

Multifidelity Aerodynamic Flow Field Prediction Using Random Forest-Based Machine Learning

Jethro Nagawkar and Leifur Leifsson*

Department of Aerospace Engineering, Iowa State University

Ames, IA, 50011, USA

Abstract

In this paper, a novel random forest (RF)-based multifidelity machine learning (ML) algorithm to predict the high-fidelity Reynolds-averaged Navier-Stokes (RANS) flow field is proposed. The RF ML algorithm is used to increase the fidelity of a low-fidelity potential flow field. Three cases are studied, the first two consist of a flow past a backward-facing step, and the third, a subsonic flow around an airfoil. In the first case, the data is generated using ten different inlet velocities, in the second using six different step heights, and in the third using 20 different airfoil shapes parameterized using B-spline curves. Input parameters to RF are case dependent. For the first case, the x and y cell-center locations and the corresponding x and y potential flow velocities, along with the specified inlet velocity, are used. For the second, the cell-center values are nondimensionalized using the step height, and the step height is used in place of the inlet velocity. The remaining two input features used are the same as in the previous case. For the third case, the potential flow stream function and velocity potential along with the B-spline control point values are used as input variables. The outputs of the RF algorithm are the same for all the cases and include the RANS velocities, pressures, and turbulent viscosities. The results in this study are compared to those generated using the tensorFlowFoam (TFF) and from directly solving the RANS equations. To quantify the errors, the absolute error and relative L_2 norm error metrics are used. The results show that for the first two cases, RF consistently has two to 30 times lower relative L_2 norm compared to TFF, with the only exception being the turbulent viscosities for the second case. For the third case, RF is better at predicting the pressure and skin friction coefficients for the RAE 2822 airfoil compared to the NACA 0012 airfoil. The relative L_2 norm error is 1.67 and 1.19 times lower for the pressure and skin friction coefficients, respectively.

*Corresponding author (leifur@purdue.edu)

Keywords: Flow field prediction, random forest, machine learning, multifidelity modeling, surrogate modeling.

Nomenclature

α	Angle of attack, (deg)
$\hat{\mathbf{f}}_{\mathbf{HF}}$	RF-based flow field prediction
ν	Kinematic viscosity, (m ² /s)
ν_t	Turbulent eddy viscosity, (m ² /s)
ϕ	Velocity potential
ψ	Stream function
ρ_∞	Free-stream density, (kg/m ³)
τ_w	Wall shear stress, (Pa)
$\mathbf{f}_{\mathbf{HF}}$	High-fidelity flow field variables
$\mathbf{f}_{\mathbf{LF}}$	Low-fidelity flow field variables
$\mathbf{u}$	Velocity vector, (m/s)
$\mathbf{X}$	Sample points of the design variables
$\mathbf{x}$	Design variables used to make RF-based flow field prediction
c	Chord length, (-)
C_d	Drag coefficient, (-) $= \frac{D}{\frac{1}{2}\rho_\infty U_\infty^2 c}$
C_f	Skin friction coefficient, (-) $= \frac{\tau_w}{\frac{1}{2}\rho_\infty U_\infty^2}$
C_l	Lift coefficient, (-) $= \frac{L}{\frac{1}{2}\rho_\infty U_\infty^2 c}$

C_m	Pitching moment coefficient, (-) $= \frac{M}{\frac{1}{2}\rho_\infty U_\infty^2 c^2}$
C_p	Pressure coefficient, (-) $= \frac{p-p_\infty}{\frac{1}{2}\rho_\infty U_\infty^2}$
D	Drag force, (N)
d	Number of design variables
L	Lift force, (N)
M	Pitching moment, (Nm)
m	Number of data samples
M_∞	Free-stream Mach number
m_{tr}	Number of training design samples
m_t	Number of testing design samples
N	Number of samples
n	Number of subsets
n_h	Number of high-fidelity flow field parameters
n_l	Number of low-fidelity flow field parameters
p	Static pressure, (Pa)
p_r	RANS pressure in the grid cell-center, (Pa)
p_∞	Free-stream pressure, (Pa)
Re_∞	Reynolds number based on chord length, $= \frac{U_\infty c}{\nu}$
Re_H	Reynolds number based on step height, $= \frac{U_{inlet} H}{\nu}$
u_p	Potential flow x coordinate velocity in the grid cell-center, (m/s)

u_r	RANS x coordinate velocity in the grid cell-center, (m/s)
U_∞	Free-stream velocity, (m/s)
U_{inlet}	Domain inlet velocity, (m/s)
v_p	Potential flow y coordinate velocity in the grid cell-center, (m/s)
v_r	RANS y coordinate velocity in the grid cell-center, (m/s)
x_c	x coordinate of the grid cell-center, (m)
x_{RAL}	Boundary layer reattachment length, (m)
y^+	Non-dimensionalized first layer cell thickness
y_c	y coordinate of the grid cell-center, (m)
$\mathbf{x}_0$	B-spline control points for NACA 0012
H	Step height, (m)
h	Baseline step height, (m)
x	x coordinate of the domain, (m)
y	y coordinate of the domain, (m)
d.c.	Drag counts, $\Delta C_d = 0.0001$
l.c.	Lift counts, $\Delta C_l = 0.01$

1 Introduction

Increasing computational power and decreasing computational cost have led to the widespread development and use of three very important areas of aerospace engineering, namely, computational fluid dynamics (CFD), machine learning (ML), and aerodynamic design optimization (ADO). While these three fields were initially developed separately, there has been an increasing effort to couple the three systems together to find optimum designs [1–5], perform uncertainty analysis [6–9], and solve inverse problems [10–12].

Traditional ADO methods rely heavily on expensive high-fidelity simulations to calculate both the cost function and constraint values [13–16]. Furthermore, these methods require multiple and

repetitive high-fidelity model evaluations during the iterative design process. When combined with a large number of design variables, these methods result in problems that can be difficult to solve in a reasonable time period.

Several surrogate modeling techniques have been introduced to overcome these challenges [17–22]. These surrogate modeling techniques can be broadly classified as either data-fit methods [23, 24] or multifidelity methods [25]. In data-fit methods, a response surface is fitted through the evaluated single-fidelity model responses at sampled points in the design space. Kriging [26–28], polynomial chaos expansions [29], and support vector regression [30] are examples of data-fit methods. Multifidelity methods [25], on the other hand, use information from multiple levels of fidelities to make accurate high-fidelity predictions. Low-fidelity data samples can be used to provide information on the trends to a relatively small number of high-fidelity data samples, increasing the overall predictive capabilities of these surrogate models. Cokriging [31, 32] and manifold mapping [33] are examples of multifidelity surrogate modeling methods.

The use of ML to augment Reynolds-averaged Navier-Stokes (RANS) CFD models, increasing their fidelity, have recently become popular [34–37]. The simplest method involves tuning the turbulence parameters by minimizing a cost function, such as the mean squared error (MSE) of velocity or temperature at a selected location [38–42]. Either a Kriging [38] or a neural network (NN) [39] surrogate model was used to find the relationship between the turbulence parameters and the cost function. Variables selected in the cost function showed prediction improvements, while other variables showed a decrease in accuracy [38, 39]. In another method, Singh et al. [34, 35] developed a data-driven technique referred to as field inversion ML to construct a deficiency (β) term from high-fidelity large eddy simulation (LES) data. This term was then multiplied to either the production or destruction terms in one of the two additional transport equations of a two-equation RANS model. Either NN [43] or Kriging [26] were used to construct a functional form of β from a set of nondimensional flow features. This approach enhanced the fidelity of the two-equation RANS model [34, 35]. A physics-informed ML framework was developed by Wang et al. [36] and Wu et al. [37]. Several mean flow features were used as inputs into a random forest (RF) [44, 45] ML algorithm with outputs being the discrepancy of the Reynolds stresses between the RANS model and direct numerical simulations (DNS). This discrepancy term was then embedded into the RANS model and the resulting RANS model showed significant improvements in field predictions. While the methods are promising, they involve solving the RANS equations every time a prediction is needed to be made. In addition,

training these ML models requires data to be generated from either LES or DNS, both of which are computationally impractical to solve for many problems.

Instead of solving the governing equations each time, some researchers [3, 4, 46] have attempted to directly predict the quantities of interest. Zhang et al. [46] tried to predict pressure coefficients on airfoils of different shapes at multiple Mach numbers, Reynolds numbers, and angle of attack, using both NN [47] and convolutional NN [43]. Li et al. [4] used gradient-enhanced Kriging with partial least squares [20] to predict the drag, lift, and pitching moment coefficients of different airfoils at varying flow conditions. Bouhlel et al. [3] performed a similar study, but used gradient-enhanced NN [48] to predict the drag, lift, and pitching moment coefficient for an airfoil. While good prediction capabilities of these algorithms were shown, the major drawback of these methods is that they require a large number of sample points to train the algorithms, typically on the order of 10^3 to 10^5 . The data generated in these studies [3, 4] was acquired by running a large number of CFD simulations for multiple different airfoil shapes and flow conditions to calculate the force coefficients. The number of CFD simulations performed, therefore, corresponds to the number of sample points required to train the ML algorithms, thereby making it challenging to be used for various optimum design and uncertainty quantification problems.

Maulik et al. [49] developed a framework, called tensorFlowFoam¹ (TFF), to predict the turbulent viscosity in fluid flow obtained from data generated using several different RANS turbulence models. Their work considers the flow past a backward-facing step for different inlet conditions and geometries. Specifically, in the first case, they train the NN using data generated with ten different inlet velocities, and in the second case, the training data is generated using six different step heights. The NN is then tested on two unseen inlet velocities and step heights, respectively. To predict the flow field for the unseen cases, they solve the continuity and momentum equations in RANS and use the turbulent viscosity predicted by the NN instead of solving additional closure equations. This reduced the simulation cost by a factor of five [49]. The major benefit of using such a method is the total number of CFD simulations required to generate all the data in order to train the NN is relatively small. This is because Maulik et al. [49] used data from each cell-center from each CFD simulation to train the NN, and each CFD simulation contained approximately 10^5 grid cells.

In this work, a novel multifidelity ML algorithm using RF [44, 45] is proposed and demonstrated on three different cases. The framework is motivated by the TFF framework developed

¹<https://github.com/argonne-lcf/TensorFlowFoam>

by Maulik et al. [49, 50], and uses data from both a high-fidelity RANS model and a low-fidelity potential flow model. The proposed framework predicts the velocity and pressure fields in addition to the turbulent viscosity field predicted using TFF. The proposed method, therefore, eliminates the need for solving the continuity and momentum equations required by TFF, further reducing the amount of time required to make a high-fidelity flow prediction. Furthermore, the proposed method uses the RF [45] in place of NN [43]. The first two cases in this paper are the same as those used by Maulik et al. [49] and the proposed RF framework is compared to TFF as well as to the RANS CFD simulations. The third case consists of flow over an airfoil at a Mach number of 0.3 and an angle of attack of ten degrees. In this case, results from RF are only compared to those from RANS simulations.

The next section describes the proposed method to predict the flow field using the multifidelity RF ML algorithm. In the following section, the results of applying the proposed method to three cases are described and compared against TFF prediction and direct RANS solutions. Lastly, the conclusion and future work are presented.

2 Methods

This section outlines the methods used to construct the multifidelity RF-based flow field prediction framework. The section begins with an outline of the workflow of the framework, followed by descriptions of the sampling plan, the high- and low-fidelity flow field modeling, the RF algorithm, the validation metric used, and, lastly, the RF-based flow field prediction.

2.1 Workflow

Figure 1 shows a flowchart of the proposed multifidelity RF-based flow field prediction framework. This framework consists of two parts, the offline part (Fig. 1(a)) and the online part (Fig. 1(b)). The offline part consists of training and validating the RF, while the online part is used to make high-fidelity flow field prediction using the trained RF, given the low-fidelity flow field and the design parameters. The process for the offline part starts by sampling the design space $\mathbf{X} \in \mathbb{R}^{m_{tr} \times d}$ first. Here, m_{tr} denotes the total number of training design samples, while d is the number of design parameters. The low- ($\mathbf{f}_{LF} \in \mathbb{R}^{m \times n_l}$) and high-fidelity models ($\mathbf{f}_{HF} \in \mathbb{R}^{m \times n_h}$) are run for the sampled designs to generate the data required to train and validate the RF. n_l and n_h are the number of low- and high-fidelity flow field parameters, respectively, while m

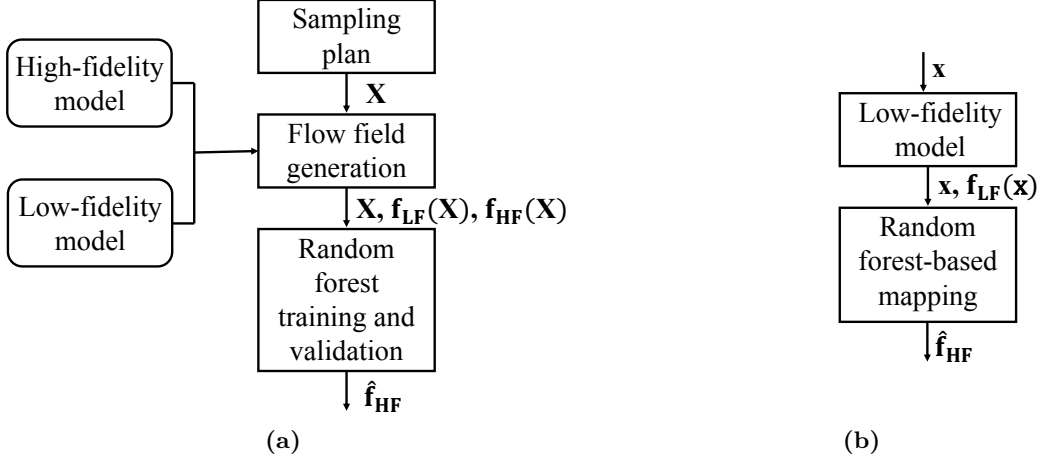


Figure 1: A flowchart of the multifidelity random forest-based flow field prediction framework: (a) training and validating the random forest, and (b) random forest-based flow field prediction.

denotes the total number of data samples. In order to tune the hyperparameters of the RF, 90% of the data is used to train the RF, while 10% is used to validate them. The accuracy of the RF algorithm is measured using the MSE. Once all the hyperparameters are tuned, the RF is retrained with the combined training and validation data and is ready to be used in the online part. In the online part, the low-fidelity model is run on previous unsampled design parameters $\mathbf{x} \in \mathbb{R}^{m_t \times d}$, where m_t is the number of testing design samples. The data from this low-fidelity model is then used as inputs to the trained RF to make high-fidelity RF-based flow field predictions $\hat{\mathbf{f}}_{\text{HF}} \in \mathbb{R}^{m \times n_h}$.

2.2 Sampling plan

Sampling is the process of selecting discrete samples in the design space [51]. It is an iteration-based process in which the design parameters are randomly drawn from probability distributions assigned to the parameters. For the first two demonstration cases in this work, the full factorial sampling (FFS) [51] plan is used to generate the design parameters, while the Latin Hypercube sampling (LHS) [52] plan is used for the same in the third case. For these sampled design parameters, both high-fidelity RANS CFD simulations and low-fidelity potential flow simulations are performed to generate the required flow field to train, validate, and make predictions using the RF framework.

2.3 Flow field modeling

This section details the high- and low-fidelity models use to model the flow field for the demonstration cases used in this study.

2.3.1 Low-fidelity flow field modeling

In this study, the low-fidelity model used is the potential flow model [53]. Potential flow equations assume that the flow is both inviscid and irrotational [54]. For the flow to be irrotational, the vorticity needs to be zero everywhere and is given by the curl of the velocity [54]

$$\nabla \times \mathbf{u} = 0, \quad (1)$$

where $\mathbf{u}$ is velocity vector of the flow. Considering the vector identity [54]

$$\nabla \times (\nabla \phi) = 0, \quad (2)$$

where ϕ is a scalar function, combining (1) and (2) leads to

$$\mathbf{u} = \nabla \phi, \quad (3)$$

where ϕ now is defined as the velocity potential [54]. Since the flow in the demonstration cases is incompressible, the conservation of mass is given by divergence of the velocity [53]

$$\nabla \cdot \mathbf{u} = 0. \quad (4)$$

Using (3) in (4) gives

$$\nabla \cdot (\nabla \phi) = 0 \quad (5)$$

or

$$\Delta \phi = 0. \quad (6)$$

In any two dimensional incompressible flow, a stream function ψ [54] can be defined such that it satisfies [54]

$$\Delta \psi = 0. \quad (7)$$

Equations (6) and (7) imply that any two dimensional potential flow problem has a velocity potential and a stream function that satisfy Laplace's equations [54].

