sensors

MDPI

Article

Wind-Induced Pressure Prediction on Tall Buildings Using Generative Adversarial Imputation Network

Bubryur Kim [1,2], N. Yuvaraj [1,*], K. R. Sri Preethaa [3], Gang Hu [4] and Dong-Eun Lee [5,*]

[1] Department of Architectural Engineering, Dong-A University, Busan 49315, Korea; rlaqjqfuf@dau.ac.kr
[2] Department of ICT Integrated Ocean Smart Cities Engineering, Dong-A University, Busan 49315, Korea
[3] Department of Artificial Intelligence and Data Science, KPR Institute of Engineering and Technology, Coimbatore 641407, India; k.r.sripreethaa@kpriet.ac.in
[4] School of Civil and Environmental Engineering, Harbin Institute of Technology, Shenzhen 518055, China; hugang@hit.edu.cn
[5] School of Architecture, Civil, Environment and Energy Engineering, Kyungpook National University, 80, Daehak-ro, Buk-gu, Daegu 41566, Korea
* Correspondence: yuvaraj.n@kpriet.ac.in (N.Y.); dolee@knu.ac.kr (D.-E.L.); Tel.: +91-98-4342-0108 (N.Y.); +82-53-950-7540 (D.-E.L.)

Abstract: Wind tunnel testing techniques are the main research tools for evaluating the wind loadings of buildings. They are significant in designing structurally safe and comfortable buildings. The wind tunnel pressure measurement technique using pressure sensors is significant for assessing the cladding pressures of buildings. However, some pressure sensors usually fail and cause loss of data, which are difficult to restore. In the literature, numerous techniques are implemented for imputing the single instance data values and data imputation for multiple instantaneous time intervals with accurate predictions needs to be addressed. Thus, the data imputation capacity of machine learning models is used to predict the missing wind pressure data for tall buildings in this study. A generative adversarial imputation network (GAIN) is proposed to predict the pressure coefficients at various instantaneous time intervals on tall buildings. The proposed model is validated by comparing the performance of GAIN with that of the K-nearest neighbor and multiple imputations by chained equation models. The experimental results show that the GAIN model provides the best fit, achieving more accurate predictions with the minimum average variance and minimum average standard deviation. The average mean-squared error for all four sides of the building was the minimum (0.016), and the average R-squared error was the maximum (0.961). The proposed model can ensure the health and prolonged existence of a structure based on wind environment.

Keywords: wind-pressure coefficients; wind load; machine learning; data imputation; generative adversarial imputation network; tall building

check for updates

Citation: Kim, B.; Yuvaraj, N.; Sri Preethaa, K.R.; Hu, G.; Lee, D.-E. Wind-Induced Pressure Prediction on Tall Buildings Using Generative Adversarial Imputation Network. *Sensors* **2021**, *21*, 2515. https://doi.org/10.3390/s21072515

Academic Editor: Chelakara Subramanian

Received: 8 February 2021
Accepted: 30 March 2021
Published: 3 April 2021

1. Introduction

With the advancements in construction technologies, numerous large-span roof structures have been built, and clusters of tall buildings are abundant. These flexible, lightweight structures with low damping ratios and low frequencies are very sensitive to wind loads [1]. When wind passes through these tall buildings, wind loads acting on the buildings will reduce or amplify the loads equally [2]. The amplified wind loads may cause severe vibrations in buildings and discomfort to the occupants [3,4]. The building shapes play a vital role in determining wind-flow patterns and wind loads [5,6]. Wind effects on these structures may be low, moderate, strong, or extremely destructive. Moderate and low winds are beneficial, whereas strong and destructive winds may cause structural damages [7]. Additionally, changes in the wind environment around a building may affect the structural durability of nearby buildings [8]. Hence, considerable research efforts have focused on the changes in the wind environment, wind flow, and wind pressure characteristics in tall buildings [9–11]. It is thus necessary to understand the characteristics of the wind effects on each structure.

The structural safety of tall buildings depends on their structural designs and their capacities to withstand wind loads and wind-induced vibrations [12]. To understand the wind effects on a structure, it is essential to obtain wind load information for that particular structure [13]. Wind loads can be obtained from a wind tunnel test on a rigid structural model [14]. Wind tunnel tests focus on the measurement of the wind loads on structures, structural responses, and cladding effects on structures under different wind conditions [15]. The wind tunnel test also helps in evaluating the wind pressures generated on the surfaces of buildings with different shapes [16]. Pressure taps are installed on the surfaces of structures to collect wind-pressure values. It is necessary to find an effective method to predict wind pressures on the entire surface of a building even in the absence of, or with faulty pressure taps [17].

Rigid model testing, namely high-frequency base balance (HFBB), provides overall wind loads. The HFBB results can be analyzed using frequency- or time-domain techniques. The HFBB technique is based on concurrent measurements at different locations on the outer surfaces of buildings [18,19]. The wind tunnel test produces wind pressure coefficients, which can be used to analyze the effects of wind loads on buildings. Technological advancements render it promising for instantaneous measurement of wind pressures at more than 1000 locations on a building model with the deployment of pressure taps [20]. The autonomous monitoring of tall buildings was increased following the installation of various types of sensors to assess structural deformation [21], model parameters, and major stress on buildings [22,23]. Long-term building monitoring and maintenance is a necessary task that increases the longevity of buildings [24–28]. However, the functionality of the sensors diminishes with the effects of time and external factors, such as intense pressure and microparticles, which can damage the electronic circuits.

In addition, the pressure sensors installed in a building for monitoring wind pressure may fail in due course owing to vibration, shock, variation in pressure, electrical interference, and chemical damage. In tall buildings, it is very difficult to identify and replace faulty pressure taps at different locations. In such cases, data relayed from pressure sensors may be corrupted or missing. Data-centric research based on these data face a tremendous threat due to these inevitable data losses. Figure 1 illustrates the need for a data imputation model to analyze the wind-induced pressure response for the buildings with malfunctioning pressure taps and wind response at the new locations of the building without pressure taps installed. Thus, numerous techniques have been developed for imputing missing pressure tap values that may arise owing to tap failure or data loss [29,30]. Missing values are imputed by analyzing the correlation between other pressure coefficients [31]. Optimization techniques estimate the randomly missing time series pressure coefficients [32,33], and sampling techniques have been used to reconstruct the lost data [34].

However, most works related to data imputation in the existing literature focus on predicting the missing data from of multiple points at single time instances. Imputing the missing data for multiple points at multiple time instances remains challenging. This study focuses on handling the data loss by predicting the wind-induced pressure coefficients at different time intervals using a generative adversarial iterative network (GAIN). The performances of the models were quantified using standard statistical measures. GAIN is a light-weight ML model, and its performance shows that it predicts the values more accurately than the other deep learning algorithms applied in the literature [35–37]. The remainder of this paper is organized as follows. In Section 2, we summarize the related works in data imputation by machine learning and deep learning algorithms. Section 3 describes the process involved in generating the wind pressure data for buildings with a square section. Section 4 presents the proposed GAIN model and its implementation for predicting missing time series values. Section 5 presents the results and a comparative study of the proposed ML models for the prediction of the missing values. Section 6 concludes the paper and discusses the future scope.

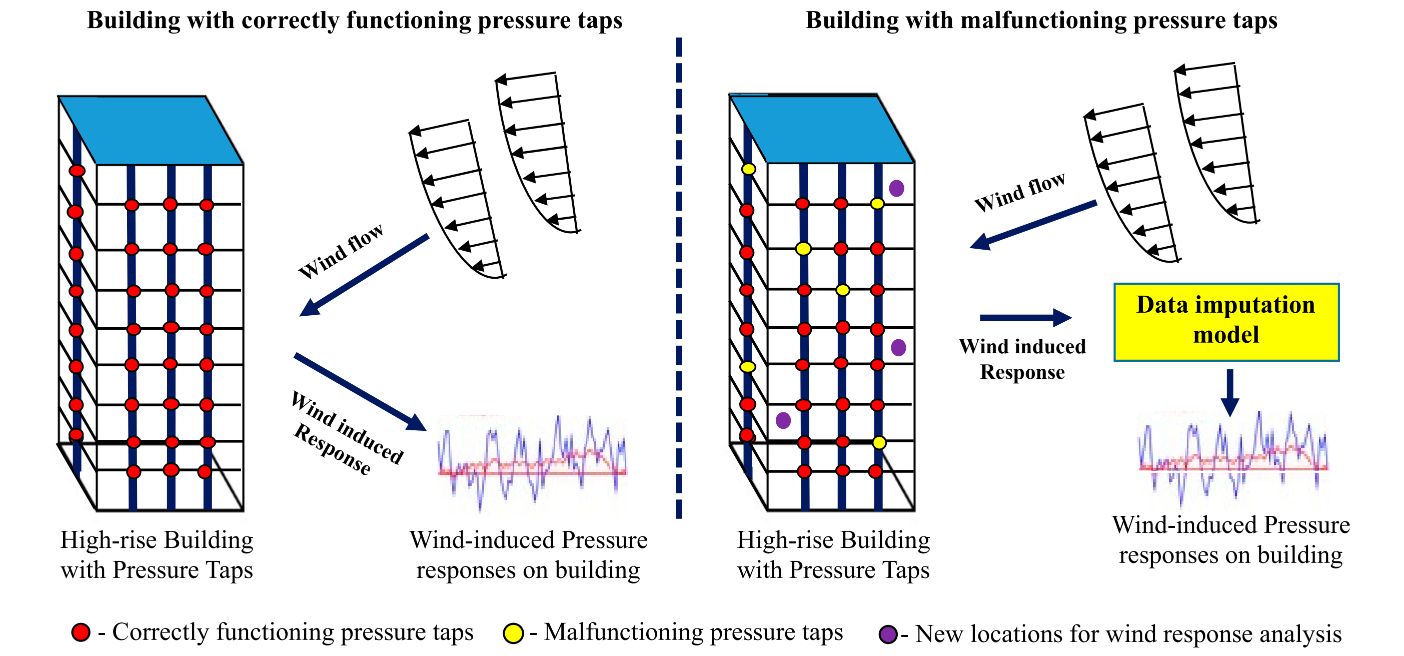

Building with correctly functioning pressure taps
Building with malfunctioning pressure taps
Wind flow
Wind induced Response
High-rise Building with Pressure Taps
Wind-induced Pressure responses on building
Wind flow
Wind induced Response
Data imputation model
High-rise Building with Pressure Taps
Wind-induced Pressure responses on building
● - Correctly functioning pressure taps
○ - Malfunctioning pressure taps
●- New locations for wind response analysis

Figure 1. Data imputation model for analyzing wind-induced response.

2. Related Works

In the literature, artificial neural network (ANN)-based multilayer perceptron (MLP) models were used to estimate the pressure coefficients on the walls and roof of a building by considering the building geometry and wind attack angle [38]. ANNs were used to predict the root-mean-square pressure coefficients of the buildings and the wind-induced pressure at different time intervals on various structures [39,40]. MLP and decision trees were implemented to analyze the nonlinear relationship between the various environmental factors for predicting the deformation of the unstable slopes [41]. Convolutional neural network (CNN) is a deep learning approach which is mostly applied for image-related applications. Regarding predictions, CNN is applied to analyze the wind-induced responses of high-rise structures. Time series data in the time and frequency domain were set as input to the CNN model. CNN models were used to predict the strains in the columns of buildings based on future wind loads with measured wind-response data [42]. Back Propagation Neural Network (BPNN) was implemented for time series predictive model to analyze the displacement based on the captured environmental factors [43]. BPNN and fuzzy neural network were developed to predict the wind pressures on a large gymnasium roof at different time intervals [44]. Subsequently, BPNN was integrated with proper orthogonal decomposition (POD-BPNN) to predict the wind pressure coefficients. The results showed that POD-BPNN was effective in predicting the individual pressure data of a tall building with the minimum number of pressure taps [45]. The Autoregressive recurrent networks (ARN) model based on time series was implemented for identifying the slope displacement [46,47]. ARN models find it difficult to operate on the missed data and impute it with meaningful time series values. Deep neural network (DNN) integrated with long short-term memory (LSTM) is used for the regression analysis to analyze the time series data [48,49]. A deep learning-based autoencoder network is implemented for landslide susceptibility prediction [50]. LSTM-based DNN manages to predict the discontinuity in the electrical resistivity, but dependency on the optimization of the large number of the hyperparameters remains as a major limitation during the time series prediction. Missing values can be imputed using any of the deep learning models, such as ANNs, CNNs, and recurrent neural networks [51–53]. Although neural-network-based models can predict wind pressures on structures, the prediction of wind pressure data at multiple points of the structures at different time intervals remains challenging [54].

