Computers and Fluids 200 (2020) 104431

Contents lists available at ScienceDirect

Computers and Fluids

journal homepage: www.elsevier.com/locate/compfluid

Flows over periodic hills of parameterized geometries: A dataset for data-driven turbulence modeling from direct simulations

Heng Xiao[a,*], Jin-Long Wu[a,1], Sylvain Laizet[b], Lian Duan[c]

[a] Kevin T. Crofton Department of Aerospace and Ocean Engineering, Virginia Tech, Blacksburg, VA, USA
[b] Department of Aeronautics, Imperial College London, UK
[c] Department of Mechanical and Aerospace Engineering, The Ohio State University, Columbus, OH, USA

ARTICLE INFO

Article history:
Received 2 October 2019
Revised 22 December 2019
Accepted 7 January 2020
Available online 9 January 2020

Keywords:
Physics-informed machine learning
Turbulence modeling
Separated flows

ABSTRACT

Computational fluid dynamics models based on Reynolds-averaged Navier–Stokes equations with turbulence closures still play important roles in engineering design and analysis. However, the development of turbulence models has been stagnant for decades. With recent advances in machine learning, data-driven turbulence models have become attractive alternatives worth further explorations. However, a major obstacle in the development of data-driven turbulence models is the lack of training data. In this work, we survey currently available public turbulent flow databases and conclude that they are inadequate for developing and validating data-driven models. Rather, we need more benchmark data from systematically and continuously varied flow conditions (e.g., Reynolds number and geometry) with maximum coverage in the parameter space for this purpose. To this end, we perform direct numerical simulations of flows over periodic hills with varying slopes, resulting in a family of flows over periodic hills which ranges from incipient to mild and massive separations. We further demonstrate the use of such a dataset by training a machine learning model that predicts Reynolds stress anisotropy based on a set of mean flow features. We expect the generated dataset, along with its design methodology and the example application presented herein, will facilitate development and comparison of future data-driven turbulence models.

1. Introduction

Although turbulence affects natural and engineered systems from sub-meter to planetary scales, fundamental understanding and predictive modeling of turbulence continue to defy and bedevil scientists and engineers. Turbulent flows are typical multi-scale physical systems that are characterized by a wide range of spatial and temporal scales. When predicting such systems, first-principle-based simulations are prohibitively expensive, and the small-scale processes must be modeled. For simulating turbulent flows, this is done by solving the Reynolds-Averaged Navier–Stokes (RANS) equations with the unresolved processes represented by turbulence model closures.

While the past two decades have witnessed a rapid development of high-fidelity turbulence simulation methods such as large eddy simulations (LES), they are still too expensive for practical systems such as the flow around an commercial airplane [1]. It is expected that using LES for engineering design will remain infeasible for decades to come. On the other hand, attempts in combining models of different fidelity levels (e.g., hybrid LES/RANS models) have shown promise, but how to achieve consistencies in the hierarchical coupling of models is still a challenge and a topic of ongoing research. Consequently, Reynolds-Averaged Navier–Stokes (RANS) equations are still the workhorse tool in engineering computational fluid dynamics for simulating turbulent flows.

It is well known that RANS turbulence models have large model-form uncertainties for a wide range of flows [2], which diminish the predictive capabilities of the RANS-based CFD models. Development of turbulence models has been stagnant for decades, which is evident from the fact that currently used turbulence models (e.g., k–ε, k–ω, and Spalart–Allmaras models [3–5]) were all developed decades ago despite unsatisfactory performance for many flows. In view of the importance and stagnation of turbulence modeling, NASA Technology Roadmap [6] called for sustained investment in improvements in the modeling of turbulent flows [7]. In particular, NASA CFD 2030 vision [8] identified the development of advanced turbulence model as a key priority.

Recent advances in machine learning techniques have enabled researchers to explore data-driven turbulence models as attractive alternatives to traditional algebraic models and Reynolds stress transport models. Duraisamy et al. [9,10] introduced a multiplica-

* Corresponding author.
E-mail address: hengxiao@vt.edu (H. Xiao).
[1] Current affiliation: California Institute of Technology, Pasadena, CA, USA.

https://doi.org/10.1016/j.compfluid.2020.104431
0045-7930/

tive discrepancy field to the source term of the turbulence transport equations. Xiao and co-workers [11,12] built a regression function for the RANS-modeled Reynolds stress discrepancies in terms of mean flow quantities. Ling et al. [13] directly built regression functions for the Reynolds stresses with data from DNS databases. Weatheritt and Sandsberg [14,15] used symbolic regression and gene expression programming for learning the coefficients in algebraic turbulence models. Schmelzer et al. [16] used deterministic symbolic sparse regression to infer algebraic stress models and perform a systematic assessment on separated flows.

While machine learning has repeatedly exceeded expert expectations in business applications from IBM's Jeopardy-winning Watson to Google's World-Champion-beating AlphaGo, their applications in physical modeling, and particularly turbulence modeling, have been hindered by two major obstacles: (1) the difficulty in incorporating domain knowledge of the physical systems (e.g., conservation laws, realizability, energy spectrum), and (2) the lack of publicly available, high-quality training flow databases. In the fields where machine learning has achieved tremendous successes (e.g., computer vision as detailed below), neither of these difficulties exist. To address the first obstacle above, Raissi et al. [17] proposed physics-informed neural networks by encoding the partial differential equations (PDEs) and the associated boundary and initial conditions to the loss function of the neural networks. Effectively such a framework amounts to representing the formal solution to the PDEs (mapping from space x and time t to the field variables, e.g., velocity u and pressure p of fluid flows) in the form of a neural network, which are then trained by using data. The approach has been successfully applied to fluid flows [18]. In a similar spirit yet different context, Xiao and co-workers [19,20] recently explored enforcing deterministic and statistical physical constraints on generative adversarial networks (GANs) by modifying the loss functions. They showed that such modifications not only improved the convergence rate and robustness of the GANs but also helped GANs better conform to the physical constraints, including both the conservation-law-like deterministic constraints and the energy-spectrum-like statistical constraints. The present work, on the other hand, will primarily discuss the second obstacle, i.e., the bottleneck of training data for data-driven turbulence modeling.