2.3.2 High-fidelity flow field modeling

The Spalart-Allmaras [55] turbulence model is used in this work. The additional transport equation is introduced as a way to model the Reynolds stresses, which are a byproduct of performing Reynolds averaging on the Navier-Stokes equations, leading to the infamous closure problem in turbulence [56]. The Boussinesq eddy viscosity assumption [57] is used to correlate the Reynolds stresses and the turbulent viscosity, ν_t . A transport equation to model this turbulent viscosity is solved in addition to the Reynolds averaged continuity and momentum equations and is given by [55]

$$\begin{aligned} \frac{\partial \tilde{\nu}}{\partial t} + \mathbf{u} \cdot \nabla \tilde{\nu} = & C_{b1}[1 - f_{t2}]\tilde{S}\tilde{\nu} + \frac{1}{\sigma}\{\nabla \cdot [(\nu + \tilde{\nu})\nabla \tilde{\nu}] + C_{b2}|\nabla \tilde{\nu}|^2\} \\ & - \left[C_{w1}f_w - \frac{C_{b1}}{\kappa^2}f_{t2} \right] \left(\frac{\tilde{\nu}}{d} \right)^2 + f_{t1}\Delta U^2, \end{aligned} \quad (8)$$

where

$$\nu_t = \tilde{\nu}f_{v1}, f_{v1} = \frac{\chi^3}{\chi^3 + C_{v1}^3}, \chi = \frac{\tilde{\nu}}{\nu}, \tilde{S} = S + \frac{\tilde{\nu}}{\kappa^2 d^2}f_{v2}, f_{v2} = 1 - \frac{\chi}{1 + \chi f_{v1}},$$

$$f_w = g \left[\frac{1 + C_{w3}^6}{g^6 + C_{w3}^6} \right]^{1/6}, g = r + C_{w2}(r^6 - r), r = \frac{\tilde{\nu}}{\tilde{S}\kappa^2 d^2}, S = \sqrt{2\Omega_{ij}\Omega_{ij}},$$

$$f_{t1} = C_{t1}g_t \exp\left(-C_{t2}\frac{\omega_t^2}{\Delta U^2}[d^2 + g_t^2 d_t^2]\right),$$

$$f_{t2} = C_{t3}\exp(-C_{t4}\chi^2), \Omega_{ij} = \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} - \frac{\partial u_j}{\partial x_i}\right), C_{w1} = \frac{C_{b1}}{\kappa^2} + \frac{1 + C_{b2}}{\sigma},$$

$$\sigma = 2/3, C_{b1} = 0.1355, C_{b2} = 0.622, \kappa = 0.41, C_{w2} = 0.3,$$

$$C_{w3} = 2, C_{v1} = 7.1, C_{t1} = 1, C_{t2} = 2, C_{t3} = 1.1, C_{t4} = 2,$$

$\tilde{\nu}$ is the turbulent viscosity like variable [55], S and Ω are the strain-rate tensor and the vorticity tensor, respectively, d is the distance from the closest surface, and ΔU^2 is the peak velocity difference [58].

2.4 Random forest

Random forest [44, 45] is an ensemble ML method [59] which can be used for both classification and regression tasks by constructing a multitude of decision trees [60]. Both decision trees [60] and random forests [45] involve stratifying or segmenting the output predictor space into simple

regions. For a new observation, the mean or mode value of training observations to which the new observation belongs is used as its predicted value. A set of splitting rules is used to segment the predictor space and this can be summarized using a tree, hence the name decision trees [61].

Figure 2 shows a schematic of a random forest with two decision trees. Each decision tree consists of a root node, followed by several intermediate nodes or decision nodes, and terminates on the leaf or terminal nodes. The root node of a decision tree represents the entire training dataset and it is used to further divide this dataset into two or more smaller datasets. This is done by iterating over each individual input feature as well as over a splitting criterion such as a mean or mode of the input feature value. The combination which results in the lowest weighted MSE over all the subsets, n , is used as the criterion to split the data at the root node. The MSE is given by

$$\mathcal{L}_{split} = \sum_{k=1}^n \frac{\sum_{i=1}^{N^{(k)}} \sum_{j=1}^{n_h} (\hat{f}_{\text{HF},j}^{(k)} - f_{\text{HF},j}^{(i)})^2}{N^{(k)} n_h}, \quad (9)$$

where n_h is the number of output predicted variables, $N^{(k)}$ is the number of samples in the k^{th} subset created after splitting at the root node, $f_{\text{HF},j}^{(i)}$ represent the high-fidelity CFD field prediction for the j^{th} field variable and the i^{th} sample in the subset, and $\hat{f}_{\text{HF},j}^{(k)}$ is the mean of the CFD field predictions for the j^{th} field variable in the k^{th} subset, and is given by

$$\hat{f}_{\text{HF},j}^{(k)} = \frac{\sum_{i=1}^{N^{(k)}} f_{\text{HF},j}^{(i)}}{N^{(k)}}. \quad (10)$$

The intermediate nodes following the root node perform the same task as the root node, but with data from a given subset. The intermediate node essentially splits the subset from the previous intermediate node or root node into a smaller subset. This process continues with smaller and smaller subsets until the leaf node is reached. Various hyperparameters can be set to decide whether a leaf node is reached. These include the minimum number of samples in a subset required to split it and the maximum tree depth. The leaf node returns the predicted value from the decision tree.

Some of the main advantages of decision trees include interpretability, the ability to handle large amounts of data, and ease of use [61]. This makes it attractive to use for various ML tasks. However, its drawbacks include high sensitivity to the training dataset, and over-fitting [61]. Random forest overcomes these problems by averaging the outputs from multiple decision trees, which is set using the number of trees hyperparameter, and in order to construct each individual

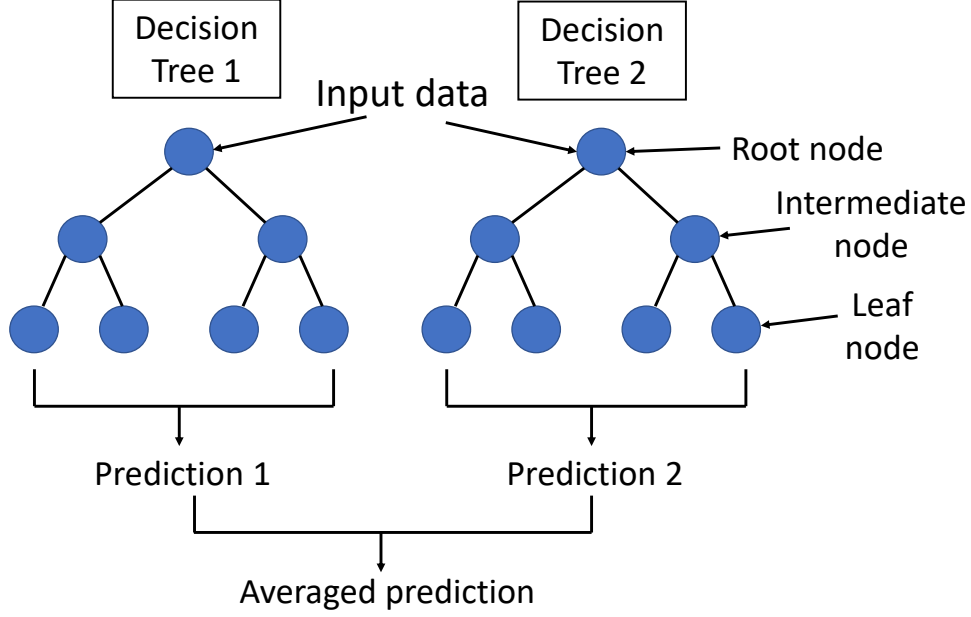


Figure 2: A schematic of a random forest with two decision trees.

decision tree, data from the training dataset is randomly selected with replacement and fed into the decision tree.

2.5 Validation

In this work, 90% of the data is used to train the RF, and 10% is used to validate its accuracy. The training and validation set accuracy is measured using the mean MSE, and is averaged over all the output predictions, and is given by

$$\mathcal{L} = \frac{\sum_{i=1}^N \sum_{j=1}^{n_h} (\hat{f}_{\text{HF},j}^{(i)} - f_{\text{HF},j}^{(i)})^2}{N n_h}, \quad (11)$$

where n_h is the number of high-fidelity flow field variables, m is the total number of data samples, $N \subset m$ is the number samples in either the training or validation data sets, $\hat{f}_{\text{HF},j}^{(i)}$ and $f_{\text{HF},j}^{(i)}$ represent the high-fidelity RF and CFD field prediction, respectively, of the j^{th} field variable and the i^{th} sample.

2.6 Multifidelity random forest-based flow field prediction

Once the RF is trained and validated, it can then be used to increase the fidelity of the low-fidelity potential flow field. To perform a high-fidelity RANS field prediction, the potential flow field values, as well as the design variables, $\mathbf{x}$, are fed into the trained RF and the corresponding RANS field can be reconstructed for the design $\mathbf{x}$. To measure the accuracy of the field predicted by

RF, both the absolute and the relative L_2 norm errors are measured at different locations in the domain. The choice of these locations is case dependant and is discussed in their corresponding sections. The absolute error is given by

$$L_{abs,j} = |\hat{f}_{HF,j} - f_{HF,j}|, \quad (12)$$

where $\hat{f}_{HF,j}$ and $f_{HF,j}$ are the high-fidelity RF and CFD flow field prediction for a given variable of interest j . The relative L_2 norm is given by

$$L_{2,rel,j} = \frac{\|\hat{\mathbf{f}}_{HF,j} - \mathbf{f}_{HF,j}\|_2}{\|\mathbf{f}_{HF,j}\|_2}. \quad (13)$$

The main benefit of using this method is the reduction in the cost associated in various engineering problems, such as aerodynamic shape optimization, uncertainty quantification and propagation, as well as global sensitivity analysis. Such problems typically need multiple and repetitive high-fidelity physics-based simulation evaluations. This methodology replaces these high-fidelity simulations with the RF model. This is done by feeding in the values of the design variables along with the low-fidelity flow field variables to the trained RF model for a new design $\mathbf{x}$, to make high-fidelity flow field predictions. This field can then be used to calculate the quantities of interest for the engineering application. Both the low-fidelity simulation cost and RF prediction time are significantly lower than the high-fidelity simulation cost, thereby significantly reducing the computational time involved in solving such problems.

3 Numerical Examples

This section presents the results of the multifidelity RF-based flow field prediction framework demonstrated on three different cases. The first two cases consider steady flow past a backward-facing step and the third a steady subsonic flow around an airfoil at ten degrees angle of attack. The results for the first two cases are compared to the TFF framework and all the cases are compared to the RANS CFD simulations.

3.1 Case I

In this case, the flow past a backward facing step is simulated. The step height, $H = 1.0h$, where h is the baseline step height of 12.7 mm , is used and the Reynolds number based on this step

height is defined as $Re_H = U_{inlet}H/\nu$, where U_{inlet} is the freestream velocity at the inlet to the domain, H is the step height, and ν is the kinematic viscosity. The Re_H for this case varies between 34,000 and 41,500, based on the selected U_{inlet} values.

3.1.1 Problem definition

The objective of this case is to replace the high-fidelity RANS simulations with the multifidelity RF ML algorithm that predicts the RANS flow field, namely, velocity (u_r and v_r), pressure (p_r) and turbulent viscosity (ν_t), in the domain. The inputs to the random forest are the cell-center locations of the grid elements, given by the horizontal, x_c , and vertical, y_c , positions, the horizontal and vertical velocities at these cell-center locations from the low-fidelity potential flow solver, given by u_p and v_p , respectively, and the inlet velocity inlet boundary condition value, U_{inlet} . The RF for this case is trained with data from ten CFD simulations with U_{inlet} varying from 40 m/s to 49 m/s in increments of 1 m/s , that is using the FFS plan. The RF is then tested on two unseen inlet speeds, U_{inlet} , of 44.2 m/s and 48.5 m/s . The results are finally compared to those from RANS CFD simulations and TFF. Note that the choice of training and testing data is the same as those used by Maulik et al. [49].

3.1.2 CFD setup and validation

In this work, the flow past a backward-facing step is simulated for the first two cases. Figure 3 shows the domain of the geometry used in this study. The step height, h , is fixed at 12.7 mm and the remaining domain is scaled as per the dimensions given in Fig. 3. The inlet velocity to the domain is set to a value of 44.2 m/s . The step height and inlet velocity values are set in

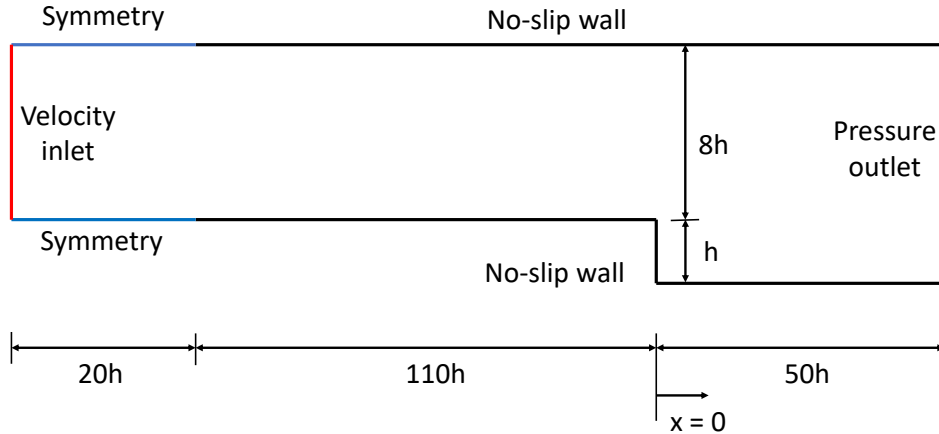


Figure 3: Domain and boundary conditions of the backward facing step.

accordance with experimental data from David and Seegmiller [62], in order to validate the CFD setup.

To generate the mesh, the blockMesh in OpenFOAM version 5.0 [63] is used and is shown in Fig. 4. Near the upper and lower walls, as well as the step, the mesh is refined to ensure that the first cell thickness is less than a y^+ value of one. A grid independent study is then performed and the results are shown in Table 1.

To simulate both the high- and low-fidelity flow, OpenFOAM version 5.0 [63] is used. In particular, the potentialflowFoam solver is used to simulate the low-fidelity potential flow field and the simpleFoam solver to simulate the high-fidelity Spalart-Allmaras [55] RANS turbulence model flow field. The boundary conditions set are shown in Fig. 3. The viscosity (ν) of the fluid used has a value of $1.56 \times 10^{-5} \text{ m}^2/\text{s}$ and the outlet has a zero pressure gradient value. The wall boundary conditions are shown in Fig. 3. The Linear-Upwind Stabilised Transport [63] scheme is used to handle the convective fluxes. The convergence criteria set is pressure and velocity residuals fall to a value below 10^{-6} for the RANS simulations. The simulation time as well as the reattachment length (x_{RAL}) of boundary layer downstream of the step is given in Table 1. Refining the mesh moves the reattachment length from the CFD simulations closer to the experimental value. From Table 1, mesh L1 is used in this study.

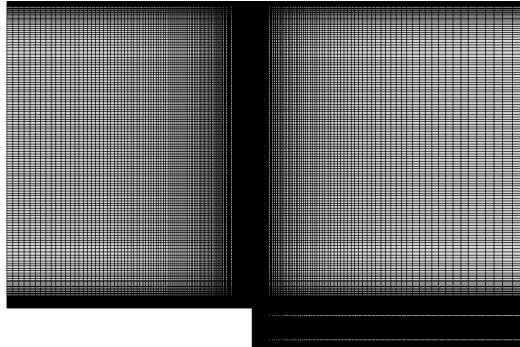


Figure 4: Mesh generated for the backward facing step.

Table 1: Grid convergence study for the backward facing step.