Machine learning (ML) models are preferred over deep learning models for the prediction of missing wind pressure values due to pressure tap failure [55,56]. Random forest (RF) is one of the most widely applied ensemble methods to train the data for prediction by aggregating multiple weak learners [57]. The Extreme Gradient Boosting Regressor (XGBoost) algorithm [58] was implemented for forecasting short-term load in the power plant units. XGBoost struggles to attain the maximum performance with the sparse time series data of a large dataset [59,60]. Although both ML and deep learning work well for data imputation, the deep learning model uses multiple parameters for imputation, resulting in data overfitting [61,62].

To avoid overfitting, an ML-based shallow approach is deployed in this study to compute missing values. In addition, the proposed ML model can be trained and implemented with a smaller amount of data [63]. The missing values in several incomplete, multivariate datasets are imputed using many ML models such as singular value decomposition, general iterative principal component imputation, regularized expectation maximization, truncated total least-squares expectation maximization, K-nearest neighbor (KNN), multiple imputations by chained equations (MICE), and generative adversarial iterative network (GAIN) [64–67]. The KNN algorithm is used for short-term wind speed forecasting [65]. MICE is found to be more flexible in imputing categorical and quantitative (including skewed) variables [64]. GAIN was implemented on five real-world datasets obtained from the University of California (Oakland, CA, USA), Irvine repository to quantitatively evaluate the imputation performance. It was observed that GAIN models significantly outperformed the other state-of-the-art data imputation methods [68–70]. A comparison about the models used in the literature for data imputation and prediction is presented in Table 1.

Table 1. Comparison of techniques used for data imputation and prediction.

Technique	Learning Scenarios	Functionality	Pros	Cons
ANN/MLP	Supervised, unsupervised, reinforcement	Modeling data with simple correlations	Naïve structure, easy to build	Slow convergence rate, high complexity, and not suitable for heavy applications
BPNN	Supervised, unsupervised	Modeling the learning derivatives	Fast and simple, efficient for a clean dataset	Sensitive to noisy data, difficult to fix the learning rate
CNN	Supervised, unsupervised, reinforcement	Spatial data modeling	Weight sharing, customizable layer stack arrangement	High computational cost, difficult to optimize the hyperparameters
RCNN	Supervised, unsupervised, reinforcement	Sequential data modeling	Good in capturing the temporal dependencies	Heavily complex model, stuck with vanishing gradient, exploding problems occurs on complex data
ARN	Supervised, unsupervised	Modeling time series and interpretable model	Operates on variety of data and various conditions	Generating variable length output is difficult
Autoencoder	Unsupervised	Dimensionality reduction, compression	Very effective in computation, powerful for unsupervised learning	Pretraining is expensive Stuck with performance for timeseries data
DNN–LSTM	Supervised, unsupervised, reinforcement	Control problems with high dimensional inputs	Fully connected layer arrangement, can overcome vanishing gradient problem.	Depends on large amount of data, very expensive in computation

Table 1. *Cont.*

Technique	Learning Scenarios	Functionality	Pros	Cons
XG-Boost	Supervised, unsupervised	Modelling less feature engineering applications	Fast in operations, less overfitting	Difficult to optimize the hyperparameters
Randomforest	Supervised, unsupervised	Modelling applications for feature selection	Very effective in highly correlated features	Depend on highly correlated features
KNN	Supervised, unsupervised	Modelling instance-based applications	Easy implementation, evolving model for new data points	Depends on homogeneous features
MICE	Supervised, unsupervised	Data imputations	Flexible, can handle variables of varying types	Sensitive to outliers, depends on homogeneous features
GAIN	Supervised, unsupervised	Data generations	Effective in generating the similar patterns	Convergence is difficult

3. Materials and Methods

Analyzing the wind-induced response on the high-rise structures remains important for observing the impact of wind on the structures. As the impact of the wind directly affects the durability of the structures, wind-induced responses on the structures plays a vital role to enhance the longevity of the structures with effective structural maintenance [71–74]. The technical improvements in implementing the pressure sensors around the high-rise structures for monitoring the wind-induced response gained its maximum attention. Furthermore, the advancements in the field of artificial intelligence (AI) leads a way to develop an integrated system to analyze the deep insights out of the data generated from the pressure sensors. The operational failures and malfunctions that occurs in the pressure taps produce the missing data. Analyzing the wind-induced response from the missing data will not lead to a meaningful insight. In addition, it is also necessary to develop a model which is capable of analyzing the wind-induced response of the building at random locations of the high-rise structures in the absence of pressure taps [75–77]. This proposed work concentrates on developing a data imputation model capable of imputing the missing values. Section 3.1 discusses the wind-tunnel test setup for generating the wind-induced response data. An overview of developing an intelligent data prediction model is discussed in Section 3.2.

3.1. Wind Tunnel Test and Wind Pressure Data

In this study, an aerodynamic database of wind pressures on tall buildings was constructed based on wind tunnel experiments. A tall building model with a square section and dimensions of height (*H*): 600 mm, width (*W*): 120 mm, and breadth (*B*): 120 mm was used in the pressure measurement tests. A synchronous multi-pressure measurement system was employed because of its capability to simultaneously measure multiple-point local pressures on the building model. The oncoming flow conditions were set in terms of Category 2 terrain according to the Australian/New Zealand standard (AS/NZS 1170.2:2011). The normalized wind characteristics between the measured profiles and target profiles are compared in Figure 2.

The mean wind speed at the top of the building model was 13 m/s. Hence, the Reynolds number was 1.03×10^5, which was higher than the minimum Reynolds number requirement as specified in AWES-QAM-1-2001 [78,79]. The sampling frequency was set to 800 Hz, and the measurement duration was 150 s. The local pressure coefficients were calculated using the following Equation:

$$C_{p\_ij} = \frac{p_{ij}(t) - p_0}{\frac{1}{2}\rho V_H^2} \tag{1}$$

where $i = 1, 2, \ldots, 10$ indicates the pressure tap level, j represents the pressure tap number in each level, p_0 is the local static pressure, ρ is the air density, and V_H is the velocity at the top of the building model. In this proposed work the incoming wind flow is considered as 0° on the building model placed in the wind tunnel test. As the wind flows at 0°, the front face of the building is perpendicular to the incoming wind.

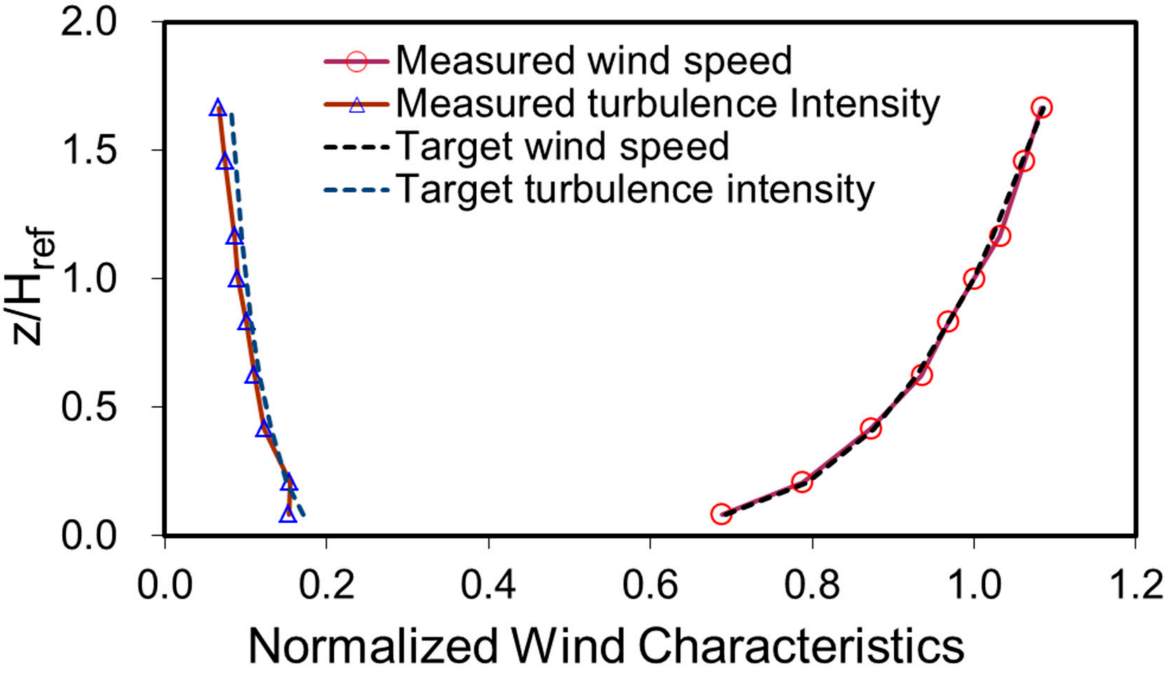


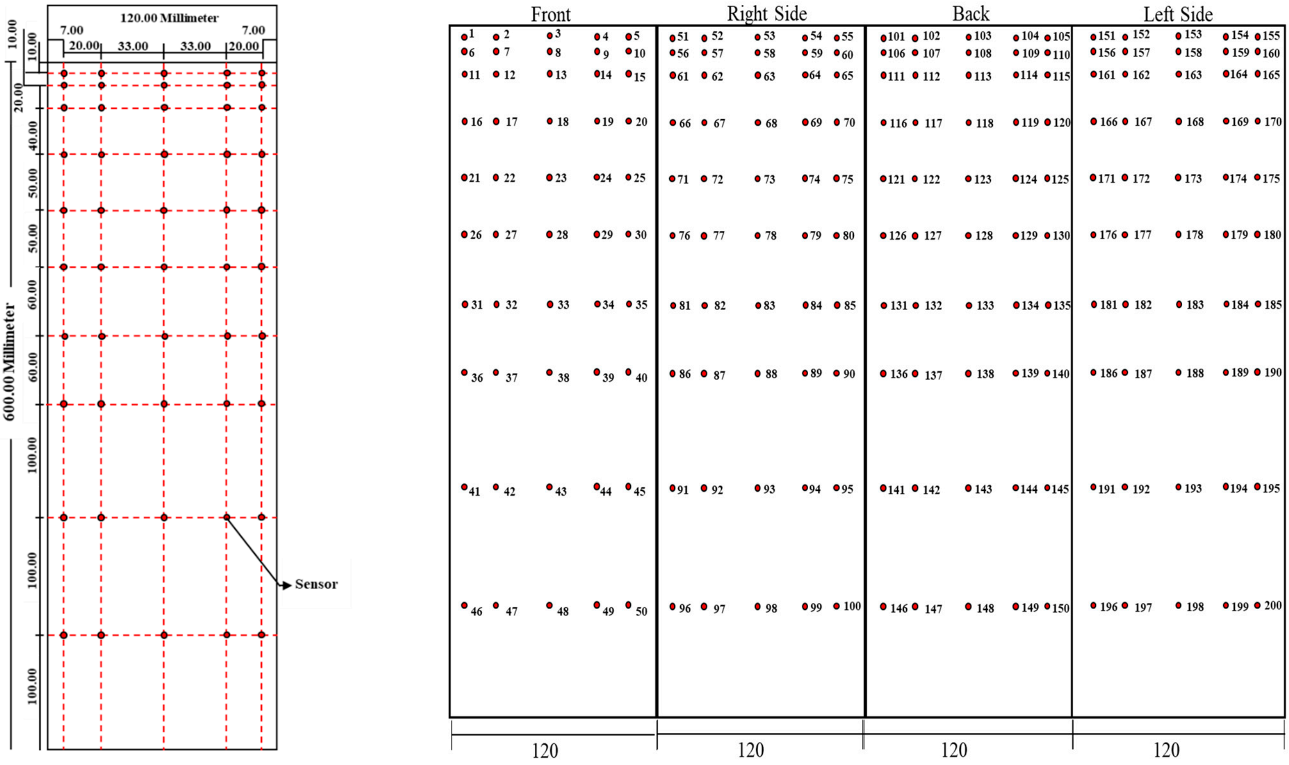
Figure 2. Wind profiles in the wind tunnel. z and H_{ref} represent the heights of the measured position and building top, respectively.

A total of 200 pressure taps were distributed on all four faces of the building, and each face had 50 pressure taps. Figure 3a illustrates the arrangement of pressure taps on the building model, and Figure 3b shows the pressure tap locations on the four faces. The measured wind pressures on the building surface were utilized as a database to train the ML model to predict wind pressures on untested locations of the proposed building.

