Wind-induced pressure prediction using quasi-steady theory and few-shot learning

Yanmo Weng, Stephanie German Paal
Zachry Department of Civil and Environmental Engineering, Texas A&M University, College Station, TX 77843, United States
yanmow@tamu.edu

Abstract. This research introduces an approach to predict the wind pressures for full-scale low-rise buildings using few-shot learning and quasi-steady theory. The proposed few-shot learning model can extrapolate wind pressure predictions from the model-scale to full-scale measurements. The extrapolation capability is typically beyond what is expected of conventional machine learning techniques such as artificial neural networks, random forests, and support vector machines. In addition, the quasi-steady theory-based model is applied to generate the accurate estimation of wind pressure coefficient data under unmeasured wind directions. The generated data helps improve the few-shot learning prediction performance. In this research, 3% of the full-scale measurements were used for training the model, and the remaining 97% of data points were used to test the prediction performance. The proposed model shows a better performance than the baseline model by reaching MSE and R-squared values equal to 0.047 and 0.740, respectively.

1. Introduction

In North America, low-rise light-frame buildings are one of the most common and most vulnerable building types under extreme wind events (He et al. 2017, van de Lindt and Dao, 2009). Therefore, proper evaluation of wind pressures on the roofs of these structures is important. Wind tunnel tests are commonly used to estimate wind pressures for low-rise buildings. By conducting wind tunnel tests, it is possible to accurately simulate realistic wind conditions and calculate surface pressure estimations. However, when applying the wind tunnel test results to a full-scale building, the scaling issue cannot be ignored. Specifically, the wind tunnel tests could underestimate the wind-induced pressures on the roof corner and edge zones (Cheung et al. 1997, Tieleman et al. 1981). The mismatch of wind pressures between the field measurements and small-scale wind tunnel tests sometimes exceeds 50% (Wang et al. 2020a, Coffman et al. 2010). To obtain a sufficiently high Reynolds number to capture realistic airflow characteristics, a large-scale model is usually required (Moravej, 2018). However, it is a time-consuming and expensive process to build a large-scale model and place high-resolution measuring taps. To reduce the reliance on experiments, computational fluid dynamics (CFD) (Aly and Gol-Zaroudi, 2020), parametric studies (Muehleisen and Patrizi, 2013), and Partial Turbulence Simulations (PTS) (Estephan et al. 2021) have all been used to simulate the wind pressures on low-rise buildings. In addition to these techniques, in more recent years, machine learning (ML) has become a more popular method for predicting wind pressure coefficients with low reliance on experimental tests or costly computational simulations.

The state-of-the-art ML techniques mainly focus on predicting wind pressures based on the existing wind tunnel test datasets. For a typical ML algorithm (e.g., random forest, artificial neural network), training and testing sets usually have a similar joint probability distribution $p(x, y)$ over the predictors $\boldsymbol{X}$ and target variable(s) $\boldsymbol{y}$. To satisfy this assumption, the training and testing datasets are usually randomly split in a ratio of 7:3. Therefore, both the training and test data points will be sampled from the wind tunnel dataset in any traditional ML algorithm. For example, Gavalda et al. (2011) trained an artificial neural network (ANN) model based on the wind tunnel data. The trained ANN model can interpolate the prediction for unmeasured wind directions that were within the range of the measured wind directions included in the dataset. Lang et al. (2021) used the random forest algorithm to predict wind pressure coefficients for high-rise buildings. They tested the model's performance for interpolating the wind pressures for measuring taps where the coordinates are within the range of the surface coordinates of the training set. Weng and Paal (2022) used the gradient-boosting decision tree to predict the wind pressures for non-isolated low-rise buildings based on the wind tunnel data. However, model scale is not considered as one of the predictors in this approach. In short, these state-of-the-art ML applications for wind pressure prediction might not be able to solve the scaling issue. Even if the prediction is sufficiently accurate, there still exists a mismatch between model-scale and full-scale (and thus, real-world) prediction. Recently, Weng and Paal (2023) proposed a few-shot learning approach to extrapolate the wind pressures from model-scale to full-scale measurements. However, only the wind pressures of roof corner taps are analyzed, and the prediction accuracy is not high. In short, there still exists a gap in using ML techniques to predict the wind pressures for full-scale experiments based on the model-scale wind tunnel tests.