Mesh	No. of cells	x_{RAL}	Simulation time*, s
L3	7,620	5.97h	12.5
L2	30,730	6.03h	49.3
L1	123,000	6.06h	343.4
L0	408,000	6.29h	2,912.5
Exp. [62]	-	6.26h	-

*Computed on a high-performance cluster with 16 processors.

3.1.3 Results

The ratio of the turbulent viscosity to kinematic viscosity, termed viscosity ratio, for TFF, RF, and CFD are shown in Figs. 5(a) and 6(a) for the two selected test cases, respectively, and at different downstream locations. Figures 5(b) and 6(b) show the corresponding absolute errors of TFF and RF with respect to the CFD results. Figures 5 and 6 show that the viscosity ratio trend for both these cases is similar, with RF having a lower absolute error when compared to TFF. At the midsection of this backward-facing step domain, with midsection referring to y/H values between 3 and 5, RF has the lowest absolute error and this error is around three orders of magnitude lower than TFF. In the remaining section of the domain, RF is still better at predicting the viscosity ratios than TFF, but this error is around four orders of magnitude higher than the midsection. Table 2 shows the relative L_2 norm of the viscosity ratio at various downstream locations for this case. RF has a lower relative L_2 norm compared to TFF. At x/H

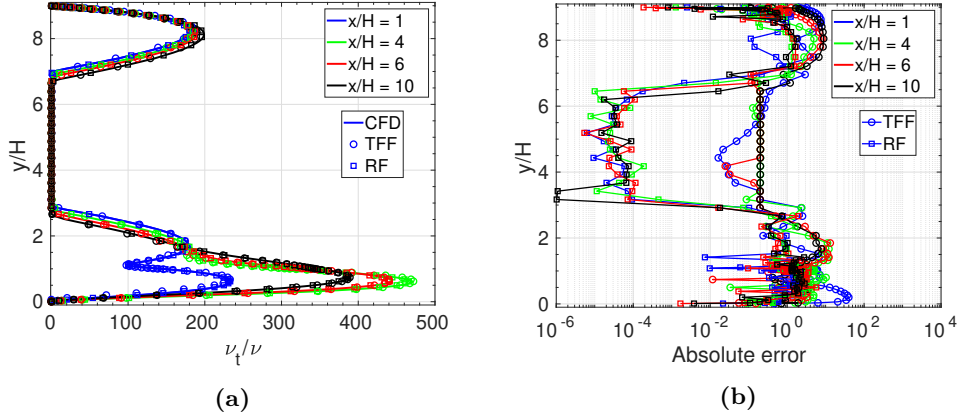


Figure 5: Case I viscosity results at different x/H locations for inlet velocity of 44.2 m/s : (a) viscosity ratio values, and (b) absolute errors.

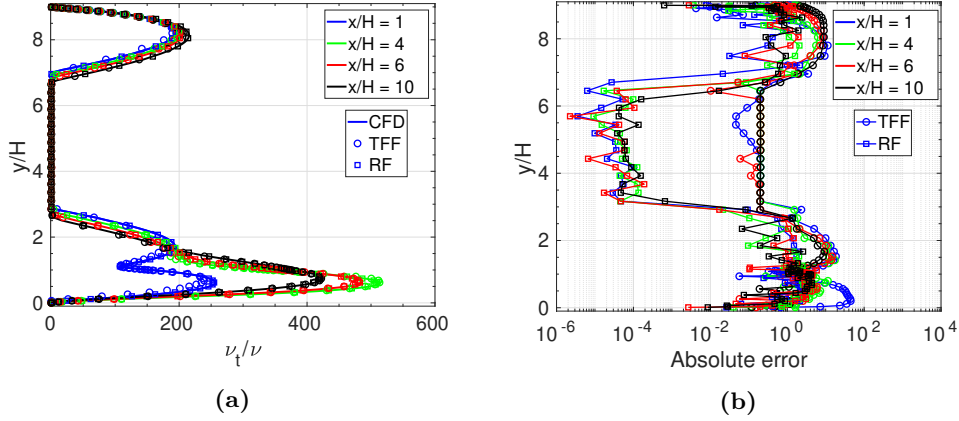


Figure 6: Case I viscosity results at different x/H locations for inlet velocity of 48.5 m/s : (a) viscosity ratio values, and (b) absolute errors.

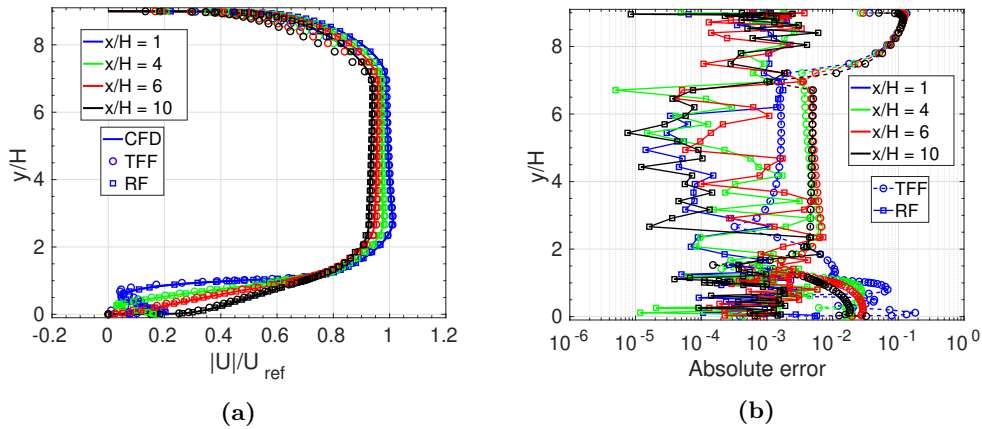
Table 2: Case I relative L_2 norm of the viscosity ratio.

Model	$U_{inlet}, m/s$	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	44.2	0.082	0.019	0.017	0.016
RF	44.2	0.008	0.007	0.007	0.006
TFF	48.5	0.104	0.020	0.018	0.017
RF	48.5	0.008	0.008	0.008	0.007

$= 1$, this norm is one order of magnitude lower for RF, while at the remaining locations, it is approximately half the value when compared to TFF.

Figure 7 shows the velocity ratio and the absolute errors at selected downstream locations for the test case with an inlet velocity of $44.2 m/s$ for the different ML algorithms and CFD simulations. The absolute errors in RF fluctuate significantly from the lower to upper walls, with this error fluctuating by approximately three orders of magnitude. However, RF outperforms TFF in predicting velocity ratios. TFF is especially poor at predicting the velocity ratios near the upper and lower walls as seen in Fig. 7(b). Figure 8 shows similar results for the test case with $U_{inlet} = 48.5 m/s$ as seen with $U_{inlet} = 44.2 m/s$ in Fig. 7. Table 3 shows that RF is more than one order of magnitude better at predicting the velocity ratios for this case when compared to TFF, given by the relative L_2 norm.

RF outperforms TFF in predicting the pressure coefficients on the lower wall of the backward-facing step. These results are shown in Figs. 9 and 10 for the two test cases selected. The pressure coefficient profile from RF matches the CFD results well, while small differences can be seen using TFF. The corresponding relative L_2 norms are shown in Table 4, with RF again

**Figure 7:** Case I velocity results at different x/H locations for inlet velocity of $44.2 m/s$: (a) velocity ratio values, and (b) absolute errors.

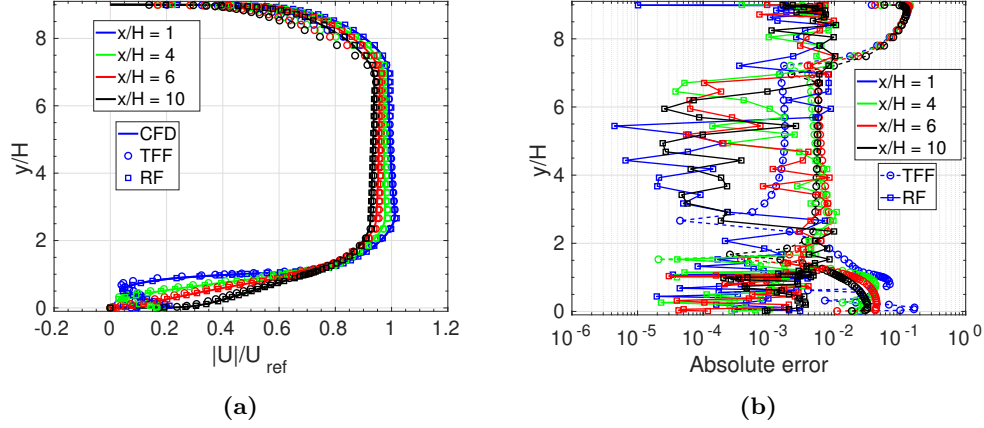


Figure 8: Case I velocity results at different x/H locations for inlet velocity of 48.5 m/s : (a) velocity ratio values, and (b) absolute errors.

Table 3: Case I relative L_2 norm of the velocity ratio.

Model	$U_{inlet}, \text{ m/s}$	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	44.2	0.086	0.072	0.071	0.066
RF	44.2	0.002	0.002	0.003	0.002
TFF	48.5	0.092	0.080	0.079	0.072
RF	48.5	0.004	0.005	0.006	0.006

having significantly lower values than the comparison method.

Figures 11 and 12 show the skin friction coefficient profile on the lower wall for the different test cases, respectively. RF again outperforms TFF at capturing this profile as seen by the lower absolute errors in Figs. 11(b) and 12(b), respectively. This is also shown by the lower relative L_2 norms in Table 4. The skin friction coefficient profile from RF is able to match CFD results well, with the exception for the case with $U_{inlet} = 44.2 \text{ m/s}$ and at x/H of 2. It is important to note

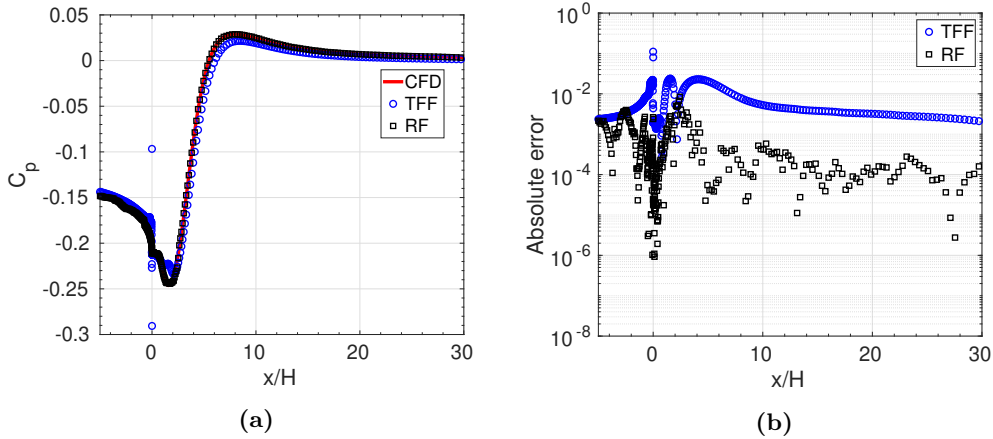


Figure 9: Case I pressure results on bottom wall for inlet velocity of 44.2 m/s : (a) pressure coefficient values, and (b) absolute errors.

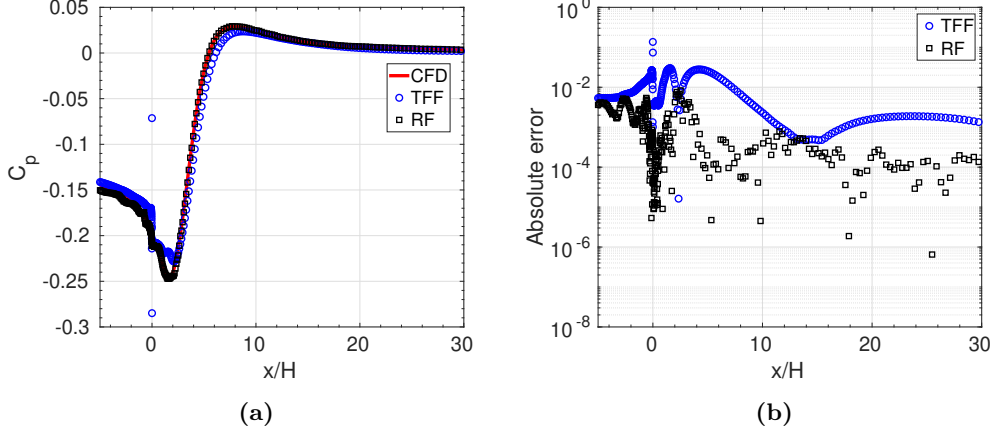


Figure 10: Case I pressure results on bottom wall for inlet velocity of 48.5 m/s: (a) pressure coefficient values, and (b) absolute errors.

Table 4: Case I relative L_2 norm of the pressure and skin friction coefficients on the bottom wall.

Model	$U_{inlet}, m/s$	C_p	C_f
TFF	44.2	0.075	1.032
RF	44.2	0.008	0.034
TFF	48.5	0.092	1.049
RF	48.5	0.012	0.031

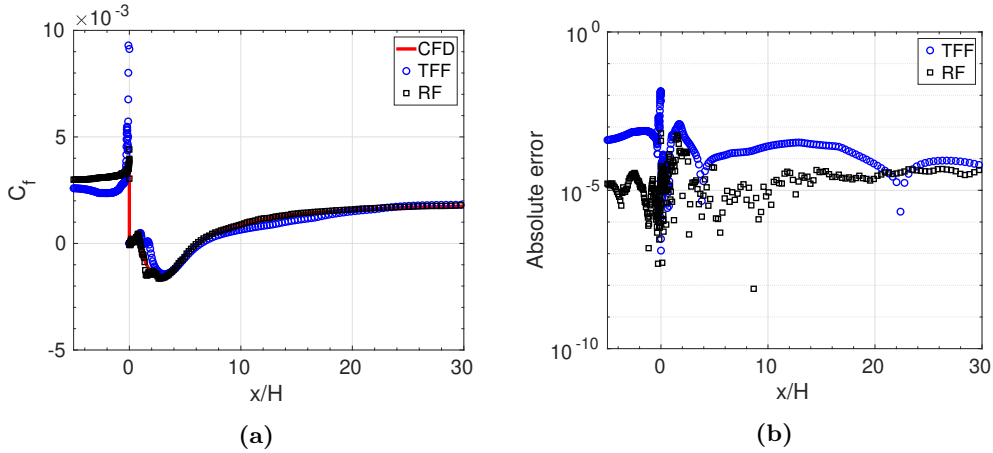


Figure 11: Case I skin friction results on bottom wall for inlet velocity of 44.2 m/s: (a) skin friction coefficient values, and (b) absolute errors.

that the skin friction coefficient values are heavily affected by the turbulent viscosity values near the wall. In this case, the turbulent viscosity is in the order of magnitude of 10^{-8} near the walls, while the maximum turbulent viscosity is in the order of magnitude of 10^{-3} . Small changes in the value of this viscosity near the walls can have a big impact on the skin friction value of the wall. Therefore, it is crucial for the ML algorithms to accurately predict this viscosity near the walls of the domain.

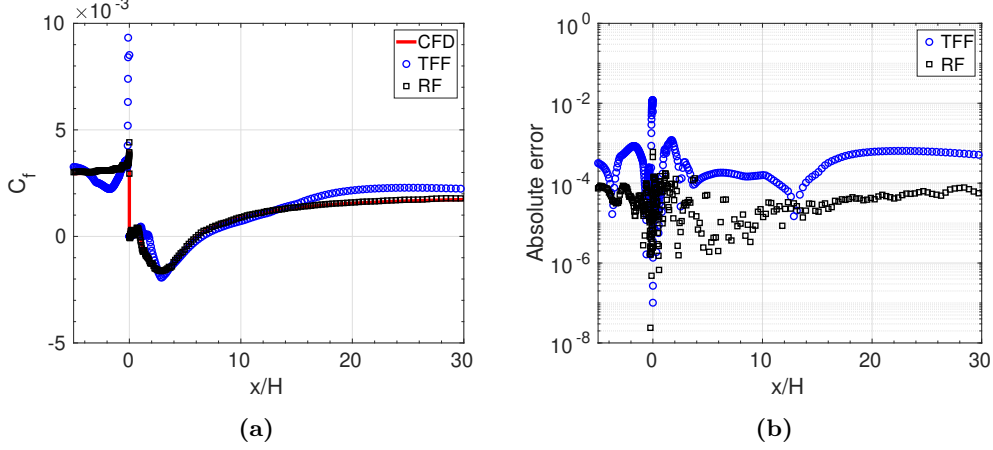


Figure 12: Case I skin friction results on bottom wall for inlet velocity of 48.5 m/s: (a) skin friction coefficient values, and (b) absolute errors.

