A multi-fidelity machine learning framework to predict wind loads on buildings

Giacomo Lamberti[a,*], Catherine Gorlé[a]

[a]*Stanford University, Y2E2 Building, 473 Via Ortega, Stanford, CA, 94305*

Abstract

Large-eddy simulations (LES) can provide accurate predictions of wind loads on buildings, but their high computational cost, and the need to explore all wind directions with a 10° resolution, limits their use in the design process. Reynolds-averaged Navier-Stokes (RANS) simulations have a low computational cost, but their accuracy can be compromised by the turbulence model and by the model required to retrieve the pressure fluctuations, that ultimately determine the design loads. This study proposes a multi-fidelity machine learning framework that combines computationally efficient RANS, for a large number of wind directions, with more expensive LES, for a small number of wind directions, to provide accurate predictions of the root mean square pressure coefficient at a reasonable computational cost. The training set includes 5 wind directions with a 20° resolution; the test set contains the 5 intermediate wind directions. A bootstrap algorithm, used to generate an ensemble of models, provides confidence intervals that encompass the majority of the LES data for the test directions. These results demonstrate that multi-fidelity machine learning frameworks provide a route to balancing accuracy and computational cost in the prediction of complex turbulent flow quantities.

Keywords: wind loading, machine learning, computational fluid dynamics

*Corresponding author

Email addresses: giacomol@stanford.edu (Giacomo Lamberti), gorle@stanford.edu (Catherine Gorlé)

1. Introduction

Computational fluid dynamics (CFD) represents an attractive tool for cladding design of high-rise buildings, given the potential to provide the complete 3-dimensional flow field in complex geometries. Nevertheless, routine use of CFD for design purposes still requires significant progress to guarantee the right balance between accuracy of the results and computational efficiency. In this respect, the most important modeling choice is whether to use Reynolds-averaged Navier-Stokes simulations (RANS) or large-eddy simulations (LES).

RANS, which solve the time-averaged Navier-Stokes equations and require modeling the entire spectrum of turbulence, are widely employed in wind engineering applications due to their low computational cost. However, their inability to accurately predict the inherently transient features of bluff body flows, such as separation and reattachment, is limiting their application to qualitative analysis or to less critical tasks like modeling wind comfort, pollutant dispersion and natural ventilation [1–5]. RANS-based calculation of wind loads is generally considered to be insufficiently accurate. It is challenging to obtain a quantitatively accurate prediction of the mean pressure field, and there is a need for additional models to retrieve an estimate of the turbulent pressure peaks that define the design loads. Different modeling approaches have been proposed, either solving the Poisson equation for the pressure fluctuations to generate artificial pressure fluctuations based on time-averaged RANS inputs [6–8], or employing empirical relationships that provide the root mean square (rms) pressure as a function of the local mean pressure, turbulent kinetic energy and velocity [9–12]. However, neither of these approaches produce accurate results in regions of flow separation and reattachment [6, 13].

LES, which apply a low-pass filter to the Navier-Stokes equations to resolve the larger energy-containing scales of turbulence, provide a significant improvement in the accuracy of the results, and a direct estimate of the pressure fluctuations. However, the computational cost can be prohibitive for wind engineering design applications, which often have to consider the loads that occur under all wind directions. Usually a wind direction resolution of 10° is required for tall buildings' cladding design [14], resulting in a total of 36 simulations.

The objective of this paper is to propose a multi-fidelity data-driven approach that combines a large number of computationally efficient RANS with a smaller number of LES to provide accurate wind load predictions at a reasonable computational cost. The approach uses machine learning to find a functional form that relates LES data of rms pressure coefficient to the mean flow variables predicted by RANS. Our full dataset consists of RANS and LES simulations of a rectangular plan high-rise building, at 10 wind directions; a subset of these wind directions is used for training, while the remainder of the simulations are used for testing the model performance.

This application of machine learning has several similarities to recent studies exploring the use of machine learning to relate RANS time-averaged quantities to high-fidelity LES or direct numerical simulation (DNS) data [15, 16]. The current analysis leverages knowledge gained in these studies, such as using Galilean invariant features [17], and using dimension reduction to analyze model behavior [18]. We also aim to explore two open questions identified in [15], namely 'what data should be used?', and 'what is the confidence in the prediction?'.

To explore the effect of the data used to train the models, we compare the performance of models trained on different subsets of training data. The first approach uses 5 wind directions to train a universal model that is then used to predict the remaining wind directions. The second approach selects 2 out of these 5 wind directions to train a model targeted at predicting a specific test wind direction. The selection of the training data is based on principal 51 component analysis (PCA) and the Kullback-Leibler divergence between the distributions of the features in the training and test data sets. To determine the effectiveness of this technique in identifying the optimal training sets, we compare the results to models trained on different combinations of the training wind directions. To quantify the confidence in the predictions, we use a bootstrapping technique to generate 1000 different models, and we report the mean and 95% confidence interval computed from the ensemble of models. The performance of this approach is assessed by comparing the results to the LES data.

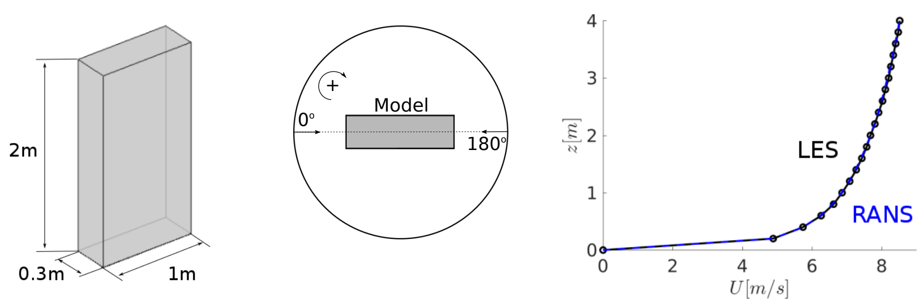
In the remainder of this paper, we first introduce the test case and the CFD set-up in Section 2. Section 3 presents the machine learning methods. Subsequently, the results for different wind directions are compared to the available LES data, in Section 4. Section 5 presents the conclusions and possible areas of future research.

2. CFD models

In this section, we first introduce the test case; subsequently, the setup of the RANS and LES simulations performed to obtain the features and output of the dataset are presented.

2.1. Test case

The numerical simulations reproduce a wind tunnel test on a high-rise building [19, 20]; the model is a 1m wide, 0.3m deep, and 2m high rectangular box, representative of a 100m tall building in full-scale. Figure 1 shows a sketch of the building model and the wind direction convention used in this study. The



(a) High-rise building model (b) Wind direction convention (c) Mean inflow velocity

Figure 1: Test case and mean inflow velocity magnitude.

experiments considered an open-terrain exposure, with a roughness length of 3.2mm, and a friction velocity of 0.49m/s. The corresponding reference velocity at building height is $\sim$ 7.8m/s. The resulting logarithmic velocity profile, shown in Figure 1c, was used to define the inflow boundary condition for the mean velocity in the RANS and LES.

