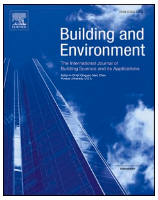
Building and Environment 188 (2021) 107500

Contents lists available at ScienceDirect

Building and Environment

journal homepage: http://www.elsevier.com/locate/buildenv



A Gaussian Process-Based emulator for modeling pedestrian-level wind field

A.U. Weerasuriya [a,b], Xuelin Zhang [c,d,e,*], Bin Lu [f], K.T. Tse [a], C.H. Liu [b]

[a] Department of Civil and Environmental Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong
[b] Department of Mechanical Engineering, The University of Hong Kong, Pokfulam, Hong Kong
[c] School of Atmospheric Sciences, Sun Yat-sen University, Zhuhai, Guangdong, China
[d] Guangdong Province Key Laboratory for Climate Change and Natural Disaster Studies, Sun Yat-sen University, Zhuhai, 519082, PR China
[e] Key Laboratory of Tropical Atmosphere-Ocean System (Sun Yat-sen University), Ministry of Education, 519000, Zhuhai, China
[f] Department of Civil and Architectural Engineering, City University of Hong Kong, Tat Chee Avenue, Kowloon Tong, Hong Kong

ARTICLE INFO

Keywords:
Gaussian process
Emulator
Pedestrian-level wind environment
Model evaluation
Lift-up building

ABSTRACT

Wind tunnel tests and computational fluid dynamics (CFD) simulations remain the main modeling techniques in wind engineering despite being expensive, time-consuming, and requiring special facilities and expert knowledge. There is a clear need for a fast, accurate, but, at the same time, computationally economical substitute. This study proposes a Gaussian Process-based (GP-based) emulator to predict the pedestrian-level wind environment near a lift-up building – an isolated, unconventionally configured building. The proposed GP-based emulator transcends the limitations of previous emulators as it can handle many inputs (8) and output parameters (384) and a large dataset (150 CFD simulations). To increase computational efficiency, the current study proposes a data reduction method based on Principal Component Analysis (PCA) and a technique to estimate hyper-parameters based on optimization. The latter can efficiently compute 250 hyper-parameters and requires no prior knowledge of their probability distributions. The emulator is faster, by a factor of 10^7 than CFD simulations in predicting wind speeds, and its accuracy is substantiated using both qualitative and quantitative analyses, which reveal that the emulator's predictions of all-prominent flow features near a building have no systematic bias, are highly accurate, and have great reproductivity.

1. Introduction

Knowledge of wind-building interactions is imperative when designing a building and its surrounding wind environment. On the one hand, wind-building interactions can threaten structural safety and affect the comfort of occupants, as wind can exert extreme wind loadings and induce excessive motions in the building [32,37,40,83]. On the other hand, the building amplifies the wind speed in its vicinity, deteriorating the quality of the wind environment and causing wind discomfort for pedestrians [13,42,51,63,84,102,103]. Conventionally, the effects of wind-building interactions are modeled using one of two popular techniques: wind tunnel tests and computational fluid dynamics (CFD) simulations. Their popularity is based mostly on the abundance of experience using them, a well-established wealth of literature and well-organized databases, and the availability of best practice guidelines. Nevertheless, conducting wind tunnel tests and CFD simulations is costly, time-consuming, and requires specific test facilities and expert knowledge, prompting engineers and researchers to find an economical, fast, and easy-to-use alternative.

Emulators [45,68,86], also known as surrogate [6,52,73], meta- [21,48], or proxy-models [9,45,66], are a fitting substitute for conventional modeling techniques. They are different from simulators (e.g. wind tunnel tests and CFD simulations). Emulators are *trained* to mimic few required outputs, rather than having to model or calculate all associated parameters of the interested case [48]. Producing only limited outputs is not necessarily a drawback of using emulators for wind engineering applications, because most applications need either wind velocity, pressure, or building responses. In fact, having limited outputs enables faster emulators and more computationally economical than their simulator counterparts. Besides, producing outputs through mimicry of simulator's results does not hamper the accuracy of the emulators, as they can estimate the uncertainty associated with predictions. Moreover, emulators are suitable for tasks that need to be repeated many

[*] Corresponding author. School of Atmospheric Sciences, Sun Yat-sen University, Zhuhai, Guangdong, China.
E-mail address: zhangxlin25@mail.sysu.edu.cn (X. Zhang).

https://doi.org/10.1016/j.buildenv.2020.107500
Received 10 August 2020; Received in revised form 6 November 2020; Accepted 28 November 2020
Available online 3 December 2020

Nomenclature

a	corner modification parameter
$\widehat{\mathbf{C}}$	matrix of truncated principle components
D	building depth
d	center core depth
FAC2	the factor of two of observations
FB	fractional bias
H	building height
h	center core height
h_a	amplitude length-scale hyperparameter
I	turbulence intensity
$\boldsymbol{I}_r$	an $r \times r$ unit matrix
K	normalized mean wind speed ratio
k	turbulent kinetic energy
l_{ij}	characteristic length-scale hyperparameter
NMSE	normalized mean square error
n_p	number of parameters
n_q	quantities of interest
n_t	number of principal components after the truncation
PCA	principal component analysis
$p_{kj}(p_{lj})$	the jth setting corresponding to the simulation run $k(l)$
PLWE	pedestrian-level wind environment
$p_{\min,j,}$ and $p_{\max,j}$	minimum and maximum of $\boldsymbol{P}_{\text{raw}}[:,j]$
$\boldsymbol{P}_{\text{raw}}$	matrix of unscaled parameters
r	number of simulation runs
R	Pearson correlation coefficient
RANS	Reynolds-Averaged Navier-Stokes [simulation]
RMSE	root mean square error
t	corner modification parameter
$\tilde{t}$	an error term of emulator predictions
U	mean wind speed
u	longitudinal mean wind speed
$\widehat{\mathbf{V}}$	matrix of truncated eigen-vectors
v	lateral mean wind speed
v_1	first shape parameter of the center core
v_2	second shape parameter of the center core
W	building width
w	center core width
$\boldsymbol{W}_{raw}$	matrix of unscaled quantities of interest (wind speed, $n_t \times r$)
z, z_{ref}	height above the ground, reference height
α	power-law exponent
β_{ij}	scaled characteristic length-scale hyperparameter
ε	turbulent kinetic energy dissipation rate
θ	building orientation/wind direction
σ_i	estimated noise
$\boldsymbol{\mu}^{\tilde{w}_i}$	mean function of GP
$\sum^{\hat{c}_i}$	covariance matrix of GP

times with minor differences in settings between iterations. This is where most simulators are found to be unfeasible [33,72]. Examples are: when assessing several building designs in the early design stage, when optimizing the configurations of a building, and when designing and disposing of buildings with standard layouts, such as cases of pre-fabricated and modular buildings.

Wind engineering applications of emulators may yet be limited, but they have already been widely used in other branches of engineering for sensitivity analysis [26,53,75], model selection [58,71], uncertainty assessment [11,44,89,97], multi-scale modeling [8,16,25], model coupling [61,74,82], system design [5,17,60], and inverse identification [2,34,54,62]. In the few applications in wind engineering, emulators have mostly been developed based on an Artificial Neural Network (ANN) and used to optimize the shapes and aerodynamics of tall buildings [27,28,93], to predict dynamic responses of tall buildings [67,69], to model bridge aerodynamics [1,43,76,94], to estimate wind pressure on buildings [15,22,24], predict interference effects [29,46], wind speed forecast [41,79] and to estimate how the effect of topography features cause wind to speed up [10]. Besides, polynomial chaos expansion based emulators have been used to model the urban wind environment [80]; estimate uncertainty in CFD simulations [31,35,85] and to model how pollutants disperse [49]. Several studies have employed support vector regression as an emulator to forecast short-term wind speeds [19,47,55] and daily air pollutant concentrations [65,70,96]. Another set of emulators have been developed based on Gaussian Process-based (GP-based) regression to predict short- [39,99] and long-term [98] wind speeds, the power generation of wind turbines [20,57,78], and wind pressure prediction [56].

Moonen & Allegrini [64] have methodically introduced the GP-based emulator to wind engineering applications. They developed the GP-based emulator to predict the pedestrian-level wind environment (PLWE) in an urban-like setting. The urban setting consists of two rectangular and 12 square buildings forming a street canyon and its surroundings. The wind environment in this urban setting is evaluated for six wind directions: $\theta = 0°$, $20°$ $30°$, $35°$, $60°$, and $90°$, using CFD simulations and the emulator [64]. point out the great promise in saving computational resources that the GP-based emulator shows in wind engineering applications. Nevertheless, certain aspects of the emulator must be further explored, as discussed below:

O'Hagan [68] points out that GP-based emulators are more efficient and flexible than other emulators despite their high computation costs handling large datasets [23,59]. This shortcoming can be overcome by employing a dimensional reduction-based approach [4]. Indeed, Moonen & Allegrini [64] employed such an approach to improve the efficiency of a GP-based emulator but it is ambivalent whether they have fully explored the potential of the emulator yet. The uncertainty arises from the simplicity of their case study, which has two input parameters – wind speed and direction – and carries out eight simulation runs – all combinations between two wind speeds and four wind directions – for a simple urban-like setting with rectangular and square buildings. Therefore, it is yet to be established how efficiently a GP-based emulator can handle many input parameters and a large dataset of many simulations conducted on unconventional building configurations.

The current study further explores the potential of using a GP-based emulator to predict the PLWE near a building with an unconventional configuration. Such an unconventional configuration serves two purposes: first, it provides many inputs for the emulator in the form of design parameters; and second, the PLWE near that building with the unconventional configuration may have special flow features that are outright absent in the wind environment near a rectangular or square-shaped building [95,103]. These features make it possible to gauge how well the GP-based emulator can handle many input parameters, and how accurately it can predict flow characteristics in a complex wind field. In this study, the unconventional configuration is an isolated building with a lift-up design [88,100,101]. The particular selected configuration has eight design parameters, quadrupling the input parameters used by Ref. [64]. The current study also investigates how efficiently a particular data reduction technique can develop the GP-based emulator by employing a large dataset based on 150 CFD simulations. In addition, a novel technique based on optimization to determine the magnitudes of hyper-parameters is proposed, as current practices are inefficient in estimating the magnitudes of hundreds of hyper-parameters arising from larger datasets and many input parameters in cases such as those presented in the current study.

The overall framework for developing a GP-based emulator is briefly discussed in Section 2, and subsequent sections describe separately the main steps of the framework. Section 3 presents the details of the CFD simulations, including building models, computational domain, boundary conditions, solver settings, and the estimation of the accuracy of the CFD simulation. Section 4 demonstrates the dimensional reduction-based approach employed to reduce the size of the dataset of this study. Section 5 discusses the underlying theories and reasoning behind the selection of model parameters of the GP-based emulator. Section 6 evaluates the prediction accuracy of the emulator with respect to the data from the CFD simulations. Section 7 demonstrates the practical significance of the proposed GP-based emulator using several examples. Section 8 discusses some limitations of the current study, and Section 9 closes with some concluding remarks.

2. Framework

Fig. 1 shows the main steps in developing the GP-based emulator. The parallelograms are the main tasks of the process, and the rectangles are tools that are used to execute the tasks. The process starts with choosing design parameters for the emulator; in this study, they are the dimensions of a lift-up building and the mean wind speed at the pedestrian level near the building (Section 4.1). Eight dimensions — height (*H*) and width (*W*) of the elevated structure; height (*h*), width (*w*), depth (*d*), two shape parameters: $v_1 = t/d$ and $v_2 = a/t$ of the center core, and wind direction (θ) (or orientation of the building) — define the design of the building (Fig. 2). The output parameters are longitudinal (*u*) and lateral (*v*) mean wind speeds of 194 points at the pedestrian level (2-m height in full scale) near the building. A database that contains different combinations of design parameters as well as corresponding outputs is developed for training the emulator.