To arrive at a highly accurate predictive model, capable of extrapolating from model-scale data to full-scale specimens, this research incorporates few-shot learning with a parametric study based on quasi-steady theory. Few-shot learning (FSL) is an ML approach that trains the machine to learn a new, similar task using a limited number of labeled data samples (Fink, 2004). In this study, the FSL model first trains a meta-learner based on a large amount of model-scaled wind tunnel data points. Then, the trained model can be quickly adapted to full-scale prediction with only a few full-scale experimental data points used in the training dataset. The key to successfully predicting the wind pressures for the full-scale buildings is to have a good, well-trained FSL meta-learner. The quasi-steady (QS) model will be used to interpolate the wind pressures under unmeasured wind directions for the model-scale dataset. The interpolated data points will then expand the dataset used for training the FSL meta-learner and finally reach a better prediction performance for full-scale measurements.

The novelty of this research could be summarized as follows:

- It is the first time that quasi-steady theory is integrated with ML techniques in the wind engineering field.
- It is the first time that a few-shot learning technique is used to predict the wind pressures over the entire roof area.

The remainder of this manuscript is organized as follows: Section 2 introduces the background of the FSL technique implemented in this work. Section 3 introduces the datasets and the framework of the proposed QS theory-enhanced FSL model. Section 4 shows the prediction results. Section 5 presents the conclusions and future work.

2. Theoretical background

Few-shot learning is a class of machine learning techniques which can make predictions based on only a limited number of data samples (Wang et al. 2020b). Meta-learning is the most widely-used framework for few-shot learning (Chen et al. 2021). Therefore, this research chooses meta-learning as the approach to achieve the goal of few-shot learning. Meta-learning also called "learning to learn", aims to train a meta-learner that can efficiently adapt to a similar unseen new task using limited data. Meta-learning has been widely used to extrapolate the prediction to out-of-distribution tasks (Finn et al. 2017, Alajaji and Alhichri, 2020). Model-Agnostic Meta-Learning (MAML) and "Reptile" are the most commonly used meta-learning algorithms proposed by Finn et al. (2017) and Nichol et al. (2018), respectively. Both algorithms sample the dataset into many small tasks used to train a meta-learner. Then several gradient decent steps are applied based on the limited number of labeled unseen data samples. After the gradient descent steps, the updated learner should be able to efficiently predict unseen data. Compared to MAML, Reptile has two advantages. Firstly, Reptile is more computationally efficient. In MAML, the second-order derivative needs to be calculated in the gradient decent steps which is computationally expensive. However, the Reptile algorithm bypasses the need to compute the second-order derivative terms while still maintaining good prediction performance. Secondly, the Reptile algorithm is easier to implement since there is no need to split the task into query-support sets. Therefore, for this research, the Reptile algorithm is selected as the few-shot learning algorithm. In order to prevent any confusion, throughout the remaining part of this manuscript, the term "few-shot learning" will specifically refer to the Reptile algorithm.

There are three sets of data used in any meta-learning problem: the training set, meta-training set, and meta-testing set. Assume there are two dataset sources (i.e., D_1 and D_2) with different distributions, where dataset D_2 is the dataset of interest. The first dataset D_1 serves as the training set. The second dataset D_2 is divided into two parts: the meta-training set ($D_{2,tr}$), and the meta-testing set ($D_{2,te}$). Specifically, a very small portion of the second dataset D_2 are randomly selected as the meta-training set $D_{2,tr}$. The remaining large portion of D_2 serves as the meta-testing set $D_{2,te}$ which will be employed to comprehensively evaluate the performance of the model on the dataset of interest. Corresponding to these three sets are two training phases and a testing phase. The first training phase is named the "training stage" while the second training phase is designated the "meta-training stage". Figure 1 illustrates the two-stage training procedure. In the first iteration of the first training stage, a random initial model parameter ϕ^0 is assigned. The first dataset D_1 is randomly split into n number of small tasks (τ_1, τ_2, …, τ_n). This sampling procedure is represented by the dashed green arrows in Figure 1. In the first training stage, model parameters are first updated via gradient decent steps on the sampled small tasks. This model parameter update direction is called the "inner-loop" update direction which is shown by the solid blue arrows in Figure 1. In other words, in the inner-loop update direction, the model parameters update in the order of $\phi^0 \rightarrow \phi^{\tau_1} \rightarrow \cdots \rightarrow \phi^{\tau_n}$. After the inner-loop update, the model parameters are updated in the direction connecting ϕ^0 and ϕ^{τ_n}. This direction is termed the "outer-loop" update direction and is shown by the black solid arrows in Figure 1. The red solid arrow in Figure 1 represents the final model parameter update direction for each iteration. The final update direction is the same as the outer-loop update direction but applies an additional learning rate to avoid the over-fitting issue. Then the obtained final model parameter of the first iteration ($\tilde{\phi}^1$) serves as the initial model parameter of the next iteration. After performing the same procedure for all m iterations, the first training stage is finished and $\tilde{\phi}^m$ is obtained as the vector of model parameters for the meta-learner.