2.2. RANS model description

2.2.1. RANS model set-up

The RANS simulations use a computational domain that is 35m in the streamwise and spanwise directions and 4m high. The mesh contains 3.7 million hexahedral cells and includes refinement regions close to the building model, which result in $\sim 42,000$ cells on the building surface. The resolution on the building surface is 25 mm, 12.5 mm and 6.25 mm in the streamwise, spanwise and vertical directions, respectively; the resulting y^+ is 500 on average. A grid dependency study was performed against a finer mesh composed of 6.2 million cells. The coefficient of determination R^2 between the mean pressure coefficient on the building's surface, computed with the two meshes, was 0.98, confirming the quality of the selected mesh size and resolution. To perform RANS simulations at different wind directions, we simply modify the velocity components imposed at the inflow, while maintaining the same computational domain and mesh.

We employ the RNG $k-\epsilon$ model together with a standard log-law wall function on the building's surface. To properly represent the neutral atmospheric boundary layer (ABL), the inlet boundary conditions specify a logarithmic velocity profile (see Figure 1c) and a turbulence kinetic energy that is constant with height and equal to $0.8\text{m}^2/\text{s}^2$ (see Figure 2) [12]. To ensure horizontal homogeneity of these profiles throughout the domain, we use a modified wall function that represents the effect of the roughness length of the terrain on the ground wall [12, 21]. The outlet is treated as a pressure-outlet with a constant relative pressure equal to zero and a zero-gradient boundary condition for the other flow variables. The side boundaries are either an inlet or an outlet, depending on the wind direction, while on the top boundary we impose a slip boundary condition.

The momentum and turbulence model equations are discretized using second order numerical schemes and iteratively solved using a linear solver with a symmetric Gauss-Seidel smoother. The Poisson equation is approximated with second order schemes and solved using the generalised geometric-algebraic multi-grid (GAMG) solver with Gauss-Seidel smoother. To monitor convergence, we select 40 points on the building surface and run the simulations until the corresponding mean pressure coefficients stopped varying; most simulations converged in ~ 2000 iterations.

2.2.2. Model for rms pressure coefficient

The solution of the RANS equations provides a prediction of the mean pressure distribution around the building, and an additional model is needed to retrieve the pressure fluctuations. Empirical models that relate the rms pressure coefficient C_p' to the local mean pressure coefficient C_P and turbulence kinetic energy k have been formulated using wind tunnel data acquired for low-rise buildings [9–13]. Since these geometries are not representative of high-rise building designs, and the measurements were limited to a small number of locations on the building, these models tend to be inaccurate when considering the lateral facades of high-rise buildings [13]. To assess the performance of our data-driven approach, we will use the empirical model that was found to achieve the highest accuracy in our test case as a reference. This model, proposed by Paterson and Holmes [11], calculates the rms pressure coefficient as follows:

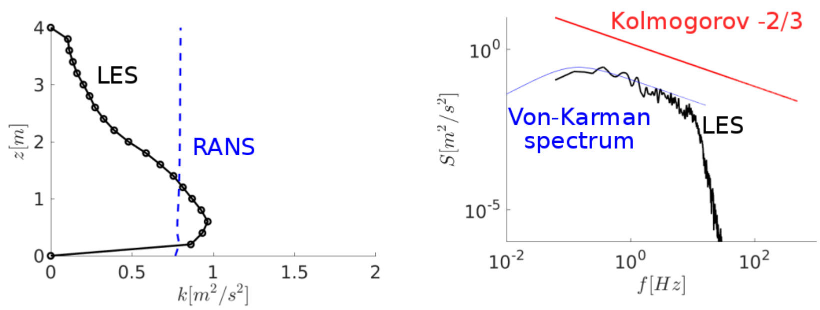
$$C_p' = \frac{k/3 + 0.816|C_P|U_0\sqrt{k_0}}{0.5U_H^2}, \tag{1}$$

where k_0 and U_0 are the turbulence kinetic energy and mean velocity magnitude of the incoming ABL, while U_H is the reference wind velocity at roof height.

2.3. LES

The computational domain for the LES is 35m in the streamwise direction and 20m in the spanwise one; the height of the domain is 4m. The high-rise building is located at a distance of 5m from the inlet boundary and 30m from the outlet [22]. Since the LES turbulent inflow generator requires the inlet to be perpendicular to the incoming ABL flow turbulence, a new mesh with a different building orientation is generated for each wind direction. The blockage ratio, i.e. the ratio between the area of the windward face of the building and the domain cross section, is always less than 2.8% [23]. The meshes include refinement regions close to the building model, resulting in a total number of $\sim 120,000$ cells on the building surface. The resolution is ~ 12.5mm, ~ 6.4mm and ~ 3.2mm in the streamwise, spanwise and vertical directions, respectively; the resulting y^+ is 170 on average. As part of a grid dependency analysis, the rms pressure coefficient computed from the present mesh was compared to the one computed from a finer mesh, with 4 times higher spatial resolution next to the corners and edges of the building; the comparison resulted in $R^2 = 0.93$.

Subgrid-scale turbulence is modeled using the dynamic k-equation model, while we employ a divergence-free digital filter [24, 25] to generate a turbulent flow field at the inflow. The divergence free digital filter method is coupled with an optimization algorithm, that adjusts the inflow parameters until desired statistics are obtained at the building location [26]. The resulting mean velocity profile matches the logarithmic law, shown in Figure 1c. Figure 2 shows the profile of turbulence kinetic energy at the building location, compared to the RANS value, together with the power spectrum of the streamwise velocity component at roof height. The power spectrum is in good agreement with a Von-Karman spectrum [27] and the inertial subrange follows the Kolmogorov hypothesis [28] up to a frequency of ~ 15Hz, after which the energy in the LES drops. On the side boundaries, we impose periodic boundary conditions, while



(a) Turbulence kinetic energy

(b) Power spectrum of the streamwise velocity component at roof height

Figure 2: Characteristics of the incoming boundary layer.

ground, top and outlet boundaries are treated as in the RANS simulations.

The momentum and turbulence model equations are discretized using second order numerical schemes and iteratively solved using a linear solver with a symmetric Gauss-Seidel smoother. The Poisson equation is approximated with second order schemes and solved using the generalised geometric-algebraic multi-grid (GAMG) solver with Gauss-Seidel smoother. To monitor convergence, we run the simulations until the maximum absolute difference in the rms pressure coefficient after a flow through is less than 0.01. Validation of the LES model against the wind tunnel data has been reported in [29].

