_i$ is the data predicted by the framework. The optimization target is to minimize the MSE loss on the training dataset.

As illustrated in Table 1, multiple network hyperparameters can be adjusted in the proposed framework. Among them, the parameter *Time_Lag* can be determined manually, which represents how far into the future the wind environment will be predicted. In this study, *Time_Lag* is set to three values of 1, 3, and 6 to establish three different prediction models, which are used to predict the wind environment in the next 10 minutes, 30 minutes, and 1 hour. For each model, other hyperparameters, i.e. *Sample_Times*, *LSTM_Dim*, and *BiLSTM_Dim* need to be tuned to optimize the performance of the model. The grid search method is applied to enumerate all value combinations of these parameters. The combination of parameters that performs best on the validation set will be used to train the final version of the framework.

V. IMPLEMENTATION CASE STUDY

As illustrated in Figure 8, a building group in Beijing, China (116.35°E, 39.99°N) is selected to verify the feasibility of the proposed framework. The total construction area of the building group is over 36,000 m^2, and the height of the highest building is 85 m. At present, the consideration of wind in the operation and management mainly relies on regional weather forecasts and human experience, rather than efficient methods for quantitative analysis. The proposed method will be applied to the building group, providing an original way to efficiently predict the wind environment.

A. MODEL ESTABLISHMENT

The proposed framework is developed based on the TensorFlow [38] and Keras [39] toolkit. TensorFlow is an open-source library that implements a variety of deep learning models including the LSTM network and the FC layer. Keras is a Python-based toolset that provides a high-level programming interface for TensorFlow. Besides, the NumPy library [40]is also used in this study for performing matrix operations. The versions of TensorFlow, Keras, and NumPy are 1.13.1, 2.2.4, and 1.16.1, respectively.

The multi-scale numerical simulation of the studied area is first carried out to generate the training, validation, and test dataset. The simulation time range for generating the training and validation data is 0:00-24:00 on January 6, 2021, and the time range for the test data is 4:00-10:00 on February 19, 2022. The training and validation data are divided randomly with a ratio of 8:2. The time interval of each numerical simulation is 10 minutes. Therefore the training, validation, and test dataset has a total of 115, 29, and 36 moments of data, respectively. As the algorithm in Figure 3, 792 sample points are created in the target area, i.e., the parameter *Sample_Points* is set to 792 in the case study. At each moment, the wind environment obtained by numerical simulation is sampled at the sampling points to generate the input and expected output data.

TABLE 3. Parameter tuning range and results.

Hyperparameter	Tuning range	Tuning results		
		Model 1 *Time_Lag* = 1	Model 2 *Time_Lag* = 3	Model 3 *Time_Lag* = 6
Sample_Times	{1,2,3,4,5,6}	3	3	6
LSTM_Dim	{64,128,256,512}	128	128	128
BiLSTM_Dim	{64,128,256,512}	256	256	256
Other parameters	*Sample_Points* = 792, *Meteorology_Params* = 15			

Based on the training and validation data, the hyperparameters are tuned to determine the optimal configuration of the framework. The grid search method is applied to carry out the process, and the hyper-parameters to be adjusted and the corresponding search ranges are illustrated in Table 3. Since this study establishes 3 models for predicting the PLWE at different times, their parameters need to be tuned separately. Taking the model 1 (*Time_Lag* = 1, or to predict the wind environment in the next 10 minutes) as an example, the parameter combination that leads to the best performance of the model is *Sample_Times* = 3, *LSTM_Dim* = 128, *BiLSTM_Dim* = 256. The optimal parameters of the other two models are shown in Table 3.

Figure 9 shows the structure of model 3 output by Keras. The number of training epochs and the initial learning rate are set to 50 and 0.001 respectively. It is proved that the model weights can reach convergence under such training settings.

B. MODEL EVALUATION AND APPLICATION

The trained models are applied to the test dataset to evaluate their performance. For models designed to predict wind environments at different times, their performance on the test dataset is illustrated in Table 4. It is noted that the predicted wind speed of the framework has been denormalized before calculating the error. The relative MAE (rMAE) in the table refers to the average percentage ratio of the prediction error relative to the actual wind speed.

From the table, it is concluded that as the prediction time interval increases, the prediction accuracy of the framework decreases. This fact is reasonable since the farther the predicted time is from the present, the more uncertainties will lead to the change of wind environment, and the greater the probability that the predicted results deviate from the expected one. However, even for the one-hour prediction task, the error between the prediction and the actual wind environment is only 10.23%. It proves that the framework can predict the future pedestrian-level wind environment with high accuracy, within at least 1 hour.