1.1. The need for parameterized datasets for data-driven turbulence modeling

Training datasets play pivoting roles in developing algorithms in machine learning and data-driven modeling. For example, in the field of image classification and computer vision, the dataset ImageNet has greatly facilitated the advancement of machine learning algorithms in the past decades [21]. As of now, the ImageNet dataset contains 14 million images hand-annotated with one of the 20,000 labels, which indicates what object is pictured [22].

Currently available public datasets are inadequate for training and testing data-driven turbulence models for RANS simulations. For example, the JHU database consists of DNS data of simple flows such as homogeneous isotropic turbulence and plane channels flow of friction Reynolds numbers ranging from $Re_\tau = 180$ to $Re_\tau = 5200$ [23,24]. Despite the high quality and large amounts of the data, such flows are not the focus of RANS modeling. The AGARD dataset [25] consists of a number of flows of engineering interests, but the data are meant for validating LES. The NASA Langley turbulence modeling portal [26] contains a number of complex flows of interest to the RANS modeling community. This database is unique in that it is intended to serve as *verification* of user-implemented RANS turbulence models rather than for validation efforts. That is, they provided the solution a particular RANS turbulence model should produce rather than the true physical solution, although the data for the latter are often available as well. All the above-mentioned datasets consist of representative flows that are clearly distinct from one another. For example, typical flows include attached boundary layers, separated flows, bluff body wakes, and jets. While such dataset are suitable for validating the implementation of traditional models, they are not suitable for developing data-driven models. In summary, existing high-fidelity simulations (DNS and LES) are performed to probe turbulent flow physics, and the simulations are often performed on representative but sufficiently different configurations. As a result, existing benchmark databases [25–29] are sparsely scattered in the parameter space, as they were generated for understanding flow physics or validating CFD solvers.

There are two noteworthy exceptions. First, Pinelli et al. [30] performed DNS of fully developed turbulent flows in a square duct at Reynolds numbers ranging from 1400 to 2500. The data are well curated and publicly available at their institutional website [31]. This flow is of great interest to the turbulence modeling community, because the in-plane secondary flows (recirculation) induced by the imbalance of Reynolds normal stresses cannot be captured by linear eddy viscosity models (the most commonly used models at present). This feature is challenging to capture even with advanced models such as nonlinear eddy viscosity models and Reynolds stress transport models. A shortcoming of this dataset is that the series of flows differ only in Reynolds numbers and not in geometry. It would be a more challenging test if a model trained on flows in square ducts are used to predict flows in rectangular ducts. The same group has performed DNS of flows in rectangular ducts, but the data are not yet publicly available. Another set of flows that can potentially be used for developing data-driven models is the flows separated from smoothly contoured surfaces. With benchmark data scattered in the NASA database [26] and the ERCOFTAC database [27], these include flow in a wavy channel [32], flow over a curved backward facing step [33], flow in a converging–diverging channel [34], and flow over periodic hills [35]. However, even the configurations in such an extensively studied class of flows can differ from each other dramatically in terms of obstacle-height based Reynolds number (ranging from $O(10^2)$ to $O(10^5)$), periodicity, and flow reattachment.

Similar to other application fields of machine learning, training and testing of data-driven turbulence models require data from *systematically* and *continuously* varied flow conditions (e.g., Reynolds number, geometry, angle of attack, and pressure gradient) with a maximum coverage of parameter space. Such a parameter-sweeping database is essential for researchers to test the predictive generality of their data-driven models. For example, while each data-driven turbulence model has its own characteristics and generalization capability, one can reasonably expect a data-driven model trained on one class of flows (or a larger dataset consisting these flows as a subset) to be able to predict other flows of the same type. For example, a model trained on attached boundary layers of various pressure gradients should be able to predict a flow with a pressure gradient unseen in the training data, possibly even higher or lower than any of the flows, i.e., involving extrapolation. However, it would be challenging if it is used to predict flows with physics absent in the training flows, e.g., flow with massive separations. Unfortunately, despite the two exceptions above, generally speaking one can state that databases tailored for data-driven modeling of turbulent flows are lacking [36]. As a result, developers of data-driven models have relied on their own in-house datasets [10,12,13], making it difficult to compare different methods rigorously due to the different datasets they used. A com-

prehensive, publicly available database specifically built for data-driven turbulence modeling is essential for the growing community.

1.2. Building a parameter-sweeping database for data-driven turbulence modeling

Data-driven turbulence modeling requires datasets of flows at systematically varied flow conditions. As a benchmark database, the quality and reliability of data are still of paramount importance. In light of such unique requirements, we aim to build such a database by using carefully performed Direct Numerical Simulations (DNS). It is expected that small-scale experiments can be leveraged to achieve the same objectives. On the other hand, large eddy simulations would be less ideal as they involve the modeling of sub-grid scale stresses and thus lead to uncertainties that are difficult to quantify. A similar recent study [37] used LES to study a parametric set of bumps, which can be useful for data-driven models that only require benchmark data for integral quantities (e.g., drag coefficient) or velocities. However, caution should be exercised when using the LES-generated Reynolds stresses due to the sub-grid scale stress modeling involved in LES. Due to the limited available resources and the high computational costs of DNS, we inevitably have to focus on turbulent flows of moderate Reynolds numbers that are much lower than those in practical flows. Despite such limitations, such a dataset is still useful for training and evaluating data-driven turbulence models.

