CS230

Multi-fidelity neural network for predicting wind loads on high-rise buildings

Mattia Fabrizio Ciarlatani
Civil and Environmental Engineering
Stanford University
mattiafc@stanford.edu

Themistoklis Vargiemezis
Civil and Environmental Engineering
Stanford University
tvarg@stanford.edu

Abstract

Large-eddy simulations (LES) can provide accurate predictions but their computational cost is prohibitively high for engineering use when wind loading at multiple wind directions need to be evaluated. This study explores a multi-fidelity neural network that combines the computational efficient, low-resolution LES, for a large number of wind directions, with the more expensive, high-resolution LES, for a subset of wind directions. The goal is to predict the the mean, root-mean-square, peak-min and peak-max pressure coefficient at wind directions where high-fidelity LES data is not available, at a reduced computational cost. The training set includes 7 wind directions and the test set the 3 intermediate wind directions. The results show that the multi-fidelity neural network provide accurate predictions for all the quantities of interest (QoIs) except for the root-mean-square (rms) pressure coefficients (C_p)

1 Introduction

Computational Fluid Dynamics (CFD) can play a major role in estimating wind loading over high-rise buildings [1]. However, to enable its routine use for building design a significant computational speed-up is required. To do so, we aim at combining data from cheap, Low Fidelity (LF) Large-Eddy Simulations (LES) with data from expensive, High Fidelity (HF) LES in a Multi-Fidelity (MF) framework to predict wind loading on the facade of a high-rise buildings for the full wind rose. In particular, we will combine LF data computed at many wind directions with HF data computed at few wind directions to provide wind loading predictions for the full wind rose with accuracy close to the HF model at a fraction of its cost (i.e. 30% to 70% cost reduction). We will consider the case of the high-rise building in Figure 2a with an incoming atmospheric boundary layer (ABL). The QoIs will be the mean, rms, peak-min and peak-max pressure coefficient on the building facade. Figure 1 sums up the proposed framework.

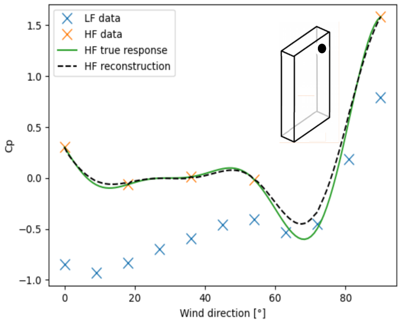


Figure 1: Goal is to reconstruct the green curve at each point on the building surface with features from the orange and blue datapoints.

Code is available upon request at: https://github.com/mattiafc/HRBMachineLearning.git ,

2 Related work

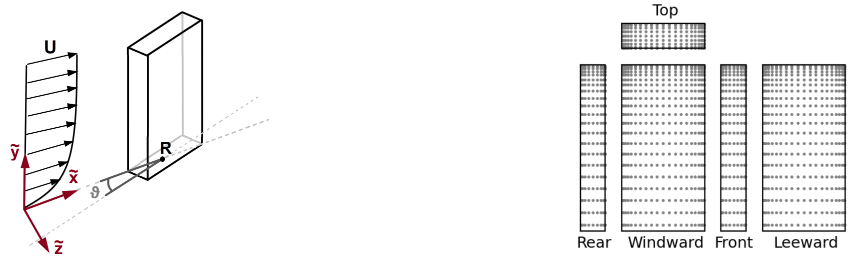
The novelty of this work is that it is the first time ever that mean, peak-min and peak-max Cp are predicted through Neural Networks (NN) in a MF framework. Almost no literature on MF modeling for wind loading predictions is available, as there is only one published work on the matter [2] and two currently under review [3], [4]. In [2], the author uses HF data at 5 wind directions and LF data at 10 wind directions to predict the HF model response at the 5 wind directions at which HF data is not available. The LF model adopted is Reynolds Averaged Navier-Stokes simulations while the HF model adopted is LES. The author uses NN to correct the LF model response for therms C_p.

