

# Multifidelity Aerodynamic Flow Field Prediction Using Random Forests

Jethro Nagawkar\*, Marc Brittain†, and Leifur Leifsson‡

*Iowa State University, Ames, Iowa, 50011, USA*

A random forest (RF)-based framework to predict the high-fidelity Reynolds-averaged Navier-Stokes (RANS) flow field, namely, the pressure, velocity and turbulent viscosity, is proposed in this work. In this framework, RF are used to increase the fidelity of a low-fidelity flow field generated using a potential flow solver. Two cases are studied, both consisting of a flow past a backward facing step. In the first case, the data is generated using ten different inlet velocities, and in the second, using six different step heights. The input parameters to the RF include the x and y cell-center locations, the corresponding x and y potential flow velocities, and the inlet velocity to the domain, for the first case. For the second case, the cell-center locations are normalized using the step height, and the step height is used in place of the inlet velocity as input parameters. The outputs of the RF are the RANS velocities, pressures, and turbulent viscosities at the corresponding x and y cell-center locations. The results from the RF framework are compared to those generated using the TensorFlowFoam (TFF) framework and from directly solving the RANS equations. The errors are quantified using both the absolute and relative  $L_2$  norm errors at different locations and for different flow quantities. RF consistently shows lower errors compared to TFF, with the exception of viscosity ratios for the second case. The relative  $L_2$  norm errors are two to 30 times lower for RF compared to TFF depending on the field variable and location. RF is also takes two to three times less time to train than TFF and takes about a second to predict the entire flow field.

## Nomenclature

$C_f$	= skin friction coefficient, $\frac{\tau_w}{q_\infty}$
$q_{ref}$	= reference dynamic pressure, $0.5\rho_{ref} V_{ref}^2$ , [N/m <sup>2</sup> ]
$Re_H$	= Reynolds number based on the step height, $\frac{U_{inlet}H}{\nu}$ , [-]
$H$	= step height, [m]
$h$	= baseline step height, [m]
$m$	= total number of samples
$N$	= number of samples
$n$	= number of subsets
$n_h$	= number of high-fidelity flow field variables
$U_{inlet}$	= domain inlet velocity, [m/s]
$V_{ref}$	= reference air velocity, [m/s]
$\mathbf{X}$	= sample points of the design variables
$\mathbf{x}$	= design variables used to make RF-based flow field prediction
$\mathbf{f}_{LF}$	= low-fidelity flow field variables
$\mathbf{f}_{HF}$	= high-fidelity flow field variables
$\hat{\mathbf{f}}_{HF}$	= RF-based flow field prediction
$x$	= x coordinate of the domain, [m]
$x_c$	= x coordinate of the grid cell-center, [m]
$y$	= y coordinate of the domain, [m]
$y_c$	= y coordinate of the grid cell-center, [m]

\*Ph.D. Student, Department of Aerospace Engineering, AIAA Student Member.

†Ph.D. Candidate, Department of Aerospace Engineering, AIAA Student Member.

‡Associate Professor, Department of Aerospace Engineering, AIAA Senior Member.

$p_r$	= RANS pressure in the grid cell-center, [N/m <sup>2</sup> ]
$u_p$	= 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]
$v_p$	= 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]
$x_{RAL}$	= boundary layer reattachment length, [m]
$y^+$	= non-dimensionalized first layer cell thickness, [-]
$\nu$	= kinematic viscosity, [m <sup>2</sup> /s]
$\nu_t$	= turbulent viscosity, [m <sup>2</sup> /s]
$\rho_{ref}$	= reference air density, [kg/m <sup>3</sup> ]
$\tau_w$	= Wall shear stress, [N/m <sup>2</sup> ]

## I. Introduction

Aerodynamic design optimization (ADO), computational fluid dynamics (CFD), and machine learning (ML) are increasing becoming an intrinsic part in the design of complex physical systems in the aerospace, automotive, and energy sectors. While these three fields were initially developed separately, recently there has been an increasing effort to couple these systems together to find optimum designs [1, 2], perform uncertainty analysis [3, 4] and solve inverse problems [5, 6].

ADO methods have traditionally relied on expensive high-fidelity simulations to calculate both the cost and constraint function values [7-9]. These methods also 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.

To overcome these challenges, several surrogate modeling techniques have been introduced [1, 10, 11]. These techniques can be classified into two types: data-fit methods [12] and multifidelity methods [13]. Data-fit methods include fitting a response surface through the evaluated cost function values at sampled points in the design space. Kriging [14], polynomial chaos expansions [15], and neural networks (NN) [16] are some examples of data-fit methods. Multifidelity surrogate modeling [13] methods use information from multiple levels of fidelity. Low-fidelity data is used to provide information on the trends to a relatively small number of high-fidelity data samples. Cokriging [17] and manifold mapping [18] are some examples of this method.

The use of ML to increasing the fidelity of Reynolds-averaged Navier-Stokes (RANS) models have increasingly become popular [19-22]. A field inversion ML method to construct a deficiency ( $\beta$ ) term from high-fidelity large eddy simulation data was developed by Singh et al. [19, 20]. This term was then used in one of the two additional transport equations of a two-equation RANS model. A NN was then used to construct a functional form of  $\beta$  from a set of non-dimensional flow features. Cases that were not trained using the NN showed improvements. Wang et al. [21] developed a physics informed ML framework. Mean flow features were used as inputs to a random forest (RF) [23] and the outputs the discrepancy of Reynolds stresses between the RANS model and direct numerical simulations. This discrepancy term was embedded into the RANS model and the resulting RANS model showed significant improvements in field prediction. While the methods above are promising, they involve solving the RANS equations every time a prediction is needed to be made.

Some researchers [1, 2] tried to directly predict the quantities of interest instead of solving the governing equations each time. Gradient-enhanced Kriging with partial least squares [1] was used by Li et al. [2] to predict the drag, lift and pitching moment coefficient of different airfoils at varying flow conditions. A similar study was performed by Bouhlel et al. [1], but gradient-enhanced NN [24] was used to predict the force coefficients for an airfoil. The data used in these approaches consisted of values generated in both subsonic and transonic flow regimes. The major drawback of these methods is that they require a large amount of data to train the algorithms used, typically in the order of  $10^3$  to  $10^5$ .

The TensorFlowFoam (TFF<sup>†</sup>) framework was developed by Maulik et al. [25, 26] to predict the turbulent viscosity obtained from data generated using RANS models. Flow past a backward facing step was used as their test case. Data for the first case was generated using ten different inlet velocities, while for the second, using six different step heights. The NN is then tested on cases with two untrained inlet velocities and step heights, respectively. In order to predict the flow field for the untrained cases, they solve the continuity and momentum equations in RANS and use the turbulent viscosity predicted by the NN, instead of solving the additional closure equations. This reduced the simulation cost by a

---

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

factor of five [25]. In that study [25], each grid cell provided one data sample used in the training of the NN. Since each CFD simulation had approximately  $10^5$  grid cells, the total number of CFD simulations required to accurately train the NN was small.

This work is motivated by Maulik et al. [25]. Instead of just predicting the turbulent viscosity, the velocity and pressure fields are predicted as well. Furthermore, the RF [27] ML algorithm is used in place of NN [16]. The results from this work are compared to TFF for the same two cases used by Maulik et al. [25] as well as RANS CFD simulations. Note that the two cases are redone using TFF and with data we generated.

The next section describes the method used to predict the flow field using RF. The following section describes the application of this RF framework two cases. This framework is benchmarked against the TFF framework and is also compared to RANS CFD simulations. Lastly, the conclusions and future work are presented.