We choose separated flows as a starting point to build a database tailored for development and evaluation of data-driven turbulence models. Turbulent flows separated from a smoothly contoured surface are ubiquitous in natural and engineered systems from rivers with complex bathymetry to airfoils and gas turbines at off-design conditions (e.g., stall and take-off). Such flows are highly complex and depend on Reynolds number, boundary geometry, the presence of mean pressure gradient and acceleration, among others. In particular, these flows are characterized by strong non-local, non-equilibrium effects, which clearly violates the equilibrium assumption (between turbulence production and dissipation) made in most eddy viscosity models. Currently, no model performs generally well in flows with equilibrium turbulence (e.g., attached boundary layer) and in those with non-equilibrium turbulence (separated flows). A dataset of separating flows of parameterized geometries would be a valuable addition to the existing benchmark databases for data-driven turbulence modeling. The design methodology proposed here for generating databases of parameterized flow configurations can be extended to other geometries such as airfoils of parameterized shapes and angles of attack, or wall-mounted objects in flows at various pressure gradients.

The rest of the paper is organized as follows. Section 2 introduces the design of the cases and the methodology used in the direct numerical simulations for generating the data. Section 3 presents an example to illustrate the process of constructing and testing a machine learning based model based on this dataset. Furthermore, selected mean flow features and the output are analyzed to shed light on the working of machine learning based flow prediction. Finally, Section 4 concludes the paper.

2. Methodology

The flow over periodic hills is widely utilized to evaluate the performance of turbulence models [e.g., 38,39] due to the comprehensive experimental and numerical benchmark data at a wide range of Reynolds numbers [35,40,41] from $Re = 700$ to 10595. The geometry of the computational domain is shown in Fig. 1a. Recently, Gloerfelt and Cinnella performed LES for flows in the same geometry [42] at Reynolds numbers[2] up to 37,000. However, benchmark data with systematically varied geometry configurations is still lacking. In this work, we performed DNS to build a database of separated flows of the periodic hill geometry [35] with various steepness ratios (Fig. 1b, c) at Reynolds number $Re = 5600$, which is based on the crest height H and the bulk velocity U_b at the crest. Periodic boundary conditions are applied in the streamwise (x) direction, and non-slip boundary conditions are applied at the walls. The spanwise (z) direction is homogeneous (periodic boundary conditions for the DNS) and thus the mean flow is two-dimensional.

The variation in geometry (as shown in Fig. 1c) causes incipiently-separated, mildly-separated, and vastly separated flows to occur (see Fig. 2). The data have been used in our previous works on developing and evaluating data-driven turbulence models based on physics-informed machine learning[12,44,45]. Great caution was exercised in ensuring the quality of the simulations by comparing with the previous benchmark data [35] in both the mean velocities and the Reynolds stresses (detailed in Section 2.3).

2.1. Numerical methods

Direct numerical simulations are performed by solving the forced incompressible Navier–Stokes equations for a fluid with a constant density ρ:

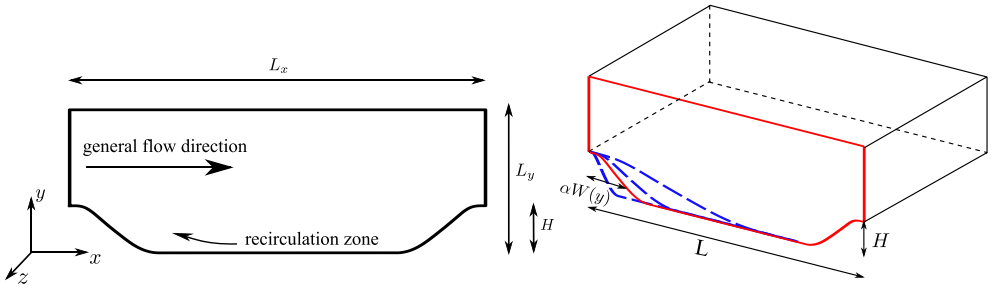
$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p - \frac{1}{2}[\nabla(\mathbf{u}\otimes\mathbf{u}) + (\mathbf{u}\cdot\nabla)\mathbf{u}] + \nu\nabla^2\mathbf{u} + \mathbf{f} \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \tag{2}$$

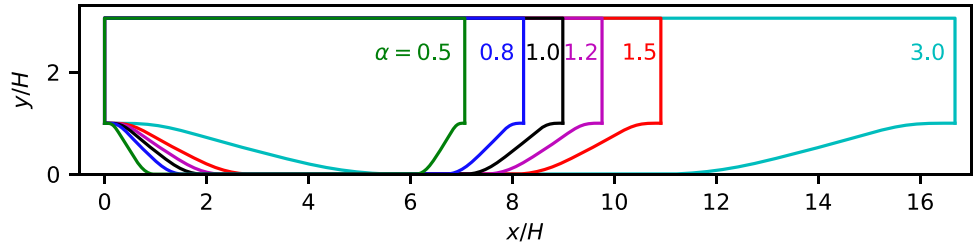
where $\otimes$ indicates outer-product of vectors; $p(\mathbf{x}, t)$ is the pressure field and $\mathbf{u}(\mathbf{x}, t)$ the velocity field; $\mathbf{x}$ and t are spatial and temporal coordinates, respectively. The forcing field $\mathbf{f}(\mathbf{x}, t)$ is used to enforce boundary conditions through an immersed boundary method and to ensure a specified mass flux. Note that convective terms are written in the skew-symmetric form, which allows the reduction of aliasing errors while retaining energy conservation for the spatial discretization permitted in the code [46].

These equations are solved on a Cartesian mesh with the high-order flow solver `Incompact3d`, which is based on sixth-order compact schemes for spatial discretization and a third-order Adams–Bashforth scheme for time advancement. The code is publicly available on GitHub [47]. To treat the incompressibility condition, a fractional step method is used, which requires solving a Poisson equation and it is solved fully in the spectral space. The divergence-free condition is ensured up to machine accuracy by utilizing the concept of modified wavenumber. The pressure mesh is staggered from the velocity mesh by half a mesh point to avoid spurious pressure oscillations. The modelling of the hill geometry is performed with a customized immersed boundary method based on a direct forcing approach that ensures a zero-velocity boundary condition at the wall. Following the strategy of [48], an artificial flow is reconstructed inside the 2D hill in order to avoid any discontinuities at the wall. More details about the present code and its validation can be found in [49]. The high computational cost of the present simulations requires the parallel version of this solver. The computational domain is split into a number of subdomains (pencils) which are each assigned to an MPI-process. The derivatives and interpolations in the x-, y-, and z- directions are performed in X-, Y-, and Z-pencils, respectively. The 3D FFTs required by the Poisson solver are also broken down as series of 1D FFTs computed in one direction at a time. The parallel version of

[2] The database [43] only contains data up to $Re = 19000$ at the time when this article is in press.