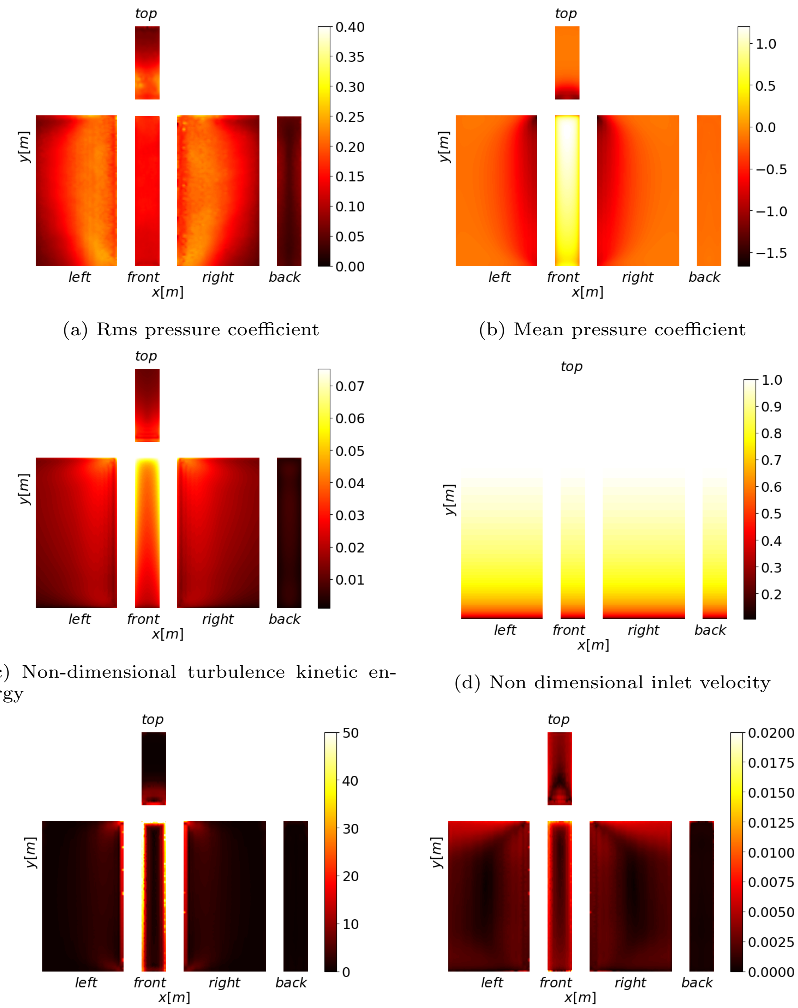
3. Machine learning method

In the following, we first present the selection of the features, i.e. the combinations of RANS mean flow variables used in the machine learning model. Subsequently, we introduce how we use PCA to visualize and quantify similarity between the distribution of the features across the different wind directions, and identify the data sets that contain the most relevant information to train models for specific wind directions. Lastly, the details of the hyperparameters and model search are presented.

3.1. Selection of features

The goal of the machine learning algorithm is to find the functional form that better relates the LES data of C_p' to 5 non-dimensional and Galilean invariant features constructed from the RANS flow variables, summarized in Table 1 [17].

Similar to standard empirical models, we select the mean pressure coefficient C_P, local turbulence kinetic energy k, and inflow velocity magnitude relative to the building U_0; in addition, we include the non-dimensional norm of the pressure gradient ∇P and the friction coefficient C_f to provide further information on the various flow regimes, i.e. separation, reattachment and fully attached flow. Figure 3 shows contour plots of the output and the features on the building's facade for the 0° wind direction; the building is unfolded to visualize the entire surface. The rms pressure coefficient (Figure 3a) is higher



(a) Rms pressure coefficient

(b) Mean pressure coefficient

(c) Non-dimensional turbulence kinetic energy

(d) Non dimensional inlet velocity

(e) Non-dimensional pressure gradient norm

(f) Friction coefficient

Figure 3: Contours of the output and features (before pre-processing) at 0° wind direction.

Mean pressure coefficient (C_P)	$\frac{P}{0.5\rho U_H^2}$
Non-dimensional turbulence kinetic energy	$\frac{k}{U_H^2}$
Non dimensional inflow velocity magnitude	$\frac{U_0}{U_H}$
Non-dimensional pressure gradient norm	$\frac{H\|\|\nabla P\|\|}{0.5\rho U_H^2}$
Friction coefficient (C_f)	$\frac{\|\|\tau_w\|\|}{0.5\rho U_H^2}$

Table 1: Features of the machine learning model, where τ_w is the wall shear stress vector and H the height of the building; ν and ρ are the kinematic viscosity and density of the air, respectively.

inside the region of flow separation; this indicates that including features that are able to distinguish regions of flow separation from regions of attached flow is essential for the accuracy of the method. Figure 3 shows that, excluding the inflow velocity profile, all remaining features provide some information on the different flow regimes. Before training the models, the features are centered and normalized by subtracting their mean value and dividing by their standard deviation across the dataset. This pre-processing step allows to obtain features with similar order of magnitude, which can significantly improve the learning process.

3.2. Selection of training data

3.2.1. Full data set and training data selection methods

The full dataset consists of RANS and LES simulations at 10 different wind directions, namely from $0°$ to $90°$, with a $10°$ resolution. These wind directions are representative of the entire wind rose, given the symmetry of the building model. For each wind direction, we have $\sim 42,000$ data points, i.e. the number of RANS cells on the building surface, consisting of 1 output and 5 features. Since our goal is to reduce the number of LES simulations necessary for cladding design, as sketched in Figure 4, we first split the dataset in two. The simulations at $10°$, $30°$, $50°$, $70°$ and $90°$ wind directions are included in the training data set, while the remaining 5 wind directions are in the test set. The machine learning model predictions for the rms pressure coefficients for the wind directions in the test set, i.e. 0°, 20°, 40°, 60° and 80°, will be compared to the corresponding LES data for validation.

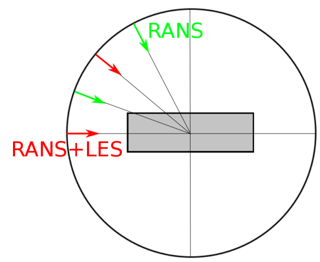


Figure 4: Sketch of the data-driven approach.

Using this initial split between training and test data, we will explore the performance of three different approaches for the selection of training data, resulting in three different models:

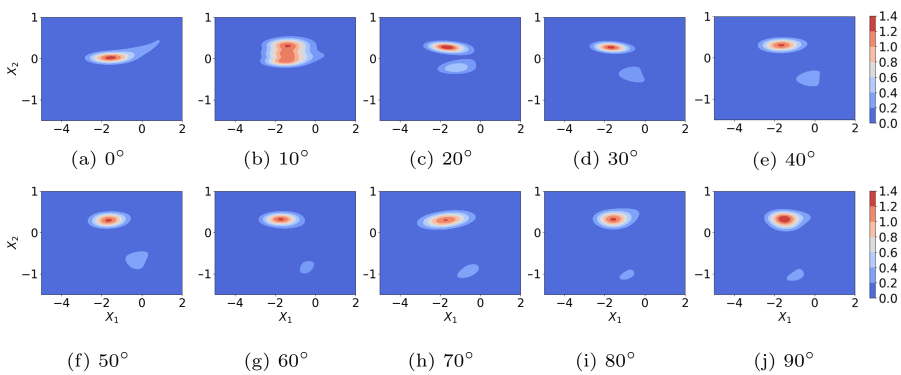
1. Universal model: uses all training data to learn a universal model that can be used to predict all wind directions in the test set
2. Lowest D_{KL} model: uses PCA and the Kullback-Leibler divergence (D_{KL}) to quantify similarity between the distributions of the features across the wind directions; then, a 2 wind directions are selected to learn a model for each wind direction in the test set.
3. Best model: perform further experimentation by considering different combinations of training wind directions, and report the results for the model that performs best for each wind direction in the test set. This approach is used to provide a reference for evaluating the performance of the universal and lowest D_{KL} models.

