An Extension to the Grid-Induced Machine Learning CFD Framework for Turbulent Flows

Chin Yik, Lee[1,2*] Vân Anh, Huynh-Thu[3] Stewart Cant[2]

1. Rolls-Royce Plc, Derby, UK (email: chinyik.lee@rolls-royce.com)
2. Department of Engineering, University of Cambridge, Cambridge, UK
3. Department of Electrical Engineering and Computer Science, University of Liège, Belgium
*Corresponding author: Chin Yik, Lee

Abstract

High-fidelity computational fluid dynamics (CFD) is widely used to understand turbulence and guide engineering design. While effective in predicting complex flow phenomena, CFD simulations at high Reynolds numbers require fine grids, resulting in prohibitive computational costs for parametric studies. To address this, we proposed a framework that uses machine learning (ML) to predict fine-grid results from coarse-grid simulations in a previous work. Coarsening the grid increases grid-induced error and affects turbulence prediction, necessitating a data-driven surrogate model to predict and correct these errors. A Random Forest (RF) regression was used to construct the surrogate model. The proposed framework was tested using a turbulent flow configuration consisting of an enclosed duct with a triangular bluff body acting as a blockage to the incoming flow. The chosen input features (IFs) were shown to be critical in predicting the turbulent flow field. In the current paper, we introduce further enhancements to the framework to allow it to be more robust in its prediction and application. These extensions also serve to reduce the computational cost of the approach without compromising on the accuracy. The proposed extensions include (i) adoption of Multivariate Random Forest (MRF) to replace the RF approach; (ii) identification and reduction of the IFs required for training and prediction using Variable IMportance Prediction (VIMP); (iii) predictions of flow field with changes in the bluff body configurations. The present paper aims to investigate the capability of the proposed extensions within the framework. We show that (i) the MRF allows for the accurate prediction of multiple outputs within one training instance but with a reduced computational cost relative to the RF approach. (ii) the impact of the IFs on the training can be understood via VIMP, and applying the MRF model with reduced IFs selected through VIMP does not cause any detriment to the accuracy of the prediction (iii) the extended framework trained with different bluff body configurations could be robustly applied to predict the flow field in an unseen configuration that is different from those trained. The predictive capability of the approach with these proposed extensions is demonstrated.

Keywords

Computational Fluid Dynamics, Input Features, Multivariate Random Forest, Variable Importance Prediction, Turbulent Bluff Body Flow

1. Introduction

Turbulence is a complex phenomenon that contains flow structures spanning a broad range of temporal and spatial scales. Such a flow is high-dimensional, chaotic and highly intermittent. Resolving all relevant scales down to the viscous cutoff limit in turbulent flows is a challenging undertaking. Direct Numerical Simulation (DNS) [1, 2] offers a pathway to fully resolve turbulence without the need of any models, yet achieving this for high Reynolds number flows remains computationally prohibitive. As such, DNS is invaluable for fundamental research, but its practical application is confined to modest computational domains and simpler flow scenarios.

More recently, Large Eddy Simulation (LES) [3] has emerged as a promising alternative to simulate turbulent flows. In LES, large energy-containing scales of the fluid motion are resolved, while unresolved length scales smaller than the computational grid are modelled using sub-grid scale (SGS) models [4]. On the other hand, Reynolds-averaged Navier-Stokes (RANS) simulation is widely applied in many industries to predict turbulent flows. Despite being significantly cheaper, RANS rely heavily on turbulence closure modelling and can struggle to predict complex flows [5, 6]. LES offers an advantage over RANS to predict unsteady phenomena. LES also helps to mitigate the high computational cost of DNS and is gaining popularity in many industrial applications [7, 8, 9].

Turbulence closure requires the mathematical modelling of the Reynolds stresses for RANS or SGS effects in the filtered equations for LES [6]. Although turbulence models are generally validated using experimental measurements or high-fidelity simulations, formulations of these models are largely based on either heuristic arguments or physical intuition. As a result, the functional forms are usually not guaranteed. In addition, constants for turbulence models are often calibrated using simplified or canonical test cases. A consequence of this is that RANS may lack the desired accuracy to predict complex flows especially with high swirl, pressure gradient or streamline curvature [10, 11]. Fine tuning of the model constants is often necessary for RANS to work across different flow configurations. For LES, such an impact is expected to be small given a sufficiently fine grid [12, 13]. However, when an inappropriate resolution is applied in LES, the error can be propagated through the SGS model resulting in an erroneous prediction. To this end, machine learning (ML) presents an alternative strategy to construct an optimal mapping between available turbulence data and statistical quantities of interest.

ML has been actively exploited to develop turbulence closure and improve flow predictions, as exemplified by the pioneering work by Milano and Koumoutaskos [14], who compared a neural network (NN) against Proper Orthogonal Decomposition (POD) in the reconstruction of near wall turbulent fields. They demonstrated the favourable performance of nonlinear NN over linear POD models. More recently, ML has been applied extensively to identify model discrepancies between the RANS-based Reynolds stress tensor and high-fidelity simulations. Tracey et al. [15] used NN to enhance a deficient Spalart-Allmaras model by substituting a source term within the model with machine-learned functional forms to simulate the transonic flow over a wing. Zhang and Duraisamy [16] explored Gaussian regression and NN to reconstruct the spatial form of correction factors for turbulent and transitional flows. Ling and Templeton [17] compared Support Vector Machine, Adaboost decision tree, and Random forest (RF) to classify and predict regions of high uncertainty in the Reynolds stress tensor. In a subsequent work, Ling et al. [18] embedded

Galilean Invariance into their tensor prediction, providing a simple approach to incorporate known physical symmetries and invariances into the learning architecture. Ling et al. [19] then further developed a model for the Reynolds stress anisotropy tensor using NN with embedded Galilean invariance. Parish and Duraisamy [20] developed the field inversion and ML modelling framework that builds corrective models based on inverse modelling. Wang et al. [21] and Wu et al. [22] applied RF to build a supervised model for the discrepancy in the Reynolds stress tensor. Kaandorp and Dwight [23] pursued similar lines, but they modified the RF algorithm to accept a tensor basis to satisfy Galilean invariance.

Extensive efforts have been made also in the development of SGS closure using ML. One of the earliest applications can be found in Sarghini et al. [24], where a NN is trained and validated using the flow fields provided by the scale-similarity model [25] to model the turbulent viscosity coefficient. Beck et al. [26] used a NN based on local convolutional filters to predict the mapping between the flow in a coarse simulation and the SGS closure terms, using a filtered DNS of decaying homogeneous isotropic turbulence. Maulik et al. [27] employed a multilayer perceptron to predict the SGS model in an LES using high fidelity numerical data to train the model. They evaluated the performance of their method on a classical two-dimensional decaying-turbulence test case known as the Kraichnan turbulence [28]. Antoine et al. [29] considered improving the prediction of the SGS model of a scalar flux. They combined a NN and a bi-objective optimisation technique to minimise the modelling error for the SGS scalar flux and allow better predictability across different flow configurations. NNs have also been successfully applied in the context of LES to understand the flow physics [30] and develop SGS closure models [31] for wall-bounded turbulence. These studies demonstrate the increasing interest in applying ML to improve the accuracy of turbulence predictions.

In a previous paper, a CFD-ML framework was proposed [32] to allow for a better prediction of the turbulent flow field from low-resolution simulations. The proposed framework was driven by two key considerations: practical design application and better physical characterisation of turbulence. In industrial settings, complex designs with lengthy iteration cycles are common. Such practice necessitates numerous simulations to converge towards an optimal design. While LES can be applied, performing a large number of runs becomes impractical due to computational constraints and time limitations. The errors from the SGS model can become large and difficult to quantify when a ‘coarse grid’ resolution is used. The proposed approach aims to predict grid-induced errors to correct flow variables using data-driven models. From a physical perspective, a fundamental challenge with ML lies in its limited ability to generalise to unseen flow, particularly in the context of data-driven turbulence. This is exacerbated in circumstances where there is insufficient resolution of the flow and the turbulence quantities are not well predicted. Previous studies have highlighted the limitations of ML models when applied to flow significantly different from those used for training [17, 22]. To address this, it is crucial for the learning algorithm to generalise effectively, even in scenarios where there is a shift in the dataset distribution between training and testing data. This is achieved through the incorporation of input features (IFs) that encapsulate the relevant turbulent physics.

The applications of the proposed framework are targeted towards turbulent flows. The test case considered is a well-validated turbulent bluff-body flow configuration [33, 34]. To construct the surrogate model, the RF regression algorithm is used. From the perspective of predicting turbulence flows, the RF-based regression is well suited because of its ability to learn nonlinear decision boundaries, a prerequisite for turbulence modelling. The algorithm has also been widely used across a broad range of ML applications for complex flows, making it a good candidate for the evaluation of turbulence [17, 22, 23]. Key benefits of RF relative to NN primarily lie in its simplicity in usage, robustness, and efficient computational cost. Numerous studies were performed to evaluate the performance of RF against deep learning techniques such as NN. Firstly, RF is easier to train and has far fewer hyperparameters to tune compared to NN [35]. The baseline hyperparameters usually work well, requiring minimal tuning and simplifying optimisation [36]. In terms of robustness, RF is less sensitive to noise, making it more desirable in handling datasets with missing values or outliers [37-38]. When it comes to tabular data, RF was shown to outperform NN in terms of predictive performance [39]. In RF, less data is needed to achieve the same level of accuracy as NN. RF was found to outperform NN when the training sample size was small [40, 41]. Most notably, RF is computationally more efficient [38, 41], with training and testing times generally an order of magnitude faster than NN [42]. These advantages highlight RF's suitability and versatility for practical engineering applications, making it a desirable candidate for predicting complex turbulent flow in the context of the current framework.

Further details of the CFD-ML framework are given in a previous paper [32]. In the current paper, the authors aim to build upon the existing framework and propose further enhancements. The framework is extended on the following basis:

1. The RF ML algorithm proposed in the original framework is a univariate ML approach. It only allows one variable to be predicted for a single training instance. As a result, a separate RF training and prediction process is required for each variable that is to be predicted. Such a process is computationally inefficient if multiple variables need to be predicted. Furthermore, training the variables separately using RF ignores the correlation between each of the predicted variables. This potentially violates the physical laws of turbulent flows, for example, as the velocity components are correlated to one another. To address this, our emphasis is on extending the originally proposed RF to the multivariate random forest (MRF) approach to predict variables that are correlated with one another within a single training instance.
2. For the initial ML framework, a considerable amount of IFs is used when training the ML model. However, not all the features/variables may be relevant for predicting the outcome of interest. Variable selection techniques involve eliminating irrelevant variables and can provide several advantages. Firstly, the model with lesser IFs is computationally cheaper to run. Secondly, a model with a small number of IFs is more interpretable. Thirdly, the risk of overfitting is potentially reduced, hence improving model's ability to generalise. Here, we leverage the interpretability of the MRF to understand the impact of the IFs onto the quality of the prediction. We apply a feature selection technique to identify the features of high importance, with creating a more economical ML model in mind: (i) we train the model using reduced IFs and evaluate the impact on the model performance; (ii) we place emphasis on understanding why certain features play a more important role than others

in the current setup. Such an understanding can be translated and applied into other configurations of turbulent flows.

3. In the previous study, we tested the approach using a single triangular bluff body configuration, but with different boundary conditions and mesh resolution. In the current work, we extend the usability of the approach to different bluff-body shapes. These consist of the original triangular-shaped bluff body, a parabolic-shaped bluff body, and a square-shaped bluff body. The trained surrogate model is tested on a different bluff body from that used in the training process. The differences in the bluff-body shape mean that the separation behaviour, recirculation structure and flow features behind the bluff-body will vary. The key motivation of this exercise is to extend the generalisability of the ML approach towards different geometries, allowing it to be applied effectively to predict the flow field of a different configuration from the trained ones.

The present paper aims to investigate the capability of the proposed extensions within the data-driven CFD-ML framework. The paper is structured as follows: The numerical setup and target configurations are first described. Next, the ML methodology is discussed in detail. The MRF regression algorithm is then introduced followed by the explanation of the IFs. In this framework, 'fine grid' and 'coarse grid' LES solutions are performed at different scenarios containing changes in inlet velocities and geometrical configurations. Validation of the 'fine grid' solution is first presented and the differences between the 'coarse grid' and the 'fine grid' solutions are highlighted. Next, the prediction using the MRF approach is presented. The accuracy of the approach is examined and compared against the RF. Then, the Variable IMportance Prediction (VIMP) of the input features is assessed. The influence of the IFs on the flow field is examined and the crucial features for the ML training are identified. The results trained using reduced IFs are compared against the full set of IFs. In the last section, the MRF approach is investigated for an extended scenario where different geometries were used for training and prediction. The accuracy of MRF is assessed to understand the predictive capability. A summary of the results and future works are presented in the final section.

2. Numerical Methodology and Geometrical Configurations

The datasets that are used to train and evaluate the ML model in this case study are based on an incompressible LES run for a bluff-body configuration in an enclosed duct. The LES is performed using the open source CFD toolkit OpenFOAM [43]. The code utilises an unstructured collocated Finite Volume Method (FVM) in which the discretisation scheme is based on Gauss's Theorem and a semi-implicit time integration scheme. The SGS turbulence is solved using the Smagorinsky model [44]. Van-Driest damping is applied in the near-wall regions [45]. Spatial differencing is achieved via a central differencing scheme. The scheme is second-order accurate in the smooth regions and flux limited in areas with steep gradients to ensure boundedness. Temporal integration is achieved via a second-order backward method. The Pressure Implicit with Splitting of Operators (PISO) pressure-correction algorithm [46] is used to solve the systems of equations.

The bluff-body configuration is shown in Fig 1. It corresponds to an experimental configuration [33,34] consisting of a rectangular duct with dimensions 1.0 m (length) × 0.12 m (height) × 0.24 m (width). Inside the duct is an equilateral triangular bluff body with sides w_f = 0.04 m, positioned 0.318 m downstream from the inlet, with one apex pointing upstream. The bluff-body acts as a blockage to the incoming flow. The flow separates at the bluff-body edges, resulting in vortex shedding behind the bluff-body.

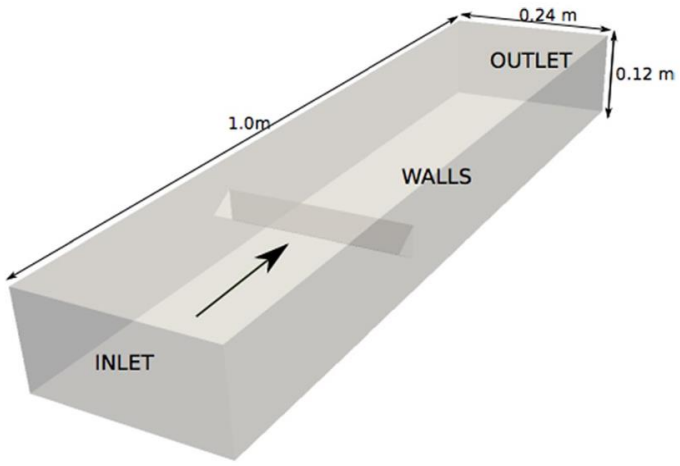


Figure 1 The configuration: enclosed duct with triangular bluff-body

In the current work, three key elements are considered when training and evaluating the CFD-ML framework. These include changes in (i) grid resolution; (ii) boundary conditions and (iii) bluff-body shape. These changes are incorporated when setting up the geometry, mesh and boundary conditions for the LES.

A 2D section through the 3D computational grid around the triangular bluff body is shown in Fig. 2. For the fine grid (Fig. 2, left), 30 cells are placed along each side of the triangular bluff body and 100 cells are placed across its width, resulting in a total cell count of 3.5 M. Two additional sets of coarse grids are constructed for the ML exercise. For the first coarse grid (Fig. 2, right), the cells along the edges are halved relative to the fine grid, resulting in 15 cells along each side of the bluff body and 50 cells across its width, giving a total cell count of 0.4 M. The second coarser grid uses 25 cells across the width instead of 50, yielding a total cell count of 0.18 M. The coarse grids are incapable of resolving the flow in LES, and are used as inputs for ML training.

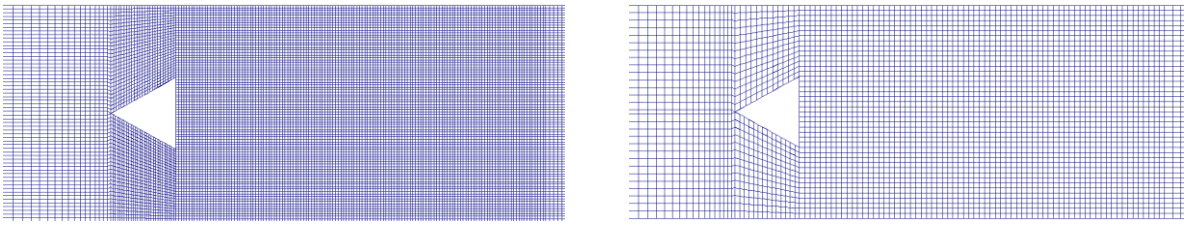


Figure 2 Zoomed-in section showing the mesh around the bluff-body for the fine mesh with 3.5 M cells (left) and coarse mesh with 0.4 M cells (right)

The axial velocity prescribed at the inlet of the duct is 17 m/s. This value is based on the data reported in the experiment [33, 34]. This yields an estimated Reynolds number of 45,000 based on the bluff body length scale. At the inlet, a turbulence intensity of 4% [47] is specified and the temperature is set at 288 K. A no-slip and adiabatic boundary condition is specified for the walls. The outlet pressure is fixed at an atmospheric condition of 1 bar with a zero gradient assumption for the velocity. For the purpose of training different cases for the ML, cases with different inlet conditions, summarised in Table 1 are considered. The Pope criterion [48] for the fine-grid has been computed for these conditions to evaluate the LES quality. For all conditions up to 34 m/s, more than 85% of the turbulent kinetic energy is resolved in the domain, indicating the adequacy of the fine mesh resolution for the current configuration.

Conditions	Inlet velocity (m/s)	Expected Re at the inlet
1	8.5	22000
2	13	33500
3	17	45000
4	34	90000

Table 1 Operating conditions considered for the bluff body turbulent flow

The shape of the bluff-body is pivotal in dictating the development of the boundary layer and flow separation, which ultimately affect pressure losses and flow field at the wake. In the original setup, the bluff-body assumes an equilateral triangular geometry. Here, we consider three new configurations where the bluff-body takes different shapes: a parabolic-shaped bluff body, a square-shaped bluff body and a bullet-shaped bluff body.

