buildings

Article

Interpretation of Machine-Learning-Based (Black-box) Wind Pressure Predictions for Low-Rise Gable-Roofed Buildings Using Shapley Additive Explanations (SHAP)

Pasindu Meddage [1,*], Imesh Ekanayake [2], Udara Sachinthana Perera [3], Hazi Md. Azamathulla [4], Md Azlin Md Said [5] and Upaka Rathnayake [6]

1 Department of Civil and Environmental Engineering, Faculty of Engineering, University of Ruhuna, Hapugala 80042, Sri Lanka
2 Department of Computer Engineering, Faculty of Engineering, University of Peradeniya, Pereadeniya 20400, Sri Lanka; imeshuek@eng.pdn.ac.lk
3 Department of Technology, Kothalawala Defense University, Rathmalana 10390, Sri Lanka; pereraus@kdu.ac.lk
4 Department of Civil and Environmental Engineering, The Faculty of Engineering, The University of West Indies, St. Augustine 32080, Trinidad and Tobago; hazi.azamathulla@sta.uwi.edu
5 School of Civil Engineering, Universiti Sains Malaysia, Nibong Tebal 14300, Penang, Malaysia; ceazlin@usm.my
6 Department of Civil Engineering, Faculty of Engineering, Sri Lanka Institute of Information Technology, Malabe 10115, Sri Lanka; upaka.r@sliit.lk
* Correspondence: meddage.p@cee.ruh.ac.lk or pasindu95dm@gmail.com

Abstract: Conventional methods of estimating pressure coefficients of buildings retain time and cost constraints. Recently, machine learning (ML) has been successfully established to predict wind pressure coefficients. However, regardless of the accuracy, ML models are incompetent in providing end-users' confidence as a result of the black-box nature of predictions. In this study, we employed tree-based regression models (Decision Tree, XGBoost, Extra-tree, LightGBM) to predict surface-averaged mean pressure coefficient ($C_{p,mean}$), fluctuation pressure coefficient ($C_{p,rms}$), and peak pressure coefficient ($C_{p,peak}$) of low-rise gable-roofed buildings. The accuracy of models was verified using Tokyo Polytechnic University (TPU) wind tunnel data. Subsequently, we used Shapley Additive Explanations (SHAP) to explain the black-box nature of the ML predictions. The comparison revealed that tree-based models are efficient and accurate in wind-predicting pressure coefficients. Interestingly, SHAP provided human-comprehensible explanations for the interaction of variables, the importance of features towards the outcome, and the underlying reasoning behind the predictions. Moreover, SHAP confirmed that tree-based predictions adhere to the flow physics of wind engineering, advancing the fidelity of ML-based predictions.

Keywords: explainable machine learning; pressure coefficient; shapley additive explanation; tree-based machine learning; gable-roofed low-rise building

check for updates

Citation: Meddage, P.; Ekanayake, I.; Perera, U.S.; Azamathulla, H.M.; Md Said, M.A.; Rathnayake, U. Interpretation of Machine-Learning-Based (Black-box) Wind Pressure Predictions for Low-Rise Gable-Roofed Buildings Using Shapley Additive Explanations (SHAP). *Buildings* **2022**, *12*, 734. https://doi.org/10.3390/buildings12060734

Academic Editor: David J. Edwards

Received: 9 May 2022
Accepted: 27 May 2022
Published: 29 May 2022

1. Introduction

Low-rise buildings are popular all over the world as they serve many sectors (e.g., residential, industrial, and commercial). Despite the ubiquitous presence, these buildings are constructed in various terrain profiles with numerous geometric configurations. From the wind engineering viewpoint, these buildings are located in the ABL with high turbulence intensities and steep velocity gradients. As a result, external wind pressure on low buildings is spatially and temporally heterogeneous. Either physical experiments (full scale or wind tunnel) or CFD simulations are often employed to investigate external pressure characteristics of low buildings. In the experiments, external pressure coefficients are recorded to investigate the external wind pressure on the building envelope. These pressure coefficients are required to estimate design loads and for natural ventilation calculations. External pressure coefficients indicate the building components that are subjected to large suction loads. However, these methods are resource-intensive and require time and significant expertise. Despite being computationally expensive, numerical simulations such as CFD are widely used for ABL modeling [1–5].

Alternatively, secondary methods including analytical models and parametric equations were developed in numerous ways. For example, DAD methods were introduced to ensure that structural designs are economical and safe [6–8]. Swami and Chandra [9] established a parametric equation to estimate $C_{p,mean}$ on walls of low-rise buildings. Subsequently, Muehleisen and Patrizi [10] developed a novel equation to improve the accuracy of the same predictions. However, both equations are permitted only for rectangular-shaped buildings and failed to predict the pressure coefficients of the roof. In consequence, the research community explored advanced data-driven methods and algorithms as an alternative.

Accordingly, a noticeable surge in ML-based modeling is observed in engineering applications. Foremost, these approaches can be categorized into supervised, semi-supervised, and unsupervised ML, based on data-labeling [11,12]. Regardless of the complexity of the approach, various ML algorithms are successively employed in structural engineering applications as evident from the comprehensive review provided by Sun et al. [13]. In addition, many recent studies combined ML techniques with building engineering [14–26]. For example, Wu et al. [27] used tree-based ensemble classifiers to predict hazardous materials in buildings. Fan and Ding [28] developed a scorecard for sick building syndrome using ML. Ji et al. [29] used ML to investigate life cycle cost and life cycle assessment of buildings. In their proposed work, deep learning models showed superior performance. Overall models achieved an R^2 between 0.932 and 0.955. Yang et al. [30] reported that supervised classification methods can be effectively used for building climate zoning. Olu-Ajyal et al. [31] argued that deep neural networks perform better compared to remaining machine learning models (Artificial Neural Network, Gradient Boosting, Decision Tree, Random Forest, Support Vector Machine, Stacking, K-Nearest Neighbour, and Linear Regression) in predicting building energy consumption in residential buildings.

Given its ability to find non-linear and complex functions, Kareem [32] stated that ML algorithms are competent with physical experiments and numerical simulations when predicting wind loads on buildings. Numerous studies have been conducted over the years on investigating ML applications in wind engineering [14,33–44]. For example, Bre et al. [33] and Chen et al. [35] developed ANN models to predict wind pressure coefficients of low-rise buildings. Lin et al. [45] suggested an ML-based method to estimate the cross-wind effect of tall buildings. They used lightGBM regression to predict crosswind spectrum, and it achieved an acceptable accuracy, complying with experimental results. Kim et al. [46] proposed clustering algorithms to identify wind pressure patterns. They noticed that the clustering algorithm reasonably captures the pressure patterns better than independent component analysis or principal component analysis. Dongmei et al. [36], and Duan et al. [37] used ANNs to predict wind loads and wind speeds, respectively. Hu and Kwok [40] proposed data-driven (tree-based) models to predict wind pressure around circular cylinders. Accordingly, all tree-based models achieved $R^2 > 0.96$ for predicting mean pressure coefficients and $R^2 > 0.97$ for predicting fluctuation (rms) pressure coefficients. Hu et al. [47] investigated the applicability of machine learning models (Decision Tree, Random Forest, XGBoost, GAN) to predict wind pressure of tall buildings in the presence of interference-effect. The proposed GAN showcased superior performance in contrast to the remaining models. The GAN achieved $R^2 = 0.988$ for mean pressure coefficient predictions and $R^2 = 0.924$ for rms pressure coefficient predictions. Wang et al. [48] integrated LSTM, Random Forests, and gaussian process regression to predict short-term wind gusts. They argued that the proposed ensemble method is more accurate than employing individual models. Tian et al. [49] introduced a novel approach using deep neural networks (DNN) to predict wind pressure coefficients of low-rise gable-roofed buildings. The method achieved

R = 0.9993 for mean pressure predictions and R = 0.9964 for peak pressure predictions. In addition, Mallick et al. [50] extended experiments from regular-shaped buildings to unconventional configurations while combining gene expression programming and ANNs. The proposed equation was intended to predict surface average pressure coefficients of a C-shaped building. Na and Son [51] predicted the atmospheric motion vector around typhoons using GAN, and the model achieved acceptable accuracy ($R^2 > 0.97$). Lamberti and Gorle [44] reported that ML can balance computation cost and accuracy in predicting complex turbulent flow quantities. Recent work done by Arul et al. [52] proposed that shapelet transformation is an effective way to represent the time series of wind. They observed that shapelet transformation is useful in identifying a wide variety of thunderstorms that cannot be detected using conventional gust factor-based methods. Interestingly, these ML models were precise and less time-consuming in contrast to conventional methods.

However, all related studies failed to explain the black-box nature of ML predictions and their underlying reasoning. ML models estimate complex functions while no insights on its inner-working methodology are provided to the end-user. Regardless of its superior performance, model transparency is inferior compared to traditional approaches. The absence of such knowledge makes implementing ML in wind engineering more difficult. For instance, end-users are confident in physical and CFD modeling as a result of the transparency of the process. Hence, ML models should be explainable in order to obtain the trust of domain experts and end-users.

As an emerging branch, explainable artificial intelligence expects to overcome the black-box nature of ML predictions. It provides insights into how an ML model performs the prediction and the causality of prediction. It works as the human agent of the ML model. Therefore, it is highly recommended among multi-stakeholders [53]. Explainable ML makes regular ML predictions interpretable to the end-user. In addition, end-users become aware of the form of relationship that exists between features. Hence, it is convenient for the user to understand the importance of the features, and the dependency among various features as a model in whole or instances [54–56]. With such advanced features, explainable ML turns *black-box* models into *glass-box* models. One such attempt has been recently given to predict the external pressure coefficients of a low-rise flat-roofed building surrounded by similar buildings by using explainable machine learning [57]. They argued that explainable models advance the ML models by revealing the causality of predictions. However, that study focused on the effect of surrounding buildings, whereas the present study focused on isolated gable-roofed low-rise buildings and the effect of its geometric parameters.

Therefore, the main objective is to predict surface-averaged pressure coefficients of low-rise gable-roofed buildings using explainable ML. We argue that explainable ML improves the transparency of the pressure coefficient predictions while exposing the inner-working of the model. On the other hand, the explainable ML is imperative to cross-validate the predictions using theoretical/experimental knowledge on low-rise gable-roofed buildings. Finally, the study demonstrates that using explainable ML to predict wind pressure coefficient does not affect model complexity or accuracy but rather enhances the fidelity by improving end users' trust in predictions.

Because the wind engineering community is new to explainable ML, Section 2 provides a brief on the concept and the explanation we used (Shapley additive explanations (SHAP) [58]). Section 3 provides the background of tree-based ML models used in this study. Sections 4 and 5 provide the performance analysis and the methodology of the study, respectively. Results and discussion are provided in Sections 6 and 7 concludes the paper. Section 8 discusses the limitations and future work of the research study.

2. Explainable ML

Explainable ML does not own a standard definition. Explainable methods can be categorized into unique approaches namely, intrinsic and post hoc. For example, models whose structure is simple are self-explainable (intrinsic) (e.g., linear regression, Decision

Trees at lower tree depths). However, when an ML model is complex, a post hoc explanation (explainable ML) is required to elucidate the predictions

Several models are already available that provide post hoc explanations. These include models such as DeepLIFT [59], LIME [60], RISE [61], and SHAP [58]. LIME and SHAP have been widely used in ML applications. However, Moradi and Samwald [62] state LIME creates dummy instances considering the neighborhood of an instance by approximating a linear behavior. Therefore, LIME interpretations do not reflect actual feature values. In this study, we used SHAP to investigate the influence of each parameter on respective prediction, its quantification, and convince underlying reasoning behind instances.

Shapley Additive Explanations (SHAP)

SHAP provides both local and global explanations of an ML model [63]. The value assigned by SHAP can be used as a unified measure of feature importance. SHAP follows core concepts in game theory when computing the feature importance. For instance, "games" can be referred to as model predictions, and features inside the ML model are represented by "players". Simply, SHAP quantifies the contribution of each player to the game [58]. Global interpretation provided by SHAP measures how a patient attribute contributes to a prediction. Liang et al. [54] provided a detailed classification of explanation models, in which SHAP is a data-driven and perturbation-based method. Therefore, it relies on input parameters and does not require understanding the operation sequence of the ML model.

Perturbation works by masking several regions of the data samples to create disturbances. Subsequently, a disordered sample will result in another set of predictions that can be compared with the original predictions. Lundberg and Lee [58] introduced unique SHAP versions (e.g., DeepSHAP, Kernel SHAP, LinearSHAP, and TreeSHAP) for specific ML model categories. For example, the current study employed TreeSHAP for ML predictions. There, a linear explanatory model is used and the corresponding Shapley value is calculated using Equation (1),

$$f(y') = \phi_o + \sum_{i=1}^{N} \phi_i {y'}_i \tag{1}$$

where f denotes the explanation model whereas $y' \in \{0,1\}^N$ denotes the simplified features of the coalition vector. N and $\phi \in \mathbb{R}$ denote the maximum size of the coalition and the feature attribution, respectively. Lundberg and Lee [58] specified Equations (2) and (3) to compute feature attribution,

$$\phi_i = \sum_{S\ \{1,\ldots,p\}\backslash\{i\}} \frac{|S|!(p-|S|-1)!}{p!}[g_x(S \cup \{i\}) - g_x(S)] \tag{2}$$

$$\text{where; } g_x(S) = E[g(x)|x_S] \tag{3}$$

In Equation (2), S represents a subset of the features (input) and x is the vector of feature values of the instance to be interpreted. Thus, the Shapley value is denoted through a value function (g_x). Here, p symbolizes the number of features and $g_x(S)$ is the prediction obtained from features in S. $E[g(x)|x_K]$ represents the expected value of the function on subset S. In addition, this study employs Scikit-learn (Sklearn), NumPy, matplot, pandas, and Shap libraries for the implementation.

Sklearn is the most efficient and robust library that is used for machine learning applications in python. Sklearn allows using numerous machine learning tools, including regression, classification, clustering, and dimensionality reduction. In addition, this library was written mostly using python over Numpy, Scipy, and Matplotlib.