In the following we further discuss the approach used for selecting the data to train the lowest D_{KL} model.

3.2.2. Principal component analysis and Kullback-Leibler divergence

To better understand how the distribution of the features changes across the wind directions, we employ PCA [30]. The objective is to identify the wind directions in the training set that represent good training candidates for predicting a specific wind direction in the test set. Using PCA, the 5-dimensional feature space can be reduced to a 2-dimensional space, since most of the variance across the full dataset (i.e. including all wind directions) is explained by the first 2 principal components. Specifically, the first principal component X_1 and the second principal component X_2 explain $\sim 97.5\%$ and $\sim 2.2\%$ of the variance, respectively.

Figure 5 shows the joint distribution of the first two principal components for all available wind directions. Even though most data points are located in a similar region of the reduced space, a variation with the wind direction can be noticed. With increasing inflow angle, the high concentration region (in red) seems to move to higher values of X_2, while a second region of relatively high probability, that arises at 10°, moves to more negative values of X_2.



(a) 0° (b) 10° (c) 20° (d) 30° (e) 40°

(f) 50° (g) 60° (h) 70° (i) 80° (j) 90°

Figure 5: Contours of the joint distribution of the first two principal components.

In order to quantify the difference between the joint distributions of Figure 5, we compute the Kullback-Leibler divergence D_{KL} [31]:

$$D_{KL}(p||q) = \sum_i p_i \log \frac{p_i}{q_i} \tag{2}$$

where p_i and q_i represent the ith data points of the probability distributions p and q. Table 2 reports the values of D_{KL} between the wind directions in the test set (rows of Table 2) and the wind directions in the training set (columns of Table 2). For each task, the values in bold indicate the two wind directions that produce the smallest values of divergence and thus represent good candidates for training the corresponding model. Within the dataset, the 0° case deviates most from the others, having an average D_{KL} of ~ 1000 (Table 2).

$D_{KL}(p\|\|q)$	$q = 10°$	$q = 30°$	$q = 50°$	$q = 70°$	$q = 90°$	Average
$p = 0°$	**290**	1500	1100	1100	**990**	1000
$p = 20°$	**350**	**210**	350	1170	1000	770
$p = 40°$	520	**120**	**30**	370	440	300
$p = 60°$	340	390	**130**	170	**120**	290
$p = 80°$	400	790	280	**140**	**60**	420

Table 2: Kullback-Leibler divergence between wind directions in the test and training sets.

3.3. Hyperparameters and model search

The model search and hyperparameters tuning was performed only once, by using the data at 30° wind direction for training and a left-out simulation at 45° for validation. The learnable parameters are computed by minimizing the squared-error loss with L2 regularization, on the training data:

$$L(\mathbf{w}) = \sum_i (y_i - f(x_i; \mathbf{w}))^2 + \lambda ||\mathbf{w}||_2^2 \quad (3)$$

where y_i and x_i are the output and features of the ith training example, $\mathbf{w}$ the vector of weights and λ the regularization strength.

A variety of models is considered, starting from less flexible ones, such as linear regression, to more advanced ones, such as random forests and neural networks. To quantify the confidence in the resulting machine learning prediction, we use the bootstrap method [32], to compute the mean and 95% confidence interval of C_p'. Specifically, we generate $1,000$ training sets, by sampling the original training set with replacement; then, the corresponding machine learning models are trained and the ensemble of results are employed to compute a mean and confidence interval of C_p'. Table 3 reports the root mean square error (RMSE) in the training and validation sets for some of the models that were considered. The RMSE is computed using the average of C_p' across the bootstrap samples. The model that achieves the best performance in the validation set is a 5-layer neural network with 10 hidden units per layer and ReLU

	Train	Valid
Dataset	30°	45°
Linear regression (RMSE)	0.0192	0.0257
Linear regression + quadratic features (RMSE)	0.0131	0.0304
Random forest (RMSE)	0.0009	0.0270
5-layer Neural Network (RMSE)	0.0143	0.0221

Table 3: Root mean square error (RMSE) of the selected models.

activation function (Figure 6). To train the model, we used the Adam optimization algorithm [33], with a learning rate of 0.001 and regularization strength of 0.01. Within the models considered, the neural network is the one that benefits most from the bootstrap technique: the technique prevents overfitting, which is critical for such a flexible model.

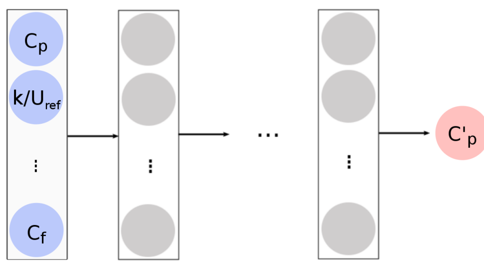
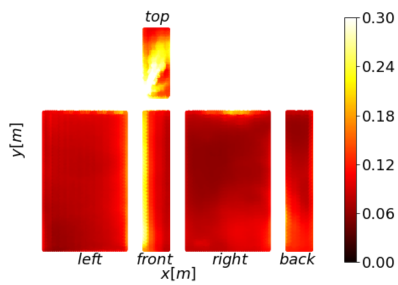
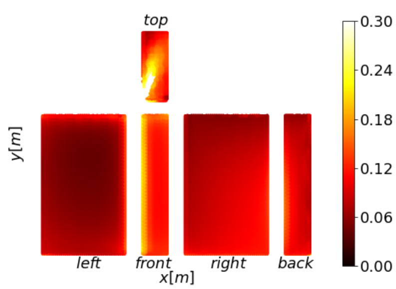


Figure 6: Neural network acrhirecture.