The bluff body geometries and their corresponding meshes are presented in Fig. 3. The number of cells on the sides of each bluff body is set to be the same as the equilateral triangular bluff body, which is 30 cells across the side length. The upstream and downstream mesh counts also are kept similar to the triangular bluff body to facilitate a consistent resolution of the flow field between cases. The parabolic bluff body (Fig. 3 left) has curved upper and bottom sections prior to the edges. The downstream facing side is equal in length to the side of the triangular bluff body. The square bluff body (Fig. 3 middle) consists of four equilateral sides. Each side has the same length as the side of the triangular bluff body. The ‘bullet’ bluff body (Fig. 3 right) features a downstream extension to the triangular bluff body. The downstream extension is a $0.185w_f$ extrusion to the original counterpart. Unlike the immediate separation typically observed at the edges of the triangular bluff-body, the elongated shape of the bullet-like bluff-body is expected to induce a delayed flow separation process. The modification in these bluff-bodies is anticipated to exert a notable influence on the flow field and recirculating structure behind the bluff body.

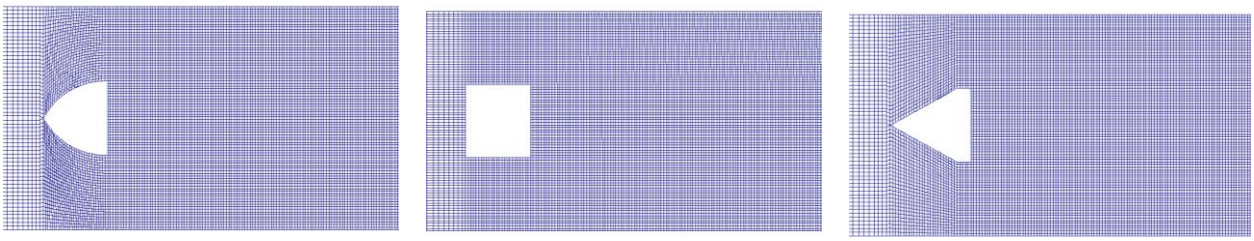


Figure 3 Zoomed-in sections showing the mesh around the parabolic (left), square (middle) and bullet (right) bluff bodies

3. ML Methodology

The fundamental concept underpinning this framework is depicted in Fig. 4. A set of LES results are used to train the ML model. The LES results could be based on cases at different operating conditions Ψ, grid spacing Δ or geometrical configuration Θ. The variables ϕ = [ϕ_1, ϕ_2, ϕ_3,...] represent a comprehensive set of features extracted from LES that encapsulate the underlying flow physics. The variables computed from the high-resolution, fine grid simulation are denoted as ϕ_f and the low-resolution, coarse grid simulation are denoted as ϕ_c. The primary motivation is to leverage the available data from both low- and high-resolution simulations to construct a data-driven surrogate model. This model aims to accurately predict the value of a new, yet unavailable variable $\phi_{\square}$, based solely on variables from corresponding low-resolution computations ϕ_c.

In the training process, both high- and low-resolution data for every cell across the entire computational domain is required to correct the coarse grid error. Ideally it would be preferable to use DNS to ensure the highest fidelity when computing ϕ_f^{tr} (where superscript tr indicates training flows). In the current work, a sufficiently resolved LES is used to obtain a solution for this. The justification for adopting LES here is based on the fact the LES results are well validated against experiment, and that performing DNS for industrial-based computation in complex geometry would not be feasible. With this in mind, the trained solution can only be as good as the fine grid solution. To obtain ϕ_c^{tr}, the same LES simulation is performed again but using a coarse grid. The difference between the fine grid solution and coarse grid solution ϕ_f^{tr}- ϕ_c^{tr} gives the coarse grid error.

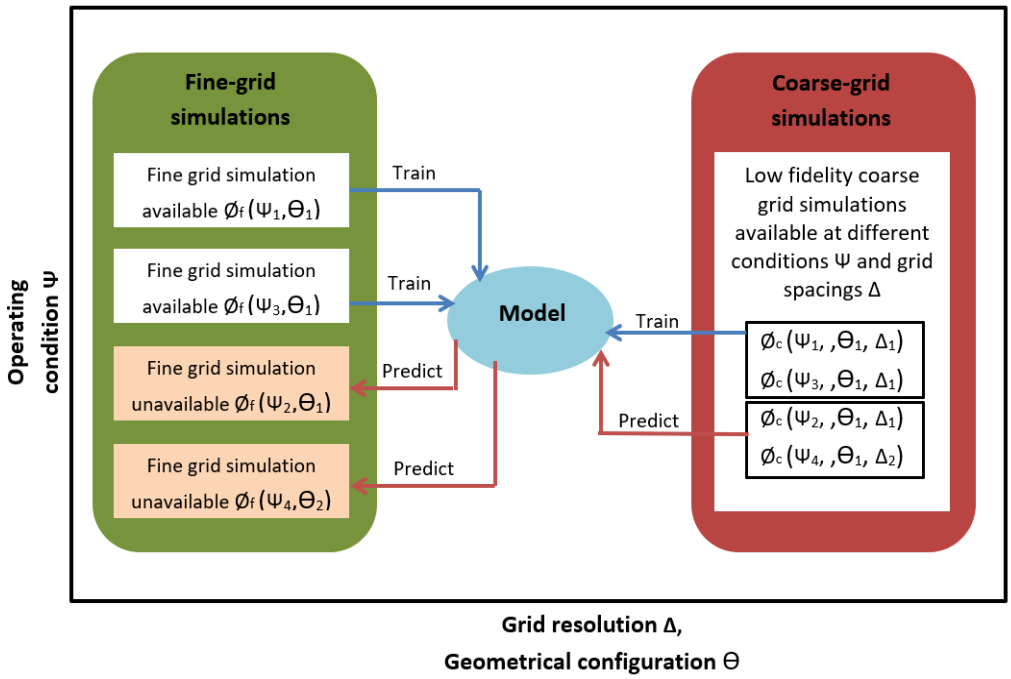


Figure 4 The workflow for data generation for the CFD-ML approach

An immediate difficulty in computing the difference between the coarse and fine grid simulations is the difference in mesh resolution between these simulations. The mesh for the fine-grid simulations is generally much finer and therefore has a much higher cell count than the coarse-grid. To overcome this, the fine-grid results are mapped onto the same computation grid as the coarse-grid simulation such that the cell counts are consistent. Mapping of the fine grid results onto the coarse grid produces $\phi_{f,m}^{tr}$. The error between the fine and coarse grid variables is $\varepsilon(\phi^{tr})$ = $\phi_{f,m}^{tr}$-ϕ_c^{tr}. Implications due to mapping of the results were discussed in the previous paper [32]. The error due to mapping has been shown to be far smaller than the coarse grid results.

Based on this information, the ML surrogate model can be constructed. The datasets used as input for training are taken purely from the coarse grid simulations ϕ_c^{tr}. These variables are formulated in the form of IFs for the ML algorithm to characterise the turbulent flow field, $IF(\phi_c^{tr})$. The datasets used as output for training are the errors between the coarse grid and fine grid simulations $\varepsilon(\phi^{tr})$. In the training process, the ML algorithm learns to predict $\varepsilon(\phi^{tr})$ through the IFs that are provided as inputs. The formulation of IFs is critical in enabling an accurate prediction of the turbulent flow field. Further details on the IFs are provided in section 5.

To predict the fine-grid solution ϕ_f^{ev}(where superscript ev denotes evaluation) at a new, unseen condition of interest, either at specific Ψ or Θ, the coarse grid results at that specific condition ϕ_c^{ev} are required. Using the IFs of the coarse grid datasets $IF(\phi_c^{ev})$, the coarse grid error of the prediction $\varepsilon(\phi^{ev})$ can be obtained using the ML surrogate model. The fine-grid solution for this new case is then computed as ϕ_{pred}^{ev}= ϕ_c^{ev}+ $\varepsilon(\phi^{ev})$.

The predictive performance of a data-driven surrogate model depends on the level of similarity between training and evaluation flows. Greater resemblance in the flow fields between the training and evaluation flows aids learning, while disparities present challenges. To assess the surrogate model, we evaluate the surrogate model across four main scenarios 1-4 of training and evaluation flows (see Table 2) to assess its robustness and generalisability. As part of the extension proposed in the study, an additional scenario 5 is considered. The results for scenario 5 are presented in Section 9.

Scenario	Training flows	Evaluation flows
1 Operating condition Interpolation	Triangular bluff body Fine grid: inlet velocity 8.5, 13, 34 m/s, mesh = 3.5 M Coarse grid: inlet velocity 8.5, 13, 34 m/s, mesh = 0.4 M	Triangular bluff body Inlet velocity = 17 m/s Mesh = 0.4 M
2 Operating condition extrapolation	Triangular bluff body Fine grid: inlet velocity 8.5, 13, 17 m/s, mesh = 3.5 M Coarse grid: inlet velocity 8.5, 13, 17 m/s, mesh = 0.4 M	Triangular bluff body Inlet velocity = 34 m/s Mesh = 0.4 M
3 Operating condition and grid size interpolation	Triangular bluff body Fine grid: inlet velocity 8.5, 13, 34 m/s, mesh = 3.5 M Coarse grid: inlet velocity 8.5, 34 m/s, mesh = 0.4 M; Inlet velocity 13 m/s, mesh = 0.18 M	Triangular bluff body Inlet velocity = 17 m/s Mesh = 0.4 M
4 Operating condition and grid size extrapolation	Triangular bluff body Fine grid: inlet velocity 8.5, 13, 17 m/s, mesh = 3.5 M Coarse grid: inlet velocity 8.5, 17 m/s, mesh = 0.4 M; Inlet velocity 13 m/s, mesh = 0.18 M	Triangular bluff body Inlet velocity = 34 m/s Mesh = 0.4 M
5 Bluff-body shape	Fine grid: triangular bluff body, square bluff body, parabolic bluff body Inlet velocity 17 m/s, mesh = 3.5 M Coarse grid: triangular bluff body, square bluff body, parabolic bluff body Inlet velocity 17 m/s, mesh = 0.4 M;	Bullet bluff body Inlet velocity = 17 m/s Mesh = 0.4 M

Table 2 Training and evaluation flows for scenarios 1-5

For each scenario, six simulations are conducted for the training flows: three fine grid simulations and three coarse grid simulations, aimed at evaluating the coarse-grid error. These simulations encompass various combinations of either operating conditions Ψ (inlet velocity), grid spacing Δ

or geometrical configuration Θ (bluff-body shape). In scenarios 1 and 2, we only consider changes in operating conditions for the training flows. In scenarios 3 and 4, we explore the combined changes in operating condition and grid resolution for the training flows. For scenarios 1-4, the geometry is maintained where the triangular bluff body is used. In scenario 5, we consider changes in the geometrical configurations. Different bluff body shapes are used in the training flows but with the inlet velocity kept at 17 m/s.

In scenario 1, the solutions from three cases, u_{inlet} = 8.5, 13 and 34 m/s at 3.5 M cells (fine grid) and 0.4 M cells (coarse grid) are considered for training. The trained surrogate model is then used to predict the coarse grid error for u_{inlet} = 17 m/s, given its coarse grid solution at 0.4 M cells. This scenario is considered the least challenging as the inlet velocity of the evaluation flow lies within the range of the training cases. Scenario 2 presents a slightly more challenging situation since the inlet velocity of the evaluation flow falls outside the range of the training data, necessitating extrapolation.

Next, the approach is tested in scenarios 3 and 4, which involve both interpolation and extrapolation of the inlet velocity and grid size. An inherent challenge with LES is the potential to result in spurious solutions when using grids that lack sufficient resolution to capture the desired length scales of the flow. Such an impact is addressed in both of these scenarios, where one of the training samples features a coarse grid count of 0.18 M. Such a resolution is inadequate for LES and allows for the ML approach to be tested. For both these scenarios, the solution at 0.18 M cells is mapped onto the mesh at 0.4 M cells to compute the coarse grid error.

In scenario 5, variations in the bluff body shape are considered. The surrogate model needs to learn the flow field associated with the differences in the bluff body shapes, namely triangular, square and parabolic bluff bodies in the training flows. The trained model is tested on a bullet-shaped bluff body. The key challenge for this scenario is to examine the capability of the ML approach to capture these aerodynamics and apply them to a configuration different from the trained datasets.

4. Multivariate Random Forest

In this section, we provide a brief overview of RF, and focus on detailing the extension of the univariate RF to multivariate response settings. The MRF approach is the ML algorithm adopted in the current work. The MRF algorithm [49] is implemented using the Comprehensive R Archive Network (CRAN) RF-SRC package [50]. The RF-SRC package is called using the R-to-Python interface into the open-source Python library Scikit-Learn [51].

The underlying idea of the RF is based on the regression trees developed by Breiman et al. [52]. The regression tree from the predictor, *x* to the response, *y* (see Fig. 5) consists of four key components: (1) a set of splits, comprising yes or no questions that serve to split the initial predictor space. The subsamples which are created by assigning learning cases according to the splits are termed as nodes. A node without any descendant nodes is called a leaf. (2) A node impurity measure to determine the splits during training. This measure is typically the variance of

the output response for a regression algorithm. (3) A split function *f(s, t)* can be evaluated for each split *s* for each node *t*. The best split aims to optimise *f(s, t)*, such that the homogeneity of response distributions among the resultant child nodes after a split is maximised. The homogeneity is evaluated based on the node impurity, as defined in (2); and lastly (4) An approach to determine the appropriate tree size.

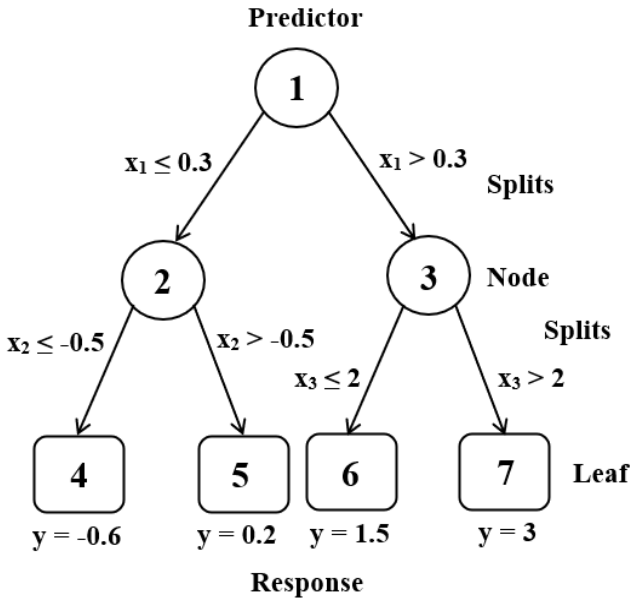


Figure 5 Example of a regression tree

For a univariate regression tree, the predictors and response are designed as x_{ij} and y_i , where *i = 1…n*, and *j = 1…p.* The variable *n* denotes the number of training samples and *p* denotes the number of predictors. When partitioning a node *t* into the child nodes, left node t_L and right node t_R during the split, the *L2* node impurity measure is taken as the sum–of-squares, given as

$$SS(t) = \sum_{i \in t} (y_i - \mu(t))^2$$

where $\mu(t)$ is the mean of y_i in node *t.*

The split function is given as

$$f(s,t) = SS(t) - SS(t_L) - SS(t_R)$$

For a multiple response approach, the response data is y_{ik}, where *i = 1…n, k = 1, ...q.* The extension of a single response to a multiple response regression tree involves the modification of the split function. A natural formulation is to replace the node impurity measure with a summation over the different output responses.

$$SS(t) = \sum_{k=1}^{q} \sum_{i \in t} (y_{ik} - \mu_k(t))^2$$

where $\mu_k(t)$ is the mean of the k^{th} output in node *t*. In the multivariate setting, this expression of sum-of-squares replaces the existing *SS(t)* for the univariate setting in the split function. Each leaf of a multi-response regression tree contains a length-*q* output prediction vector, which consists of the average output vector values taken over the training samples that reach that leaf.

The RF is a collection of regression tree predictors, defined as

$$h(x; \theta_m) \text{ where } m = 1 \ldots, M$$

where *M* is the number of trees, *x* represents the observed input/predictor of length *p* and the term θ_m represents the parameters of the m-th tree. The RF prediction is the unweighted average over the collection of trees.

$$\bar{h}(x) = (1/M) \sum_{m=1}^{M} h(x; \theta_m)$$

The key requirements to achieve an accurate RF regression are (i) low prediction error for the individual trees in the forest and (ii) low correlation between residuals of different trees within the forest. To achieve these, the following is adopted: (i) minimise the individual error by growing the trees to the maximum depth and (ii) keep the residual correlation low through the randomisation processes including bagging and split randomisation.

In a RF or MRF, each tree is trained from a bootstrap sample of the training data, so each tree is trained based on a different dataset. Moreover, at each tree node, a subset of predictors is selected at random among all the features before determining the best split. Here, the size of the subset used is *p*/3. This randomness in the selected data and splits yields tree diversity. Averaging over diversified models improves the robustness of the predictions and reduces overfitting. Another key hyperparameter to set for this algorithm is the number of trees. In general, the accuracy of the RF will increase with increasing number of trees, but with increasing computational cost as the forest grows larger. We set the number of trees to 500.

One advantage of the MRF lies in the availability of the variable importance summary from the trained model. Here, we use the importance calculation method proposed within the RF-SRC package [53]. In this method, the VIMP of each input variable is calculated as the difference between the prediction error of a model that does not use that variable and the prediction error of the original trained model. To simulate the situation where the model does not include the variable, samples are propagated down each trained tree, by assigning the samples to the wrong child nodes (opposite split) whenever the encountered split is done based on the variable of interest. If a variable is relevant for the output prediction, such opposite splits should result in an increase in the prediction error and hence a higher VIMP value. The VIMP hence allow for an understanding

of which predictors have a significant impact on each output, hence contributing to a more straightforward interpretation of the model.

5. Input Features

To predict regions of high uncertainty in the flow solution, ML algorithms require information about the flow. Selection of appropriate IFs enhances the ability of the ML model to better predict flow patterns. However, no universally optimal method exists for selecting the ideal set of IFs to characterise flow phenomena. To maximise the applicability of these algorithms, inputs should encompass the key physics of the flow and ideally satisfy the governing laws of such flows. The inputs should also be non-dimensional to allow the results to exhibit good generalisation properties.

The inputs use only local flow variables so that they can operate on unstructured grids. The velocity field is the quantity of interest. The MRF is used to predict all three velocity components. To compare different scenarios with different sets of CFD data, dimensionless IFs are used. Here, the cell Reynolds number Re_{Δ} is taken as a local dimensionless feature, defined as

$$Re_{\Delta} = |u|\Delta/\nu$$

where $|u|$ is the cell velocity, Δ is the cell size and ν is the kinematic viscosity. The cell size is given as

$$\Delta = \sqrt{(\Delta x)^2 + (\Delta y)^2 + (\Delta z)^2}$$

where $\Delta x, \Delta y, \Delta z$ are the length of each cell in the x, y and z directions, respectively.