* The left figure is shot on the field, and the right is the digital model of the target analysis area

FIGURE 8. Dormitory building group for case study.

TABLE 4. Performance of the MFF framework on the test dataset.

Model	RMSE (m/s)	MAE (m/s)	rMAE (%)
Model 1 (prediction in 10 minutes)	0.047	0.038	3.52
Model 2 (prediction in 30 minutes)	0.092	0.079	7.14
Model 3 (prediction in 1 hour)	0.135	0.116	10.23

Taking the No. 500 and No. 788 sampling points as examples, Figure 10 shows the variation of the predicted and actual wind speed. The No. 500 sampling point is located in the interstitial area between two buildings with a high wind speed, while the No. 788 sampling point is located in the leeward area of the building with a relatively low wind speed. For Model 1/2 which predicts the wind field in 10/30 minutes, the predicted wind speed is basically consistent with the actual data, conforming to the good accuracy of the framework. For Model 3 which predicts the wind field in 1 hour, there is a certain deviation between the predicted and actual wind speed in the first 2 hours. This is mainly because of the obvious increase in wind speed during this period, but there is no similar process in the training dataset, resulting in insufficient training of the model. It is believed that the accuracy of the framework will further increase with the expansion of the training data.

To further illustrate the performance of the framework, several machine learning models are applied to predict the pedestrian-level wind speeds as benchmarks. The comparison between the proposed framework and benchmarks is shown in Table 5. Among them, the ''input as output'' model directly takes the input wind speed at the nearest historical moment as the prediction result in the future without any machine learning structure. The BPNN is a classic machine learning model, and LSTM and GCN are both widely applied deep learning networks. Applications of these models in the field of wind environment prediction have been found in the literature [41], [42], and [43]. In this study, the input data of the

TABLE 5. Performance comparison between the proposed framework and benchmarks.

Time_Lag		1	3	6
This framework	MAE	0.038	0.079	0.116
	rMAE	3.52	7.14	10.23
"Input as output"	MAE	0.042	0.087	0.15
	rMAE	3.91	8.05	13.44
BPNN	MAE	0.075	0.119	0.166
	rMAE	6.96	10.64	15.27
LSTM	MAE	0.036	0.08	0.151
	rMAE	3.39	7.27	13.1
GCN	MAE	0.103	0.135	0.184
	rMAE	8.97	11.87	15.82

LSTM network is the same as the temporal factor branch of this framework, and the input of GCN is consistent with the spatial factor branch. The input of the BPNN model is a vector composed of historical wind speed data and meteorology data.

It can be concluded from the table that the performance of the proposed framework is better than all benchmark models, no matter how far into the future the prediction task is. Moreover, the farther the prediction moment is from the current moment, the more significant the performance advantage of the framework. It proves that the proposed framework can effectively improve the accuracy of wind environment prediction based on its hybrid structure. The prediction result of the framework outperforms any original machine learning model.

The entire pedestrian-level wind prediction process is run on a medium-configured computer (2.2 GHz GPU, 16 GB of RAM). The training process of the framework takes about 15 minutes, and the prediction of the wind environment can be completed within seconds. As a comparison, in order to obtain the 6-hour wind environment data to generate the test dataset, the numerical simulation program based on WRF and CFD runs for more than 10 hours. In conclusion, the proposed method can considerably improve the efficiency

Layer(type)	Output Shape	Param #	Connected to
wind_input(InputLayer)	(None,6,792)	0	
time_branch_ALSTM(LSTM)	(None,6,128)	471552	wind_input[0][0]
graph_input(InputLayer)	(None,6,792)	0	
meteorology_input(InputLayer)	(None,15)	0	
time_branch_FCNN(Dense)	(None,6,792)	102168	time_branch_ALSTM[0][0]
spatial_branch_FCNN(Dense)	(None,6,792)	628056	graph_input[0][0]
meteorology_branch_FCNN (Dense)	(None,792)	12672	meteorology_input[0][0]
core_feature_fusion(Add)	(None,6,792)	0	time_branch_FCNN[0][0] spatial_branch_FCNN[0][0] meteorology_branch_FCNN[0][0]
core_permute(Permute)	(None,792,6)	0	core_feature_fusion[0][0]
core_BiLSTM(Bidirectional)	(None,256)	138240	core_permute[0][0]
core_FCNN(Dense)	(None,792)	203544	core_BiLSTM[0][0]
Total params:1,556,232			

Framework parameters

$Sample\_Times = 6$

$Sample\_Points = 792$

$Time\_Lag = 6$

$LSTM\_Dim = 128$

$LSTM\_Dim = 128$

$Meteorology\_Params = 15$

$BiLSTM\_Dim = 256$

$Total\_Params = 1{,}556{,}232$

FIGURE 9. The framework structure of model 3.