Data preprocessing yields the most significant outcome of any data-based study. Corrupted data and missing values remain a challenge for data-driven research. Computing a missing value has been a lifesaver for such studies. Conventionally, data imputation is performed with simple mathematical computations, such as imputation using mean/median values, using the most frequent values, or simply using a constant value. However, these conventional imputation techniques have flaws, such as data inaccuracies, data bias, and effects on data correlations. ML-based imputation models, such as GAIN, KNN, and MICE, have been proven to be effective techniques for predicting missing values while maintaining the data accuracy and intactness. In recent years, GAIN has revolutionized many fields through its implementation of the ML paradigm [70]. GAIN has transformed data imputation with its highly accurate imputation predictions.

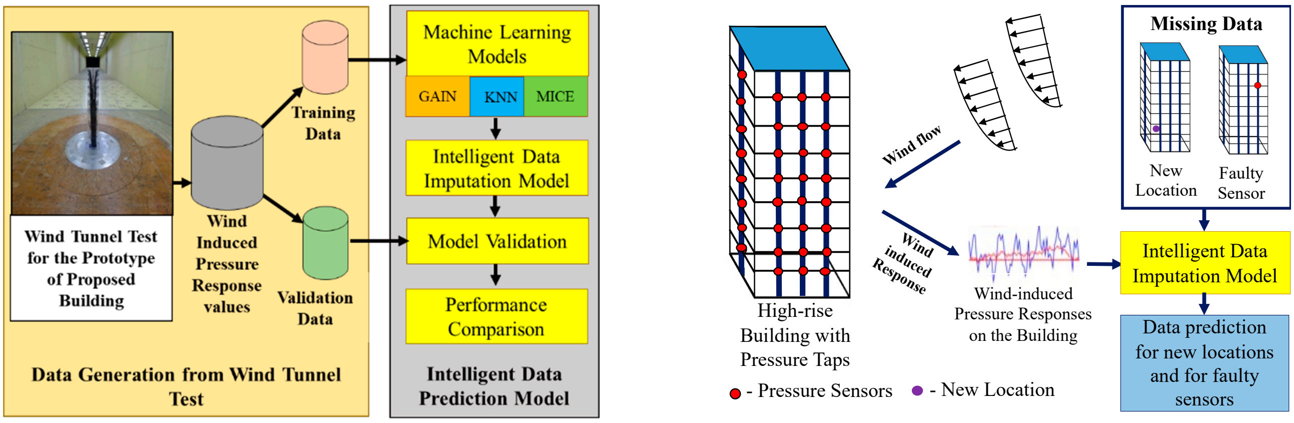
3.2. *Intelligent Data Prediction Model*

This study aims to build an intelligent data prediction model (IDPM) using ML algorithms. The GAIN-based data prediction model is adopted to impute the missing pressure data at locations with faulty sensors or without pressure taps. The pressure values predicted by GAIN are validated by comparing the model's performance with those of KNN and MICE. Figure 4a presents the workflow of the ML-based intelligent data prediction model. Eighty-five percent of the measured pressure data were used for training the model and the rest of the data were used for model validation.



(**a**) (**b**)

Figure 3. Pressure tap locations (**a**). Relative locations of pressure taps on the proposed building model; (**b**) pressure tap locations on all four sides of the building.


Wind Tunnel Test for the Prototype of Proposed Building
Wind Induced Pressure Response values
Training Data
Validation Data
Data Generation from Wind Tunnel Test
Machine Learning Models
GAIN
KNN
MICE
Intelligent Data Imputation Model
Model Validation
Performance Comparison
Intelligent Data Prediction Model
High-rise Building with Pressure Taps
Wind flow
Wind induced Response
Wind-induced Pressure Responses on the Building
Missing Data
New Location
Faulty Sensor
Intelligent Data Imputation Model
Data prediction for new locations and for faulty sensors
● - Pressure Sensors
● - New Location

(**a**) (**b**)

Figure 4. IDPM model. (**a**) Workflow of machine-learning-based intelligent data prediction model for wind pressures; (**b**) workflow of intelligent data prediction model to predict the missing values.

Figure 4b presents the workflow of the IDPM. For each face with 50 pressure taps, the data from 1000 instantaneous time intervals data of 50 pressure taps for each side of the building is considered. For each face with 50 pressure taps, wind pressures acquired from all 50 pressure taps were used for training the model among which the values from 7 pressure taps were intentionally imputed with NaN (Not a Number) values to represent the missing values and used for model validation. The NaN was included at two different patterns, namely, missing at random (MAR) and missing completely at random (MCAR). This ensures that 15% of the data contains missing values among the entire data considered for training. Eighty-five percent of the wind pressures measured at 1000 time intervals in the wind tunnel tests were used for training; the rest were used for testing.

4. Construction of Wind-Induced Pressure Prediction Model

4.1. GAIN

GAIN is a method that generalizes the well-known generative adversarial network (GAN) framework, which can impute data when complete data are unavailable. GAIN is composed of two networks: generator and discriminator. The goal of generators is to impute the missing data, and the discriminator attempts to distinguish the observed components and imputed components [53]. These two networks are trained in an adversarial manner so that the discriminator minimizes the classification loss, and the generator is trained to maximize the classification loss of the discriminator.

In the process of data imputation, the generator (G) imputes the missing components and outputs a completed vector to the discriminator (D). Then, D attempts to determine which components were actually observed and which were imputed. The hint vector (H) is used to provide additional information to ensure that D forces G to learn. H reveals partial missing information to D to focus its attention on imputation quality. In addition, H ensures that G learns to generate the true data distribution.

GAIN is the new state-of-the-art algorithm for data imputation, which uses a modified GAN architecture (Figure 5). The pressure coefficients are obtained from 50 pressure sensors in the form of a 10 × 5 matrix. Some specific values in the matrix are converted to NaN values to extrapolate the malfunctioning sensor and missing sensor values. The reshaped matrix of pressure coefficients is transformed into data, random, and mask matrices. The data matrix is an instance of the input that retains all the available sensor values in position but is filled with zeros instead of NaN for missing values. The random matrix consists of randomly generated values in place of the missing values, and the rest of the sensors' actual values are changed to zeros. The mask matrix is a binary matrix that indicates the missing positions in a matrix using zeros in place of the missing values and ones for occupied positions.

Using these three matrices, G then generates an imputed matrix. This imputed matrix and the data matrix are compared to calculate the mean-squared error (MSE). The hint generator generates a hint matrix depending on the mask matrix, which ensures that D forces G to learn. D outputs the estimated imputed matrix, which is compared with the mask matrix, and the loss is calculated in terms of D's performance of correctly identifying the missing value. It is used to back propagate to change D's weights. This loss found in D is also given in summation with the MSE found before G for back propagation. This back propagation is performed until the optimum results are achieved. Both networks are said to learn the parameters during back propagation.

Missing Data Imputation Using GAIN

The prediction of the pressure values at the missing locations is formulated as a data imputation problem. Consider a d-dimensional space X = X1 × . . . × Xd. Represent X = (X1, . . . , Xd) as a random variable taking values in χ with a distribution P(X).

Let M = (M1, . . . , Md) be a random variable that takes values as {0,1} in d-dimensional space. X represents the data vector and M is the mask vector. For each i ϵ {1, . . . , d}, denote $\widetilde{\chi}_i$ = χi ⋃ {*}, where * is a point that does not belong in any χ_{i}, and it represents an unobserved value.

Let $\widetilde{\chi} = \widetilde{\chi}_1 \times \ldots \times \widetilde{\chi}_d$ and $\widetilde{\mathrm{X}} = (\widetilde{X}_1, \ldots, \widetilde{X}_d)$, where

$$\overline{\mathrm{X}}\mathrm{i} = \begin{cases} X_{i,\, if\ M_i = 1} \\ *_{,}\ Otherwise \end{cases} \tag{2}$$

M indicates which components of X are observed.

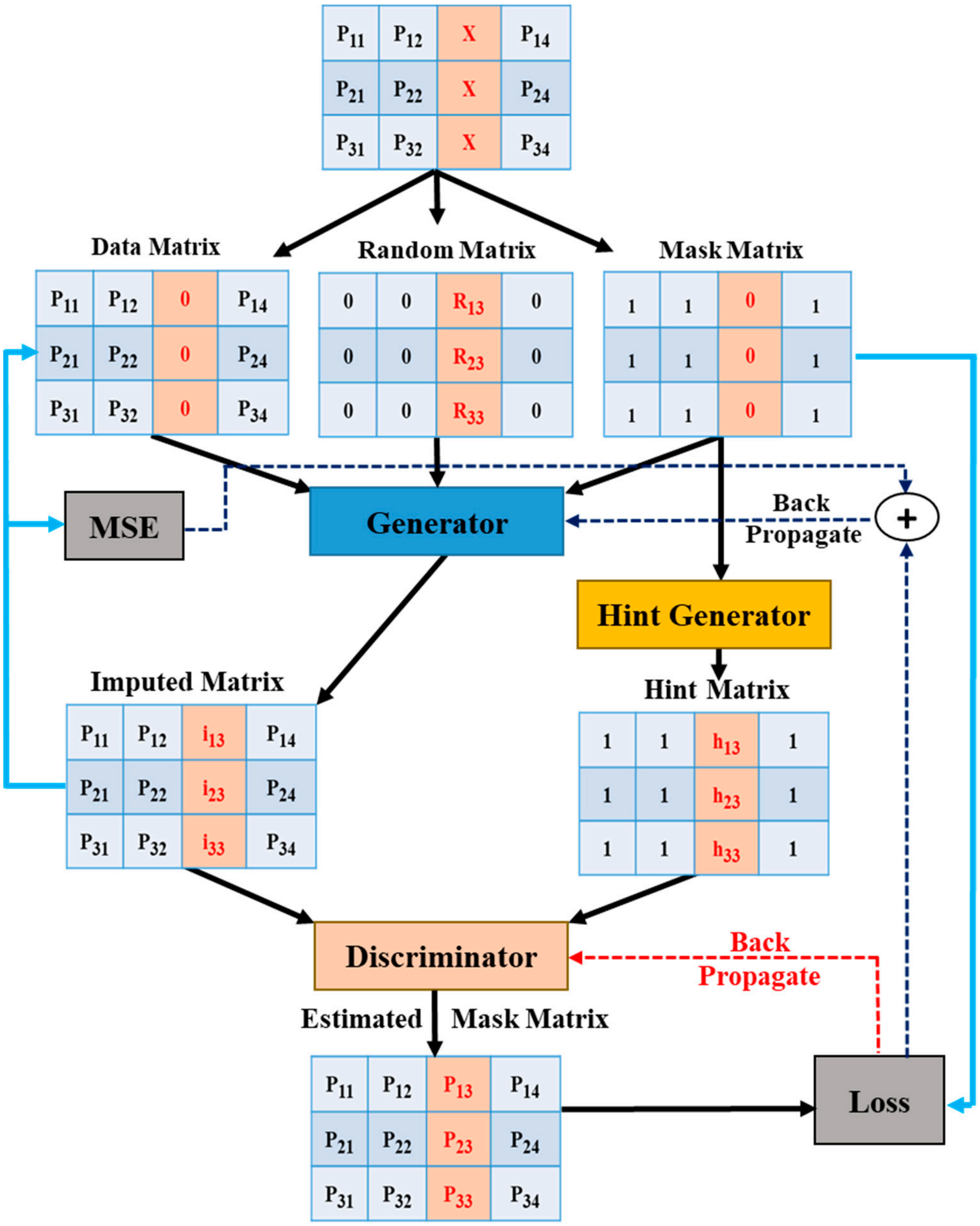


Figure 5. Operation of generative adversarial imputation network (GAIN).

In imputation, the goal is to impute the unobserved values in each χ_i. To compute the unobserved values, it is necessary to generate samples according to $P(X \,|\, \widetilde{X} = \widetilde{x}^i\ )$. The conditional distribution of X is represented as $\widetilde{X} = \widetilde{x}^i$ and is used to fill missing data points. To impute the missing values in the data collected, a vector with real data with some missing values is provided as an input to G, and the missing values are imputed accordingly. D takes the imputed data and determines which data were originally missing.

G takes the value of $\widetilde{X}$ as input along with the Z and M values. M is a noise variable, Z denotes the noise vector, and $\widetilde{P}$ is the output variable with the missing values being imputed. Let G: $\widetilde{p}$i × {0,1}n × [0, 1]n → X be a function, and Z = (Z1, ..., Zn) be a n dimensional noise variable. The random variables $\widetilde{P}$, $\hat{P}$ € X are expressed by

$$\overline{P} = G\left(\widetilde{P}, M, (1-M)\odot Z\right) \tag{3}$$

$$\hat{P} = M \odot \hat{P} + (1-M)\odot \overline{P} \tag{4}$$

where $\odot$ denotes element-wise multiplication, $\overline{\mathrm{P}}$ corresponds to the vector of the imputed values, and $\hat{\mathrm{P}}$ corresponds to the completed data vector. This is similar to the standard GAN with Z being analogous to the noise variables in the framework. D is introduced and used as an adversary to train G. In a standard GAN framework, the output of the generator is either completely real or completely fake. To identify the entire vector as real or fake, the discriminator attempts to distinguish the real components and fake components for predicting the mask vector, m. The discriminator function is given as D: $P \to [0,1]^{\mathrm{n}}$ with the i-th component of D ($\hat{P}$) corresponding to the probability that the i-th component of $\hat{X}$ is observed.