3 Dataset and Features

The raw dataset consists in pressure and velocity time series for 80s at 1449 building building points from numerical simulations (see Figure 2b). The dataset features data at 10 different wind directions θ (see Figure 2a) and 2 different LES resolutions, name the Coarsest (CC), Coarse (C) simulations. Thus, there are a total of $1449 \cdot 10 = 14490$ datapoints per LES resolution. We will refer to the Coarsest LES as the LF model and the Coarse LES as the HF model. To train the model, LES data is post-processed to extract Galilean invariant features to help and boost the NN accuracy and facilitates its training [5]. If Galilean invariant features are not used and the model is evaluated on identical flows but different frame of reference, the NN would yield different predictions. Thus, the use of Galilean invariant features allow the NN to naturally embed symmetry properties of the problem of interest (e.g. $\theta = 0°$ and $\theta = 180°$ NN predictions should yield the same results). The features selected for this study are the following:

$$||\nabla\tilde{P}|| = \frac{H||\nabla\overline{P}||}{\frac{1}{2}\rho U_H^2}; \quad \tilde{k} = \frac{\overline{k}}{U_H^2}; \quad U_{in} = \frac{u_*}{U_H\kappa}\log(\frac{y+y_0}{y_0}); \quad \tilde{\tau}_w = \frac{||\nabla\tau_w||}{\frac{1}{2}\rho U_H^2}.;$$
$$\tilde{A}_{tap} = \frac{A_{tap}}{A}; \quad \theta\tilde{n} = \frac{\vec{U_{in}}\cdot\hat{n}}{||\vec{U_{in}}\cdot\hat{n}||}; \quad C_p^{mean} = \frac{\overline{P}}{\frac{1}{2}\rho U_H^2}; \quad C_p^{rms} = \frac{P^{rms}}{\frac{1}{2}\rho U_H^2};$$
$$C_p^{peak,Max} = \frac{P^{peak,Max}}{\frac{1}{2}\rho U_H^2}; \quad C_p^{peak,min} = \frac{P^{peak,min}}{\frac{1}{2}\rho U_H^2}; \quad (1)$$

Here, $\tilde{(.)}$ and $\overline{(.)}$ denotes adimensional and time averaged quantities, respectively. P, ρ, k denotes the pressure, air density, and turbulent kinetic energy on the building surface, and the ABL velocity at roof height, while U_H, u_*, y_0 ,and y represent the ABL velocity at roof height, the friction speed, the roughness length, and the vertical coordinate. This implies that U_{in} is a function of height, y ($U_{in}(y)$). For what concerns the other quantities, τ_w is the wall stress on the building surface, A_{tap} is the tributary area of each tap and $\hat{n}$ its normal, A is the building surface, and θ the wind direction. Finally, C_p denotes the pressure coefficient and the subscript peak,Max and peak,min denotes the maximum and minimum peak coefficients, respectively. These peak values are computed by using the method in [6].



(a) High-rise building schematic.

(b) Taps distribution on the building surface.

4 Methods

Our aim is to train a NN that is able to correct the LF model predictions at the wind directions for which we don't have HF data, and provide at those wind directions wind loading predictions with HF accuracy. Thus, we want our NN predictions to be more accurate than the LF model predictions. The QoIs for our learning task are the mean, rms, peak-max, and peak-min C_p of the HF model. This work uses data from LF simulations at 10 wind directions and HF simulations at 7 wind directions to predict HF responses at the other 3 wind direction (HF data withheld from training). Two different neural network architectures have been built from scratch and explored: (1) A single neural network for each QoI, and (2) a neural network with shared layers that branches out into four branches (one for each QoI). The shared layers improve the accuracy of the neural network and thus, the results of this architecture are presented in this report. We further explored two possible approaches by assuming that the QoIs are correlated: (1) End-to-end learning task where we predict directly the HF response directly, and (2) predict the discrepancy, ΔC_p, between LF and HF (intermediate step), and then correct the predictions by adding it to the LF data ($C_p^{HF} = C_p^{LF} + \Delta Cp$). The learning task of the first approach uses only LF data as features and HF data (of the same features) as labels. The second approach uses both LF and HF features and the labels are the discrepancies between LF and HF features. The latter approach outperforms the end-to-end learning task, and thus only results of approach (2) will be presented.We hypothesize that this is the case because approach (2) leverages more data. A graphical representation of the neural network is given in Fig. 3