The local discrepancies of a flow variable due to insufficient mesh resolution are accounted for by considering a Taylor series expansion for a variable ϕ along the x-direction in a grid with a cell spacing of Δx [32].

$$\phi = \phi_0 + (\Delta x)\frac{d\phi}{dx}|_{\Delta x} + \frac{(\Delta x)^2}{2}\frac{d^2\phi}{dx^2}|_{\Delta x} + \cdots$$

Considering the first and second order terms from the Taylor series leads to the incorporation of the first derivative of local velocity multiplied by the cell size and the second derivative of local velocity multiplied by the cell size squared as IFs. These quantities are normalised by u_{inlet}.

$$\frac{(\Delta x_j)}{u_{inlet}}\frac{du_i}{dx_j}|_{\Delta x_j}$$

$$\frac{(\Delta x_j)^2}{u_{inlet}}\frac{du_{ik}}{dx_j}|_{\Delta x_j}$$

The term $\frac{(\Delta x_j)}{u_{inlet}}\frac{du_i}{dx_j}|_{\Delta x_j}$ comprises nine features as the velocity has three components and the derivative is performed with respect to x, y and z directions. Similarly, the term $\frac{(\Delta x_j)^2}{u_{inlet}}\frac{du_{ik}}{dx_j}|_{\Delta x_j}$ comprises 27 features, accounting for all the second derivatives for the three velocity components. These are the terms that characterise the grid-induced error.

To characterise the local regions of high turbulence, Reynolds stress terms are used as IFs. The Reynolds stress is normalised by the inlet velocity squared u_{inlet}^2, given as

$$\frac{R_{ij}}{u_{inlet}^2} = \frac{\overline{u'_{xx}}}{u_{inlet}^2}, \frac{\overline{u'_{yy}}}{u_{inlet}^2}, \frac{\overline{u'_{zz}}}{u_{inlet}^2}, \frac{\overline{u'_{xy}}}{u_{inlet}^2}, \frac{\overline{u'_{xz}}}{u_{inlet}^2}, \frac{\overline{u'_{yz}}}{u_{inlet}^2}$$

Further inputs are introduced to enable the results to exhibit good generalisation properties for turbulent flows. These inputs should be rotationally invariant and non-dimensional [17]. The following local variables are considered.

(1) The normalised Q-criterion

$$\frac{\|\Omega\|^2 - \|S\|^2}{\|\Omega\|^2 + \|S\|^2}$$

where $\Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right)$ is the anti-symmetric rotational tensor and $\|\Omega\|$ is its Frobenius norm. The term $S_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)$ is the symmetric rotational tensor and $\|S\|$ is the Frobenius norm.

(2) The pressure gradient along a streamline

$$\frac{u_k \frac{dP}{dx}_k}{\sqrt{\frac{dP}{dx}_j \frac{dP}{dx}_j u_i u_i} + |u_l \frac{dP}{dx_l}|}$$

where P is the pressure and u is the velocity.
(3) The ratio of the pressure normal stresses to normal shear stresses

$$\frac{\sqrt{\frac{dP}{dx}_i \frac{dP}{dx}_i}}{\sqrt{\frac{dP}{dx}_j \frac{dP}{dx}_j} + \frac{1}{2}\rho u_{kk}^2}$$

where ρ is the density.

(4) Vortex stretching

$$\frac{\sqrt{\omega_j \frac{du_i}{dx_j} \omega_k \frac{du_i}{dx_k}}}{\sqrt{\omega_l \frac{du_n}{dx_l} \omega_m \frac{du_n}{dx_m} + \|S\|}}$$

where ω is the vorticity and $\|S\|$ is the Frobenius norm of the symmetrical rotational tensor.

(5) Deviation from parallel shear flow

$$\frac{|u_k \frac{du_k}{dx_l}|}{\sqrt{u_n u_n u_i \frac{du_i}{dx_j} u_m \frac{du_m}{dx_j}} + |u_i u_j \frac{du_i}{dx_j}|}$$

(6) The normalised vorticity

$$\frac{|\omega| w_f}{u_{inlet}}$$

where ω is the vorticity and w_f is the bluff body width of the flameholder.

These inputs are crafted to convey as much of the relevant local flow information as possible in terms of physically relevant quantities. The terms also encapsulate the local pressure variations and vortical structures, which are critical quantities to characterise the bluff-body blow. The process of selecting these IFs is heuristic; we therefore investigate the proposed IFs through the current work using the VIMP approach.

In the original framework [32], we compared two sets of IFs. These IFs are highlighted in Table 3. The first set of IFs takes into account the local flow properties and grid-induced error. These variables are the local Reynolds number Re_Δ, the normalised first derivatives of the velocity vector $\frac{(\Delta x_j)}{u_{inlet}} \frac{du_i}{dx_j}|_{\Delta x_j}$ (9 terms) and the normalised second derivatives of the velocity vector $\frac{(\Delta x_j)^2}{u_{inlet}} \frac{du_{ik}}{dx_j}|_{\Delta x_j}$ (27 terms). The total number of IFs considered in the first set is 37. The second set of IFs augments the first set with additional variables that serve to capture and generalise turbulence. The

Scenario	Variables	Expression and Definition	Total number of features
Input Features Set 1 (IF1)	Local Reynolds number	Re_Δ	37
	Normalised first derivative of velocity	$\frac{(\Delta x_j)}{u_{inlet}} \frac{du_i}{dx_j}|_{\Delta x_j}$ (9 terms)	
	Normalised second derivative of velocity	$\frac{(\Delta x_j)^2}{u_{inlet}} \frac{du_{ik}}{dx_j}|_{\Delta x_j}$ (27 terms)	
Input Features	Local Reynolds number	Re_Δ	49

Set 2 (IF2)	Normalised first derivative of velocity	$\frac{(\Delta x_j)}{u_{inlet}} \frac{du_i}{dx_j} \mid_{\Delta x_j}$ (9 terms)	
	Normalised second derivative of velocity	$\frac{(\Delta x_j)^2}{u_{inlet}} \frac{du_{ik}}{dx_j} \mid_{\Delta x_j}$ (27 terms)	
	Normalised Reynolds stresses	$\frac{R_{ij}}{u_{inlet}^2}$ (6 terms)	
	Non-dimensionalised Q-criterion	$\frac{\|\Omega\|^2 - \|S\|^2}{\|\Omega\|^2 + \|S\|^2}$	
	Pressure gradient along streamline	$\frac{u_k \frac{dP}{dx_k}}{\sqrt{\frac{dP}{dx_j}\frac{dP}{dx_j} u_i u_i + \mid u_l \frac{dP}{dx_l} \mid}}$	
	Ratio of pressure normal stresses to normal shear stresses	$\frac{\sqrt{\frac{dP}{dx_i}\frac{dP}{dx_i}}}{\sqrt{\frac{dP}{dx_j}\frac{dP}{dx_j}} + 0.5\rho u_{kk}^2}$	
	Vortex stretching	$\frac{\sqrt{\omega_j \frac{du_i}{dx_j} \omega_k \frac{du_i}{dx_k}}}{\sqrt{\omega_l \frac{du_n}{dx_l} \omega_m \frac{du_n}{dx_m}} + \|S\|}$	
	Deviation from parallel shear flow	$\frac{\mid u_k \frac{du_k}{dx_l} \mid}{\sqrt{u_n u_n u_i \frac{du_i}{dx_j} u_m \frac{du_m}{dx_j}} + \mid u_i u_j \frac{du_i}{dx_j} \mid}$	
	Normalised vorticity	$\frac{\mid \omega \mid w_f}{u_{inlet}}$	

Table 3 Input feature sets 1 and 2 used in the original framework

Scenario	Variables	Expression and Definition	Total number of features
Input Features Set 3 (IP3)	Local Reynolds number	Re_Δ	55
	Normalised first derivative of velocity	$\frac{(\Delta x_j)}{u_{inlet}} \frac{du_i}{dx_j} \mid_{\Delta x_j}$ (9 terms)	
	Normalised second derivative of velocity	$\frac{(\Delta x_j)^2}{u_{inlet}} \frac{du_{ik}}{dx_j} \mid_{\Delta x_j}$ (27 terms)	
	Normalised Reynolds stresses	$\frac{R_{ij}}{u_{inlet}^2}$ (6 terms)	

	Non-dimensionalised Q-criterion	$\frac{\|\Omega\|^2 - \|S\|^2}{\|\Omega\|^2 + \|S\|^2}$	
	Pressure gradient along streamline	$\frac{u_k \frac{dP}{dx}_k}{\sqrt{\frac{dP}{dx}_j \frac{dP}{dx}_j u_i u_i + \lvert u_l \frac{dP}{dx_l} \rvert}}$	
	Ratio of pressure normal stresses to normal shear stresses	$\frac{\sqrt{\frac{dP}{dx}_i \frac{dP}{dx}_i}}{\sqrt{\frac{dP}{dx}_j \frac{dP}{dx}_j} + 0.5\rho u_{kk}^2}$	
	Vortex stretching	$\frac{\sqrt{\omega_j \frac{du_i}{dx_j} \omega_k \frac{du_i}{dx_k}}}{\sqrt{\omega_l \frac{du_n}{dx_l} \omega_m \frac{du_n}{dx_m}} + \|S\|}$	
	Deviation from parallel shear flow	$\frac{\lvert u_k \frac{du_k}{dx_l} \rvert}{\sqrt{u_n u_n u_i \frac{du_i}{dx_j} u_m \frac{du_m}{dx_j}} + \lvert u_i u_j \frac{du_i}{dx_j} \rvert}$	
	Normalised vorticity	$\frac{\lvert \omega \rvert w_f}{u_{inlet}}$	
	Basis of tensor invariants	$I = I_1, I_2, I_3, I_4, I_5, I_6$ (6 terms)	

Table 4 Input feature set 3 used in the current framework

additional input features include the normalised Reynolds stress terms $\frac{R_{ij}}{u_{inlet}^2}$ (6 terms) and non-dimensionalised parameters to characterise the turbulence. We have demonstrated that the full set of input features IF2 better characterise the turbulent flow and is superior at correcting for the biases due to insufficient resolution and spurious flow behaviour, providing more accurate and consistent predictions [32].

For completeness, we address the need to account for invariance properties for the fluid flow in the current work. An effective method of embedding an invariance property into a ML model is to formulate the inputs of the model such that they are invariant [18]. A function is invariant under a given group if its output value does not change when its inputs are subjected to the transformation specified by any element in that group. Details into the formulations of the invariance properties can be found in relevant literatures [54-56].

The Navier-Stokes equations obey Galilean invariance. This invariance dictates that the Reynolds stress anisotropy invariant at a given point in the flow will not change if the reference frame is translated, rotated or reflected. To enforce this, the basis of the invariants I constructed using the symmetric rotational tensor $S = \frac{1}{2}(\frac{du_i}{dx_j} + \frac{du_j}{dx_i})$ and anti-symmetric rotational tensor $\Omega = \frac{1}{2}(\frac{du_i}{dx_j} - \frac{du_j}{dx_i})$ are considered, where both S and Ω are normalised by $\frac{(\Delta x_j)}{u_{inlet}}$.

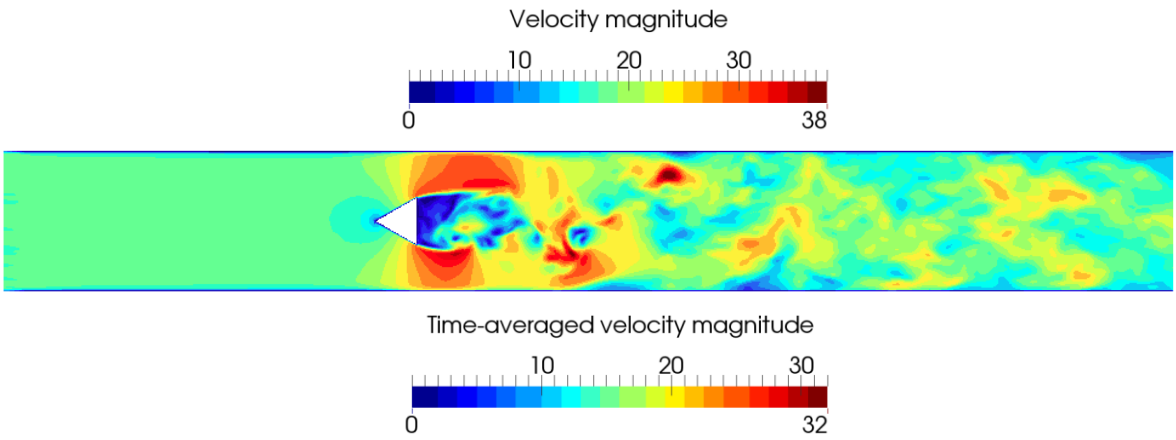
$$I_1 = Tr(S^2)$$
$$I_2 = Tr(S^3)$$
$$I_3 = Tr(\Omega^2)$$
$$I_4 = Tr(\Omega^2 S)$$
$$I_5 = Tr(\Omega^2 S^2)$$
$$I_6 = Tr(\Omega^2 S \Omega S^2)$$

In this work, we include the additional six bases of invariance into the second set of IFs described earlier. The full set of IFs considered in the current work is summarised in Table 4.

6. Results and Observations

The instantaneous and time-averaged velocity magnitude for the fine grid at u_{inlet} = 17 m/s is shown in Fig. 6. The instantaneous solution (top row) depicts vortex shedding behind the bluff body and the breakdown of vortices into smaller-scale turbulence structures further downstream. In the time-averaged solution (bottom row), the shedding behaviour is manifested as high-velocity flow separation around the bluff-body edges, accompanied by a low-velocity region behind the bluff-body. The ML-CFD framework intends to reproduce these flow features.

Figure 7 presents the profiles of the time-averaged axial velocity (top) and time-averaged transverse velocity (bottom) at five different axial locations downstream of the bluff body. These axial locations correspond to 0.375 w_f, 0.95 w_f, 1.5 w_f, 3.75 w_f and 9.4 w_f downstream of the bluff-body, where w_f is the bluff body side length. Results from the fine grid (solid line) and coarse grid (dashed line) are compared against the experimental measurements (black dots) [33, 34]. The axial and transverse velocities are normalised against the inlet velocity u_{inlet} and the height is normalised against the bluff body side length w_f.



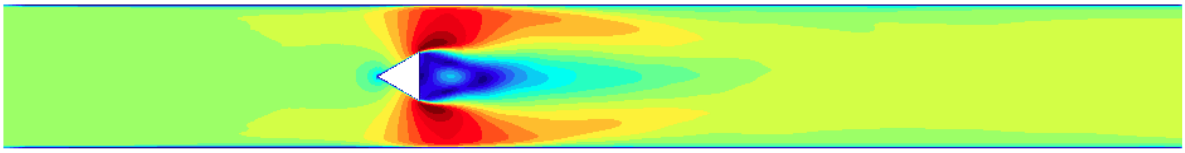


Figure 6 Instantaneous velocity magnitude (top) and time-averaged velocity magnitude (bottom) on the centreline cut plane at u_{inlet} = 17 m/s using the fine grid with 3.5 M cells. Units in [m/s]

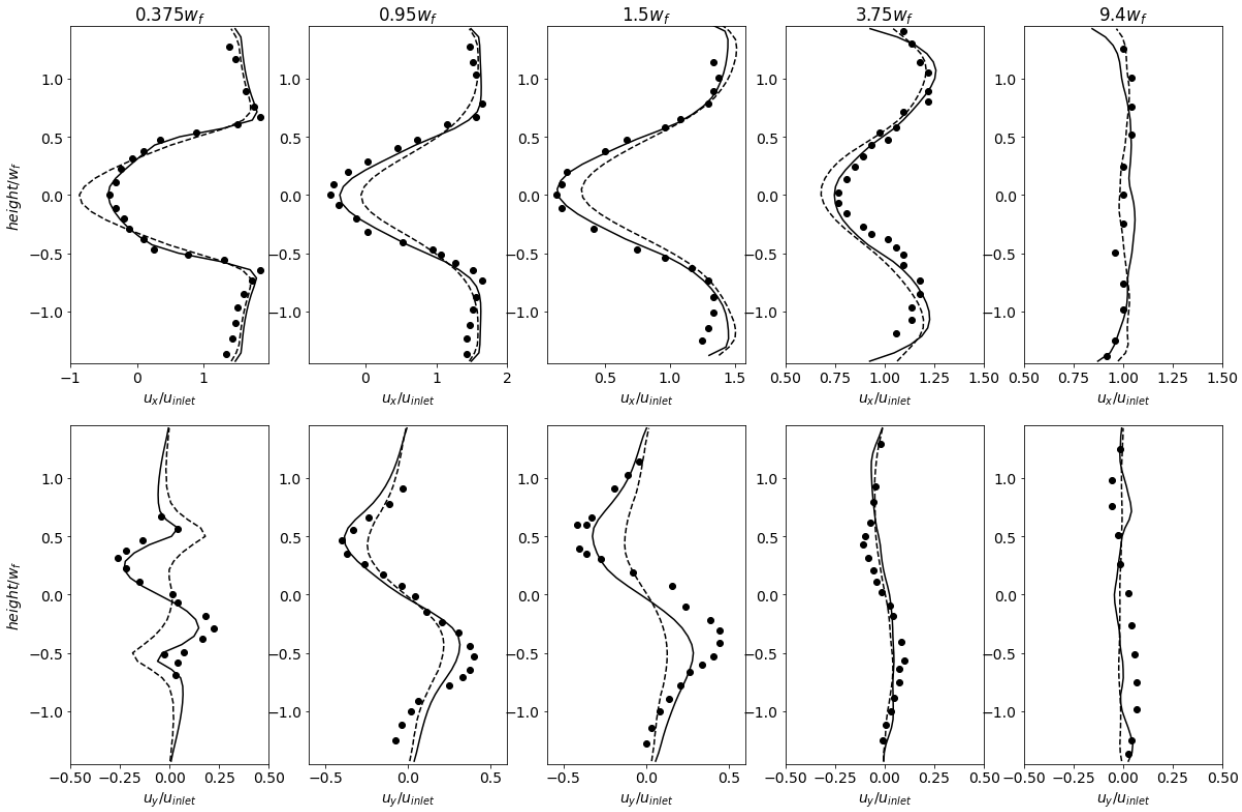


Figure 7 Profiles of the time-averaged axial velocity (top) and time-averaged transverse velocity (bottom) for u_{inlet} 17 m/s. Fine mesh (solid line), coarse mesh (dashed line) and black dots (experiment)

A strong flow reversal is present behind the bluff-body, as indicated by the negative values in the time-averaged axial velocity profiles (top row). The fine grid solution (solid line) matches the experimental measurements across all axial locations, while the coarse grid (dashed line) shows discrepancies particularly in centreline values where height/w_f = 0. For the time-averaged transverse velocity (bottom row), anti-symmetrical peaks are evident in the shear layer where height/w_f = 0.5 and -0.5. The fine grid (solid line) aligns reasonably well with measurements, though some under-predictions occur in the downstream locations. By contrast, the coarse grid (dashed line) fails to capture the transverse velocity profile accurately, showing notable discrepancies at 0.375 w_f, 0.95 w_f and 1.5 w_f. Overall, the comparison highlights the limitations of the coarse grid in replicating the flow field compared to the fine grid solution.