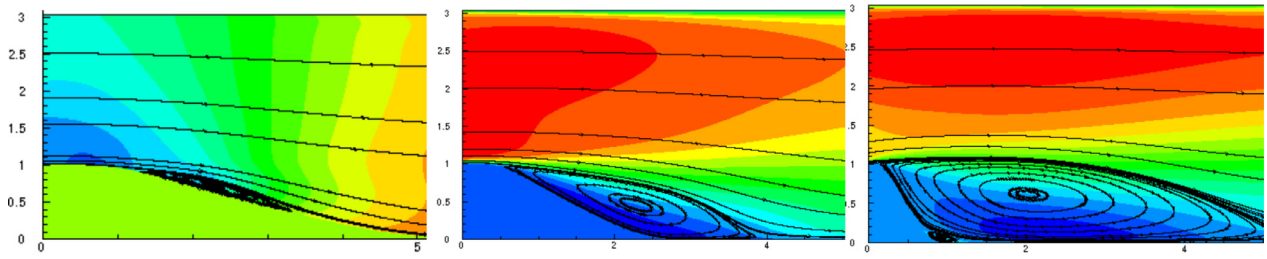


(a) Schematic of baseline geometry (b) Scheme of slope variation



(c) Geometries on which DNS are performed

Fig. 1. Schematic of the periodic hill geometry and the proposed scheme to achieve different separation patterns by systematically varying the steepness of the hill. (a) Computational domain for the flow over the baseline geometry of periodic hills. The x, y and z coordinates are aligned with streamwise, wall-normal, and spanwise, respectively. The dimensions of the original geometry are normalized with H with $L_x/H = 9$, $L_y/H = 3.036$, and $L_z/H = 0.1$. (b) Scheme for slope variation. The solid line denotes the hill profile of the baseline geometry. The dashed lines show different steepness of the profiles, parameterized by multiplying with a factor α to the hill width $W(y)$ of the baseline profile. The length of the bottom flat region is kept constant in this work but can potentially be varied in future efforts. (c) Geometry variations obtained by scaling the width of the hill by a factor of $\alpha = 0.5$, 0.8, 1.0 (baseline), 1.2, 1.5, and 3.0, for which DNS data are generated. As the length of the flat section is constant, the total horizontal length of the domain is $L_x/H = 3.858\alpha + 5.142$.



(a) $\alpha = 3$ (b) $\alpha = 1.5$ (c) $\alpha = 0.5$

Fig. 2. Separated flows come in many sub-regimes depending on the Reynolds number and slope geometry, among others. This figure shows that the size of separation bubble increases as the hill slope becomes steeper in the flow over periodic hills at $Re = 5600$, covering a range from (a) incipient separation to (b) mild separation and (c)massive separation.

`Incompact3d` can be used with up to one million computational cores [50].

The solver `Incompact3d` has been used recently for a variety of projects ranging from the study of the main features of gravity currents [51–53], performance analysis of active and passive drag reduction techniques [54–56], heat transfer feature of an impinging jet [57,58], wake-to-wake interactions in wind farms [59,60], and non-equilibrium scalings of turbulent wakes [61–63].

2.2. Flow set-up

Following the design of the baseline periodic hill geometry, the coordinates of the hill are represented as piecewise third-order polynomial functions (detailed in Appendix A). The second hill geometry is described by the same equation with a horizontal translation.

When varying the slope of the hills, we keep its height H constant and change its width. The length of the flat section between the hills, which is 5.142 in the baseline geometry, is kept constant as well. Therefore, total horizontal lengths L_x of the computational domain for the varied geometries are thus given by

$$L_x/H = 3.858\alpha + 5.142.$$

The parameter α changes the width of the hill by elongating the x-coordinates of the geometry. The distance between the first hill geometry and the second hill geometry thus changes with α. The reference case corresponds to $\alpha = 1$ and a crest-to-crest distance of $L_x = 9$. The meshes used for hill geometries of various slopes are presented in Table 1.

The initial condition for the streamwise velocity field is given by

$$u(y) = 1 - \left(\frac{y}{H}\right)^2$$

while the initial conditions for the vertical and spanwise velocity field are equal to zero. The mass flow rate for all the simulations