Similar to the first training stage, in the meta-training stage, the meta-training set $D_{2,tr}$ is randomly split into n small tasks ($\tau_{meta,1}$, $\tau_{meta,2}$, …, $\tau_{meta,n}$). The final model parameters $\tilde{\phi}^m$ obtained from the first training stage serves as the initial set of model parameters for the meta-training stage. In the meta-training stage, the model parameters are only updated along the inner-loop direction via training on the sampled small tasks. The red circle in Figure 1 represents the last set of updated model parameters ($\tilde{\phi}^f$) which will be used to assess the prediction performance on the meta-testing set. The goal for the first training stage is to train a meta-learner that can understand the general information across different tasks from D_1. Then, the goal of the meta-training stage is to fine-tune or optimize the trained learner based on limited data points, so that it can quickly adapt to a new unseen task or distribution. The objective function of this optimization problem is shown as follows (Eq. 1):

$$argmin_{\phi} E_{\tau}[\mathcal{L}_{\tau}(U_{\tau}^{k}(\phi))] \tag{1}$$

Where $\mathcal{L}_{\tau}(U_{\tau}^{k}(\phi))$ is the loss for task τ after k stochastic gradient descent (SGD) steps.

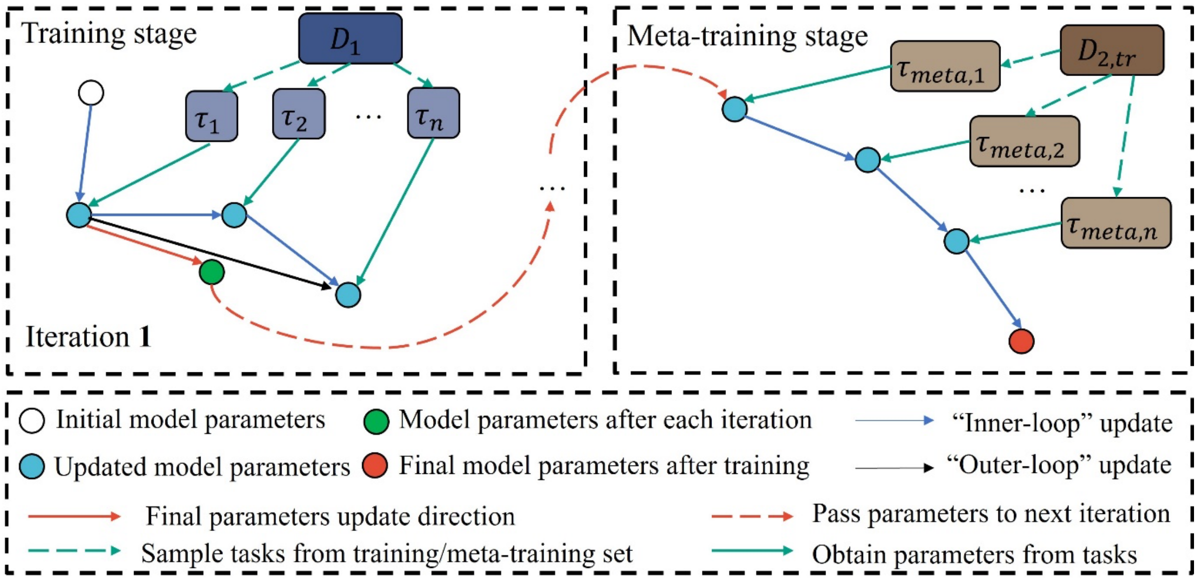

Figure 1: Two-stage training procedure for meta-learning

3. Dataset and framework

There are two datasets used in the research. The first dataset is the Wall of Wind (WOW) dataset (Chowdhury et al., 2017) which has a series of large-scale wind tunnel experiments with five different model scales (1:100, 1:50, 1:20, 1:10, and 1:6). The second dataset is the full-scale test data from the Texas Tech University (TTU) aerodynamic dataset (Smith et al., 2018). The WOW and TTU datasets both have the same building geometry and measurement tap layout. Also, both datasets use the wind pressure coefficients (C_p) to represent the wind-induced pressure as follows (Eq. 2):