A hint mechanism is introduced to avoid the failure of the missing data imputation algorithm. If sufficient information about **M** is not provided to D, then multiple distributions can be generated by G that are optimal with respect to D. To overcome the data insufficiency, the hint-matrix mechanism is followed. H depends on M, and for each imputed sample ($\hat{P}$, m), h is drawn according to the distribution H∣M = m. h is passed as an additional input to D, and thus it becomes a function D: P × H→ $[0,1]^{\mathrm{d}}$, where the i-th component of D ($\hat{P}$, h) corresponds to the probability that the i-th component of $\hat{P}$ was observed subject to the condition that $\hat{\mathrm{P}}$ = $\hat{P}$ and H = h. By defining H in different ways, the amount of information contained in H about M is controlled.

The goal of the discriminator is to maximize the probability of the successful prediction of M, and the goal of the generator is to minimize this probability. Then, define the quantity V(D,G) as

$$V(\mathrm{D},\mathrm{G}) = \mathrm{E}\hat{p}, M, H\left[\mathrm{M}, \log \mathrm{D}(\hat{\mathrm{X}},\mathrm{H}) + (1-\mathrm{M}), \log(1-\mathrm{D}((\hat{\mathrm{X}},\mathrm{H})))\right] \quad (5)$$

where *log* is the element-wise algorithm, and $\hat{\mathrm{X}}$ depends on G. The imputation model is associated with the standard GAN, and the objective of GAIN is then defined to minimax the problem as

$$\min_{G}\max_{D} V(D,\ G) \quad (6)$$

Define the loss function $\mathcal{L}$: $\{0,1\}^{\mathrm{d}} \times [0,1]^{\mathrm{d}\to}$ R by,

$$\mathcal{L}(a,b) = \sum_{i=1}^{d} a_i\ \log b_i\ + (1-\ a_i)\log(1-b_i) \quad (7)$$

By expressing $\hat{\mathrm{M}}$ = D ($\hat{\mathrm{X}}$, H), (6) can be rewritten as,

$$\min_{G}\max_{D} E\left[\mathcal{L}(M,\ \hat{M})\right] \quad (8)$$

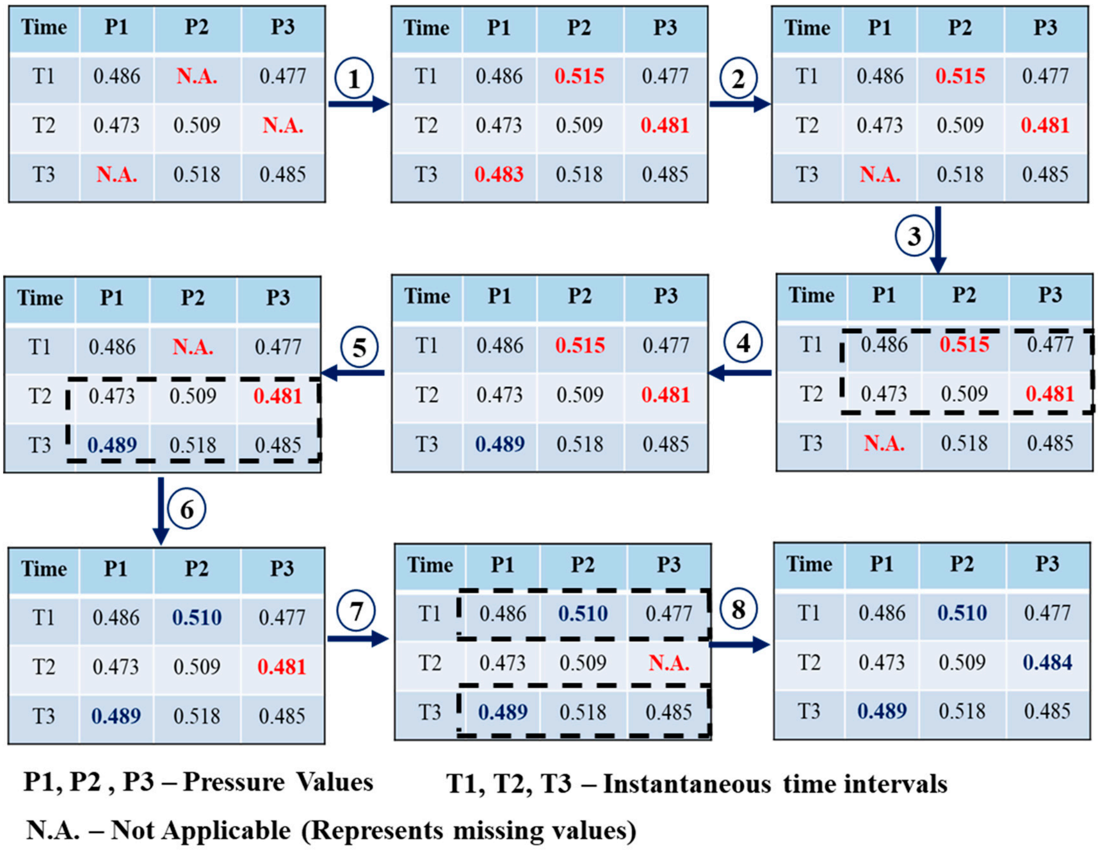
The working code explaining the functionality of GAIN (Algorithms 1) for the prediction of the wind-induced pressure values is presented in the following section.

In MICE, each missing value was predicted multiple times by multiple imputations. The analyses of imputation values were considered to handle the uncertainty and achieve the minimum standard errors. If there is no meaningful information regarding the missing values, the imputation will vary and will lead to high standard errors. In contrast, if the observed data are meaningful and highly predictable, the imputations will be more consistent with more accurate values. MICE can operate based on the assumption that the given variables used in the imputation procedure are missing-at-random and missing-at-complete. The probability of the missing value depends only on the observed values and not on the missing values. The MICE method runs a series of regression procedures, wherein each missing variable is modeled conditionally on other data variables. The regression procedure normally uses logistic regression for binary variables and linear regression for continuous variables.

The MICE algorithm, which operates on the pressure coefficients at various time intervals, is shown in Figure 6.

Algorithms 1 GAIN for data imputation.

```
1. While training loss has not converged do
2. Discriminator (D)
3. Get $k_D$ samples from the dataset $\{(\overline{x}(j),\ m(j))\}_{j=1}^{k_D}$
4. Get $k_D$ independent and identically distributed samples $\{z(j)\}_{j=1}^{k_D}$ of Z
5. Get $k_D$ independent and identically distributed samples $\{b(j)\}_{j=1}^{k_D}$ of B
6. For j = 1 . . . $k_D$ do
7.     $\overline{x}(j) \leftarrow G(\tilde{x}(j),\ m(j),\ z(j))$
8.     $\hat{x}(j)\ \leftarrow m(j) \odot \tilde{x}(j) + (1 - m(j)) \odot \overline{x}(j)$
9.     $h(j) = b(j) \odot m(j) + 0.5(1 - b(j))$
10. End for
11. Update D using adaptive moment estimation optimization (Adam) using the loss obtained
from the loss function of D
12.   $\nabla_D - \sum_{j=1}^{k_D} \mathcal{L}_D(m(j),\ D(\hat{x}(j),\ h(j)),\ b(j))$
13. Generator (G)
14. Draw $k_G$ samples from the dataset $\{(\overline{x}(j),\ m(j))\}_{j=1}^{k_G}$
15. Draw $k_G$ independent and identically distributed samples $\{z(j)\}_{j=1}^{k_G}$ of Z
16. Draw $k_G$ independent and identically distributed samples $\{b(j)\}_{j=1}^{k_G}$ of B
17. For j = 1 . . . $k_G$ do
18.     $h(j) = b(j) \odot m(j) + 0.5\ (1 - b(j))$
19. End for
20. Update G using Adam (for fixed D) based on the loss obtained from the loss function of G
21. $\nabla_G - \sum_{j=1}^{k_G} \mathcal{L}_G(m(j),\ \hat{m}(j),\ b(j)) +\ \alpha\mathcal{L}_M(x(j),\ \tilde{x}(j))$
22. End while
```



Time	P1	P2	P3
T1	0.486	N.A.	0.477
T2	0.473	0.509	N.A.
T3	N.A.	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.515	0.477
T2	0.473	0.509	0.481
T3	0.483	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.515	0.477
T2	0.473	0.509	0.481
T3	N.A.	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.515	0.477
T2	0.473	0.509	0.481
T3	N.A.	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.515	0.477
T2	0.473	0.509	0.481
T3	0.489	0.518	0.485

Time	P1	P2	P3
T1	0.486	N.A.	0.477
T2	0.473	0.509	0.481
T3	0.489	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.510	0.477
T2	0.473	0.509	0.481
T3	0.489	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.510	0.477
T2	0.473	0.509	N.A.
T3	0.489	0.518	0.485

Time	P1	P2	P3
T1	0.486	0.510	0.477
T2	0.473	0.509	0.484
T3	0.489	0.518	0.485

P1, P2 , P3 – Pressure Values **T1, T2, T3 – Instantaneous time intervals**

N.A. – Not Applicable (Represents missing values)

Figure 6. Workflow of multiple imputation by chained equations (MICE).

The chained equation process is explained based on the MICE algorithm for a set of time series pressure values $p_1, \ldots., p_{k.}$ Assuming that some of the pressure values are missing, all missing values are initially filled randomly or with the mean values.

$$\overline{P} = \left(\sum p_i\right)/\mathrm{n} \tag{9}$$

where p_i is the time series value and n is the total number of values. This means that the imputation can be considered as the set of filler values. The filler values are obtained by the imputation of one variable and are reset to missing. The first variable with at least one missing value, p_1, is regressed on the other variables $p_2, \ldots., p_k$. In this regression model, given in Equation (3), variable p_1 is a dependent variable, and the other variables act as independent variables. The model operates based on the same assumptions of linear regression while imputing the missing data. The missing value for p_1 is predicted by the regression model, and the dataset is imputed with the new predicted values. All missing data are imputed as described above.

$$p_1 = \mathrm{a} + \mathrm{b}\ p_2 + \epsilon \tag{10}$$

where p_1 is the dependent variable, p_2 is the independent variable, a is the intercept, b is the slope, and ϵ is the residual.

Filling each of the missing variables constitutes one iteration. Missing values are predicted at the end of one iteration using the regression relationship observed in the data. The number of iterations is determined based on the error rate calculated at the end of each iteration. The iteration procedure is stopped at a minimum error rate with stable values. The optimum number of iterations depends on the type of data and the number of data values missing in a specific dataset.

4.2. KNN

The KNN algorithm is an effective supervised ML algorithm used for data imputation. KNN relies on labeled input data to build a function that produces an appropriate output while handling the missing data. KNN operates based on the assumption that similar data values are in close proximity with each other [57,58]. The KNN measures the similarity between two points by measuring their distance. Although there are different methods to quantify the distance between two points, the most commonly used is the Euclidean distance.

The time series of wind pressures are represented as vectors in a multidimensional feature space. An ML model for data imputation can be developed by using KNN and by considering all the available time series values for training. The value *K* in KNN is a user-defined constant. During the model training process, the KNN algorithm is iterated with multiple *K* values, and the error associated with each *K* value is monitored. This iteration process will be stopped at a particular *K* value when the values are stable and the error rate is minimal. The prediction accuracy of KNN can be improved significantly by selecting the optimum value of *K*.

The KNN algorithm takes wind pressures as input. The dataset is split into a training and testing data. The random number of neighbor values is initialized as *K*. The distance between the pressure points is calculated using the Euclidean measure according to Equation (4).

$$\mathrm{d}(\mathrm{x},\mathrm{y}) \ = \ \sqrt{\sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2} \tag{11}$$

where Y_i and $\hat{Y}_i$ are Euclidean vectors that start from the origin of the space and have specific ending points. *K* nearest points are selected based on the minimum distance measure, and the mean pressure value for all available *k* points is then calculated. The error rate is calculated for the associated *k* values. The procedure is repeated for the next *k* values, and iterations are performed for different seed values. The iteration procedure is stopped when the error rate is minimized. The MSE is calculated for all associated *k* values according to Equation (5).