3. Tree-Based ML Algorithms

We proposed four tree-based ML models for the present study. All four models are Decision Tree-based models. Tree-based models follow a deterministic process in decision-making. Patel and Prajapati [64] reported that a Decision Tree mimics the human thinking process. Despite the complexity that grows with tree-depth, the decision-tree structure is self-explainable. Moreover, tree-based models work efficiently for structured data.

3.1. Decision Tree Regressor

Decision Trees serve for both regression and classification problems [65–67]. The working principle of a Decision Tree is to split a comprehensive task into several simplified versions. Evolved structure of the Decision Tree is hierarchical from roots to end leaves and generates a model based on logical postulations that can be subsequently employed to predict new data.

Recursive breakdown and multiple regression are performed to train a decision-tree regression model. Until end criteria are met, splitting takes place at each interior node, starting from the root node of a Decision Tree. Primarily, each leaf node of the tree represents a simple regression model. Trimming low information gain branches (pruning) is applied to enhance the generalization of the model. Furthermore, the Decision Tree compacts each possible alternative toward the conclusion.

Per each partition, the response variable y is separated into two sets, namely, S_1 and S_2. Subsequently, the Decision Tree examines a predictor variable x concerning the split threshold,

$$SSE = \sum_{i \in S_1} (y_i - \overline{y_1})^2 + \sum_{i \in S_2} (y_i - \overline{y_2})^2 \tag{4}$$

where $\overline{y_1}$ and $\overline{y_2}$ are the average values of the response for each set. The tree generally intends to minimize the sum of squared error (SSE) (refer to Equation (4)) for each split. The tree starts growing with recursive splits and split thresholds. The terminal node represents the average of y values of samples collected within the node.

3.2. XGBoost Regressor

XGBoost is a gradient boosting implementation that boosts weak learners. It is more often preferred due to its fast execution process [68]. Chakraborty and Elzarka [69], Mo et al. [70], and Xia et al. [69–71] have successfully used XGBoost in their respective studies. The regressor itself can handle overfitting or underfitting issues, and regularizations are better than Decision Tree and Random Forest algorithms.

The regularization function of XGBoost assists to control the complexity of the model and to select predictive functions. The objective function is defined as a regularization term together with a loss function and it is optimized using gradient descent. XGBoost provides column subsampling compared to conventional Gradient Boosting [72]. At each level, the tree structure is formed by estimating leaf score, objective function, and regularization. Hence, it is difficult to evaluate all possible combinations at the same time. Subsequently, the tree structure will be re-employed in an iterative manner that helps to reduce computational expense. Information gain at each node is estimated during the splitting process and seeks the best splitting node until it reaches maximum depth. Later, the pruning process is executed in bottom-up order. The objective function in terms of loss and regularization can be expressed as given in Equation (5),

$$\text{objective} = \sum_{j=1}^{N} l(y_j, \hat{y}_j) + \sum_{j=1}^{K} \Omega(f_j) \tag{5}$$

The summation ($\sum_{j=1}^{N} l(y_j, \hat{y}_j)$) is the loss function that represents the difference between predicted $(\hat{y}_j)$ and actual values (y_j). The summation ($\sum_{j=1}^{K} \Omega(f_j)$) is the regularization term that decides the complexity of XGBoost model.

3.3. Extra-Tree Regressor

Extra-tree regressor is classified under ensemble methods of supervised learning [73–75]. It builds random trees whose primary structure is independent of the outcome of the learning sample. Okoro et al. [74] stated randomized trees are adopted for numerical inputs that improve precision and substantial reduction in computational burden.

Extra-tree regressor follows the classical top-down approach to fit disarrayed Decision Trees on subsamples of learning data. Random split points make the extra-tree regressor unique from other tree-based ensembles. Afterward, the tree grows using the whole learning sample. In particular, final predictions are done using a voting process in classification and regression. John et al. [76] and Seyyedattar et al. [77] described random subset features and detailed structure and their importance, respectively. Interestingly, explicit randomization can reduce similarities in contrast to weaker randomization of other methods.

For regression, relative variance reduction is used, and the score is expressed in Equation (6). The terms U_i and U_j represent subsets of cases from U that correspond to the outcome of a splits,

$$\mathrm{Score_R(s,\ U)} = \frac{\mathrm{var}\{y|U\} - \frac{|U_i|}{|U|}\mathrm{var}\{y|U_i\} - \frac{|U_j|}{|U|}\mathrm{var}\left\{y|U_j\right\}}{\mathrm{var}\{y|U\}} \tag{6}$$

3.4. LightGBM Regressor

LightGBM is an efficient gradient boosting structure formed on boosting and Decision Trees [78–80]. It uses histogram-based algorithms in contrast to XGBoost, to accelerate the training process, reducing memory consumption. Given that Decision Tree itself is a weak model, the accuracy of the segmentation point is not important. A coarse segmentation process can influence the regularization effect that avoids over-fitting. Leaf orientation at the downside can grow deeper Decision Trees leading to over-fitting situations. Hence, LightGBM tackles this issue by constricting maximum depth to the top of leaves. The model not only enables a higher efficiency but also handles non-linear relationships, ensuring a higher precision.

4. Model Performance

The following analysis was carried out to evaluate the performance of predictions obtained from machine learning models.

Performance Evaluation

Ebtehaj et al. [81] specified several indices to compare model efficiencies in terms of predictions. Hyperparameter optimization and model training were performed based on R^2. For validation predictions, we proposed R, MAE, and RMSE. R^2 expresses how well predictions fit actual data. R closer to +1 or −1 indicates a strong positive or negative correlation, respectively. MAE evaluates direct residual between wind tunnel and ML predictions while RMSE considers standard deviation of residuals, indicating how far predictions lie from experimental values. These four indices are mathematically formulated as shown in Equations (7)–(10),

$$R^2 = \frac{\sum_{i=1}^{N}\left(C_{p,ML} - \overline{C}_{p,WT}\right)^2}{\sum_{i=1}^{N}\left(C_{p,WT} - \overline{C}_{p,WT}\right)^2} = \frac{\text{Model Sum of Squares (MSS)}}{\text{Total Sum of Squares (TSS)}} \tag{7}$$

$$R = \frac{N\sum_{i=1}^{N}(C_{p,WT}.C_{p,ML}) - (\sum_{i=1}^{N} C_{p,WT}.\sum_{i=1}^{N} C_{p,ML})}{\sqrt{(N\sum_{i=1}^{N} C_{p,WT}{}^{2} - (\sum_{i=1}^{N} C_{p,WT})^{2})}.\sqrt{N\sum_{i=1}^{N} C_{p,ML}{}^{2} - (\sum_{i=1}^{N} C_{p,ML})^{2})}} \quad (8)$$

$$MAE = \frac{\sum_{i=1}^{N}|C_{p,WT} - C_{p,ML}|}{N} \quad (9)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(C_{p,WT} - C_{p,ML})^{2}}{N}} \quad (10)$$

N denotes the number of data samples and subscripts WT and ML refer to "Wind Tunnel" and "ML" predictions, respectively. C_p represents all three predicting variables, including mean, fluctuating (rms), and peak (minimum) components. $\overline{C}_P$ refers to the average value of the dependent variable in the validation data set.

5. Methodology

For each ML application, the quality and reliability of training data are very crucial. In terms of wind engineering applications, data sets should consist of a wide range of geometric configurations with multi parameters to obtain a generalized solution. The explanation becomes more comprehensible with a wide range of parameters.

Several wind tunnel databases are freely available for reference. NIST [82] and TPU [83] databases are well-known as reliable databases, especially for bluff body aerodynamics related to buildings. Both databases provide time histories of wind pressure including various geometric configurations of buildings. However, TPU provides wind tunnel data of gable-roofed low-buildings involving a wide range of roof pitches. Therefore, the TPU data set was selected for the application purpose.

Mean, fluctuating, and peak pressure coefficients on the corresponding surface were averaged and denoted as surface averaged pressure coefficients. Equations (11)–(14) were used to calculate the pressure coefficients from the time histories. It is noteworthy that the experiments had been conducted under fixed wind velocity. Therefore, the study cannot assess the effect of approaching wind velocity.

$$C_{p,mean} = \frac{\sum_{i=1}^{n} C_{p,i}}{n} \quad (11)$$

$$C_{p,rms} = \sqrt{\frac{\sum_{i=1}^{n}(C_{p,i} - C_{p,mean})^{2}}{n}} \quad (12)$$

$$C_{p,peak} = \min\{C_{p,i}\}\ ;\ (\text{sin gle worst minimum}) \quad (13)$$

$$\text{Surface avaraged } C_P = \frac{\sum_{i=1}^{n} C_P A_i}{\sum_{i=1}^{n} A_i};\ \text{for mean, rms, and peak} \quad (14)$$

where $C_{p,i}$ represents instantaneous external wind pressure, u_h denotes wind velocity at roof-mid height and ρ air density. A_i refers to the tributary area of a pressure tap and n is the total number of time steps).

Each model consists of four H/B ratios (0.25, 0.5, 0.75, and 1), three D/B ratios (1, 1.5, 2.5) and seven distinct wind directions (0°, 15°, 30°, 45°, 60°, 75°, 90°). Next, the roof pitch (α) of each model was altered eight times (5°, 10°, 14°, 18°, 22°, 27°, 30°, 45°). The last independent variable of the data set is surface, including walls and building roof (S1 to S6), marked in Figure 1. In addition, $C_{p,mean}$, $C_{p,rms}$, and $C_{p,peak}$ are prediction variables.

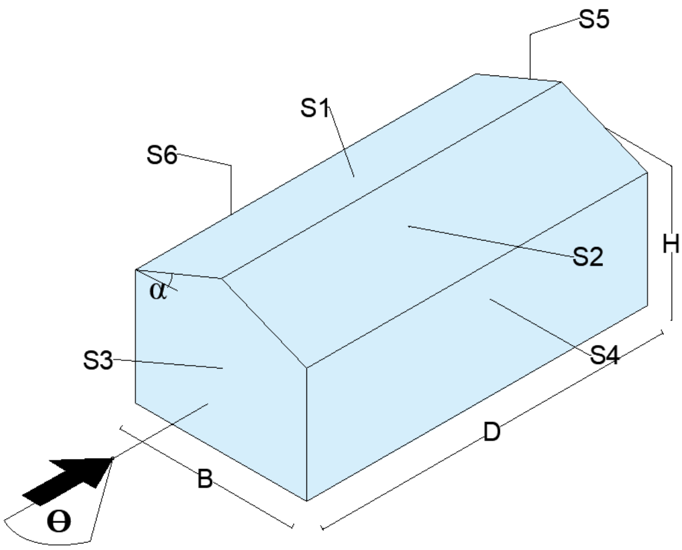


Figure 1. Low-rise gable roof building: (H—Height to mid-roof from ground level, B—Breadth of the building, D—Depth of building, θ—Direction of wind, α—Roof pitch, and "S" denotes surface).

Table 1 provides a summary of the input and output variables of the present study. Since surface 1 to surface 6 is categorical, we used one-hot encoding. For occasions where an ordinary relationship does not exist, one-hot encoding is a better option compared to integer encoding. All independent and predicting variables were tabulated and 60% of data samples (2420 out of 4032 total samples) were fed into the training sequence while the remaining 40% is employed for the validation (*out-of-bag data*) process. All variables were provided to four tree-based algorithms through the sci-kit learn library in python [84].

Table 1. Descriptive statistics of the dataset.

Type	Variable	Details
Independent	Wind direction (θ)	0°, 15°, 30°, 45°, 60°, 75°, 90°
	Roof Pitch (α)	5°, 10°, 14°, 18°, 22°, 27°, 30°, 45°
	H/B	0.25, 0.5, 0.75, 1
	D/B	1, 1.5, 2.5
	S1	One-hot encoding is active. A value of "1" is used when a particular surface is referred whereas remaining surfaces hold "0".
	S2	
	S3	
	S4	
	S5	
	S6	
Dependant	$C_{p,mean}$	Ranges from −1.39 to 0.74
	$C_{p,rms}$	Ranges from 0.09 to 0.57
	$C_{p,peak}$ [Single worst minimum]	Ranges from −4.91 to −0.13

In addition, hyperparameters are required to optimize all four models. All required hyperparameters of four tree-based models were chosen based on a *grid search*. Optimized model predictions were compared using performance indices. Subsequently, we followed the model explanatory process to elucidate the causality of predictions. Figure 2 summarizes the workflow of this study. The methodology shows the existing research gap in the wind engineering field. Therefore, the authors strongly believe the novelty-*explainable ML*-would advance the wind engineering research community.