## II. Methods

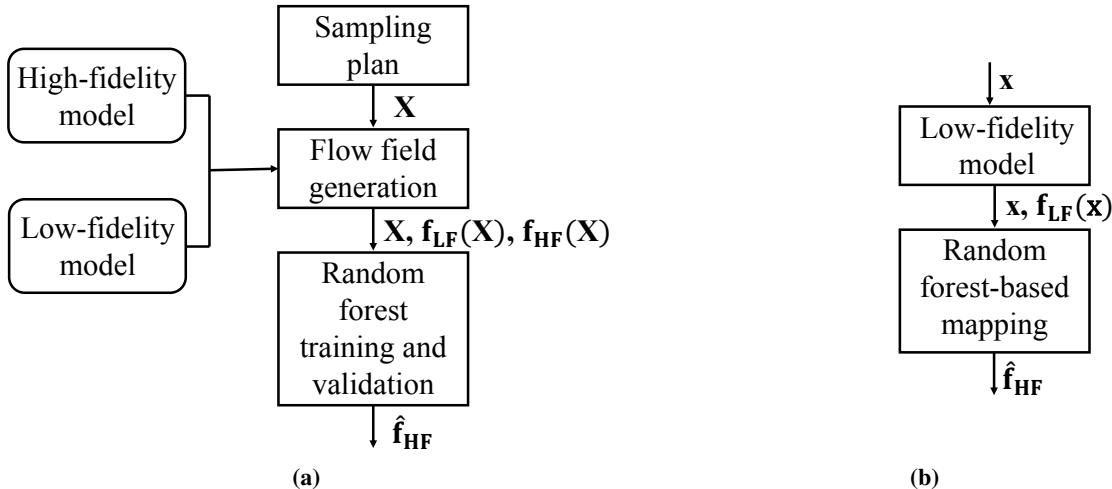
This section outlines the methods used to construct the RF-based flow field prediction framework. The section begins with the workflow, followed by the sampling plan, a description of the RF used, the CFD setup and validation, the validation metric used, and finally ends with the RF-based flow field prediction.

### A. Workflow

A flowchart of the proposed RF-based flow field prediction framework is shown in Fig. 1. It consists of two parts, the offline part (Fig. 1(a)) and the online part (Fig. 1(b)). In the offline part, the RF is trained and validated, while the online part, the trained RF is used to make flow field predictions, given the low-fidelity flow field data and the design parameters. The process for the offline part starts by sampling the design space. Both low and high-fidelity models are run for sampled designs to generate the data required to train and validate the RF. The RF is trained using 90% of the this data, while 10% is used to validate the RF. The is done in order to tune the hyperparameters in order to improve the predictive capabilities of the RF. The mean squared error metric is used to quantify the training and validation dataset accuracy. Once the hyperparameters are tuned, the RF is retrained with both the training and validation data, and the RF is now ready to be used in the online part. In this part, the low-fidelity model is run on previous unsampled design parameters, and the data generated is used as inputs to the trained RF to make high-fidelity flow field predictions.

### B. Sampling plan

Sampling is the process of selecting discrete samples in the design space [28]. It is an iteration-based process in which the design parameters are randomly drawn from probability distributions assigned to the parameters. In this work, the full-factorial sampling [28] plan is used to generate the design parameters. This is the same sampling plan used by



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

Maulik et al. [25]. 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.

### C. Random forest

Random forest [23, 27] is an ensemble ML method [29] which can be used for both regression and classification tasks and is constructed using a multitude of decision trees [30]. The output predictor space is segmented by both decision trees [30] and random forests [27]. The mean or mode value of the training observation in which the new observation belongs to is used as its predicted value, for a new observation.

A schematic of a random forest with two decision trees is shown in Fig. 2. Decision trees consist of a root node, followed by several intermediate nodes, and terminates on the leaf. The root node of a decision tree divides the entire training dataset into smaller datasets or subsets. Each individual input feature and a splitting criterion such as a mean or mode of the input feature value is iterated over and the combination which results in the lowest weighted mean squared error over all the subsets is used as the criterion to split the data at the root node [31]. The weighted mean squared error is given by

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

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

$$\hat{f}_{HF,j}^{(k)} = \frac{\sum_{i=1}^{N^{(k)}} f_{HF,j}^{(i)}}{N^{(k)}}. \quad (2)$$

The intermediate nodes perform the same task as the root node, but with data from a given subset created from the previous intermediate or root node. Again, the subset is split into a smaller subset. This process continues with smaller and smaller subsets till the leaf node is reached. Hyperparameters such as minimum number of samples in a subset required to split it, and maximum tree depth can be set to decide whether a leaf node is reached. The leaf node returns the predicted value from the decision tree. This value is the averaged value of all the samples in the subset of the leaf node.

The main advantages of decision trees include ability to handle large amounts of data, interpretability, and ease of use [31]. This makes it attractive to use for various ML tasks. Its drawbacks, however, include high sensitivity to the

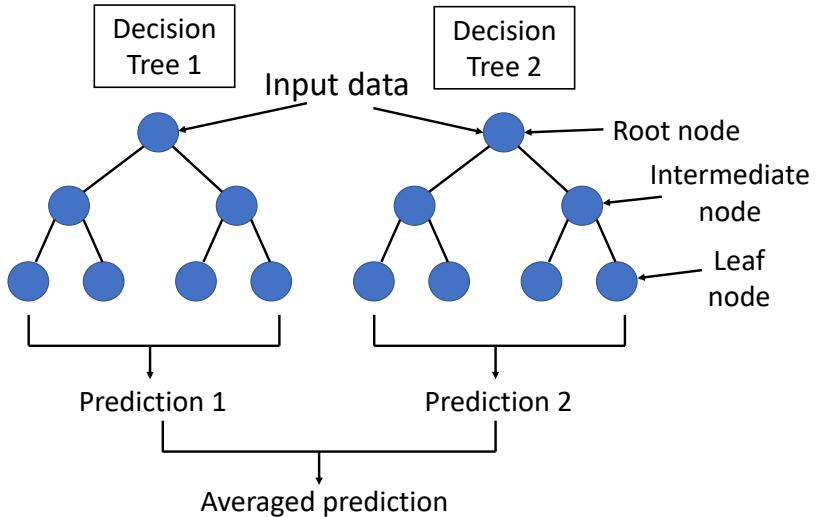


Fig. 2 A schematic of a random forest.

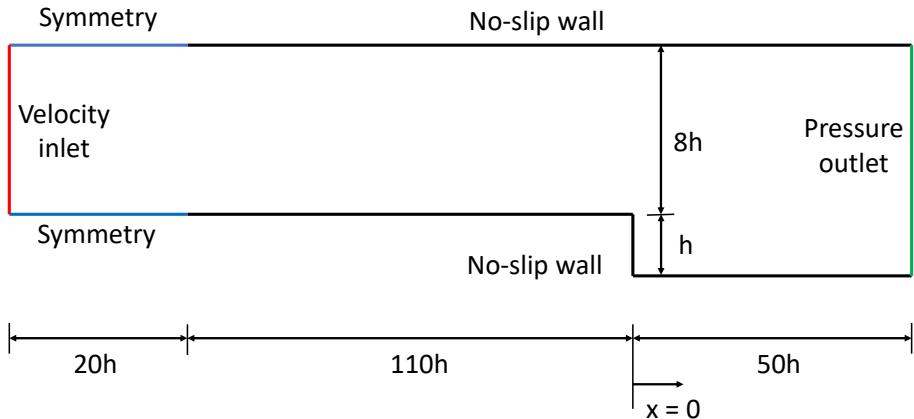
training dataset, and over-fitting [31]. Random forest overcomes these problems by averaging the outputs from multiple decision trees, which is set using the number of trees hyperparameter, and in order to construct each individual decision tree, data from the training dataset is randomly selected with replacement and fed into the decision tree.