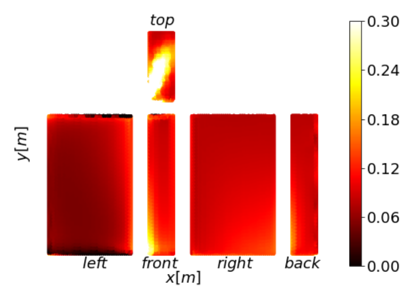
The better performance of the neural network model is confirmed in Figure 7, where the distribution of rms pressure coefficient around the building's surface is shown. The LES result (Figure 7a) is compared to the prediction given by the models listed in Table 3. The linear regression model with extended feature space and the random forests model manifest some degree of overfitting, which is also reflected in their low RMSE in the training set and high RMSE in the validation set (Table 3). Both the simple linear regression and neural network models are able to provide accurate prediction of C_p' around the building; nonetheless, the neural network slightly outperforms the linear regression model on the top and side walls, where separation and reattachment occur. Therefore, the remainder of the paper will present results obtained with the neural network. The model will be re-trained for each of the tasks in the test set, but



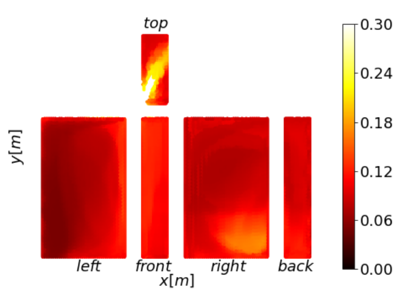
(a) LES



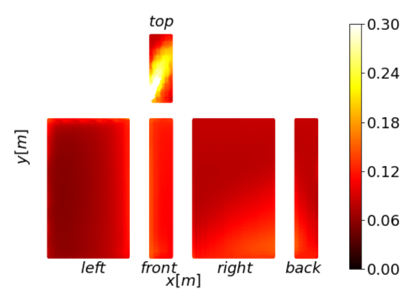
(b) Linear regression



(c) Linear regression + quadratic features



(d) Random forests



(e) Neural network

Figure 7: Contours of rms pressure coefficient at 45° wind direction.

keeping the architecture and hyperparameters fixed.

4. Results

In this section we first quantify the overall performance of the neural networks trained on different subsets of the training data in terms of RMSE. Subsequently, we present a more comprehensive comparison between the rms pressure coefficient computed by the different neural networks, the LES result and the empirical model prediction for the 0°, 40°, and 80° wind directions.

4.1. Comparison of RMSE

0°	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0897
Neural network (universal)	10°, 30°, 50°, 70°, 90°	0.0252	0.0520
Neural network (lowest D_{KL} = best)	10°, 90°	0.0163	0.0492
20°	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0570
Neural network (universal)	10°, 30°, 50°, 70°, 90°	0.0252	0.0260
Neural network (lowest D_{KL} = best)	10°, 30°	0.0197	0.0252
40°	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0438
Neural network (universal)	10°, 30°, 50°, 70°, 90°	0.0252	0.0334
Neural network (lowest D_{KL} = best)	30°, 50°	0.0149	0.0159
60°	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0546
Neural network (universal)	10°, 30°, 50°, 70°, 90°	0.0252	0.0300
Neural network (lowest D_{KL})	50°, 90°	0.0147	0.0259
Neural network (best)	50°	0.0097	0.0241
80°	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0475
Neural network (universal)	10°, 30°, 50°, 70°, 90°	0.0252	0.0210
Neural network (lowest D_{KL})	70°, 90°	0.0167	0.0273
Neural network (best)	50°, 70°, 90°	0.0241	0.0195
Average	**Training data**	**Train RMSE**	**Test RMSE**
Paterson-Holmes	-	-	0.0585
Neural network (universal)	-	0.0252	0.0325
Neural network (lowest D_{KL})	-	0.0166	0.0288
Neural network (best)	-	0.0166	0.0269

Table 4: Root mean square error (RMSE) of the neural network and Paterson-Holmes models, for each of the 5 wind directions in the test set.

Table 4 reports the training and test sets, and the RMSE for each of the models, i.e. the Paterson-Holmes empirical model, and the universal, lowest D_{KL}, and best neural network models. The universal model, which is trained using all 5 wind directions in the training set, achieves an average RMSE of ~ 0.0325 on the test set, which represents a $\sim 44\%$ improvement over the Paterson-Holmes model. By performing PCA and computing the Kullback-Leibler divergence to identify the 2 training wind directions that are most similar to a single test wind direction, the average RMSE can be further decreased by $\sim 12\%$. Finally, by considering additional combinations of wind directions, we are able to further improve the performance by $\sim 7\%$.

The test case that exhibits the worst neural network performance is the $0°$ wind direction; as explained in Section 3.2, this wind direction is characterized by a distribution of data that is significantly different from the remaining wind directions; the results will be further analyzed in Subsection 4.2. The lowest RMSE is achieved for the $40°$ wind direction; these results will be presented in more detail in Subsection 4.3.

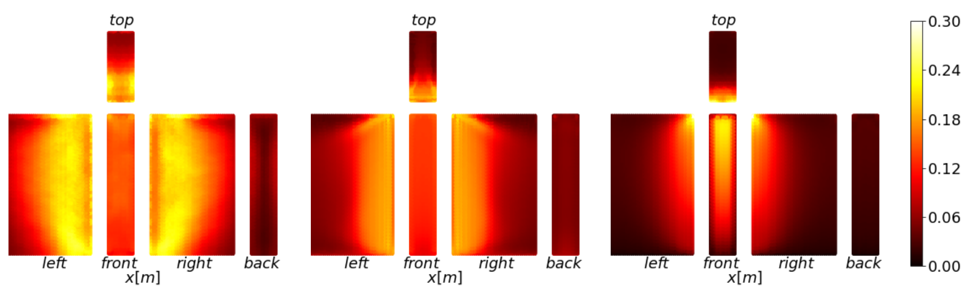
The potential benefit of carefully selecting training data based on the feature space is demonstrated by the fact that in 4 out of 5 cases, the lowest D_{KL} model produces a lower RMSE than the universal model. In addition, for 3 out of 5 cases, i.e. the $0°$, $20°$, and $40°$ wind directions, the lowest D_{KL} model corresponds to the best model found. For the $60°$ wind direction the RMSE decreases by 7% when reducing the training data set to only include the $50°$ wind direction. For the $80°$ wind direction, a more significant 28% decrease in RMSE is observed when adding the $50°$ wind direction. This is also the only wind direction at which the universal model outperforms the lowest D_{KL} model.

4.2. Results for $0°$ *wind direction*

The $0°$ wind direction represents a particular case in which the mean flow is symmetric and the side walls are sufficiently long for the flow to recover from separation. The best training set includes the two wind directions associated with the smallest value of D_{KL}: $10°$ and $90°$. The $10°$ wind direction represents the closest inflow condition, while the $90°$ wind direction is also characterized by a symmetric mean flow.

Figure 8 shows the contours of C_p' on the unfolded building surface at $0°$

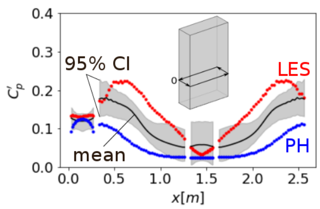
wind direction; the figure shows the LES data (Figure 8a), the mean obtained from the ensemble of neural networks (Figure 8b) and the empirical model result (Figure 8c). Compared to the empirical model, the neural network provides a better representation of the trend of C_p' on the entire building. This is confirmed by the RMSE of 0.0492 on the test set, compared to 0.0897 of the Paterson-Holmes model. The Paterson-Holmes model inherits the limitations of the RANS solution for the mean flow, which predicts a stronger suction at the windward edge of the lateral facades and a faster recovery than the LES; this too small separation region is similarly reflected in the features, indicated by the dark region of Figure 3b, and the model output, indicated by the yellow region of Figure 8c. On the other hand, the machine learning model partially compensates for the incorrect mean flow prediction of the RANS, as evident from the larger yellow region of Figure 8b. Despite this improvement, comparison to the LES result indicates that the neural network still under-predicts the size of the region with high C_p', and the corresponding magnitude of C_p'.