$$C_p(i,t) = \frac{p(i,t)}{0.5\rho U_{mean}^2} \tag{2}$$

Where $C_p(i,t)$ is the wind pressure coefficient of tap i at time t; $p(i,t)$ is the wind pressure of tap i at time t; ρ is the air density; and U_{mean} is the wind speed measured at the mean roof height.

The predictor of the proposed model includes the wind characteristics (i.e., wind speed, turbulence intensity, wind direction), scale, and tap coordinate information. The tap coordinate information is converted into dimensionless variables to avoid the influence of scale changes. Specifically, the centroid of the entire roof area is set as the origin point. The tap coordinate is represented by X/H and Y/H. H denotes the building height while X and Y represent the distance between the chosen tap and the origin position along the width and length of the building. The possible values of each selected predictor are demonstrated in Table 1. All the variables in Table 1 are numeric values. All the scale values are converted to float values during the data pre-processing stage. For example, the 1:100 scale is converted to 0.01 and the 1:50 scale is converted to 0.02. In total, the WOW dataset contains 28,523 data points from the wind tunnel experiments. The TTU dataset contains 2,430 data points from the full-scale measurements.

Table 1: Values of selected predictors.

Predictors	Potential values
X/H	0, ±0.39, ±0.77, ±0.96, ±1.15
Y/H	±0.19, ±0.57, ±0.95, ±1.33, ±1.71
Scale	1:100, 1:50, 1:20, 1:10, 1:6, 1:1
Wind direction	0° to 195° at a 3° interval
Wind speed (U_{mean})	7.66, 13.62, 14.8, 18.3, 19.9, 20.7 (m/s)
Turbulence intensity (I_u)	0.102, 0.110, 0.181, 0.197, 0.199, 0.216

Although the few-shot learning model only needs a very limited number of data samples from the second dataset in the training process, it still needs plenty of data samples from the first dataset to train the meta-learner. One method for expanding the dataset is to generate wind pressures using parametric equations derived from existing empirical or mechanical relationships. Wu and Kopp (2016) considered the wind direction as the most significant factor influencing the wind pressure coefficients. They proposed a QS theory-based wind pressure coefficient estimation equation to represent the correlation between wind direction and wind pressure coefficients using the following Fourier series (Eq. 3):

$$C_p(\theta) = \sum_{k=0}^{N_1} a_{1k} \cos(k\theta) + b_{1k} \sin(k\theta) \tag{3}$$

Where θ is the wind direction; $C_p(\theta)$ is the wind pressure coefficient under wind direction θ; and a_{1k} and b_{1k} are Fourier coefficients.

The QS estimation equation is able to estimate the instantaneous area-averaged roof surface pressures considering the variation of magnitudes and wind directions measured at the reference location. The QS estimation equation is derived based on several physical assumptions. Wu and Kopp (2019) explained the physical assumptions based on simple algebraic manipulations of the time-averaged integral momentum equation for turbulence flow. Some studies have indicates that the QS estimation equation can reasonably compute the mean pressure coefficient (Richards and Hoxey, 2004, Wu and Kopp, 2016, 2019). To apply the QS model equation in this approach, the WOW dataset is first split into many subsets grouped by the scale and tap coordinate information. Therefore, in each subset, the wind pressure coefficient varies with the wind direction. Then, the number of terms for the Fourier coefficients (i.e., k) is set as seven. In other words, there are seven terms for a_{1k} and b_{1k}. By fitting the data points in the subset, a unique group of Fourier coefficients is produced for each subset. Then, the fitted QS theory-based estimation equation can be used to expand the dataset. Specifically, the WOW dataset includes data extracted at 3° intervals in the wind direction. With the updated estimation equation, this interval decreases from 3° to 1° by evenly generating data points for each subset. The WOW dataset contains 28,523 data points in total, and the QS theory-based estimation equation adds 31,500 data points to this dataset. Then, adding the data samples generated from the QS estimation equation can help the model construct a better learner. With a well-trained learner, the prediction for the out-of-distribution data will potentially be more accurate.