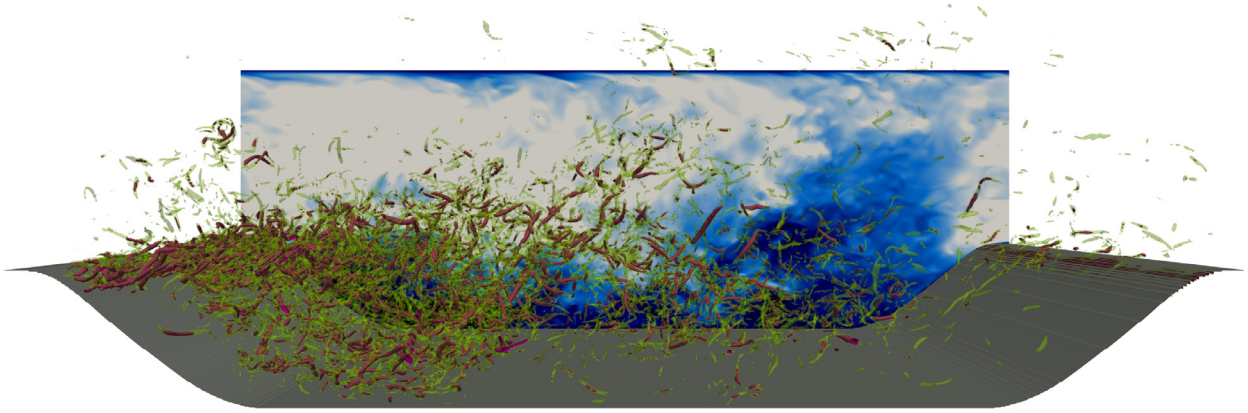


Fig. 3. Perspective view of vorticity Ω modulus isosurfaces $|\Omega| = 100U_b/H$ (green/lighter grey) and $|\Omega| = 200U_b/H$ (pink/darker grey) for the simulation with the baseline geometry ($\alpha = 1$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 1
Summary of mesh resolution along streamwise- (x-), wall-normal (y), and spanwise- (z-) directions. A stretched mesh in the vertical direction is used with a refinement towards the walls. The height of the first cell along wall-normal direction is $\Delta y_{\text{wall}}/H \approx 6.22 \times 10^{-3}$. The spatial resolution is comparable to those found in the benchmark numerical simulations [35] at the same Reynolds number $Re = 5600$.

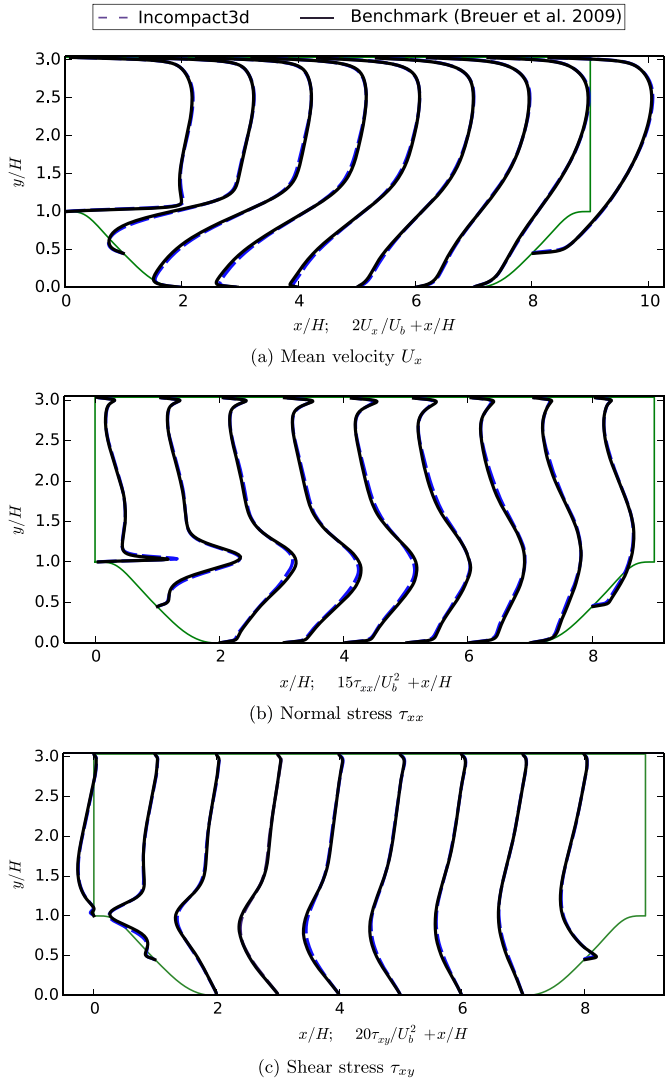
α	mesh ($n_x \times n_y \times n_z$)
0.5	$768 \times 385 \times 128$
0.8	$704 \times 385 \times 128$
1.0	$768 \times 385 \times 128$
1.2	$832 \times 385 \times 128$
1.5	$768 \times 385 \times 128$
3.0	$768 \times 385 \times 128$

is kept constant in time via the imposition of a constant pressure gradient at each time step. The data are collected after a transitional period, from the point when the flow is fully developed. A simulation time step $\Delta t = 0.0005H/U_b$ is used. For all the calculations, turbulent statistical data have been collected over a time period $T = 150H/U_b$.

An example visualization of the instantaneous flow field is presented in Fig. 3 from the simulation with the baseline geometry ($\alpha = 1$). One can clearly see the highly three-dimensional nature of the flow structures that develop around the recirculation region of the flow. Intense elongated small-scale flow structures are present when the flow separates at the first hill crest while there is only a small number of intense turbulent structures after the reattachment of the flow.

2.3. Mesh sensitivity study and validation to benchmark data

We have performed careful validations with existing simulations of flow over periodic hills. Fig. 4 shows a comparison of mean horizontal velocity and Reynolds stress components between the results computed by `Incompact3d` and those presented by Breuer et al. [35] at $Re = 5600$. It can be seen that excellent agreements were obtained for both the mean velocity and Reynolds stress components. The comparison of mean vertical velocity U_y shows similar agreement as comparison to U_x and is thus omitted here. It should be noted that the normal stress component τ_{xx} is representative of the strength of turbulent kinetic energy, while the shear stress component τ_{xy} reflects the strength of the shear flow downstream of the hill crest. Both of these two components



(a) Mean velocity U_x

(b) Normal stress τ_{xx}

(c) Shear stress τ_{xy}

Fig. 4. Validation of `Incompact3d` (dashed lines) against the benchmark data of Breuer et al. [35] (solid lines), showing profiles of (a) mean velocities U_x, (b) turbulent normal stresses τ_{xx}, and (c) turbulent shear stresses τ_{xy} at nine different streamwise locations $x/H = 0, 1, \ldots, 8$.