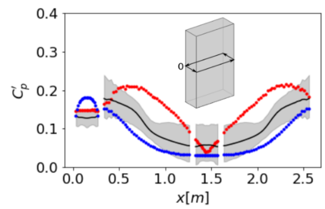
(a) LES (b) Neural network (c) Paterson-Holmes

Figure 8: Contours of rms pressure coefficient at 0° wind direction.

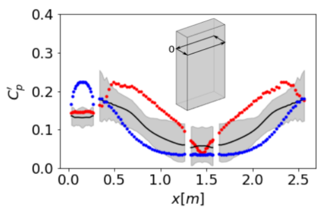
Figure 9 presents a more quantitative comparison along different building perimeters. The mean and the 95% confidence interval obtained from the bootstrap procedure are compared to the LES data and the empirical model. The plots confirm the good performance of the neural network on the *front*, *top*, and *back* faces; on these faces the confidence interval obtained from the bootstrap procedure encompasses the LES in most locations. Along the *left* and



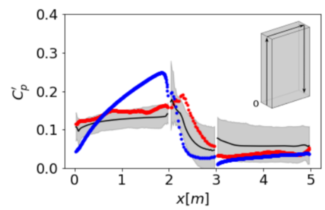
(a) Horizontal perimeter at 0.5m height



(b) Horizontal perimeter at 1.0m height



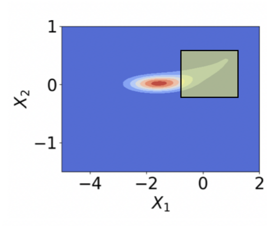
(c) Horizontal perimeter at 1.5m height



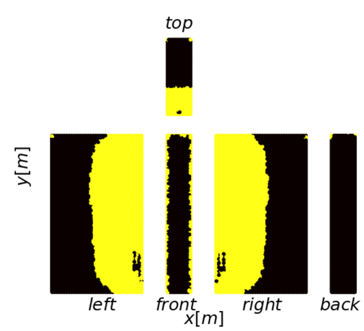
(d) Vertical perimeter

Figure 9: Profiles of rms pressure coefficient at 0° wind direction.

right facades the rms pressure coefficient is underpredicted, and the bootstrap confidence interval fails to fully represent the uncertainty in the results. Overall, the bootstrap confidence interval contains 51% of the data across the entire dataset; the maximum discrepancy between the confidence interval and the LES data is ~ 0.1, compared to ~ 0.2 of the empirical model. The size of the interval confirms the importance of uncertainty in the input data, especially where separation and reattachment occur.



(a) Joint distribution of the first two principal components at 0°; the box highlights the region of the features space that is unique to the 0° wind direction



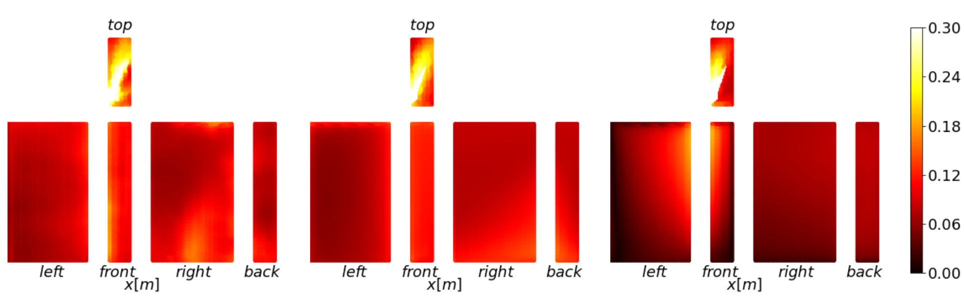
(b) The region in yellow highlights the location of the points that are inside the box of Figure 10a

Figure 10: Principal components at 0°.

The model inaccuracies on the *left* and *right* facades are likely due to the significantly different distribution of the features that characterize these facades for the 0° wind direction. To visualize this effect, Figure 10a shows the joint probability distribution of the first two principal components; as highlighted by the box in the figure, the 0° wind direction is characterized by a peculiar distribution of the principal components, that does not appear at any of the remaining wind directions in the dataset. The points within the box correspond to the regions of flow separation and reattachment, as shown in yellow in Figure 10b. The 0° wind direction is the only configuration at which the flow is able to reattach along the side walls after separating at the windward edge; as a result, when trying to predict the C_p' in these locations, the machine learning model is extrapolating. This also explains why the bootstrap procedure, which samples from the training data, cannot reflect the model uncertainty.

4.3. Results for 40° *wind direction*

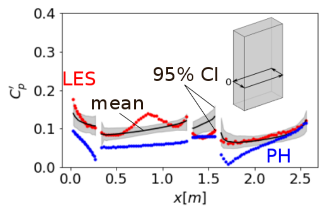
To predict the 40° wind direction, the best training set includes the two wind directions associated with the smallest value of D_{KL}: 30° and 50°. Figure 11b shows the contour plot comparing the LES data to the neural network and the Paterson-Holmes model results. There is significant qualitative improvement in the neural network predictions compared to the empirical model; the resulting RMSE is 0.0159, compared to 0.0438 achieved by the Paterson-Holmes model.



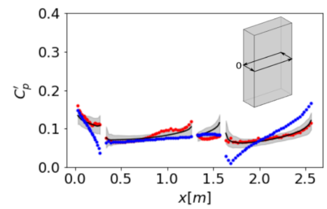
(a) LES (b) Neural network (c) Paterson-Holmes

Figure 11: Contours of rms pressure coefficient at 40° wind direction.

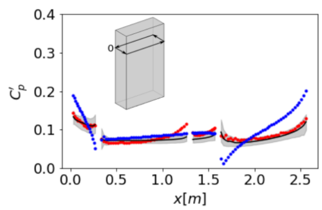
Figure 12 further confirms that good quantitative agreement is obtained: the



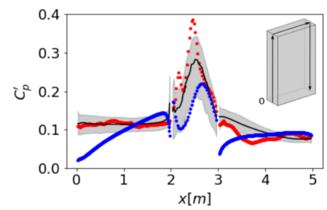
(a) Horizontal perimeter at 0.5m height



(b) Horizontal perimeter at 1.0m height



(c) Horizontal perimeter at 1.5m height



(d) Vertical perimeter

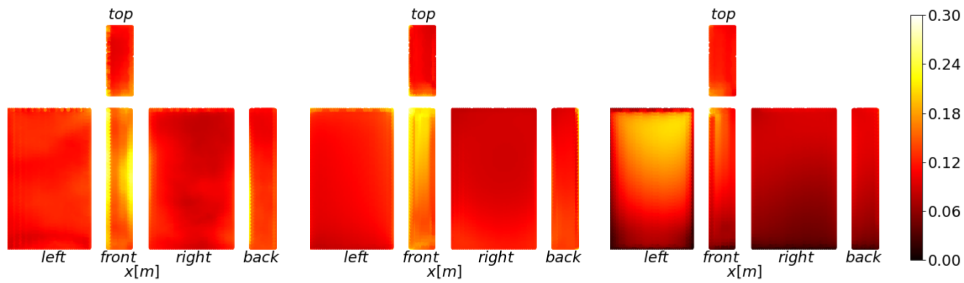
Figure 12: Profiles of rms pressure coefficient at 40° wind direction.