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 \tag{12}$$

The optimum *K* value is selected based on the accuracy of data imputing. The KNN model is imputed several times. Different *K* values and the optimum values are selected based on the error rate. The prediction becomes less stable when the *K* value decreases. For example, if the value of *K* is selected to be one, it simply replaces the missing value with the nearest value. This will become an unstable value. As the value of *K* increases, the predictions become more stable owing to averaging done by the majority of the neighboring values, and thus more accurate predictions are likely to be made. However, it is possible to witness an increasing value of errors when the *K* value is pushed too far.

5. Performance Discussions

This section presents the experimental results of GAIN and the performance comparison of proposed GAIN-based wind-induced pressure prediction with other ML models. In this proposed work, the GAIN model is proposed as an imputation model for filling the missing values generated from the pressure sensors. The performance of the GAIN model is validated by comparing its performance with other familiar ML models, such as MICE and KNN (Figure 7).

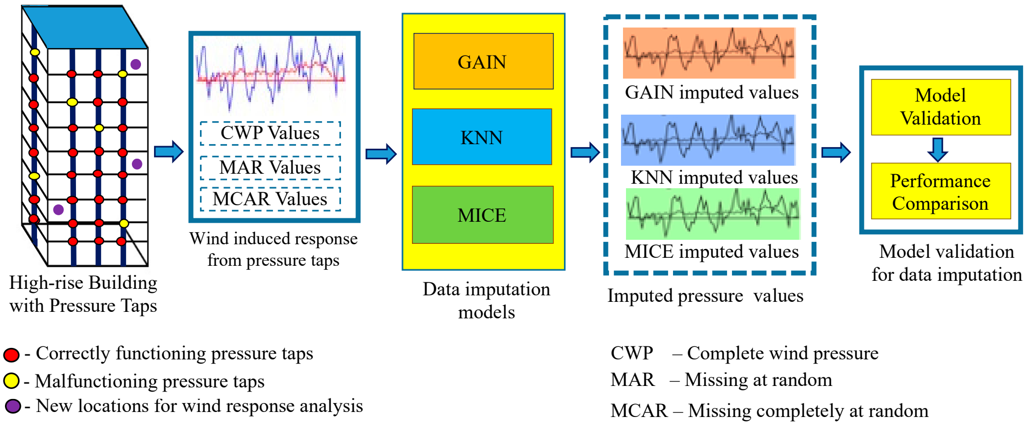

High-rise Building with Pressure Taps
CWP Values
MAR Values
MCAR Values
Wind induced response from pressure taps
GAIN
KNN
MICE
Data imputation models
GAIN imputed values
KNN imputed values
MICE imputed values
Imputed pressure values
Model Validation
Performance Comparison
Model validation for data imputation
● - Correctly functioning pressure taps
● - Malfunctioning pressure taps
● - New locations for wind response analysis
CWP – Complete wind pressure
MAR – Missing at random
MCAR – Missing completely at random

Figure 7. Data imputation models and performance comparison.

5.1. *Experimental Results of GAIN*

The performance of the GAIN algorithm is quantified by calculating the average mean-squared error (AMSE) and average R-squared error (ARSE) values for all four sides of the building. The AMSE of the building was measured to be 0.011, 0.013, 0.021, and 0.019 for front, back, side 1, and side 2 of the building, respectively, as shown in Figure 8. The ARSE values were 0.95, 0.962, 0.972, and 0.96, for front, back, side 1, and side 2 of the building, respectively, as shown in Figure 9.

The time series plots of the AMSE and ARSE values of GAIN for all four sides of the building are shown in Figures 10 and 11 respectively.

In addition to the AMSE and ARSE values, the performance of the GAIN algorithm was quantified in terms of accuracy and loss values at different iterations of GAIN. The GAIN algorithm was executed with 5000 iterations and the accuracy and loss values were calculated (Figures 12–14).

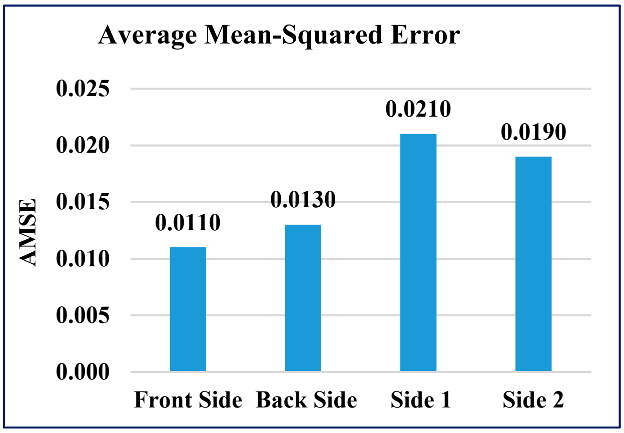


Figure 8. Average mean-squared error (AMSE) of GAIN.

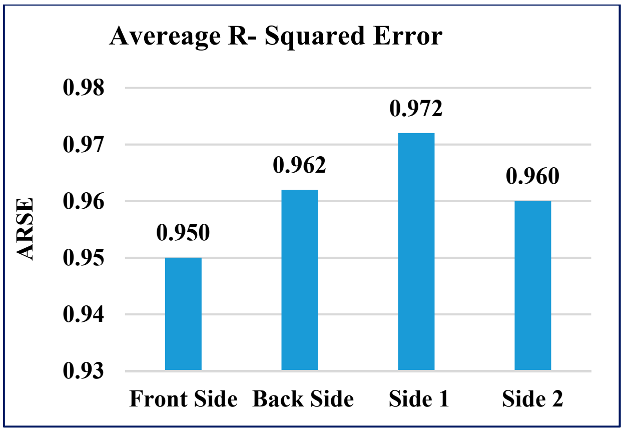


Figure 9. Average R-squared error (ARSE) of GAIN.

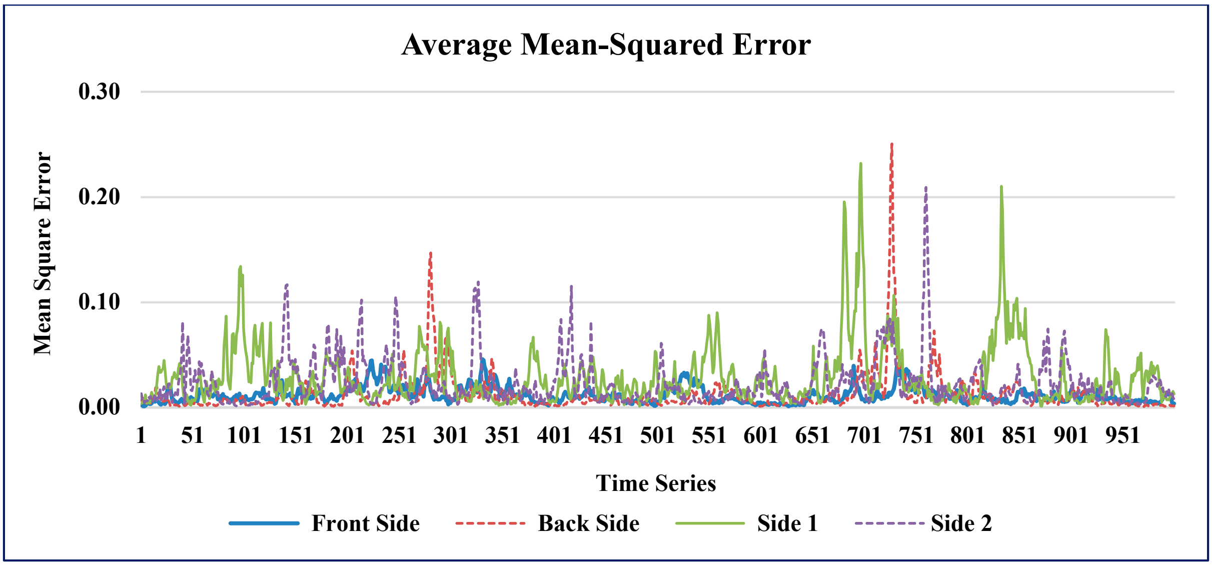


Figure 10. AMSE predicted by GAIN.

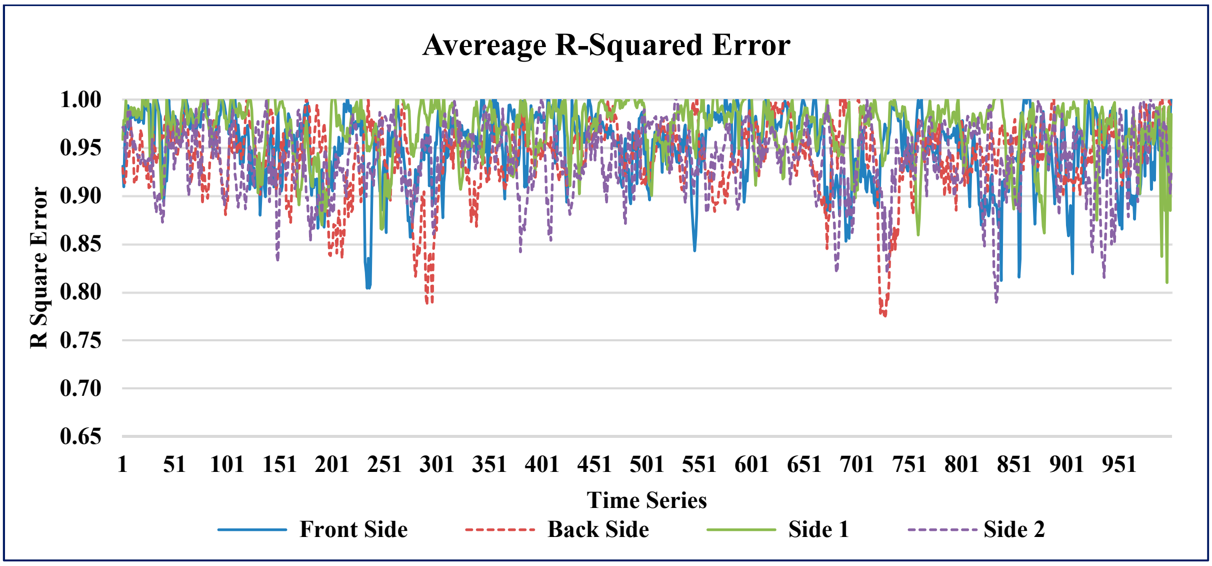


Figure 11. ARSE predicted by GAIN.

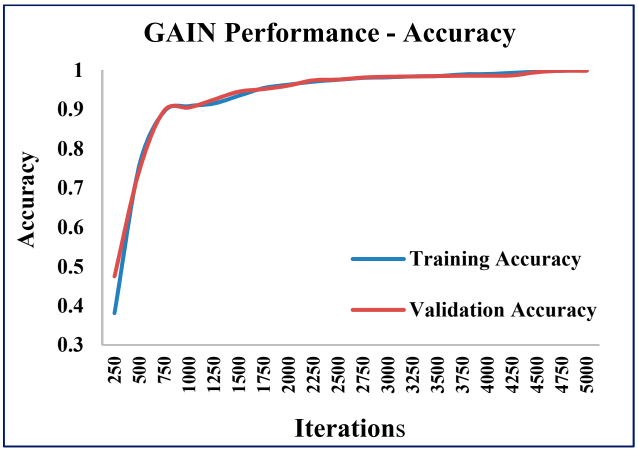


Figure 12. GAIN accuracy measure.

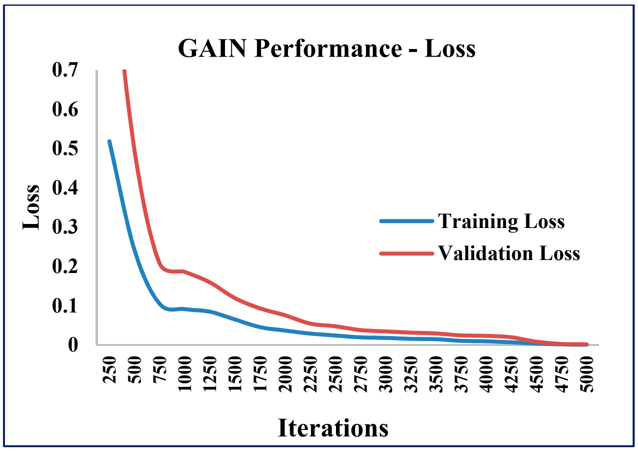


Figure 13. GAIN loss measure.