The initial WOW dataset and QS model generated dataset are combined and then used as the first dataset D_1 (as denoted in the previous section and Figure 1). Then, a small portion of the TTU dataset serves as the meta-training set $D_{2,tr}$, and the remaining large portion of the TTU dataset is used as the meta-testing set $D_{2,te}$. Different from the previous work presented by Weng and Paal (2023), the data points in the meta-training set are only selected from a limit number (n) of taps (rather than all the taps). The selected number of taps (n) is a predefined number that ranges from one to 90 (i.e., all the taps). In this study, the data points in the meta-training sets are drawn from 20 taps (i.e., $n = 20$) for demonstration purposes. These 20 taps are chosen from the center and endpoints of each row of taps on the roof surface; then the data points in the meta-training set are randomly selected from these taps. All the unselected data points form the meta-testing set. The benefit of limiting the data points to be chosen only from the pre-selected taps is that no information about points other than the selected taps is required. Only a very low resolution of measuring taps is required for the chosen locations when implementing the proposed model in real-world scenarios (and thus, we can greatly reduce the complexity of future testing regimens). The detailed framework of the proposed algorithm is shown in Figure 2. Before splitting the dataset, the Max-min scaler function is first performed to scale each feature in the dataset. In this work, 20 out of 90 taps are first selected. Then 80 data points from these 20 taps are randomly selected. In other words, the size of the meta-training set is 80, which represents only 3% of the total data in the TTU dataset. As shown in Figure 2, after splitting the training set, meta-training set, and meta-testing set, the training set is first applied to train a meta-learner. In this work, the multilayer perceptron (MLP) is selected as the meta-learner. The mean squared error (MSE) loss is selected as the loss function. When the two-stage training process is completed, the meta-testing set is applied to test the prediction performance on the full-scale measurements.