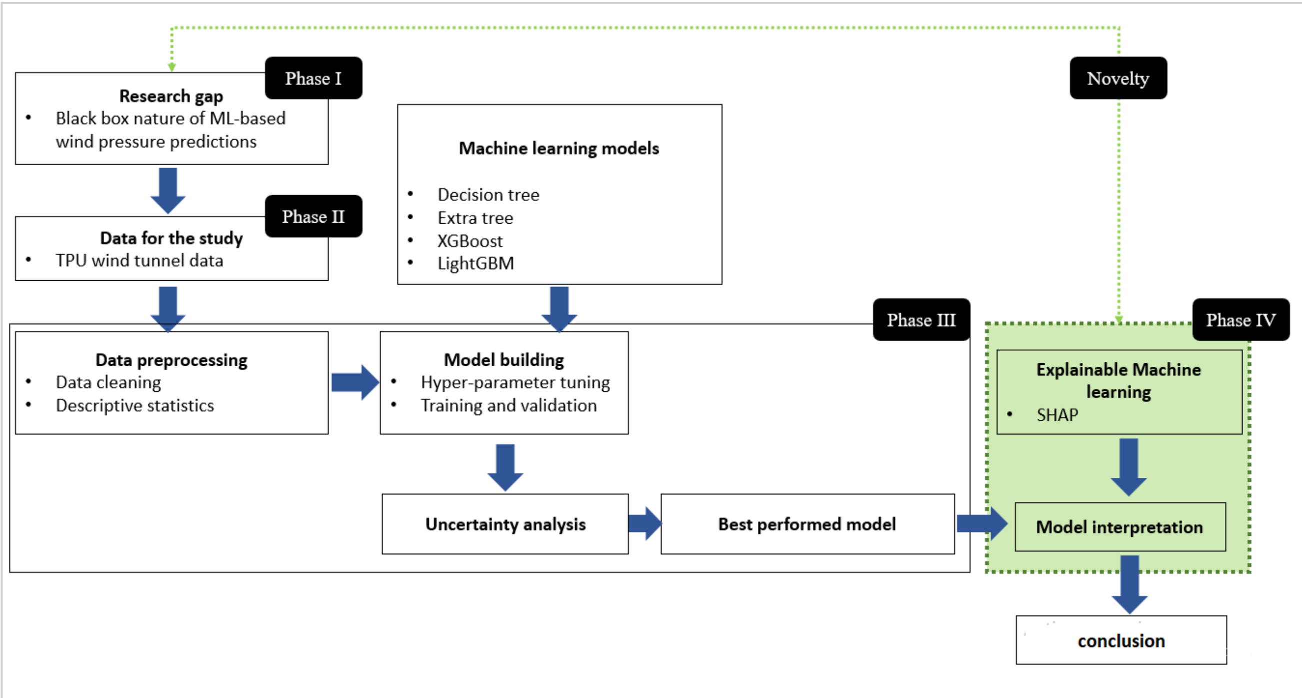
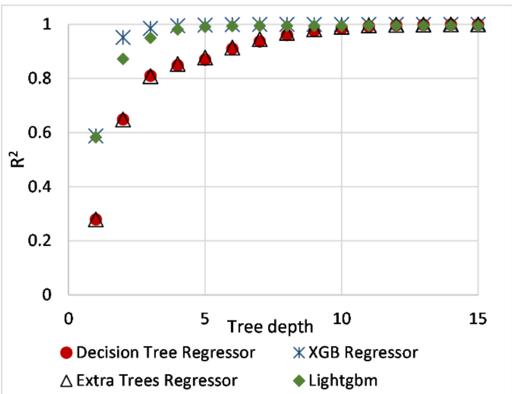


Figure 2. Proposed workflow of this study.

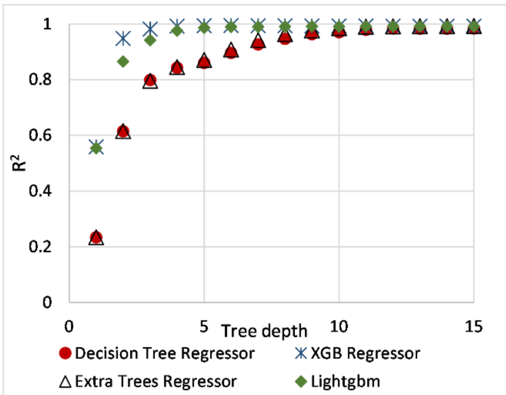
6. Results and Discussion

6.1. Hyperparameter Optimization

The grid search method was used to optimize the hyperparameters for all four tree-based models. Grid search considers different combinations of parameters and evaluated their respective output to obtain optimum values. Table 2 presents the optimized hyperparameters for each model. Their definitions are presented in Appendix A. Interestingly, tree-depth held a greater significance compared to the remaining hyperparameters. Figure 3 depicts how the accuracy of training and validation depends on tree-depth.

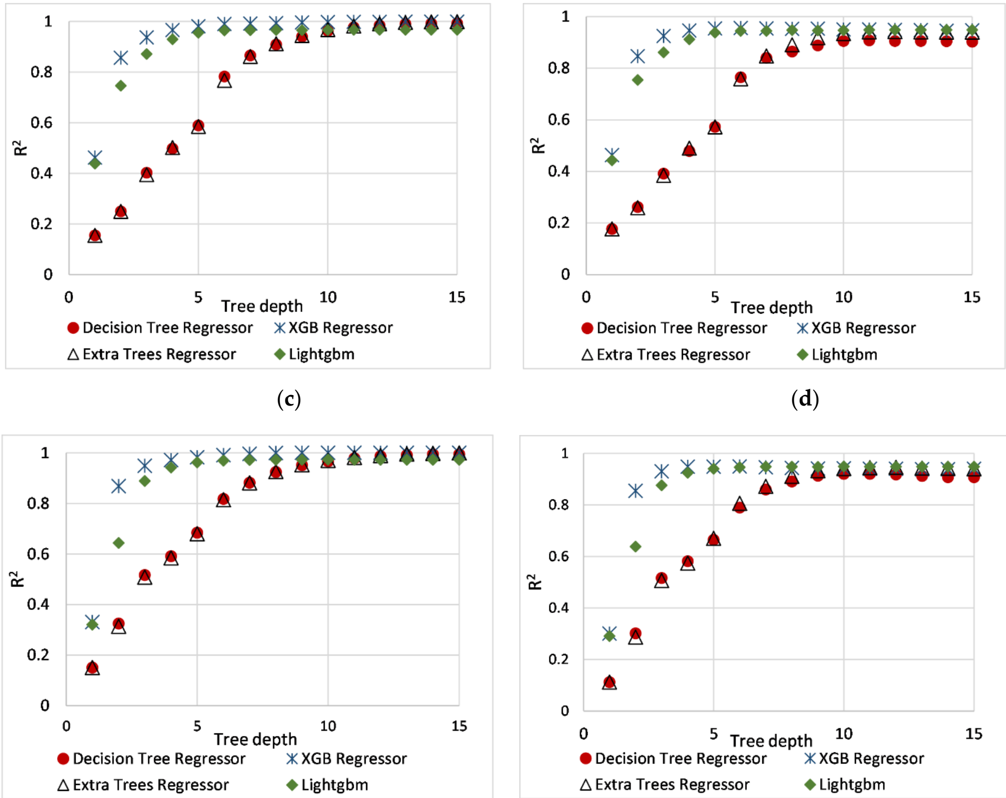


(**a**)



(**b**)

Figure 3. *Cont.*



(c) (d)

(e) (f)

Figure 3. Variation of R^2 with the depth of each tree model; training split = 60%, and validation split = 40%. (**a**) The training process of $C_{p,mean}$, (**b**) The validation process of $C_{p,mean}$, (**c**) The training process of $C_{p,rms}$, (**d**) The validation process of $C_{p,rms}$, (**e**) The training process e of $C_{p,peak}$, (**f**) The validation process of $C_{p,peak}$.

Table 2. Optimized hyperparameters for tree-based ML models.

	Decision Tree		**Extra Tree**		**XGBoost**		**LightGBM**	
	Hyperparameter	**Value**	**Hyperparameter**	**Value**	**Hyperparameter**	**Value**	**Hyperparameter**	**Value**
$C_{p,mean}$	criterion	MSE	criterion	MSE	Maximum depth	4	Learning rate	0.1
	splitter	Best	Maximum depth	10	Gamma	0.0002	Maximum depth	4
	Maximum depth	10	Minimum samples leaf	2	Learning rate	0.3	Random state	5464
	Minimum samples leaf	2	Minimum sample split	2	Number of Estimators	50	Number of jobs	none
	Minimum sample split	2	Number of Estimators	50	Random state	154	Reg_Alpha	1×10^{-4}
	Maximum Features	5	Bootstrap	FALSE	Reg_Alpha	0.0001		
	Minimum impurity decrease	0	Minimum impurity decrease	0	Base score	0.5		
	Random state	5464	Random state	5464				
	CC alpha	0	Number of jobs	none				
$C_{p,rms}$	criterion	MSE	criterion	MSE	Maximum depth	5	Learning rate	0.1
	splitter	Best	Maximum depth	10	Gamma	0.0001	Maximum depth	5
	Maximum depth	10	Minimum samples leaf	2	Learning rate	0.4	Random state	5464
	Minimum samples leaf	2	Minimum sample split	2	Number of Estimators	400	Number of jobs	none
	Minimum sample split	2	Number of Estimators	100	Random state	154	Reg_Alpha	1×10^{-4}
	Maximum Features	5	Bootstrap	True	Reg_Alpha	0.0001		
	Minimum impurity decrease	0	Minimum impurity decrease	0	Base score	0.5		
	Random state	5464	Random state	4745				
	CC alpha	0	Number of jobs	100				

Table 2. *Cont.*

	Decision Tree		Extra Tree		XGBoost		LightGBM	
	Hyperparameter	**Value**	**Hyperparameter**	**Value**	**Hyperparameter**	**Value**	**Hyperparameter**	**Value**
$C_{p,peak}$	criterion	MSE	criterion	MSE	Maximum depth	4	Learning rate	0.1
	splitter	Best	Maximum depth	10	Gamma	0.0001	Maximum depth	5
	Maximum depth	10	Minimum samples leaf	2	Learning rate	0.053	Random state	5464
	Minimum samples leaf	2	Minimum sample split	2	Number of Estimators	400	Number of jobs	none
	Minimum sample split	2	Number of Estimators	100	Random state	154	Reg_Alpha	1×10^{-4}
	Maximum Features	5	Bootstrap	True	Reg_Alpha	0.0001		
	Minimum impurity decrease	0	Minimum impurity decrease	0	Base score	0.5		
	Random state	5464	Random state	4745				
	CC alpha	0	Number of jobs	100				

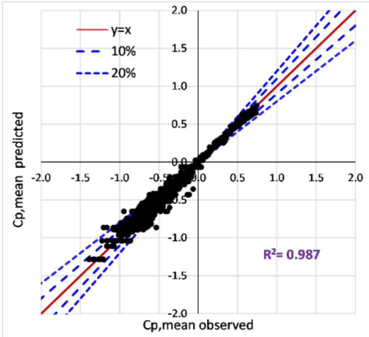
6.2. Training and Validation of Tree-Based Models

Figure 3 depicts the performance of tree-based regressors in terms of training and validation. Both XGBoost and LightGBM achieve an $R^2 > 0.8$ at a depth of 3, showing greater adaptability to wind pressure data. The Decision Tree regressor comprises enhanced predictions at depths between 5 and 10, though it is considered a weak learner compared to the remaining models. Both the Decision Tree and the Extra Tree slowly increase prediction accuracy with respect to remaining tree-based models. Compared to the training process of $C_{p,mean}$, Decision Tree, Extra Tree, and LightGBM decelerated training phase of $C_{p,rms}$, and $C_{p,peak}$ as shown in Figure 3. However, the same performance was maintained by the XGBoost regressor throughout the entire training process. Despite different forms of pressure coefficients, all models exceed R^2 beyond 0.95 and become stalled between depths of 5 to 10. It is noteworthy that the stalled R^2 values beyond depths of 10 can lead to overfit the model.

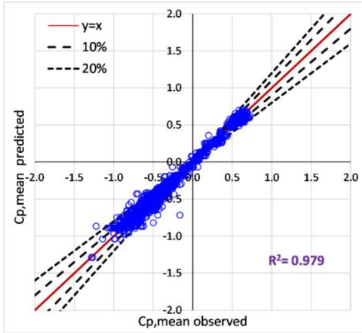
6.3. Prediction of Surface-Averaged Pressure Coefficients

According to Figure 4, all models provide accurate predictions to the validation set of $C_{p,mean}$ compared to wind tunnel data. XGBoost and Extra-Tree exhibit improved predictions with moderately higher R^2 values of 0.992 and 0.985, respectively. The Extra-Tree and XGBoost maintain the consistency of validation predictions for both negative and positive values within a 20% margin, most of the time. Slight inconsistencies can be observed for negative predictions obtained from the Decision Tree regressor. The accuracy of LightGBM is also satisfactory for $C_{p,mean}$ predictions.

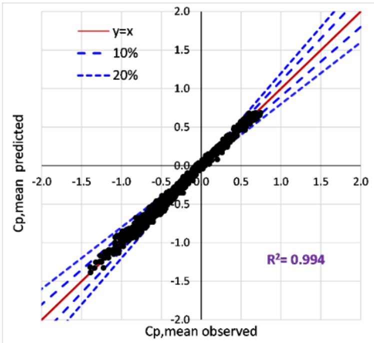
The coefficient $C_{p,rms}$ represents turbulent fluctuations. $C_{p,rms}$ is important, especially along perimeter regions where high turbulence is present. However, such localization was not feasible with the available data set due to the distinct pressure tap arrangement of each geometry configuration. Hence, for the present study, we had to employ (area-averaged) $C_{p,rms}$ coefficient, which may not reflect localized effects. Nevertheless, the accuracy of validation predictions remains within a 20% error margin except for the Decision Tree regressor (see Figure 5). Both the XGBoost and LightGBM validation predictions are consistent, achieving relatively higher R^2 values.



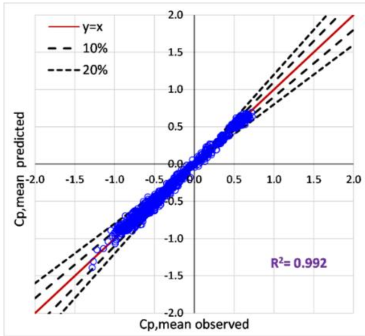
Decision Tree training



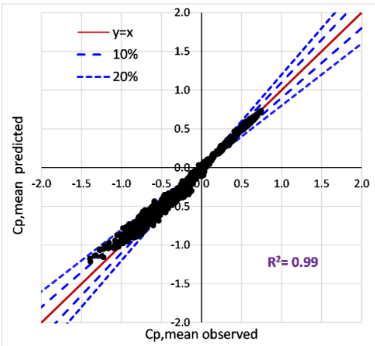
Decision Tree validation



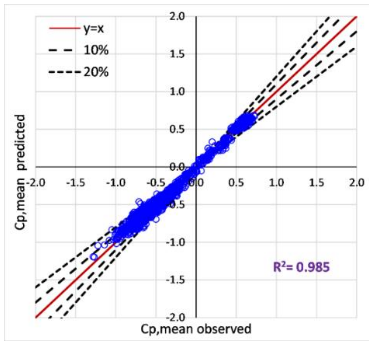
XGBoost training



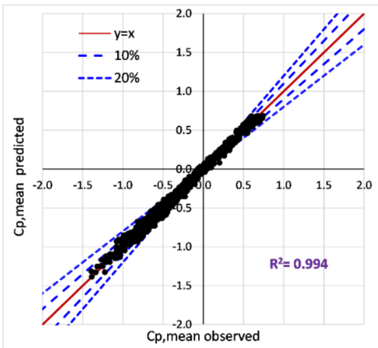
XGBoost validation



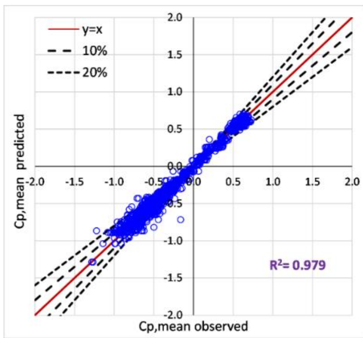
Extra tree training



Extra tree validation



LightGBM training



LightGBM validation

Figure 4. Comparison of overall tree-based predictions and wind tunnel data: ($C_{p,mean}$).