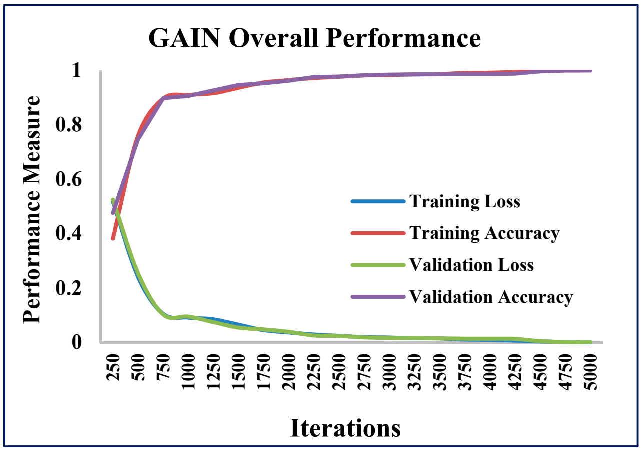


Figure 14. GAIN overall performance.

As the iterations increased, the validation accuracy also increased, and the validation loss decreased. The training and validation accuracies reached the maximum (Figure 12) and stabilized as the iterations increased. The training and validation losses reached the minimum (Figure 13) as the iterations increased.

5.2. Experimental Comparisons

The computational performance of GAIN was compared with those of the MICE and KNN models. Initially, the performances of the proposed algorithms were evaluated in terms of AMSE and ARSE by comparing the ground truth and predicted values. During experimentation, MICE was iterated with various *i* values, such as 3, 5, 7, 8, 9, and 10, to identify the best fit for missing values. Among all the iterations, the results at the eighth iteration (*i* = 8) produced the best fit with the AMSE and ARSE values. Similarly, KNN was experimented for different k values ranging from 2 to 9. The results at k = 3 produced the best fit with the minimum AMSE and maximum ARSE values.

Considering the accuracy of the proposed models, the maximum accuracy in this study was 0.95, and it was achieved by the GAIN prediction model. The MICE and KNN models struggle to attain the maximum accuracy using an iteration model separately for distinct iteration in MICE and by using different k values in KNN. The GAIN model was iterated and the discriminator provided better accuracy with increasing iterations. In addition, the maximum accuracy and minimum loss values were achieved by the GAIN model in successive iterations. However, other models exhibited variations and decreased accuracy, although the k and i values increased. In summary, it is found that the GAIN model works well for finding missing values with high ARSE, low AMSE, decreased loss, and increased accuracy values. GAIN maintains the highest ARSE values (Figure 15) of 0.95, 0.962, 0.972, and 0.962 for the front, back, side 1, and side 2 of the building, respectively. In this proposed work, the building perpendicular to the wind flow is considered for developing the intelligent data imputation model. So, the wind induced responses generated by the pressure taps in the front face of the building is dispersed compared with the other faces of the building. Hence the variance and standard deviation values at the side of the building differs from the sides of the building.

GAIN produces the minimum AMSE values (Figure 16) of approximately 0.011, 0013, 0.021, and 0.019 for front, back, side 1, and side 2, respectively. The KNN algorithm maintains an average AMSE of approximately 0.015, 006, 0.039, and 0.031 for front, back, side 1, and side 2, respectively. The MICE algorithm values deviate considerably from the original values based on the maximum AMSE values in comparison with GAIN and KNN. It is observed that the variance and standard deviation values at the sides of the building remains much closer to each other (Figures 17 and 18). In addition, GAIN manages to produce the minimum standard deviation (Figure 17) and minimum variance (Figure 18).

These analyses indicate that the GAIN model is well suited to the prediction of missing pressure value.

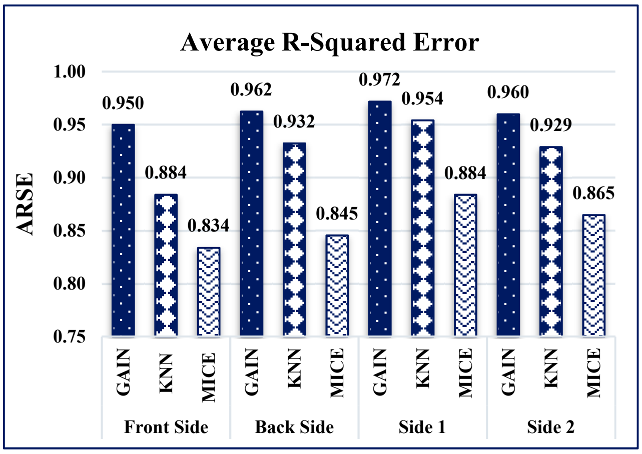


Figure 15. ARSE comparison.

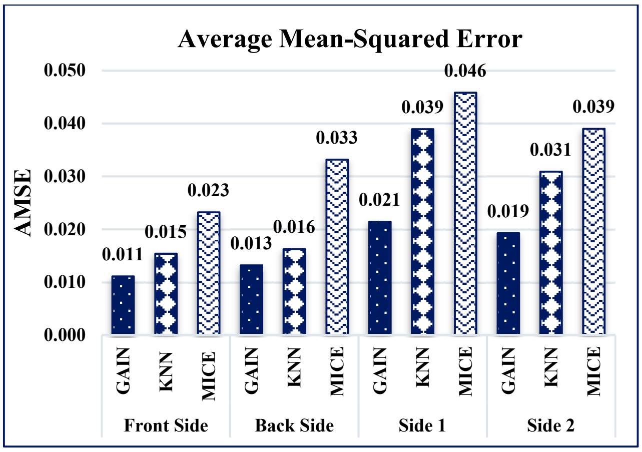


Figure 16. AMSE comparison.

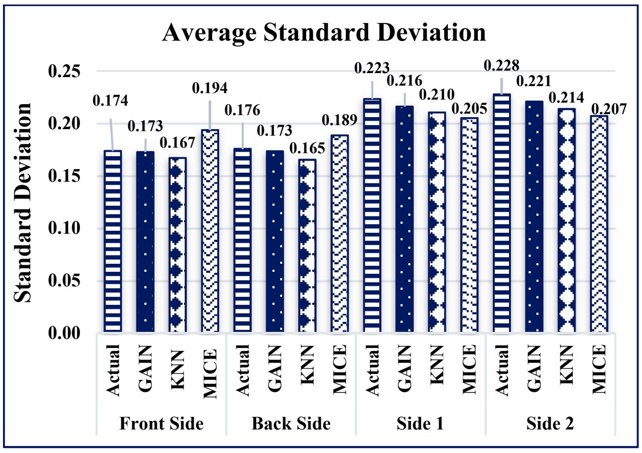


Figure 17. Average standard deviation comparison.

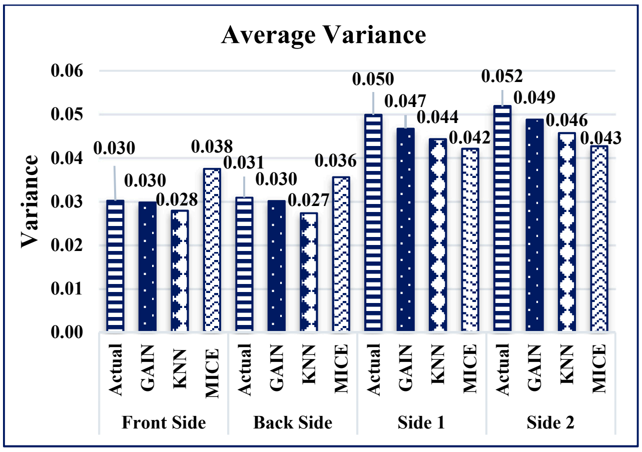


Figure 18. Average variance comparison.

The comparison of the original pressure values and the imputed pressure values of the MICE, KNN, and GAIN models at different pressure tap locations for all four sides of the building is depicted in Figure 19. It is clearly shown that the GAIN predicted average produces closer imputed values with the actual values when compared with MICE and KNN.

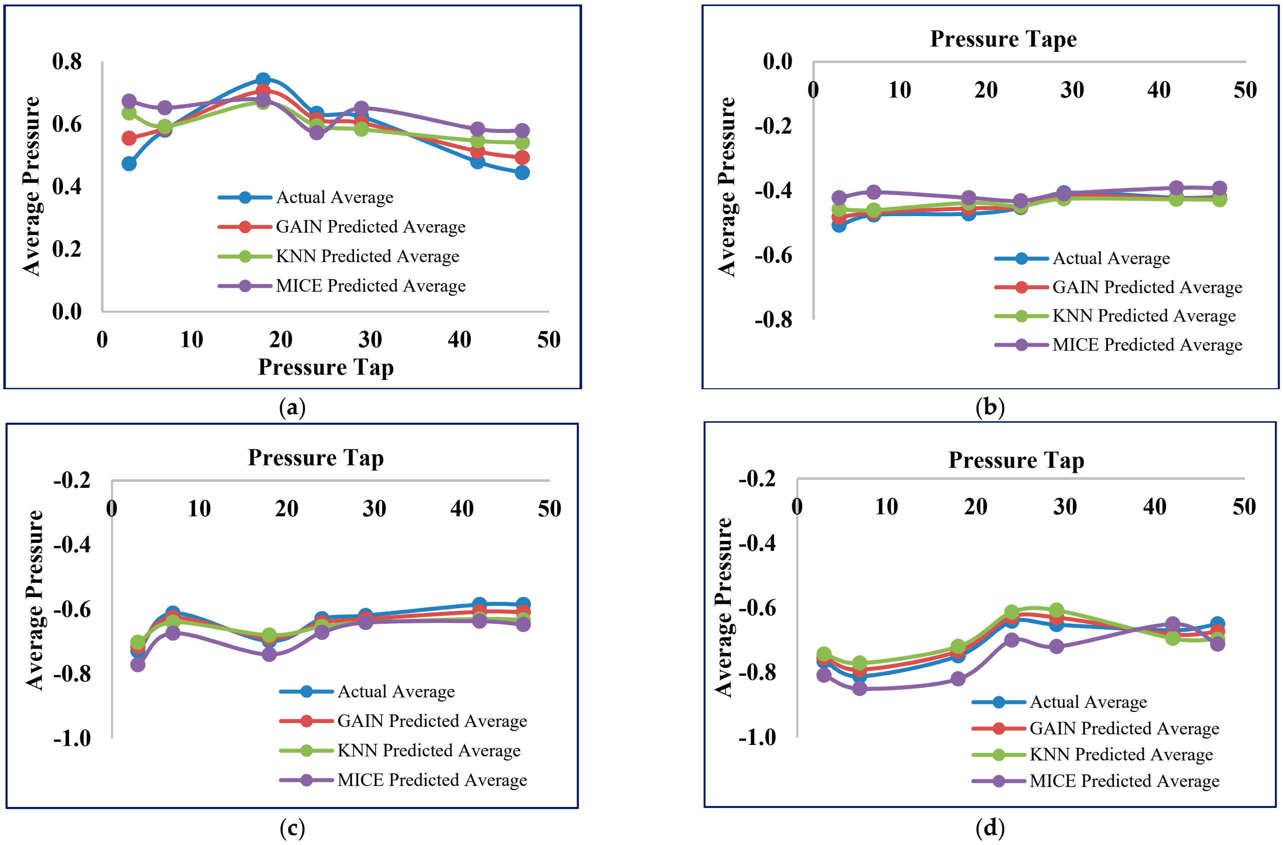


Figure 19. Average pressure value comparison on all sides. (**a**) Average pressure—front side; (**b**) Average pressure—back side; (**c**) Average pressure—side 1; (**d**) Average pressure—side 2.

The regression analysis values of the test and predicted mean C_P values of KNN, MICE, and GAIN are shown in Figures 20–22. It is observed that the test and predicted mean C_P values exhibit the best fit for the GAIN model. The MICE model performs overfitting, and the predicted mean C_P values are scattered. The KNN algorithm can manage the predictions with minimum deviations. The GAIN performance clearly shows that the test and predicted mean C_P values are close to each other with a minimal deviation in comparison with the KNN and MICE algorithms. The performance analyses depicts that the GAIN algorithm can impute the pressure data more accurately than the other ML models.