In this study, Design of Experiment (DoE) is used to select the design parameters of different lift-up designs while the three-dimensional steady-state Reynolds-averaged Navier-Stokes equation-based simulation (3D SRANS) is employed to model the PLWE (Section 3.1). Generally, the input and output data are inhomogeneous in terms of magnitude and units. This necessitates the next major task – data preprocessing, which eliminates the differences in design parameters and outputs. This study employs normalization and standardization as techniques for data preprocessing of the input and output (Section 4.3). In the next step, the preprocessed data are subjected to data reduction, which extracts all important variances in the output space using a smaller dataset than the database. Principal Component Analysis (PCA) is employed for data reduction in this study (Section 4.3). Nevertheless, any of the following techniques could have served the same purpose: low variance filter, high correlation filter, random forests/ensemble trees, backward feature elimination, forward feature construction [90]. The emulator is developed as a stochastic model based on the Gaussian Process (GP) to predict the mean wind speeds (Section 4.4). Hyper-parameters, whose individual values can be determined using prior knowledge of their probability distributions and predefined upper and lower bounds, form an essential part of the GP-based emulator [64]. In the current study, an optimization-based technique is used to estimate the hyper-parameters (Section 4.5). Unlike the method followed by Ref. [64]; the novel method does not require prior knowledge of the properties of the hyper-parameters. It is therefore advantageous for determining a large set of hyper-parameters, whose probability distributions are difficult to be determined in advance. In the last step, the accuracy of the GP-based emulator is estimated by conducting performance evaluation and sensitivity analysis. The performance evaluation has qualitative and quantitative components. In the qualitative analysis, visual differences are identified between the outputs of the emulator and the high-quality CFD simulations (Section 5.1). They are then quantified using statistical indices (Section 5.2) and validation metrics (Section 5.3). The sensitivity analysis ensures that subjective decisions, such as the size of the database and degree of data reduction made during emulator development, do not affect the emulator's predictions (Section 6). If the results of the performance evaluation and sensitivity analysis are satisfactory, the emulator development process is complete, and the emulator is ready for use.

3. CFD simulation

The CFD simulation of this study is carried out using the commercial software package ANSYS FLUENT v.19.2 using three-dimensional steady-state Reynolds-averaged Navier-Stokes (3D SRANS) equations and a 2-equation turbulence closure model. The CFD simulations serve two purposes: first, pedestrian-level wind speed data from the CFD simulations form a database based on which the GP-based emulator will be developed; second, they offer an independent dataset for estimating

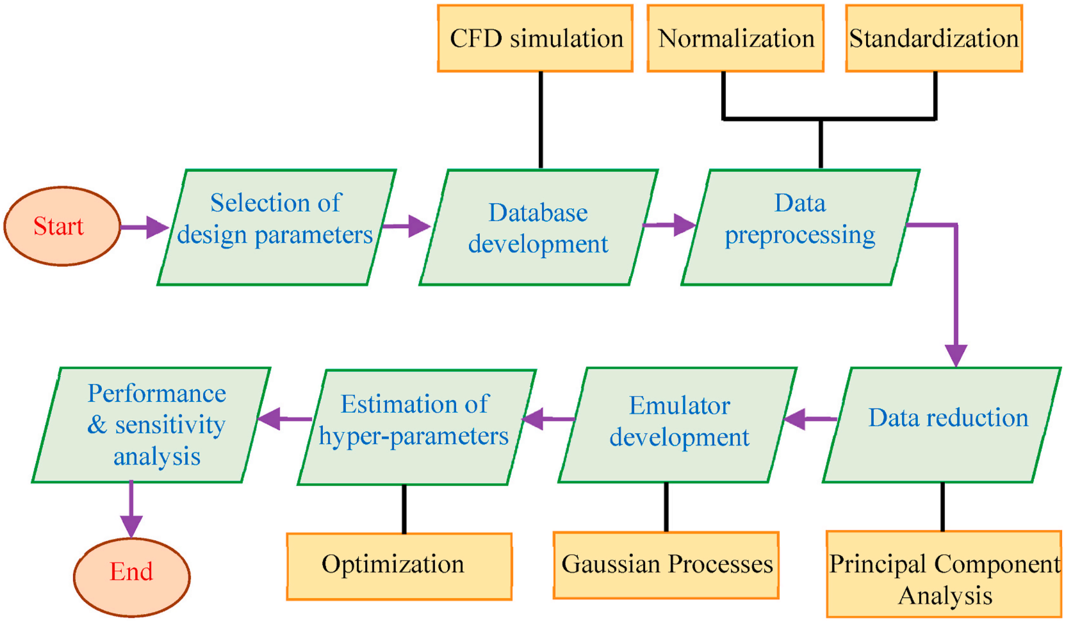


Fig. 1. The framework for developing the GP-based emulator.

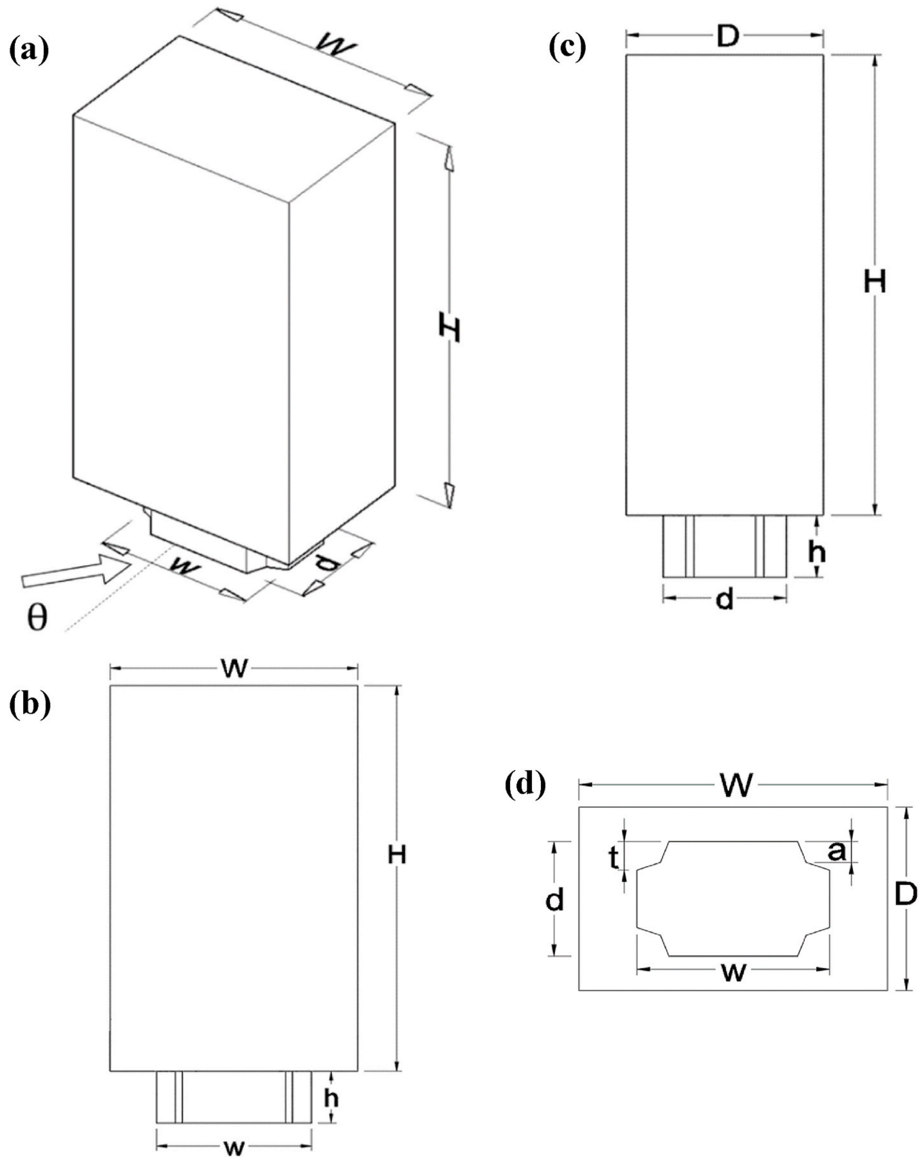


Fig. 2. (a) Schematic diagram, (b) front elevation, (c) side elevation, and (d) plan view of a lift-up building.

the accuracy of the emulator. Before creating the database, the accuracy of the CFD simulations is estimated using the pedestrian-level wind speed data from a wind tunnel test. The selected wind tunnel test modeled a reduced-scale model (1:200 length scale) of a lift-up building with dimensions $H = 600$ mm × $W = 150$ mm × $D = 100$ mm and $h = 30$ mm × $w = 75$ mm × $d = 50$ mm in an atmospheric boundary layer wind flow. The mean wind speed averages data measured at 90 points at the pedestrian level (10 mm in model scale) near the lift-up building using Irwin sensors and thermistor wind speed sensors. Complete details of the wind tunnel test can be found in Ref. [88,101]; and [100].

The same lift-up building is modeled in CFD simulation using a computational domain with these dimensions: length = 13H × width = 6H × height = 4H, where H is the height of the building (600 mm) (Fig. 3). The lift-up building is located at distances of 3H and 10H from the inlet and outlet, and the lateral and top boundaries are 3H away from the nearest building walls. The blockage ratio of this setup is 1%, which satisfies the maximum allowable blockage ratio of 3% as recommended by the best practice guidelines [30,87]. Of note is that the current study did not attempt to simulate the wind tunnel's test section as the computational domain because the dimensions of the test section (5 m × 4 m) were substantial compared to the largest building model (0.15 m

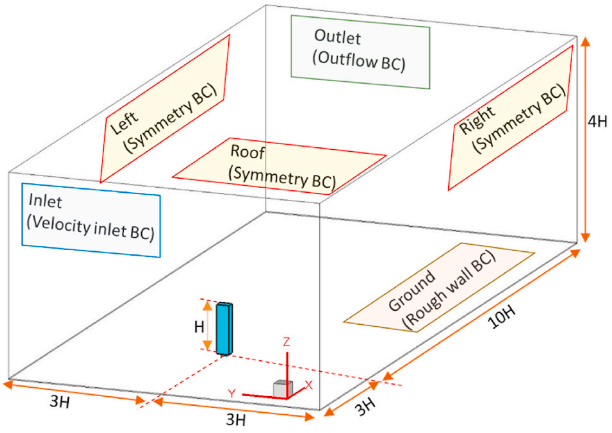


Fig. 3. The dimensions of the computational domain.

× 0.6 m) tested in the wind tunnel. Therefore, it is reasonable to assume that the wind fields near the buildings are free from the influence of the sidewalls and ceiling of the wind tunnel, and that conditions can be equated to the symmetry boundary condition used for the lateral and top boundaries of the computational domain. The computational domain is discretized into small hexahedral cells using the FLUENT MESHING tool, such that cell sizes gradually increase from smallest near the lift-up building to largest near the boundaries of the computational domain (Fig. 4). Five prism layers are created between the ground and the pedestrian level (10 mm height) to improve the accuracy of modeling wind flow close to the ground.

The realizable k-ε turbulence model is selected as the turbulence closure model because of its superior performance in modeling mean flows pertaining to complex structures, flows that involve rotations, boundary layers under strong adverse pressure gradients, and flow separation and recirculation (Davis et al., 2012). The profiles of mean wind speed (U), turbulent kinetic energy (k), and turbulent kinetic energy dissipation rate (ε) shown in Fig. 5 are defined as boundary conditions at the inlet. The derivations of these profiles follow Eqs. (1)–(3) and utilize mean profiles of wind speed and turbulence intensity measured in the wind tunnel.

$$U(z) = U_{ref}\left(\frac{z}{z_{ref}}\right)^{\alpha} \tag{1}$$

where U_{ref} is the reference wind speed, $U_{ref} = 7.5\ \mathrm{ms}^{-1}$ at the reference height $z_{ref} = 0.6$ m, and α is the power-law exponent, which equals 0.12.

$$k(z) = 1.5 \times (I(z)U(z))^2 \tag{2}$$

where $I(z)$ is the vertical profile of the turbulence intensity measured in the wind tunnel.

$$\varepsilon(z) = C_{\mu}^{1/2}k(z)\frac{U_{ref}}{z_{ref}}\alpha\left(\frac{z}{z_{ref}}\right)^{(\alpha-1)} \tag{3}$$

where C_{μ} is a constant that equals 0.09.

The outflow boundary conditions $\partial(u,v,w,k,\varepsilon)/\partial x$ and $\partial(u,v,w,k,\varepsilon)/\partial y = 0$ are applied to the outlet, and the symmetry boundary conditions $\partial(u,v,w,k,\varepsilon)/\partial y$ and $\partial(u,v,w,k,\varepsilon)/\partial z = 0$ are assigned for the lateral and top boundaries of the computational domain. The ground is modeled as a rough wall with the standard wall function [50] and the sand-grain equivalent roughness height [18] $K_s = 0.27$ mm with the roughness constant $C_s = 0.5$. The walls of the building are modeled as smooth walls. The semi-implicit method for pressure linked equations (SIMPLE) algorithm is used for pressure-velocity coupling, and the pressure interpolation is of second order. Second-order discretization schemes are used to solve for the convection and viscous terms in the governing equations. The results of the CFD simulations are considered to have converged when the residuals of iteration have reached these values: continuity – 10^{-6}, x-, y-, z-momentum – 10^{-7}, k – 10^{-7}, and ε – 10^{-7}, and show no further reduction as the number of iterations increased.