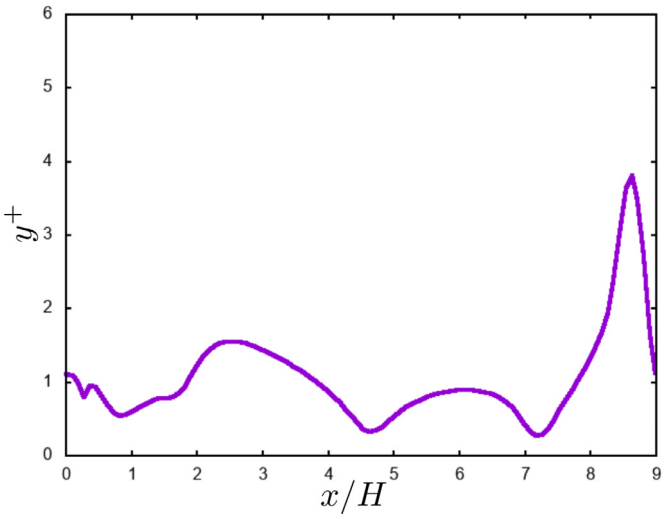


Fig. 5. Distribution of y^+ at the bottom wall of the periodic hill domain for the baseline geometry case with $\alpha = 1$. The quantity y^+ is defined as $y^+ = \Delta y_{\text{wall}} u_\tau / \nu$ with Δy_{wall} denoting the distance of the first mesh node from the bottom wall and $u_\tau = \sqrt{\tau_{xy}/\rho}$ denoting the shear stress velocity.

from the results computed by `Incompact3d` show a good agreement with those of Breuer et al. [35]. The comparison of normal stress component τ_{yy} also showed a similar good agreement as in Fig. 4b and is omitted here for brevity. For completeness, we present in Fig. 5 the distribution of non-dimensional y^+ values for the case with $\alpha = 1$, where y^+ is defined as $y^+ = \Delta y_{\text{wall}} u_\tau / \nu$ with Δy_{wall} denoting the distance of the first mesh node from the bottom wall and $u_\tau = \sqrt{\tau_{xy}/\rho}$ denoting the shear stress velocity. It can be seen that the values obtained in the present study are consistent with those reported in the literature [35].

In order to demonstrate the convergence of resolved turbulence stresses in our database, we further performed simulations with either a larger time step $\Delta t' = 2\Delta t$ or coarser spatial resolutions. The results from these simulations are compared with the results of the original simulation in Figs. 6 and 7. It can be seen that the turbulence stresses τ_{xx} and τ_{xy} do not show any noticible changes by coarsening the temporal or spatial resolutions of the original simulation. Therefore, the convergence of resolved turbulence stresses has been achieved with the resolutions of the simulations presented here. It should be noted that the gradients of velocity and pressure fields used to construct the features (e.g., q_4, q_6, and q_7 in Table 2 in Section 3) are obtained by taking gradient of the mean flow fields on the coarse mesh (e.g., RANS mesh) and not on the DNS mesh.

3. Data analysis, machine learning, and interpretation

The data generated by performing DNS on the designed cases as described in Section 2 are made available in a public GitHub repository [64] and will also be distributed through the NASA Turbulence Modeling Portal [26]. The data include mean pressure field, mean velocities fields, and second order statistics (Reynolds stress fields). As such, our dataset are primarily valuable for developing RANS-based turbulence models. Based on the current literature, most data-driven turbulence models are constructed from these quantities or those derived therefrom, such as strain-rate and rotation-rate (the symmetric and anti-symmetric parts of velocity gradient), pressure gradient, and streamline curvature. We recognize that future models may need high-order statistics (e.g., velocity triple correlation tensor, pressure-strain rate tensor), which will be recorded in future simulations.

From these mean pressure and velocity fields, various other features can be derived. Table 2 shows a set of five hand-crafted mean flow features following that of Ling and Templeton [65] and Wang et al. [11]. In a typical RANS simulation, these quantities would be available, and such mean flow features have been used to predict the reliability of linear eddy viscosity models [65], the discrepancies of the RANS-modeled Reynolds stresses [11,66]. They can also be used to predict Reynolds stress itself or the eddy viscosity [67] without relying on any baseline RANS models [13]. The specific choice of mean flow features (inputs) and the output quantities (Reynolds stress, eddy viscosity, or their discrepancies, or the symbolic form of the Reynolds stress–strain-rate function) are what differentiate the data-driven models.

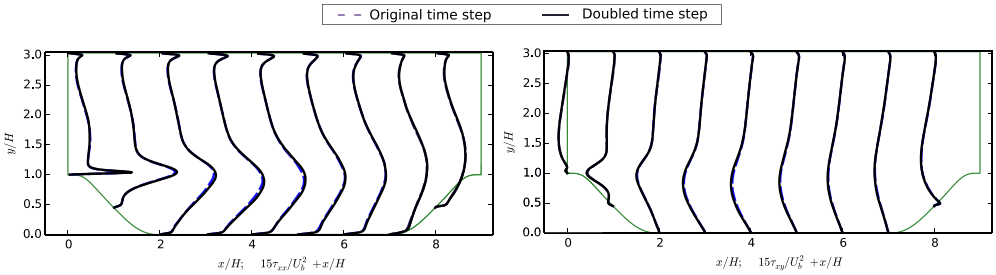
3.1. Problem formulation

We aim to present a methodology of generating DNS benchmark datasets of systematically varying flow configures for data-driven turbulence modeling. As such, we refrain from advocating any particular data-driven turbulence model. Here we use an example to illustrate the possible usage of our dataset for training and testing data-driven models. To this end, machine learning models are constructed to predict Reynolds stress anisotropy based on the mean flow features $\mathbf{q}$ shown in Table 2, which is a subset of what were used in [11]. The original features that are based on quantities from RANS solvers (e.g., turbulent kinetic energy k and dissipation rate ε) are omitted here as no RANS simulations are performed in this work. Only DNS data are used in this example. The output quantities as the anisotropy of the Reynolds stress tensor defined based on the following eigen-decomposition:

$$\boldsymbol{\tau} = 2k\left(\frac{1}{3}\mathbf{I} + \mathbf{V}\Lambda\mathbf{V}^T\right) \tag{3}$$