95% confidence interval obtained using the bootstrap method contains 84% of the LES data; the maximum discrepancy between the confidence interval bound and the LES is ~ 0.08 across the entire dataset. In contrast, the maximum absolute error experienced by the empirical model is ~ 0.28.

4.4. Results for 80° wind direction

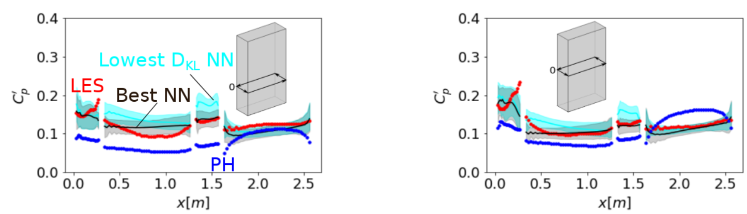
For the 80° wind direction, the best results are obtained by training the model using the data at 50°, 70° and 90°, i.e. the three wind directions that produce the lowest values of D_{KL}. Figure 13b shows the resulting contours of C_p' on the building surface, comparing the LES data to the neural network and Paterson-Holmes model results. The neural network, which has a RMSE is 0.0195 compared to the LES, provides a much better representation of C_p' than the Paterson-Holmes model, which has a RMSE of 0.0475.

The profiles of Figure 14, which compare the mean and 95% confidence interval predicted by the neural network to the LES data and the empirical model result, quantitatively confirm the improvement obtained by the neural network. Across the entire dataset, the bootstrap confidence interval contains 77% of the data; the maximum discrepancy between the confidence interval

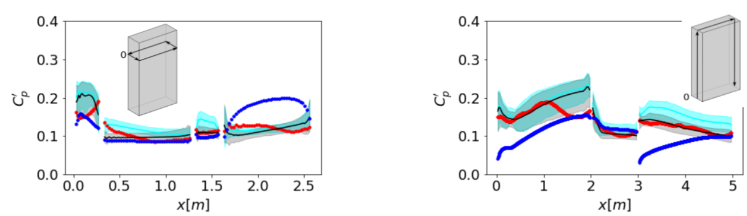


(a) LES (b) Neural network (c) Paterson-Holmes

Figure 13: Contours of rms pressure coefficient at 80° wind direction.



(a) Horizontal perimeter at 0.5m height (b) Horizontal perimeter at 1.0m height

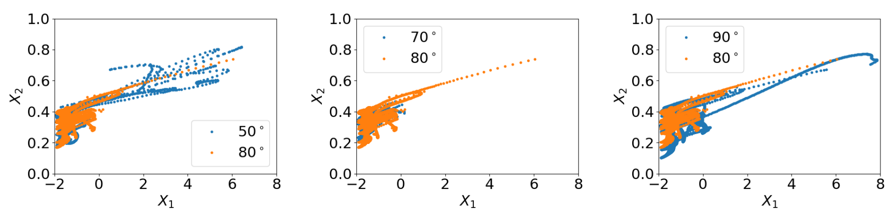


(c) Horizontal perimeter at 1.5m height (d) Vertical perimeter

Figure 14: Profiles of rms pressure coefficient at 80° wind direction.

and LES data is ~ 0.06, against a maximum absolute error of ~ 0.13 for the Paterson-Holmes model.

Figure 14 also shows the profiles of C_p' predicted by the neural network trained on the two wind directions with lowest D_{KL}, i.e. 70° and 90°. The model achieves a RMSE of 0.0273, i.e. $\sim 28\%$ higher than the best result. Across the entire dataset, the 95% confidence interval predicted by this model contains only 52% of the data, while the maximum discrepancy remains ~ 0.06. The lowest D_{KL} result seems biased towards higher C_p' values on the right and back faces, where the flow is fully separated for this wind direction.



(a) 50° and 80° (b) 70° and 80° (c) 90° and 80°

Figure 15: Scatter plot of the first two principal components relative to the *right* and *back* faces.

To understand why adding the 50° wind direction to the training set can remove this bias and improve the performance of the model, we can reconsider the distribution of the first two principal components of the features. Figures 15a, 15b and 15c compare the scatter plot of the first two principal components for the 80° configuration, to the 50°, 70° and 90° wind directions, respectively. The plots are restricted to the *right* and *back* faces, i.e. the locations where the best neural network outperforms the neural network trained on the 2 wind directions with lowest D_{KL}. The plots indicate that the 50° wind direction adds a significant number of data points for $X_1 > 0$ and $X_2 > 0.5$, thereby avoiding extrapolation when predicting the 80° case. Given the relatively low number of points in this region, this was not directly evident from Figure 5 and did not have a significant impact on the Kullback-Leibler divergence (Table 2).

5. Conclusions and future work

A multi-fidelity machine learning approach has been proposed to combine a large number of computationally efficient RANS with a smaller number of LES, to predict the rms pressure coefficient C_p' on a high-rise building. The model is trained to relate the C_p' obtained from LES to 5 non-dimensional and Galilean invariant features calculated from RANS.

The full data set consists of RANS and LES simulations at a 10° wind direction resolution. The data for the 10°, 30°, 50°, 70° and 90° wind directions are used to train the model, while the LES for the 0°, 20°, 40°, 60° and 80° wind directions are only used to evaluate the model performance. Based on model search and hyperparameters tuning, performed by employing a left-out simulation at 45°, a 5-layer neural network with 10 hidden units per layer and ReLU activation function is found to achieve the lowest root mean square error (RMSE) on the test set. A bootstrap technique, that samples with replacement the training data, is used to produce an ensemble of 1000 models and support reporting a mean and confidence interval for the model predictions. Subsequently, the model is re-trained to predict each of the 5 wind directions in the test set.

When training a universal model on all 5 wind directions in the training set and using it to predict all 5 wind directions in the test set, the RMSE is on average 2 times smaller than the RMSE of a standard empirical model. When training a targeted model on a select subset of 2 wind directions in the training set, to predict a specific wind direction in the test set, this RMSE is further reduced by 20%. In this case, the 2 wind directions used for training are selected by considering the similarity between the joint distributions of the first two principal components of the features, for the wind directions in the training and test sets; these first two principal components explain 99.7% of the variance in the dataset, and the similarity between their distributions can be quantified using the Kullback-Leibler divergence. Comparison to models trained on various combinations of wind directions in the training set, indicates that this strategy for selecting the training data results in optimal performance for 3 of the 5 test wind directions. For the remaining wind directions, the best model still relies on training data from wind directions with a low Kullback-Leibler divergence, but it uses data from either only 1 or 3 wind directions.