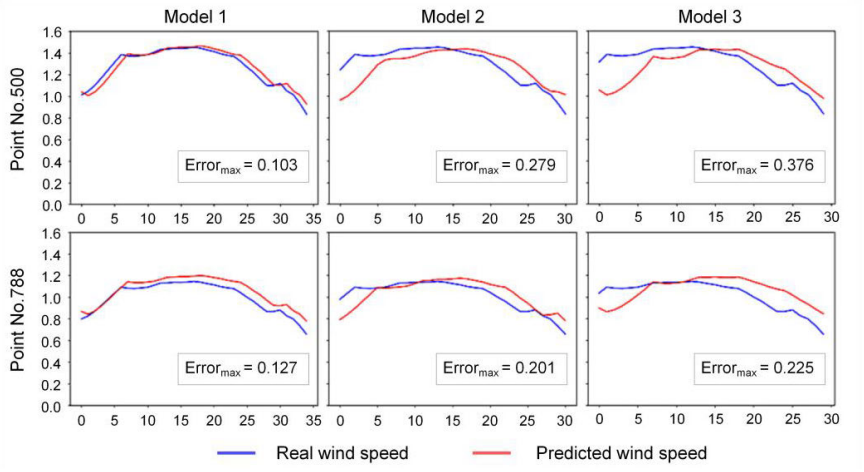

FIGURE 10. Variation of the predicted and actual wind speed.

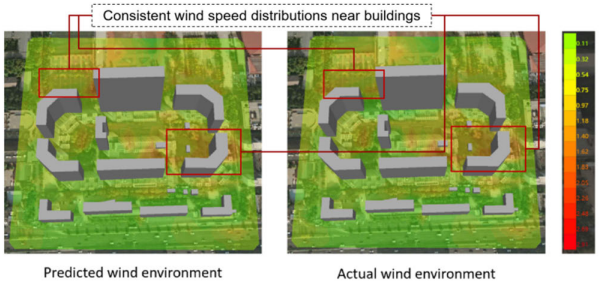

FIGURE 11. Visualization of wind environment at 10:00 on February 19, 2022.

of pedestrian-level wind environment analysis tasks, while keeping the prediction error within a reasonable threshold.

Figure 11 visualizes the predicted and actual wind environment based on the web platform developed in the author's previous work [28]. The predicted wind speed is output by model 3 ($Time\_Lag = 6$). The web platform uses color maps to illustrate the distribution of wind speed, where the speed difference of each color gradient is about 0.22m/s. Under such a visualization scheme, the visual effects of the predicted and actual wind environment are basically the same. This implies that the proposed method is accurate enough to replace the numerical simulation and perform fast pedestrian-level wind environment prediction in actual engineering scenarios.

VI. DISCUSSION AND CONCLUSION

This study introduces deep learning to propose a framework for building wind environment analysis, effectively addressing the current challenges in the existing analysis model and methods. However, there are still limitations in the proposed framework.

It is considered to improve the framework performance by exploring the use of more advanced deep learning techniques. Meanwhile, the prediction is made only based on three branches with limited features for wind environment

analysis, which should be further enriched. Additionally, a weighted combination is applied to fuse features from the three branches, and the accuracy of wind environment analysis can be enhanced by stacking models of the same type, thus reducing the prediction error in the future. Only buildings and structures are taken into consideration, with subgrid scale obstacles neglected, which also affect the local wind environment. In further works, parameterization schemes can be used to address the subgrid obstacles, namely introducing corresponding parameters or sub models, and adaptive refined mesh technology can be simultaneously adopted to ensure the capture of their influence.

Meanwhile, improvements can be made through parallel computing. Limited by the server performance, the analysis task is only supported to run in a single process. Based on parallel computing, a distributed parallel computing framework can be utilized to increase the efficiency of building wind environment analysis. Furthermore, the exploration of the application of building wind environment data is still limited. In subsequent studies, we will further leverage the application potential of the framework, including testing its reliability and practicality in a broader and more varied urban environment, investigating its application throughout the entire life cycle of buildings, and developing real-time smart city applications based on actual needs to expand its service scope.