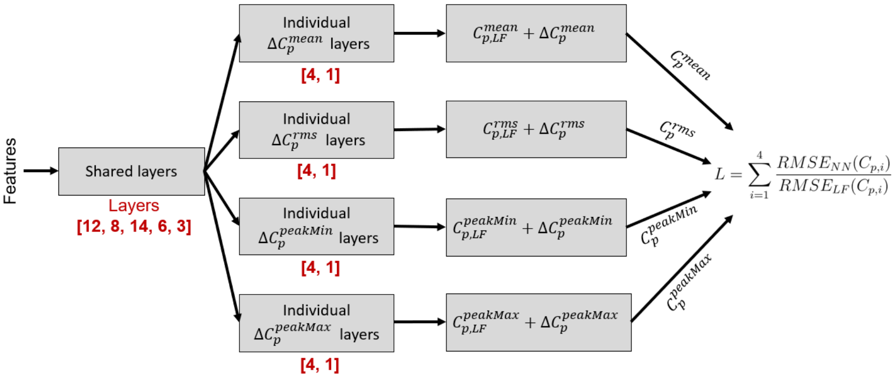

Features
Shared layers
Layers
[12, 8, 14, 6, 3]
Individual ΔC_p^{mean} layers
[4, 1]
Individual ΔC_p^{rms} layers
[4, 1]
Individual $\Delta C_p^{peakMin}$ layers
[4, 1]
Individual $\Delta C_p^{peakMax}$ layers
[4, 1]
$C_{p,LF}^{mean} + \Delta C_p^{mean}$
$C_{p,LF}^{rms} + \Delta C_p^{rms}$
$C_{p,LF}^{peakMin} + \Delta C_p^{peakMin}$
$C_{p,LF}^{peakMax} + \Delta C_p^{peakMax}$
C_p^{mean}
C_p^{rms}
$C_p^{peakMin}$
$C_p^{peakMax}$
$L = \sum_{i=1}^{4} \frac{RMSE_{NN}(C_{p,i})}{RMSE_{LF}(C_{p,i})}$

Figure 3: Implemented neural network architecture

The first step in setting up our neural network is to identify the 7 wind directions at which HF data is available (Θ) and the 3 wind directions at which we want to predict HF responses (Θ^C). As of now, we are using equispaced wind directions, meaning $\Theta = \{0°; 10°; 30°; 40°; 60°; 70°; 90°\}$ and $\Theta_C = \{20°; 50°; 80°\}$. Figure 4 shows of how data is used in our neural network. The train-dev split is 80/20, while the test set is Θ_C . To make predictions we use features from the LF model evaluated at Θ^C as input for our neural net, and then compare its predictions against the label from HF data evaluated at Θ^C.

Since we aim at predicting the wind load over the building, the root-mean-square error (RMSE) is the correct choice as both loss function and evaluation metric. However, since the pressure tap distribution on the building is uneven, their tributary area is different. Thus, taps with larger tributary area will contribute more to the RMSE than taps with a small tributary area. To account for this, we used a discretized version of the integral RMSE. In addition, since we used mini-batch gradient descent, and each batch contributes differently to the building RMSE, we normalized the RMSE by the LF error over the batch. This results in the following loss:

$$\frac{RMSE_{NN}}{RMSE_{LF}} = \sqrt{\frac{\sum_{i=1}^{m} A_{tap}^{(i)} (C_{p,i}^{NN} - C_{p,i}^{HF})^2}{\sum_{i=1}^{m} A_{tap}^{(i)} (C_{p,i}^{LF} - C_{p,i}^{HF})^2}}. \qquad (2)$$

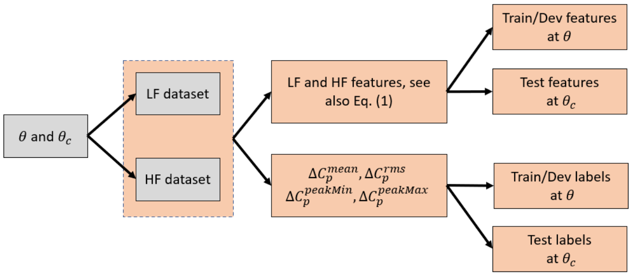

θ and θ_c
LF dataset
HF dataset
LF and HF features, see also Eq. (1)
$\Delta C_p^{mean}, \Delta C_p^{rms}$ $\Delta C_p^{peakMin}, \Delta C_p^{peakMax}$
Train/Dev features at θ
Test features at θ_c
Train/Dev labels at θ
Test labels at θ_c

Figure 4: Data split for the neural network.