#### D. CFD setup and validation

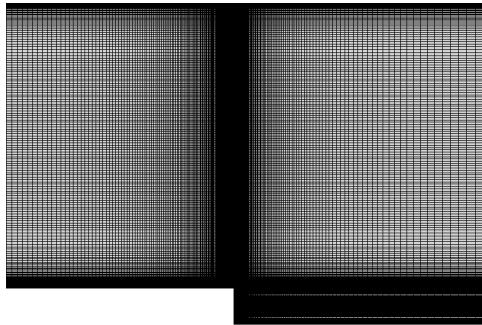
In this work, the flow past a backward facing step is simulated. The domain of the geometry used in this study is shown in Fig. 3. The step height  $h$  is fixed at  $12.7\text{ mm}$  and the remaining domain is scaled as per the dimensions shown in Fig. 3. The inlet velocity to the domain is set to a value of  $44.2\text{ m/s}$ . The domain dimensions as well as the inlet velocity are set in accordance to experimental data from David and Seegmiller [32], in order to validate the CFD setup.

This work uses blockMesh in OpenFOAM version 5.0 [33] to generate the mesh and is shown in Fig. 4. Near the wall boundaries, the mesh is refined to ensure that the first cell thickness is less than a  $y^+$  value of one. A grid independent study is performed and is shown in Table I.

OpenFOAM version 5.0 [33] is used to simulate both the high and low-fidelity flow. The potentialflowFoam and simpleFoam solvers are used to simulate the low-fidelity potential flow field and the high-fidelity Spalart-Allmaras [34] RANS turbulence model flow field, respectively. Figure 3 shows the boundary conditions set for this case. The outlet has a zero pressure gradient value, and the viscosity ( $\nu$ ) of the fluid used has a value of  $1.56 \times 10^{-5}\text{ m}^2/\text{s}$ . The convergence criteria set is pressure and velocity residuals fall to a value below  $10^{-6}$  for the RANS simulations. Table I shows the simulation time as well as the reattachment length ( $x_{RAL}$ ) of boundary layer downstream of the step for different mesh sizes. On refining the mesh, the reattachment length from the CFD simulations closer to the experimental value. For this study, mesh L1 is used for both the cases.



**Fig. 3** Domain and boundary conditions of the backward facing step.



**Fig. 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 [32]	-	6.26h	-

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

## E. Validation

This work uses 90% of the data to train the RF, and 10% to validate its accuracy. The mean squared error (MSE) metric is used to measure the training and validation dataset accuracy, and is averaged over all the output predictions and is given by

$$\mathcal{L} = \frac{\sum_{i=1}^N \sum_{j=1}^{n_h} (\hat{f}_{HF,j}^{(i)} - f_{HF,j}^{(i)})^2}{N n_h}, \quad (3)$$

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

## F. Random forest-based flow field prediction

Once trained, the RF can be used to increase the fidelity of the low-fidelity potential flow field to make high-fidelity RANS field prediction. In order to do so, the potential flow field variables and the design variable are fed into the trained RF and the corresponding RANS field is reconstructed. In this study, the absolute error as well as the relative  $L_2$  norm errors are measured at different locations in the domain, in order to quantify the accuracy of the predicted RF flow field. The absolute error is given by

$$L_{abs,j} = |\hat{f}_{HF,j} - f_{HF,j}|, \quad (4)$$

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

$$L_{2,rel,j} = \frac{\|\hat{\mathbf{f}}_{HF,j} - \mathbf{f}_{HF,j}\|_2}{\|\mathbf{f}_{HF,j}\|_2}. \quad (5)$$

Engineering outer-loop problems, such as aerodynamic shape optimization, uncertainty quantification and propagation as well as global sensitivity analysis, typically require multiple and repetitive high-fidelity physics-based simulations. This makes it challenging to use for different engineering applications. Replacing these high-fidelity simulations with the RF-based flow field prediction reduces this cost dramatically.

## III. Numerical Examples

This section presents the two cases used to demonstrate the RF-based flow field prediction framework. Both these cases involves a flow past a backward facing step. The results are compared to both the TFF framework as well as CFD simulations.

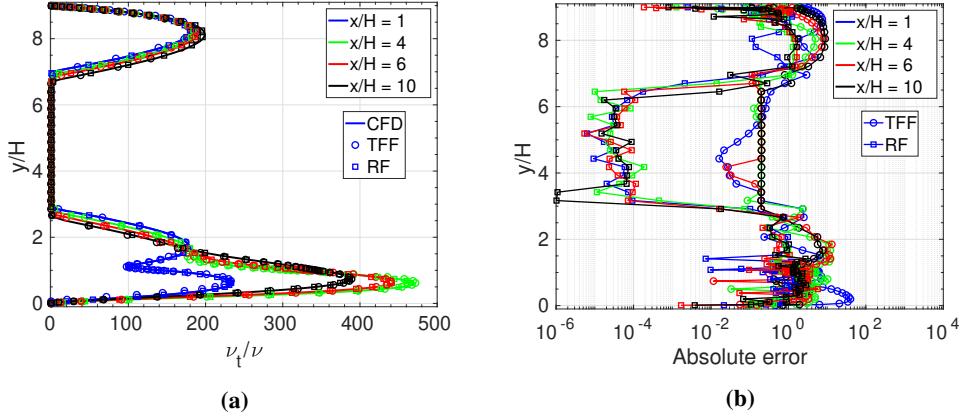
### A. Case I

In this case, the flow past a backward facing step is simulated. The step height,  $H = 1.0h$ , 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  $\nu$  is the kinematic viscosity. The  $Re_H$  for this case varies between 34,000 and 41,500, based on the selected  $U_{inlet}$  values.

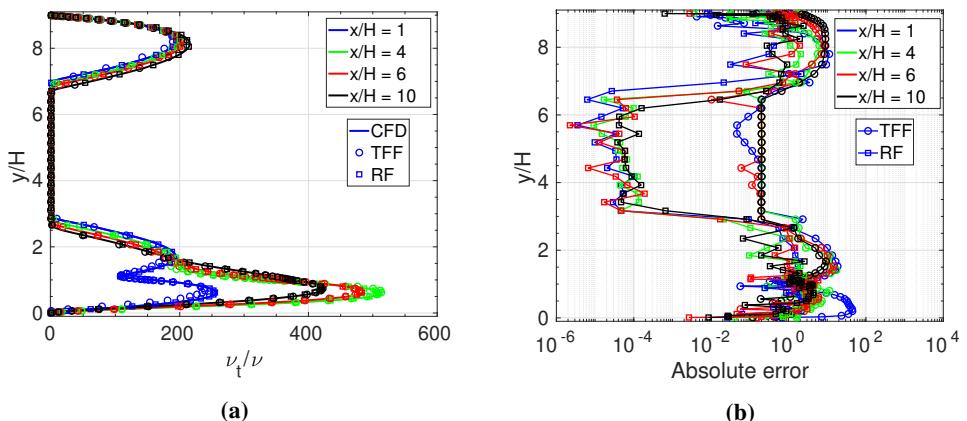
In this case, the high-fidelity RANS simulations are replaced with a RF that predicts the RANS flow field, namely, velocity ( $u_r$  and  $v_r$ ), pressure ( $p_r$ ) and turbulent viscosity ( $\nu_t$ ), in the domain. The inputs to the RF are the cell-center locations of the grid,  $x_c$  and  $y_c$ , respectively, the velocities at these grid locations from a low-fidelity potential flow solver, given by  $u_p$  and  $v_p$ , respectively, and  $U_{inlet}$ . 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. The RF is then tested on two unseen  $U_{inlet}$  of 44.2 m/s and 48.5 m/s, respectively. The results are then compared to those from RANS CFD simulations and TFF.