A comparison of the time-averaged velocity magnitude between the fine grid (top), mapped solution (middle) and coarse grid (bottom) is shown in Fig. 8 for cases with u_{inlet} = 17 m/s. The fine grid solution, consisting of 3.5 M cells (top), is considered the 'ground truth'. To compute the error between the fine grid and coarse grid in the ML training process, the fine grid solution is mapped onto the same computational grid used for the coarse grid solution. This results in the mapped solution (middle). The coarse grid solution (bottom), run with a reduced cell count of 0.4 M, produces a lower-accuracy solution. The differences between the fine grid (top) and mapped solution (middle) can be regarded as small and are primarily due to interpolation errors between these cases. By contrast, noticeable differences in the flow field and recirculation structure immediately behind the bluff body are evident between the mapped solution (middle) and coarse grid (bottom) cases.

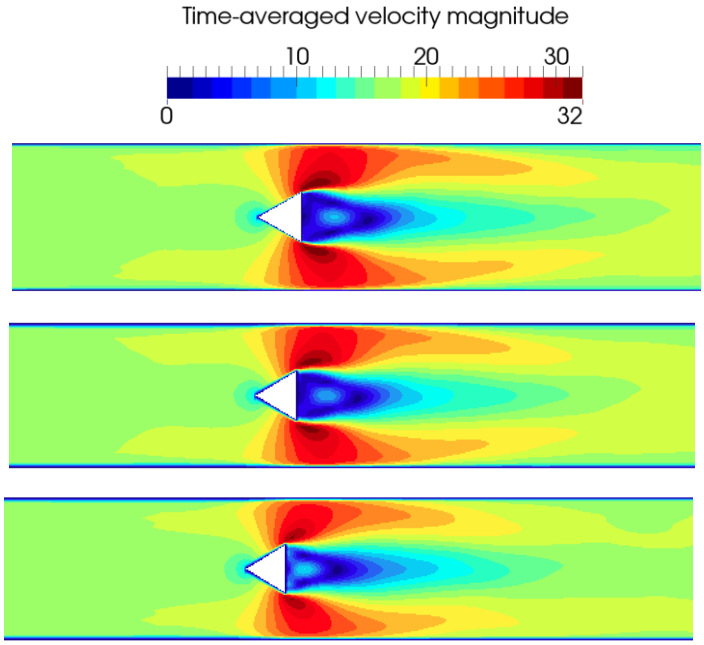


Figure 8 Time-averaged velocity magnitude for the fine grid (top), mapped solution (middle) and coarse grid (bottom) on the centreline cut plane at u_{inlet} = 17 m/s. Units in [m/s].

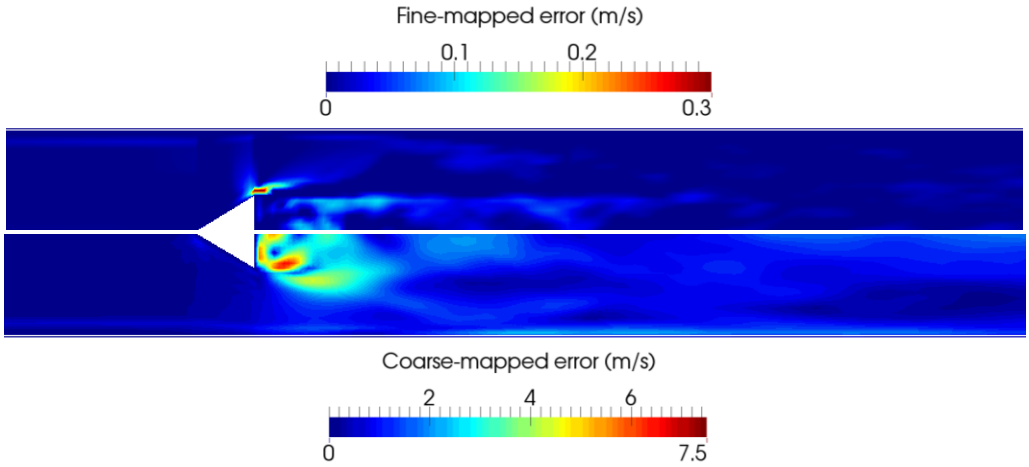


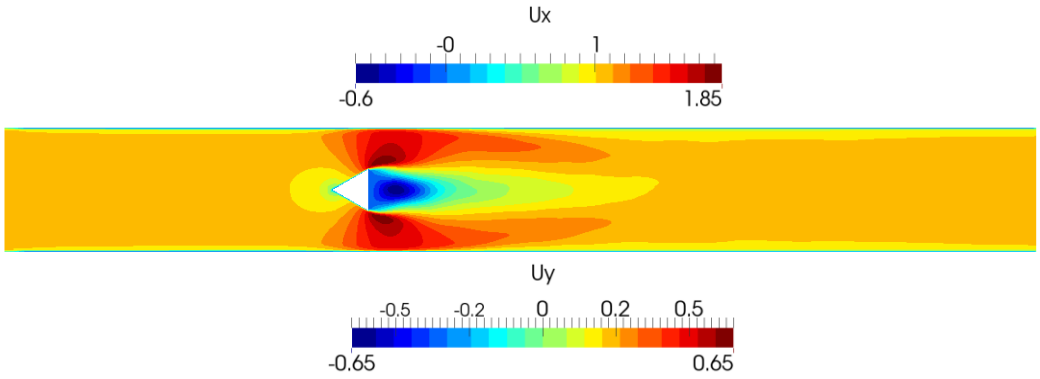
Figure 9 The error between fine grid and mapped solution (top half) and the error between the coarse grid and mapped solution (bottom half) on the centreline cut plane at u_{inlet} = 17 m/s. Units in [m/s].

Figure 9 depicts the error between the fine grid and mapped solution (top half) and the error between the coarse grid and mapped solution (bottom half) for u_{inlet} = 17 m/s. The errors due to mapping (top half) are significantly smaller in magnitude than the coarse grid (bottom half). Results from the fine grid are mapped onto the coarse grid using a second-order interpolation scheme. As the order-of-accuracy of the LES is consistent with the interpolation method used for the mapping, the error has a minimal impact on the accuracy of the results. This is evident in the fine-mapped error (top half), where a peak value of 0.3 m/s can be seen in the separation region. By contrast, the coarse-mapped error (bottom half) contains considerably larger values. A peak value of 7.5 m/s is evident. This value is 23% of the peak velocity magnitude of 32 m/s. The averaged error, within the recirculation region, takes a value of 3 m/s which is 10% of the peak velocity magnitude.

No attempt is made to compare the LES results at different inlet velocity conditions, as such a comparison has already been described [32]. As demonstrated previously, the coarse grid results exhibit deviations from the fine grid results and are not able to predict the physics of the flow accurately.

7. MRF Prediction

The MRF algorithm enables the prediction of the response of multiple variables simultaneously. In contrast to the RF model that treats each variable independently, the MRF algorithm considers the correlation and dependencies between variables when making predictions, leading to a more robust framework compared to the RF model. Here it is used to predict all three time-averaged, normalised velocity components for every cell in the computational domain. In the context of ML training and predictions, normalisation is achieved by the inlet velocity. The velocity components and velocity magnitude presented here are the time-averaged, normalised values denoted as U_x (axial velocity), U_y (transverse velocity), U_z, (spanwise velocity) and U_{mag} (velocity magnitude). We first focus on scenario 1. In this scenario, results from u_{inlet} = 8.5, 13 and 34 m/s are used to train the RF model, which is then evaluated on the data simulated with u_{inlet} = 17 m/s.



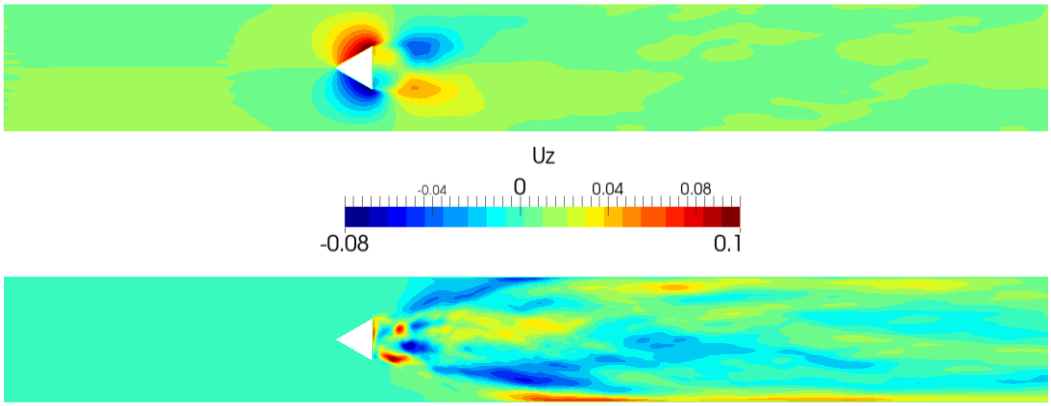


Figure 10 Time-averaged normalised axial velocity, U_x (top row), normalised traverse velocity, U_y (middle) and normalised spanwise velocity, U_z (bottom) on the centreline cut plane at u_{inlet} = 17 m/s for the fine-grid

Cross-sectional centreline planes showing the normalised velocity components: axial velocity U_x (top), transverse velocity U_y (middle) and spanwise velocity U_z (bottom) for the fine grid at u_{inlet} = 17 m/s are presented in Fig. 10. The axial velocity (top) indicates the presence of flow acceleration and reversal. Positive values are present at the edge of the bluff-body due to the flow acceleration from the separation process. Negative values are seen behind the bluff-body due to the existence of the recirculation bubble. The transverse velocity (middle) indicates translational flow movement, as demonstrated by alternating signs on the top and bottom of the bluff-body. A positive value above the y-axis suggests an upward movement of the flow, and vice versa. The translational movement, combined with the axial movement results in the flow rotation. In contrast to the axial and transverse velocity components that exhibit distinct behaviour, the spanwise velocity U_z (bottom) takes a small value and is less coherent. This velocity component is primarily driven by the vortex breakdown in the spanwise direction. Such a behaviour is dictated by turbulent small-scale vortices that are chaotic in nature.

Figure 11 shows the scatter plots for U_x (left), U_y (middle) and U_z (right). Each scatter plot shows a comparison of the fine-grid velocity component $U_{f,m}$ on the y-axis against three variables on the x-axis: (i) fine-grid velocity component $U_{f,m}$ (black symbols); (ii) coarse-grid velocity component U_c (blue symbols) and predicted velocity component U_{pred} (red symbols) obtained using the MRF algorithm. There are a few notable observations from this plot. Firstly, the values of the velocity components decrease in the order of U_x (left), U_y (middle) to U_z (right), evident from the peak values on the x- and y- axes from each plot. This observation is consistent with the cross-sectional planes presented in Fig. 10. Second, despite the decrease in the absolute value, the level of scatter increases from U_x, U_y and U_z (left to right). To quantify the level of scatter, we compute and compare the error in the next section. Lastly, the ML prediction (red) provides improved prediction for all three velocity components, showing reduced scatter relative to the coarse-grid CFD (blue).

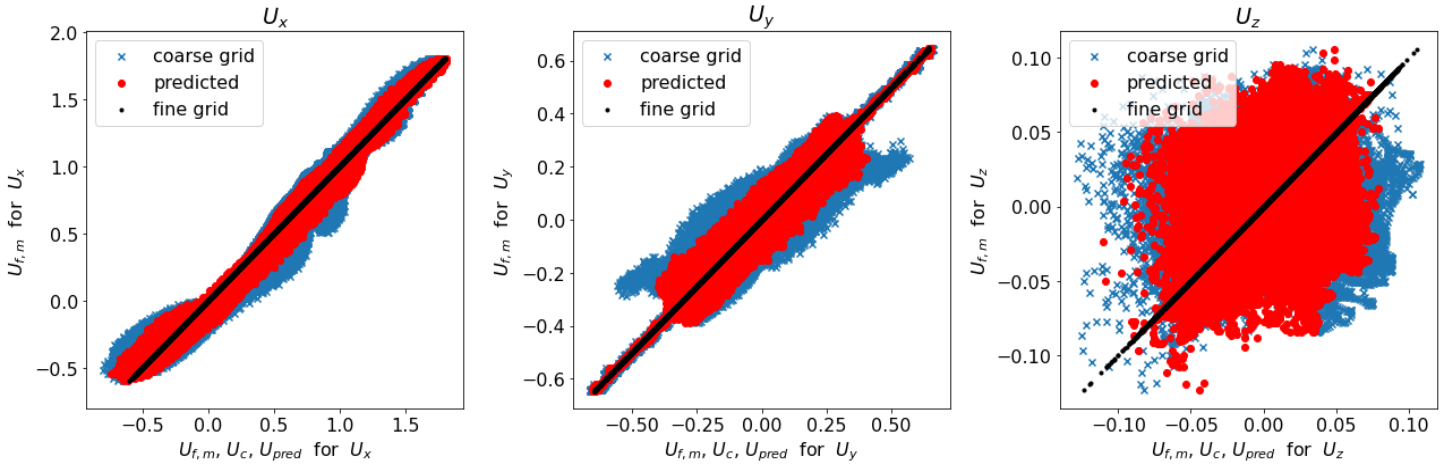


Figure 11 Scatter plot showing the distribution of the normalised axial velocity (left), normalised traverse velocity (middle) and normalised spanwise velocity (right) for the evaluated flow at u_{inlet} = 17 m/s in scenario 1: fine grid (black lines), coarse grid (blue cross) and MRF predictions (red circle)

Next, we compute (i) the local coarse-grid error ε_c, defined as the error between coarse grid and fine grid results $|U_c^{ev} - U_{f,m}^{ev}|$, and (ii) the local prediction error ε_{pred}, defined as the error between prediction and fine grid results $|U_{pred}^{ev} - U_{f,m}^{ev}|$ for each normalised velocity component. The computed errors across the centreline cross-sectional plane are shown in Fig. 12 for U_x (top), U_y (middle) and U_z (bottom). On the left column is the local coarse grid error and on the right column is the local prediction error. The colourbar in the left and right columns is set to a consistent scale. The local errors attributed to predictions (right column) are lower than those from the coarse grid (left column) for the x- and y-velocity components (top and middle rows). The peak prediction error (right column) has decreased by almost 50% relative to the coarse grid (left column) for both these velocity components. This level of improvement is particularly significant in key regions of the flow such as the recirculation zones. However, for the z-velocity component (bottom), the prediction errors are comparable to those of the coarse grid solution.

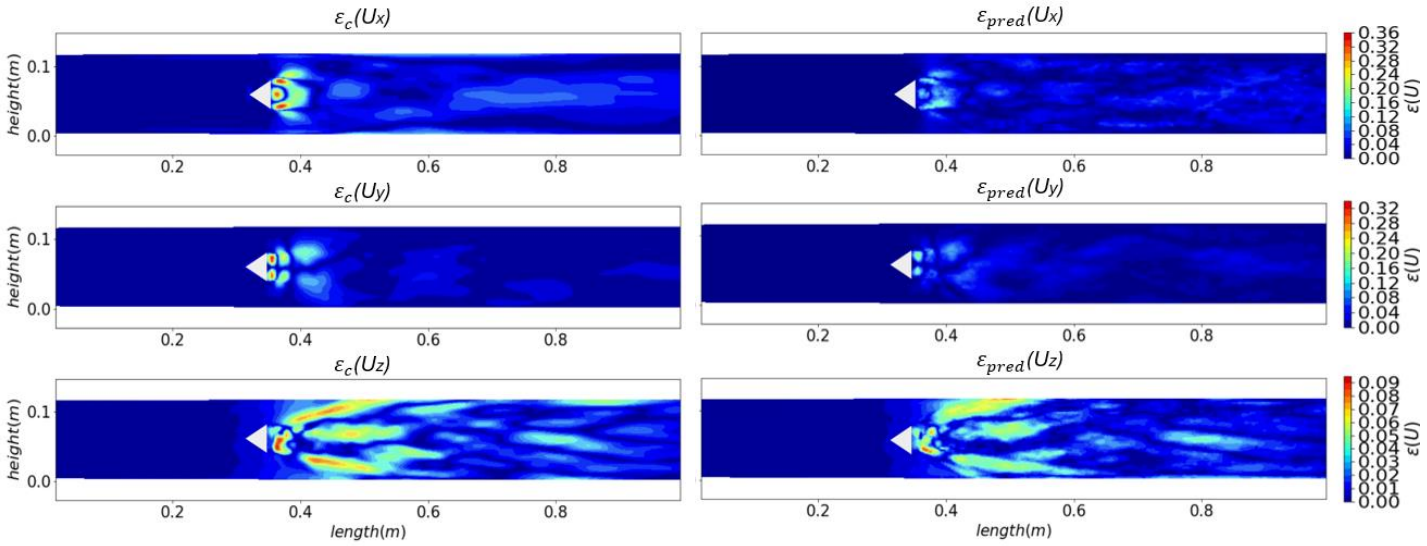


Figure 12 Local distribution of errors across the centreline cut plane (i) between coarse grid and fine grid (left); and (ii) between prediction and fine grid (right) for the normalised axial velocity (top), normalised traverse velocity (middle) and normalised spanwise velocity (bottom) for u_{inlet} = 17 m/s in scenario 1

To quantify the global error, the error metrics *E* are introduced. The metrics are computed for the prediction and coarse grid results, respectively. These metrics are the prediction error E_{pred} and coarse grid error E_c, defined as:

$$E_{pred}(U) = \overline{\varepsilon_{pred}(U)} = \overline{\left|U_{pred}^{ev} - U_{f,m}^{ev}\right|}$$

$$E_c(U) = \overline{\varepsilon_c(U)} = \overline{\left|U_c^{ev} - U_{f,m}^{ev}\right|}$$

The prediction error describes the error between the prediction and fine grid velocities $\overline{\left|U_{pred}^{ev} - U_{f,m}^{ev}\right|}$. The coarse grid error describes the error between the coarse grid and fine grid velocities $\overline{\left|U_c^{ev} - U_{f,m}^{ev}\right|}$. To obtain these errors, the predicted velocity or coarse grid velocity relative to the fine grid velocity is computed for every cell and averaged across the entire computational domain, denoted by the overbars. These metrics provide a measure of averaged error within the computational domain and allow the ML performance for different cases to be compared.

The standard deviations SD associated with the prediction error and coarse grid error, defined below, are also computed.

$$SD_{pred}(U) = \sqrt{\frac{\sum_{n=1}^{N}\left(\varepsilon_{pred}(U) - \overline{\varepsilon_{pred}(U)}\right)^2}{N}}$$

$$SD_c(U) = \sqrt{\frac{\sum_{n=1}^{N}\left(\varepsilon_c(U) - \overline{\varepsilon_c(U)}\right)^2}{N}}$$

where *N* is the number of cells in the computational domain and *n = 1…N*. The standard deviation provides a measure of the scatter in the error relative to its mean global value. A low scatter is required to ensure that a robust prediction is achieved. In the results section, all reported global errors will be accompanied by the standard deviation, and presented in the form of $E_{pred}(U) \pm SD_{pred}(U)$ for the prediction and $E_c(U) \pm SD_c(U)$ for the coarse grid. The global error and standard deviation are formulated for each velocity component and the velocity magnitude.