Pedestrian-level wind environment around the building complex is closely related to human living safety and comfort. However, the current analysis and prediction for PLWE still rely on numerical simulation methods represented by CFD, which pose challenges in terms of computational efficiency and cost. This study proposes the MFF framework to promote a balance between the accuracy and efficiency of pedestrian-level wind environment analysis. The main contributions of the MFF framework are as follows:

An accurate and rapid prediction method for PLWE based on end-to-end networks: traditional multi-scale simulation methods require mesh generation and iterative calculations to realize time-consuming wind field prediction. The MFF framework proposed in this study makes direct predictions from macroscopic meteorological information to the current wind environment through end-to-end neural networks, bypassing these cumbersome steps and significantly improving the efficiency of wind environment analysis.

2) A multi-branch network structure to extract wind environment features: the framework innovatively combines various deep neural networks including GCN, and LSTM, to construct three branches and extract three features respectively, thus comprehensively considering potential influence factors to the current wind environment.

3) Calibration based on multi-scale numerical simulation data: referring to the nudging technology, results of the multi-scale numerical simulation are added to the deep learning model to reduce error accumulation and improve the model's prediction performance.

Apply the MFF framework to the case study and make a comparison with the corresponding multi-scale numerical simulation results. The results show that the MFF framework exhibits high accuracy in PLWE prediction and reduces the time and computational costs for numerical simulation, thus providing a new efficient solution for building wind environment analysis.

REFERENCES

[1] Y. Du, C. M. Mak, K. Kwok, K.-T. Tse, T.-C. Lee, Z. Ai, J. Liu, and J. Niu, "New criteria for assessing low wind environment at pedestrian level in Hong Kong," *Building Environ.*, vol. 123, pp. 23–36, Oct. 2017.
[2] Q. M. Zahid Iqbal and A. L. S. Chan, "Pedestrian level wind environment assessment around group of high-rise cross-shaped buildings: Effect of building shape, separation and orientation," *Building Environ.*, vol. 101, pp. 45–63, May 2016.
[3] X. Zhang, K. T. Tse, A. U. Weerasuriya, S. W. Li, K. C. S. Kwok, C. M. Mak, J. Niu, and Z. Lin, "Evaluation of pedestrian wind comfort near 'lift-up' buildings with different aspect ratios and central core modifications," *Building Environ.*, vol. 124, pp. 245–257, Nov. 2017.
[4] A. Abd Razak, A. Hagishima, N. Ikegaya, and J. Tanimoto, "Analysis of airflow over building arrays for assessment of urban wind environment," *Building Environ.*, vol. 59, pp. 56–65, Jan. 2013.
[5] A. U. Weerasuriya, Z. Z. Hu, X. L. Zhang, K. T. Tse, S. Li, and P. W. Chan, "New inflow boundary conditions for modeling twisted wind profiles in CFD simulation for evaluating the pedestrian-level wind field near an isolated building," *Building Environ.*, vol. 132, pp. 303–318, Mar. 2018.
[6] B. Blocken, "50 years of computational wind engineering: Past, present and future," *J. Wind Eng. Ind. Aerodynamics*, vol. 129, pp. 69–102, Jun. 2014.
[7] Y. Li, "Computational fluid dynamics technology and its application in wind environment analysis," *J. Urban Technol.*, vol. 17, no. 3, pp. 67–81, Dec. 2010.
[8] Ismail, J. John, E. A. Pane, B. M. Suyitno, G. H. N. N. Rahayu, D. Rhakasywi, and A. Suwandi, "Computational fluid dynamics simulation of the turbulence models in the tested section on wind tunnel," *Ain Shams Eng. J.*, vol. 11, no. 4, pp. 1201–1209, Dec. 2020.
[9] Y. Du, C. M. Mak, and Z. Ai, "Modelling of pedestrian level wind environment on a high-quality mesh: A case study for the HKPolyU campus," *Environ. Model. Softw.*, vol. 103, pp. 105–119, May 2018.
[10] X. Zhang, A. U. Weerasuriya, B. Lu, K. T. Tse, C. H. Liu, and Y. Tamura, "Pedestrian-level wind environment near a super-tall building with unconventional configurations in a regular urban area," in *Proc. Building Simul.*, vol. 13, 2020, pp. 439–456.
[11] H. Mittal, A. Sharma, and A. Gairola, "Numerical simulation of pedestrian level wind flow around buildings: Effect of corner modification and orientation," *J. Building Eng.*, vol. 22, pp. 314–326, Mar. 2019.
[12] J. Zhong, J. Liu, Y. Zhao, J. Niu, and J. Carmeliet, "Recent advances in modeling turbulent wind flow at pedestrian-level in the built environment," *Architectural Intell.*, vol. 1, no. 1, p. 5, Dec. 2022.
[13] S. Du, X. Zhang, X. Jin, X. Zhou, and X. Shi, "A review of multi-scale modelling, assessment, and improvement methods of the urban thermal and wind environment," *Building Environ.*, vol. 213, Apr. 2022, Art. no. 108860.
[14] K. H. Schlünzen, D. Grawe, S. I. Bohnenstengel, I. Schlüter, and R. Koppmann, "Joint modelling of obstacle induced and mesoscale changes—Current limits and challenges," *J. Wind Eng. Ind. Aerodynamics*, vol. 99, no. 4, pp. 217–225, Apr. 2011.
[15] C. Huang, J. Yao, B. Fu, J. K. Calautit, C. Zhao, J. Huang, and Q. Ban, "Sensitivity analysis of WRF-CFD-based downscaling methods for evaluation of urban pedestrian-level wind," *Urban Climate*, vol. 49, May 2023, Art. no. 101569.
[16] Z. Zhao, Y. Xiao, C. Li, P. W. Chan, G. Hu, and Q. Zhou, "Multiscale simulation of the urban wind environment under typhoon weather conditions," *Building Simul.*, vol. 16, no. 9, pp. 1713–1734, Sep. 2023.
[17] M. Mortezazadeh, Z. Jandaghian, and L. L. Wang, "Integrating CityFFD and WRF for modeling urban microclimate under heatwaves," *Sustain. Cities Soc.*, vol. 66, Mar. 2021, Art. no. 102670.
[18] P. A. Mirzaei, "CFD modeling of micro and urban climates: Problems to be solved in the new decade," *Sustain. Cities Soc.*, vol. 69, p. 102839, Jun. 2021.