The viscosity ratios for TFF, RF, and CFD are shown in Figs. 5(a) and 6(a) for the  $U_{inlet}$  of 44.2 m/s and 48.5 m/s, respectively, and at different downstream locations. The corresponding absolute errors are shown in Figs. 5(b) and 6(b), respectively. The viscosity ratio trend for both these cases are similar, with RF having a lower absolute error when compared to TFF. At the midsection of this domain, with midsection referring to  $y/H$  values between 3 and 5, RF has the lowest absolute error. This error is around three orders of magnitude lower compared to TFF. In the remaining part of the domain, RF is still better at predicting the viscosity ratios than TFF. However, this error is around four orders of magnitude higher than the midsection. The  $L_{2,rel}$  of the viscosity ratio at various downstream locations for this case is shown in Table 2. RF has approximately half the  $L_{2,rel}$  compared to TFF, with the exception of  $x/H = 1$ , where this error is an order of magnitude lower for RF than for TFF.

Figures 7 and 8 show the velocity ratios and its absolute errors at selected downstream locations, for the two selected  $U_{inlet}$ . RF again outperforms TFF in predicting the velocity ratios. However, unlike the viscosity ratios, the absolute



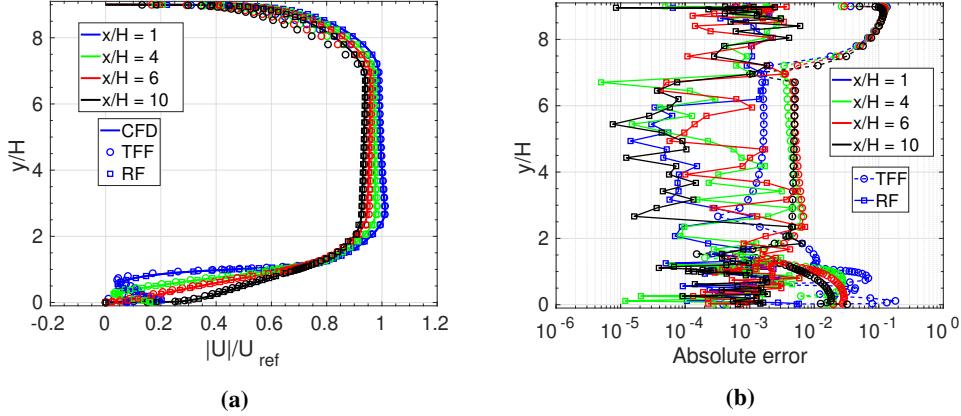
**Fig. 5 Case I viscosity results at different  $x/H$  locations for inlet velocity of 44.2 m/s: (a) viscosity ratio values, and (b) absolute errors.**



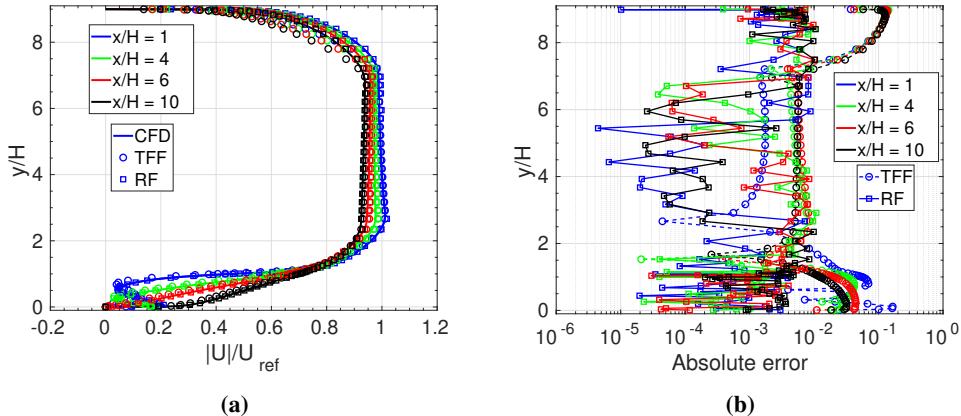
**Fig. 6 Case I viscosity results at different  $x/H$  locations for inlet velocity of 48.5 m/s: (a) viscosity ratio values, and (b) absolute errors.**

**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



**Fig. 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.**



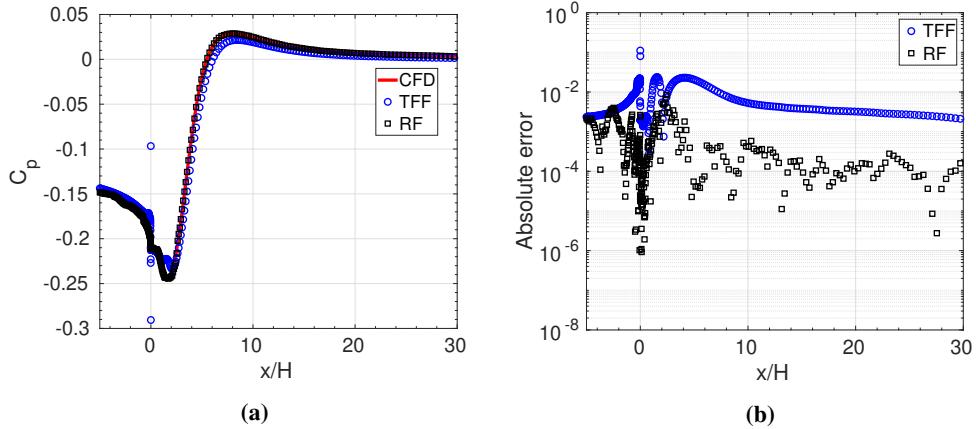
**Fig. 8 Case I velocity results at different  $x/H$  locations for inlet velocity of 48.5 m/s: (a) velocity ratio values, and (b) absolute errors.**

errors in velocity ratios for RF fluctuate by approximately three orders of magnitude, from lower to upper walls. The  $L_{2,rel}$  for the velocity ratios for RF is more than a magnitude smaller than TFF and is shown in Table 3.

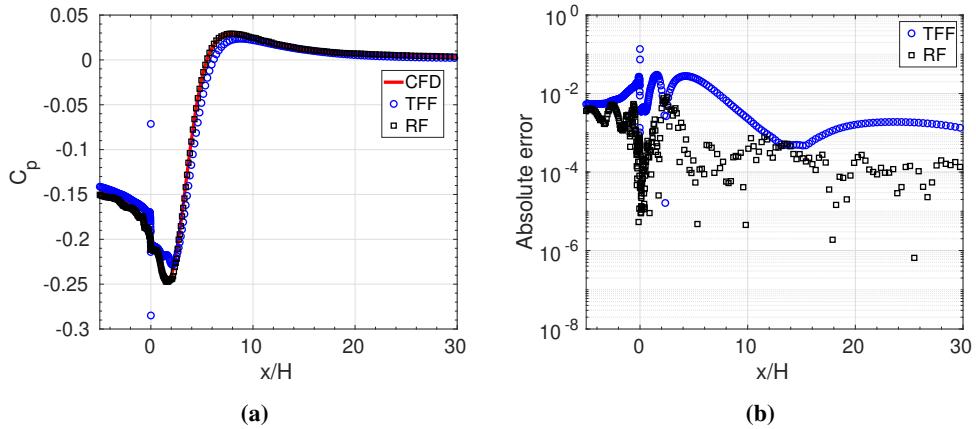
RF is better as at predicting the pressure and skin friction coefficients on the lower wall of the backward facing step for the different  $U_{inlet}$ . This is shown in Figs. 9, 10, 11 and 12, respectively. The pressure coefficient profile from RF matches the CFD results well, while small differences can be seen using TFF (Figs. 9 and 10). The skin friction coefficient profile too from RF matches the CFD results well as seen in Figs. 11 and 12. The skin friction values are heavily affected by the turbulent viscosity values near the wall. Near the walls, the turbulent viscosity is in the order of