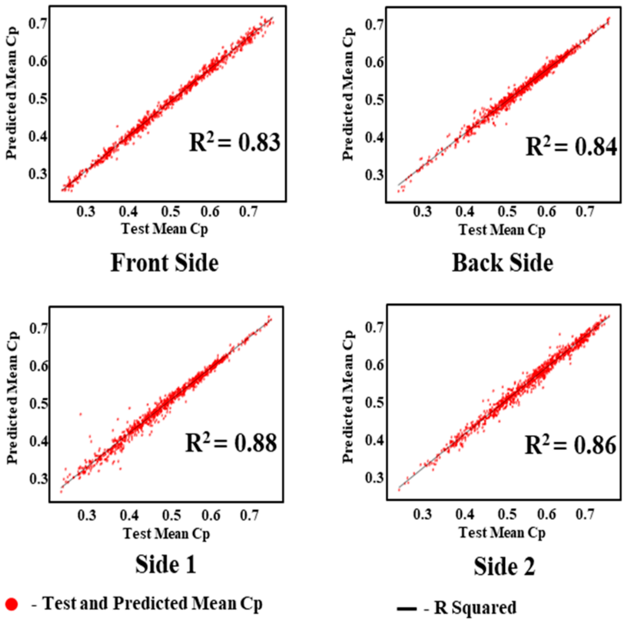


Figure 20. Predicted mean pressure coefficient by MICE.

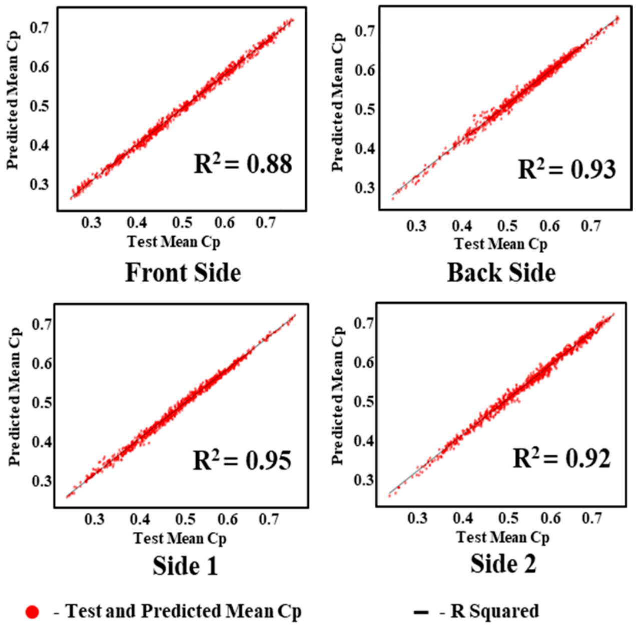


Figure 21. Predicted mean pressure coefficient by KNN.

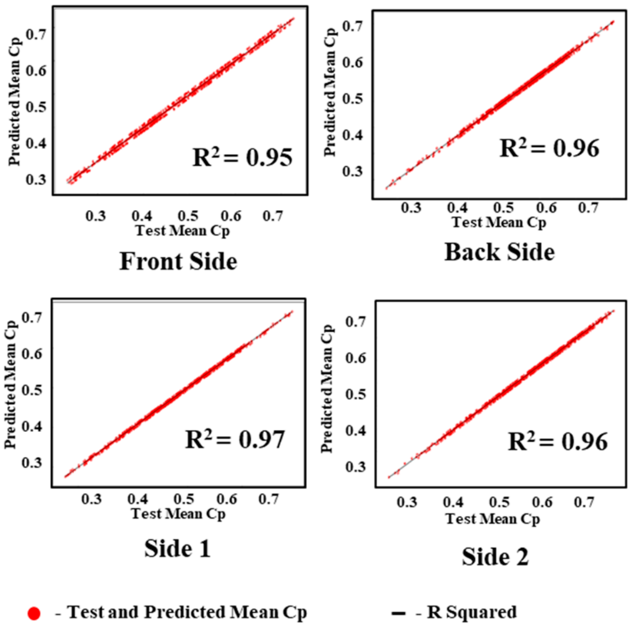


Figure 22. Predicted mean pressure coefficient by GAIN.

6. Conclusions

Predicting the wind pressure coefficient values at the missing locations plays a vital role in the continuous monitoring of the wind environment around structures. Inevitable data losses and corruption in structural monitoring sensors have tremendous effects on structural health research experiments. In this study, missing wind-induced pressure data at various time intervals of a square-shaped building were predicted by constructing an intelligent data prediction model (IDPM) with ML algorithms. This proposed IDPM model could predict the missing values that occurred owing to faulty sensors. In addition, the IDPM model predicted the wind pressure coefficients on a building by considering the nearby pressure tap values without using multiple pressure taps. A prototype of a proposed building was constructed to conduct the wind tunnel test for data generation. The ML algorithms, namely KNN, MICE, and GAIN, were implemented and validated for predicting missing values. Performance analysis of the proposed ML models indicated that GAIN was more accurate in imputing the missing pressure coefficient values compared with KNN and MICE. GAIN provided a minimum AMSE of 0.01 and a maximum ARSE of 0.96 in comparison with other models. In addition, GAIN exhibited excellent performance with minimum standard deviation and minimum variance between actual and imputed pressure coefficients.

The proposed work applied the best ML algorithms to predict the missing wind pressure values. This study provides a pathway to predict the wind-induced pressure coefficient values of faulty and malfunctioning pressure taps. In addition, it is viable and can predict the wind response at a particular locations on the structure even in the absence of pressure taps. Therefore, the future scope of this work is to minimize the usage of pressure sensors on structures for continuous monitoring of the wind environment.

Author Contributions: Conceptualization, B.K. and N.Y.; methodology, N.Y. and K.R.S.P.; software, N.Y. and K.R.S.P.; validation, B.K., N.Y. and G.H.; formal analysis, G.H.; investigation, D.-E.L.; resources, B.K.; data curation, D.-E.L.; writing—original draft preparation, N.Y.; writing—review and editing, D.-E.L.; visualization, K.R.S.P.; supervision, B.K.; project administration, N.Y.; funding acquisition, B.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT), grant number NRF-2018R1A5A1025137).

Institutional Review Board Statement: Not Applicable.

Informed Consent Statement: Not Applicable.

Data Availability Statement: Not Applicable.

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

References

1. Dongmei, H.; Shiqing, H.; Xuhui, H.; Xue, Z. Prediction of wind loads on high-rise building using a BP neural network combined with POD. *J. Wind. Eng. Ind. Aerodyn.* **2017**, *170*, 1–17. [CrossRef]
2. Kwok, K.C.S.; Hitchcock, P.A.; Burton, M.D. Perception of vibration and occupant comfort in wind-excited tall buildings. *J. Wind. Eng. Ind. Aerodyn.* **2009**, *97*, 368–380. [CrossRef]
3. Tse, K.; Zhang, X.; Weerasuriya, A.; Li, S.; Kwok, K.; Mak, C.M.; Niu, J. Adopting 'lift-up' building design to improve the surrounding pedestrian-level wind environment. *Build. Environ.* **2017**, *117*, 154–165. [CrossRef]
4. Zhang, X.; Tse, K.; Weerasuriya, A.; Li, S.; Kwok, K.; Mak, C.M.; Niu, J.; Lin, Z. Evaluation of pedestrian wind comfort near 'lift-up' buildings with different aspect ratios and central core modifications. *Build. Environ.* **2017**, *124*, 245–257. [CrossRef]
5. Kim, B.; Tse, K.; Yoshida, A.; Chen, Z.; Van Phuc, P.; Park, H.S. Investigation of flow visualization around linked tall buildings with circular sections. *Build. Environ.* **2019**, *153*, 60–76. [CrossRef]
6. Weerasuriya, A.; Tse, K.; Zhang, X.; Li, S. A wind tunnel study of effects of twisted wind flows on the pedestrian-level wind field in an urban environment. *Build. Environ.* **2018**, *128*, 225–235. [CrossRef]
7. Aly, A.M. Pressure integration technique for predicting wind-induced response in high-rise buildings. *Alex. Eng. J.* **2013**, *52*, 717–731. [CrossRef]
8. Weerasuriya, A.; Tse, K.; Zhang, X.; Kwok, K. Integrating twisted wind profiles to Air Ventilation Assessment (AVA): The current status. *Build. Environ.* **2018**, *135*, 297–307. [CrossRef]
9. Beranek, W.J. Wind environment around single buildings of rectangular shape. *Heron* **1984**, *29*, 3–31.
10. Weerasuriya, A.; Hu, Z.; Zhang, X.; Tse, K.; Li, S.; Chan, P. New inflow boundary conditions for modeling twisted wind profiles in CFD simulation for evaluating the pedestrian-level wind field near an isolated building. *Build. Environ.* **2018**, *132*, 303–318. [CrossRef]
11. Hang, J.; Li, Y. Ventilation strategy and air change rates in idealized high-rise compact urban areas. *Build. Environ.* **2010**, *45*, 2754–2767. [CrossRef]
12. Ng, E. Policies and technical guidelines for urban planning of high-density citiesair ventilation assessment (AVA) of Hong Kong. *Build. Environ.* **2009**, *44*, 1478–1488. [CrossRef] [PubMed]
13. Yim, S.; Fung, J.; Lau, A.; Kot, S. Air ventilation impacts of the "wall effect" resulting from the alignment of high-rise buildings. *Atmos. Environ.* **2009**, *43*, 4982–4994. [CrossRef]
14. Tsang, C.; Kwok, K.; Hitchcock, P. Wind tunnel study of pedestrian level wind environment around tall buildings: Effects of building dimensions, separation and podium. *Build. Environ.* **2012**, *49*, 167–181. [CrossRef]
15. Niu, Y.; Fritzen, C.-P.; Jung, H.; Buethe, I.; Ni, Y.-Q.; Wang, Y.-W. Online Simultaneous Reconstruction of Wind Load and Structural Responses-Theory and Application to Canton Tower. *Comput. Civ. Infrastruct. Eng.* **2015**, *30*, 666–681. [CrossRef]
16. Graf, K.; Müller, O. Photogrammetric Investigation of the Flying Shape of Spinnakers in a Twisted Flow Wind Tunnel. In Proceedings of the 19th Chesapeake Sailing Yacht Symposium, Annapolis, MD, USA, 20–21 March 2009.
17. Kato, Y.; Kanda, M. Development of a modified hybrid aerodynamic vibration technique for simulating aerodynamic vibration of structures in a wind tunnel. *J. Wind. Eng. Ind. Aerodyn.* **2014**, *135*, 10–21. [CrossRef]
18. Lam, K.M.; Li, A. Mode shape correction for wind-induced dynamic responses of tall buildings using time-domain computation and wind tunnel tests. *J. Sound Vib.* **2009**, *322*, 740–755. [CrossRef]
19. Tse, K.; Hitchcock, P.; Kwok, K. Mode shape linearization for HFBB analysis of wind-excited complex tall buildings. *Eng. Struct.* **2009**, *31*, 675–685. [CrossRef]
20. Ho, T.C.E.; Surry, D. *Factory Mutual-High Resolution Pressure Measurements on Roof Panels*; Boundary Layer Wind Tunnel Laboratory Report; The University of Western Ontario: London, ON, Canada, 2000.
21. Park, H.S.; Sohn, H.G.; Kim, I.S.; Park, J.H. Application of GPS to monitoring of wind-induced responses of high-rise buildings. *Struct. Des. Tall Spéc. Build.* **2008**, *17*, 117–132. [CrossRef]
22. Amezquita-Sanchez, J.P.; Park, H.S.; Adeli, H. A novel methodology for modal parameters identification of large smart structures using MUSIC, empirical wavelet transform, and Hilbert transform. *Eng. Struct.* **2017**, *147*, 148–159. [CrossRef]
23. Xia, Y.; Ni, Y.-Q.; Zhang, P.; Liao, W.-Y.; Ko, J.-M. Stress Development of a Supertall Structure during Construction: Field Monitoring and Numerical Analysis. *Comput. Civ. Infrastruct. Eng.* **2011**, *26*, 542–559. [CrossRef]
24. Amezquita-Sanchez, J.P.; Adeli, H. Signal Processing Techniques for Vibration-Based Health Monitoring of Smart Structures. *Arch. Comput. Methods Eng.* **2016**, *23*, 1–15. [CrossRef]
25. Gao, Y.; Mosalam, K.M. Deep Transfer Learning for Image-Based Structural Damage Recognition. *Comput. Civ. Infrastruct. Eng.* **2018**, *33*, 748–768. [CrossRef]
26. Rafiei, M.H.; Adeli, H. A novel unsupervised deep learning model for global and local health condition assessment of structures. *Eng. Struct.* **2018**, *156*, 598–607. [CrossRef]