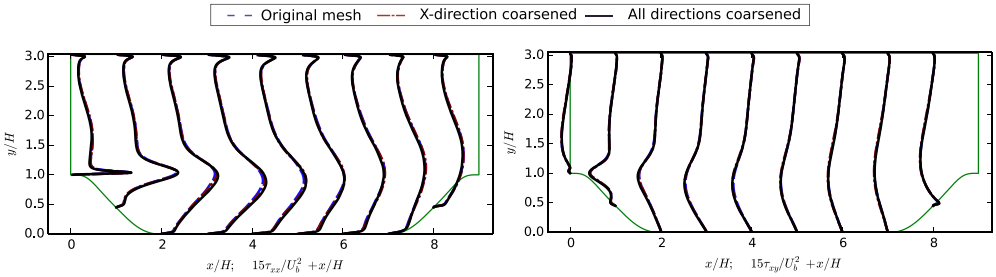
Table 2
Non-dimensional flow features used as input in the example machine-learning models. The normalized feature q_β is obtained by normalizing the corresponding raw features value $\hat{q}_\beta$ with normalization factor q^*_β according to $q_\beta = \hat{q}_\beta / (|\hat{q}_\beta| + |q^*_\beta|)$. Repeated indices i, j, k, l, imply summation. Notations are as follows: U_i is mean velocity, $\mathbf{S}$ is the strain rate tensor, $\boldsymbol{\Omega}$ is the rotation rate tensor, $\boldsymbol{\Gamma}$ is unit tangential velocity vector, D denotes material derivative, and L_c is the characteristic length scale of the mean flow. $\|\cdot\|$ and $|\cdot|$ indicate matrix and vector norms, respectively.

feature (q_β)	description	raw feature ($\hat{q}_\beta$)	normalization factor (q^*_β)
q_1	ratio of excess rotation rate to strain rate (Q-criterion)	$\frac{1}{2}(\|\boldsymbol{\Omega}\|^2 - \|\mathbf{S}\|^2)$	$\|\mathbf{S}\|^2$
q_4	pressure gradient along streamline	$U_k \dfrac{\partial P}{\partial x_k}$	$\sqrt{\dfrac{\partial P}{\partial x_j}\dfrac{\partial P}{\partial x_j} U_i U_i}$
q_6	ratio of pressure normal stresses to shear stresses	$\sqrt{\dfrac{\partial P}{\partial x_i}\dfrac{\partial P}{\partial x_i}}$	$\dfrac{1}{2}\rho \dfrac{\partial U_k^2}{\partial x_k}$
q_7	non-orthogonality between velocity and its gradient	$\left\lvert U_i U_j \dfrac{\partial U_i}{\partial x_j} \right\rvert$	$\sqrt{U_l U_l U_i \dfrac{\partial U_i}{\partial x_j} U_k \dfrac{\partial U_k}{\partial x_j}}$
q_{10}	streamline curvature	$\left\lvert \frac{D\boldsymbol{\Gamma}}{Ds} \right\rvert$ where $\boldsymbol{\Gamma} \equiv \mathbf{U}/\lvert\mathbf{U}\rvert$, $Ds = \lvert\mathbf{U}\rvert Dt$	$\dfrac{1}{L_c}$



(a) Normal stress τ_{xx} (b) Shear stress τ_{xy}

Fig. 6. Demonstration of **time-step size sensitivity** for `Incompact3d` results with baseline time step Δt and doubled time step $\Delta t' = 2\Delta t$, showing profiles of (a) turbulent normal stresses τ_{xx} and (b) turbulent shear stresses τ_{xy} at nine different streamwise locations $x/H = 0, 1, \ldots, 8$.



(a) Normal stress τ_{xx} (b) Shear stress τ_{xy}

Fig. 7. Mesh convergence study on the baseline geometry ($\alpha = 1$) for `Incompact3d` results with original mesh ($n_x \times n_y = 768 \times 385$), mesh coarsened along x direction (512×385) and mesh coarsened along both x and y directions (512×257), showing profiles of (a) turbulent normal stresses τ_{xx} and (b) turbulent shear stresses τ_{xy} at nine different streamwise locations $x/H = 0, 1, \ldots, 8$.

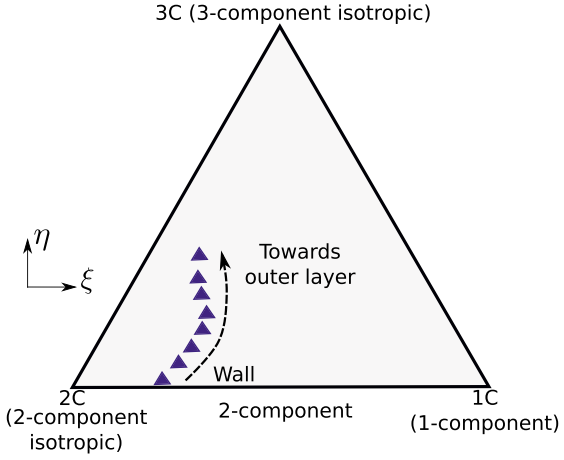


Fig. 8. The Barycentric triangle encloses physically realizable Reynolds stresses [69]. The position in the Barycentric triangle represents the componentality of the Reynolds stress anisotropy, e.g., three-component isotropic state (vertex 3C), two-component isotropic state (vertex 2C), one-component state (vertex 1C), and a combination thereof (interior). The coordinate system places the triangle in the $\xi - \eta$ plane with the origin (0, 0) located at 2C.

where k is the turbulent kinetic energy; $\mathbf{I}$ is the second order identity tensor; $\mathbf{V}$ and $\Lambda = \mathrm{diag}[\lambda_1, \lambda_2, \lambda_3]$ are the eigenvectors and eigenvalues of the Reynolds stress anisotropy tensor. The eigenvalues are further mapped to the Barycentric coordinates (C_1, C_2, C_3) as follows [68]:

$$C_1 = \lambda_1 - \lambda_2, \quad C_2 = 2(\lambda_2 - \lambda_3), \quad \text{and} \quad C_3 = 3\lambda_3 + 1. \tag{4}$$

The coordinate of a point $\boldsymbol{\xi} \equiv (\xi, \eta)$ in the Barycentric triangle can be expressed as that of the three vertices ($\boldsymbol{\xi}_{1c}$, $\boldsymbol{\xi}_{2c}$, and $\boldsymbol{\xi}_{3c}$), i.e., $\boldsymbol{\xi} = C_1\boldsymbol{\xi}_{1c} + C_2\boldsymbol{\xi}_{2c} + C_3\boldsymbol{\xi}_{3c}$. The Barycentric triangle and the $\xi - \eta$ coordinate system are illustrated in Fig. 8. Our example problem thus consists of (1) learning the functional mapping $\mathbf{q} \mapsto \boldsymbol{\xi}$ from mean flow features $\mathbf{q}$ to a frame-independent quantity $\boldsymbol{\xi}$ of the Reynolds stress from training data and (2) assessing the predictive performance of the learned function on a new flow.