Table 5 summarises the coarse grid error E_c and prediction error E_{pred} for each velocity component, U_x, U_y, U_z (rows 1-3) and the velocity magnitude U_{mag} (row 4) for scenario 1. The standard deviations are presented alongside the errors. We compare the coarse grid error (left) against the errors from the RF predictions (middle) and MRF predictions (right). For both MRF and RF, the axial velocity U_x (row 1) exhibits the largest reduction in error, followed by U_y (row 2) and U_z (row 3). This observation is consistent with the error plot for the centreline cross-sectional plane in Fig. 12. The reason for the minimal improvement in U_z is because U_z takes small values and is governed by the chaotic nature of the small-scale vortices (see Fig 10). The velocity magnitude U_{mag} is reconstructed based on each velocity component and its error is presented in row 4. The prediction error for U_{mag} is lower for MRF (right) and RF (middle) compared to the coarse grid results (left). The reduction in error for U_{mag} is primarily contributed by the greatest reduction obtained from U_x. More importantly, MRF (right) delivered a better prediction than RF (middle). The prediction error for MRF is almost 33% lower than the coarse grid counterpart, compared to 26% for RF. Lower standard deviations are also evident for all velocity components in the MRF and RF predictions. The lower standard deviations in MRF relative to RF demonstrate its robustness in reducing the scatter.

	Coarse grid error, E_c and standard deviation, SD_c (%)	**RF prediction error, E_{pred} and standard deviation, SD_{pred} (%)**	**MRF prediction error, E_{pred} and standard deviation, SD_{pred} (%)**
U_x	4.273 ± 0.007	3.139 ± 0.0058	2.889 ± 0.0048
U_y	1.909 ± 0.0042	1.266 ± 0.0029	1.252 ± 0.0027
U_z	1.667 ± 0.0027	1.466 ± 0.0025	1.474 ± 0.0024
U_{mag}	4.282 ± 0.0071	3.142 ± 0.0058	2.894 ± 0.0048

Table 5 Coarse grid error and standard deviation (left), prediction error and standard deviation for RF (middle) and prediction error and standard deviation for MRF (right) in scenario 1. Normalised velocity components (rows1-3) and normalised velocity magnitude (row 4)

Next, we compare the performance of MRF against RF for the velocity magnitude U_{mag} in scenarios 1-4. The value of U_{mag} is reconstructed using the three velocity components U_x, U_y and U_z. In MRF, all three velocity components can be obtained within a single training instance. By contrast, three separate training and prediction instances are required to obtain all three velocity components for RF. Table 6 summarises the coarse grid error (left), prediction error for RF (middle) and prediction error for MRF (right) for scenarios 1-4. The reported errors are

accompanied by their standard deviations. For both RF and MRF, the prediction errors for the interpolation scenarios (1 and 3) are lower compared to the extrapolation scenarios (2 and 4). Also, the differences in prediction error between MRF and RF are lower for scenarios involving extrapolation (2 and 4). These observations suggest that extrapolation is inherently more challenging to predict. However, irrespective of whether an interpolation or extrapolation mode, MRF provided improvements relative to RF for all cases due to its ability to account for the correlation between the velocity components. The low standard deviation values for both RF and MRF predictions highlight that the reduction in scatter is also achieved compared to the coarse-grid values.

	Coarse grid error and standard deviation $E_c \pm SD_c$ (%)	**RF prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**	**MRF prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**
Scenario1	4.282 ± 0.0071	3.142 ± 0.0058	2.894 ± 0.0048
Scenario2	5.841 ± 0.0091	3.552 ± 0.0059	3.426 ± 0.0059
Scenario3	4.282 ± 0.0071	3.357 ± 0.0058	2.944 ± 0.0049
Scenario4	5.841 ± 0.0091	3.691 ± 0.0061	3.401 ± 0.0057

Table 6 Coarse grid error and standard deviation (left), prediction error and standard deviation for RF (middle) and prediction error and standard deviation for MRF (right) for the normalised velocity magnitude in scenarios 1-4

	RF runtime (s)	**MRF runtime (s)**
Scenario1	12257	3476
Scenario2	12418	3534
Scenario3	12727	3750
Scenario4	12606	3634

Table 7 The training runtimes of RF (left) and MRF (right) for scenarios 1-4

Table 7 shows the training runtimes of RF and MRF for scenarios 1-4. The runtimes correspond to the total time taken during training to compute all three velocity components U_x, U_y and U_z used to reconstruct U_{mag}. MRF achieves a significant computational saving of over 70% relative to RF. The savings offered are consistent across all scenarios. The MRF method is advantageous due to its capability to predict multiple outputs (all three velocity components) within a single training instance compared to the RF method which produces a single output at a time (one velocity component). The flexibility of the MRF model allows for straightforward extension to predict additional variables, further enhancing its utility and versatility.

8. Feature Interpretation and Reduction

In this section, we focus on interpreting the MRF model to identify the input features/variables that are crucial in characterising the flow physics. The aim is to reduce the dimensionality of the MRF model and remove any unnecessary noise in the data that might impact the accuracy of the model. Ultimately, this enables faster training and inference processes without compromising the accuracy of the predictions. To achieve this, a variable selection process is first conducted to identify and extract key features.

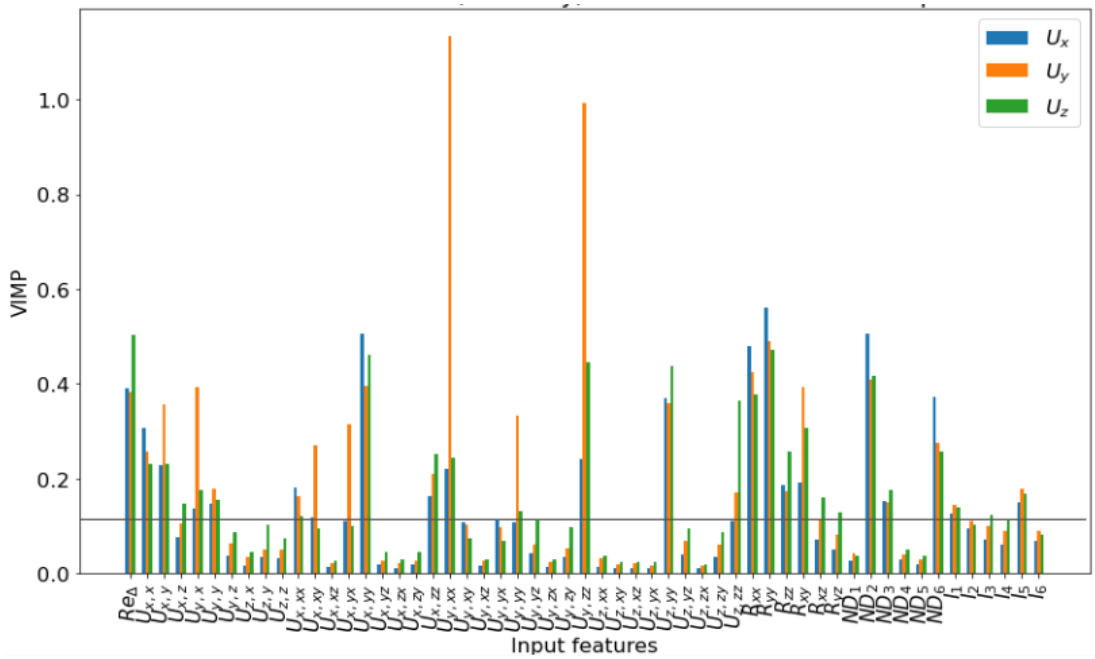


Figure 13 Bar chart of the VIMP values for all IFs used for the MRF prediction in scenario 1. U_x (blue bar), U_y (orange bar) and U_z (green bar)

The VIMP for scenario 1 is presented in Fig. 13. The y-axis denotes the VIMP value, and the x-axis denotes the IFs. Recall that for the IFs (see Table 4), we used the local Reynolds number Re_Δ , 9 terms for the normalised first derivative of velocity ($U_{x,x}$, - $U_{z,z}$), 27 terms for the normalised second derivative of velocity ($U_{x,xx}$ – $U_{z,zz}$), 6 terms for the normalised Reynolds stresses (R_{xx} – R_{yz}), 6 terms for the non-dimensionalised parameters to generalise turbulent flow (ND_1 - ND_6) and 6 terms for the basis of invariances I_1 - I_6. The IFs are arranged in this order on the x-axis. The VIMP is computed for all three velocity components U_x, U_y and U_z. Scenarios 2-4 exhibit similar trends in terms of feature importance and are therefore not shown here.

Features with high importance (large VIMP) and low importance (small VIMP) are present in Fig. 13. To discern the features that have larger contributions towards ML, an averaged VIMP is computed from all IFs, defined by the horizontal line at 0.12. Features that lie above this value are retained for subsequent training and those below are eliminated, apart from the bases of invariance which are retained. This gives rise to 30 important features, compared to the original 55. This represents almost a 50% reduction in the original IFs required for MRF prediction, which translates into a significant decrease in the computational time for training.

The physical relevance of the IFs is first interpreted. The magnitudes for the normalised first derivative of velocity (left), normalised second derivative of velocity (middle) and normalised Reynolds stresses (right) for the mapped solution at u_{inlet} = 17 m/s in scenario 1 are presented in Fig. 14.

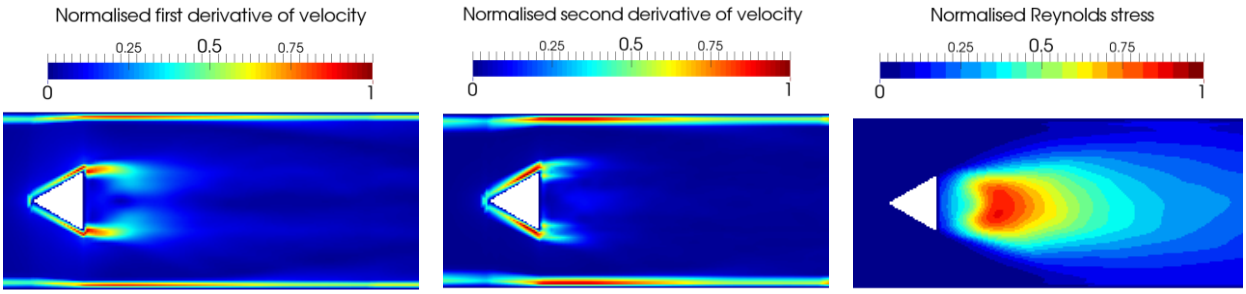


Figure 14 The magnitudes for the normalised first derivative of velocity (left), normalised second derivative of velocity (middle) and the normalised Reynolds stresses (right) on the centreline cut plane for the mapped solution at u_{inlet} = 17 m/s in scenario 1

For the normalised first derivative of velocity (left), high values are observed in three key regions: (i) the boundary layers along the bluff body surface, (ii) the separated shear layer, and (iii) the boundary layers along upper and lower walls of the duct. This behaviour arises due to the interaction of flow with the no-slip walls. At the bluff body edges where flow separation occurs, high values are initially present but reduce along the shear layer due to viscous dissipation. For the normalised second derivative of velocity (middle), high values are seen primarily along the boundary layers and separation zones at the bluff body edges. This term contributes to the viscous dissipation of the flow. The normalised Reynolds stresses (right) exhibit high values in the recirculation region behind the bluff body, indicating substantial turbulent fluctuations and momentum transfer in this region. Collectively, these IFs characterise the flow behaviour close to the bluff body and address the regions where the errors are significant (see Fig. 9).

From the VIMP analysis, the normalised first derivatives of velocity, normalised second derivatives of velocity and normalised Reynolds stresses in the z-direction exhibit low importance. To investigate this, we compare the normalised first derivatives of the velocity (top), normalised second derivatives of the velocity (middle) and normalised Reynolds stress (bottom) in the x-direction (left) and in the z-direction (right) for the mapped solution at u_{inlet} = 17 m/s in Fig. 15. Two key observations can be made. Firstly, the quantities in the x-direction (left) capture the key flow variations around the bluff body, which are predominantly in the boundary layers, shear layers, and recirculation zone. By contrast, the quantities in the z-direction (right) only reproduce the incoherent, small-scale features in the flow. Secondly, the magnitudes for the quantities in the z-direction (right) are significantly lower than those in the x-direction (left). From these observations, it is evident the quantities in the z-direction have minimal contributions towards the ML predictions.

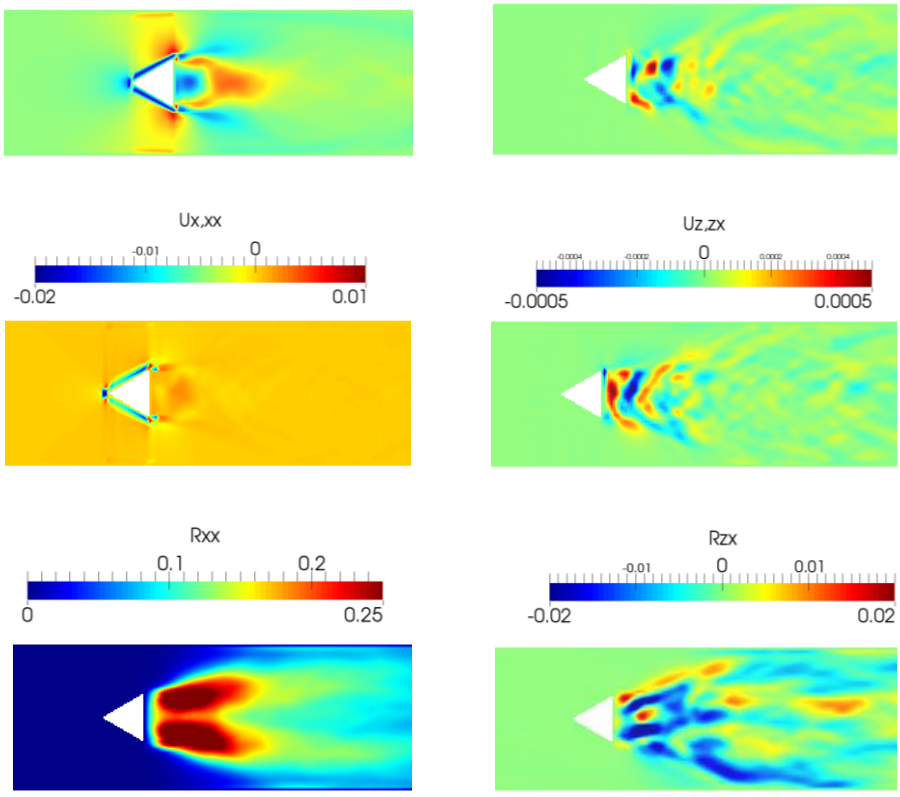
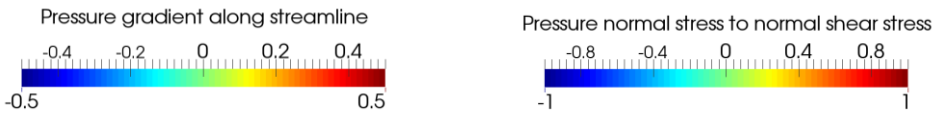


Figure 15 The normalised first derivative of velocity component (top), normalised second derivative of velocity component (middle) and the normalised Reynolds stress component (bottom) on the centreline cut plane for the mapped solution at u_{inlet} = 17 m/s in scenario 1. Components in the x-direction (left) and components in the z-direction (right)

Additional non-dimensionalised inputs were employed to encapsulate the local information essential for characterising turbulent flow. Figure 16 shows two IFs identified as important in the VIMP analysis: pressure gradient along the streamline ND_2 (left) and the ratio of pressure normal stresses to normal shear stress ND_3 (right). Both these quantities highlight the local pressure variations. A stagnation region is formed around the bluff body apex facing the incoming flow. In this region, steep pressure gradients are present. Such behaviour is expected due to the deceleration of the approaching flow, which generates a high localised static pressure at the stagnation point. Large pressure gradients are also evident close to the bluff body edges and extend to the top and bottom walls of the duct. In these regions, the flow separation, rotation and acceleration due to vortex shedding occur. Further variations are also present in the recirculation zone, where a low-pressure region exists behind the bluff body that acts as a blockage. Pressure variations are also apparent in the downstream region. These findings highlight the critical role of pressure variations in characterising the bluff body flow.



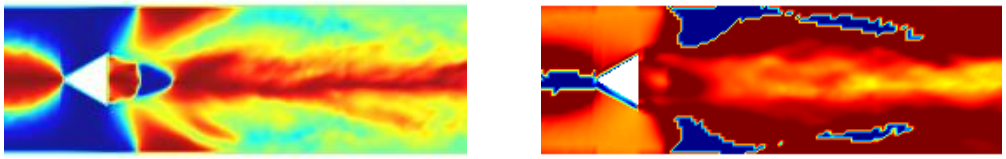


Figure 16 The pressure gradient along the streamline (left) and the ratio of pressure normal stress to normal shear stress (right) on the centreline cutplane for the mapped solution at u_{inlet} = 17 m/s in scenario 1

Figure 17 illustrates two tensor invariances I_1 (left) and I_2 (right) for the mapped solution at u_{inlet} = 17 m/s. These invariances capture the flow variations along the bluff body walls and the shear layer. A similar trend is observed for other invariances bases I_3 - I_6. The flow features captured by these invariances closely resemble those of the velocity derivatives magnitude (see Fig 14). This observation suggests that these quantities can identify the localised flow changes near the bluff body, particularly regions with large velocity gradients, but are limited in capturing variations over a larger extent in the domain. Consequently, these invariances may be less effective than other quantities in describing the flow features at different regions for the current configuration, as indicated by the VIMP analysis.

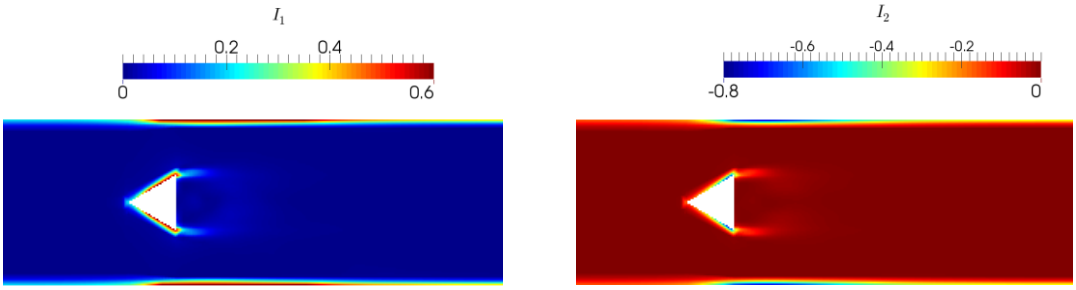


Figure 17 The first tensor invariance I_1 (left) and the second tensor invariance I_2 (right) on the centreline cut plane for the mapped solution at u_{inlet} = 17 m/s in scenario 1

Next, we assess the impact of the IFs of high importance (above the horizontal line in Fig. 13) and low importance (below the horizontal line in Fig. 13) on the flow field. To do this, the flow is first partitioned into distinct regions using two fundamental markers: normalised velocity magnitude U_{mag} and normalised vorticity $|\omega|w_f/u_{inlet}$. Threshold values are selected through an exploratory basis for these markers to partition the flow into distinct regions. The four regions of the flow, summarised below, are illustrated in Fig. 18.