The viscosity ratio contours for CFD, TFF and RF are shown in Figs. 13, 14, and 15, respectively. No noticeable differences in these contours plots between the ML algorithms and CFD can be seen in these figures. The velocity ratio contours similarly do not show any noticeable difference between the two ML algorithms and CFD as seen in Figs. 16, 17, and 18. The pressure coefficient contours in Figs. 19, 20, and 21 show that RF is better at predicting pressures than TFF. Downstream of the step, at around $x/H = 2$, the pressure coefficients for TFF are higher than CFD as seen in blue, while at $x/H = 10$ are lower than CFD as seen in red.

Table 5 list the computational cost involved in Case I. The data accumulation time is the same for RF and TFF as the same training data is used. RF is faster to train than TFF (which uses NN) and is also significantly faster at prediction compared to both TFF and CFD.

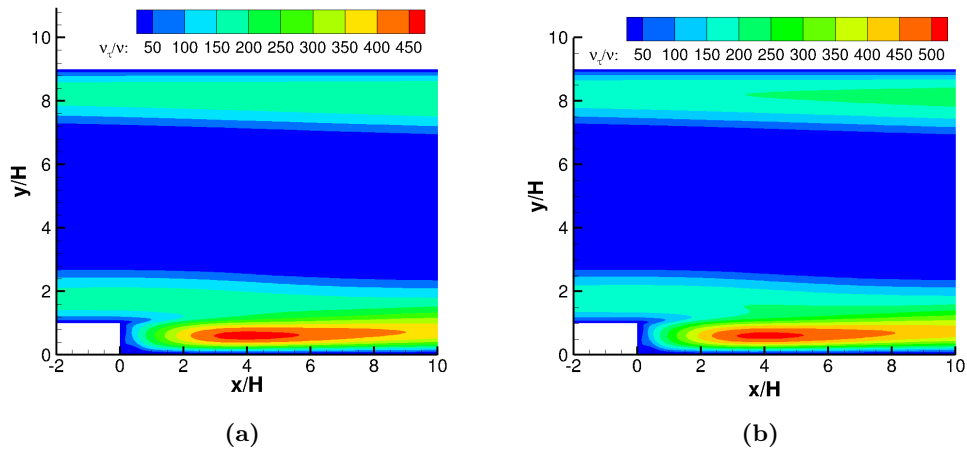


Figure 13: Case I CFD viscosity ratio contours for inlet velocities of: (a) 44.2 m/s, and (b) 48.5 m/s.

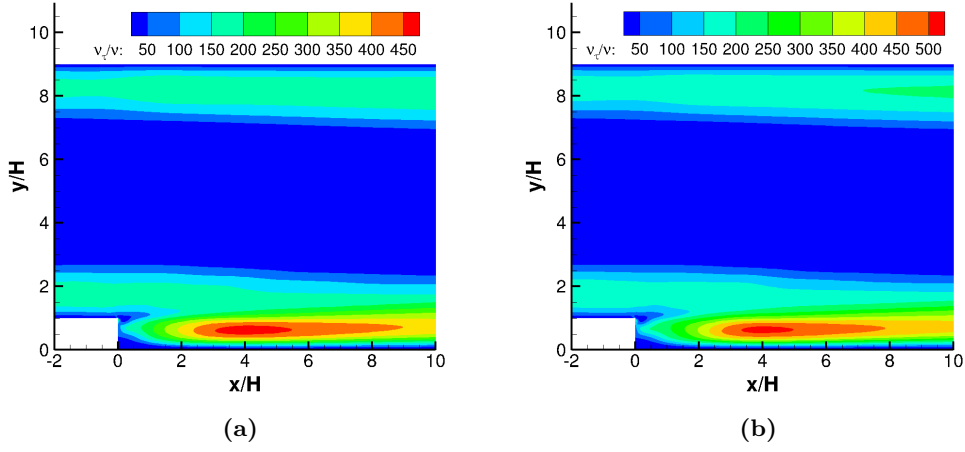


Figure 14: Case I TFF viscosity ratio contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

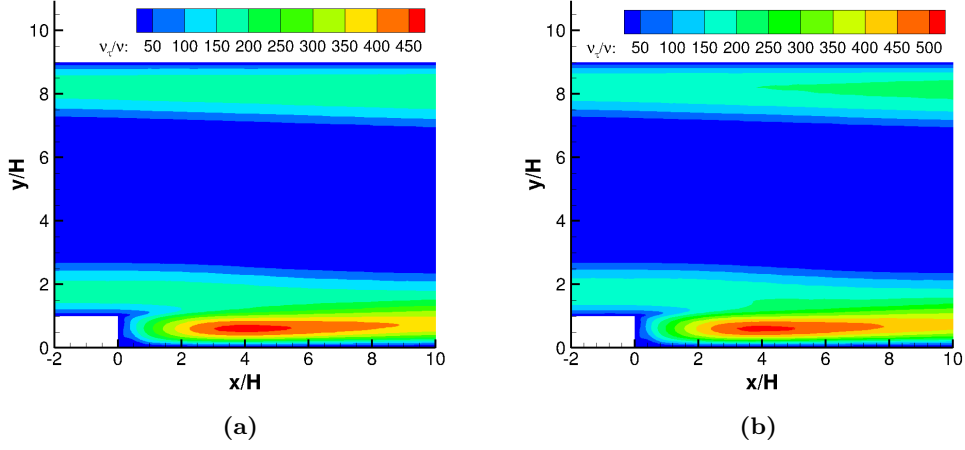


Figure 15: Case I RF viscosity ratio contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

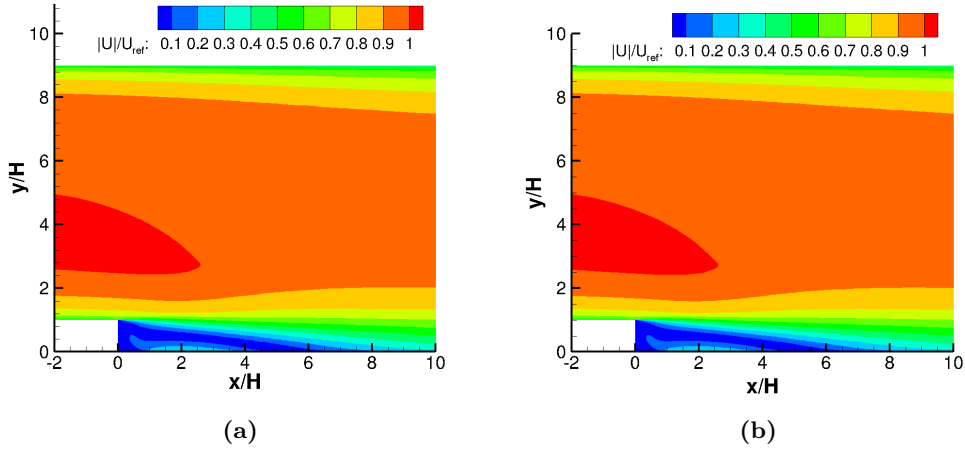


Figure 16: Case I CFD velocity ratio contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

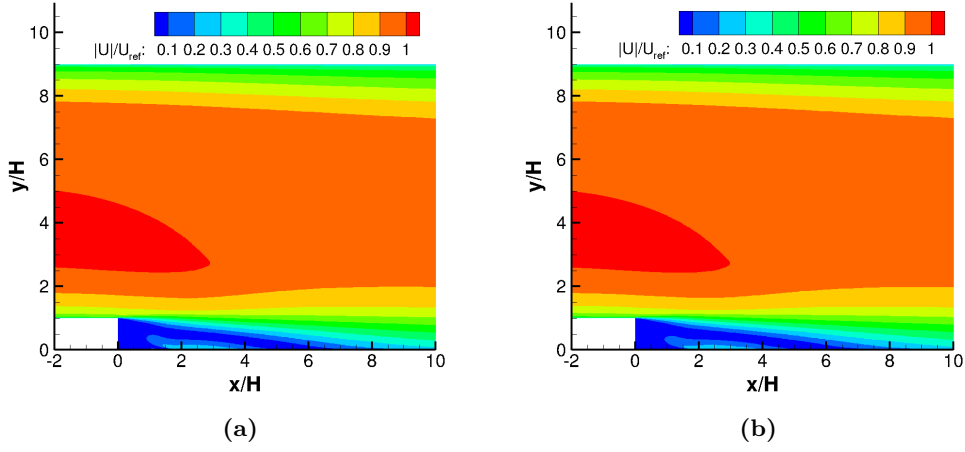


Figure 17: Case I TFF velocity ratio contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

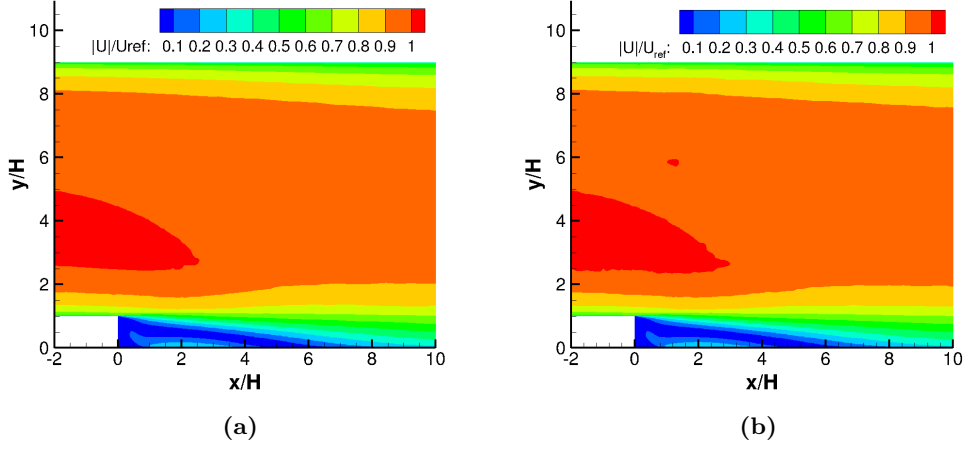


Figure 18: Case I RF velocity ratio contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

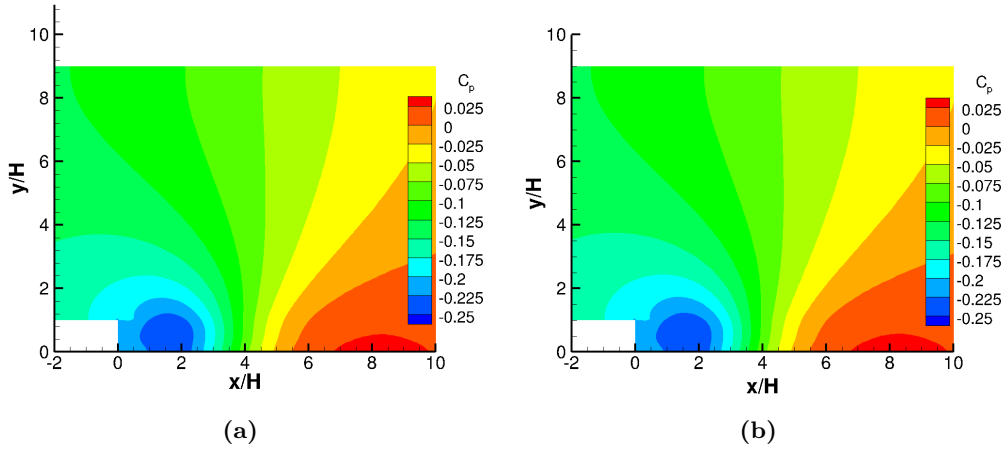


Figure 19: Case I CFD pressure coefficient contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

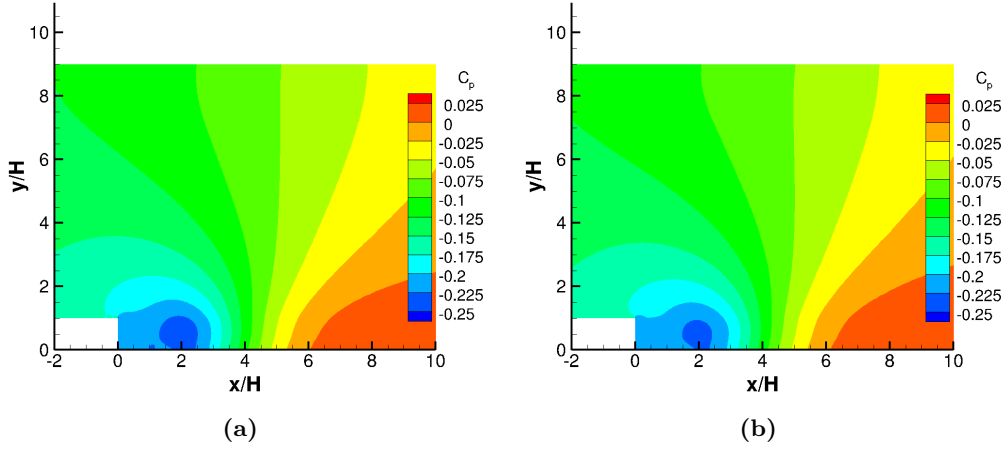


Figure 20: Case I TFF pressure coefficient contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

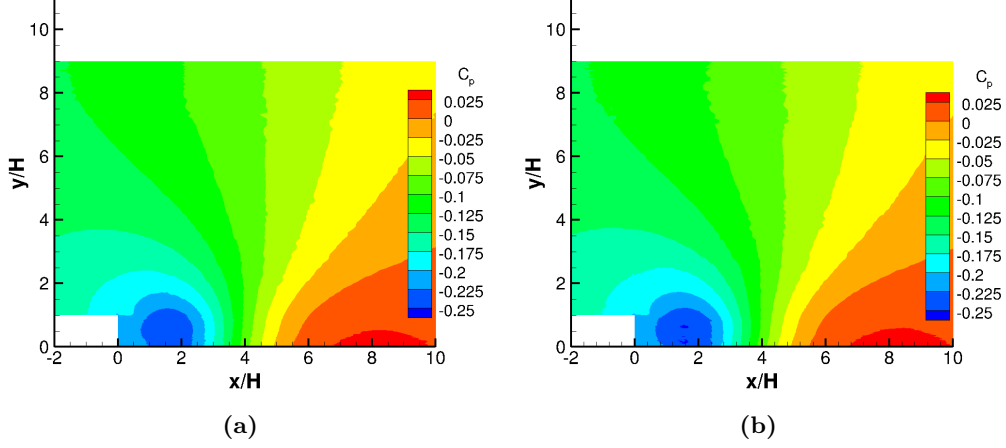


Figure 21: Case I RF pressure coefficient contours for inlet velocities of: (a) 44.2 m/s , and (b) 48.5 m/s .

Table 5: Case I computational cost.

Model	U_{inlet} , m/s	Data accumulation, s	Training, s	Prediction, s
CFD	44.2	-	-	474
TFF	44.2	4,740	170	90
RF	44.2	4,740	70	1
CFD	48.5	-	-	473
TFF	48.5	4,740	170	79
RF	48.5	4,740	70	1

3.2 Case II

The flow past a backward-facing step is simulated in this case. However, this case considers the effect of varying the step height, H , instead of U_{inlet} . The Reynolds number based on the step height is defined as $Re_H = U_{inlet}H/\nu$, where U_{inlet} is the freestream velocity at the inlet to

the domain, H is the step height, and ν is the kinematic viscosity. The Re_H for this case varies between 18,000 and 72,000, based on the selected H values.

3.2.1 Problem definition

Similar to Case I, the objective of this case is to replace the high-fidelity RANS simulations with a RF ML algorithm that predicts the RANS flow field. This case, however, considers the effect of varying the step height, H , instead of U_{inlet} . U_{inlet} is fixed to a value of 44.2 m/s . The inputs to the RF include x_c/H , y_c/H , u_p and v_p , and H . The RF is trained with data from six different step heights, namely, $H = 0.5h$, $0.75h$, $1.25h$, $1.5h$, $1.75h$, and $2.0h$, respectively. This training dataset is sampled at $0.25h$ difference apart, except between $0.75h$ and $1.25h$, where the difference is $0.5h$. This trained RF is then tested with step heights $1.0h$ and $1.9h$, respectively. The results are then compared to those from RANS CFD simulations and TFF. The training and testing sets are the same as those used by Maulik et al. [49]. The CFD and mesh setup is similar to Case I.