[19] B. Courbet, C. Benoit, V. Couaillier, and F. Haider, "Space discretization methods," *Aerospace Lab*, vol. 2011, no. 2, pp. 1–14, 2011.
[20] Y. Ashie and T. Kono, "Urban-scale CFD analysis in support of a climate-sensitive design for the Tokyo bay area," *Int. J. Climatol.*, vol. 31, no. 2, pp. 174–188, Feb. 2011.
[21] H. Wang, H. Wang, F. Gao, P. Zhou, and Z. Zhai, "Literature review on pressure–velocity decoupling algorithms applied to built-environment CFD simulation," *Building Environ.*, vol. 143, pp. 671–678, Oct. 2018.
[22] J. Liu and J. Niu, "CFD simulation of the wind environment around an isolated high-rise building: An evaluation of SRANS, LES and DES models," *Building Environ.*, vol. 96, pp. 91–106, Mar. 2016.
[23] G. Calzolari and W. Liu, "Deep learning to replace, improve, or aid CFD analysis in built environment applications: A review," *Building Environ.*, vol. 206, p. 108315, Mar. 2021.
[24] X. Shao, Z. Liu, S. Zhang, Z. Zhao, and C. Hu, "PIGNN-CFD: A physics-informed graph neural network for rapid predicting urban wind field defined on unstructured mesh," *Building Environ.*, vol. 232, Mar. 2023, Art. no. 110056.
[25] Y. He, X.-H. Liu, H.-L. Zhang, W. Zheng, F.-Y. Zhao, M. Aurel Schnabel, and Y. Mei, "Hybrid framework for rapid evaluation of wind environment around buildings through parametric design, CFD simulation, image processing and machine learning," *Sustain. Cities Soc.*, vol. 73, Oct. 2021, Art. no. 103092.
[26] M. Mortezazadeh, J. Zou, M. Hosseini, S. Yang, and L. Wang, "Estimating urban wind speeds and wind power potentials based on machine learning with city fast fluid dynamics training data," *Atmosphere*, vol. 13, no. 2, p. 214, Jan. 2022.
[27] R. K. Reja, R. Amin, Z. Tasneem, M. F. Ali, M. R. Islam, D. K. Saha, F. R. Badal, M. H. Ahamed, S. I. Moyeen, and S. K. Das, "A review of the evaluation of urban wind resources: Challenges and perspectives," *Energy Buildings*, vol. 257, Feb. 2022, Art. no. 111781.
[28] S. Leng, S.-W. Li, Z.-Z. Hu, H.-Y. Wu, and B.-B. Li, "Development of a micro-in-meso-scale framework for simulating pollutant dispersion and wind environment in building groups," *J. Cleaner Prod.*, vol. 364, Sep. 2022, Art. no. 132661.
[29] J. Franke, A. Hellsten, K. H. Schlunzen, and B. Carissimo, "The COST 732 best practice guideline for CFD simulation of flows in the urban environment: A summary," *Int. J. Environ. Pollut.*, vol. 44, no. 4, p. 419, 2011.
[30] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A search space Odyssey," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 10, pp. 2222–2232, Oct. 2017.
[31] Q. Wang and Y. Hao, "ALSTM: An attention-based long short-term memory framework for knowledge base reasoning," *Neurocomputing*, vol. 399, pp. 342–351, Jul. 2020.
[32] M. Shimrat, "Algorithm 112: Position of point relative to polygon," *Commun. ACM*, vol. 5, no. 8, p. 434, Aug. 1962.
[33] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[34] H. Hersbach et al., "The ERA5 global reanalysis," *Quart. J. Roy. Meteorol. Soc.*, vol. 146, no. 730, pp. 1999–2049, Jul. 2020.
[35] (2023). *Baidu Map*. Accessed: Jun. 5, 2023. [Online]. Available: https://map.baidu.com/
[36] Y. Xue and C. Li, "Extracting Chinese geographic data from Baidu map API," *Stata J., Promoting Commun. Statist. Stata*, vol. 20, no. 4, pp. 805–811, Dec. 2020.
[37] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[38] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.
[39] N. Ketkar and N. Ketkar, "Introduction to keras," in *Deep Learning With Python: A Hands-on Introduction*. Berkeley, CA, USA: Apress, 2017, pp. 97–111.
[40] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, and R. Kern, "Array programming with NumPy," *Nature*, vol. 585, no. 7825, pp. 357–362, 2020.
[41] Z.-H. Guo, J. Wu, H.-Y. Lu, and J.-Z. Wang, "A case study on a hybrid wind speed forecasting method using BP neural network," *Knowl.-Based Syst.*, vol. 24, no. 7, pp. 1048–1056, Oct. 2011.
[42] T. Liang, Q. Zhao, Q. Lv, and H. Sun, "A novel wind speed prediction strategy based on bi-LSTM, MOOFADA and transfer learning for centralized control centers," *Energy*, vol. 230, Sep. 2021, Art. no. 120904.
[43] T. Stańczyk and S. Mehrkanoon, "Deep graph convolutional networks for wind speed prediction," 2021, *arXiv:2101.10041*.