The test case that experiences the worst agreement is the 0° wind direction. This wind direction has the highest average Kullback-Leibler divergence, and it is shown that the model breaks down in regions where it is extrapolating in the space defined by the first and two principal components. In these regions, the lack of data also implies that the bootstrap method can not provide accurate information on the uncertainty in the model. At the remaining wind directions, the 95% confidence interval predicted by the bootstrap procedure encompasses between 70% and 84% of the data, and the maximum absolute error remains limited to 0.08.

In summary, the proposed multi-fidelity framework has the potential to significantly reduce the number of LES simulations needed for design, while retaining a significantly higher accuracy than standard empirical models. The findings from this study also have broader relevance to machine learning for turbulence modeling. First, the use of principal component analysis to select training data with optimal similarity to test data is shown to improve model performance; it could also be used to identify a need for additional training data, or to identify regions where the model should not be trusted. Second, the use of a bootstrap procedure to create an ensemble of machine learning models provides useful confidence intervals, as long as the model is not extrapolating beyond the training data in the space of the first two principal components. Future work will focus on further customization of the selection of optimal training data, and on applying the procedure to the peak pressure coefficient as the quantity of interest.

6. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number 1635137, and used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by National

Science Foundation grant number CI-1548562.

[1] Bert Blocken, Ted Stathopoulos, P Saathoff, and X Wang. Numerical evaluation of pollutant dispersion in the built environment: comparisons between models and experiments. *Journal of Wind Engineering and Industrial Aerodynamics*, 96(10-11):1817–1831, 2008.

[2] Theodore Stathopoulos. The numerical wind tunnel for industrial aerodynamics: Real or virtual in the new millennium? *Wind and Structures*, 5(2_3_4):193–208, 2002.

[3] Akashi Mochida and Isaac YF Lun. Prediction of wind environment and thermal comfort at pedestrian level in urban area. *Journal of wind engineering and industrial aerodynamics*, 96(10-11):1498–1527, 2008.

[4] Bert Blocken, Ted Stathopoulos, Jan Carmeliet, and Jan LM Hensen. Application of computational fluid dynamics in building performance simulation for the outdoor environment: an overview. *Journal of Building Performance Simulation*, 4(2):157–184, 2011.

[5] Bert Blocken. 50 years of computational wind engineering: past, present and future. *Journal of Wind Engineering and Industrial Aerodynamics*, 129:69–102, 2014.

[6] S Senthooran, Dong-Dae Lee, and S Parameswaran. A computational model to calculate the flow-induced pressure fluctuations on buildings. *Journal of wind engineering and industrial aerodynamics*, 92(13):1131–1145, 2004.

[7] Yu-Tai Lee, William K Blake, and Theodore M Farabee. Modeling of wall pressure fluctuations based on time mean flow field. *Journal of Fluids Engineering*, 127(2):233–240, 2005.

[8] MF Huang, IWH Lau, Chun Man Chan, Kenny CS Kwok, and Guojie Li. A hybrid rans and kinematic simulation of wind load effects on full-scale tall buildings. *Journal of wind engineering and industrial aerodynamics*, 99(11):1126–1138, 2011.

[9] D. A. Paterson, J. Holmes, et al. Computation of wind flow around the texas tech building. 1989.

[10] R Panneer Selvam. Computation of pressures on texas tech building. *Journal of Wind Engineering and Industrial Aerodynamics*, 43(1-3):1619–1627, 1992.

[11] DA Paterson. Predicting rms pressures from computed velocities and mean pressures. pages 431–437, 1993.

[12] PJ Richards and BS Wanigaratne. A comparison of computer and wind-tunnel models of turbulence around the silsoe structures building. *Journal of Wind Engineering and Industrial Aerodynamics*, 46:439–447, 1993.

[13] IM Kalkman, AJ Bronkhorst, CA van Bentum, and J Franke. A comparison of rans computations and wind tunnel tests for rms pressures on a high-rise building model. 2012.

[14] Peter Irwin, Roy Denoon, and David Scott. *Wind Tunnel Testing of High-Rise Buildings*. Routledge, 2013.

[15] Karthik Duraisamy, Gianluca Iaccarino, and Heng Xiao. Turbulence modeling in the age of data. *Annual Review of Fluid Mechanics*, 51:357–377, 2019.

[16] Steven L Brunton, Bernd R Noack, and Petros Koumoutsakos. Machine learning for fluid mechanics. *Annual Review of Fluid Mechanics*, 52:477–508, 2020.

[17] Julia Ling and J Templeton. Evaluation of machine learning algorithms for prediction of regions of high reynolds averaged navier stokes uncertainty. *Physics of Fluids*, 27(8):085103, 2015.

[18] Pedro M Milani, Julia Ling, and John K Eaton. Physical interpretation of machine learning models applied to film cooling flows. *Journal of Turbomachinery*, 141(1):011004, 2019.

[19] L. Amerio. Experimental high resolution analysis of the pressure peaks on a building scale model façades. 2018.

[20] G. Lamberti, L. Amerio, G. Pomaranzi, A. Zasso, and C. Gorlé. Comparison of high resolution pressure measurements on a high-rise building in a closed and open-section wind tunnel. *Journal of wind engineering and industrial aerodynamics*, (under review).

[21] A. Parente, C. Gorle, J. van Beeck, and C. Benocci. Improved kappa-epsilon model and wall function formulation for the RANS simulation of ABL flows. *Journal of Wind Engineering and Industrial Aerodynamics*, 99:267–278, 2011.

[22] Agerneh K Dagnew and Girma T Bitsuamlak. Computational evaluation of wind loads on a standard tall building using les. *Wind and Structures*, 18(5):567–598, 2014.

[23] Jörg Franke. *Best practice guideline for the CFD simulation of flows in the urban environment*. Meteorological Inst., 2007.

[24] Z. T. Xie and I. P. Castro. Efficient generation of inflow conditions for large eddy simulation of street-scale flows. *Flow, turbulence and combustion*, 81(3):449–470, 2008.

[25] Steven J Daniels, Ian P Castro, and Zheng-Tong Xie. Peak loading and surface pressure fluctuations of a tall model building. *Journal of Wind Engineering and Industrial Aerodynamics*, 120:19–28, 2013.

[26] G. Lamberti, C. García-Sánchez, J. Sousa, and C. Gorlé. Optimizing turbulent inflow conditions for large-eddy simulations of the atmospheric boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*, 177:32–44, 2018.

[27] R. Harris. On the spectrum and auto-correlation function of gustiness in high winds. *Electrical Research Association*, 1968.

[28] Stephen B Pope. *Turbulent flows*. IOP Publishing, 2001.

[29] G. Lamberti and C. Gorlé. Sensitivity of les predictions of wind loading on a high-rise building to the inflow boundary condition. *Journal of Wind Engineering and Industrial Aerodynamics*, (under review).

[30] Svante Wold, Kim Esbensen, and Paul Geladi. Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52, 1987.

[31] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86, 1951.

[32] Bradley Efron. Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics*, pages 569–593. Springer, 1992.

[33] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.