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

Model	$U_{inlet}, m/s$	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
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



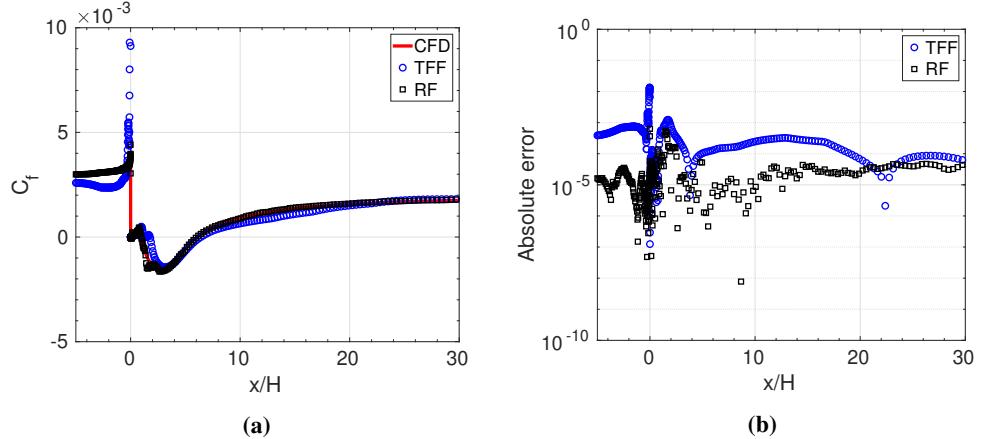
**Fig. 9 Case I pressure results on bottom wall for inlet velocity of 44.2 m/s: (a) pressure coefficient values, and (b) absolute errors.**



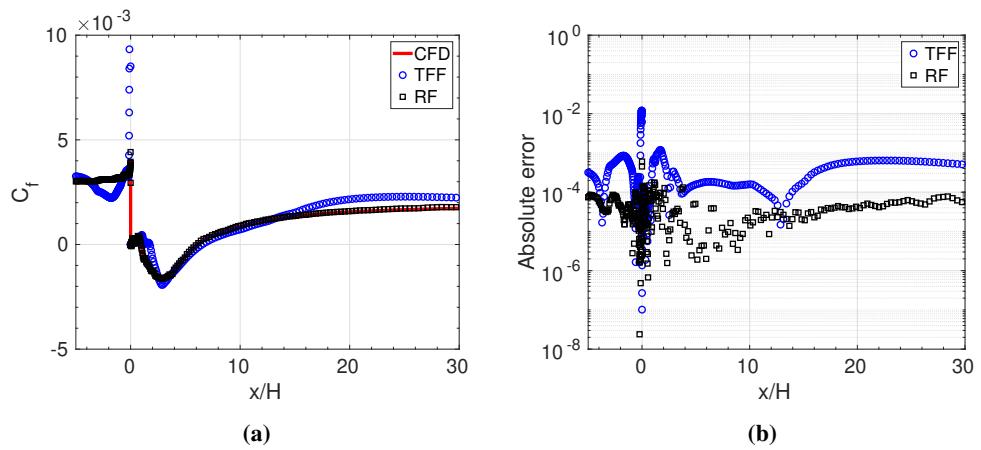
**Fig. 10 Case I pressure results on bottom wall for inlet velocity of 48.5 m/s: (a) pressure coefficient values, and (b) absolute errors.**

magnitude of  $10^{-8}$ , while the maximum turbulent viscosity is in the order of magnitude of  $10^{-3}$ . Small changes in this viscosity values near the wall can heavily impact the skin friction values. It is therefore crucial for the ML algorithms to accurately predict the viscosity values near the walls of the domain. Table 4 shows the  $L_{2,rel}$  for both the pressure and skin friction coefficients. The pressure coefficients for RF is slightly less than one order of magnitude lower than TFF, while for the skin friction coefficients, it is more than a magnitude smaller than TFF.

The computational cost involved in Case I is listed in Table 5. Since the same data is used for both TFF and RF, the data accumulation time is the same. RF is faster to train than TFF. RF takes about 1 second to predict the entire flow



**Fig. 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.



**Fig. 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.

field. TFF, however, requires significantly more time for prediction due to the need to solve the continuity and moment equations. This shows the enormous benefits of using RF in predicting the flow fields.

**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

**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

## B. Case II

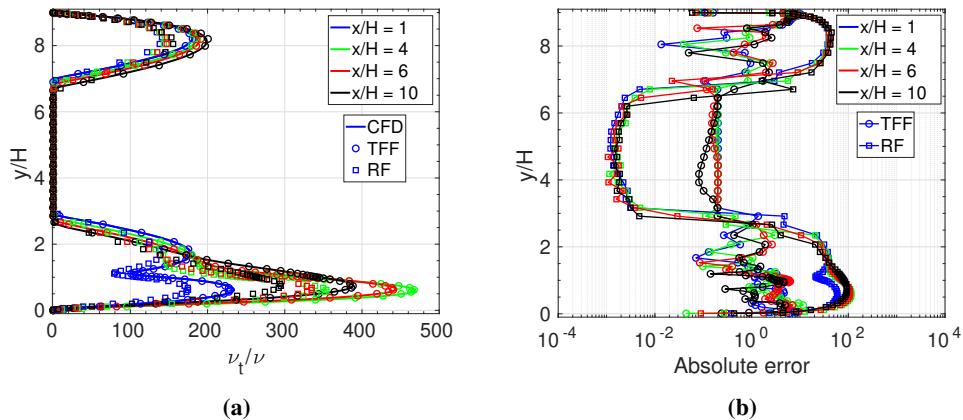
For this case, the flow past a backward facing step is simulated as well. However, instead of varying  $U_{inlet}$ ,  $H$  is varied. The remaining domain is also scaled accordingly as shown in Fig. 3. The  $Re_H$  for this case varies between 18,000 and 72,000, based on the selected  $H$  values.

The objective of this case is the same as the previous case, however, instead of predicting the flow field for a given  $U_{inlet}$ , it is predicted for a given  $H$  with a fixed  $U_{inlet}$  of 44.2 m/s. The inputs to the RF for this case include  $x_c/H$ ,  $y_c/H$ ,  $u_p$ ,  $v_p$ , and  $H$ . The RF is trained with data from six different step heights, namely,  $H = 0.5h, 0.75h, 1.25h, 1.5h, 1.75h, 2h$ , respectively. The RF is then tested with step heights 1.0h and 1.9h. The results are then compared to those from RANS CFD simulations and TFF.