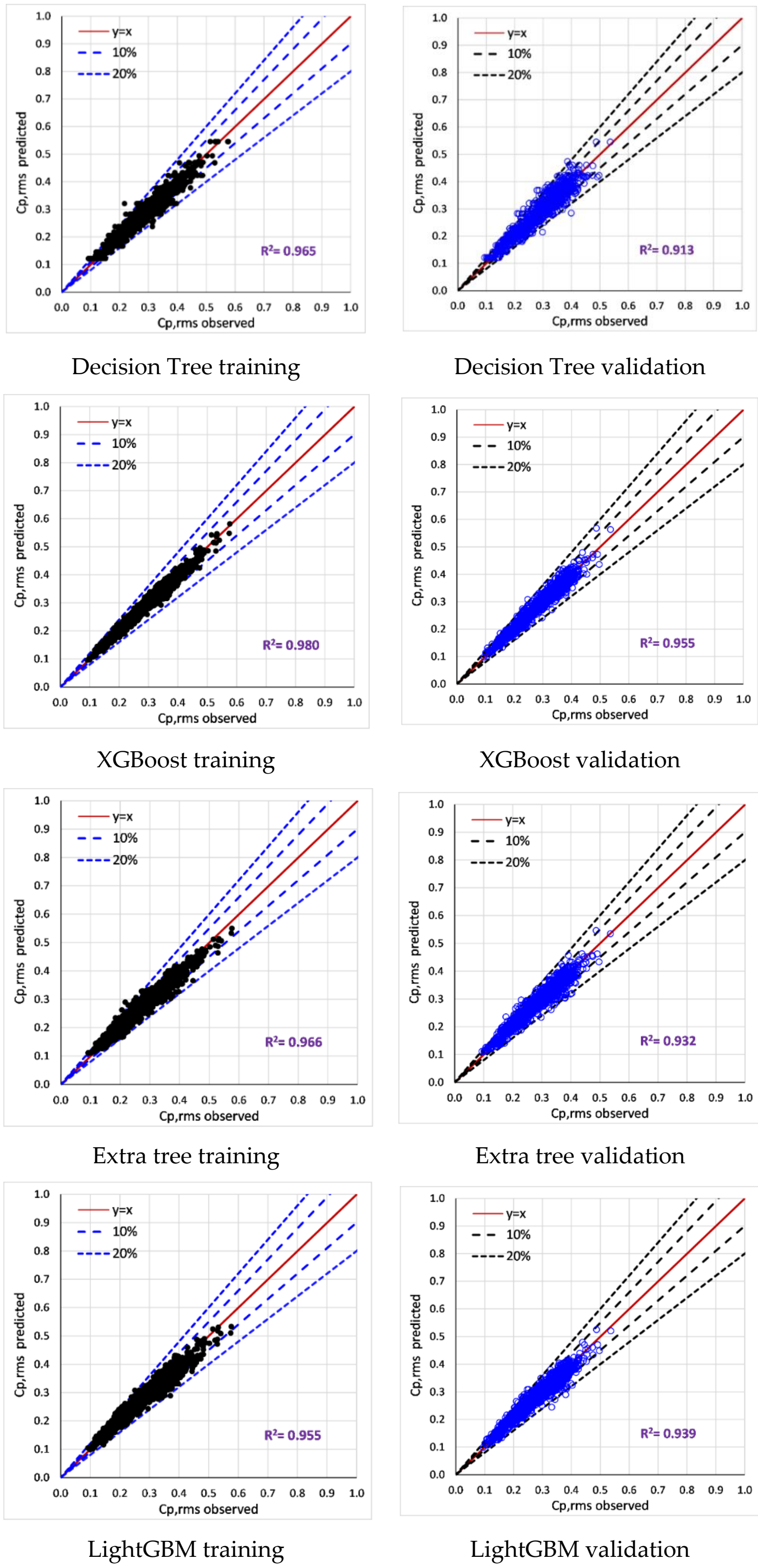
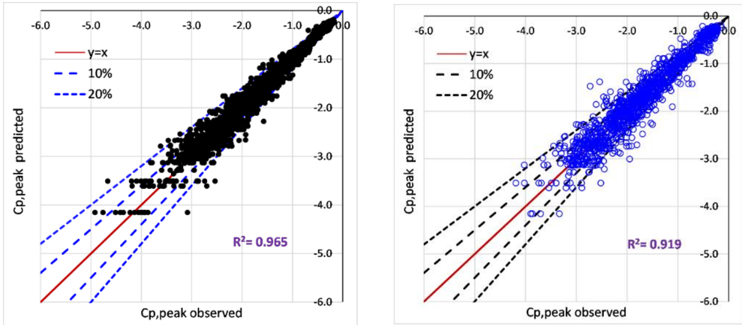
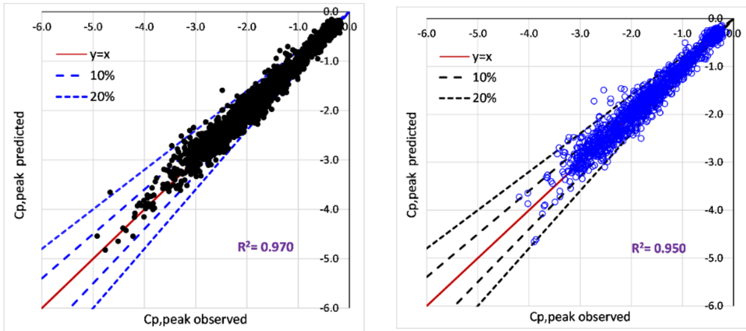


Figure 5. Comparison of overall tree-based predictions and wind tunnel data: ($C_{p,rms}$).

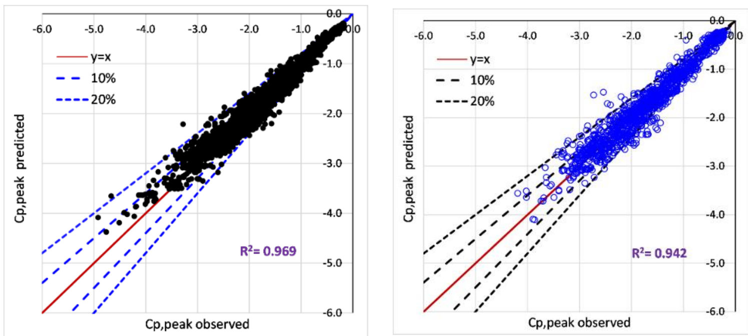
For $C_{p,peak}$ predictions, negative peaks were only considered as they induce large uplift forces on the external envelope [84]. Generally, the external pressure on roofs is more important than the pressure on walls as the roof is subjected to relatively larger negative pressure compared to walls. Especially negative pressure on the roof surface is critical to assess vulnerability for progressive failures [85]. Even with surface averaging, values closer to −5 (See Figure 6) have been obtained. Compared to mean and fluctuation predictions, peak pressure predictions (validation) have deviations beyond the 20% error limit for all tree-based models. In the case of the Decision Tree model, substantial deviations are visible for lower $C_{p,peak}$ values, compared to wind tunnel data. Repeatedly, the XGBoost model showcased superior performance (R^2 = 0.955) with respect to the remaining models.



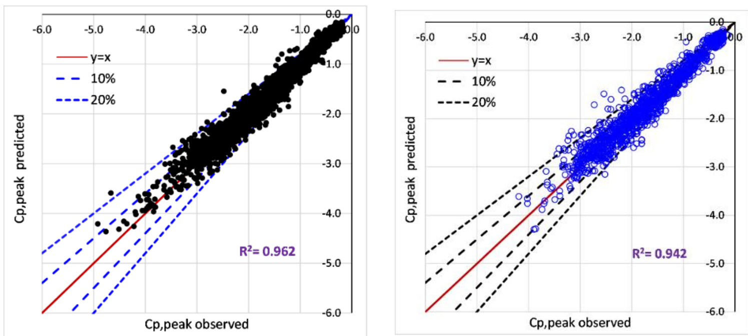
Decision Tree training Decision Tree validation



XGBoost training XGBoost validation



Extra tree training Extra tree validation



LightGBM training LightGBM validation

Figure 6. Comparison of overall tree-based predictions and wind tunnel data: ($C_{p,peak}$).

6.4. Performance Evaluation of Tree-Based Models

Figures 4–6 provide insights into the applicability of tree-based models to predict wind pressure coefficients. However, models require performance analysis to explain the uncertainty associated with each model. The summary of model performance indices (validation predictions) is shown in Table 3.

Table 3. Performance indices of optimized tree-based models.

C_p	Performance Indicators	Decision Tree	Extra Tree	LightGBM	XGBoost
$C_{p,mean}$	Correlation (R)	0.990	0.993	0.977	0.996
	MAE	0.049	0.043	0.051	0.033
	RMSE	0.067	0.055	0.07	0.043
$C_{p,rms}$	Correlation (R)	0.955	0.966	0.969	0.977
	MAE	0.018	0.016	0.015	0.013
	RMSE	0.023	0.020	0.019	0.017
$C_{p,peak}$	Correlation (R)	0.957	0.970	0.970	0.974
	MAE	0.177	0.150	0.149	0.143
	RMSE	0.246	0.206	0.205	0.192

Based on lower tree depth, XGBoost and LightGBM perform better than the remaining two models. In terms of validation predictions, XGBoost achieved the highest correlation. All models are adaptable to $C_{p,rms}$ predictions as observed from lower MAE and RMSE values. Among four models, the Decision Tree showcased the lowest accuracy for $C_{p,rms}$, and $C_{p,peak}$ whereas lightGBM showcased the lowest accuracy for $C_{p,mean}$ predictions.

Bre et al. [33] obtained an R of 0.9995 for $C_{p,mean}$ predictions of gable-roofed low-rise buildings using an ANN. According to Table 3, the same task is performed using tree-based ML models (R > 0.98) while expanding predictions to $C_{p,peak}$, and $C_{p,rms}$. In this study, we have separately provided training and validation accuracies because reliability is a measure in terms of the accuracy of training and validation (refer to Figure 7). For example, if training accuracy is considerably higher than validation, it leads to over-fitting. Lower accuracies of both imply under-fitting occasions. Therefore, obtaining similar accuracies in training and validation is more important (see Figure 3). Based upon R, RMSE, and MAE values, the XGBoost model was selected as the superior model and continued to the explanatory process.

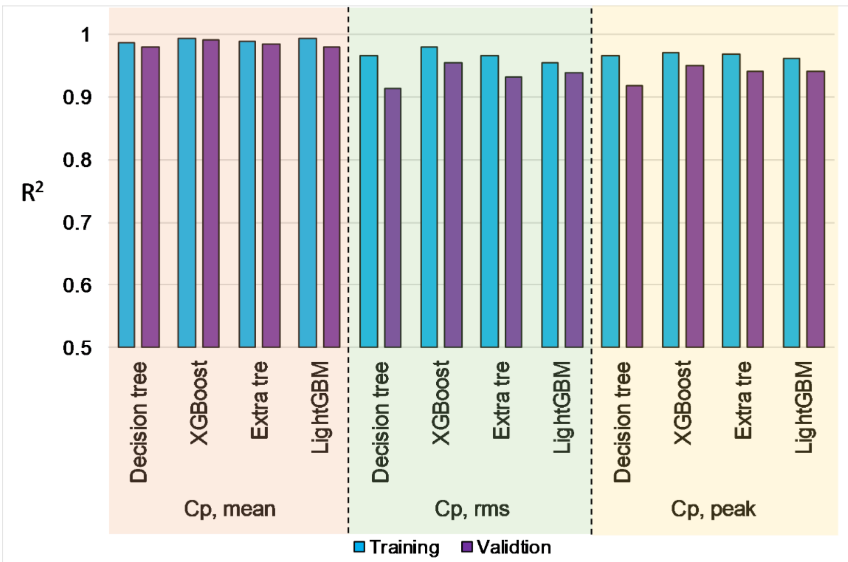


Figure 7. Training and validation score of tree-based models.

7. Model Explanations

7.1. Model-Based (Intrinsic) Explanations

All tree-based algorithms used in this study are based on the Decision Tree structure. A simple Decision Tree is explainable without a post hoc agent. Figure 8 provides a Decision Tree regressor formed at a depth of three for the current study. X[number] denotes the number assigned to a variable in the selected order. (0—wind angle (θ); 1—D/B; 2—H/B; 3—Roof angle (α); 4—Surface 1 (S1); 5—Surface 2 (S2); 6—Surface 3 (S3); 7—Surface 4 (S4); 8—Surface 5 (S5); 9—Surface 6 (S6)). Value in the box refers to the mean value of the predictor variable of the samples passing through each box.