3.2.2 Results

Figure 22 shows the viscosity ratio plots for the step height of $1.0h$ at different downstream locations. At the midsection, RF has lower absolute errors compared to TFF by approximately two orders of magnitude. In the remaining sections, TFF has two orders of magnitude lower errors than RF. For the step height of $1.9h$, similar trends can be seen in the midsection as shown in Fig. 23. In the remaining section, however, the difference in absolute errors between RF and TFF is lower compared to a step height of $1.0h$. However, RF still has higher errors than TFF.

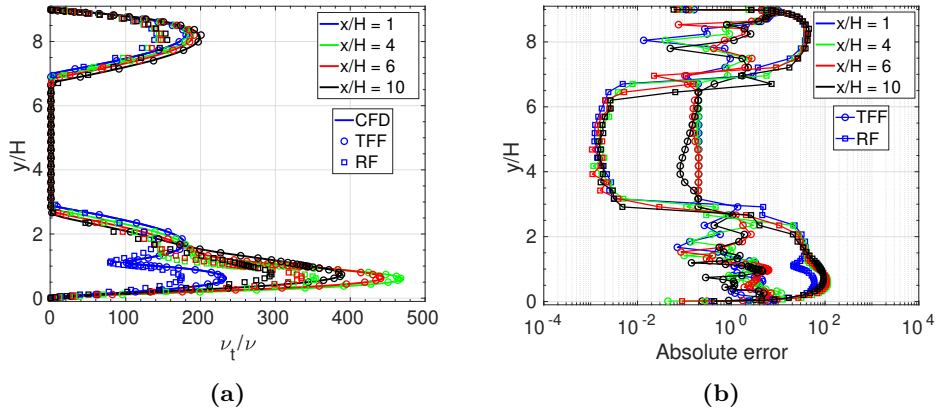


Figure 22: Case II viscosity results at different x/H locations for step height of $1.0h$: (a) viscosity ratio values, and (b) absolute errors.

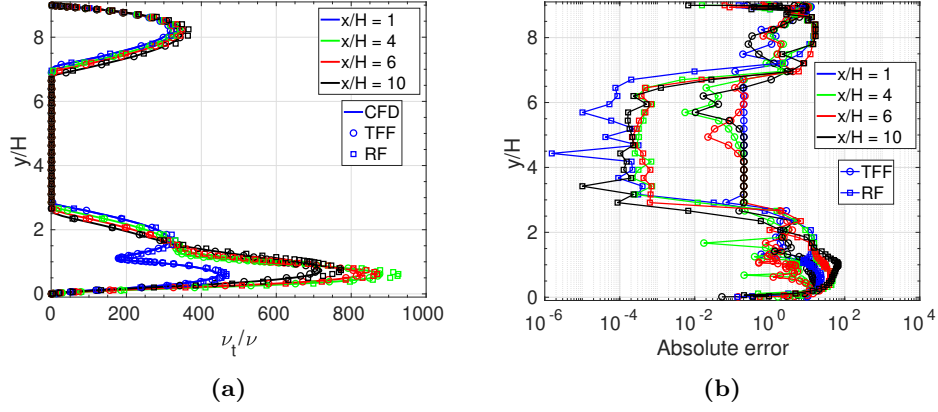


Figure 23: Case II viscosity results at different x/H locations for a step height of $1.9h$: (a) viscosity ratio values, and (b) absolute errors.

The corresponding relative L_2 norm for the viscosity ratios is shown in Table 6. TFF has a lower relative L_2 norm when compared to RF. Between the two step heights, for RF, the step height of $1.9h$ has significantly lower relative L_2 norm values when compared to $1.0h$. This is due to the lack of training data in the vicinity of $1.0h$ when compared to $1.9h$.

RF is better at predicting the velocity ratios than TFF for both the step heights as seen in Figs. 24 and 25. This is confirmed by the lower relative L_2 norm values shown in Table 7.

Table 6: Case II relative L_2 norm of the viscosity ratio.

Model	H	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	$1.0h$	0.032	0.018	0.017	0.014
RF	$1.0h$	0.236	0.238	0.234	0.233
TFF	$1.9h$	0.038	0.014	0.015	0.023
RF	$1.9h$	0.050	0.051	0.049	0.082

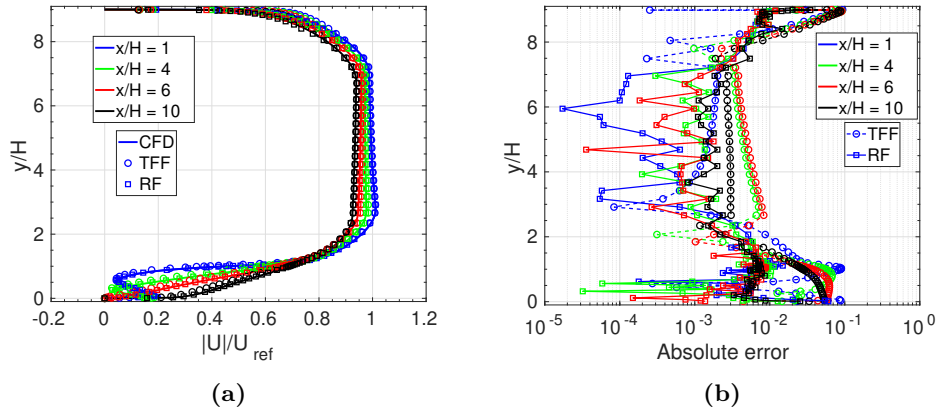


Figure 24: Case II velocity results at different x/H locations for a step height of $1.0h$: (a) velocity ratio values, and (b) absolute errors.

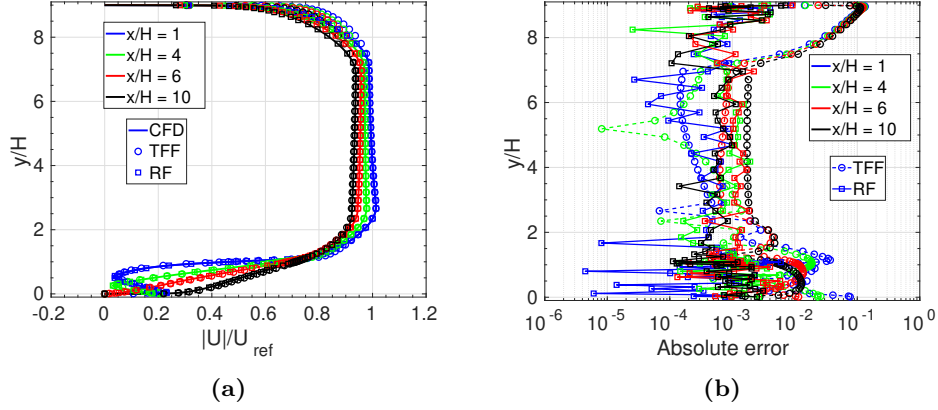


Figure 25: Case II velocity ratio at different x/H locations for a step height of 1.9h: (a) velocity ratio values, and (b) absolute errors.

Table 7: Case II relative L_2 norm of the velocity ratio.

Model	H	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	1.0h	0.063	0.055	0.056	0.050
RF	1.0h	0.016	0.010	0.010	0.011
TFF	1.9h	0.065	0.061	0.059	0.056
RF	1.9h	0.002	0.002	0.003	0.002

These results are similar to those in Case I as shown in Figs. 7 and 8. TFF is especially poor at predicting the velocities in the region between $y/H = 7$ and the upper wall, as well as between $y/H = 3$ and the lower wall.

RF is able to capture the pressure coefficient trend on the lower wall more accurately than TFF as seen in Figs. 26(a) and 27(a) for step heights of 1.0h and 1.9h, respectively. RF has about one order of magnitude lower absolute errors compared to TFF as shown in Figs. 26(b)

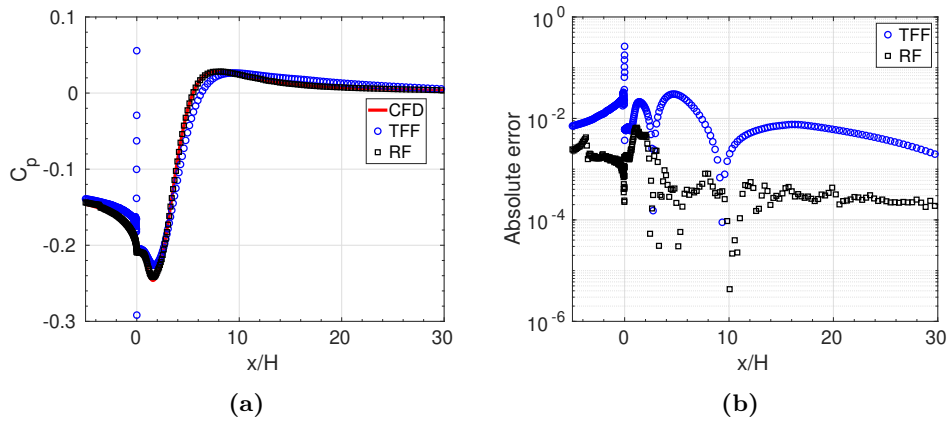


Figure 26: Case II pressure results on bottom wall for a step height of 1.0h: (a) pressure coefficient values, and (b) absolute errors.

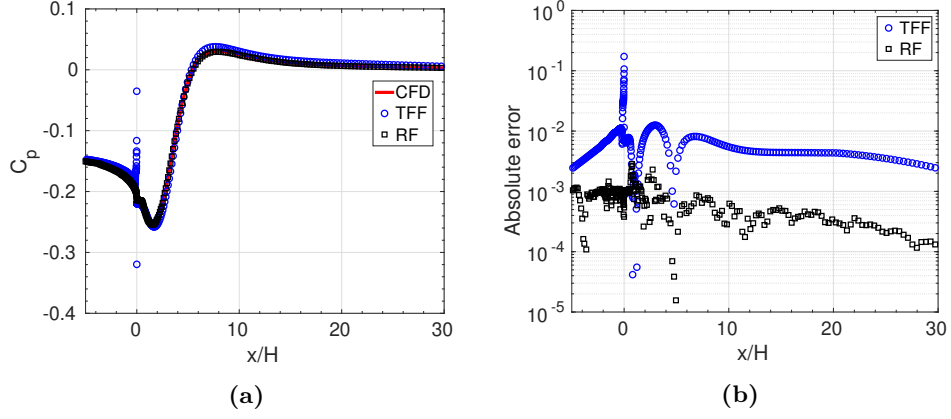


Figure 27: Case II pressure results on bottom wall for a step height of 1.9h: (a) pressure coefficient values, and (b) absolute errors.

and 27(b). Figures 28 and 29 show that RF and TFF are better at predicting the skin friction coefficient on the bottom wall for the case with a step height of 1.9h compared to 1.0h. This is due to the lack of training data in the vicinity of 1.0h, as mentioned above. The corresponding

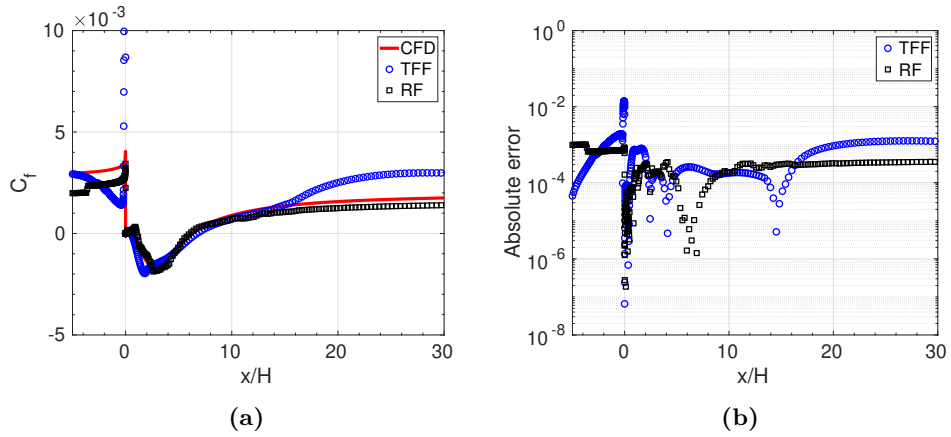


Figure 28: Case II skin friction results on bottom wall for a step height of 1.0h: (a) skin friction coefficient values, and (b) absolute errors.

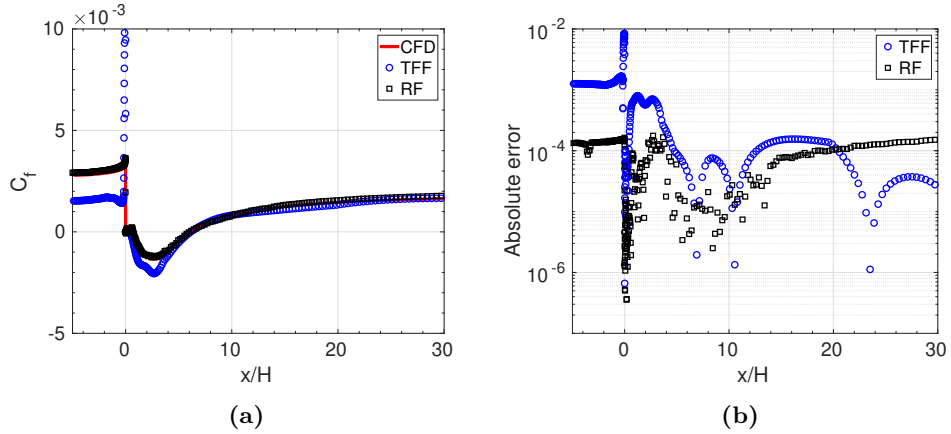


Figure 29: Case II skin friction results on bottom wall for a step height of 1.9h: (a) skin friction coefficient values, and (b) absolute errors.

relative L_2 norms for both the coefficients are shown in Table 8, with RF having significantly lower values than TFF.

Figures. 30, 31, and 32 show the viscosity ratio contours from CFD, TFF and RF, respectively. For the step height of 1.0h, TFF viscosity ratio contours (Fig. 31(a)) match those from CFD (Fig. 30(a)) well. While RF under predicts the viscosities in regions between $y/H = 7$ and the upper wall, as well as between $y/H = 3$ and the lower wall, as seen in Fig. 32(a). For the step height of 1.9h, both TFF and RF are better at predicting the viscosities when compared to a step height of 1.0h. The velocity ratio contours for the ML algorithms match CFD results well as seen in Figs. 33, 34, and 35. TFF, however, is poor at predicting the pressure contours as seen in Figs. 36, 37, and 38.

Table 9 shows the computational cost involved in Case II. Similar to Case I, RF is cheaper to both train on the given dataset and to predict the flow field, when compared to TFF. RF is approximately three times cheaper to train than TFF. RF takes about one second to predict the entire flow field.

Table 8: Case II relative L_2 norm of the pressure and skin friction coefficients on the bottom wall.