A grid sensitivity test is conducted to select a suitable grid from Coarse, Intermediate, and Fine grids for the CFD simulation. The Coarse grid has 496,556 cells with a minimum cell size of 10 mm, while the Intermediate and Fine grids have 1,061,679 and 3,896,841 cells and minimum cell sizes of 8 mm and 5 mm, respectively. The grid independence results are assessed to calculate root-mean-square error (RSME) (Eq. (4)) using the K values at corresponding locations in any two grids.

$$RMSE = \sqrt{\frac{1}{n}\sum\nolimits_{i=1}^{n}\left(K_{1,i} - K_{2,i}\right)^2} \tag{4}$$

where $K_{1,I}$, and $K_{2,i}$ are normalized wind speed ratios at similar locations in grid 1 and 2. The normalized mean wind speed ratio K is defined as:

$$Normalized\ \ mean\ \ wind\ \ speed\ \ ratio\ \ (K) = \frac{U_{i,2m}}{U_{i,2m,ambient}} \tag{5}$$

where U_{2m} is mean wind speed at location i at a 2 m height (pedestrian level) and $U_{i,2m,\ ambient}$ is the mean wind speed at the same location and height but in the absence of the building.

Fig. 6(a) shows the locations where K values are compared, and Fig. 6(b)–(d) compare the K values of the three grids. The differences in K values of the Coarse-Intermediate and the Corse-Fine grids are more than ±20%, and the corresponding RMSE values are higher than 0.2621 and 0.2183, respectively. Conversely, the Fine and Intermediate grids have similar K values over a wide range where the difference is less than ±10%, and the RSME value is 0.0837. Similar K values and smaller RMSE imply that the Intermediate grid can produce grid-independent results. Taking consideration of the accuracy and smaller computational cost, this study uses the Intermediate grid for the rest of the CFD simulation.

Fig. 7 shows the comparison of the pedestrian-level wind speeds between the CFD simulation and wind tunnel data. The wind speeds at the pedestrian-level ($z = 10$ mm) are extracted at 75 mm intervals on 11 horizontal lines within 375 mm distances upstream and downstream of the building. The CFD and wind tunnel test data show good agreement in the area with high wind speeds ($K \geq 1$), where their discrepancy is less than 15%. However, the discrepancy exceeds 15% downstream of the building, particularly in the wake of the building where wind speeds are low ($K \leq 1$). CFD simulation tends to underpredict the wind speeds in the wake of the building because 3D SRANS cannot reproduce the vortex shedding that occurs behind the building, consequently underestimating the kinetic energy in the wake and overestimation of the wake of the building [12,87]. Barring this shortcoming, 3D SRANS is sufficiently accurate and computationally competent to build the pedestrian-level wind speed database for developing and validating the GP-based emulator.

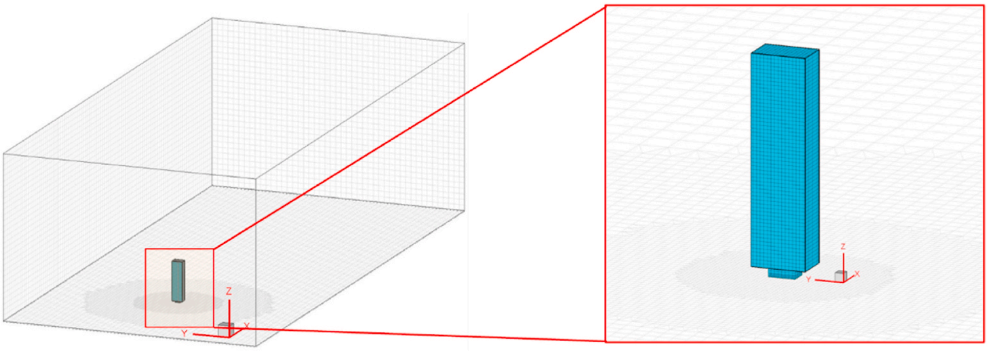


Fig. 4. The grid arrangement in the computational domain and around the lift-up building.

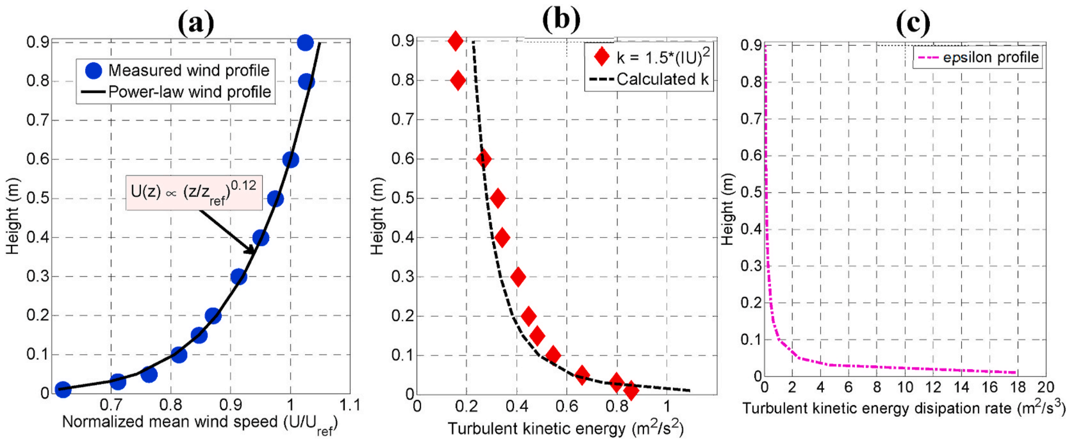


Fig. 5. Inflow boundary conditions of CFD simulations (a) mean wind speed profile (U), (b) turbulent kinetic energy (k) profile, (c) turbulent kinetic energy dissipation (ε) profile.

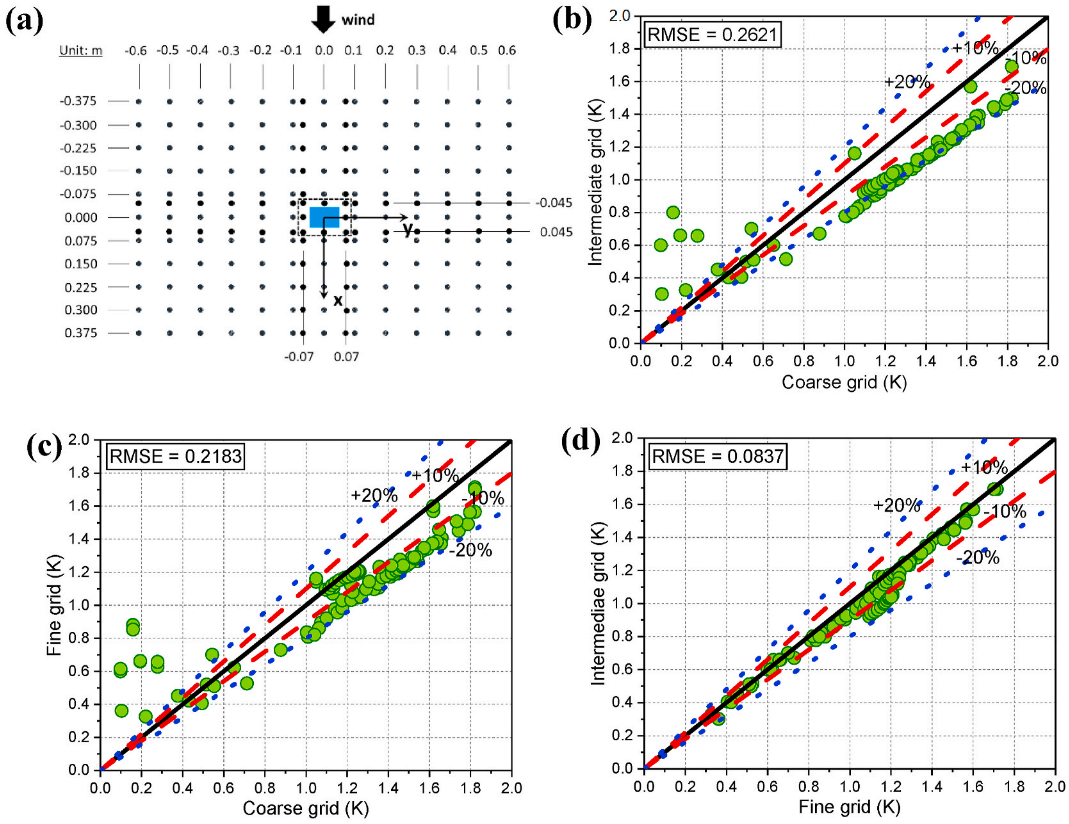


Fig. 6. (a) 194 locations of K value extraction (in model scale), and the comparison of K value (b) Coarse – Intermediate grids, (c) Coarse-Fine grids, (d) Fine-Intermediate grids.

4. Emulator development

4.1. Design space and input parameters

The eight parameters of the lift-up design result in $n_p = 8$ unique simulation settings. Each input has its upper and lower bounds, which are similar to the largest and smallest dimensions of the lift-up buildings tested in the wind tunnel by Refs. [88,101]; and [100] (Table 1).

A total of 150 CFD simulations ($r = 150$) are conducted, covering the upper and lower bounds of the eight parameters, to create the database (Fig. 8). The settings of each simulation follow a hybrid design of experiment (DoE) technique which combines Latin Hypercube Sampling (LHS) and random generation of settings. The database is subsequently divided into various sizes of design space ranging from 10 to 150 simulations for training and testing the emulator. The following subsections detail the development of the emulator, and its results based on a design space of 50 simulations ($r = 50$). The impact of the size of the design space on emulator performance is discussed in detail in the sensitivity analysis (Section 6).

4.2. Output space

The output space contains the quantities of interest stored in an $r \times n_q$ matrix $\boldsymbol{W}_{\text{raw}}$, where n_q denotes the number of quantities of interest, i.e.,

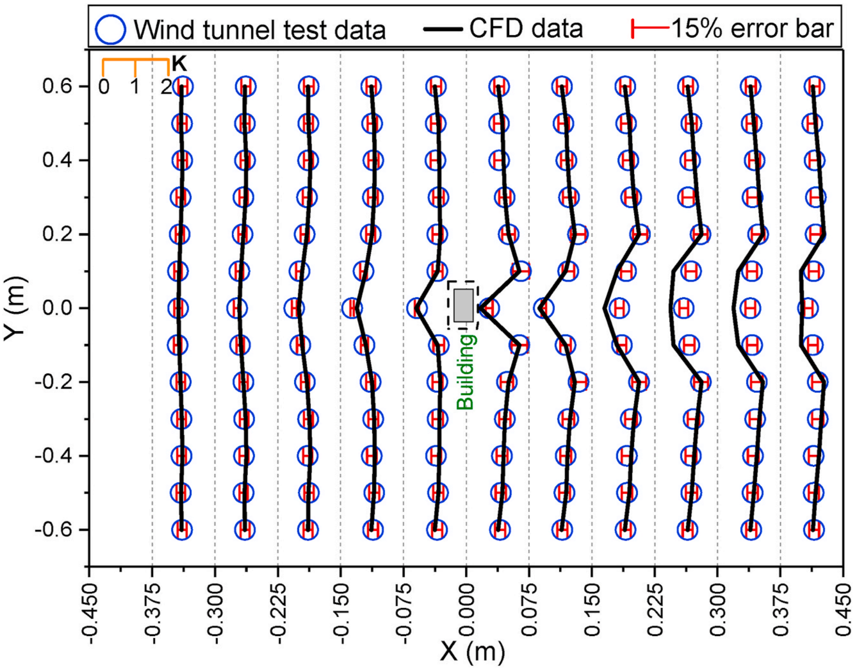


Fig. 7. Comparison of normalized wind speed (*K*) at the pedestrian level in the wind tunnel test and CFD simulation (building not to scale).

Table 1
Design parameters and their upper and lower bounds.

	Design parameter	Upper and lower bounds
Building	Height (*H*)	45 m < *H* < 120 m
	Width (*W*)	30 m < *W* < 90 m
Center core	Height (*h*)	3 m < *h* < 9 m
	Width (*w*)	9 m < *w* < *W*
	Depth (*d*)	6 m < *d* < min (20 m, *w*)
	$v_1 = t/d$	$0 < v_1 < 1/3$
	$v_2 = t/a$	$-1 < v_2 < 0$
Orientation	θ	0° < θ < 45°

$\boldsymbol{W}_{\mathrm{raw}}$ has r rows of n_q quantities of interest extracted from an r number of CFD simulations. In this study, n_q columns consist of u and v components of mean wind speed at the pedestrian level extracted from 194 points in a 240 m × 150 m area around the lift-up building (Fig. 6(a)). The parameter settings are stored in an $r \times n_p$ matrix $\boldsymbol{P}_{\mathrm{raw}}$, where each row corresponds to the outputs in $\boldsymbol{W}_{\mathrm{raw}}$.