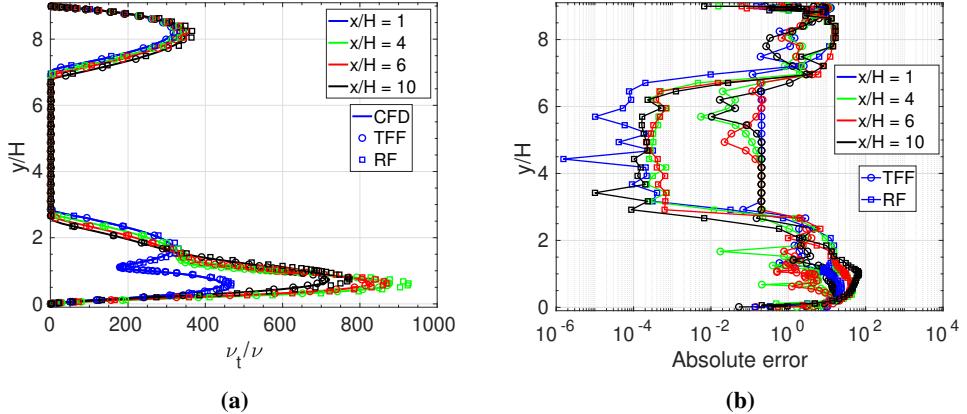
Figure 13 shows the viscosity ratio plots for  $H$  of 1.0h at different downstream locations. At the midsection, RF has approximately two orders of magnitude lower absolute errors compared to TFF, while in the remaining sections, TFF has two order of magnitude lower errors than RF. For the step height of 1.9h, similar trends can be seen in the midsection as seen in Fig. 14. However, in the remaining section, the difference in absolute errors between RF and TFF is lower compared to a step height of 1.0h, with RF still having higher errors than TFF. TFF has a lower  $L_{2,rel}$  when compared to RF. Between the two  $H$ , however,  $H$  of 1.9h has a significantly lower  $L_{2,rel}$  than 1.0h, for RF. This is due to the lack of training data in the vicinity of 1.0h when compared to 1.9h.

Similar to the previous case, RF is better at predicting the velocity ratios than TFF as seen in Figs. 15 and 16. This can also be seen in the lower  $L_{2,rel}$  for RF in Table 7. TFF is especially poor at prediction the velocities in the region between  $y/H = 7$  and the upper wall, as well as between  $y/H = 3$  and the lower wall.

The pressure coefficient trends on the lower wall predicted by RF is more accurate compared to TFF as seen in Figs. 17 and 18. Similar results can be seen for the skin friction coefficient profiles in Figs. 19 and 20. For both the  $H$ , RF is



**Fig. 13 Case II viscosity results at different  $x/H$  locations for step height of 1.0h: (a) viscosity ratio values, and (b) absolute errors.**



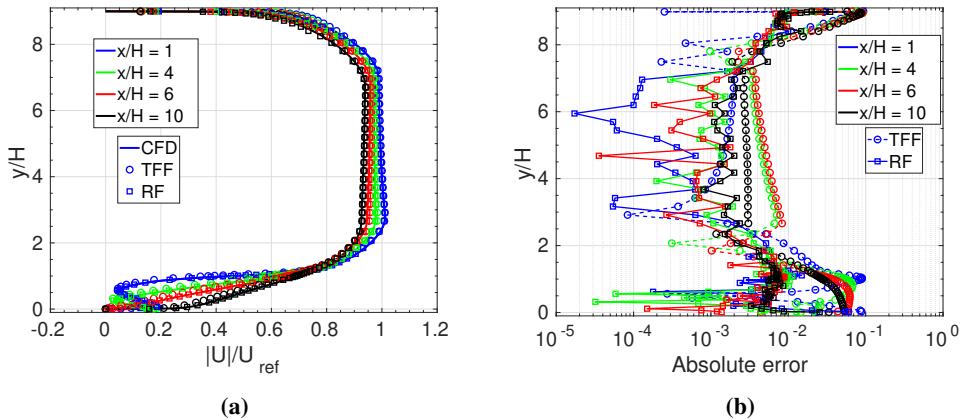
**Fig. 14** Case II viscosity results at different  $x/H$  locations for a step height of  $1.9h$ : (a) viscosity ratio values, and (b) absolute errors.

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

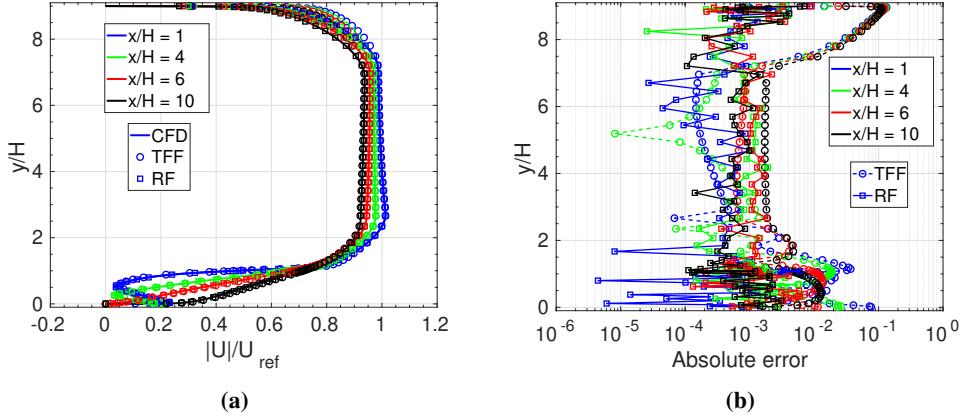
Model	H	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	$1.0h$	0.032	0.018	0.017	0.014
RF	$1.0h$	0.236	0.238	0.234	0.233
TFF	$1.9h$	0.038	0.014	0.015	0.023
RF	$1.9h$	0.050	0.051	0.049	0.082

able to capture the pressure coefficient profile well, however, for the skin friction profile,  $H$  of  $1.9h$  is captured better than  $1.0h$ . This is due to the lack of training data in the vicinity of  $1.0h$ , as mentioned above. Table 8 shows that RF has consistently lower  $L_{2,rel}$  than TFF.

The computational cost involved in Case II is listed in Table 9. 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, while TFF takes about 70 to 80 seconds for the same.



**Fig. 15** Case II velocity results at different  $x/H$  locations for a step height of  $1.0h$ : (a) velocity ratio values, and (b) absolute errors.



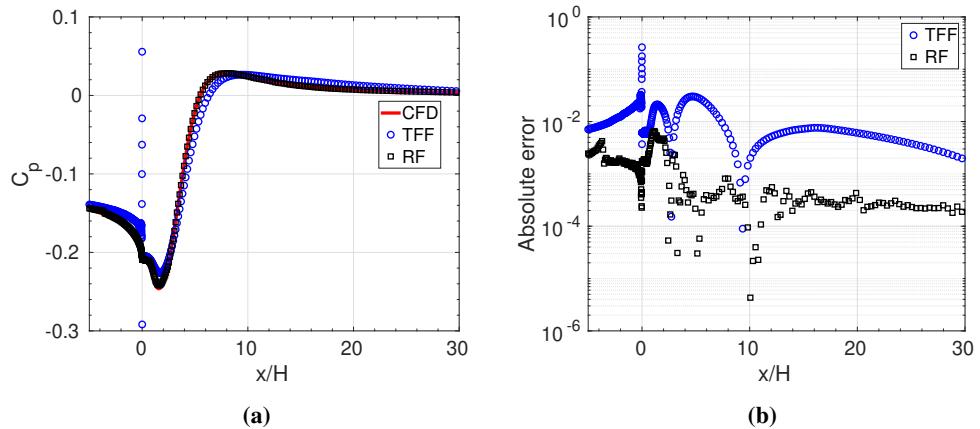
**Fig. 16** Case II velocity ratio at different  $x/H$  locations for a step height of  $1.9h$ : (a) velocity ratio values, and (b) absolute errors.