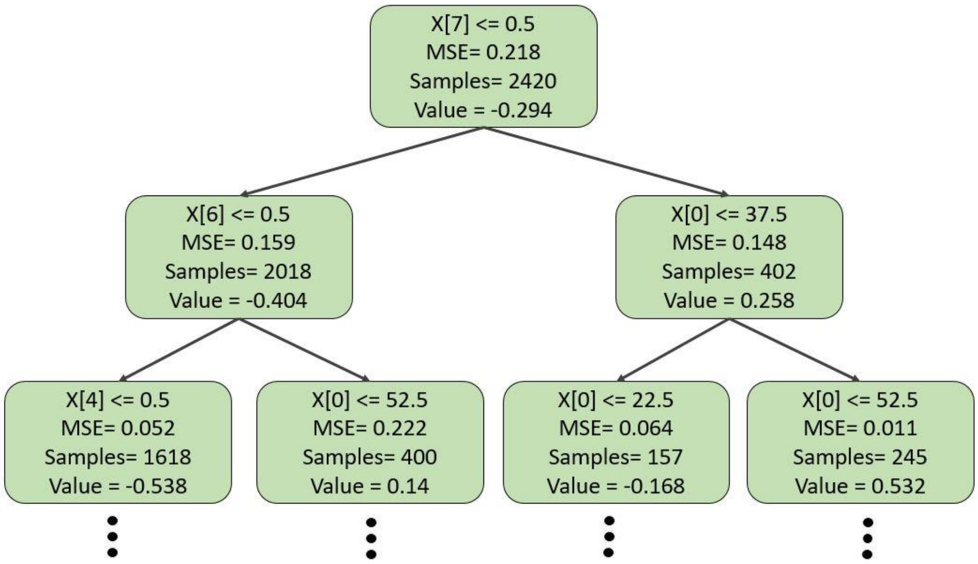


Figure 8. Tree structure of Decision Tree at up to first three layers.

Splitting takes place at each node based on MSE. The selection criteria are followed by "IF-THEN" and "ELSE" statements. For instance, at the root node of the Decision Tree, variables are clustered based on whether it is surface 4 (X [7]) or not. After confirming value does not belong to surface 4, the tree decides wind direction (X [0]) should be examined. There it leads to initiate clustering depending on whether the $\theta < 37.5°$. Subsequently, ($\theta < 37.5°$) is clustered farther towards the left whereas the remaining will be clustered to the right. Interestingly, layer-wise splitting is performed by separating large deviations into a single group, gradually reducing MSE.

Though XGBoost, Extra-tree, and LightGBM are Decision Tree-based architectures, within those tree-based models, various methods such as bagging and boosting are implemented, which lead to perceive different outcomes. For example, XGBoost forms numerous Decision Trees at a particular point and conducts voting criteria to obtain a suitable one.

7.2. SHAP Explanations

As previously highlighted, a post hoc component is required to explain complex models. The authors used SHAP for this purpose. The explanatory process is divided into two stages. First, the global explanation is provided using SHAP explanations. Secondly, instance SHAP values are explained to compare the causality of predictions with respect to experimental behavior. Figure 9 depicts SHAP explanation on $C_{p,mean}$ prediction of XGBoost.

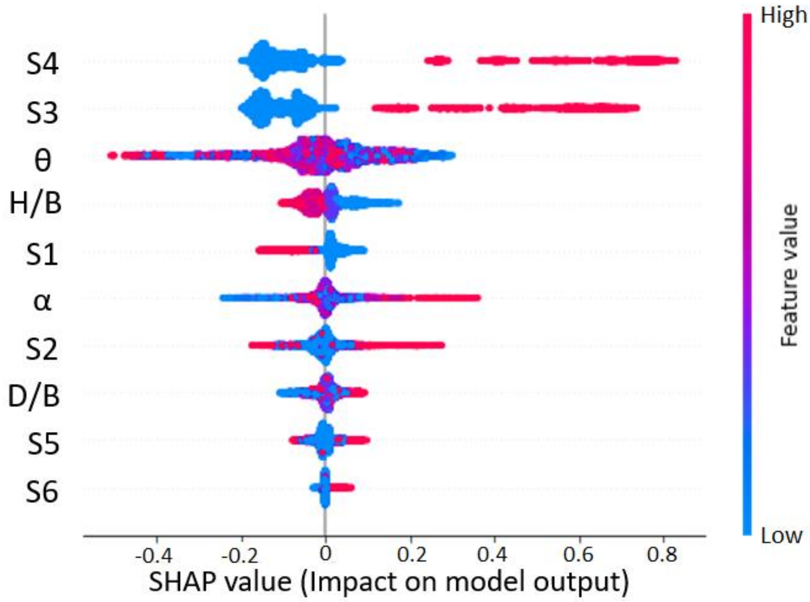


Figure 9. SHAP Global interpretation on XGBoost regressor ($C_{p,mean}$ prediction).

The horizontal axis determines whether the impact is negative or positive. Feature values are denoted using red and blue color, whereas the former indicates higher feature values and the latter represents lower values. For example, wind angle contains several distinct values. Therefore, SHAP identifies values closer to 90° as higher feature values and values closer to 0° as lower feature values. Further, all surfaces are binary variables such that "1" (Higher feature value) means that the instance belongs to a particular surface and "0" (Lower feature value) indicates that the instance does not belong to the corresponding surface. SHAP identified that if instances belong to surface 4, there is a higher possibility to make a positive impact on the model whereas "not being surface 4" influences a low negative impact on the model. Generally, surface 3 and surface 4 experience positive $C_{p,mean}$ values when it is facing a windward direction. Therefore, a considerable positive effect can be expected on the model output as identified by SHAP. Subsequently, concatenated blue-colored values explicate that there are more instances where a low negative impact occurs due to lower feature values.

When the wind angle (θ) is considered, the concatenated portion is observed towards the middle of the horizontal axis. That explains more instances exist where a neutral effect is observed regardless of feature values. Nevertheless, higher wind angles (closer to 90°) have influenced a moderately higher negative effect on model output. Fascinatingly, SHAP recognizes that higher roof pitch influences a positive effect on model output. According to SHAP explanation, surface 2, surface 3, surface 4, roof angle and wind angle have a substantial contribution to $C_{p,mean}$ compared to the remaining variables.

A markedly different explanation was obtained for $C_{p,rms}$ (See Figure 10). Surface 3 has become a crucial factor for model output. If a particular instance belongs to surface 3, it results in a positive impact on the model. Surface 2 has influenced a considerable positive (SHAP value > 0.125) effect on $C_{p,rms}$. Lower values of wind angle and H/B ratio cause a negative influence on the output. Surface 5 creates a mixed effect based on feature values, and that effect is more pronounced in roof angle and wind angle. However, concatenated regions indicate a neutral effect from wind angle and roof angle regardless of feature value. Overall, the effect of surface 6 was less notable on $C_{p,rms}$. Effects of H/B and D/B ratios on $C_{p,rms}$ contradict the effect that appeared on $C_{p,mean}$.

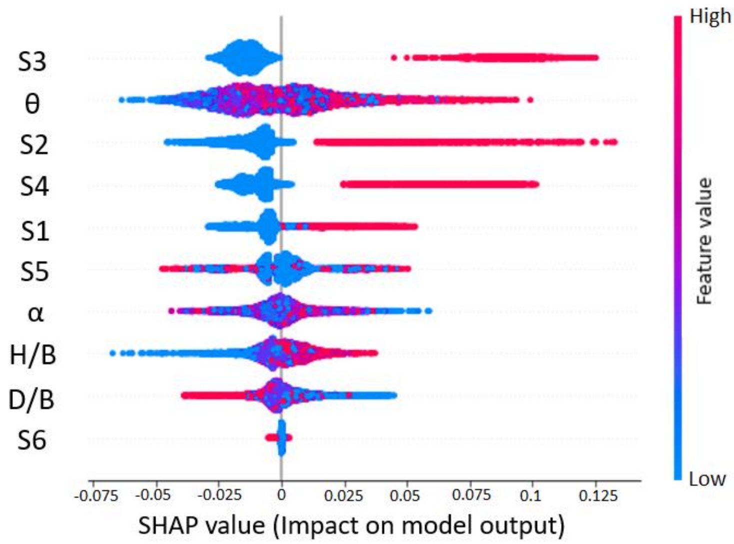


Figure 10. SHAP Global interpretation on XGBoost regressor ($C_{p,rms}$ prediction).

Repeatedly, Surface 3 (S3) and Surface 4 (S4) provided a comparable effect on $C_{p,peak}$ (See Figure 11). In addition, Surface 5 and Surface 6 (roof halves) show a similar effect on model output except for the fact that surface 5 can influence both positive and negative effects on $C_{p,peak}$. A notable difference was observed from the θ that influences a relatively higher negative effect on the ML model regardless of feature value. On the contrary, roof pitch (α) creates a dissimilar effect compared to θ, whereas higher feature values create a positive impact and lower features create a negative impact. H/B and D/B ratio creates a completely different impact on $C_{p,peak}$. Compared to the $C_{p,rms}$, the effect of surfaces 1 and 2 has been reversed. Less notable features observed in $C_{p,mean}$ have become considerably important for the $C_{p,peak}$. The overall effect of surfaces 3 and 4 was similar for all forms of predictions. The effects of H/B and D/B were comparable for $C_{p,mean}$, and $C_{p,peak}$ but completely reversed for $C_{p,rms}$. There exist instances in which the effect of wind direction and roof pitch appeared to be less dependent on feature values. However, we recognized that a completely different impact was observed from wind direction and roof pitch. The effect of roof surfaces becomes pronounced $C_{p,peak}$.

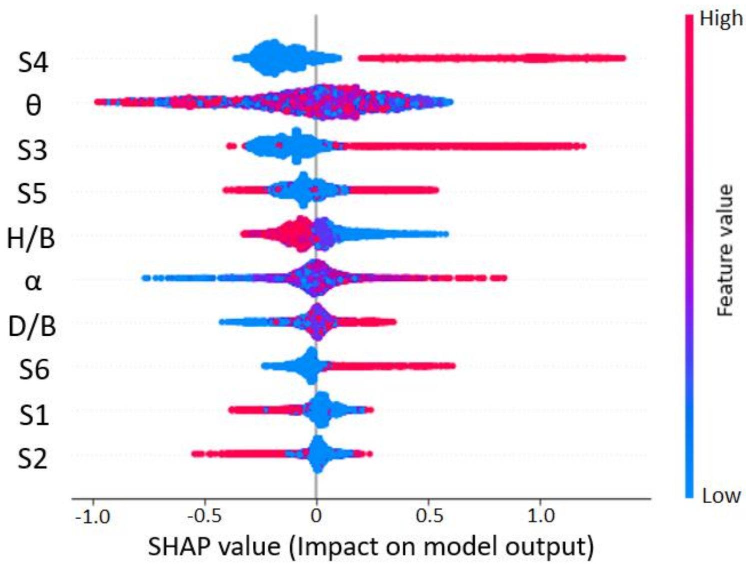


Figure 11. SHAP Global interpretation on XGBoost regressor ($C_{p,peak}$ prediction).

Heretofore, a generic overview of XGBoost models was described with the aid of SHAP explanations. Further, SHAP can provide a force plot picture based on each instance. That elucidates how each parameter forces the base value. The base value is defined as the average value observed during the training sequence. Red color features force the base value to be higher and blue color features force the lower side. The length of each segment is proportionate to its contribution (see Figure 12). SHAP can quantify the largest influential parameters at any given instance, assisting decision-making.

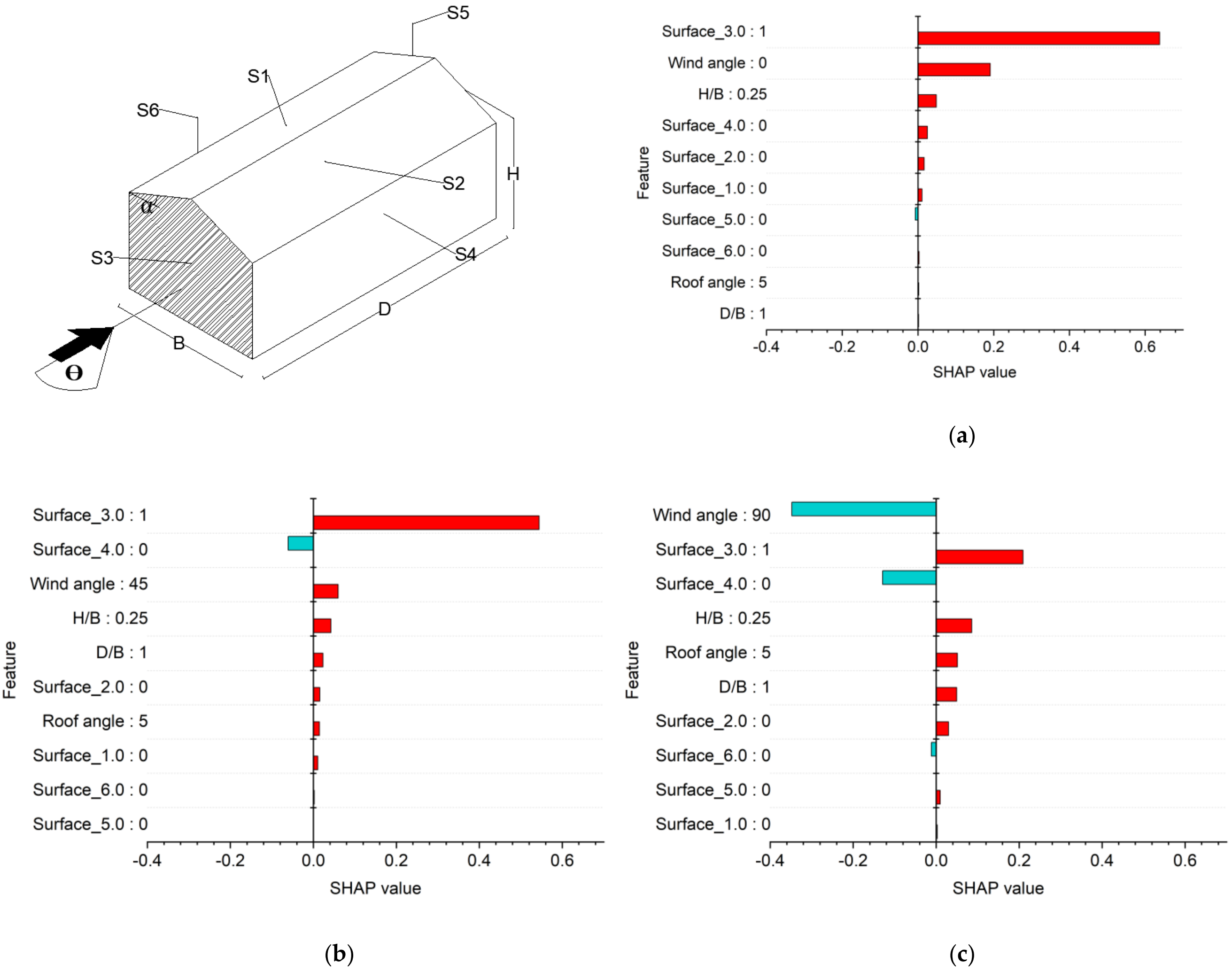


Figure 12. SHAP force plot for $C_{p,mean}$ on surface 3 (D/B = 1, H/B = 0.25, α = 5°). (**a**) Predicted $C_{p,mean}$ = 0.58; (**b**) Predicted $C_{p,mean}$ = 0.33; (**c**) Predicted $C_{p,mean}$ = −0.35.