4.3. Data preprocessing: standardization, normalization, and data reduction

Data preprocessing increases the efficiency of the emulator by removing differences in units and order of magnitude and removes any bias in the input space. Data scaling, through normalization and standardization, is employed to preprocessing the input and output spaces. Normalization is applied to the input space to map the data onto the interval [0, 1], as in Eq. (6):

$$\mathbf{P}[:,j] = \frac{\mathbf{P}_{\mathrm{raw}}[:,j] - p_{\min,j}}{p_{\max,j} - p_{\min,j}} \tag{6}$$

where $\boldsymbol{P}_{\mathrm{raw}}[:,j]$ is the vector of the unscaled values of the parameter j. $p_{\min,j}$ and $p_{\max,j}$ are the respective minimum and maximum of $\boldsymbol{P}_{\mathrm{raw}}[:,j]$.

Standardization scales the output space such that it has zero mean and unit variance. Standardization follows Eq. (7):

$$\mathbf{W}[:,j] = \frac{\mathbf{W}_{\mathrm{raw}}[:,j] - \mathrm{w}_{\mathrm{avg},j}}{w_{\mathrm{std},j}} \tag{7}$$

where $\boldsymbol{W}_{\mathrm{raw}}[:, j]$ is the vector of the unscaled values of the quantity of interest j. $w_{\mathrm{avg},j,}$ and $w_{\mathrm{std},j}$ are the average and standard deviations of $\boldsymbol{W}_{\mathrm{raw}}[:, j]$, respectively.

Principal Component Analysis (PCA) is employed for data reduction in this study. PCA decomposes $\boldsymbol{W}[r, n_q]$ into a score matrix $\boldsymbol{C}[r, n_q]$ of the principal components and a loading matrix $\boldsymbol{V}[n_q, n_q]$, whose columns are the eigenvectors of $\boldsymbol{W}^{\mathrm{T}}\boldsymbol{W}$ such that $\boldsymbol{W} = \boldsymbol{C}\boldsymbol{V}^{\mathrm{T}}$. PCA rearranges the original dataset by variance in descending order, i.e., the first principal component or the first column of $\boldsymbol{C}$ represents the largest variance of the

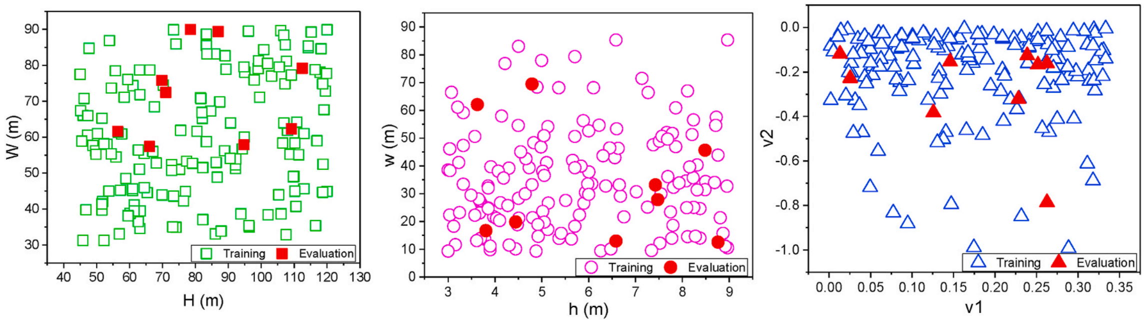


Fig. 8. Distributions of six design parameters: *H*, *W*, *h*, *w*, v_1, and v_2 of training (open markers) and evaluation (filled markers) datasets in the design space.

dataset. Fig. 9 shows the contribution of each principal component to the variance of the dataset, of which more than 98% is captured by the first 25 principal components (98.11%). As shown in Fig. 8, the first principal component reproduces more than 48.2% of the variance of the original dataset, and the contribution of the first 10 principal components is more than 91.79%. By using the first 25 principal components, the GP-based emulator achieves (388–25)/388 = 93.56% data reduction without losing more than 2% of the original dataset. To limit this loss to 1% (i.e., covering 99% of the variance), the emulator needs to use the first 31 principal components. This leads to little information gain (0.91%) but a higher computation cost. Taking this extra computation cost into consideration, the study develops the emulator using the first 25 principal components. The impact of the degree of data reduction is assessed in a sensitivity analysis (Section 6).

The data reduction limits the number of PCA to $n_t < n_q$ but introduces a truncation error to the emulator [81]. With the data reduction and the truncation error, $\boldsymbol{W}$ can be rearranged as:

$$\mathbf{W} = \widehat{\mathbf{C}}\widehat{\mathbf{V}}^{\mathrm{T}} + \widehat{\mathbf{T}} \tag{8}$$

where $\widehat{\mathbf{C}}$ is an $r \times n_t$ truncated score matrix, $\widehat{\mathbf{V}}$ is an $n_q \times n_t$ truncated loading matrix, and $\widehat{\mathbf{T}}$ is an $r \times n_q$ matrix of the truncation error.

4.4. The emulator

The objective of the emulator is to predict an output $\boldsymbol{y}_{em}$ based on an unknown function of inputs, $f(\boldsymbol{x})$, to estimate any uncertainty, ε, associated with the prediction. The most straightforward method is Bayesian Linear Regression (BLR) [14]. This process can be mathematically expressed as:

$$y_{em} = \mathbf{w}\mathbf{x}^{\mathrm{T}} + \boldsymbol{\varepsilon} = f(\mathbf{x}) + \boldsymbol{\varepsilon} \tag{9}$$

where $\boldsymbol{y}_{\mathrm{em}}$ can be expressed as a product of the input $\boldsymbol{x}$ and weight matrix $\boldsymbol{w}^{\mathrm{T}}$. Here, the unknown function $f(\boldsymbol{x})$ can be anything from a simple linear regression function to a specific high-order, nonlinear regression. However, the selected function should be flexible enough to capture underlying trends in the input rather than fitting data to mimic the simulator. To deal with these highly nonlinear problems, an approach based on the BLR and kernel method (nonlinear transformation) is adopted for the current study. Inspired by the study of Moonen & Allegrini [64]; this study has selected the Gaussian Process (GP) as the stochastic process to determine the unknown function $f(\boldsymbol{x})$. The GP defines a Gaussian-type probability distribution for each function value, and its finite collection of function values is a multivariate normal distribution. Owing to these inherent characteristics, GP can be specified by a mean function and a covariance function.

Since the current study adopts a data reduction approach, the GPs can be used to estimate the wind speed by constructing a statistic model as shown in Eqs. (10) and (11).

$$\tilde{\mathbf{w}}_{\mathrm{em}} = \tilde{\mathbf{c}}\widehat{\mathbf{V}}^{\mathrm{T}} + \tilde{\mathbf{t}} \tag{10}$$

with $\tilde{\mathbf{c}} = [\tilde{\mathbf{c}}_1, \tilde{\mathbf{c}}_2, \ldots, \tilde{\mathbf{c}}_{n_t}]$ and

$$\tilde{\mathbf{c}}_i = \mathrm{GP}\left(\boldsymbol{\mu}^{\tilde{\mathbf{c}}_i}, \sum^{\tilde{\mathbf{c}}_i}\right) \tag{11}$$

where $\tilde{\mathbf{w}}_{\mathrm{em}}$ is an $r \times n_q$ matrix of the emulator's results; $\tilde{\mathbf{c}}_i$ is the ith ($i = 1, 2, \ldots, n_t$) independent GP with mean function $\mu^{\tilde{\mathbf{c}}_i}$ and covariance function $\sum^{\tilde{\mathbf{c}}_i}$; $\tilde{\mathbf{t}}$ represents the error term to account for the approximation and truncation errors of the emulator, which will be in the estimation of $\tilde{\mathbf{c}}_i$ as a zero-mean normal distribution with finite variance $N(0, \boldsymbol{\Sigma}^{\tilde{e}_i})$. In this study, $\boldsymbol{\Sigma}^{\tilde{e}_i}$ is estimated to be $\sigma_i\mathbf{I}_r$, and the estimation of σ_i is similar to other parameters, as discussed in Section 4.5.

The term $\boldsymbol{\mu}^{\tilde{\mathbf{c}}_i}$ is the mean value of GP, about which variations occur. The covariance function ($\sum^{\tilde{\mathbf{c}}_i}$) denotes the correlation of different simulation runs, and it is assumed to be in the form of:

$$\sum^{\tilde{\mathbf{c}}_i} = h_a^i \prod_{j=1}^{n_p} \exp\left(-l_{ij}\left(p_{kj} - p_{lj}\right)^2\right) \tag{12}$$

where h_a and l_{ij} are parameters that need to be estimated; $p_{kj}(p_{lj})$ is the jth setting of the simulator run $k(l)$. The selected covariance function may lead to a large or small value depending on the spatial correlation between the two sets of parameter settings for simulation runs k and l. The parameter l_{ij} controls the correlation decay rate and has values between 0 and infinity ($l_{i,j} \in [0, +\infty]$). The value of l_{ij} is determined using Eq. (13).

$$l_{ij} = -4\ln\left(\rho_{ij}\right) \tag{13}$$

where $\rho_{\mathrm{ij}} \in [0, 1]$ and its values close to 1 indicate the weight ith principal component strongly depends on setting j. The extreme values of ρ_{ij}, i.e., 0 and 1, indicate rapid changes in the emulator predictions and uncertainty, and irrelevant settings that exhibit no correlation with predictions, respectively.

To complete the emulator development, the values of many model

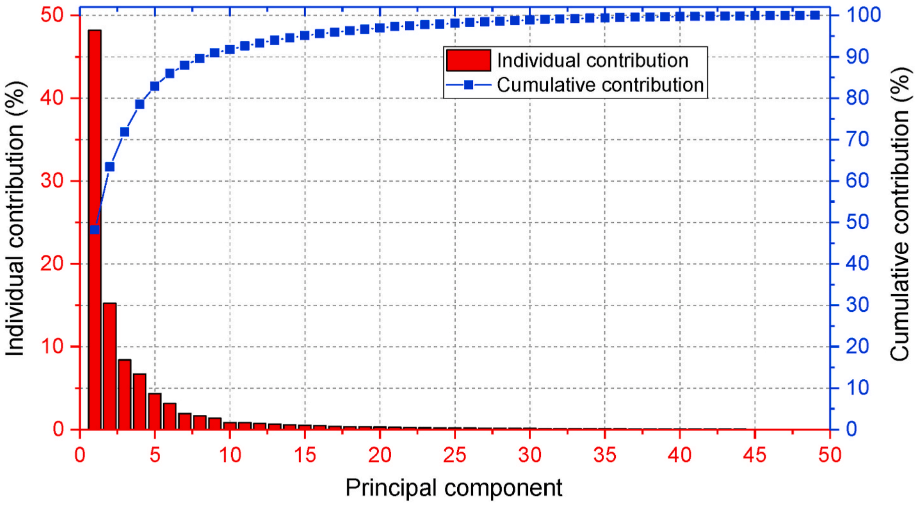


Fig. 9. The individual and cumulative contributions of principal components to the original dataset (r = 50).

parameters must be determined through model calibration. These parameters, commonly known as hyper-parameters, consist of one amplitude length-scale hyper-parameter (h_a) and n_p scaling parameters ρ_{ij} for each of the principal components and a precision parameter that controls the variance of the error (σ_i). The total number of hyper-parameters of the proposed emulator is 250 as determined by the equation $n_t(2 + n_p)$. It should be noted that the number of hyper-parameters in the current study (250) is significantly higher than that (16) of [64]. Determining their prior distributions is therefore a daunting task, as discussed in the following section.

4.5. Calibration of hyper-parameters

Moonen & Allegrini [64] used the Bayesian concept to calibrate the 16 hyper-parameters by multiplying their joint prior distributions and likelihood. However, this method cannot be adopted for the current study because of the large number of hyper-parameters. Defining their prior distribution would be impractical. Alternatively, the marginal likelihood method is employed in this study because it does not require knowledge of the prior distributions of the hyper-parameters. Given a specific group of hyper-parameters, the marginal likelihood assumes a multi-variable normal distribution of the hyper-parameters (Williams and Rasmussen 2006) and can be calculated as:

$$\mathrm{ML}\left(\tilde{\mathbf{c}}_i\middle|\mathbf{P},\mathbf{h}^i\right)=\frac{1}{\sqrt{(2\pi)^r}\left|\sum^{\tilde{\mathbf{c}}_\mathbf{i}}\right|^{\frac{1}{2}}}\mathrm{e}^{-\frac{\left(\tilde{\mathbf{c}}_i-\boldsymbol{\mu}\right)\left(\sum^{\tilde{\mathbf{c}}_i}\right)^{-1}\left(\tilde{\mathbf{c}}_i-\boldsymbol{\mu}\right)}{2}} \tag{14}$$

where h^i is the hyper-parameters for the covariance of the ith set of principal components $\tilde{\mathbf{c}}_i$, and $\boldsymbol{P}$ is the parameter matrix. $\boldsymbol{\mu}$ is the mean value vector of principle components. Because of data standardization, $\boldsymbol{\mu}$ is equal to zero.