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

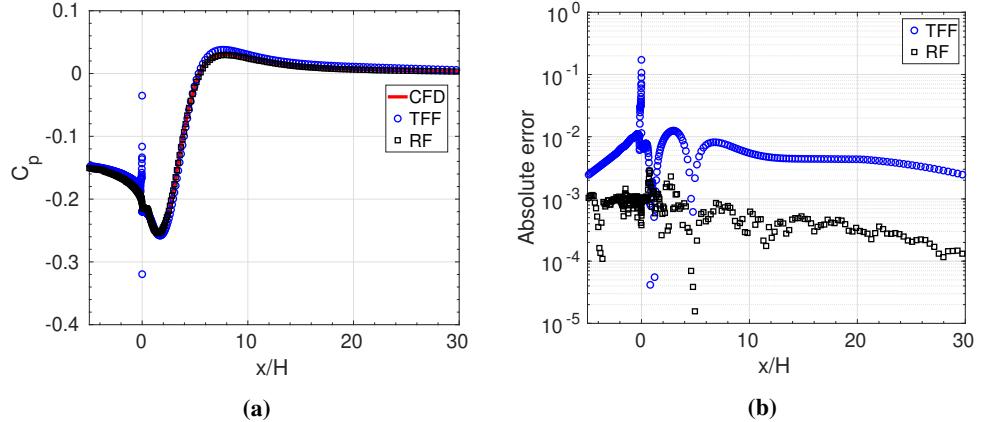
Model	H	$x/H = 1$	$x/H = 4$	$x/H = 6$	$x/H = 10$
TFF	$1.0h$	0.063	0.055	0.056	0.050
RF	$1.0h$	0.016	0.010	0.010	0.011
TFF	$1.9h$	0.065	0.061	0.059	0.056
RF	$1.9h$	0.002	0.002	0.003	0.002

#### IV. Conclusion

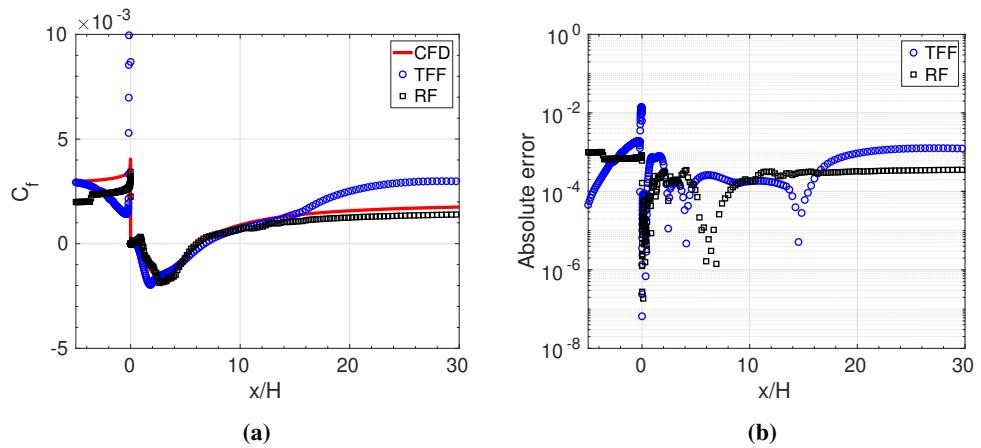
The multifidelity RF ML framework is presented in this study. Two different cases, both with flow past a backward facing step is studied. In the first case, the inlet velocity to the domain is varied, while for the latter case, the step height is varied. The low- and high-fidelity models used are the potential flow and the Spalart-Allmaras RANS model, respectively. The results from this framework is compared to the TFF framework as well as RANS CFD simulations. The errors from both these frameworks are quantified using the absolute and relative  $L_2$  norm errors.



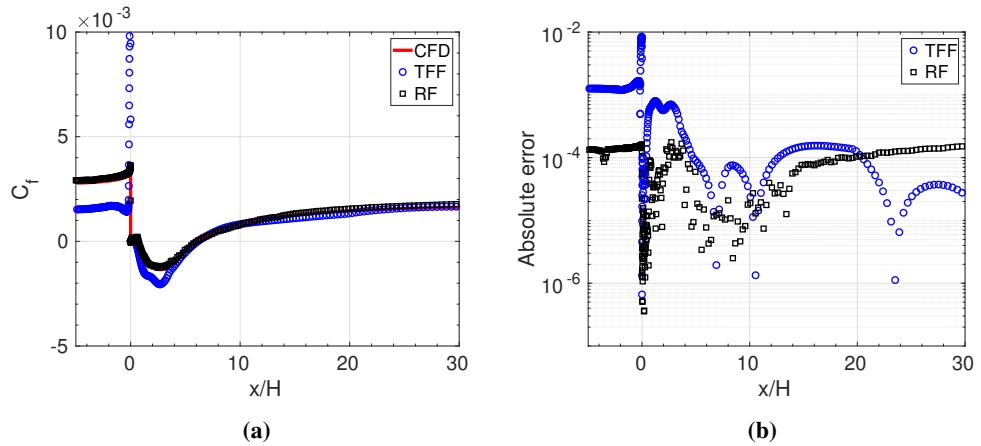
**Fig. 17** Case II pressure results on bottom wall for a step height of  $1.0h$ : (a) pressure coefficient values, and (b) absolute errors.



**Fig. 18** Case II pressure results on bottom wall for a step height of  $1.9h$ : (a) pressure coefficient values, and (b) absolute errors.



**Fig. 19** Case II skin friction results on bottom wall for a step height of  $1.0h$ : (a) skin friction coefficient values, and (b) absolute errors.



**Fig. 20** Case II skin friction results on bottom wall for a step height of  $1.9h$ : (a) skin friction coefficient values, and (b) absolute errors.

**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

**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

For the first case, ten different inlet velocities are used to generate the data to train both RF and TFF. For the second case, six different step heights are used. The inputs to the RF 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 RF algorithm. The output for both these cases are the same and include the RANS velocities, pressures, and turbulent viscosities. For these two cases, RF outperformed TFF in predicting the various flow quantities, with the exception of the viscosity ratios for the second case. RF also was cheaper to train on the given data and predict the flow field compared to TFF.

This study does not investigate the choice of input variables to the ML algorithms. The choice of these variables significantly affects the predictive performance of the ML algorithms. A thorough investigation of these variables is needed. Furthermore, this study does not investigate the number of design variables needed in order to accurately predict the flow field. This needs to be done in order to quantify the appropriate number of CFD simulations needed to generate the training data. Currently, only a flow past a backward facing step is studied. Other problems need to be considered. In addition, the online half of this framework needs to be developed and applied to problems involving global sensitivity analysis, uncertainty quantification and propagation, and aerodynamic design optimization.

### Acknowledgements

The first and last authors acknowledge the support of the National Science Foundation award No. 1846862. The second author is partially funded by the National Science Foundation under Award No. 1846862 and the NASA Iowa Space Grant under Award No. NNX16AL88H.

### References

- [1] Bouhlel, M. A., He, S., and Martins, J., “Scalable gradient-enhanced artificial neural networks for airfoil shape design in the subsonic and transonic regimes,” *Structural and Multidisciplinary Optimization*, 2020. doi:10.1007/s00158-020-02488-5.
- [2] Li, J., Bouhlel, M. A., and Martins, J. R. R. A., “Data-Based Approach for Fast Airfoil Analysis and Optimization,” *AIAA Journal*, Vol. 57, No. 2, 2019, pp. 581–596. doi:10.2514/1.J057129.
- [3] Feldstein, A., Lazzara, D., Princen, N., and Willcox, K., “Multifidelity Data Fusion: Application to Blended-Wing-Body Multidisciplinary Analysis Under Uncertainty,” *AIAA Journal*, Vol. 58, No. 2, 2020, pp. 889–906. doi:10.2514/1.J058388.