For example, we can consider three instances on surface 3, varying the only direction of the wind as 0°, 45°, and 90°. Figure 12 illustrates the SHAP force plot on these instances, considering $C_{p,mean}$. The base value tends to reduce when θ increases. For example, at θ = 0°, approaching wind directly attacks surface 3, where maximum positive pressure is obtained. Subsequently, positive pressure decreases for oblique wind directions as observed at θ = 45°. Thus, the maximum negative value is obtained at θ = 90° when surface 3 becomes a sidewall. From θ = 0° to θ = 90°, the effect of flow separation causes negative pressure to dominate on surface 3. Similar variation is observed based on the SHAP explanation where the effect of wind angle is positive (forcing towards higher values; Figure 12a) and gradually becomes negative (forcing towards lower values; Figure 12c). SHAP claims negative contribution is substantial at θ = 90° compared to the magnitude of contribution at θ = 0°. Moreover, SHAP has quantified the contribution of each parameter toward the instance value.

Figure 13 displays four instances of $C_{p,mean}$ on surface 2 at θ = 90° (along wind direction for surface 2). More importantly, SHAP confirmed the effect of roof pitch as one of the influencing parameters in each instance. From α = 5° to α = 45°, the effect of roof pitch shifts from negative (forcing towards lower) to positive (forcing towards higher). From the wind engineering viewpoint, flat roofs induce large negative pressure due to the effect of flow separation along the upwind edge. When roof pitch increases, the effect of flow separation on the upwind half of the roof becomes less pronounced [85]. Further, at α = 45°, positive values dominate on the upwind half of the building roof, indicating flow reattachments.

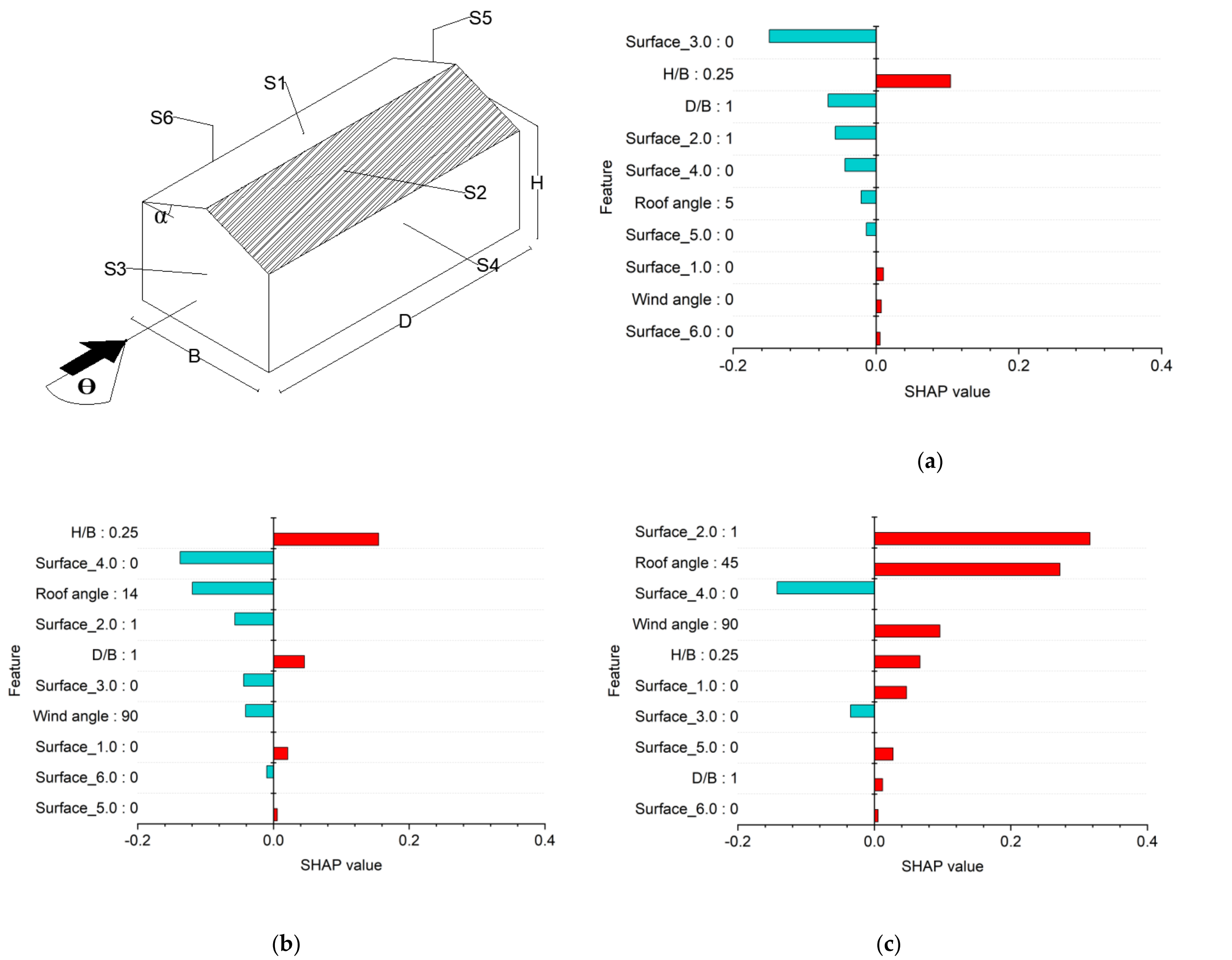


Figure 13. SHAP force plot for $C_{p,mean}$ on surface 2 (D/B = 1, H/B = 0.25, θ = 90°). (**a**) Predicted $C_{p,mean}$ = −0.64; (**b**) Predicted $C_{p,mean}$ = −0.48; (**c**) Predicted $C_{p,mean}$ = 0.28.

Figures 14 and 15 describe three instances of $C_{p,rms}$, and $C_{p,peak}$ on surface 1 at θ = 0° (crosswind direction for surface 1), for three different roof pitches (α = 5°, 14°, 27°). SHAP identifies that both $C_{p,rms}$, and $C_{p,peak}$ have increased due to increased roof pitch. Interestingly, wind tunnel results confirmed the same observations as a result of an increase in roof pitch. Except for roof pitch and H/B, the remaining parameters force the base value towards the lower side (Figure 15). $C_{p,rms}$ are critical in zones where high turbulence is expected. Peak values are critical, especially near roof corners and perimeter zones.

The overall explanations indicate that SHAP adheres to what is generally observed in the external pressure of the low-rise gable-roofed building.

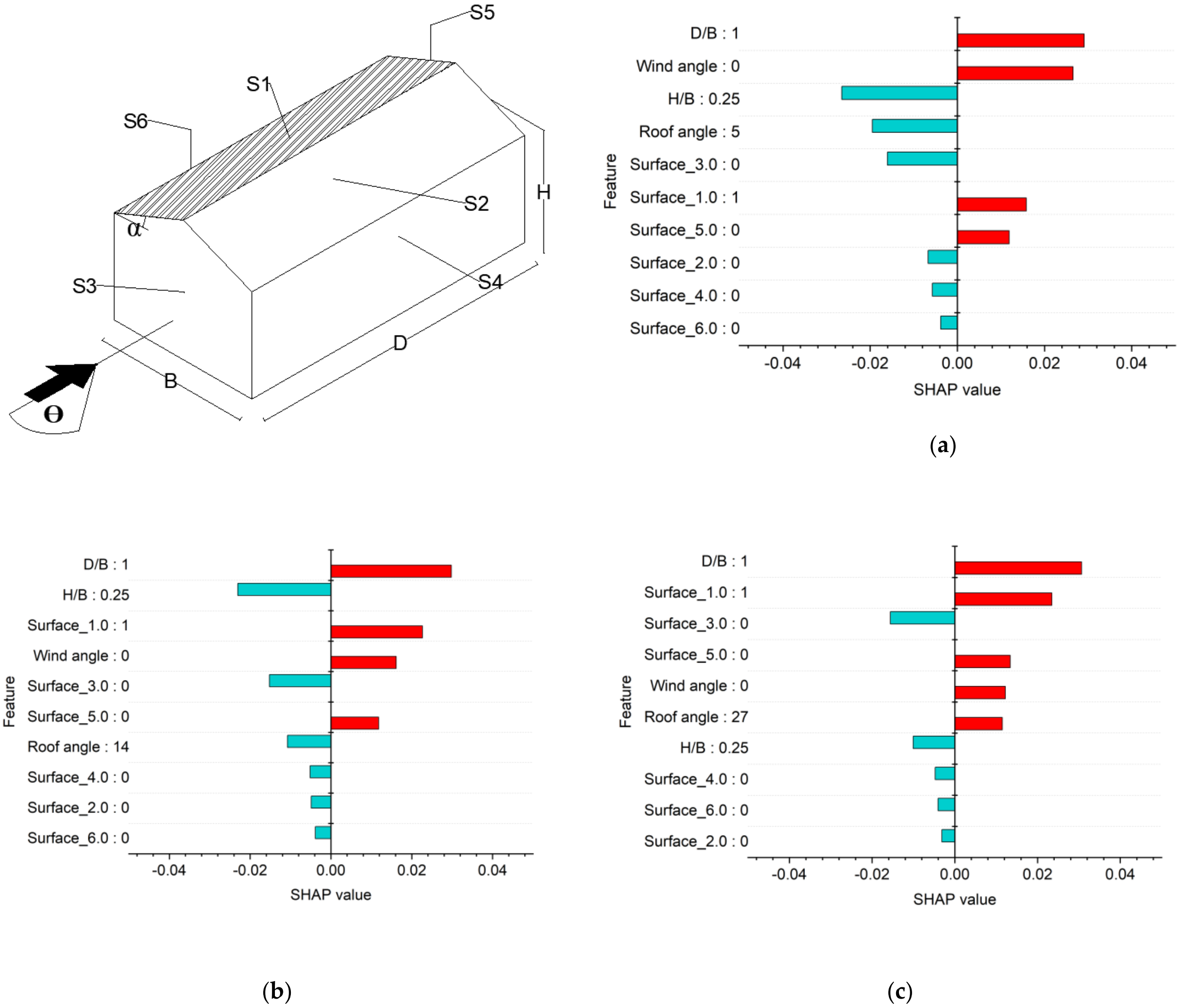


Figure 14. SHAP force plot for $C_{p,rms}$ on surface 1 (D/B = 1, H/B = 0.25, θ = 0). (**a**) Predicted $C_{p,rms}$ = 0.29; (**b**) Predicted $C_{p,rms}$ = 0.29; (**c**) Predicted $C_{p,rms}$ = 0.37.

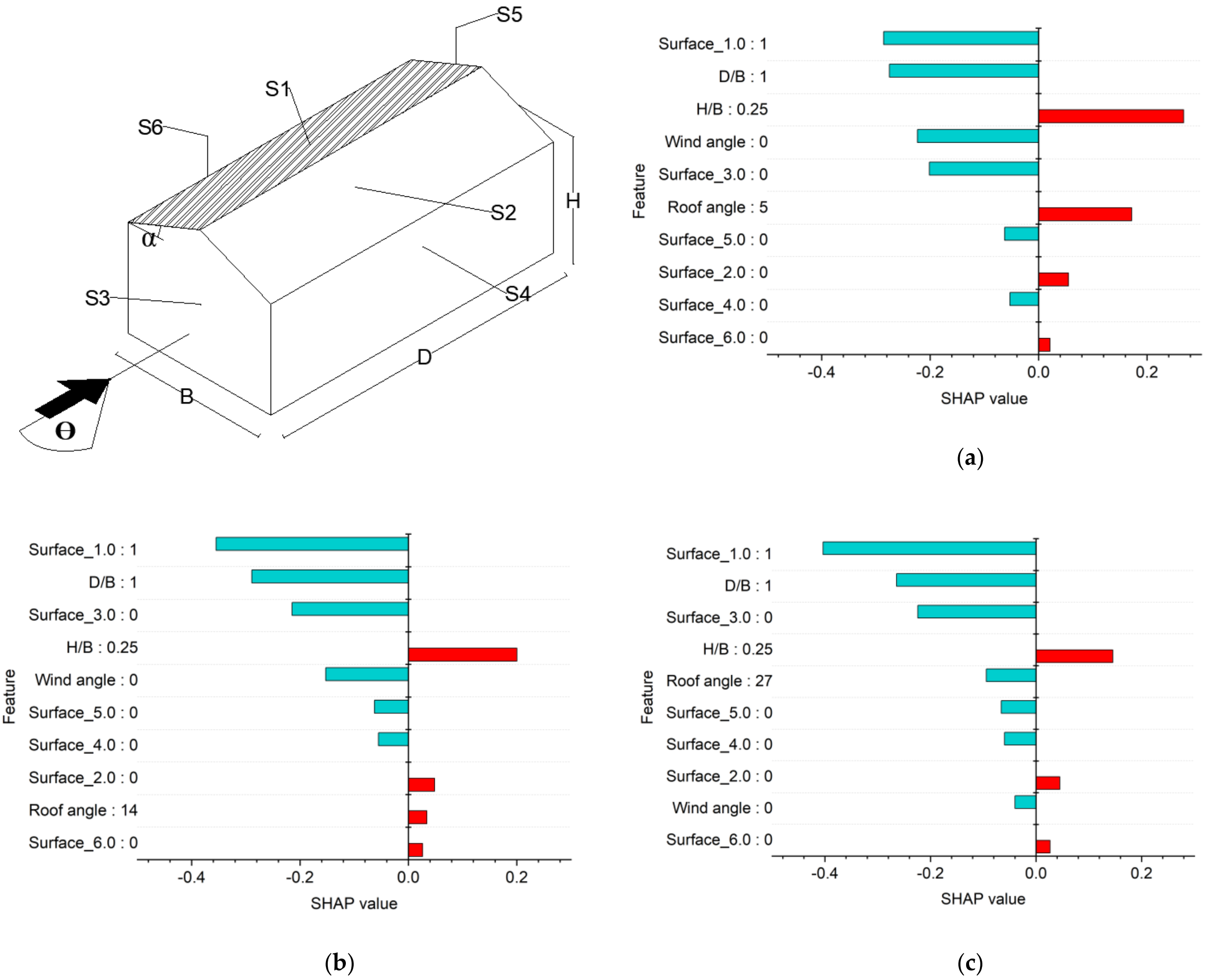


Figure 15. SHAP force plot for $C_{p,peak}$ on surface 1 (D/B = 1, H/B = 0.25, $\theta = 0°$). (**a**) Predicted $C_{p,peak} = -2.26$; (**b**) Predicted $C_{p,peak} = -2.34$; (**c**) Predicted $C_{p,peak} = -2.75$.