For ease of the mathematical operation, Eq. (14) is expressed in natural log, as shown in Eq. (15).

$$\mathrm{logML}\left(\tilde{\mathbf{c}}_i\middle|\mathbf{P},\mathbf{h}^i\right)=-\frac{1}{2}\tilde{\mathbf{c}}_i\left(\sum^{\tilde{\mathbf{c}}_i}\right)^{-1}\tilde{\mathbf{c}}_i-\frac{1}{2}\log\left|\sum^{\tilde{\mathbf{c}}_i}\right|-\frac{r}{2}\log 2\,\pi \tag{15}$$

Considering the noise of the model,

$$\mathrm{logML}\left(\tilde{\mathbf{c}}_i\middle|\mathbf{P},\mathbf{h}^i\right)=-\frac{1}{2}\tilde{\mathbf{c}}_i\left(\sum^{\tilde{\mathbf{c}}_i}+\sigma_i\mathbf{I}_r\right)^{-1}\tilde{\mathbf{c}}_i-\frac{1}{2}\log\left|\sum^{\tilde{\mathbf{c}}_i}+\sigma_i\mathbf{I}\right|-\frac{r}{2}\log 2\,\pi \tag{16}$$

In Eq. (16), $\frac{1}{2}\tilde{c}_i^T(\sum^{\tilde{c}_i}+\sigma_i\boldsymbol{I}_r)^{-1}\tilde{c}_i$ measures the data fit condition, $\frac{1}{2}\log\left|\sum^{\tilde{c}_i}+\sigma_i\boldsymbol{I}\right|$ is the complexity penalty, $\frac{r}{2}\log 2\,\pi$ is a normalization constant, which can be neglected when maximizing the marginal likelihood (Williams and Rasmussen 2006). The remnant part, which only depends on $\sum^{\tilde{c}_i}$, is subsequently determined by the hyper-parameters. It should be noted that the current study does not determine the value of each hyper-parameter separately, but estimates their values as a certain combination to obtain the highest logarithmic marginal likelihood. This concept deduces the calculation process of hyper-parameters into optimization, which estimates the global optimal set of hyper-parameters to maximize the log marginal likelihood.

The Genetic Algorithm (GA) is used as the optimization algorithm because of its ability to determine the global optimum solution without being trapped in local maxima. This superior ability of GA is attributed to three parameters: selection, crossover, and mutation, which enable it to produce offspring with excellent performance. The process starts by generating an initial population containing different hyper-parameters. After establishing the initial population in the binary system, GA creates generations using mutation and crossover to maximize the log marginal likelihood. The optimum numbers of mutation and crossover are estimated by trial and error, and the size of the population and the maximum number of iterations are set as 300 and 150, respectively. The optimal settings of the hyper-parameters are determined as the best-fitted set of offspring at the maximum number of iterations or the last iteration, after which point the value of the log marginal likelihood does not change even when the number of iterations continues to increase. The reliability of the optimization is ensured by running the optimization four times and obtaining identical optimal solutions across the runs.

4.6. Posterior distribution

With the hyper-parameters now known, the emulator can predict the posterior distribution of the object function. The construction of the covariance matrix of ith principal components can be done as follows:

$$\sum\nolimits_*^{\tilde{\mathbf{c}}_i}=h_a^i\prod_{j=1}^{n_p}\exp\left(-l_{ij}\left(p_{kj}^{\mathrm{t}}-p_{lj}^{\mathrm{p}}\right)^2\right) \tag{17}$$

where p_{kj}^t represents the jth parameter in the kth simulation in the training database, and p_{ij}^p represents the jth parameter in the lth setting of the parameters, which are required to predict the performance.

$$\begin{bmatrix}\tilde{\mathbf{c}}_i^{\mathrm{t}}\\ \tilde{\mathbf{c}}_i^{\mathrm{p}}\end{bmatrix}\sim\mathrm{N}\left[\begin{matrix}0\\0\end{matrix},\begin{matrix}\sum^{\tilde{\mathbf{c}}_i}+\sigma_i\mathbf{I} & \sum_*^{\tilde{\mathbf{c}}_i}\\ \left(\sum_*^{\tilde{\mathbf{c}}_i}\right)^{\mathrm{T}} & \sum_{**}^{\tilde{\mathbf{c}}_i}\end{matrix}\right] \tag{18}$$

where $\sum_*^{\tilde{\mathbf{c}}_i}$ is the covariance matrix of the training and prediction data, in which the columns represent the training parameters, and the rows are the settings that are supposed to be predicted. $\sigma_i\mathbf{I}_r$ represents the $r \times r$ noise matrix.

The prediction of the principal components should be:

$$\tilde{\mathbf{c}}_i^{\mathrm{p}}=\left(\sum\nolimits_*^{\tilde{\mathbf{c}}_i}\right)^{\mathrm{T}}\left(\sum^{\tilde{\mathbf{c}}_i}+\sigma_i\mathbf{I}\right)^{-1}\tilde{c}_i^{\mathrm{t}} \tag{19}$$

Then, the principal components can be transformed back to the standardized wind speed using the following equation:

$$\mathrm{w}_{\mathrm{em}}^{\mathrm{p}}=\tilde{\mathbf{c}}^{\mathrm{p}}\,\widehat{V}^{\mathrm{T}} \tag{20}$$

where $\tilde{\mathbf{c}}^p$ consists of n_t predicted principal components:

$$\tilde{c}^p=\left(c_1^p,c_2^p,\ldots,c_{n_t}^p\right) \tag{21}$$

Next, the wind speed at location j can be estimated by:

$$W_{\mathrm{prediction}}(\,:,j)=w_{\mathrm{em}}^{\mathrm{p}}(\,:,j)w_{\mathrm{std},j}+w_{\mathrm{avg},j} \tag{22}$$

5. Emulator evaluation

The accuracy of the proposed emulator is evaluated qualitatively and quantitatively in the following subsections. The training data set (150 CFD simulations) and an evaluation dataset (nine CFD simulations) are subjected to qualitative and quantitative analyses. The evaluation dataset has lift-up designs that are completely different from the training dataset as shown in Fig. 8 and Table A1 in Appendix A. The qualitative analysis compares the velocity fields near the lift-up buildings obtained from the CFD simulations and as predicted by the emulator to identify any visible differences between the two wind fields. The quantitative analysis estimates the similarity between the two velocity fields by calculating statistical indices and validation metrics.

5.1. Qualitative evaluation

Fig. 10 shows four cases of the wind field near a lift-up building, two of which are modeled by CFD simulations, and the other two are predicted by the GP-based emulator: T1 and T2 stem from the training data of the emulator; E1 and E2 are from the independent dataset. These wind fields are carefully selected to represent the influence of the lift-up designs on the surroundings as the selected buildings have different building dimensions and center core designs. To identify the visible differences in the wind fields, the standard deviation of the emulator (3rd row) and absolute difference in wind speeds of the CFD simulations and the emulator (4th row) are shown in Fig. 10. The visual inspection suggests that the emulator captures all prominent flow features such as upstream and downstream low wind speed zones, corner streams as well as wind direction near the lift-up building and predicts wind fields that are similar to those of the CFD simulations. The standard deviation of the emulator (Emulator (std)) is considerably small (<0.5 m/s) over a large portion of the interrogated area even though some large discrepancies between the emulator and CFD predictions are observed locally in the evaluation cases. It should be noted that the absolute difference between the emulator and CFD predictions coincides with Emulator (std), thus the latter is a reliable indicator for estimating the uncertainty in the emulator predictions. It is noteworthy that the emulator on a typical desktop computer (Intel® Core™ i7-6700k CPU @ 4.00 GHz with 32 GB RAM and Windows10 64-bit operating system) requires about 0.02 s to predict the wind field near a lift-up building, while its CFD counterpart entails six CPU hours on the same desktop computer. Therefore, the emulator is faster by a factor of 10^7 than the CFD simulations.

5.2. Statistical analysis

Fig. 11 shows the comparison of longitudinal and lateral wind speeds (u and v) of the CFD simulation and the emulator for the training and evaluation datasets. In each case, u and v are measured in CFD simulation and predicted in the emulator at the same locations near the lift-up buildings and are paired up for comparison. The distribution of data points of the two datasets is approximately symmetrical about the diagonal line, showing a perfect match between the CFD-simulated and emulator-predicted wind speeds. The symmetric distribution of the data points indicates the absence of systematic error in the emulator's predictions in both the training and evaluation datasets. Moreover, the similar distributions of data clusters in the two datasets prove that the emulator has successfully learned from the training dataset to predict wind speeds in the evaluation cases. The accuracy of the emulator is more than 91% of the training and 89% of the evaluation data, within a factor of 2 of the corresponding CFD simulation data.

5.3. Validation metrics

The similarity between the emulator predictions and CFD simulation, i.e., the correlation between the two sets of data and degree of deviation between the emulator and CFD simulation results are estimated using validation metrics proposed by COST732 [77]. Four indices, namely, Factor of 2 of observations (FAC2), Normalized mean square error (NMSE), Fractional bias (FB), and Pearson correlation coefficient (R) are calculated as per Eqs. 23–26:

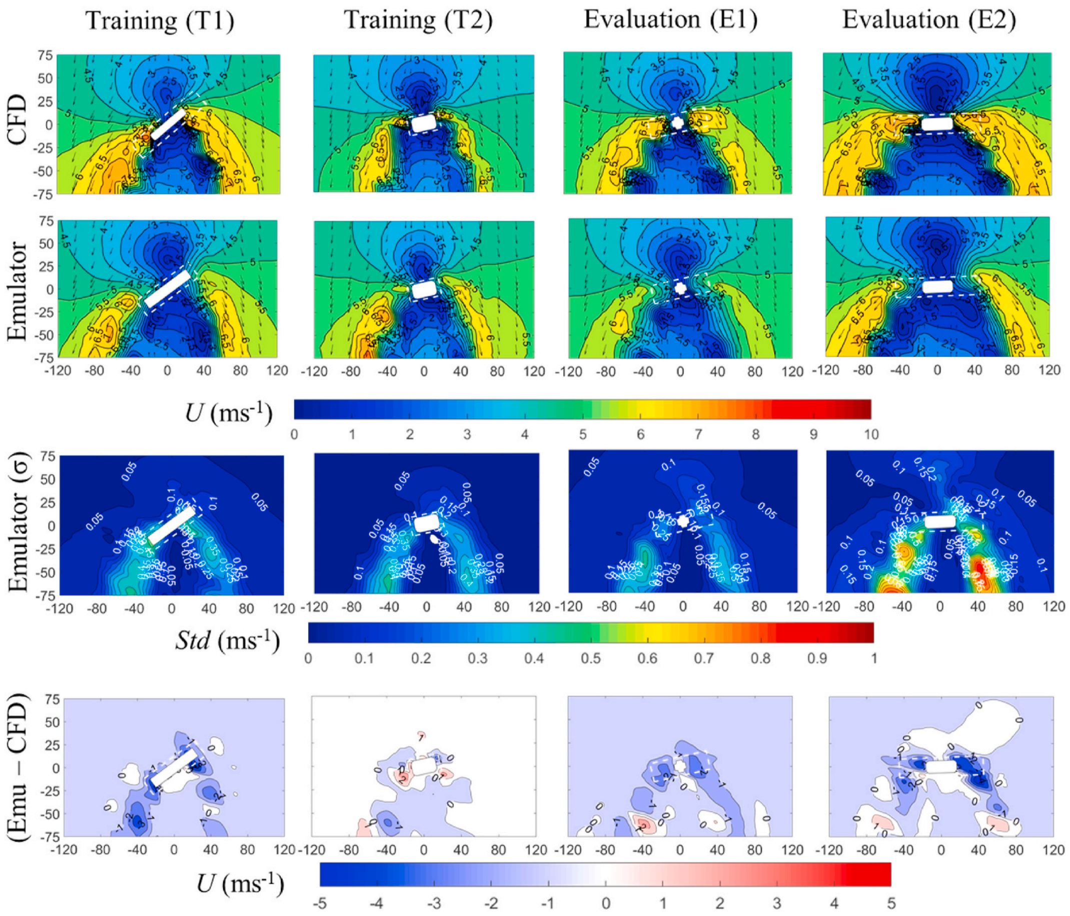


Fig. 10. Comparison of pedestrian-level wind fields modeled by CFD simulation (1st row), predicted by the emulator (2nd row), prediction uncertainty of the emulator (3rd row), and absolute difference in wind speeds of CFD simulation and the emulator (4th row).