27. Tsogka, C.; Daskalakis, E.; Comanducci, G.; Ubertini, F. The Stretching Method for Vibration-Based Structural Health Monitoring of Civil Structures. *Comput. Civ. Infrastruct. Eng.* **2017**, *32*, 288–303. [CrossRef]
28. Ni, Y.Q.; Xia, Y.; Liao, W.Y.; Ko, J.M. Technology innovation in developing the structural health monitoring system for Guangzhou New TV Tower. *Struct. Control. Health Monit.* **2009**, *16*, 73–98. [CrossRef]
29. Yuvaraj, N.; Kim, B.; Preethaa, S.; Transfer, K.R. Learning based Real-Time Crack Detection Using Unmanned Arial System. *Int. J. High Rise Build.* **2021**, *9*, 4. [CrossRef]
30. Hu, G.; Kwok, K. Predicting wind pressures around circular cylinders using machine learning techniques. *J. Wind. Eng. Ind. Aerodyn.* **2020**, *198*, 104099. [CrossRef]
31. Lee, K.J.; Carlin, J.B. Multiple Imputation for Missing Data: Fully Conditional Specification Versus Multivariate Normal Imputation. *Am. J. Epidemiol.* **2010**, *171*, 624–632. [CrossRef]
32. Zhang, Z.; Luo, Y. Restoring method for missing data of spatial structural stress monitoring based on correlation. *Mech. Syst. Signal Process.* **2017**, *91*, 266–277. [CrossRef]
33. Yang, Y.; Nagarajaiah, S. Harnessing data structure for recovery of randomly missing structural vibration responses time history: Sparse representation versus low-rank structure. *Mech. Syst. Signal Process.* **2016**, *74*, 165–182. [CrossRef]
34. Yu, Y.; Han, F.; Bao, Y.; Ou, J. A Study on Data Loss Compensation of WiFi-Based Wireless Sensor Networks for Structural Health Monitoring. *IEEE Sens. J.* **2015**, *16*, 3811–3818. [CrossRef]
35. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.L.; Courville, A.; Bengio, Y. Generative adversarial nets. *arXiv* **2014**, arXiv:1406.2661v1.
36. Preethaa, K.R.S.; Sabari, A. Intelligent video analysis for enhanced pedestrian detection by hybrid metaheuristic approach. *Soft Comput.* **2020**, *24*, 12303–12311. [CrossRef]
37. Lichman, M. UCI Machine Learning Repository. 2013. Available online: http://archive.ics.uci.edu/ml (accessed on 18 January 2021).
38. Bre, F.; Gimenez, J.M.; Fachinotti, V.D. Prediction of wind pressure coefficients on building surfaces using artificial neural networks. *Energy Build.* **2018**, *158*, 1429–1441. [CrossRef]
39. Chen, Y.; Kopp, G.; Surry, D. Prediction of pressure coefficients on roofs of low buildings using artificial neural networks. *J. Wind. Eng. Ind. Aerodyn.* **2003**, *91*, 423–441. [CrossRef]
40. Fu, J.; Liang, S.; Li, Q. Prediction of wind-induced pressures on a large gymnasium roof using artificial neural networks. *Comput. Struct.* **2007**, *85*, 179–192. [CrossRef]
41. Thrun, M.C.; Ultsch, A.; Breuer, L. Explainable AI Framework for Multivariate Hydrochemical Time Series. *Mach. Learn. Knowl. Extr.* **2021**, *3*, 170–205. [CrossRef]
42. Oh, B.K.; Glisic, B.; Kim, Y.; Park, H.S. Convolutional neural network-based wind-induced response estimation model for tall buildings. *Comput. Civ. Infrastruct. Eng.* **2019**, *34*, 843–858. [CrossRef]
43. Yang, X.; Zhang, Y.; Lv, W.; Wang, D. Image recognition of wind turbine blade damage based on a deep learning model with transfer learning and an ensemble learning classifier. *Renew. Energy* **2021**, *163*, 386–397. [CrossRef]
44. Zhang, C.; Patras, P.; Haddadi, H. Deep Learning in Mobile and Wireless Networking: A Survey. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 2224–2287. [CrossRef]
45. Kim, B.; Yuvaraj, N.; Preethaa, K.R.S.; Pandian, R.A. Surface crack detection using deep learning with shallow CNN architecture for enhanced computation. *Neural Comput. Appl.* **2021**, 1–17. [CrossRef]
46. Dong, M.; Wu, H.; Hu, H.; Azzam, R.; Zhang, L.; Zheng, Z.; Gong, X. Deformation Prediction of Unstable Slopes Based on Real-Time Monitoring and DeepAR Model. *Sensors* **2020**, *21*, 14. [CrossRef] [PubMed]
47. Salinas, D.; Flunkert, V.; Gasthaus, J.; Januschowski, T. DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *Int. J. Forecast.* **2020**, *36*, 1181–1191. [CrossRef]
48. Kim, B.; Yuvaraj, N.; Preethaa, K.R.S.; Santhosh, R.; Sabari, A. Enhanced pedestrian detection using optimized deep convolution neural network for smart building surveillance. *Soft Comput.* **2020**, *24*, 17081–17092. [CrossRef]
49. Lee, S.-J.; Yoon, H.-K. Discontinuity Predictions of Porosity and Hydraulic Conductivity Based on Electrical Resistivity in Slopes through Deep Learning Algorithms. *Sensors* **2021**, *21*, 1412. [CrossRef] [PubMed]
50. Huang, F.; Zhang, J.; Zhou, C.; Wang, Y.; Huang, J.; Zhu, L. A deep learning algorithm using a fully connected sparse autoencoder neural network for landslide susceptibility prediction. *Landslides* **2020**, *17*, 217–229. [CrossRef]
51. Ahmed, R.; El Sayed, M.; Gadsden, S.A.; Tjong, J.; Habibi, S. Automotive Internal-Combustion-Engine Fault Detection and Classification Using Artificial Neural Network Techniques. *IEEE Trans. Veh. Technol.* **2014**, *64*, 21–33. [CrossRef]
52. Royston, P.; Carlin, J.B.; White, I.R. Multiple Imputation of Missing Values: New Features for Mim. *Stata J. Promot. Commun. Stat. Stata* **2009**, *9*, 252–264. [CrossRef]
53. Yuvaraj, N.; SriPreethaa, K.R. Diabetes prediction in healthcare systems using machine learning algorithms on Hadoop cluster. *Clust. Comput.* **2017**, *22*, 1–9. [CrossRef]
54. Thakker, D.; Mishra, B.K.; Abdullatif, A.; Mazumdar, S.; Simpson, S. Explainable Artificial Intelligence for Developing Smart Cities Solutions. *Smart Cities* **2020**, *3*, 1353–1382. [CrossRef]
55. Jin, X.; Cheng, P.; Chen, W.-L.; Wen-Li, C. Prediction model of velocity field around circular cylinder over various Reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder. *Phys. Fluids* **2018**, *30*, 047105. [CrossRef]

56. Farahmandpour, Z.; Seyedmahmoudian, M.; Stojcevski, A.; Moser, I.; Schneider, J.-G. Cognitive Service Virtualisation: A New Machine Learning-Based Virtualisation to Generate Numeric Values. *Sensors* **2020**, *20*, 5664. [CrossRef]
57. Park, H.; Son, J.-H. Machine Learning Techniques for THz Imaging and Time-Domain Spectroscopy. *Sensors* **2021**, *21*, 1186. [CrossRef] [PubMed]
58. Dou, J.; Yunus, A.P.; Bui, D.T.; Merghadi, A.; Sahana, M.; Zhu, Z.; Chen, C.-W.; Han, Z.; Pham, B.T. Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan. *Landslides* **2020**, *17*, 641–658. [CrossRef]
59. Liang, Z.; Wang, C.; Khan, K.U.J. Application and comparison of different ensemble learning machines combining with a novel sampling strategy for shallow landslide susceptibility mapping. *Stoch. Environ. Res. Risk Assess.* **2020**, 1–14. [CrossRef]
60. Le, N.Q.K.; Do, D.T.; Chiu, F.-Y.; Yapp, E.K.Y.; Yeh, H.-Y.; Chen, C.-Y. XGBoost Improves Classification of MGMT Promoter Methylation Status in IDH1 Wildtype Glioblastoma. *J. Pers. Med.* **2020**, *10*, 128. [CrossRef] [PubMed]
61. White, I.R.; Royston, P.; Wood, A.M. Multiple imputation using chained equations: Issues and guidance for practice. *Stat. Med.* **2010**, *30*, 377–399. [CrossRef]
62. Enişer, H.F.; Sen, A. Virtualization of stateful services via machine learning. *Softw. Qual. J.* **2019**, *28*, 283–306. [CrossRef]
63. Loyola-Gonzalez, O.; Gutierrez-Rodriguez, A.E.; Medina-Perez, M.A.; Monroy, R.; Martinez-Trinidad, J.F.; Carrasco-Ochoa, J.A.; Garcia-Borroto, M. An Explainable Artificial Intelligence Model for Clustering Numerical Databases. *IEEE Access* **2020**, *8*, 52370–52384. [CrossRef]
64. Soman, S.S.; Zareipour, H.; Malik, O.P.; Mandal, P. A review of wind power and wind speed forecasting methods with different time horizons. In Proceedings of the North American Power Symposium 2010, Arlington, TX, USA, 26–28 September 2010; pp. 1–8.
65. Wen, Y.; Song, M.; Wang, J. A combined AR-kNN model for short-term wind speed forecasting. In Proceedings of the 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 12–14 December 2016; pp. 6342–6346.
66. Dasgupta, S.; Frost, N.; Moshkovitz, M.; Rashtchian, C. Explainable k-Means and k-Medians Clustering. In Proceedings of the 37th International Conference on Machine Learning, Vienna, Austria, 12–18 July 2020.
67. Thrun, M.C.; Ultsch, A. Using Projection-Based Clustering to Find Distance- and Density-Based Clusters in High-Dimensional Data. *J. Classif.* **2020**, 1–33. [CrossRef]
68. Hu, G.; Liu, L.; Tao, D.; Song, J.; Tse, K.; Kwok, K. Deep learning-based investigation of wind pressures on tall building under interference effects. *J. Wind. Eng. Ind. Aerodyn.* **2020**, *201*, 104138. [CrossRef]
69. Stivaktakis, R.; Tsagkatakis, G.; Tsakalides, P. Semantic Predictive Coding with Arbitrated Generative Adversarial Networks. *Mach. Learn. Knowl. Extr.* **2020**, *2*, 307–326. [CrossRef]
70. Allen, A.; Li, W. Generative Adversarial Denoising Autoencoder for Face Completion. 2016. Available online: https://www.cc.gatech.edu/~hays/7476/projects/Avery_Wenchen/ (accessed on 18 January 2021).
71. Chen, Z.; Fu, X.; Xu, Y.; Li, C.Y.; Kim, B.; Tse, K. A perspective on the aerodynamics and aeroelasticity of tapering: Partial reattachment. *J. Wind. Eng. Ind. Aerodyn.* **2021**, *212*, 104590. [CrossRef]
72. Chen, Z.; Tse, K.; Kwok, K.; Kareem, A.; Kim, B. Measurement of unsteady aerodynamic force on a galloping prism in a turbulent flow: A hybrid aeroelastic-pressure balance. *J. Fluids Struct.* **2021**, *102*, 103232. [CrossRef]
73. Tse, K.T.; Hu, G.; Song, J.; Park, H.S.; Kim, B. Effects of corner modifications on wind loads and local pressures on walls of tall buildings. *Build. Simul.* **2020**, 1–18. [CrossRef]
74. Kim, B.; Tse, K.; Chen, Z.; Park, H.S. Multi-objective optimization of a structural link for a linked tall building system. *J. Build. Eng.* **2020**, *31*, 101382. [CrossRef]
75. Chen, Z.; Xu, Y.; Tse, K.; Hu, L.; Kwok, K.; Kim, B. Non-wind-induced nonlinear damping and stiffness on slender prisms: A forced vibration-pressure balance. *Eng. Struct.* **2020**, *207*, 110107. [CrossRef]
76. Chen, Z.; Tse, K.; Kwok, K.; Kim, B.; Kareem, A. Modelling unsteady self-excited wind force on slender prisms in a turbulent flow. *Eng. Struct.* **2020**, *202*, 109855. [CrossRef]
77. Kim, B.; Tse, K.; Yoshida, A.; Tamura, Y.; Chen, Z.; Van Phuc, P.; Park, H.S. Statistical analysis of wind-induced pressure fields and PIV measurements on two buildings. *J. Wind. Eng. Ind. Aerodyn.* **2019**, *188*, 161–174. [CrossRef]
78. Kim, B.; Tse, K. POD analysis of aerodynamic correlations and wind-induced responses of two tall linked buildings. *Eng. Struct.* **2018**, *176*, 369–384. [CrossRef]
79. Kim, B.; Tse, K.; Tamura, Y. POD analysis for aerodynamic characteristics of tall linked buildings. *J. Wind. Eng. Ind. Aerodyn.* **2018**, *181*, 126–140. [CrossRef]