8. Conclusions

Implementation of ML in wind engineering needs to advance the end-user's trust in the predictions. In this study, we predicted surface-averaged pressure coefficients (mean, fluctuation, peak) of low-rise gable-roofed buildings, using tree-based ML architectures. The explanation method—SHAP—was used to elucidate the inner working and predictions of tree-based ML models. Following are the key conclusions of this study:

- Ensemble methods such as XGBoost and Extra tree are more accurate in estimating surface averaged pressure coefficients than a Decision Tree and LightGBM model. However, decision tree and extra-tree models require a deeper tree structure to achieve good accuracy. Despite the complexity at higher depths, the decision-tree structure is self-explainable. However, complex tree formations (ensemble method: XGBoost, Extra Tree, LightgGBM) require a post hoc explanation method.
- All tree-based models (Decision Tree, Extra tree, XGBoost, LightGBM) accurately predict surface-averaged wind pressure coefficients ($C_{p,mean}$, $C_{p,rms}$, $C_{p,peak}$). For example, tree-based models reproduce surface averaged mean, fluctuating, and peak pressure coefficients ($R > 0.955$). XGBoost model achieved the best performance ($R > 0.974$).
- SHAP explanations convinced predictions to adhere to the elementary flow physics of wind engineering. This provides causality of predictions, the importance of features, and interaction between the features to assist the decision-making process. The

knowledge offered by SHAP is highly imperative to optimize the features at the design stage. Further, combining a post hoc explanation with ML provides confidence to its end-user on "*how a particular instance is predicted*".

9. Limitations of the Study

- According to the obtained TPU data set, the pressure tap configuration is not uniform for each geometry configuration. Therefore, investigating point pressure predictions is difficult. Further, the data set has a limited number of features. Hence, we suggest a comprehensive study to examine the performance of explainable ML, addressing these drawbacks. Adding more parameters would assist in understanding complex behaviors of external wind pressure around low-rise buildings. For example, the external pressure distribution strongly depends on wind velocity and turbulence intensity [86,87]. Therefore, the authors suggest future studies, incorporating these two parameters.
- We used SHAP explanations for the model interpretations. However, there are many explanatory models available to perform the same task. Each model might result in a unique feature importance value. For example, Moradi and Samweld [62] explained the difference between LIME and SHAP on how those models explain an instance. Therefore, a separate study can be conducted using several interpretable (*post hoc*) models to evaluate the explanations. In addition, we recommend comparing intrinsic explanations to investigate the effect of model building and the training process of ML models.
- The present study chooses tree-based ordinary and ensemble methods to predict wind pressure coefficients. As highlighted in the introduction section, many authors have employed sophisticated models (Neural Network Architectures: ANN, DNN) to predict wind pressure characteristics. Given their opinion, we suggest combining interpretation methods with such advanced methods to examine the difference between different ML models.

Author Contributions: Conceptualization, P.M. and U.R.; methodology, P.M.; software, I.E.; validation, P.M. and U.S.P.; formal analysis, U.S.P.; resources and data curation, I.E.; writing—original draft preparation, P.M. and U.S.P.; writing—review and editing, U.R., H.M.A. and M.A.M.S.; visualization, I.E. and P.M.; supervision, U.R.; project administration, U.R.; funding acquisition, U.R. and H.M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The wind pressure data related to analysis are available on TPU (open) wind tunnel database: http://www.wind.arch.t-kougei.ac.jp/info_center/windpressure/lowrise/mainpage.html.

Acknowledgments: The research work and analysis were carried out at the Department of Civil and Environmental Engineering, University of Moratuwa, Sri Lanka.

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

Abbreviations

ABL	Atmospheric Boundary Layer	LSTM	Long short term memory
AI	Artificial Intelligence	ML	Machine learning
ANN	Artificial Neural network	MAE	Mean Absolute Error
CFD	Computational Fluid Dynamics	MSE	Mean Square Error

$C_{p,mean}$	Surface-averaged mean pressure coefficient	NIST	National Institute of Standards and Technology
$C_{p,rms}$	Surface-averaged fluctuation pressure coefficient	R	Coefficient of Correlation
$C_{p,peak}$	Surface-averaged peak pressure coefficient	R^2	Coefficient of Determination
DAD	Database Assist Design	RISE	Real-time Intelligence with Secure Explainable
DNN	Deep neural network	RMSE	Root Mean Square Error
DeepLIFT	Deep Learning Important FeaTures	SHAP	Shapley Additive explainations
GAN	Generative adversarial network	TPU	Tokyo Polytechnic University
LIME	Local Interpretable Model-Agnostic Explanations		

Appendix A

Definition of hyperparameters

- Criterion: The function to measure the quality of a split. Supported criteria are "mse" for the mean squared error.
- Splitter: The strategy used to choose the split at each node. Supported strategies are "best" to choose the best split and "random" to choose the best random split.
- Minimum_samples_split: The minimum number of samples required to split an internal node.
- Minimum sample leaf: The minimum number of samples required to be at a leaf node.
- Random_state: Controls the randomness of the estimator. The features are always randomly permuted at each split, even if "splitter" is set to "best".
- Maximum_depth: The maximum depth of the tree. If none, then nodes are expanded until all leaves are pure or until all leaves contain less than min_samples_split samples.
- Maximum features: The number of features to consider when looking for the best split.
- Minimum impurity decrease: A node will be split if this split induces a decrease of the impurity greater than or equal to this value.
- CC alpha: Complexity parameter used for Minimal Cost-Complexity Pruning.
- Bootstrap: Whether bootstrap samples are used when building trees. If False, the whole dataset is used to build each tree.
- Number of estimators: The number of trees in the forest.
- Number of jobs: The number of jobs to run in parallel.
- Gamma: Minimum loss reduction required to make a further partition on a leaf node of the tree. The larger gamma is, the more conservative the algorithm will be; range: $[0, \infty]$.
- Reg_Alpha: L1 regularization term on weights. Increasing this value will make model more conservative.
- Learning_rate: Learning rate shrinks the contribution of each tree by "learning rate".
- Base score: The initial prediction score of all instances, global bias.

References

1. Fouad, N.S.; Mahmoud, G.H.; Nasr, N.E. Comparative study of international codes wind loads and CFD results for low rise buildings. *Alex. Eng. J.* **2018**, *57*, 3623–3639. [CrossRef]
2. Franke, J.; Hellsten, A.; Schlunzen, K.H.; Carissimo, B. The COST 732 Best Practice Guideline for CFD simulation of flows in the urban environment: A summary. *Int. J. Environ. Pollut.* **2011**, *44*, 419–427. [CrossRef]
3. Liu, S.; Pan, W.; Zhao, X.; Zhang, H.; Cheng, X.; Long, Z.; Chen, Q. Influence of surrounding buildings on wind flow around a building predicted by CFD simulations. *Build. Environ.* **2018**, *140*, 1–10. [CrossRef]
4. Parente, A.; Longo, R.; Ferrarotti, M. Turbulence model formulation and dispersion modelling for the CFD simulation of flows around obstacles and on complex terrains. In *CFD for Atmospheric Flows and Wind Engineering*; Von Karman Institute for Fluid Dynamics: Rhode Saint Genèse, Belgium, 2019. [CrossRef]
5. Tong, Z.; Chen, Y.; Malkawi, A. Defining the Influence Region in neighborhood-scale CFD simulations for natural ventilation design. *Appl. Energy* **2016**, *182*, 625–633. [CrossRef]

6. Rigato, A.; Chang, P.; Simiu, E. Database-assisted design, standardization and wind direction effects. *J. Strucutral Eng. ASCE* **2001**, *127*, 855–860. [CrossRef]
7. Simiu, E.; Stathopoulos, T. Codification of wind loads on low buildings using bluff body aerodynamics and climatological data base. *J. Wind Eng. Ind. Aerodyn.* **1997**, *69*, 497–506. [CrossRef]
8. Whalen, T.; Simiu, E.; Harris, G.; Lin, J.; Surry, D. The use of aerodynamic databases for the effective estimation of wind effects in main wind-force resisting systems: Application to low buildings. *J. Wind. Eng. Ind. Aerodyn.* **1998**, *77*, 685–693. [CrossRef]
9. Swami, M.V.; Chandra, S. Procedures for Calculating Natural Ventilation Airflow Rates in Buildings. *ASHRAE Res. Proj.* **1987**, *130*. Available online: http://www.fsec.ucf.edu/en/publications/pdf/fsec-cr-163-86.pdf (accessed on 1 April 2022).
10. Muehleisen, R.T.; Patrizi, S. A new parametric equation for the wind pressure coefficient for low-rise buildings. *Energy Build.* **2013**, *57*, 245–249. [CrossRef]
11. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009. [CrossRef]
12. Kotsiantis, S.; Zaharakis, I.; Pintelas, P. Supervised Machine Learning: A Review of Classification Techniques. *Emerg. Artif. Intell. Appl. Comput. Eng.* **2007**, *160*, 3–24.
13. Sun, H.; Burton, H.V.; Huang, H. Machine learning applications for building structural design and performance assessment: State-of-the-art review. *J. Build. Eng.* **2021**, *33*, 101816. [CrossRef]
14. Fu, J.Y.; Li, Q.S.; Xie, Z.N. Prediction of wind loads on a large flat roof using fuzzy neural networks. *Eng. Struct.* **2006**, *28*, 153–161. [CrossRef]
15. Gong, M.; Wang, J.; Bai, Y.; Li, B.; Zhang, L. Heat load prediction of residential buildings based on discrete wavelet transform and tree-based ensemble learning. *J. Build. Eng.* **2020**, *32*, 101455. [CrossRef]
16. Gupta, A.; Badr, Y.; Negahban, A.; Qiu, R.G. Energy-efficient heating control for smart buildings with deep reinforcement learning. *J. Build. Eng.* **2021**, *34*, 101739. [CrossRef]
17. Huang, H.; Burton, H.V. Classification of in-plane failure modes for reinforced concrete frames with infills using machine learning. *J. Build. Eng.* **2019**, *25*, 100767. [CrossRef]
18. Hwang, S.-H.; Mangalathu, S.; Shin, J.; Jeon, J.-S. Machine learning-based approaches for seismic demand and collapse of ductile reinforced concrete building frames. *J. Build. Eng.* **2021**, *34*, 101905. [CrossRef]
19. Naser, S.S.A.; Lmursheidi, H.A. A Knowledge Based System for Neck Pain Diagnosis. *J. Multidiscip. Res. Dev.* **2016**, *2*, 12–18.
20. Sadhukhan, D.; Peri, S.; Sugunaraj, N.; Biswas, A.; Selvaraj, D.F.; Koiner, K.; Rosener, A.; Dunlevy, M.; Goveas, N.; Flynn, D.; et al. Estimating surface temperature from thermal imagery of buildings for accurate thermal transmittance (U-value): A machine learning perspective. *J. Build. Eng.* **2020**, *32*, 101637. [CrossRef]
21. Sanhudo, L.; Calvetti, D.; Martins, J.P.; Ramos, N.M.; Mêda, P.; Gonçalves, M.C.; Sousa, H. Activity classification using accelerometers and machine learning for complex construction worker activities. *J. Build. Eng.* **2021**, *35*, 102001. [CrossRef]
22. Sargam, Y.; Wang, K.; Cho, I.H. Machine learning based prediction model for thermal conductivity of concrete. *J. Build. Eng.* **2021**, *34*, 101956. [CrossRef]
23. Xuan, Z.; Xuehui, Z.; Liequan, L.; Zubing, F.; Junwei, Y.; Dongmei, P. Forecasting performance comparison of two hybrid machine learning models for cooling load of a large-scale commercial building. *J. Build. Eng.* **2019**, *21*, 64–73. [CrossRef]
24. Yigit, S. A machine-learning-based method for thermal design optimization of residential buildings in highly urbanized areas of Turkey. *J. Build. Eng.* **2021**, *38*, 102225. [CrossRef]
25. Yucel, M.; Bekdaş, G.; Nigdeli, S.M.; Sevgen, S. Estimation of optimum tuned mass damper parameters via machine learning. *J. Build. Eng.* **2019**, *26*, 100847. [CrossRef]
26. Zhou, X.; Ren, J.; An, J.; Yan, D.; Shi, X.; Jin, X. Predicting open-plan office window operating behavior using the random forest algorithm. *J. Build. Eng.* **2021**, *42*, 102514. [CrossRef]
27. Wu, P.-Y.; Sandels, C.; Mjörnell, K.; Mangold, M.; Johansson, T. Predicting the presence of hazardous materials in buildings using machine learning. *Build. Environ.* **2022**, *213*, 108894. [CrossRef]
28. Fan, L.; Ding, Y. Research on risk scorecard of sick building syndrome based on machine learning. *Build. Environ.* **2022**, *211*, 108710. [CrossRef]
29. Ji, S.; Lee, B.; Yi, M.Y. Building life-span prediction for life cycle assessment and life cycle cost using machine learning: A big data approach. *Build. Environ.* **2021**, *205*, 108267. [CrossRef]
30. Yang, L.; Lyu, K.; Li, H.; Liu, Y. Building climate zoning in China using supervised classification-based machine learning. *Build. Environ.* **2020**, *171*, 106663. [CrossRef]
31. Olu-Ajayi, R.; Alaka, H.; Sulaimon, I.; Sunmola, F.; Ajayi, S. Building energy consumption prediction for residential buildings using deep learning and other machine learning techniques. *J. Build. Eng.* **2022**, *45*, 103406. [CrossRef]
32. Kareem, A. Emerging frontiers in wind engineering: Computing, stochastics, machine learning and beyond. *J. Wind Eng. Ind. Aerodyn.* **2020**, *206*, 104320. [CrossRef]
33. 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]
34. Chen, Y.; Kopp, G.A.; Surry, D. Interpolation of wind-induced pressure time series with an artificial neural network. *J. Wind Eng. Ind. Aerodyn.* **2002**, *90*, 589–615. [CrossRef]