Recirculation zone - $U_{mag} < 0.9$
Separation region - $U_{mag} > 1.1$
Shear and boundary layers - $0.9 \leq U_{mag} \leq 1.1$ conditioned upon normalised vorticity > 3.0
Upstream and wake - $0.9 \leq U_{mag} \leq 1.1$ conditioned upon normalised vorticity ≤ 3.0

To understand the influence of the IFs on the flow field in these specific regions, two IFs are selected from Fig. 13: one with high importance ($U_{x,x}$) and one with low importance ($U_{z,x}$), as summarised in Table 8. The histograms corresponding to different regions of the flow (i) recirculation zone (ii) upstream and wake (iii) shear and boundary layers (iv) separation region for $U_{x,x}$ and $U_{z,x}$ are presented in Fig. 19.

Importance	Input features
High	$U_{x,x}$
Low	$U_{z,x}$

Table 8 The selected input features used to understand the flow field in different regions in the domain

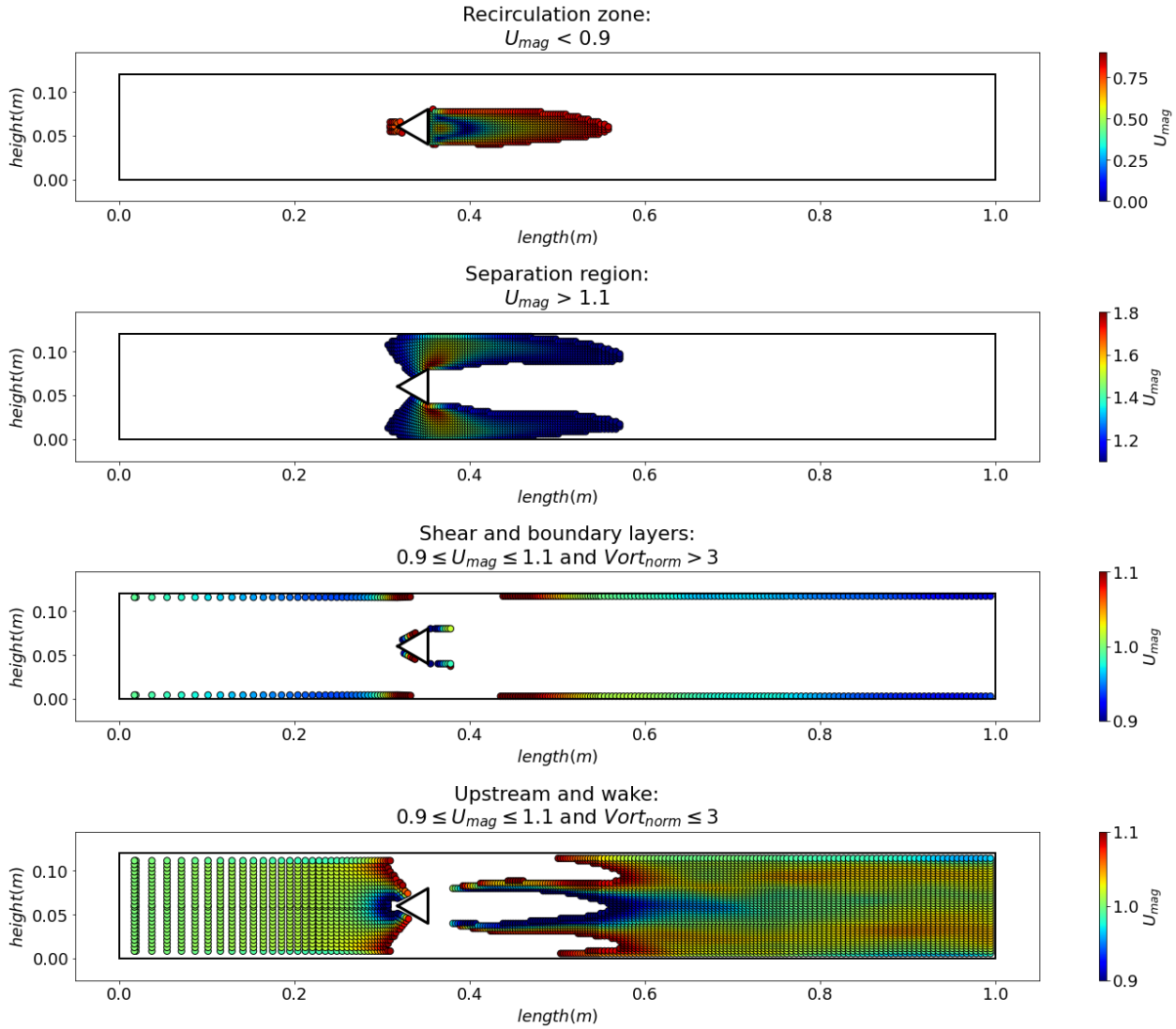


Figure 18 Contours indicating different regions of the flow on the centreline cut plane. The recirculation zone (first row), separation region (second row), shear and boundary layers (third row) and upstream and wake region (fourth row) for u_{inlet} = 17 m/s

The histogram for $U_{x,x}$ (Fig. 19, top row) highlights its contribution in different flow regions. Its contribution in the recirculation zone (green) and separation region (red) is larger. The increased contributions are characterised by the distribution of the histogram, which tends to be positively skewed for the recirculation (green), and largely negatively skewed for the separation region (red). This is followed by the boundary and shear layers (blue), where some skewness in the distribution remains. The lowest contribution is evident for the upstream region and wake (yellow), where

most of the values are centred around zero. This behaviour is expected since the flow variations in the upstream and wake regions are smaller, and less impact would be present.

The same histogram for $U_{z,x}$ (Fig. 19, bottom row) shows that most of its values tend to concentrate around zero. This behaviour is in contrast to $U_{x,x}$ (Fig. 19, top row) where skewness in the histogram is present. Here, the absence of a skewed distribution in different regions of the flow for $U_{z,x}$ suggests that the low-importance features only capture random noise instead of meaningful flow indicators. As such, these features are expected not to have an impact on the ML prediction and could be eliminated for training.

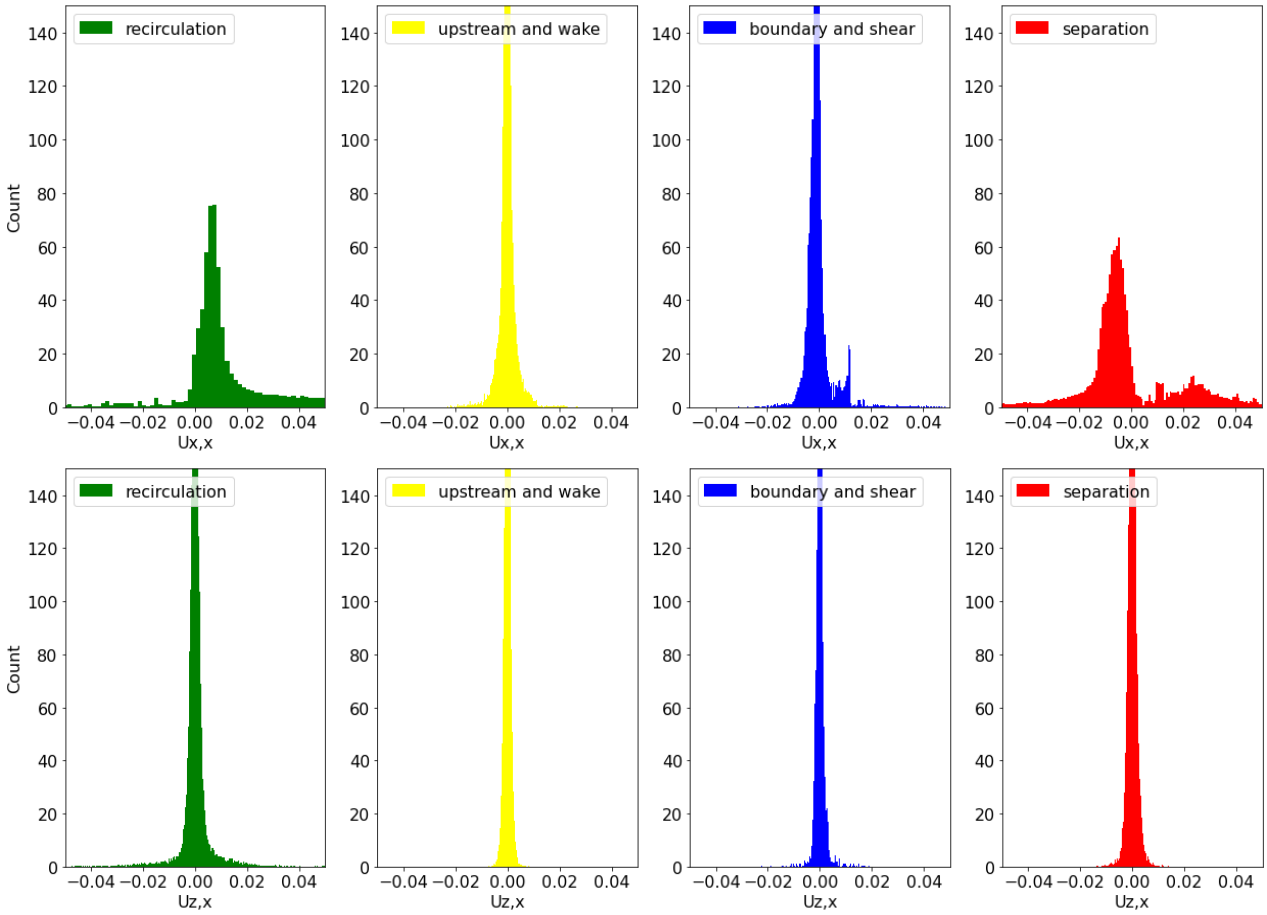
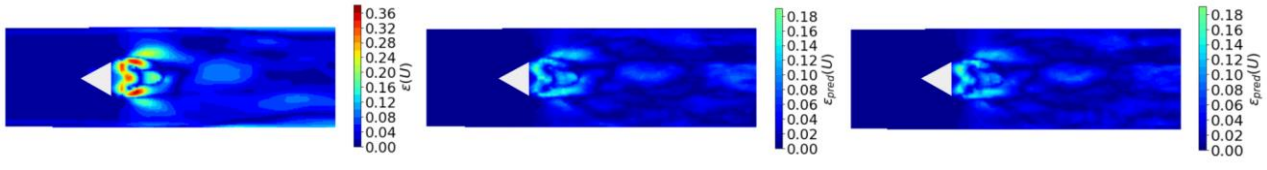


Figure 19 Histogram of $U_{x,x}$ (top) and $U_{z,x}$ (bottom) for four regions of the flow in scenario1: the recirculation zone (green), upstream and wake (yellow), boundary and shear layers (blue) and flow separation region (red)



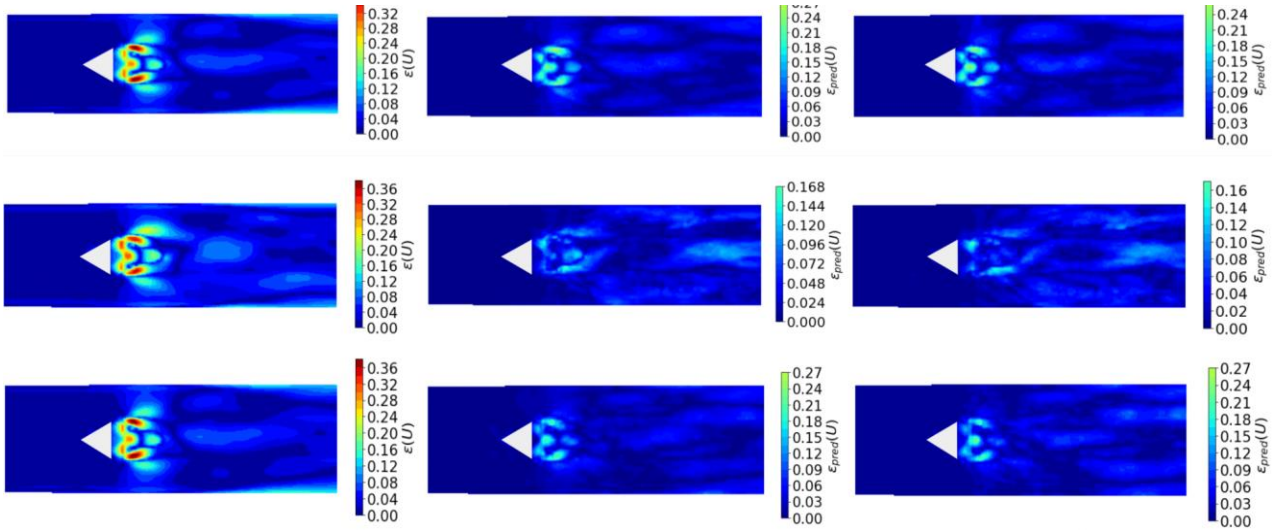


Figure 20 Local distribution of errors for the normalised velocity magnitude at the centreline cut plane. Scenarios 1-4 (top to bottom): (i) error between coarse grid and fine grid (left); (ii) error between MRF prediction with full set of IFs and fine grid (middle), error between MRF prediction with reduced set of IFs and fine grid (right)

Next, we train the ML model using the reduced set of IFs selected based on the VIMP. The performance of the MRF model trained with reduced IFs is compared with the original model trained using the full set of IFs. Figure 20 illustrates the local errors ε of the normalised velocity magnitude U_{mag} for scenarios 1-4 (top to bottom) at the centreline cross-sectional planes. For each scenario, the left column represents the local coarse-grid error ε_c, calculated as the difference between the coarse grid and the fine grid. The middle column represents the local prediction error ε_{pred}, calculated as the difference between the prediction and the fine grid, for the original case trained using the full set of IFs. The right column illustrates the local prediction error for the case trained with reduced IFs.

For each scenario, it is evident that the local prediction error (middle) is lower than the local coarse grid error (left), particularly around the recirculation zone behind the bluff body. The peak local prediction error (middle) has decreased by approximately 50% relative to the coarse grid (left). For the cases trained with reduced IFs (right), the locations of the local prediction error remain consistent with the original prediction (middle). The peak error is also comparable to the original prediction (middle). This observation suggests that no detriment in the prediction is observed when the IFs are reduced, as long as the key features capable of capturing the flow field are retained when training the model.

The computed global errors for U_{mag} in scenarios 1-4 are summarised in Table 9. Table 9 comprises the coarse grid error E_c (left column); prediction error E_{pred} using full set of IFs (middle); and prediction error E_{pred} using reduced set of IFs (right). The standard deviations are presented alongside the global errors. Around 33% reduction in error is achieved with the predictions (middle and right columns) relative to coarse grid (left column). Additionally, the reduction in IFs does not affect the prediction error, corroborating previous observations in Fig. 20 for the cross-sectional

planes. The standard deviations also remained almost unchanged. The consistency in prediction error and standard deviation underscores the robustness of the MRF model, even when trained with a reduced set of IFs.

	Coarse grid error and standard deviation $E_c \pm SD_c$ (%)	**MRF prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**	**MRF reduced IFs prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**
Scenario1	4.282 ± 0.0071	2.894 ± 0.0048	2.905 ± 0.0047
Scenario2	5.841 ± 0.0091	3.426 ± 0.0059	3.423 ± 0.0059
Scenario3	4.282 ± 0.0071	2.944 ± 0.0049	3.012 ± 0.005
Scenario4	5.841 ± 0.0091	3.401 ± 0.0057	3.377 ± 0.0058

Table 9 Coarse grid error and standard deviation (left), prediction error and standard deviation for MRF with full IFs (middle) and prediction error and standard deviation for MRF with reduced IFs (right) for the normalised velocity magnitude in scenarios 1-4

Table 10 presents the training runtimes for both the original MRF model and the MRF model trained with reduced IFs across scenarios 1-4. In this instance, the runtime has decreased by approximately 50% with reduced IFs, while retaining the accuracy of the original model. It is evident that the computational runtime for the MRF model scales approximately with the number of IFs. The current findings highlight the efficiency gains achieved by reducing IFs without compromising the predictive accuracy of the MRF model. Such an optimisation not only reduces computational overhead but also maintains the effectiveness of the model in capturing the underlying physics in the data.

	MRF runtime (s)	**MRF reduced IFs runtime (s)**
Scenario1	3476	1738
Scenario2	3534	1659
Scenario3	3750	1685
Scenario4	3634	1817

Table 10 The training runtimes for MRF with full IFs (left) and MRF with reduced IFs (right) for scenarios 1-4

9. Assessment of Different Geometry

The capability of the proposed ML-CFD framework is extended to enable predictions of the flow field with different bluff body geometries. To achieve this, datasets from the parabolic, triangle and square bluff body shapes are used for training. The trained surrogate model is tested on an unseen bullet-shaped bluff body. The inlet velocity for all these cases is kept the same at u_{inlet} = 17 m/s. In this section, we investigate the flow field from the various geometries and present the ML prediction. The primary challenge is the ability of the MRF approach to learn the physics associated with the variations in geometry and apply the prediction to a new configuration.

Figure 21 presents the instantaneous (top) and time-averaged (bottom) velocity magnitude for all three cases used for ML training: parabolic bluff body (left), triangular bluff body (middle) and square bluff body (right). In the instantaneous velocity contours (top), vortex shedding can be seen for all three cases. However, the flow separation locations are different between these cases. For the parabolic bluff body (left), the incoming flow first impinges on the apex facing the inlet direction resulting in a low velocity stagnation bubble. The flow then follows the contours of the curved upper and lower surfaces of the bluff body. Initially, the flow remains attached to the surface but begins to detach before reaching the bluff-body edges due to the curvature of the bluff body. For the triangular bluff body (middle), the flow separates at the upper and lower edges. By contrast, the square bluff body (right) presents a strong blockage to the incoming flow. The flow detaches immediately as it approaches the separation points at the first top and bottom edges of the square structure, creating a high separation velocity and short recirculation structure immediately behind the bluff body.