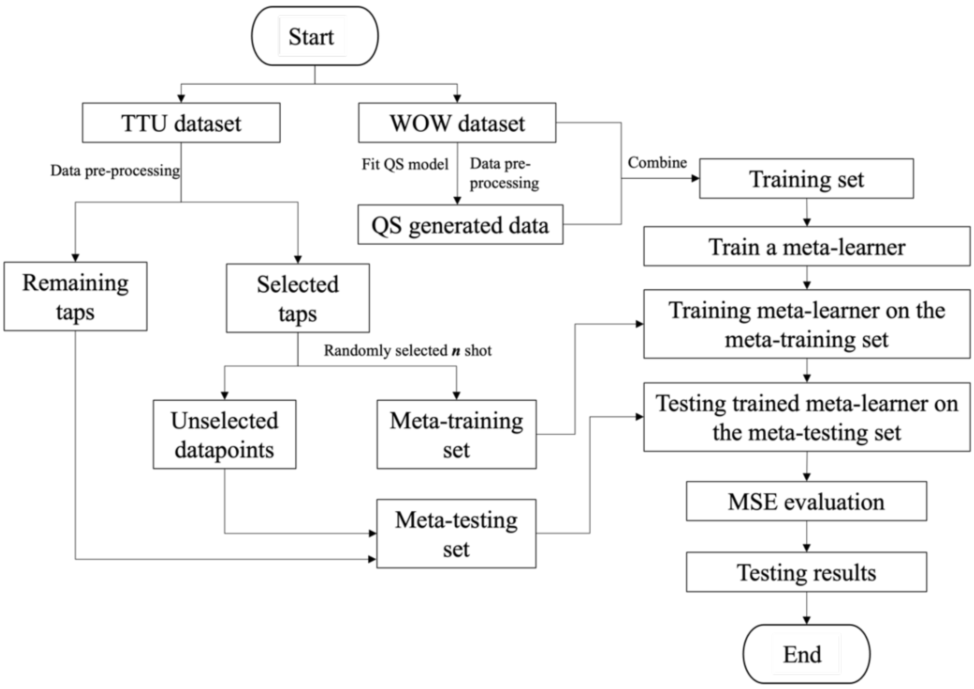

Figure 2: Framework of the proposed few-shot learning model

4. Results and discussion

The mean-squared error (MSE) and the coefficient of determination (R^2) are selected as the evaluation metrics. A smaller MSE value indicates a better prediction performance while a higher R^2 value represents a higher correlation between predicted and observed values. The final goal of this work is to improve the prediction performance of the few-shot learning model by incorporating QS-model-generated data points. Therefore, the previous work presented by Weng and Paal (2023) is employed as the baseline model for comparison. The baseline model and the proposed few-shot learning model utilize the same 80 data points as the meta-training set, and the meta-testing set is consistent across these two models. Compared to the baseline model, the proposed model has a larger training set, with the inclusion of the QS-model-generated data points. The mean wind pressure coefficient ($C_{p,mean}$) is selected as the response variable and the predictors are mentioned in Table 1. The proposed few-shot learning model is implemented in Python on a Windows system with a 2.60 GHz Intel Core i7-9750H processor. The training and testing process for the baseline model and proposed few-shot learning model take approximately 600 and 630 minutes to complete, respectively. The MSE and R^2 metrics are defined in Eq. 4 and Eq. 5 as follows:

$$MSE = \frac{1}{N}\sum_i^N (y_{obs,i} - y_{pred,i})^2 \tag{4}$$

$$R^2 = 1 - \frac{\sum_i^N (y_{obs,i} - y_{pred,i})^2}{\sum_i^N (y_{obs,i} - \widehat{y_{obs}})^2} \tag{5}$$

Where N denotes the size of the meta-testing set; $y_{obs,i}$ and $y_{pred,i}$ denote the observed ground truth and model prediction for ith data in the meta-testing set, respectively.

The MSE and R^2 values for the baseline model are equal to 0.057 and 0.685, respectively. The proposed QS theory did in fact enhance the few-shot learning model which is represented by the MSE value equal to 0.047 and R^2 value equal to 0.740. The scatter plots of the relationship between prediction and observation for the meta-testing set are shown in Figure 3. The solid red line in Figure 3 (a) and (b) signifies the zero-error line which means the prediction equals observation. Since the negative sign for the $C_{p,mean}$ values only represents the direction of the pressure, the proposed model overpredicts the $C_{p,mean}$ if the point lies above the red line. On the contrary, if the point is located below the red line, the proposed model underpredicts the $C_{p,mean}$ value. Several observations can be drawn by comparing Figure 3 (a) and (b). First, the proposed QS-theory-enhanced few-shot learning model has a higher R^2 and lower MSE than the baseline model. The increase in prediction accuracy is most noticeable for data points with observation values between -2 and 0. These data points are closer to the zero-error line compared to the prediction in the baseline model. Secondly, when compared to the baseline model, the proposed model does not improve the prediction performance of extreme value cases. Specifically, Figure 3 (a) and (b) show that the predictions are very close for data points with observations greater than 0 or less than -3. These data points are from the roof corners which usually experience the highest wind pressures. One possible explanation for the poor prediction for these data points is that there is still a lack of sufficient extreme value data points accounted for in the training sets. Even after generating the data points to expand the training set, the amount of data points that have extreme values is still not large enough for the meta-learner to ascertain the wind pressure performance. Overall, except for these extreme values cases, there is still a noticeable improvement in the proposed model. Since the majority of the data points shown in Figure 3 (b) are close to the zero-error line and the prediction errors mainly come from the data points with extremely high wind pressure coefficient values, it is reasonable to conclude that the proposed model is reliable for predicting wind pressure for full-scale buildings in most cases. However, further improvement is needed to handle extreme value cases.