ZHEN-ZHONG HU received the Ph.D. degree from the Department of Civil Engineering, Tsinghua University, China. He is currently an Associate Professor with Shenzhen International Graduate School, Tsinghua University, and the Secretary General of the BIM Specialty Committee, China Graphics Society. His research interests include information technologies in civil and marine engineering, building information modeling (BIM), and digital disaster prevention and mitigation.

YAN-TAO MIN received the B.S. degree in civil engineering from the Department of Civil Engineering, Tsinghua University, China, in 2022. She is currently pursuing the Ph.D. degree with Shenzhen International Graduate School, Tsinghua University. Her research interests include building information model (BIM) and information technologies in civil and marine engineering.

SHUO LENG received the Ph.D. degree from the Department of Civil Engineering, Tsinghua University, China. He is currently with Guangzhou Metro Construction Management Company Ltd. His research interests include building information model (BIM) and information technologies in civil engineering.

SUNWEI LI received the Ph.D. degree from the Department of Civil and Environmental Engineering, Western University, in 2012. He is currently an Associate Professor with Shenzhen International Graduate School, Tsinghua University. His research interests include offshore wind energy development and urban wind environment.

JIA-RUI LIN received the Ph.D. degree from the Department of Civil Engineering, Tsinghua University, China, in 2016. He is currently an Associate Professor with the Department of Civil Engineering, Tsinghua University. His research interests include information technology for building and civil engineering, including BIM, augmented reality (AR), cloud computing, and the Internet of Things (IoT).