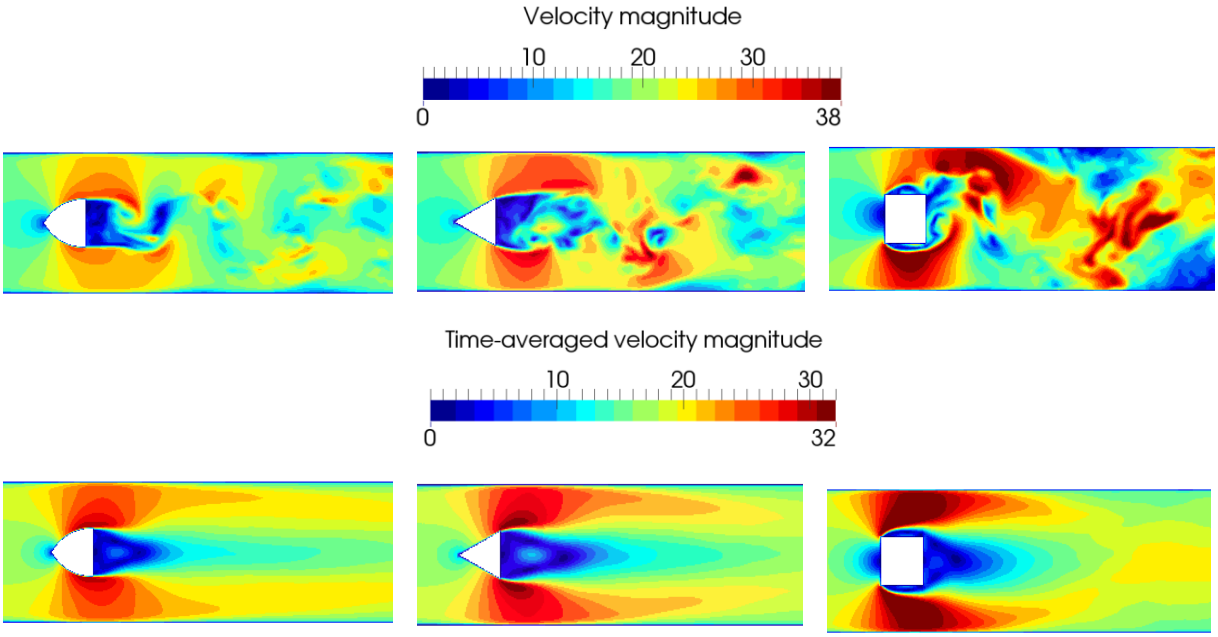


Figure 21 Instantaneous velocity magnitude (top) and time-averaged velocity magnitude (bottom) for the parabolic bluff body (left); triangular bluff-body (centre) and square bluff-body (right) on the centreline cut plane at u_{inlet} = 17 m/s using the fine grid with 3.5 M cells. Units in [m/s]

In the time-averaged contours (bottom), increasing magnitude in separation velocity can be seen in the order of the parabolic (left), triangle (middle) and square (right) bluff bodies. Each case also exhibits a distinctive recirculation structure due to the flow separation behaviour. The parabolic bluff body has the shortest recirculation length as a result of earlier flow detachment from the bluff body surface. By contrast, the square bluff body comprises a large stagnation bubble at the impingement point. Low velocity regions can also be seen at the upper and lower sections of the square bluff body due to the flow separation at the upper and lower edges.

Figure 22 depicts the instantaneous (top) and time-averaged (bottom) velocity magnitude of the bullet-shaped bluff body, which we intend to predict the flow field for. The extended rectangular section delays the detachment of the flow. Instead of separating at the edges as in the original triangular bluff body, the flow remains attached, allowing the boundary layer to grow in the extended section of the triangular structure. This leads to a smoother flow transition. As a result, a lower separation velocity and a smaller recirculation structure are evident compared to the triangular bluff body.

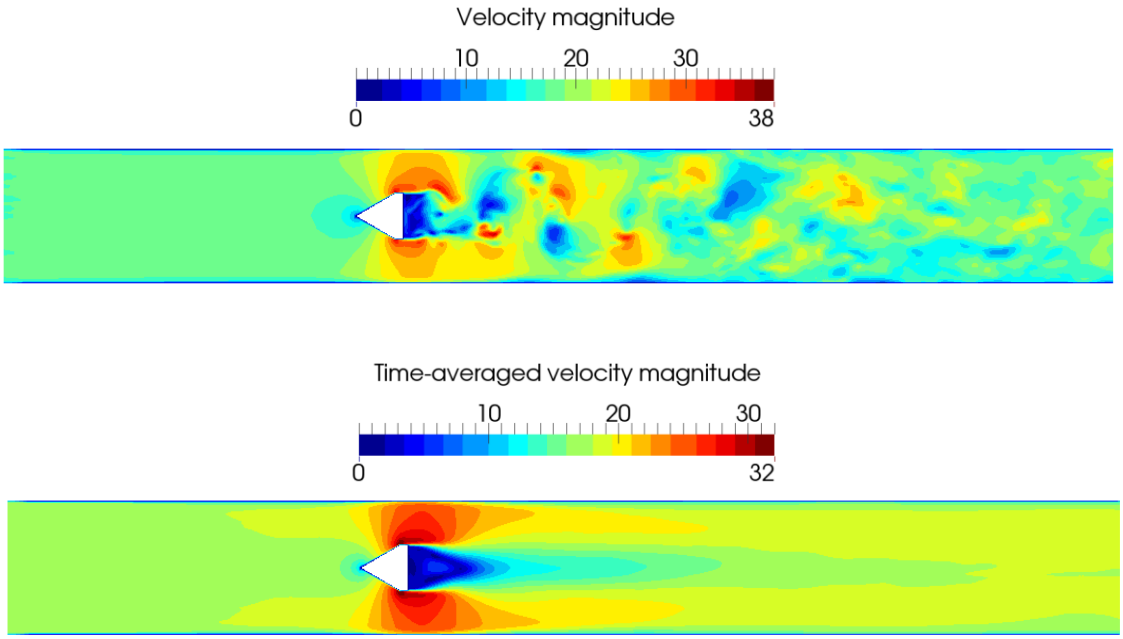
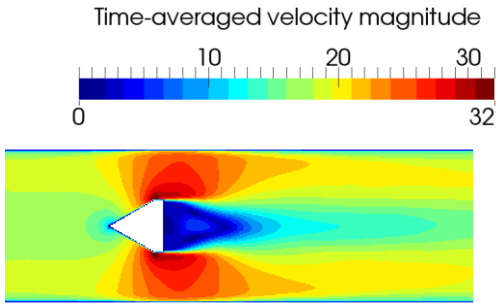


Figure 22 Instantaneous velocity magnitude (top) and time-averaged velocity magnitude (bottom) for the bullet-shape bluff body on the centreline cut plane at u_{inlet} = 17 m/s using the fine grid with 3.5 M cells. Units in [m/s]

A comparison of the time-averaged velocity magnitude between the mapped solution (top) and the coarse grid (bottom) for the bullet-shaped bluff body is presented in Fig 23. The coarse grid (bottom) reveals an elongated recirculation structure behind the bluff body. Additionally, the coarse grid predicts a zone of larger flow acceleration, extending marginally downstream along the top and bottom edges of the bluff body. These disparities stem from the inadequacy of the coarse grid in capturing the flow separation phenomena, highlighting the limitations due to insufficient grid resolution.



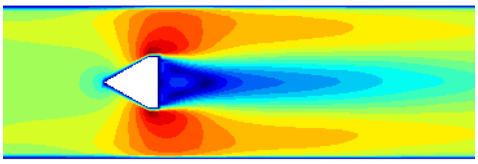


Figure 23 Time-averaged velocity magnitude for the bullet-shaped bluff-body at u_{inlet} = 17 m/s on the centreline cut plane: mapped solution (top) and coarse-grid (bottom)

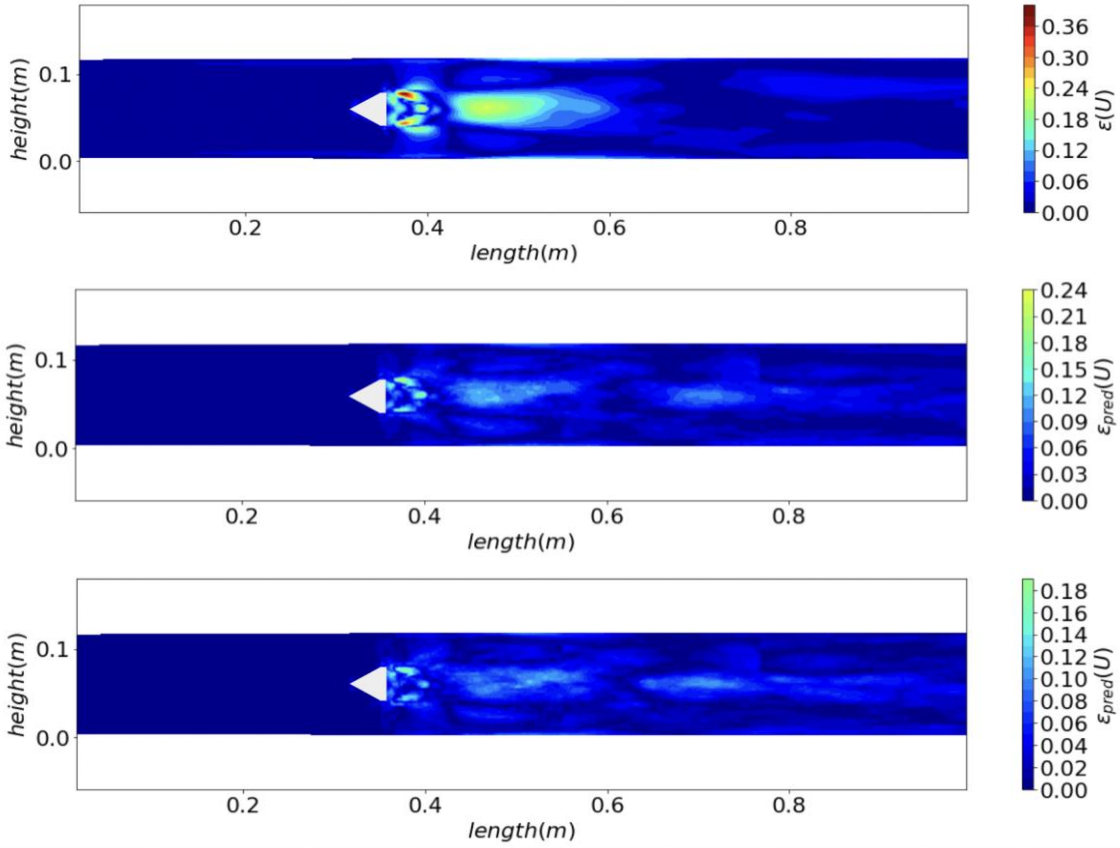


Figure 24 Local distribution of errors for the normalised velocity magnitude at the centreline cut plane for the bullet bluff body in scenario 5: error between coarse grid and fine grid (top); (ii) error between MRF prediction with full IFs and fine grid (middle), error between MRF prediction with reduced IFs and fine grid (bottom)

Next, we focus on the predictive capability of the proposed ML approach in rectifying the flow field derived from the coarse grid. Figure 24 shows the distribution of local errors for the normalised velocity magnitude U_{mag} at the centreline cross-sectional planes. The top row is the local coarse-grid errors ε_c, while the middle and bottom rows are the local prediction errors ε_{pred} obtained using MRF with the full set of IFs and a reduced set of IFs, respectively. The case with reduced IFs contains the same set of IFs as in the previous section. The local errors predominantly manifest in the shear layer and recirculation region. While the spatial locations of these errors remain consistent between the coarse grid (top) and predictions (middle and bottom), the peak errors from the predictions have notably diminished. Also evident are the negligible differences in

the local prediction errors between full IFs (middle) and reduced IFs case (bottom). This observation demonstrates that when training and predicting using datasets featuring configurations with different bluff body shapes, only the key IFs suffice to achieve prediction as accurate as the full set of IFs.

	Coarse grid error and standard deviation $E_c \pm SD_c$ (%)	**RF prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**	**MRF prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**	**MRF reduced IFs prediction error and standard deviation $E_{pred} \pm SD_{pred}$ (%)**
Scenario 5	4.051 ± 0.0067	$3.095\% \pm 0.0055$	$2.749\% \pm 0.0051$	2.839 ± 0.0051

Table 11 Coarse grid error and standard deviation (left), prediction error and standard deviation for MRF with full IFs (middle) and prediction error and standard deviation for MRF with reduced IFs (right) for the normalised velocity magnitude in scenario 5

Table 11 summarises the global error *E* and standard deviation *SD* of U_{mag} for the coarse grid (first column), RF predictions with full IFs (second column), MRF predictions with full IFs (third column) and MRF predictions with reduced IFs (fourth column). Two principal observations can be made. Firstly, both RF with full IFs (second column) and MRF with full IFs (third column) produce a lower error compared to the original coarse-grid (first column). However, MRF delivered improvement with further error reduction of 30% compared to RF at 25%, reflecting the increased ability of MRF to generalise and predict the flow field due to geometry changes. Secondly, MRF predictions using full IFs (third column) and a reduced set of IFs (fourth column) result in a comparable reduction in error compared to the original coarse-grid (first column). This demonstrates that using reduced IFs has minimal impact on prediction accuracy. The standard deviations across the MRF predictions using the full IFs (third column) and reduced IFs (fourth column) are also similar. Based on the findings in the present section, we demonstrate that the current framework can predict the flow field of a new, unseen configuration based on trained datasets from different geometries.

To quantify the accuracy of the predictions, profiles for the time-averaged velocity magnitude at locations 0.375 w_f, 0.95 w_f, 1.5 w_f, 3.75 w_f and 9.4 w_f downstream of the bullet-shape bluff body are shown in Fig. 25. The axial locations of these profiles are indicated by the cross-sectional view in Fig. 26. These profiles are compared for the fine grid (blue solid line), coarse grid (blue circle) and MRF predictions (orange circle). Close to the bluff-body at 0.375 w_f, and 0.95 w_f, an improved match is obtained between the predictions and fine grid profiles. These correspond to the key regions of interest for the current configuration. In the downstream location, the agreement of the prediction to the fine grid remains much better than the coarse grid, despite a slight deterioration in the accuracy compared to the fine grid solution. While the predictions may not

exactly resemble the fine grid solution, visible improvements are delivered using the current approach.

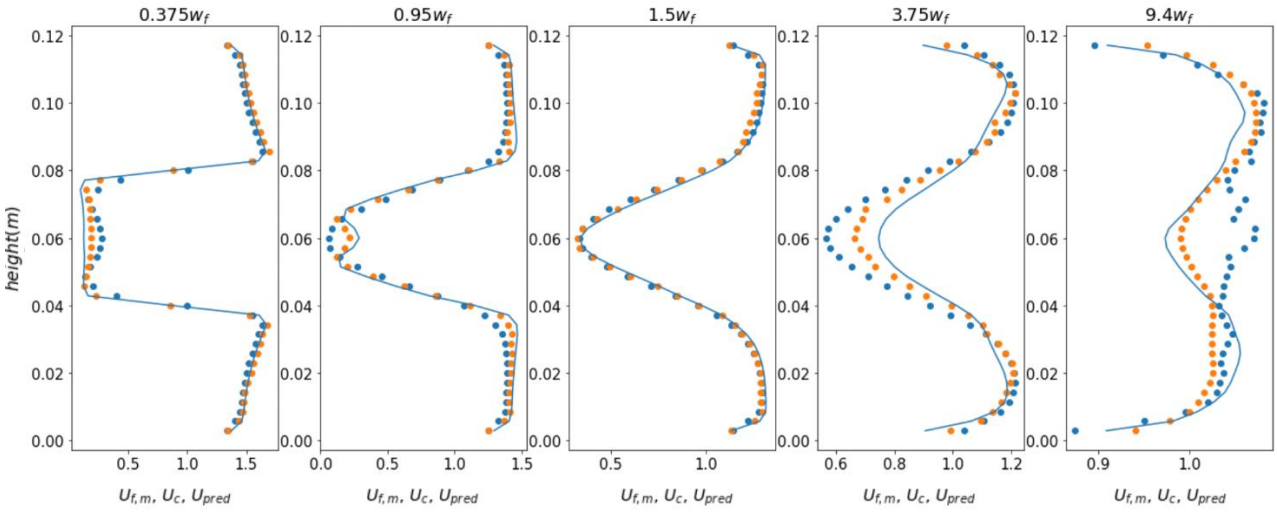


Figure 25 Profiles for the time-averaged velocity magnitude at locations 0.375 w_f, 0.95 w_f, 1.5 w_f, 3.75 w_f and 9.4 w_f (top to bottom) for bullet-shaped bluff body. Fine mesh (blue solid line), coarse mesh (blue circle) and MRF predictions (orange circle)

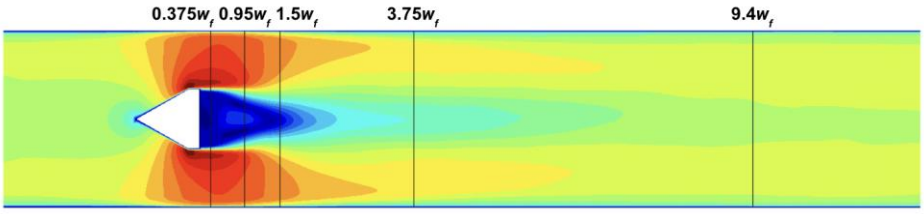


Figure 26 Cross-sectional view indicating the locations for the extracted profiles at locations 0.375 w_f, 0.95 w_f, 1.5 w_f, 3.75 w_f and 9.4 w_f for bullet-shaped bluff body.

The validation serves as a first step to demonstrate the feasibility of this approach to simulate turbulent flows with varying geometries. There is further scope to test this approach for configurations with larger geometrical deviations and develop methodologies to improve prediction accuracy. While the geometrical changes considered here may be small, these changes are crucial in practical design applications due to constraints in the design space. Such design alteration can lead to localised impact in the flow field, which in turn, could significantly impact the overall performance of a system. This is the reason why the current work includes such geometrical changes.

In the present investigation, it was demonstrated that the extensions proposed to the CFD-ML framework yielded further savings in computational runtime for training and inference, while maintaining prediction accuracy. The enhancements also enabled the generalisation of the framework to predict multiple variables within a single training instance and predict flow fields in geometries different from those in the training datasets. This extension addresses the limitations

and enhances the utility of the CFD-ML framework, particularly for industrial-based CFD applications. Often, when aiming to improve system performance, changes in the flow field due to modifications of the design space are as crucial as flow field changes due to operating conditions. Additionally, the extension proposed adds further value to the ability of the framework to predict and compare multiple variables of interest when optimising design performance.

For the current work, a significant reduction in computational time is observed when comparing the coarse grid simulation with the fine grid simulation. Here, a total of 200 computational cores is used. The fine grid simulation (3.5 M cells) takes approximately 16 hours to complete, whereas the combined time for the coarse grid simulation (0.4 M cells) and MRF is around 2.8 hours. This results in a computational savings of over 80%. Consider a scenario where six different runs with various operating conditions are required. Performing all simulations with the fine grid takes a total time of 96 hours. In contrast, the proposed approach leverages MRF to correct variables, allows for three fine grid simulations (for training) supplemented by three coarse grid simulations, reducing the total computational time to 57 hours. As demonstrated here, the approach is particularly advantageous for sensitivity analysis where simulations of flows with similar physics are required across varying conditions or changes in geometry. High resolution simulations for a case study can be supplemented by coarse grid simulations for other cases of interest. Moreover, the availability of more data enhances the overall predictive capability of the ML model to predict the complex flow.