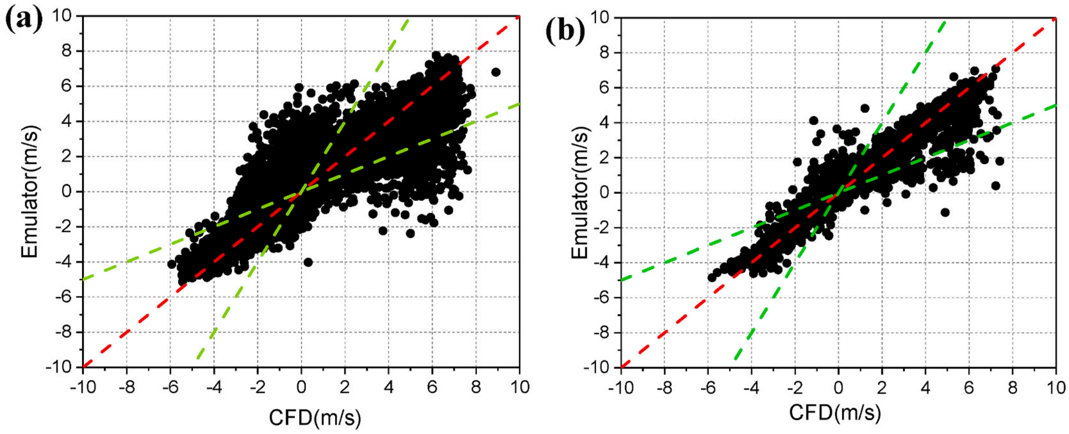


Fig. 11. Comparison of wind speeds (u and v) by the CFD simulation and the emulator for (a) training dataset, and (b) evaluation dataset.

$$FAC2 = \frac{N}{n} = \frac{1}{n}\sum_{i=1}^{n} N_i \quad with\ N_i = \begin{cases} 1\ for\ 0.5 \le \dfrac{CFD_i}{Emu_i} \le 2 \\ 1\ for\ Emu_i \le W\ and\ CFD_i \le W \\ 0\ else \end{cases} \tag{23}$$

where N_i is the number of data points that satisfy the specified criteria, n is the total number of data points, and W is a threshold wind speed.

$$NMSE = \frac{\langle (Emu - CFD)^2 \rangle}{\langle Emu \rangle \langle CFD \rangle} \tag{24}$$

Here, the angular bracket denotes the average over all data points.

$$FB = \frac{\langle Emu \rangle - \langle CFD \rangle}{0.5(\langle Emu \rangle + \langle CFD \rangle)} \tag{25}$$

$$R = \frac{\sum_{i=1}^{n} (CFD_i - \langle CFD \rangle)(Emu_i - \langle Emu \rangle)}{\sqrt{(CFD_i - \langle CFD \rangle)^2}\sqrt{(Emu_i - \langle Emu \rangle)^2}} \tag{26}$$

The four indices serve different purposes in the model evaluation. For example, FAC2 evaluates the overall emulator performance, while NMSE estimates the average discrepancy between CFD simulation and emulator. FB is an indicator of systematic bias between modeled and predicted wind speeds, and R measures the degree of correlation between the wind speeds from the CFD simulations and the emulator. In addition to these indications, threshold values are defined for the four indices to estimate whether the emulator performance is acceptable or not. This study adopts the following thresholds: FAC2 > 0.5, $-0.3 <$ FB < 0.3, NMSE <1.5, and R > 0.8 as recommended by Chang and Hanna [104], Goricsán et al. [105] and Moonen and Allegrini [64].

Table 2 shows the excellent performance of the emulator in predicting the mean wind speeds in both the training and evaluation datasets. For instance, the validation metrics FAC2 and R are close to 1, and NMSE and FB are close to 0, indicating great reproductivity of the emulator. The validation metrics of the evaluation dataset are inferior compared to the training dataset but still confirm the competitiveness of the emulator in predicting pedestrian-level mean wind speeds near lift-up buildings. For example, more than 89% of the emulator's prediction is within a factor of 2 of the simulation results with a high correlation of 0.969. The emulator has no bias towards over- or under-estimating wind speeds, as indicated by the near-zero FB and NMSE values. Based on the selected acceptance criteria of the validation metrics, it can be concluded that the proposed GP-based emulator is an excellent substitute for CFD simulation in modeling PLWEs.

6. Sensitivity analysis

The sensitivity analysis investigates how two subjective decisions – the number of principal components, and the size of the training dataset – made during emulator development impact the accuracy of the emulator's predictions. It is noteworthy that, except for the number of principal components and the size of the training dataset, all calculations and calibration of the model parameters in the sensitivity analysis are similar to those described in Section 5.

6.1. Number of principal components

The data reduction using PCA entails a truncation error and induces uncertainty in the emulator predictions. It is prudent to assume that large data reduction results in a computationally efficient emulator but causes high uncertainty in the emulator's predictions. However, it fails to determine the tradeoff point where sufficiently large data reduction can be achieved without compromising the accuracy of the emulator.

Table 3 shows the validation metrics calculated for the emulator, which is developed using five different values of n_t: 3, 5, 9, 30, and 50 in addition to the initial selection of $n_t = 25$. The validation metrics show a steady improvement of prediction accuracy for both the training and evaluation datasets as n_t increases from 3 to 25, indicating high accuracy of the emulator's prediction for those with a sufficiently large number of n_t. This trend is plausible as the number of n_t defines the set of basic functions that represent the output space. If the number of basic functions is low, then the emulator cannot reproduce the simulator's responses, resulting in considerable deviations in validation metrics from their ideal values such as in the case $n_t = 3$. However, increasing n_t enlarges the size of the dataset handled by the emulator, and the number of hyper-parameters of the emulator. Moreover, the increase of n_t does not significantly improve the emulator's performance, as can be seen

Table 2
Validation metrics calculated for training and evaluation datasets for the emulator with settings ($n_t = 25$, $r = 50$).

	Training dataset				Evaluation dataset			
Metric	FAC2	NMSE	FB	R	FAC2	NMSE	FB	R
Aim	1	0	0	1	1	0	0	1
Range	>0.5	<1.5	[-0.3,0.3]	>0.8	>0.5	<1.5	[-0.3,0.3]	>0.8
Calculated	0.9104	0.0582	0.0186	0.9751	0.8910	0.0781	0.0600	0.9690

Table 3
Validation metrics calculated for the training and evaluation datasets for the emulator with settings $r = 50$ and various numbers of n_t.

	Training dataset				Evaluation dataset			
Metric	FAC2	NMSE	FB	R	FAC2	NMSE	FB	R
Aim	1	0	0	1	1	0	0	1
Range	>0.5	<1.5	[-0.3,0.3]	>0.8	>0.5	<1.5	[-0.3,0.3]	>0.8
$n_t = 3$	0.8916	0.0865	0.0180	0.9627	0.8807	0.0917	0.0672	0.9637
$n_t = 5$	0.9011	0.0696	0.0223	0.9703	0.8861	0.0871	0.0618	0.9652
$n_t = 9$	0.9064	0.0625	0.0198	0.9733	0.8874	0.0795	0.0615	0.9685
$n_t = 25$	0.9104	0.0582	0.0186	0.9751	0.8910	0.0781	0.0600	0.9690
$n_t = 30$	0.9110	0.0578	0.0185	0.9753	0.8933	0.0785	0.0598	0.9688
$n_t = 50$	0.9110	0.0576	0.0186	0.9753	0.8920	0.0785	0.0598	0.9688

from the approximately similar validation metrics between the cases with $n_t = 25$, 30, and 50. By considering the accuracy and computational cost, this study chose $n_t = 25$ for the emulator developed in this study.

6.2. *Number of training datasets*

Table 4 shows the calculated validation metrics for the emulator developed using various sizes of datasets. It should be noted that each dataset is subjected to 98% data reduction via Principal Component Analysis, and all other emulator parameters are the same as the original emulator. The validation metrics in Table 4 show improvement as the size of the dataset increases, and this improvement is more noticeable than that obtained from increasing p_y in Section 6.1. The comparison of validation metrics in Tables 3 and 4 indicates that the accuracy of the emulator's predictions strongly depends on the size of the training dataset, rather than on the degree of data reduction. If the training dataset is too small such as in the case $r = 3$, it cannot map the variations in the simulator outputs, resulting in significant deterioration of the accuracy of the emulator's prediction. With the increase of the size of the dataset, the emulator improves the prediction accuracy in both the training and evaluation data sets, because more data points warrant capturing precisely the underlying trends in input and output spaces precisely. However, this gain is capped by the computational cost of running additional simulations. In the current study, each CFD simulation requires six CPU hours on a typical desktop computer. Given the computational cost and the accuracy of the emulator beyond the size of the dataset larger than 50 ($r > 50$), the current study limits the size of the training data set to $r = 50$, beyond which the rate of improvement of the validation metrics slows down considerably.

7. Practical significance

The proposed GP-based emulator can be used for practical purposes in various branches of engineering. The followings are three examples particularly related to the applications in environmental wind engineering.

- The emulator can be employed to predict the micro-wind climate in fully-developed urban areas based on the meteorological wind speed data from the nearest anemometer station. Based on the magnitude and direction of wind velocities, the emulator can quickly and accurately predict wind speeds at hundreds of locations in the urban area. The predicted wind speed data can be used for issuing safety warnings of the occurrence of high wind speeds, operating safety measures such as wind barriers at particular locations, and managing building-mounted wind turbines in the area [64]. have demonstrated a similar application of a GP-based emulator for predicting wind speeds in an idealized urban canyon.
- Because the emulator is a fully computerized program, it can be easily integrated into other computer programs and computer-based processes. Optimization is such a process, in which the emulator can serve as a surrogate model to predict the values of objective functions in each iteration. The emulator can also perform calculations based on the input data received from another computer program. For example, the proposed GP-based emulator can be used to predict the thermal comfort of pedestrians passing by lift-up buildings based on the data from the online calculator of the Universal Thermal Climate Index (http://www.utci.org/utcineu/utcineu.php).
- The emulator's ability to handle many input and output parameters comes in handy in managing the wind environment in a site. Such a tool can predict the areas with wind discomfort, wind danger, outdoor thermal discomfort due to lack of wind circulation in the site based on daily and extreme meteorological data (e.g. 10-min wind speed and typhoon track data), respectively. Besides, it can be used to manage window opening schedules of buildings and predict concentration levels of stack emissions on the site.

8. Discussion

The current study demonstrates the development of a computationally economical GP-based emulator for assessing the PLWE near an isolated building with an unconventional configuration. Although the GP-based emulator shows promise in modeling the PLWE, the following facets should be considered during its development.

- The developed emulator is indeed quicker by a factor of 10^7 in predicting the PLWE compared with its CFD simulation counterpart. With the time required for the one-time emulator calibration, the speedup factor decreases to 10^2, but the emulator is still more than 400 times faster than the CFD simulation.

Table 4
Validation metrics calculated for the training and evaluation datasets for the emulator with various sizes of datasets.

	Training dataset				Evaluation dataset			
Metric	FAC2	NMSE	FB	R	FAC2	NMSE	FB	R
Aim	1	0	0	1	1	0	0	1
Range	>0.5	<1.5	[-0.3,0.3]	>0.8	>0.5	<1.5	[-0.3,0.3]	>0.8
$r = 5$	0.8110	0.1780	0.0046	0.9242	0.7912	0.2390	0.1003	0.9038
$r = 10$	0.8726	0.1067	0.0027	0.9542	0.8428	0.1525	0.0821	0.9380
$r = 15$	0.8875	0.0981	0.0020	0.9578	0.8678	0.1327	0.0591	0.9448
$r = 50$	0.9105	0.0599	0.0018	0.9744	0.8920	0.0837	0.0599	0.9666
$r = 100$	0.9312	0.0378	0.0012	0.9838	0.8984	0.0703	0.0531	0.9718
$r = 150$	0.9447	0.0196	0.0011	0.9917	0.8995	0.0693	0.0512	0.9722

- The development of the emulator requires a substantial number of hours as its training and testing datasets are based on dozens or even hundreds of simulations. Therefore, the GP-based emulator is particularly beneficial for the cases, which require many simulations with similar input conditions. For instance, a GP-based emulator can be efficient in evaluating the PLWE in public housing estates in Hong Kong, where the residential high-rise buildings have standard building designs [92]. The emulator is also beneficial for assessing indoor and outdoor wind circulation near prefabricated or modular buildings, of which the designs have slight differences. Another strategy is to use a database based on existing experimental and numerical simulation data for emulator development as described by Ref. [36,38].
- The emulator's performance and accuracy can be significantly affected by the subjective decisions made in the selection of inputs, outputs, the size of training datasets, and the number of principal components as well as techniques used for data reduction and estimating hyper-parameters. Although this study has attempted to minimize the influence of subjective decisions by conducting a sensitivity analysis on the emulator predictions, another set of parameters could develop an emulator with considerably different performance and accuracy to the one proposed in this study.
- Similar to many other surrogate models, the proposed GP-based emulator has some limitations due to its kernel function, inefficiency in performing calculations in high-dimensional spaces, inability to perform the non-parametric calculation. For example Alamaniotis et al. [3], tested four kernel functions: the Matérn, Neural Net, Gaussian, and Linear kernels for a GP-based regression model and found some dependencies of the output on the selected kernel function. Therefore, the results of the proposed GP-based emulator may depend on its kernel function – the squared exponential kernel. To minimize such a dependency, it is prudent to test a few other kernel functions such as rational quadratic, and Periodic kernels for the emulator. It is a well-observed fact that the Gaussian process results in high computational cost in handling large datasets [7]. This shortcoming can be minimized using the sparse GP method, which reduces the size of the training dataset by intelligently selecting a subset of the training dataset – the active set – for the calculation [91]. Because the Gaussian process is not non-parametric, it requires loading the whole training dataset to manipulate the kernel function before each calculation. This is a drawback of the Gaussian process compared to ANN, which can store all parameters in it after the first calculation.