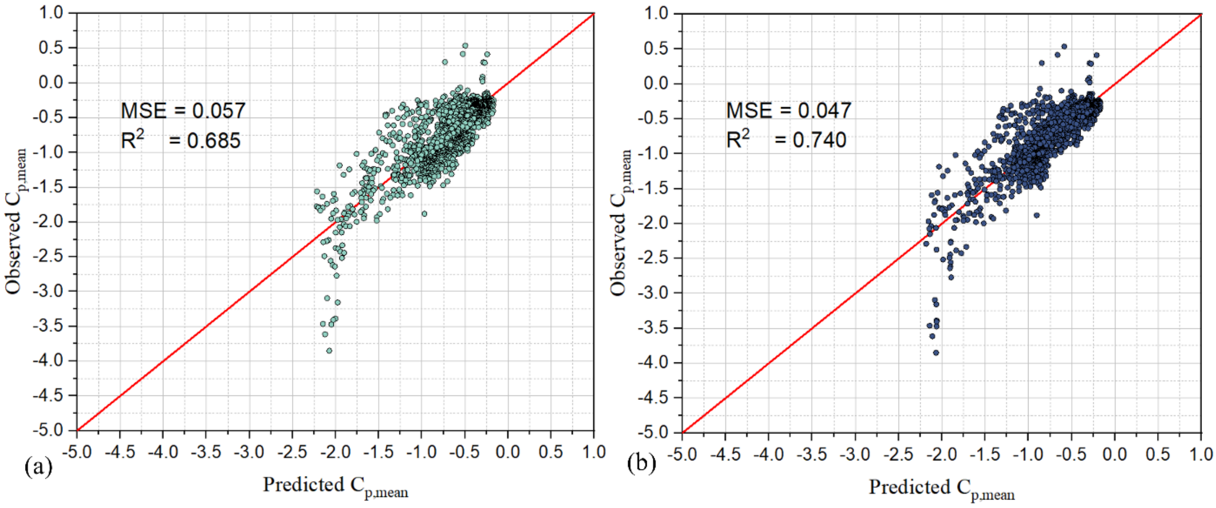

Figure 3: Mean wind pressure coefficient prediction: (a). Baseline; (b). QS theory enhanced few-shot learning.

5. Conclusions

This work presents a novel ML technique to predict wind pressures on full-scale buildings based on model-scale wind tunnel experiments. The QS theory-based estimation equation is incorporated to generate reliable data points for training the meta-learner of the few-shot learning model. Expanding the dataset using the QS theory-based estimation equation can be meaningful for this application. When compared to simple linear or quadratic interpolation, the QS theory-based estimation equation provides a more reasonable way to generate convincing wind pressure coefficients. Also, a baseline few-shot learning model is employed for comparison. With just 3% of the TTU data included in the two-stage training procedure, the baseline model predicts the mean wind pressure coefficients with an MSE equal to 0.057 and R^2 value equal to 0.685. With the help of the QS theory-based estimation equation, the proposed model shows a better prediction performance with an MSE equal to 0.047 and R^2 value equal to 0.740 (increases of 17.5% and 8%, respectively). Even though this improvement is small, it is still meaningful. Because it can reduce the reliance on physical tests by generating data points with reasonable wind pressure coefficients. Furthermore, the QS theory-based estimation equation is easily integrated with other methods that improve predictions in ways other than expanding the dataset (e.g., improving the performance via optimizing the model architectures). In conclusion, although the scaling issue causes a mismatch in wind pressure of full-scale and scaled model buildings in some cases, the proposed model yields satisfying performance with only 3% of TTU data involves in training.

However, there are still limitations that should be addressed in future work. First, the predictions for the extreme value cases are not great. One potential solution could be integrating additional physics-based information into the few-shot learning model. Specifically, in the proposed model, the data sample size for extreme values cases is still not sufficiently large for the meta-learner to deduce the difference in the behavior of the wind at the center from that of the corners of the roof. Therefore, the physical knowledge from design codes and standards could be helpful in clustering the whole roof area into several small areas. Then, implementing the proposed few-shot learning model for each small roof area might show a better prediction. Second, the WOW and TTU datasets used in this work have the same building geometry, therefore the trained few-shot learning model can only reliably be used to extrapolate the wind pressure from the scaled model to the full-scale building which has the same geometry as the WOW dataset. When comparing the real-world gable or hip roof building, this building geometry might be too simple. To make this few-shot learning model have a broader impact as an application, multiple model-scale datasets with different building geometries can be used in the training set at the first training stage as a future improvement.

Acknowledgments

The authors gratefully acknowledge the support in the form of data sharing and general advice from Dr. Chowdhury's research group at Florida International University.

References

Alajaji, D. A. & Alhichri, H. (2020) Few Shot Scene Classification in Remote Sensing using Meta-Agnostic Machine. In 2020 6th Conference on Data Science and Machine Learning Applications pp. 77-80.

Aly, A. M. & Gol-Zaroudi, H. (2020) Peak pressures on low rise buildings: CFD with LES versus full scale and wind tunnel measurements. Wind and Structures 30(1), pp. 99.

Chen, Y., Liu, Z., Xu, H., Darrell, T. & Wang, X. (2021) Meta-baseline: Exploring simple meta-learning for few-shot learning. pp. 9062-9071.

Cheung, J. C. K., Holmes, J. D., Melbourne, W. H., Lakshmanan, N. & Bowditch, P. (1997) Pressures on a 110 scale model of the Texas Tech Building. Journal of Wind Engineering and Industrial Aerodynamics 69-71, pp. 529-538.