35. Chen, Y.; Kopp, G.A.; Surry, D. Prediction of pressure coefficients on roofs of low buildings using artificial neural networks. Journal of wind engineering and industrial aerodynamics. *J. Wind Eng. Ind. Aerodyn.* **2003**, *91*, 423–441. [CrossRef]
36. 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]
37. Duan, J.; Zuo, H.; Bai, Y.; Duan, J.; Chang, M.; Chen, B. Short-term wind speed forecasting using recurrent neural networks with error correction. *Energy* **2021**, *217*, 119397. [CrossRef]
38. Fu, J.Y.; Liang, S.G.; Li, Q.S. Prediction of wind-induced pressures on a large gymnasium roof using artificial neural networks. *Comput. Struct.* **2007**, *85*, 179–192. [CrossRef]
39. Gavalda, X.; Ferrer-Gener, J.; Kopp, G.A.; Giralt, F. Interpolation of pressure coefficients for low-rise buildings of different plan dimensions and roof slopes using artificial neural networks. *J. Wind Eng. Ind. Aerodyn.* **2011**, *99*, 658–664. [CrossRef]
40. Hu, G.; Kwok, K.C.S. Predicting wind pressures around circular cylinders using machine learning techniques. *J. Wind Eng. Ind. Aerodyn.* **2020**, *198*, 104099. [CrossRef]
41. Kalogirou, S.; Eftekhari, M.; Marjanovic, L. Predicting the pressure coefficients in a naturally ventilated test room using artificial neural networks. *Build. Environ.* **2003**, *38*, 399–407. [CrossRef]
42. Sang, J.; Pan, X.; Lin, T.; Liang, W.; Liu, G.R. A data-driven artificial neural network model for predicting wind load of buildings using GSM-CFD solver. *Eur. J. Mech. Fluids* **2021**, *87*, 24–36. [CrossRef]
43. Zhang, A.; Zhang, L. RBF neural networks for the prediction of building interference effects. *Comput. Struct.* **2004**, *82*, 2333–2339. [CrossRef]
44. Lamberti, G.; Gorlé, C. A multi-fidelity machine learning framework to predict wind loads on buildings. *J. Wind Eng. Ind. Aerodyn.* **2021**, *214*, 104647. [CrossRef]
45. Lin, P.; Ding, F.; Hu, G.; Li, C.; Xiao, Y.; Tse, K.T.; Kwok, K.C.S.; Kareem, A. Machine learning-enabled estimation of crosswind load effect on tall buildings. *J. Wind Eng. Ind. Aerodyn.* **2022**, *220*, 104860. [CrossRef]
46. Kim, B.; Yuvaraj, N.; Tse, K.T.; Lee, D.-E.; Hu, G. Pressure pattern recognition in buildings using an unsupervised machine-learning algorithm. *J. Wind Eng. Ind. Aerodyn.* **2021**, *214*, 104629. [CrossRef]
47. Hu, G.; Liu, L.; Tao, D.; Song, J.; Tse, K.T.; Kwok, K.C.S. Deep learning-based investigation of wind pressures on tall building under interference effects. *J. Wind Eng. Ind. Aerodyn.* **2020**, *201*, 104138. [CrossRef]
48. Wang, H.; Zhang, Y.-M.; Mao, J.-X.; Wan, H.-P. A probabilistic approach for short-term prediction of wind gust speed using ensemble learning. *J. Wind Eng. Ind. Aerodyn.* **2020**, *202*, 104198. [CrossRef]
49. Tian, J.; Gurley, K.R.; Diaz, M.T.; Fernández-Cabán, P.L.; Masters, F.J.; Fang, R. Low-rise gable roof buildings pressure prediction using deep neural networks. *J. Wind Eng. Ind. Aerodyn.* **2020**, *196*, 104026. [CrossRef]
50. Mallick, M.; Mohanta, A.; Kumar, A.; Patra, K.C. Prediction of Wind-Induced Mean Pressure Coefficients Using GMDH Neural Network. *J. Aerosp. Eng.* **2020**, *33*, 04019104. [CrossRef]
51. Na, B.; Son, S. Prediction of atmospheric motion vectors around typhoons using generative adversarial network. *J. Wind Eng. Ind. Aerodyn.* **2021**, *214*, 104643. [CrossRef]
52. Arul, M.; Kareem, A.; Burlando, M.; Solari, G. Machine learning based automated identification of thunderstorms from anemometric records using shapelet transform. *J. Wind Eng. Ind. Aerodyn.* **2022**, *220*, 104856. [CrossRef]
53. Belle, V.; Papantonis, I. Principles and Practice of Explainable Machine Learning. *Front. Big Data* **2021**, *4*, 688969. [CrossRef] [PubMed]
54. Liang, Y.; Li, S.; Yan, C.; Li, M.; Jiang, C. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. *Neurocomputing* **2021**, *419*, 168–182. [CrossRef]
55. Roscher, R.; Bohn, B.; Duarte, M.F.; Garcke, J. Explainable Machine Learning for Scientific Insights and Discoveries. *IEEE Access* **2020**, *8*, 42200–42216. [CrossRef]
56. Xu, F.; Uszkoreit, H.; Du, Y.; Fan, W.; Zhao, D.; Zhu, J. Explainable AI: A Brief Survey on History, Research Areas, Approaches and Challenges. In *Natural Language Processing and Chinese Computing*; Springer: Cham, Switzerland, 2019; pp. 563–574. [CrossRef]
57. Meddage, D.P.; Ekanayake, I.U.; Weerasuriya, A.U.; Lewangamage, C.S.; Tse, K.T.; Miyanawala, T.P.; Ramanayaka, C.D. Explainable Machine Learning (XML) to predict external wind pressure of a low-rise building in urban-like settings. *J. Wind Eng. Ind. Aerodyn.* **2022**, *226*, 105027. [CrossRef]
58. Lundberg, S.M.; Lee, S.-I. A unified approach to interpreting model predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 4–9 December 2017; pp. 4768–4777.
59. Shrikumar, A.; Greenside, P.; Kundaje, A. Learning important features through propagating activation differences. In Proceedings of the 34th International Conference on Machine Learning—Volume 70, Sydney, NSW, Australia, 6–11 August 2017; pp. 3145–3153.
60. Ribeiro, M.T.; Singh, S.; Guestrin, C. 'Why Should I Trust You?': Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–16 August 2016; pp. 1135–1144. [CrossRef]
61. Petsiuk, V.; Das, A.; Saenko, K. RISE: Randomized Input Sampling for Explanation of Black-Box Models. Available online: http://arxiv.org/abs/1806.07421 (accessed on 11 April 2021).
62. Moradi, M.; Samwald, M. Post-hoc explanation of black-box classifiers using confident itemsets. *Expert Syst. Appl.* **2021**, *165*, 113941. [CrossRef]

63. Yap, M.; Johnston, R.L.; Foley, H.; MacDonald, S.; Kondrashova, O.; Tran, K.A.; Nones, K.; Koufariotis, L.T.; Bean, C.; Pearson, J.V.; et al. Verifying explainability of a deep learning tissue classifier trained on RNA-seq data. *Sci. Rep.* **2021**, *11*, 1–2. [CrossRef]
64. Patel, H.H.; Prajapati, P. Study and Analysis of Decision Tree Based Classification Algorithms. *Int. J. Comput. Sci. Eng.* **2018**, *6*, 74–78. [CrossRef]
65. Ahmad, M.W.; Mourshed, M.; Rezgui, Y. Trees vs Neurons: Comparison between random forest and ANN for high-resolution prediction of building energy consumption. *Energy Build.* **2017**, *147*, 77–89. [CrossRef]
66. Breiman, L.; Friedman, J.; Olshen, R.; Stone, C.J. *Classification and Regression Trees*; Routledge: Oxfordshire, UK, 1983. [CrossRef]
67. Xu, M.; Watanachaturaporn, P.; Varshney, P.K.; Arora, M.K. Decision tree regression for soft classification of remote sensing data. *Remote Sens. Environ.* **2005**, *97*, 322–336. [CrossRef]
68. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
69. Chakraborty, D.; Elzarka, H. Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold. *Energy Build.* **2019**, *185*, 326–344. [CrossRef]
70. Mo, H.; Sun, H.; Liu, J.; Wei, S. Developing window behavior models for residential buildings using XGBoost algorithm. *Energy Build.* **2019**, *205*, 109564. [CrossRef]
71. Xia, Y.; Liu, C.; Li, Y.; Liu, N. A boosted decision tree approach using Bayesian hyper-parameter optimization for credit scoring. *Expert Syst. Appl.* **2017**, *78*, 225–241. [CrossRef]
72. Zięba, M.; Tomczak, S.K.; Tomczak, J.M. Ensemble boosted trees with synthetic features generation in application to bankruptcy prediction. *Expert Syst. Appl.* **2016**, *58*, 93–101. [CrossRef]
73. Maree, R.; Geurts, P.; Piater, J.; Wehenkel, L. A Generic Approach for Image Classification Based On Decision Tree Ensembles And Local Sub-Windows. In Proceedings of the 6th Asian Conference on Computer Vision, Jeju, Korea, 27–30 January 2004; pp. 860–865.
74. Okoro, E.E.; Obomanu, T.; Sanni, S.E.; Olatunji, D.I.; Igbinedion, P. Application of artificial intelligence in predicting the dynamics of bottom hole pressure for under-balanced drilling: Extra tree compared with feed forward neural network model. *Petroleum* **2021**. [CrossRef]
75. Sagi, O.; Rokach, L. Explainable decision forest: Transforming a decision forest into an interpretable tree. *Inf. Fusion* **2020**, *61*, 124–138. [CrossRef]
76. John, V.; Liu, Z.; Guo, C.; Mita, S.; Kidono, K. Real-Time Lane Estimation Using Deep Features and Extra Trees Regression. In *Image and Video Technology*; Springer: Cham, Switzerland, 2015; pp. 721–733. [CrossRef]
77. Seyyedattar, M.; Ghiasi, M.M.; Zendehboudi, S.; Butt, S. Determination of bubble point pressure and oil formation volume factor: Extra trees compared with LSSVM-CSA hybrid and ANFIS models. *Fuel* **2020**, *269*, 116834. [CrossRef]
78. Cai, J.; Li, X.; Tan, Z.; Peng, S. An assembly-level neutronic calculation method based on LightGBM algorithm. *Ann. Nucl. Energy* **2021**, *150*, 107871. [CrossRef]
79. Fan, J.; Ma, X.; Wu, L.; Zhang, F.; Yu, X.; Zeng, W. Light Gradient Boosting Machine: An efficient soft computing model for estimating daily reference evapotranspiration with local and external meteorological data. *Agric. Water Manag.* **2019**, *225*, 105758. [CrossRef]
80. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Available online: https://www.microsoft.com/en-us/research/publication/lightgbm-a-highly-efficient-gradient-boosting-decision-tree/ (accessed on 11 April 2021).
81. Ebtehaj, I.H.; Bonakdari, H.; Zaji, A.H.; Azimi, H.; Khoshbin, F. GMDH-type neural network approach for modeling the discharge coefficient of rectangular sharp-crested side weirs. *Int. J. Eng. Sci. Technol.* **2015**, *18*, 746–757. [CrossRef]
82. NIST Aerodynamic Database. Available online: https://www.nist.gov/el/materials-and-structural-systems-division-73100/nist-aerodynamic-database (accessed on 23 January 2022).
83. Tokyo Polytechnic University (TPU). Aerodynamic Database for Low-Rise Buildings. Available online: http://www.wind.arch.t-kougei.ac.jp/info_center/windpressure/lowrise (accessed on 1 April 2022).
84. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
85. Liu, H. *Wind Engineering: A Handbook for Structural Engineers*; Prentice Hall: Hoboken, NJ, USA, 1991.
86. Saathoff, P.J.; Melbourne, W.H. Effects of free-stream turbulence on surface pressure fluctuations in a separation bubble. *J. Fluid Mech.* **1997**, *337*, 1–24. [CrossRef]
87. Akon, A.F.; Kopp, G.A. Mean pressure distributions and reattachment lengths for roof-separation bubbles on low-rise buildings. *J. Wind Eng. Ind. Aerodyn.* **2016**, *155*, 115–125. [CrossRef]