9. Concluding remarks

This study transcends the knowledge of capacity, performance, and accuracy of GP-based emulators that can be employed for typical wind engineering applications. The knowledge is gained through developing a GP-based emulator to predict complex flow features near an isolated building with an unconventional building configuration. The development process indicates that the GP-based emulator can handle many input parameters, large datasets, and the possibility of data reduction using PCA for efficient computation. Furthermore, a novel technique based on optimization has been proposed to estimate 250 hyper-parameters of the GP-based emulator. The proposed technique is advantageous because it does not require prior knowledge of the probability distributions and lower and upper bounds of the individual hyper-parameters and is less vulnerable to any subjective decisions made on selecting the lower and upper bounds of the hyper-parameters. The accuracy of the emulator has been estimated qualitatively as well as quantitatively with respect to CFD simulations. The performance evaluation reveals good agreement between the emulator's results and the CFD simulations, and the former shows no systematic bias, great reproductivity, and high prediction accuracy. The emulator is quicker by a factor of 10^7 in mimicking the PLWE near an isolated building, compared with its counterpart, which needs more than six CPU hours on a typical desktop computer (Intel® Core™ i7-6700k CPU @ 4.00 GHz with 32 GB RAM and Windows10 64-bit operating system).

Although the performance of the proposed GP-based emulator is promising, it is prudent to further explore its compatibility in model coupling, for example, as a surrogate model in optimization (see Ref. [27,28,93]) and compare its accuracy with emulators developed based on other techniques such as Artificial Neural Network (ANN), Polynomial Chaos Expansion, or Support Vector Regression (SVR).

Declaration of competing interest

The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Acknowledgment

The work described in this paper was partially supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (HKU C7064-18G) and Guangdong Province Key Laboratory for Climate Change and Natural Disaster Studies (Grant 2020B1212060025).

APPENDIX A

Table A.1
Dimensions of nine lift-up buildings of the evaluation dataset.

Building	H (m)	W (m)	h (m)	w (m)	d (m)	v_1	v_2	θ (deg)
1	86.934	89.399	8.483	45.667	8.497	0.252	−0.167	40.454
2	109.178	62.345	3.625	62.099	11.022	0.013	−0.119	3.182
3	71.002	72.445	7.473	27.865	6.786	0.026	−0.228	10.910
4	78.517	90.000	7.425	33.158	13.295	0.126	−0.382	2.272
5	94.749	57.896	6.583	12.954	8.918	0.146	−0.153	40.455
6	69.800	75.812	4.792	69.474	18.681	0.239	−0.125	8.636
7	112.485	79.178	3.806	16.660	15.455	0.263	−0.162	4.091
8	56.423	61.623	8.759	12.578	13.267	0.262	−0.786	12.727
9	66.042	57.415	4.443	19.849	19.214	0.229	−0.322	30.455

References

[1] T. Abbas, I. Kavrakov, G. Morgenthal, T. Lahmer, Prediction of aeroelastic response of bridge decks using artificial neural networks, Comput. Struct. 231 (2020) 106198.

[2] J. Aernouts, I. Couckuyt, K. Crombecq, J.J. Dirckx, Elastic characterization of membranes with a complex shape using point indentation measurements and inverse modelling, Int. J. Eng. Sci. 48 (6) (2010) 599–611.

[3] M. Alamaniotis, S. Chatzidakis, L.H. Tsoukalas, Monthly Load Forecasting Using Kernel Based Gaussian Process Regression, *MedPower* 2014, Athens, 2014, pp. 1–8, https://doi.org/10.1049/cp.2014.1693.

[4] A.C. Antoulas, D.C. Sorensen, Approximation of large-scale dynamical systems: an overview, Int. J. Appl. Math. Comput. Sci. 11 (5) (2001) 1093–1121.

[5] F. Antunes, M. Amorim, F.C. Pereira, B. Ribeiro, Active learning metamodeling for policy analysis: application to an emergency medical service simulator, Simulat. Model. Pract. Theor. 97 (2019) 101947.

[6] P.G. Asteris, K.G. Kolovos, Self-compacting concrete strength prediction using surrogate models, Neural Comput. Appl. 31 (1) (2019) 409–424.

[7] M. Bauer, M. van der Wilk, C.E. Rasmussen, Understanding probabilistic sparse Gaussian process approximations, Adv. Neural Inform. Process. Syst. (2016) 1533–1541.

[8] L.T. Biegler, Y.D. Lang, W. Lin, Multi-scale optimization for process systems engineering, Comput. Chem. Eng. 60 (2014) 17–30.

[9] H.P. Bieker, O. Slupphaug, T.A. Johansen, Real-time production optimization of oil and gas production systems: a technology survey, SPE Prod. Oper. 22 (4) (2007) 382–391.

[10] G.T. Bitsuamlak, C. Bédard, T. Stathopoulos, Modeling the effect of topography on wind flow using a combined numerical–neural network approach, J. Comput. Civ. Eng. 21 (6) (2007) 384–392.

[11] G. Blatman, B. Sudret, An adaptive algorithm to build up sparse polynomial chaos expansions for stochastic finite element analysis, Probabilist. Eng. Mech. 25 (2) (2010) 183–197.

[12] B. Blocken, LES over RANS in building simulation for outdoor and indoor applications: a foregone conclusion? Build. Simulation 11 (5) (2018) 821–870. Tsinghua University Press.

[13] M. Bottema, Towards rules of thumb for wind comfort and air quality, Atmos. Environ. 33 (24–25) (1999) 4009–4017.

[14] G.E. Box, G.C. Tiao, Bayesian Inference in Statistical Analysis, 40, John Wiley & Sons, 2011.

[15] F. Bre, J.M. Gimenez, V.D. Fachinotti, Prediction of wind pressure coefficients on building surfaces using artificial neural networks, Energy Build. 158 (2018) 1429–1441.

[16] L. Cappelli, G. Balokas, M. Montemurro, F. Dau, L. Guillaumat, Multi-scale identification of the elastic properties variability for composite materials through a hybrid optimisation strategy, Compos. B Eng. 176 (2019) 107193.

[17] A. Castelletti, S. Galelli, M. Restelli, R. Soncini-Sessa, Data-driven dynamic emulation modelling for the optimal management of environmental systems, Environ. Model. Software 34 (2012) 30–43.

[18] T. Cebeci, P. Bradshaw, Momentum Transfer in Boundary Layers, Hemisphere Publishing, Washington, DC, 1977.

[19] K. Chen, J. Yu, Short-term wind speed prediction using an unscented Kalman filter based state-space support vector regression approach, Appl. Energy 113 (2014) 690–705.

[20] N. Chen, Z. Qian, I.T. Nabney, X. Meng, Wind power forecasts using Gaussian processes and numerical weather prediction, IEEE Trans. Power Syst. 29 (2) (2013) 656–665.

[21] X. Chen, H. Yang, K. Sun, Developing a meta-model for sensitivity analyses and prediction of building performance for passively designed high-rise residential buildings, Appl. Energy 194 (2017) 422–439.

[22] Y. Chen, G.A. Kopp, D. Surry, Prediction of pressure coefficients on roofs of low buildings using artificial neural networks, J. Wind Eng. Ind. Aerod. 91 (3) (2003) 423–441.

[23] L. Csató, M. Opper, Sparse on-line Gaussian processes, Neural Comput. 14 (3) (2002) 641–668.

[24] H. Dongmei, H. Shiqing, H. Xuhui, Z. Xue, Prediction of wind loads on high-rise building using a BP neural network combined with POD, J. Wind Eng. Ind. Aerod. 170 (2017) 1–17.

[25] W. Du, N. Xue, W. Shyy, J.R. Martins, A surrogate-based multi-scale model for mass transport and electrochemical kinetics in lithium-ion battery electrodes, J. Electrochem. Soc. 161 (8) (2014) E3086–E3096.

[26] V. Dubourg, B. Sudret, Meta-model-based importance sampling for reliability sensitivity analysis, Struct. Saf. 49 (2014) 27–36.

[27] A. Elshaer, G. Bitsuamlak, Multiobjective aerodynamic optimization of tall building openings for wind-induced load reduction, J. Struct. Eng. 144 (10) (2018), 04018198.

[28] A. Elshaer, G. Bitsuamlak, A. El Damatty, Enhancing wind performance of tall buildings using corner aerodynamic optimization, Eng. Struct. 136 (2017) 133–148.

[29] E.C. English, F.R. Fricke, The interference index and its prediction using a neural network analysis of wind-tunnel data, J. Wind Eng. Ind. Aerod. 83 (1–3) (1999) 567–575.

[30] J. Franke, A. Hellsten, K.H. Schlunzen, B. Carissimo, The COST 732 Best Practice Guideline for CFD simulation of flows in the urban environment: a summary, Int. J. Environ. Pollut. 44 (1–4) (2011) 419–427.

[31] C. García-Sánchez, D.A. Philips, C. Gorlé, Quantifying inflow uncertainties for CFD simulations of the flow in downtown Oklahoma city, Build. Environ. 78 (2014) 118–129.

[32] A. Giaralis, F. Petrini, Wind-induced vibration mitigation in tall buildings using the tuned mass-damper-inerter, J. Struct. Eng. 143 (9) (2017), 04017127.

[33] J.M. Gimenez, F. Bre, Optimization of RANS turbulence models using genetic algorithms to improve the prediction of wind pressure coefficients on low-rise buildings, J. Wind Eng. Ind. Aerod. 193 (2019) 103978.

[34] D. Ginsbourger, B. Rosspopoff, G. Pirot, N. Durrande, P. Renard, Distance-based kriging relying on proxy simulations for inverse conditioning, Adv. Water Resour. 52 (2013) 275–291.

[35] C. Gorlé, C. Garcia-Sanchez, G. Iaccarino, Quantifying inflow and RANS turbulence model form uncertainties for wind engineering flows, J. Wind Eng. Ind. Aerod. 144 (2015) 202–212.

[36] G. Hu, K.C.S. Kwok, Predicting wind pressures around circular cylinders using machine learning techniques, J. Wind Eng. Ind. Aerod. 198 (2020) 104099.

[37] G. Hu, S. Hassanli, K.C. Kwok, K.T. Tse, Wind-induced responses of a tall building with a double-skin façade system, J. Wind Eng. Ind. Aerod. 168 (2017) 91–100.

[38] G. Hu, L. Liu, D. Tao, J. Song, K.T. Tse, K.C.S. Kwok, Deep learning-based investigation of wind pressures on tall building under interference effects, J. Wind Eng. Ind. Aerod. 201 (2020) 104138.

[39] J. Hu, J. Wang, Short-term wind speed prediction using empirical wavelet transform and Gaussian process regression, Energy 93 (2015) 1456–1466.

[40] S. Huang, R. Li, Q.S. Li, Numerical simulation on fluid-structure interaction of wind around super-tall building at high Reynolds number conditions, Struct. Eng. Mech. 46 (2) (2013) 197–212.

[41] M. Jamil, M. Zeeshan, A comparative analysis of ANN and chaotic approach-based wind speed prediction in India, Neural Comput. Appl. 31 (10) (2019) 6807–6819.

[42] W.D. Janssen, B. Blocken, T. van Hooff, Pedestrian wind comfort around buildings: comparison of wind comfort criteria based on whole-flow field data for a complex case study, Build. Environ. 59 (2013) 547–562.

[43] S. Jung, J. Ghaboussi, S.D. Kwon, Estimation of aeroelastic parameters of bridge decks using neural networks, J. Eng. Mech. 130 (11) (2004) 1356–1364.

[44] A. Kalinina, M. Spada, D.F. Vetsch, S. Marelli, C. Whealton, P. Burgherr, B. Sudret, Metamodeling for uncertainty quantification of a flood wave model for concrete dam breaks, Energies 13 (14) (2020) 3685.

[45] C.A. Kang, A.R. Brandt, L.J. Durlofsky, A new carbon capture proxy model for optimizing the design and time-varying operation of a coal-natural gas power station, International Journal of Greenhouse Gas Control 48 (2016) 234–252.

[46] A.C. Khanduri, C. Bédard, T. Stathopoulos, Modelling wind-induced interference effects using backpropagation neural networks, J. Wind Eng. Ind. Aerod. 72 (1997) 71–79.