[4] Panzeri, M., Savylyev, A., Anisimov, K., d’Ippolito, R., and Mirzoyan, A., “Uncertainty quantification and robust design optimization applied to aircraft propulsion systems,” *Transportation Research Procedia*, Vol. 29, 2018, pp. 289–302. doi:10.1016/j.trpro.2018.02.026.

[5] Sekar, V., Zhang, M., Shu, C., and Khoo, B. C., “Inverse Design of Airfoil Using a Deep Convolutional Neural Network,” *AIAA Journal*, Vol. 57, No. 3, 2019, pp. 993–1003. doi:10.2514/1.J057894.

[6] Sun, G., Sun, Y., and Wang, S., “Artificial neural network based inverse design: Airfoils and wings,” *Aerospace Science and Technology*, Vol. 42, 2015, pp. 415–428. doi:10.1016/j.ast.2015.01.030.

[7] Skinner, S., and Zare-Behtash, H., “State-of-the-art in aerodynamic shape optimisation methods,” *Applied Soft Computing*, Vol. 62, 2018, pp. 933 – 962. doi:10.1016/j.asoc.2017.09.030.

[8] Mader, C. A., and Martins, J. R. R. A., “Derivatives for Time-Spectral Computational Fluid Dynamics Using an Automatic Differentiation Adjoint,” *AIAA Journal*, Vol. 50, 2012, pp. 2809–2819. doi:10.2514/1.J051658.

[9] Leung, T. M., and Zingg, D. W., “Aerodynamic shape optimization of wings using a parallel newton-krylov approach,” *AIAA Journal*, Vol. 50, No. 3, 2012, pp. 540–550.

[10] Bouhlel, M. A., Bartoli, N., Otsmane, A., and Morlier, J., “Improving Kriging Surrogates of High-Dimensional Design Models by Partial Least Squares Dimension Reduction,” *Structural and Multidisciplinary Optimization*, Vol. 53, No. 5, 2016, p. 935–952. doi:10.1007/s00158-015-1395-9.

[11] Bouhlel, M. A., and Martins, J. R. R. A., “Gradient-enhanced kriging for high-dimensional problems,” *Engineering with Computers*, Vol. 1, No. 35, 2019, pp. 157–173. doi:10.1007/s00366-018-0590-x.

[12] Queipo, N. V., Haftka, R. T., Shyy, W., Goel, T., Vaidyanathan, R., and Tucker, P. K., “Surrogate-Based Analysis and Optimization,” *Progress in Aerospace Sciences*, Vol. 41, No. 1, 2005, pp. 1–28. doi:10.1016/j.paerosci.2005.02.001.

[13] Peherstorfer, B., Willcox, K., and Gunzburger, M., “Survey of Multifidelity Methods in Uncertainty Propagation, Inference, and Optimization,” *Society for Industrial and Applied Mathematics*, Vol. 60, No. 3, 2018, pp. 550–591. doi:10.1137/16M1082469.

[14] Krige, D. G., “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, 1951, pp. 119–139.

[15] Blatman, G., “Adaptive Sparse Polynomial Chaos Expansion for Uncertainty Propagation and Sensitivity Analysis,” *Ph.D. Thesis, Blaise Pascal University - Clermont II. 3, 8, 9, France*, 2009.

[16] Goodfellow, I., Bengio, Y., and Courville, A., *Deep Learning*, The MIT Press, Cambridge, MA, 2016.

[17] Kennedy, C. M., and O’Hagan, A., “Predicting the Output from a Complex Computer Code When Fast Approximations are available,” *Biometrika*, Vol. 87, No. 1, 2000, pp. 1–13. doi:10.1093/biomet/87.1.1.

[18] Echeverria, D., and Hemker, P., “Manifold Mapping: A Two-Level Optimization Technique,” *Computing and Visualization in Science*, Vol. 11, 2008, pp. 193–206. doi:10.1007/s00791-008-0096-y.

[19] Singh, A. P., Medida, S., and Duraisamy, K., “Machine-Learning-Augmented Predictive Modeling of Turbulent Separated Flows over Airfoils,” *AIAA Journal*, Vol. 55, No. 7, 2017, pp. 2215–2227. doi:0.2514/1.J055595.

[20] Singh, A. P., and Duraisamy, K., “Using field inversion to quantify functional errors in turbulence closures,” *Physics of Fluids*, Vol. 28, 2016, pp. 045110–21. doi:10.1063/1.4947045.

[21] Wang, J.-X., Wu, J.-L., and Xiao, H., “Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data,” *Phys. Rev. Fluids*, Vol. 2, 2017, p. 034603. doi:10.1103/PhysRevFluids.2.034603.

[22] Wu, J.-L., Xiao, H., and Paterson, E., “Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework,” *Phys. Rev. Fluids*, Vol. 3, 2018, p. 074602. doi:10.1103/PhysRevFluids.3.074602.

[23] Ho, T. K., “Random Decision Forests,” *Proceedings of the Third International Conference on Document Analysis and Recognition*, Montreal, QC, 14–16 August 1995, pp. 278–282.

[24] Czarnecki, W. M., Osindero, S., Jaderberg, M., Świrszcz, G., and Pascanu, R., “Sobolev Training for Neural Networks,” *31st Conference on Neural Information Processing Systems*, Long Beach, CA, 4–9 December 2017.

[25] Maulik, R., Sharma, H., Patel, S., Lusch, B., and Jennings, E., “A turbulent eddy-viscosity surrogate modeling framework for Reynolds-Averaged Navier-Stokes simulations,” *Computers Fluids, In Press*, 2020, p. 104777. doi:10.1016/j.compfluid.2020.104777.

[26] Maulik, R., Sharma, H., Patel, S., Lusch, B., and Jennings, E., “Deploying deep learning in OpenFOAM with TensorFlow,” *AIAA Scitech 2021 Forum*, Vol. 2021-1485, Virtual, 11-15 and 19-21 January 2021. doi:10.2514/6.2021-1485.

[27] Breiman, L., “Random Forests,” *Machine Learning*, Vol. 45, No. 1, 2001, pp. 5–32.

[28] Forrester, A., Sobester, A., and Keane, A., *Engineering Design via Surrogate Modelling: A Practical Guide*, John Wiley and Sons, Ltd., United Kingdom, 2008.

[29] Opitz, D., and Maclin, R., “Popular ensemble methods: an empirical study,” *Journal of Artificial Intelligence Research*, Vol. 11, 1999, pp. 169–198. doi:10.1613/jair.614.

[30] Quinlan, J. R., “Simplifying decision trees,” *International Journal of Man-Machine Studies*, Vol. 27, No. 3, 1987, pp. 169–198. doi:10.1016/S0020-7373(87)80053-6.

[31] James, G., Witten, D., Hastie, T., and Tibshirani, R., *An Introduction to Statistical Learning with Applications in R*, Springer Science+Business Media, New York, United States, 2013.

[32] Driver, D. M., and Seegmiller, H. L., “Features of a reattaching turbulent shear layer in divergent channelflow,” *AIAA Journal*, Vol. 23, No. 2, 1985, pp. 163–171. doi:10.2514/3.8890.

[33] Weller, H. G., Tabor, G., Jasak, H., and Fureby, C., “A Tensorial Approach to Computational Continuum Mechanics using Object-Oriented Techniques,” *Computers in Physics*, Vol. 12, No. 6, 1998, pp. 620–631.

[34] Spalart, P. R., and Allmaras, S. R., “A One Equation Turbulence Model for Aerodynamic Flows,” *30th AIAA Aerospace Sciences Meeting and Exhibit*, Vol. 92-0439, Reno, NV, 6-9 January 1992. doi:10.2514/6.1992-439.