Model	H	C_p	C_f
TFF	1.0h	0.155	1.434
RF	1.0h	0.012	0.222
TFF	1.9h	0.102	0.927
RF	1.9h	0.005	0.055

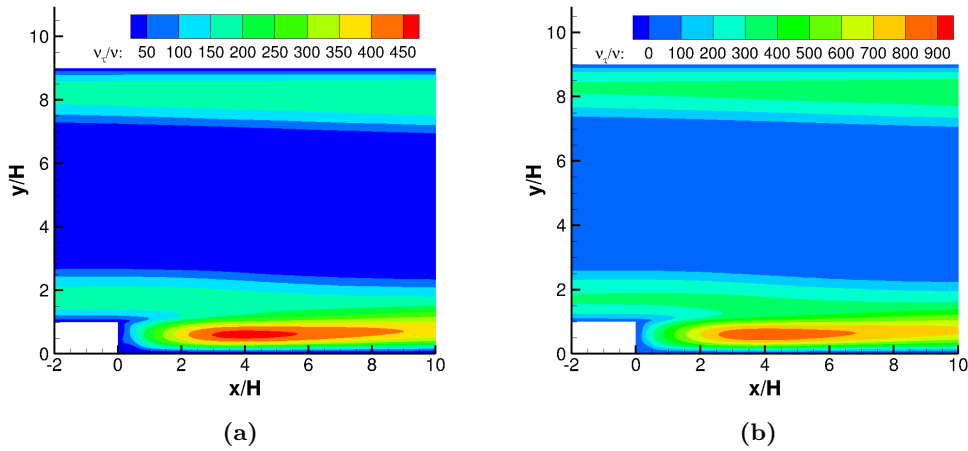


Figure 30: Case II CFD viscosity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

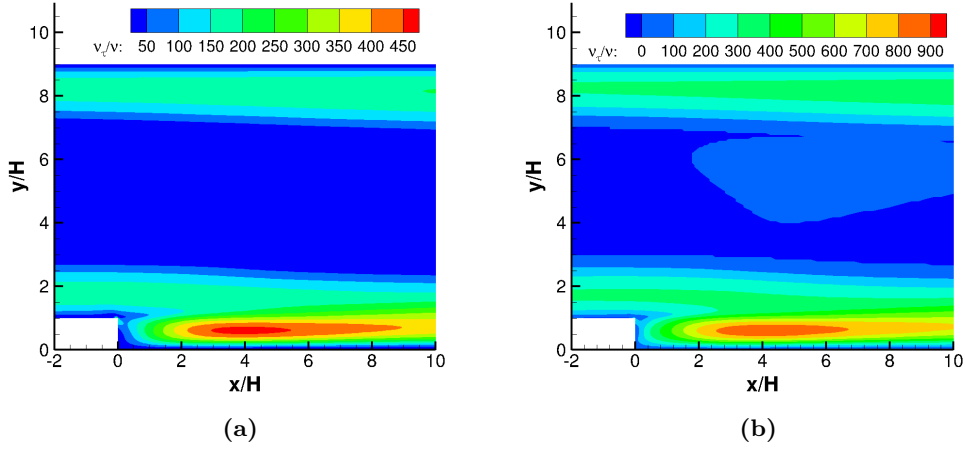


Figure 31: Case II TFF viscosity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

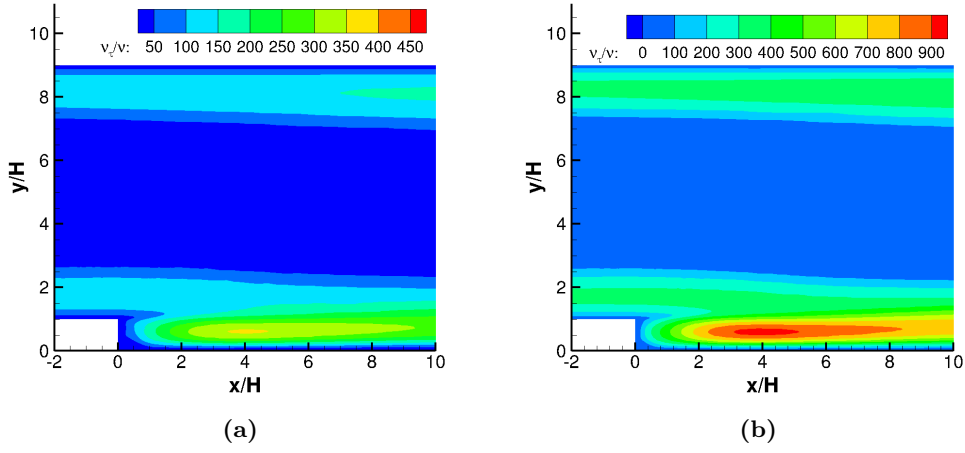


Figure 32: Case II RF viscosity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

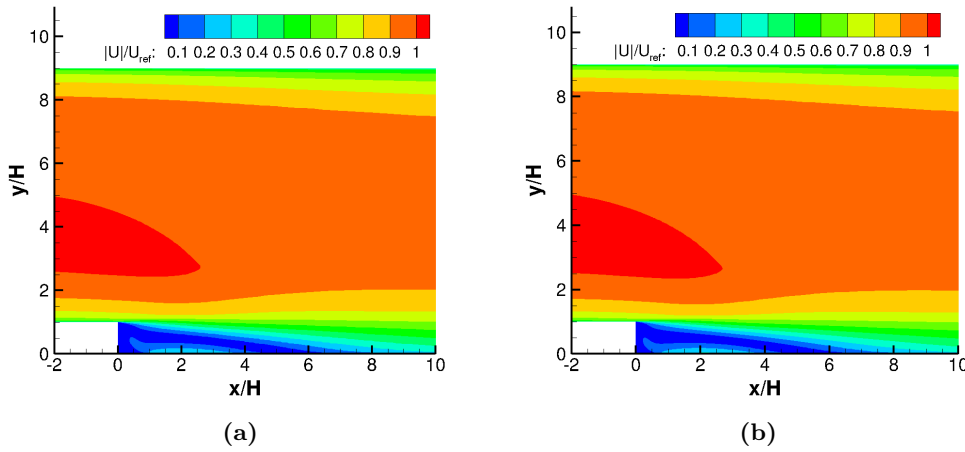


Figure 33: Case II CFD velocity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

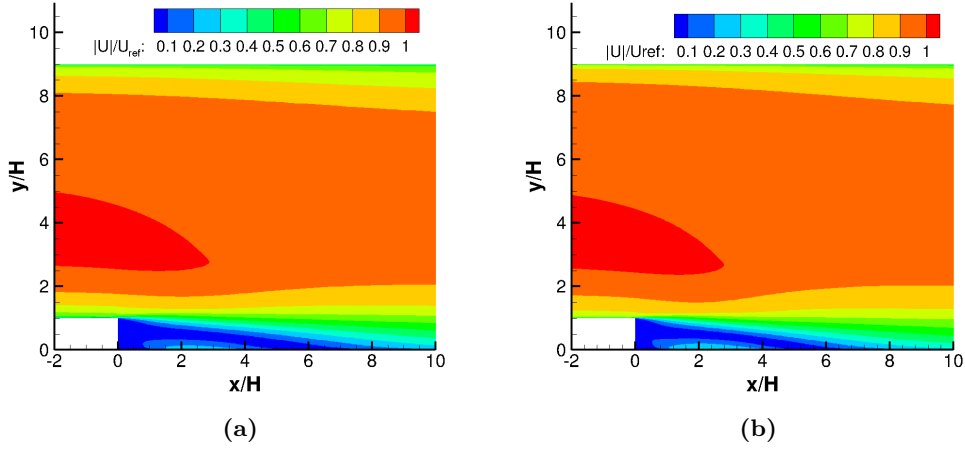


Figure 34: Case II TFF velocity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

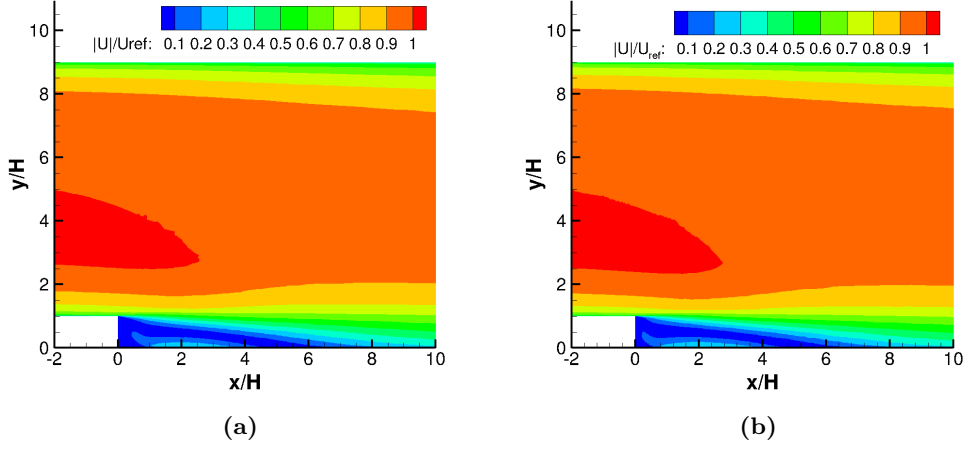


Figure 35: Case II RF velocity ratio contours for step heights of: (a) 1.0h, and (b) 1.9h.

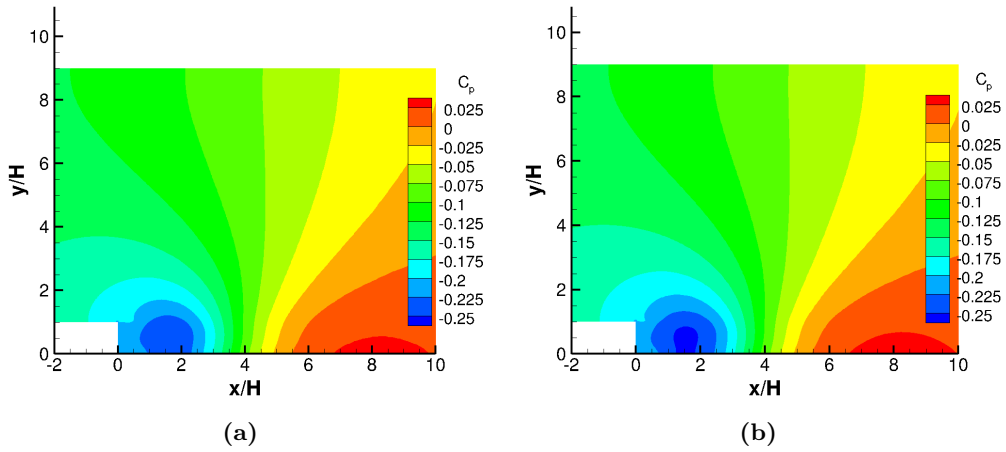


Figure 36: Case II CFD pressure coefficient contours for step heights of: (a) 1.0h, and (b) 1.9h.

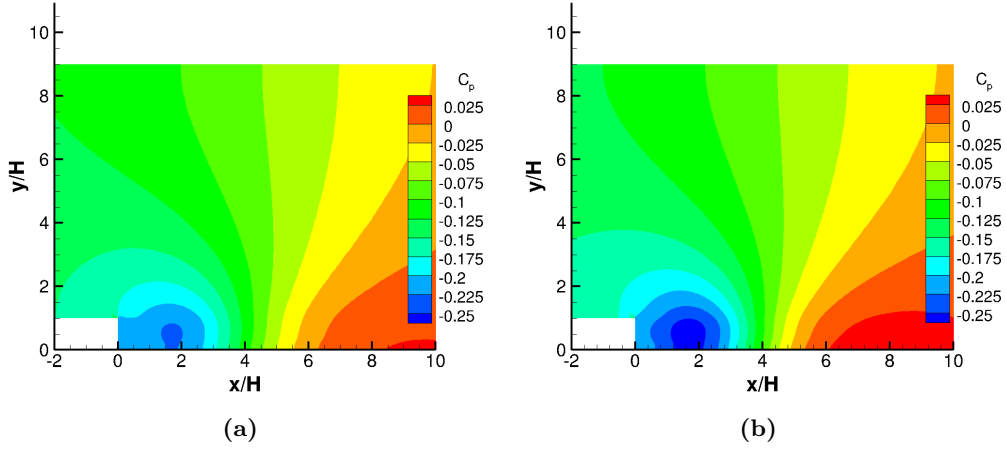


Figure 37: Case II TFF pressure coefficient contours for step heights of: (a) 1.0h, and (b) 1.9h.

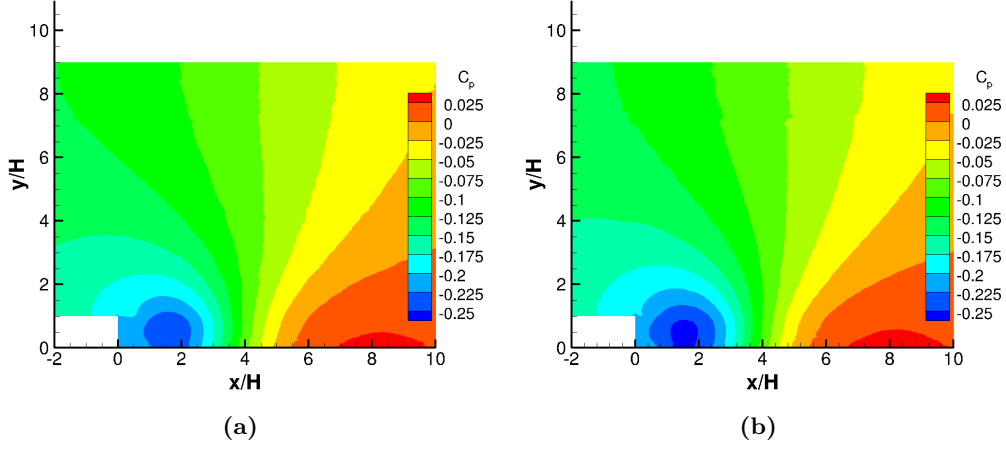


Figure 38: Case II RF pressure coefficient contours for step heights of: (a) 1.0h, and (b) 1.9h.

Table 9: Case II computational cost.

Model	H	Data accumulation, s	Training, s	Prediction, s
CFD	1.0h	-	-	476
TFF	1.0h	2,856	126	77
RF	1.0h	2,856	40	1
CFD	1.9h	-	-	473
TFF	1.9h	2,856	126	70
RF	1.9h	2,856	40	1

3.3 Case III

In this case, the flow around an airfoil is simulated. The flow is steady, subsonic, and incompressible with a Mach number of 0.3. The angle of attack of the flow is 10 degrees. The Reynolds number based on the chord length, c , is defined as $Re_c = U_\infty c / \nu$ and has a value of 6×10^6 , where

U_∞ is the freestream velocity with a magnitude of 51.5 m/s , and ν is the kinematic viscosity with a value of $8.58 \times 10^{-6} \text{ m}^2/\text{s}$.

3.3.1 Problem definition

Similar to the previous two cases, the objective of this case is to replace the RANS model with the RF ML algorithm. The RF algorithm is trained using 20 different airfoil shapes parameterized using B-spline curves [64] with eight control points. These 20 shapes are generated using the LHS [52] plan, where the upper and lower bound values are set to $(1 \pm 25\%) \mathbf{x}_0$. $\mathbf{x}_0$ is the vector of B-spline control points corresponding to the NACA 0012 airfoil. The inputs to the RF model include ϕ , ψ , and the eight B-spline control point locations, and the outputs include u_r , v_r , p_r , and ν_t . This trained RF model is tested on the NACA 0012 airfoil and the RAE 2822 airfoil. The results are compared to RANS CFD simulations.

3.3.2 CFD setup and validation

To simulate the flow around an airfoil, an O-grid mesh is generated and shown in Fig. 39 using pyHyp [65]. The y^+ value is less than two and the far-field is located $55c$ from the center of the airfoil. The mesh and CFD setup is validated with experimental data from [66]. The results from the grid independence study are shown in Table 10.

Both the high- and low-fidelity flow is simulated using OpenFOAM version 5.0 [63]. potentialFlowFoam is the low-fidelity solver, while simpleFoam is the high-fidelity solver. The high-fidelity solver simulates the Spalart-Allmaras [55] RANS turbulence model. The far-field of the domain is set to a pressure far-field boundary condition, while the walls of the airfoil are set

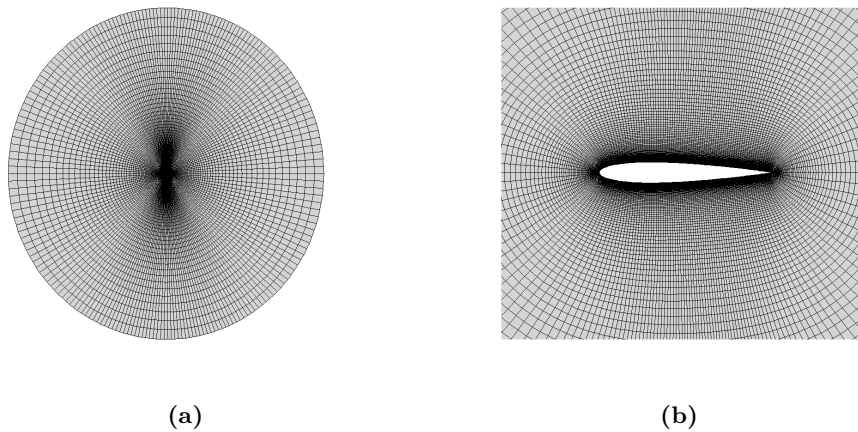


Figure 39: O-grid for Case III: (a) domain, and (b) near the airfoil.