Here NN denotes the neural network predictions (output), while HF denotes the HF model predictions (label), and i denotes one of the 4 C_p of interest. We performed random grid search considering learning rate $\in [10^{-2.5}; 10^{-5}]$, number shared hidden layers $\in [1, 4]$, number of neuron per shared layer $\in [2, 16]$, number of hidden individual layers $\in [1, 3]$, and batch size $\in [2^6, 2^8]$. To simplify the problem, we used the same architecture for all the individual hidden layers. We used *tanh* activation function for every neuron except for the output layer, where we use a single neuron with no activation.

5 Results & Discussion

The best neural network setup is a batch size of 32 examples, a learning rate of 0.00059, a neuron distribution for the hidden shared layers of [12, 8, 14, 6, 3] neurons and a hidden layer with 4 neurons with one output neuron with no activation for each hidden individual layers. The network was trained for 550 epochs. Table 1 shows the comparison between the RMSE of the NN train, NN test, NN train&test, and LF predictions error with respect to the HF response for every wind directions in Θ for the NN found through grid search with the smallest train&test RMSE. To avoid variance problems we used early stopping.

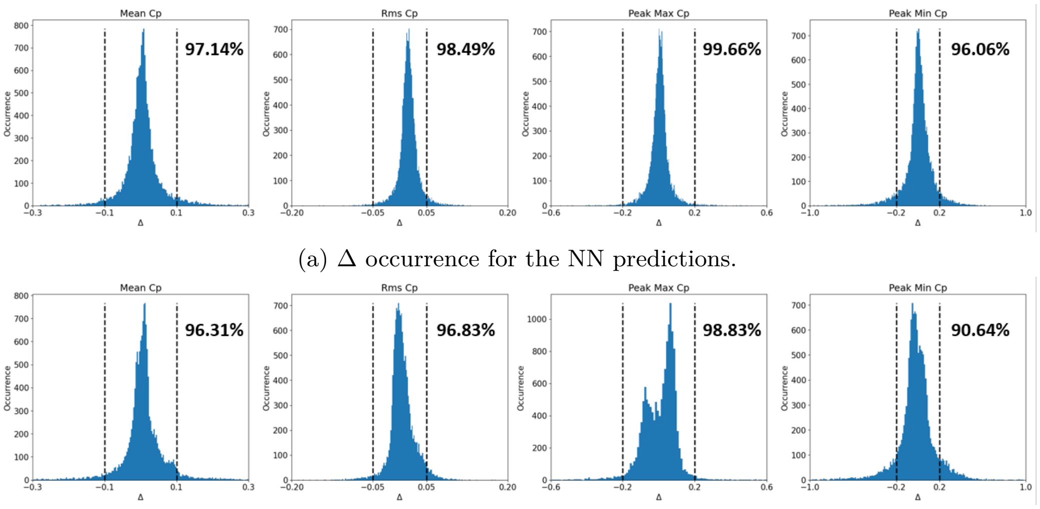
	C_p^{mean}	C_p^{rms}	$C_p^{peak,max}$	$C_p^{peak,min}$
RMSE NN Train set	0.0392	0.0214	0.0422	0.0949
RMSE LF Train set	0.0563	0.0194	0.0812	0.135
RMSE NN Dev set	0.0385	0.0212	0.0419	0.096
RMSE LF Dev set	0.0562	0.0189	0.0815	0.132
RMSE NN Test set	0.0260	0.0151	0.0283	0.065
RMSE LF test set	0.0376	0.0137	0.0571	0.093

Table 1: Root mean square error of the NN and the LF model with respect to HF predictions.