Chowdhury, A. G., Zisis, I., Irwin, P., Bitsuamlak, G., Pinelli, J.-P., Hajra, B. & Moravej, M. (2017) Large-Scale Experimentation Using the 12-Fan Wall of Wind to Assess and Mitigate Hurricane Wind and Rain Impacts on Buildings and Infrastructure Systems. Journal of Structural Engineering 143(7), pp. 04017053.

Coffman, B. F., Main, J. A., Duthinh, D. & Simiu, E. (2010) Wind Effects on Low-Rise Metal Buildings: Database-Assisted Design versus ASCE 7-05 Standard Estimates. Journal of Structural Engineering 136(6), pp. 744-748.

Estephan, J., Feng, C., Gan Chowdhury, A., Chavez, M., Baskaran, A. & Moravej, M. (2021) Characterization of wind-induced pressure on membrane roofs based on full-scale wind tunnel testing. Engineering Structures 235, pp. 112101.

Fink, M. (2004) Object classification from a single example utilizing class relevance metrics. Advances in neural information processing systems 17.

Finn, C., Abbeel, P. & Levine, S. (2017) Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. In Proceedings of Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 1126--1135.

Gavalda, X., Ferrer-Gener, J., Kopp, G. A. & Giralt, F. (2011) Interpolation of pressure coefficients for low-rise buildings of different plan dimensions and roof slopes using artificial neural networks. Journal of Wind Engineering and Industrial Aerodynamics 99(5), pp. 658-664.

He, J., Pan, F. & Cai, C. S. (2017) A review of wood-frame low-rise building performance study under hurricane winds. Engineering Structures 141, pp. 512-529.

Lang, L., Tiancai, L., Shan, A. & Xiangyan, T. (2021) An improved random forest algorithm and its application to wind pressure prediction. International Journal of Intelligent Systems 36(8), pp. 4016-4032.

Moravej, M. (2018) Investigating scale effects on analytical methods of predicting peak wind loads on buildings.

Muehleisen, R. T. & Patrizi, S. (2013) A new parametric equation for the wind pressure coefficient for low-rise buildings. Energy and Buildings 57, pp. 245-249.

Nichol, A., Achiam, J. & Schulman, J. (2018) On first-order meta-learning algorithms. arXiv preprint arXiv:1803.02999.

Richards, P. J. & Hoxey, R. P. (2004). Quasi-steady theory and point pressures on a cubic building. Journal of wind engineering and industrial aerodynamics, 92(14-15), pp. 1173-1190.

Smith, D., Mehta, K. & Morse, S. (2018) Wind Engineering Research Field Laboratory Selected Data Sets for Comparison to Model-Scale, Full-Scale and Computational Fluid Dynamics Simulations. DesignSafe-CI.

Tieleman, H. W., Akins, R. E. & Sparks, P. R. (1981) A comparison of wind-tunnel and full-scale wind pressure measurements on low-rise structures. Journal of Wind Engineering and Industrial Aerodynamics 8(1), pp. 3-19.

Van De Lindt, J. W. & Dao, T. N. (2009) Performance-based wind engineering for wood-frame buildings. Journal of Structural Engineering 135(2), pp. 169-177.

Wang, X. J., Li, Q. S. & Yan, B. W. (2020a) Full-Scale Measurements of Wind Pressures on a Low-Rise Building during Typhoons and Comparison with Wind Tunnel Test Results and Aerodynamic Database. Journal of Structural Engineering 146(10), pp. 04020196.

Wang, Y., Yao, Q., Kwok, J. T. & Ni, L. M. (2020b) Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys 53(3), pp. 1-34.

Weng, Y. & Paal, S. G. (2022) Machine learning-based wind pressure prediction of low-rise non-isolated buildings. Engineering Structures 258, pp. 114148.

Weng, Y. & Paal, S. G. (2023) Extrapolation of wind pressure for low-rise buildings at different scales using few-shot learning. Wind and Structures, in production.

Wu, C.-H. & Kopp, G. A. (2016) Estimation of wind-induced pressures on a low-rise building using quasi-steady theory. Frontiers in Built Environment 2, pp. 5.

Wu, C.-H. & Kopp, G. A. (2019) Examination of the physical assumptions of a quasi-steady vector model using the integral momentum equation. Journal of Wind Engineering and Industrial Aerodynamics 187, pp. 73-84.