In terms of future work, there are several areas where the proposed framework can be extended. Firstly, we have demonstrated here that the approach can be used to predict the current configuration with different bluff body shapes. It is useful to assess the validity of the approach in scenarios involving more elaborate and complex geometries. Such an exercise would demonstrate the approach to generalise and ensure the robustness of the approach across a broader spectrum of configurations. Secondly, the present study considers an incompressible and isothermal flow. The similarities of the tested flow physics provide confidence in the ML predictions. The applicability of this approach could be gradually extended and validated in more diverse and complex real-world flows, including those involving phase changes or chemical reactions. Expanding the scope of validation to such scenarios would build confidence in applying this method to a wider range of practical applications.

10. Conclusions

In the present paper, three extensions to the CFD-ML framework were considered. The extensions enabled computationally more efficient prediction of the turbulent flow field in the present bluff-body configuration, but without compromising on the accuracy of the predictions. The extensions included (i) adoption of the MRF; (ii) identification and reduction of IFs required for training and prediction via VIMP; and (iii) prediction of flow field with different bluff body shapes.

The MRF was introduced to replace the RF ML model originally used for prediction. The approach allows the prediction of multiple variables in a single training instance and accounts for the dependencies between variables. In the current study, the MRF was used to predict all three velocity components. Its performance was compared against the RF. For both RF and MRF, a notable reduction in error was achieved for velocity components U_x and U_y. However, the

improvements for U_z were minimal. The reason was that U_z was primarily dictated by the small-scale vortices and did not exhibit any coherent pattern. The velocity magnitude U_{mag} was then reconstructed from the velocity components. For all three velocity components and the velocity magnitude, MRF achieved a lower prediction error compared to the RF, indicating its superior performance. The improvement for MRF was achieved across all scenarios 1-4 with different operating conditions and mesh resolution. More importantly, MRF was computationally cheaper than RF. MRF enabled the prediction of all three velocity components with a runtime that was around 30% of RF's runtime. The flexibility of the MRF model allowed for a straightforward extension to predict additional variables, further enhancing the utility and versatility of the CFD-ML framework.

Training the ML model using a reduced set of IFs improves interpretability, reduces runtimes, and prevents overfitting. As such, the VIMP of the MRF model was computed and the key IFs with high importance were identified. A total of 30 key IFs were identified, compared to the 55 used in the original framework. The key IFs effectively captured the flow variations and turbulence in critical flow regions, particularly in the boundary layer, shear layer and recirculation zone. Additional non-dimensional IFs also accounted for the pressure variations in this configuration. However, IFs such as the first and second derivatives of velocity in the z-direction, only captured the incoherent small-scale features and have low importance via the VIMP. To further understand the influence of IFs on the flow field, the histograms of the IFs with low and high importance in different regions of the flow were examined. High-importance IFs exhibited skewness in the histogram, particularly in the separation region, shear layer and recirculation zone. By contrast, low-importance IFs displayed a distribution concentrating at zero, suggesting that only random noise was captured. A 33% reduction in global error relative to the coarse grid was achieved in the MRF prediction. The error obtained using a full set and a reduced set of IFs determined from VIMP was comparable, demonstrating that the accuracy of the MRF prediction was maintained even when using a reduced set of IFs. Additionally, the runtime was halved. This optimisation reduced the computational overhead while preserving the effectiveness of the model in capturing the underlying physics.

The framework was expanded to allow the prediction of configurations featuring different bluff body geometries. In the current work, the MRF model was trained using datasets from three distinct bluff body shapes - parabolic, triangle, and square, respectively. Differences in the flow detachment point led to distinct flow features and recirculation structures for each configuration. The surrogate model trained to learn the physics associated with the changes in geometry was applied to predict the flow field of a bullet bluff body configuration. Both MRF and RF predictions yielded an improvement relative to the coarse grid results, with MRF delivering a lower reduction in global error by 30% compared to RF by 25%. No detriment was also seen when MRF with a reduced set of IFs was employed for this configuration. A quantitative comparison of the time-averaged velocity magnitude profiles at different axial locations confirmed that good agreement was obtained for the predictions over the coarse grid, especially for profiles near the bluff body.

The predictive capability of the CFD-ML framework for the proposed extensions was demonstrated in the current work. These additional capabilities enhanced the existing framework

to predict a new, unseen case given IFs from the coarse grid simulations. These extensions not only maintained the accuracy of the predictions, but also managed to extend the robustness and flexibility of the CFD-ML framework while offering substantial savings in computational runtime.

11. Declarations

Funding
The author(s) received no financial support for the research, authorship, and/or publication of this article.

Conflict of Interest
The author(s) have no competing interest to declare that are relevant to the content of the article**.**

Ethical Approval
Not applicable. The article does not contain any studies on human and animal subjects.

Informed Consent
Not applicable.

12. Author Contributions

CYL – conceptualisation, methodology, calculation, formal analysis, interpretation of results, and writing of manuscript.
VAH-T – feedback on theory and formulation, interpretation of results, review and editing of manuscript.
SC – feedback on theory and formulation, interpretation of results, review and editing of manuscript.

13. References

[1] Moin, P., Mahesh, K.: Direct Numerical Simulation: A Tool in turbulence research, Annu. Rev. Fluid Mech. 30, 539–578 (1998)
[2] Lesieur, M., Métais, O.: Introduction to direct numerical simulations of turbulent flows, Annu. Rev. Fluid Mech. 28, 45–82 (1996).
[3] Ferziger, J. H. Perić, M.: Introduction to large eddy simulation of turbulent flows, Lect. Notes Comput. Sci. Eng. 2, 1–51 (1999).
[4] Sagaut, P.: Large eddy simulation for incompressible flows, Springer, Berlin, Heidelberg (2006)
[5] Craft, T., Launder, B., and Suga, K.: Development and application of a cubic eddy-viscosity model of turbulence, Int. J. Heat Fluid Flow 17, 108–115 (1996)
[6] Pope, S.B.: Turbulent flows, Cambridge University Press, (2000)
[7] Gourdain, N., Sicot, F., Duchaine F., Gicquel L.: Large eddy simulation of flows in industrial compressors: a path from 2015 to 2035, Phil. Trans. R. Soc. A372: 20130323 (2014)
[8] Pitsch, H.: Large-eddy simulation of turbulent combustion, Annu. Rev. Fluid Mech. 38, 453–482 (2006)

[9] Wu, Y. T., Porté-Agel, F.: Large-eddy simulation of wind-turbine wakes: evaluation of turbine parametrisations, Boud.-layer Meteorol. 138, 345–366 (2011)
[10] Hunt, J. C., Savill, A. M.: Chapter 8: Guidelines and criteria for the use of turbulence models in complex flows, in Predictions of Turbulent Flows, in J.C. Vassilicos (eds) (2005)
[11] Tucker, P. G.: Trends in turbomachinery turbulence treatments, Prog. Aerosp. Sci.63 1–32 (2013)
[12] Klein, M.: An attempt to assess the quality of large eddy simulations in the context of implicit filtering, Flow Turb. Combust. 75, 131–147 (2005)
[13] Bose, S.T., Moin, P., You, D.: Grid-independent large-eddy simulation using explicit filtering, Phys. Fluids 22, 1–6 (2010)
[14] Milano, M., Koumoutsakos P.: Neural network modeling for near wall turbulent flow, J. Comput. Phys., 182, 1–26 (2002)
[15] Tracey, B., Duraisamy, K., Alonso, J.: A machine learning strategy to assist turbulence model development, in: 53rd AIAA Aerospace Sciences Meeting , Kissimmee, Florida, AIAA 2015-1287 (2013)
[16] Zhang, Z. J., Duraisamy, K.: Machine learning methods for data-driven turbulence modeling. In: 22nd AIAA Computational Fluid Dynamics Conference, Dallas, TX, AIAA 2015-2460 (2015)
[17] Ling, J., Templeton, J.: Evaluation of machine learning algorithms for prediction of regions of high Reynolds averaged Navier Stokes uncertainty, Phys. Fluids 27, 1–8 (2015)
[18] Ling J., Jones, R., Templeton, J.: Machine learning strategies for systems with invariance properties, J. Comput. Phys. 318, 22–35 (2016)
[19] Ling, J., Kurzawski, A., Templeton, J.: Reynolds averaged turbulence modelling using deep neural networks with embedded invariance, J. Fluid Mech. 807, 155–166 (2016)
[20] Parish E. J., Duraisamy K.: A paradigm for data-driven predictive modeling using field inversion and machine learning, J. Comput. Phys. 305, 758–74 (2016)
[21] Wang J. X., Wu J. L., Xiao H.: Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data, Phys. Rev. Fluids 2, 034603 (2017)
[22] Wu, J-L., Xiao, H., Paterson, E.: Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework, Phys. Rev. Fluids 3, 1–42 (2018)
[23] Kaandorp, M. L., Dwight, R. P.: Data-driven modelling of the Reynolds stress tensor using random forest with invariance, Comp. Fluids 202, 1–16 (2020)
[24] Sarghini, F., De Felice, G., Santini, S.: Neural networks based subgrid scale modeling in large eddy simulations, Comp. Fluids 32, 97–108 (2003)
[25] Bardina, J., Ferziger, J. H., Reynolds, W. C.: Improved subgrid-scale models for large-eddy simulation, AIAA Paper No. 80–1357 (1980)
[26] Beck, A. D., Flad, D. G., Munz C.-D.: Deep neural networks for data-driven LES closure models,
J. Comput. Phys. 398, 1-23 (2019)
[27] Maulik, R., San, O., Rasheed, A., Vedula, P.: Subgrid modelling for two-dimensional turbulence using neural networks, J. Fluid Mech. 858, 122–144 (2019)
[28] Kraichnan, R. H.: Inertial ranges in two-dimensional turbulence, Phys. Fluids 10, 1417–1423 (1967)
[29] Antoine, D., Aufort, J., Balarac, G., and Métais, O.: Subgrid-scale scalar flux modelling based on optimal estimation theory and machine-learning procedures, J. Turb. 20, 229-262 (2019)

[30] Cremades, A., Hoyas, S., Quintero, P., Lellep, M., Linkmann, M., Vinuesa, R.: Explaining wall-bounded turbulence through deep learning, arXiv:2302.01250 (2023)
[31] Balasubramanian, A. G., Gastonia, L., Schlatter, P., Azizpour, H., Vinuesa, R.: Predicting the wall-shear stress and wall pressure through convolutional neural networks, Int. J. Heat Fluid Flow 103, 109200 (2023)
[32] Lee, C. Y., Cant, S.: A grid-induced and physics-informed machine learning CFD framework for turbulent flows, Flow Turb. Combust. 112, 407–442 (2024)
[33] Sjunnesson, A., Olovsson, S., Sj¨oblom, S.: Validation rig - A tool for flame studies. In: ISABE Conference, Nottingham, UK (1991)
[34] Sjunnesson, A., Olovsson, S., Max, E.: Measurements of velocities and turbulence in a bluff body stabilized flame. In: Fourth International Conference on Laser Anemometry - Advances and Application, ASME Cleveland, US (1991)
[35] Wu, J., Chen, X-Y., Zhang, H., Xiong, L-D., Lei, H., Deng, S-H.: Hyperparameter optimization for machine learning models based on bayesian optimization, J. Elec. Sci. Tech. 17, 26-40 (2019)
[36] Probst, P., Wright, M. N., Boulesteix, A-L.: Hyperparameters and tuning strategies for random forest, WIREs Data Mining Know Discov. (2019)
[37] Volpiani, P. S.: Are random forests better suited than neural networks to augment RANS turbulence models? Int. J. Heat Fluid Flow 107, 109348 (2024)
[38] Han, T., Jiang, D., Zhao, Q., Wang, L., Yin, K.: Comparison of random forest, artificial neural networks and support vector machine for intelligent diagnosis of rotating machinery, Trans. Ins. Measurement Control 40, 2681-2693 (2018)
[39] Grinsztajn, L., Oyallon, E., Varoquaux, G.: Why do tree-based models still outperform deep learning on typical tabular data? Advances in neural information processing systems, 35, 1-14 (2022)

[40] Evans, J. S., Murphy, M. A., Holden, Z. A., Cushman, S. A.: Modeling species distribution and change using Random Forest. In: Drew, C., Wiersma, Y., Huettmann, F. (eds) Predictive Species and Habitat Modeling in Landscape Ecology. Springer, New York, NY.
[41] Ahmad, M. W., Mourshed, M., Rezgui Y.: Trees vs neurons: comparison between random forest and ANN for high-resolution prediction of building energy consumption, Ener. Buildings 147, 77-89 (2017)
[42] Kiener, A., Langer, S., Bekemeyer, P.: Data-driven correction of coarse grid CFD simulations. Comp. Fluids 264, 1-14 (2023)
[43] Weller, H.G., Tabor, G.R., Jasak, H., Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques. J. Comput. Phys. 12, 620–631 (1998)
[44] Smagorinsky, J.: General circulation experiments with the primitive equations: I. The basic experiment, Mon. Weather Rev. 91, 99–164 (1963)
[45] Van Driest, E. R.: On turbulent flow near a wall, J. Aeronau. Sci., *23*, 1007-1011 (1956)
[46] Issa, R.I.: Solution of the implicitly discretised fluid flow equations by operator-splitting. J. Comput. Phys. 62, 40–65 (1985)
[47] Jarrin, N., Benhamadouche, S., Laurence, D., Prosser, R.: A synthetic-eddy-method for generating Inflow conditions for large-eddy simulations. Int. J. Heat Fluid Flow 27, 585–593 (2006)
[48] Pope, S. B.: Ten questions concerning the large-eddy simulation of turbulent flows, New J. Phys. 6 (2004).
[49] Segal, M., Xiao, Y.: Multivariate random forests, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 1, 80–87 (2011)

[50] Ishwaran H., Kogalur U.B.: Fast unified random forests for survival, regression, and Classification (RF-SRC), R package version 3.2.0 (2023)
[51] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python, J. Mach. Learn. Research 12, 2825–2830 (2011)
[52] Breiman, L.: Random forests. Mach. Learn. 45, 5–32 (2001)
[53] Ishwaran H., Lu M., Kogalur U. B.: randomForestSRC: Variable importance (VIMP) with subsampling inference vignette (2021) http://randomforestsrc.org/articles/vimp.html.
[54] Johnson, R.: The Handbook of Fluid Dynamics, CRC Press, (1998)
[55] Spencer, A., Rivlin, R.: Isotropic integrity bases for vectors and second-order tensors, Arch. Ration. Mech. Anal. 9, 45–63 (1962)
[56] Zheng, Q.: Theory of Representations for Tensor Functions — A unified invariant approach to constitutive equations, Appl. Mech. Rev. 47, 545–587 (1994)

14. Appendix

14.1 Hyperparameter Study

A hyperparameter study was conducted for the MRF. For the hyperparameter 'features chosen at each split', three common options are used: (i) total number of IFs/3, which is the default value used; (ii) √(total number of IFs); and (iii) total number of IFs. We performed a sensitivity study to investigate their impact on the results. The global error and RMSE for these options are summarised in Table 12 for different scenarios. The results do not change much with the different hyperparameter values, although a slightly larger impact is observed for the extrapolation scenario.

Interpolation scenario

Features chosen at each split	Global error	RMSE
Total number of IFs/3	2.894	0.0413
√ (total number of IFs)	2.901	0.0416
Total number of IFs	2.981	0.0425

Extrapolation scenario

Features chosen at each split	Global error	RMSE
Total number of IFs/3	3.426	0.0493
√ (total number of IFs)	3.640	0.0519
Total number of IFs	3.856	0.0545

Geometry changes scenario

Features chosen at each split	Global error	RMSE
Total number of IFs/3	2.839	0.0425
√ (total number of IFs)	2.855	0.0431
Total number of IFs	2.925	0.0440

Table 12 Global error and RMSE of the hyperparameter 'features chosen at each split' for the interpolation scenario, extrapolation scenario and geometrical changes scenario.

For the 'total number of trees', a higher value (up to a certain point) will decrease the generalisation error by reducing the overfitting. After a certain value, a further increase in the 'total number of trees' will not improve further the results but instead only lead to higher computational costs. An example of the global error and RMSE with different 'total number of trees' is shown in Table 13 for the interpolation scenario. A further increase in the number of trees does not provide further improvement in the results. The same trend is seen for all the other scenarios. The hyperparameter studies confirm that overfitting is not present in the study.

Interpolation scenario

Number of trees	Global error	RMSE
400	2.913	0.0416
500 (default)	2.894	0.0413
550	2.893	0.0413
600	2.891	0.0412

Table 13 Global error and RMSE of the hyperparameter 'total number of trees' for the interpolation scenario

14.2 Comparison of Prediction Method – Reconstructed Velocity Magnitude vs Direct Prediction

Table 14 summarises the coarse grid error E_c and prediction error E_{pred} for each velocity component, U_x, U_y, U_z (rows 1-3) and the velocity magnitude U_{mag} (row 4) for scenario 1. The standard deviations are presented alongside the errors. We compare the coarse grid error (first column) against the errors from the RF predictions (second column), MRF predictions (third column) and direct RF predictions for the velocity magnitude (fourth column). For the RF and MRF predictions (second and third column), the velocity magnitude U_{mag} is reconstructed based on each predicted velocity component and its error is presented in row 4. For the direct RF predictions (fourth column), the velocity magnitude is obtained with a RF model that was trained to predict the magnitude directly. The MRF (third column) delivered a better prediction than RF (second column). The direct RF prediction (fourth column) produces the lowest error for the velocity magnitude. Such an approach is useful when only a single variable needs to be predicted and no additional information on the velocity components is required. Based on the observation, MRF is advantageous when we intend to predict multiple correlated variables, i.e. three velocity components, and not just the velocity magnitude.

	Coarse grid error, E_c and standard deviation, SD_c (%)	**RF prediction error, E_{pred} and standard deviation, SD_{pred} (%)**	**MRF prediction error, E_{pred} and standard deviation, SD_{pred} (%)**	**Direct RF prediction error E_{pred} and standard deviation, SD_{pred} (%)**

U_x	4.273 ± 0.007	3.139 ± 0.0058	2.889 ± 0.0048	-
U_y	1.909 ± 0.0042	$1.266 + 0.0029$	1.252 ± 0.0027	-
U_z	1.667 ± 0.0027	1.466 ± 0.0025	1.474 ± 0.0024	-
U_{mag}	4.282 ± 0.0071	3.142 ± 0.0058	2.894 ± 0.0048	2.675 ± 0.0049

Table 14 Coarse grid error and standard deviation (first column), prediction error and standard deviation for RF (second column), prediction error and standard deviation for MRF (third column) and prediction error and standard deviation for direct RF (fourth column) in scenario 1. Normalised velocity components (rows1-3) and normalised velocity magnitude (row 4)