[47] A. Khosravi, R.N.N. Koury, L. Machado, J.J.G. Pabon, Prediction of wind speed and wind direction using artificial neural network, support vector regression and adaptive neuro-fuzzy inference system, Sustainable Energy Technologies and Assessments 25 (2018) 146–160.

[48] J.P. Kleijnen, Kriging metamodeling in simulation: a review, Eur. J. Oper. Res. 192 (3) (2009) 707–716.

[49] U. Konda, T. Singh, P. Singla, P. Scott, Uncertainty propagation in puff-based dispersion models using polynomial chaos, Environ. Model. Software 25 (12) (2010) 1608–1618.

[50] B.E. Launder, D.B. Spalding, The numerical computation of turbulent flows, in: S. V. Patankar, A. Pollard, A.K. Singhal, S.P. Vanka (Eds.), Numerical Prediction of Flow, Heat Transfer, Turbulence and Combustion, Pergamon Press, New York, 1983, pp. 96–116.

[51] T.V. Lawson, A.D. Penwarden, The effects of wind on people in the vicinity of buildings, in: Proceedings 4th International Conference on Wind Effects on Buildings and Structures, Cambridge University Press, Heathrow, 1975, pp. 605–622.

[52] L. Leifsson, S. Koziel, Multi-fidelity design optimization of transonic airfoils using physics-based surrogate modeling and shape-preserving response prediction, Journal of Computational Science 1 (2) (2010) 98–106.

[53] M. Li, G. Jia, R.Q. Wang, Surrogate Modeling for Sensitivity Analysis of Models with High-Dimensional Outputs, 13th International Conference on Applications of Statistics and Probability in Civil Engineering, ICASP13, Seoul, South Korea, 2019. May 26-30, 2019.

[54] N. Linde, D. Ginsbourger, J. Irving, F. Nobile, A. Doucet, On uncertainty quantification in hydrogeology and hydrogeophysics, Adv. Water Resour. 110 (2017) 166–181.

[55] D. Liu, D. Niu, H. Wang, L. Fan, Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm, Renew. Energy 62 (2014) 592–597.

[56] X. Ma, F. Xu, B. Chen, Interpolation of wind pressures using Gaussian process regression, J. Wind Eng. Ind. Aerod. 188 (2019) 30–42.

[57] B. Manobel, F. Sehnke, J.A. Lazzús, I. Salfate, M. Felder, S. Montecinos, Wind turbine power curve modeling based on Gaussian processes and artificial neural networks, Renew. Energy 125 (2018) 1015–1020.

[58] J.D. Martin, T.W. Simpson, Use of kriging models to approximate deterministic computer models, AIAA J. 43 (4) (2005) 853–863.

[59] M. McIntire, D. Ratner, S. Ermon, Sparse Gaussian processes for bayesian optimization, UAI (2016). https://www-cs.stanford.edu/~ermon/papers/sparse-gp-uai.pdf.

[60] J. Meirlaen, B. Huyghebaert, F. Sforzi, L. Benedetti, P. Vanrolleghem, Fast, simultaneous simulation of the integrated urban wastewater system using mechanistic surrogate models, Water Sci. Technol. 43 (7) (2001) 301–309.

[61] T. Mengistu, W. Ghaly, Aerodynamic optimization of turbomachinery blades using evolutionary methods and ANN-based surrogate models, Optim. Eng. 9 (3) (2008) 239–255.
[62] B.Y. Mirghani, E.M. Zechman, R.S. Ranjithan, G. Mahinthakumar, Enhanced simulation-optimization approach using surrogate modeling for solving inverse problems, Environ. Forensics 13 (4) (2012) 348–363.
[63] H. Mittal, A. Sharma, A. Gairola, A review on the study of urban wind at the pedestrian level around buildings, Journal of Building Engineering 18 (2018) 154–163.
[64] P. Moonen, J. Allegrini, Employing statistical model emulation as a surrogate for CFD, Environ. Model. Software 72 (2015) 77–91.
[65] J. Murillo-Escobar, J.P. Sepulveda-Suescun, M.A. Correa, D. Orrego-Metaute, Forecasting concentrations of air pollutants using support vector regression improved with particle swarm optimization: case study in Aburrá Valley, Colombia, Urban Climate 29 (2019) 100473.
[66] B.M. Negash, L.D. Tufa, M. Ramasamy, M.B. Awang, System identification based proxy model of a reservoir under water injection, Model. Simulat. Eng. 2017 (2017), https://doi.org/10.1155/2017/7645470.
[67] T.J. Nikose, R.S. Sonparote, Dynamic wind response of tall buildings using artificial neural network, Struct. Des. Tall Special Build. 28 (13) (2019) e1657.
[68] A. O'Hagan, Bayesian analysis of computer code outputs: a tutorial, Reliab. Eng. Syst. Saf. 91 (10–11) (2006) 1290–1300.
[69] B.K. Oh, B. Glisic, Y. Kim, H.S. Park, Convolutional neural network-based wind-induced response estimation model for tall buildings, Comput. Aided Civ. Infrastruct. Eng. 34 (10) (2019) 843–858.
[70] S. Osowski, K. Garanty, Forecasting of the daily meteorological pollution using wavelets and support vector machine, Eng. Appl. Artif. Intell. 20 (6) (2007) 745–755.
[71] A.M. Overstall, D.C. Woods, Multivariate emulation of computer simulators: model selection and diagnostics with application to a humanitarian relief model, J. Roy. Stat. Soc. C Appl. Stat. 65 (4) (2016) 483.
[72] J.E. Pacheco, C.H. Amon, S. Finger, Bayesian surrogates applied to conceptual stages of the engineering design process, J. Mech. Des. 125 (4) (2003) 664–672.
[73] S. Razavi, B.A. Tolson, D.H. Burn, Review of surrogate modeling in water resources, Water Resour. Res. 48 (7) (2012).
[74] K. Redouane, N. Zeraibi, M.N. Amar, Adaptive surrogate modeling with evolutionary algorithm for well placement optimization in fractured reservoirs, Appl. Soft Comput. 80 (2019) 177–191.
[75] R. Rikards, H. Abramovich, J. Auzins, A. Korjakins, O. Ozolinsh, K. Kalnins, T. Green, Surrogate models for optimum design of stiffened composite shells, Compos. Struct. 63 (2) (2004) 243–251.
[76] F. Rizzo, L. Caracoglia, Artificial Neural Network model to predict the flutter velocity of suspension bridges, Comput. Struct. 233 (2020) 106236.
[77] M. Schatzmann, H. Olesen, J. Franke (Eds.), COST 732 Model Evaluation Case Studies: Approach and Results, Meteorological Institute, 2010.
[78] M. Sharifzadeh, A. Sikinioti-Lock, N. Shah, Machine-learning methods for integrated renewable power generation: a comparative study of artificial neural networks, support vector regression, and Gaussian process regression, Renew. Sustain. Energy Rev. 108 (2019) 513–538.
[79] O.B. Shukur, M.H. Lee, Daily wind speed forecasting through hybrid KF-ANN model based on ARIMA, Renew. Energy 76 (2015) 637–647.
[80] J. Sousa, C. García-Sánchez, C. Gorlé, Improving urban flow predictions through data assimilation, Build. Environ. 132 (2018) 282–290.
[81] J. Spiegelberg, J. Rusz, Can we use PCA to detect small signals in noisy data? Ultramicroscopy 172 (2017) 40–46.
[82] J. Sreekanth, B. Datta, Comparative evaluation of genetic programming and neural network as potential surrogate models for coastal aquifer management, Water Resour. Manag. 25 (13) (2011) 3201–3218.
[83] T. Stathopoulos, Introduction to wind engineering, wind structure, wind-building interaction, in: Wind Effects on Buildings and Design of Wind-Sensitive Structures, Springer, Vienna, 2007, pp. 1–30.
[84] T. Stathopoulos, B. Blocken, Pedestrian wind environment around tall buildings, in: Advanced Environmental Wind Engineering, Springer, Tokyo, 2016, pp. 101–127.
[85] X. Sun, J. Park, J.I. Choi, G.H. Rhee, Uncertainty quantification of upstream wind effects on single-sided ventilation in a building using generalized polynomial chaos method, Build. Environ. 125 (2017) 153–167.
[86] P.M. Tagade, B.M. Jeong, H.L. Choi, A Gaussian process emulator approach for rapid contaminant characterization with an integrated multizone-CFD model, Build. Environ. 70 (2013) 232–244.
[87] Y. Tominaga, A. Mochida, R. Yoshie, H. Kataoka, T. Nozu, M. Yoshikawa, T. Shirasawa, AIJ guidelines for practical applications of CFD to pedestrian wind environment around buildings, J. Wind Eng. Ind. Aerod. 96 (10–11) (2008) 1749–1761.
[88] K.T. Tse, X. Zhang, A.U. Weerasuriya, S.W. Li, K.C. Kwok, C.M. Mak, J. Niu, Adopting 'lift-up' building design to improve the surrounding pedestrian-level wind environment, Build. Environ. 117 (2017) 154–165.
[89] L. Uusitalo, A. Lehikoinen, I. Helle, K. Myrberg, An overview of methods to evaluate uncertainty of deterministic models in decision support, Environ. Model. Software 63 (2015) 24–31.
[90] L. Van Der Maaten, E. Postma, J. Van den Herik, Dimensionality reduction: a comparative, J. Mach. Learn. Res. 10 (66–71) (2009) 13.
[91] M. Van der Wilk, Sparse Gaussian Process Approximations and Applications, Doctoral dissertation, University of Cambridge, 2019.
[92] A.U. Weerasuriya, X. Zhang, V.J. Gan, Y. Tan, A holistic framework to utilize natural ventilation to optimize energy performance of residential high-rise buildings, Build. Environ. 153 (2019) 218–232.
[93] A.U. Weerasuriya, X. Zhang, B. Lu, K.T. Tse, C.H. Liu, Optimizing Lift-Up Design to Maximize Pedestrian Wind and Thermal Comfort in 'Hot-Calm'and 'Cold-Windy'Climates, Sustainable Cities and Society, 2020, p. 102146.
[94] T. Wu, A. Kareem, Modeling hysteretic nonlinear behavior of bridge aerodynamics via cellular automata nested neural network, J. Wind Eng. Ind. Aerod. 99 (4) (2011) 378–388.
[95] X. Xu, Q. Yang, A. Yoshida, Y. Tamura, Characteristics of pedestrian-level wind around super-tall buildings with various configurations, J. Wind Eng. Ind. Aerod. 166 (2017) 61–73.
[96] W. Yang, M. Deng, F. Xu, H. Wang, Prediction of hourly PM2. 5 using a space-time support vector regression model, Atmos. Environ. 181 (2018) 12–19.
[97] J.J. Yu, X.S. Qin, O. Larsen, Uncertainty analysis of flood inundation modelling using GLUE with surrogate models in stochastic sampling, Hydrol. Process. 29 (6) (2015) 1267–1279.
[98] J. Yu, K. Chen, J. Mori, M.M. Rashid, A Gaussian mixture copula model based localized Gaussian process regression approach for long-term wind speed prediction, Energy 61 (2013) 673–686.
[99] C. Zhang, H. Wei, X. Zhao, T. Liu, K. Zhang, A Gaussian process regression based hybrid approach for short-term wind speed prediction, Energy Convers. Manag. 126 (2016) 1084–1092.
[100] X. Zhang, K.T. Tse, A.U. Weerasuriya, K.C.S. Kwok, J. Niu, Z. Lin, C.M. Mak, Pedestrian-level wind conditions in the space underneath lift-up buildings, J. Wind Eng. Ind. Aerod. 179 (2018) 58–69.
[101] X. Zhang, K.T. Tse, A.U. Weerasuriya, S.W. Li, K.C. Kwok, C.M. Mak, Z. Lin, Evaluation of pedestrian wind comfort near 'lift-up' buildings with different aspect ratios and central core modifications, Build. Environ. 124 (2017) 245–257.
[102] X. Zhang, A.U. Weerasuriya, B. Lu, K.T. Tse, C.H. Liu, Y. Tamura, Pedestrian-level wind environment near a super-tall building with unconventional configurations in a regular urban area, Build. Simulat. (2019) 1–18. Tsinghua University Press.
[103] X. Zhang, A.U. Weerasuriya, X. Zhang, K.T. Tse, B. Lu, C.Y. Li, C.H. Liu, Pedestrian wind comfort near a super-tall building with various configurations in an urban-like setting, Build. Simulation (2020) 1, https://doi.org/10.1007/s12273-020-0658-6. Nature Publishing Group.
[104] J.C. Chang, S.R. Hanna, Technical descriptions and user's guide for the BOOT statistical model evaluation software package, version 2.0, George Mason University and Harvard School of Public Health, Fairfax, Virginia, USA, 2005.
[105] I. Goricsán, M. Balczó, M. Balogh, K. Czáder, A. Rákai, C. Tonkó, Simulation of flow in an idealised city using various CFD codes, Int. J. Environ. Pollut. 44 (1–4) (2011) 359–367.