Table 10: Grid convergence study for the NACA 0012 airfoil in subsonic flow.

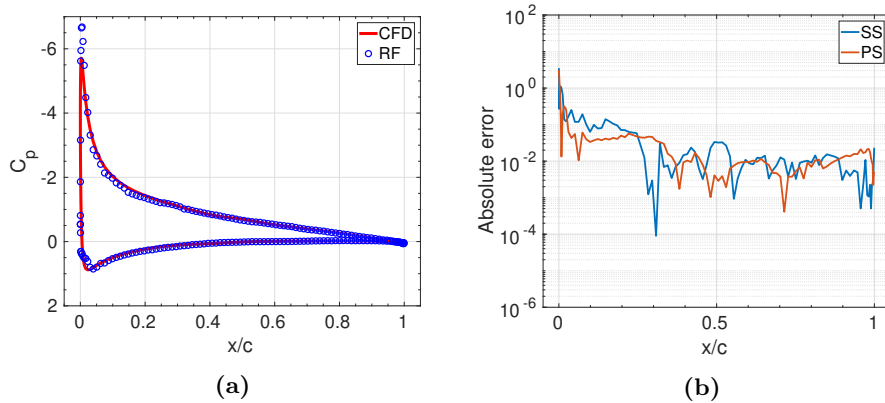
Mesh	No. of cells	α , deg	C_l , l.c.	C_d , d.c.	Simulation time*, s
L3	11,286	10.18	109.4	148.6	77
L2	39,402	10.18	109.7	139.2	232
L1	145,236	10.18	109.5	137.0	1,120
L0	557,702	10.18	110.8	135.0	6,016
Exp. [66]	-	10.18	108.1	116.5	-

*Computed on a high-performance cluster with 4 processors.

to no-slip. The convergence criteria is set to either of the two; (a) pressure and velocity residuals fall below 10^{-6} or (b) a maximum number of iterations of 10,000 is met. Table 10 shows the lift and drag coefficient values. The RANS model is better at predicting the lift compared to the drag. From Table 10, mesh L2 is used in this study.

3.3.3 Results

Figure 40(a) shows the pressure coefficient profile for the RF algorithm matches the CFD results well for the NACA 0012 test case, except at the leading edge. The absolute errors are highest at the leading edge of the airfoil and decrease to around mid-chord. From the mid-chord to the trailing edge, the errors are nearly constant, with fluctuations around an order of magnitude seen in Fig. 40(b). For the RAE 2822 case, RF is able to capture the pressure coefficient profile well, again with the exception of the leading edge as shown in Fig. 41(a). For the suction side (SS), the pressure is nearly constant in-between the leading and trailing edges, while on the pressure side (PS) decreases to about mid-chord, then increases back again. These trends are observed in Fig. 41(b). The relative L_2 norm for both the airfoils is shown in Table 11. RF has lower errors for the RAE 2822 airfoil compared to the NACA 0012 airfoil.

**Figure 40:** Case III NACA 0012 airfoil pressure results: (a) pressure coefficient values, and (b) absolute errors on the suction side and pressure side.

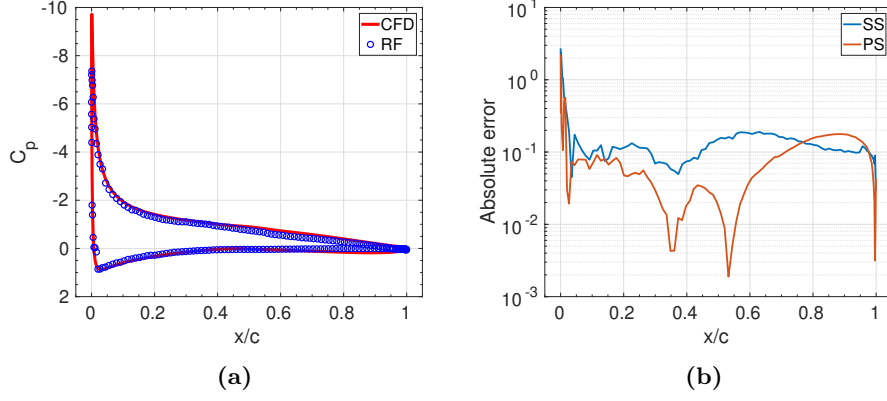


Figure 41: Case III RAE 2822 airfoil pressure results: (a) pressure coefficient values, and (b) absolute errors on the suction side and pressure side.

The skin friction results for the NACA 0012 airfoil are shown in Fig. 42. RF is poor at predicting the skin friction at the leading edge, whereas, on the remaining surface, it matches the trends from CFD results. The absolute errors for the PS and SS decrease from the leading edge to 0.25 chord and then remain nearly constant till the trailing edge. These errors fluctuate by an order of magnitude from leading to trailing edges. The corresponding results for the RAE 2822 airfoil are shown in Fig. 43. In this case, RF poorly predicts the skin friction at both leading and trailing edges. On the remaining surface of the airfoil, the skin friction profile from RF matches the CFD results, with absolute errors shown in 43(b). Similar to the pressure coefficients, the skin friction coefficient has a lower relative L_2 norm for the RAE 2822 airfoil than the NACA 0012 airfoil and is shown in Table 11.

Pressure coefficient contours for the two airfoil cases are shown in Figs. 44 and 45. These contours from RF match the CFD results better for the NACA 0012 airfoil than for the RAE

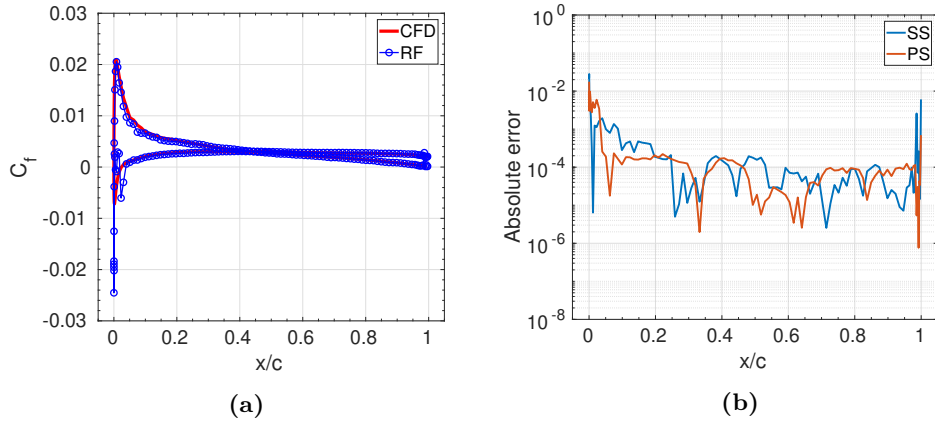


Figure 42: Case III NACA 0012 airfoil skin friction results: (a) skin friction coefficient values, and (b) absolute errors on the suction side and pressure side.

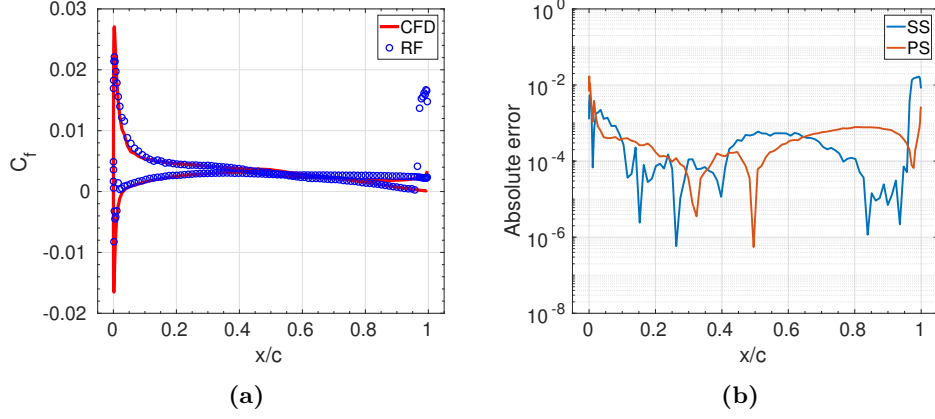


Figure 43: Case III RAE 2822 airfoil skin friction results: (a) skin friction coefficient values, and (b) absolute errors on the suction side and pressure side.

Table 11: Case III relative L_2 norm of the pressure and skin friction coefficients on the airfoil.

Airfoil	C_p	C_f
NACA 0012	0.366	0.781
RAE 2822	0.219	0.654

2822 airfoil. The most noticeable differences are seen in Fig. 45 on the PS and around the trailing edge. The corresponding Mach numbers are shown in Figs. 46 and 47. No noticeable results can be observed for the NACA 0012 airfoil, while for the RAE 2822 airfoil, small differences can be seen around the PS of the airfoil.

The computational cost involved in the case is shown in Table 12. Similar to the previous two cases, the RF algorithm takes only a second to predict the entire flow field.

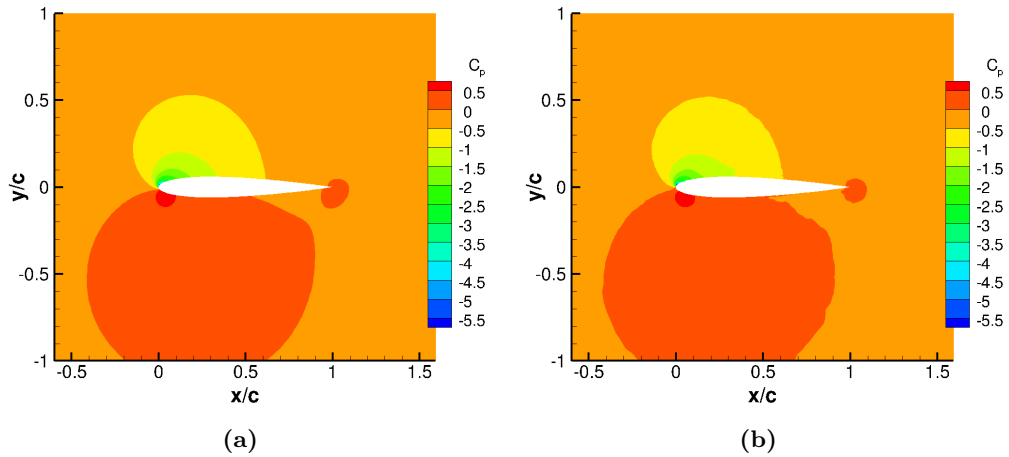


Figure 44: Case III NACA 0012 pressure coefficient contours for: (a) CFD, and (b) random forest.

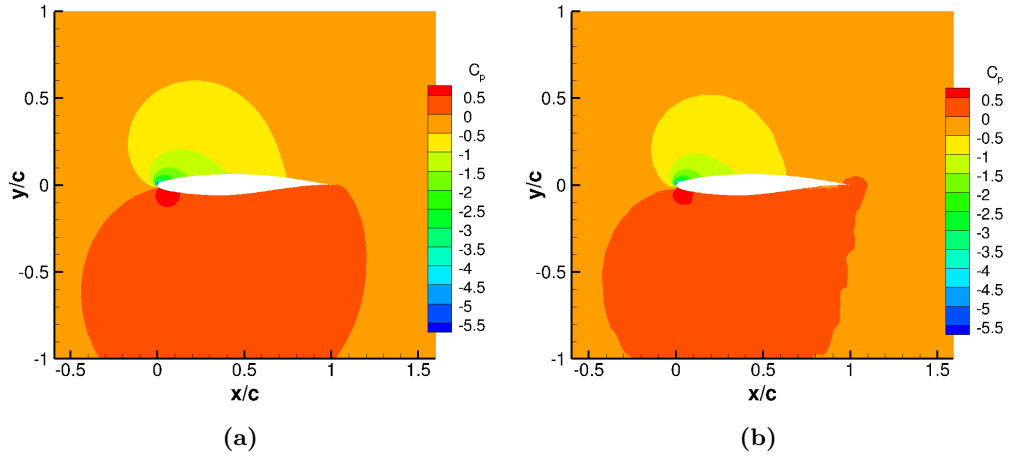


Figure 45: Case III RAE 2822 pressure coefficient contours for: (a) CFD, and (b) random forest.

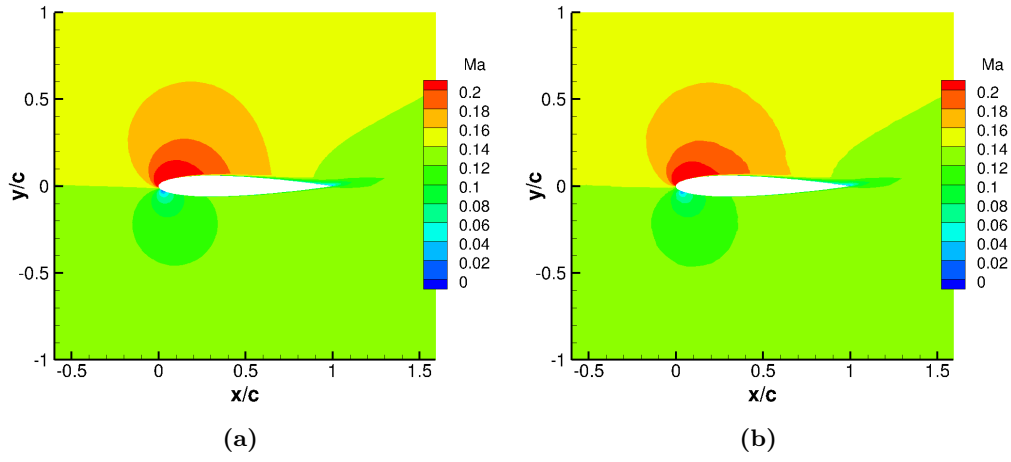


Figure 46: Case III NACA 0012 Mach number contours for: (a) CFD, and (b) random forest.

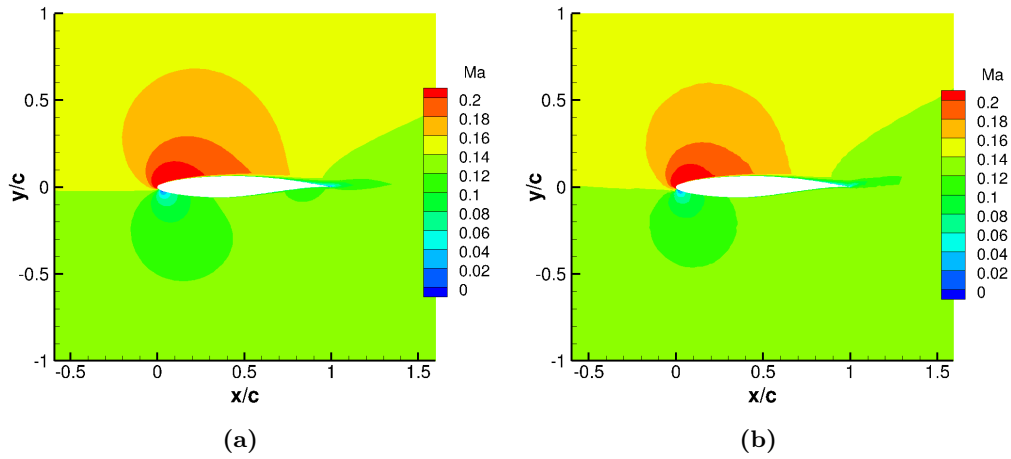


Figure 47: Case III RAE 2822 Mach number contours for: (a) CFD, and (b) random forest.

Table 12: Case III computational cost.