The results show that the NN is able to improve the LF model response for every QoI except for the rms Cp, which needs to be further investigated. To make a more meaningfull comparison of the LF and the NN model predictions we can define the metric

$$\Delta_i^m = \frac{|C_{p,i}^m - C_{p,i}^{HF}|}{sign(|C_{p,i}^m| - |C_{p,i}^{HF}|)} \tag{3}$$

where m indicates either the LF or the NN data and i the relative Cp. The model overpredicts HF data in terms of magnitude when Δ^m is positive while it undepredicts HF data when Δ^m is negative. Figure 5 shows the plot of the probability density function of Δ for both the LF and the NN predictions. Most of the Δ histograms show that the error of the NN is within reasonable margin of the HF predictions. These margins are ± 0.1 for the mean C_p, ± 0.05 for the rms C_p and ± 0.2 for the two peaks C_p. It is also possible to see that the NN outperforms the LF model predictions, as its distribution of Δ is much narrower than the one of the LF model. The only exception is the pdf of Δ for the rms Cp predictions is skewed toward positive values for the NN while centered on 0 for the LF model. To make the comparison between the LF and the NN predictions more quantitative, we can compute



(a) Δ occurrence for the NN predictions.

(b) Δ occurrence for the LF predictions

Figure 5: Occurence of Δ for the Neural network and the Low fidelity model predictions. The dashed black line represents the limit for which predictions are considered accurate. The percentages show the area of the pdf within the given limits. The percentage in the figures represent the fraction of the surface of the building over which the QoI is predicted with reasonable accuracy

the percentage of the building surface over which Cp predictions are within the reasonable interval defined before. The area is reported on the single Δ histograms.

6 Conclusions & Future Work

The present work evaluate the use of a Neural Network Multi-Fidelity framework to predict mean, rms, peak-max and peak-min C_p on the surface of a high-rise building for multiple wind directions. The Neural Network leverages the correlation between LF and HF model responses to correct the LF predictions and provide wind loading predictions with improved accuracy. The results clearly shows that the framework considered in this report is able to predict mean, peak max and peak min pressure coefficients on the high-rise building surface with accuracy close to that of HF data. In fact, the Neural Network provides satisfactory predictions over more than 96% of the building area for every QoI, while the LF model only do so for 90% of the building surface. The cost reduction of the wind loading evaluation provided by this framework is 30%, as HF data from only 7 out of 10 wind direction was used during training.

There are multiple paths we are considering going forward. First is to add a regularization term to the loss rather than using early stopping. Second, we want to explore meta-heuristic optimization algorithms such as Differential Evolution [7] to make the hyperparameters more methodical. We will also consider clustering points over the feature space and fit one neural network for each cluster to make it the task of predicting C_p^{HF} less end to end.

7 Contributions

Mattia Ciarlatani: Conceptualization, Methodology, LES run, Feature design, Coding, Post-processing.

Themistoklis Vargiemezis: Conceptualization, Methodology, Coding, Post-processing.

References

[1] Z. Huang, M. Ciarlatani, D. Philips, and C. Gorlé. Peak wind loading on high-rise building. *Journal of Wind Engineering and Industrial Aerodynamics. In preparation.*, 2021.

[2] Giacomo Lamberti and Catherine Gorlé. A multi-fidelity machine learning framework to predict wind loads on buildings. *Journal of Wind Engineering and Industrial Aerodynamics*, 214:104647, 2021.

[3] Themistoklis Vargiemezis and Catherine Gorlé. Evaluation of a multi-fidelity simulation framework for predicting wind pressure loads on buildings. 2021.

[4] Mattia Ciarlatani and Catherine Gorlé. Evaluation of a gaussian process-based multi fidelity framework for wind loading predictions. *To be published*, 2022.

[5] Julia Ling, Andrew Kurzawski, and Jeremy Templeton. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. *Journal of Fluid Mechanics*, 807:155–166, 2016.

[6] NJ Cook and JR Mayne. A refined working approach to the assessment of wind loads for equivalent static design. *Journal of Wind Engineering and Industrial Aerodynamics*, 6(1-2):125–137, 1980.

[7] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.