Model	Airfoil	Data accumulation, s	Training, s	Prediction, s
CFD	NACA 0012	-	-	230
RF	NACA 0012	4,600	50	1
CFD	RAE 2822	-	-	220
RF	RAE 2822	4,600	50	1

4 Conclusion

In this work, a novel multifidelity random forest algorithm is proposed and results on three cases are shown. The first two include flow past a backward-facing step, and the third, subsonic flow around an airfoil at ten degrees angle of attack. This framework is compared to the tensorFlowFoam (TFF) framework for the first two cases, and to RANS CFD simulations for all the cases. Potential flow is used as the low-fidelity model, while the Spalart-Allmaras RANS model is used as the high-fidelity model. The inputs to the ML algorithms varied from case to case, however, the outputs from these algorithms were kept the same. The outputs include the cell-center velocities, pressures, and turbulent viscosities. The performance of the ML algorithms were measured using the absolute and relative L_2 norm errors at different locations and for different flow quantities, as well as the computational cost involved in training and predicting the flow fields using these algorithms.

In the first case, the training data is generated using ten different inlet velocities to the backward-facing step, while in the second, six different step heights are used. The inputs to the ML algorithms for the first case include the cell-center locations, the potential flow velocities at the cell-centers, along with the inlet velocity value. For the second case, the cell-centers are normalized with the step height, and the step height is used in place of the inlet velocity as inputs to the ML algorithms. For these two cases, RF outperformed TFF in predicting the various flow quantities, with the exception of the viscosity ratios for the second case.

For the third case, the velocity potential, the stream function, and eight B-spline control points are used as inputs to the RF algorithm. The pressure and skin friction coefficient profile on the surface of the airfoils from RF matches those from RANS CFD simulations well, with the exception of the leading edge. The contours of these coefficients also match the CFD simulations. However, small differences can be seen on the pressure side of the RAE 2822 airfoil.

One major benefit of using RF to make high-fidelity flow field predictions is the number of

CFD simulations required to generate the data to train the RF algorithm. For cases 1, 2, and 3, ten, six, and 20 CFD simulations were used to train the RF algorithm. Significant time is saved during the data generation process which is often the bottleneck in surrogate modeling methods. The other benefit is the low field prediction cost of RF. Once trained, the RF algorithm takes about a second to predict the entire flow field. This is useful in engineering problems that require multiple and repetitive model evaluations.

The biggest limitation of the developed RF algorithm is in choosing the input variables. Each of the cases shown in this study uses different input variables. The input variables affect the prediction capabilities of the RF algorithm tremendously and it is difficult to know which input variables to choose before training the RF algorithm. A thorough investigation of these variables is needed in order to help improve the RF performance as well as make this framework more generic to various engineering problems. Currently, only the offline part of the framework is developed. The online part needs to be developed and applied to various problems involving global sensitivity analysis, uncertainty quantification and propagation, and aerodynamic design optimization.

Acknowledgements

The authors of this work acknowledge the support of the National Science Foundation award No. 1846862.

References

- [1] X. Yan, J. Zhu, M. Kuang, and X. Wang, “Aerodynamic shape optimization using a novel optimizer based on machine learning techniques,” *Aerospace Science and Technology*, vol. 86, pp. 826–835, 2019.
- [2] V. Raul and L. Leifsson, “Surrogate-based aerodynamic shape optimization for delaying airfoil dynamic stall using kriging regression and infill criteria,” *Aerospace Science and Technology*, vol. 111, p. 106555, 2021.
- [3] M. A. Bouhlef, S. He, and J. R. R. A. Martins, “Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes,” *Structural and Multidisciplinary Optimization*, vol. 61, pp. 1363–1376, 2020.

- [4] J. Li, M. A. Bouhlel, and J. R. R. A. Martins, “Data-based approach for fast airfoil analysis and optimization,” *AIAA Journal*, vol. 57, no. 2, pp. 581–596, 2019.
- [5] J. Li, M. Zhang, J. R. R. A. Martins, and C. Shu, “Efficient aerodynamic shape optimization with deep-learning-based filtering,” *AIAA Journal*, vol. 58, no. 10, pp. 4243–4259, 2020.
- [6] X. Du and L. Leifsson, “Optimum aerodynamic shape design under uncertainty by utility theory and metamodeling,” *Aerospace Science and Technology*, vol. 95, p. 105464, 2019.
- [7] A. Feldstein, D. Lazzara, N. Princen, and K. Willcox, “Multifidelity data fusion: Application to blended-wing-body multidisciplinary analysis under uncertainty,” *AIAA Journal*, vol. 58, no. 2, pp. 889–906, 2020.
- [8] M. Panzeri, A. Savelyev, K. Anisimov, R. d’Ippolito, and A. Mirzoyan, “Uncertainty quantification and robust design optimization applied to aircraft propulsion systems,” *Transportation Research Procedia*, vol. 29, pp. 289–302, 2018.
- [9] Y. Ju, C. Zhang, and L. Ma, “Artificial intelligence metamodel comparison and application to wind turbine airfoil uncertainty analysis,” *Advances in Mechanical Engineering*, vol. 8, no. 5, pp. 1–14, 2016.
- [10] V. Sekar, M. Zhang, C. Shu, and B. C. Khoo, “Inverse design of airfoil using a deep convolutional neural network,” *AIAA Journal*, vol. 57, no. 3, pp. 993–1003, 2019.
- [11] G. Sun, Y. Sun, and S. Wang, “Artificial neural network based inverse design: Airfoils and wings,” *Aerospace Science and Technology*, vol. 42, pp. 415–428, 2015.
- [12] X. Wang, S. Wang, J. Tao, G. Sun, and J. Mao, “A pca-ann-based inverse design model of stall lift robustness for high-lift device,” *Aerospace Science and Technology*, vol. 81, pp. 272–283, 2018.
- [13] G. L. O. Halila, J. R. R. A. Martins, and K. J. Fidkowski, “Adjoint-based aerodynamic shape optimization including transition to turbulence effects,” *Aerospace Science and Technology*, vol. 107, pp. 1–15, 2020.
- [14] X. He, J. Li, C. A. Mader, A. Yildirim, and J. R. R. A. Martins, “Robust aerodynamic shape optimization—from a circle to an airfoil,” *Aerospace Science and Technology*, vol. 87, pp. 48–61, 2019.

- [15] S. Skinner and H. Zare-Behtash, “State-of-the-art in aerodynamic shape optimisation methods,” *Applied Soft Computing*, vol. 62, pp. 933 – 962, 2018.
- [16] C. A. Mader and J. R. R. A. Martins, “Derivatives for time-spectral computational fluid dynamics using an automatic differentiation adjoint,” *AIAA Journal*, vol. 50, pp. 2809–2819, 2012.
- [17] I. M. Jonsson, L. Leifsson, S. Koziel, Y. A. Tesfahunegn, and A. Bekasiewicz, “Trawl-door shape optimization by space-mapping-corrected cfd models and kriging surrogates,” *Procedia Computer Science*, vol. 80, pp. 1061 – 1070, 2016.
- [18] S. Koziel, Y. A. Tesfahunegn, and L. Leifsson, “Variable-fidelity cfd models and co-kriging for expedited multi-objective aerodynamic design optimization,” *Engineering Computations*, vol. 33, 2016, pp. 2320-2338.
- [19] J. Nagawkar, J. Ren, X. Du, L. Leifsson, and S. Koziel, “Single- and multipoint aerodynamic shape optimization using multifidelity models and manifold mapping,” *Journal of Aircraft*, vol. 58, no. 3, pp. 591–608, 2021.
- [20] M. A. Bouhlef and J. R. R. A. Martins, “Gradient-enhanced kriging for high-dimensional problems,” *Engineering with Computers*, vol. 1, no. 35, pp. 157–173, 2019.
- [21] M. A. Bouhlef, N. Bartoli, A. Otsmane, and J. Morlier, “Improving kriging surrogates of high-dimensional design models by partial least squares dimension reduction,” *Structural and Multidisciplinary Optimization*, vol. 53, no. 5, p. 935–952, 2016.
- [22] X. Du and L. Leifsson, “Multifidelity modeling by polynomial chaos-based cokriging to enable efficient model-based reliability analysis of ndt systems,” *Journal of Nondestructive Evaluation*, vol. 39, no. 13, pp. 1–30, 2020.
- [23] N. V. Queipo, R. T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P. K. Tucker, “Surrogate-Based Analysis and Optimization,” *Progress in Aerospace Sciences*, vol. 41, no. 1, pp. 1–28, 2005.
- [24] N. Queipo, S. Pintos, and E. Nava, “Setting Targets in Surrogate-Based Optimization,” *Journal of Global Optimization*, vol. 55, no. 4, pp. 857–875, 2013.

- [25] B. Peherstorfer, K. Willcox, and M. Gunzburger, “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization,” *Society for Industrial and Applied Mathematics*, vol. 60, no. 3, pp. 550–591, 2018.
- [26] D. G. Krige, “A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand,” *Journal of the Chemical, Metallurgical and Mining Engineering Society of South Africa*, vol. 52, no. 6, pp. 119–139, 1951.
- [27] G. Matheron, “Principles of geostatistics,” *Economic Geology*, vol. 58, pp. 1246–1266, 1963.
- [28] J. Sacks, W. Welch, J. T. Michell, and P. H. Wynn, “Design and Analysis of Computer Experiments,” *Statistical Science*, vol. 4, pp. 409–423, 1989.
- [29] G. Blatman, “Adaptive Sparse Polynomial Chaos Expansion for Uncertainty Propagation and Sensitivity Analysis,” *Ph.D. Thesis, Blaise Pascal University - Clermont II. 3, 8, 9, France*, 2009.
- [30] A. J. Smola and B. Scholkopf, “A Tutorial on Support Vector Regression,” *Statistics and Computing*, vol. 14, pp. 199–222, 2004.
- [31] A. I. J. Forrester and A. J. Keane, “Recent Advances in Surrogate-Based Optimization,” *Progress in Aerospace Sciences*, vol. 45, no. 1-3, pp. 50–79, 2009.
- [32] C. M. Kennedy and A. O’Hagan, “Predicting the Output from a Complex Computer Code When Fast Approximations are available,” *Biometrika*, vol. 87, no. 1, pp. 1–13, 2000.
- [33] D. Echeverria and P. Hemker, “Manifold mapping: A two-level optimization technique,” *Computing and Visualization in Science*, vol. 11, pp. 193–206, 2008.
- [34] A. P. Singh, S. Medida, and K. Duraisamy, “Machine-learning-augmented predictive modeling of turbulent separated flows over airfoils,” *AIAA Journal*, vol. 55, no. 7, pp. 2215–2227, 2017.
- [35] A. P. Singh and K. Duraisamy, “Using field inversion to quantify functional errors in turbulence closures,” *Physics of Fluids*, vol. 28, pp. 045110–21, 2016.
- [36] J.-X. Wang, J.-L. Wu, and H. Xiao, “Physics-informed machine learning approach for reconstructing reynolds stress modeling discrepancies based on dns data,” *Phys. Rev. Fluids*, vol. 2, p. 034603, March 2017.

- [37] J.-L. Wu, H. Xiao, and E. Paterson, “Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework,” *Phys. Rev. Fluids*, vol. 3, p. 074602, July 2018.
- [38] J. C. Jouhaud, P. Sagaut, B. Enaux, and J. Laurenceau, “Sensitivity analysis and multiobjective optimization for les numerical parameters,” *Journal of Fluid Engineering*, vol. 130, no. 9-10, pp. 021401–021401–9, 2008.
- [39] S. Yarlanki, B. Rajendran, and H. Hamann, “Estimation of turbulence closure coefficients for data centers using machine learning algorithms,” in *13th IEEE Intersociety Conference on Thermal and Thermomechanical Phenomena in Electronic Systems*, San Diego, CA, 30 May - 1 June 2012.
- [40] J. Ray, S. Lefantzi, S. Arunajatesan, and L. Dechant, “Bayesian parameter estimation of a $k-\epsilon$ model for accurate jet-in-crossflow simulations,” *AIAA Journal*, vol. 54, no. 8, pp. 2432–2448, 2016.
- [41] J. Ray, S. Lefantzi, S. Arunajatesan, and L. J. DeChant, “Bayesian calibration of a $k - \epsilon$ turbulence model for predictive jet-in-crossflow simulations,” in *44th AIAA Fluid Dynamics Conference*, Atlanta, GA, 16-20 June 2014.
- [42] J. Ray, S. Lefantzi, S. Arunajatesan, and L. J. DeChant, “Bayesian calibration of a RANS model with a complex response surface-a case study with jet-in-crossflow configuration,” in *45th AIAA Fluid Dynamics Conference*, p. 2784, Dallas, TX, 22-26 June 2015.
- [43] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA: The MIT Press, 2016.
- [44] T. K. Ho, “Random decision forests,” in *Proceedings of the Third International Conference on Document Analysis and Recognition*, pp. 278–282, Montreal, Canada, 14-16 August 1995.
- [45] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [46] Y. Zhang, W. J. Sung, and D. N. Mavris, “Application of convolutional neural network to predict airfoil lift coefficient,” in *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Kissimmee, Florida, 8 - 12 January 2018.
- [47] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85 – 117, 2015.

- [48] W. M. Czarnecki, S. Osindero, M. Jaderberg, G. Świrszcz, and R. Pascanu, “Sobolev training for neural networks,” in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, 4-9 December 2017.
- [49] R. Maulik, H. Sharma, S. Patel, B. Lusch, and E. Jennings, “A turbulent eddy-viscosity surrogate modeling framework for reynolds-averaged navier-stokes simulations,” *Computers & Fluids*, vol. 227, p. 104777, 2021.
- [50] R. Maulik, H. Sharma, S. Patel, B. Lusch, and E. Jennings, “Deploying deep learning in openfoam with tensorflow,” in *AIAA Scitech 2021 Forum*, Virtual, 11-15 and 19-21 January 2021.
- [51] A. Forrester, A. Sobester, and A. Keane, *Engineering Design via Surrogate Modelling: A Practical Guide*. United Kingdom: John Wiley and Sons, Ltd., 2008.
- [52] M. D. McKay, R. J. Beckman, and W. J. Conover, “A comparison of three methods for selecting values of input variables in the analysis of output from a computer code,” *Technometrics*, vol. 21, no. 2, pp. 239–245, 1979.
- [53] J. D. Anderson, *Fundamentals of Aerodynamics*. McGraw Hill, New York, United States, 2007.
- [54] R. L. Panton, *Incompressible Flow*. John Wiley and Sons, New Jersey, United States, 2013.
- [55] P. R. Spalart and S. R. Allmaras, “A one equation turbulence model for aerodynamic flows,” in *30th AIAA Aerospace Sciences Meeting and Exhibit*, vol. 92-0439, Reno, NV, 6-9 January 1992.
- [56] H. K. Versteeg and W. Malalasekera, *An introduction to computational fluid dynamics: The finite volume method*. United Kingdom: Harlow, England: Pearson Education Ltd, 2007.
- [57] F. G. Schmitt, “About boussinesq’s turbulent viscosity hypothesis: historical remarks and a direct evaluation of its validity,” *Comptes Rendus Mécanique*, vol. 335, no. 9, pp. 617–627, 2007.
- [58] A. A. Townsend, *The structure of turbulent shear flow*. United States: Cambridge New York: Cambridge University Press, 1976.

- [59] D. Opitz and R. Maclin, “Popular ensemble methods: an empirical study,” *Journal of Artificial Intelligence Research*, vol. 11, pp. 169–198, 1999.
- [60] J. R. Quinlan, “Simplifying decision trees,” *International Journal of Man-Machine Studies*, vol. 27, no. 3, pp. 169–198, 1987.
- [61] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning with Applications in R*. Springer Science+Business Media, New York, United States, 2013.
- [62] D. M. Driver and H. L. Seegmiller, “Features of a reattaching turbulent shear layer in divergent channel flow,” *AIAA Journal*, vol. 23, no. 2, pp. 163–171, 1985.
- [63] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby, “A tensorial approach to computational continuum mechanics using object-oriented techniques,” *Computers in Physics*, vol. 12, no. 6, pp. 620–631, 1998.
- [64] G. Farin, *Curves and surfaces for computer aided geometric design: A practical guide*. San Diego, CA: Academic Press Professional, Inc., 1988.
- [65] N. Secco, G. K. W. Kenway, P. He, C. A. Mader, and J. R. R. A. Martins, “Efficient mesh generation and deformation for aerodynamic shape optimization,” *AIAA Journal*, vol. 59, no. 4, pp. 1151–1168, 2021.
- [66] C. L. Ladson, *Effects of independent variation of Mach and Reynolds numbers on the low-speed aerodynamic characteristics of the NACA 0012 airfoil section*. NASA TM 4074, Washington, D.C, 1988.