3.2. Machine learning models used in this work

We use two machine learning models, random forests [70] and fully-connected neural networks, to build such mappings and to compare their performances in predictions. A brief introduction of the two models are given below for readers who are not familiar with machine learning.

Neural networks are nonlinear functions parameterized by weights W (and biases) that can be learned from data. They are built by consecutive composition of linear functions (matrix-vector multiplication) followed by nonlinear activation functions. A neural network in its simplest form is the linear model $\mathbf{y} = \mathbf{W}\mathbf{q}$, which is controlled by weight matrix $\mathbf{W}$ and maps input vector $\mathbf{q}$ to output vector $\mathbf{y}$. However, such a simple model may lack the flexibility to represent complex functions. This difficulty can be addressed by introducing one or more intermediate vectors. For example, $\mathbf{h} = \sigma\left(\mathbf{W}^{(1)}\mathbf{q}\right)$ and $\mathbf{y} = \mathbf{W}^{(2)}\mathbf{h}$, or written as composite function $\mathbf{y} = \mathbf{W}^{(2)}\,\sigma\left(\mathbf{W}^{(1)}\mathbf{q}\right)$, where σ is an activation function such as sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ or rectifier linear unit (ReLU), $\sigma(x) = \max(0, x)$. The functional mapping can be represented by the network diagram in Fig. 9, with each layer corresponding to a vector (e.g., $\mathbf{q}$ (input layer), $\mathbf{h}$ (hidden layer), or $\mathbf{y}$ (output layer)) and each neuron an element therein. Deep learning utilizes neural networks with many layers. Neural networks with at least one hidden layer are universal approximators, i.e., they can represent any continuous function on a compact domain to arbitrary accuracy, given enough hidden neurons [71]. In this work, the input and output layers have 5 and 2 neurons, respectively, which are dictated by the given problem. We use seven hidden layers with 36 neurons in each layer. The ReLu activation function is used. In the training we used 1000 epochs with a learning rate of 0.01.

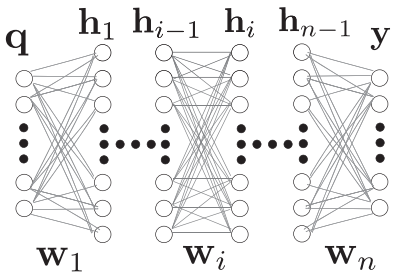


Fig. 9. Architecture of the fully-connected neural network with n layers (of weights) with $n = 7$ used in this work. Each hidden layer has 36 neurons.

Random forests are an ensemble learning method based on decision-tree regression models. Decision-tree models stratify the feature space into different non-overlapping regions so as to minimize the in-region variance of the training data. Then they predict the response for a given test input using the mean of the training observations in the region to which that test input belongs. Regression tree models are simple and useful for interpretation, but their prediction accuracy is generally not competitive compared to, e.g., neural networks. Random forests improve upon simple regression trees by building an ensemble of trees with bootstrap samples (i.e., sampling with replacement) drawn from the training data [72]. Moreover, when building each tree, it utilizes only a subset of randomly chosen features to reduce the correlation among the trees in the ensemble and thus decreases the bias of the ensemble prediction. Random forests have much lower computational costs compared to neural networks and are most robust with only two tuning parameters, i.e., the number N_{rf} of trees in the ensemble and the number M of selected features. In this work we used an ensemble of $N_{\text{rf}} = 300$ trees and a subset of features (i.e., $M = 3$) in each stratification (splitting) of the feature space.

Table 3
Overview of the scenarios for training and testing the data-driven model. Case 1 is an interpolation in the hill slope, while Case 2 involves extrapolation in the hill slope.

	Training set	Testing set
Case 1	$\alpha = \{0.5, 0.8, 1.2, 1.5\}$	$\alpha = 1.0$
Case 2	$\alpha = \{0.5, 0.8, 1.0, 1.2\}$	$\alpha = 1.5$

3.3. Predictive performance of machine learning models

We consider two cases (training/prediction scenarios):

(a) Case 1 involves an interpolation in flow configuration. The training set consists of data from four geometries: $\alpha = \{0.5, 0.8, 1.2, 1.5\}$ and the trained model is used to predict for the flow at $\alpha = 1.0$ (the baseline geometry).
(b) Case 2 involves an extrapolation in flow configuration. The training set consists of $\alpha = \{0.5, 0.8, 1.0, 1.2\}$, while the flow at $\alpha = 1.5$ is reserved for testing.

This is summarized in Table 3. In both cases, random forest and neural networks are trained on identical datasets for a fair comparison.

The predicted anisotropy as represented by the Barycentric plot trajectories are presented in Fig. 10 for three representative locations: $x/H = 2$, 5, and 7. In both flows ($\alpha = 1.0$ and 1.5, which are the predictive target of Case 1 and Case 2, respectively), the three locations correspond to the middle of the separation bubble, near the reattachment point, and after the reattachment, respectively. Overall, the Barycentric plots suggest that both random forest and neural network showed satisfactory agreement as compared to the DNS data for both Case 1 and Case 2. This is further detailed in Table 4, which shows that the relative prediction errors of both

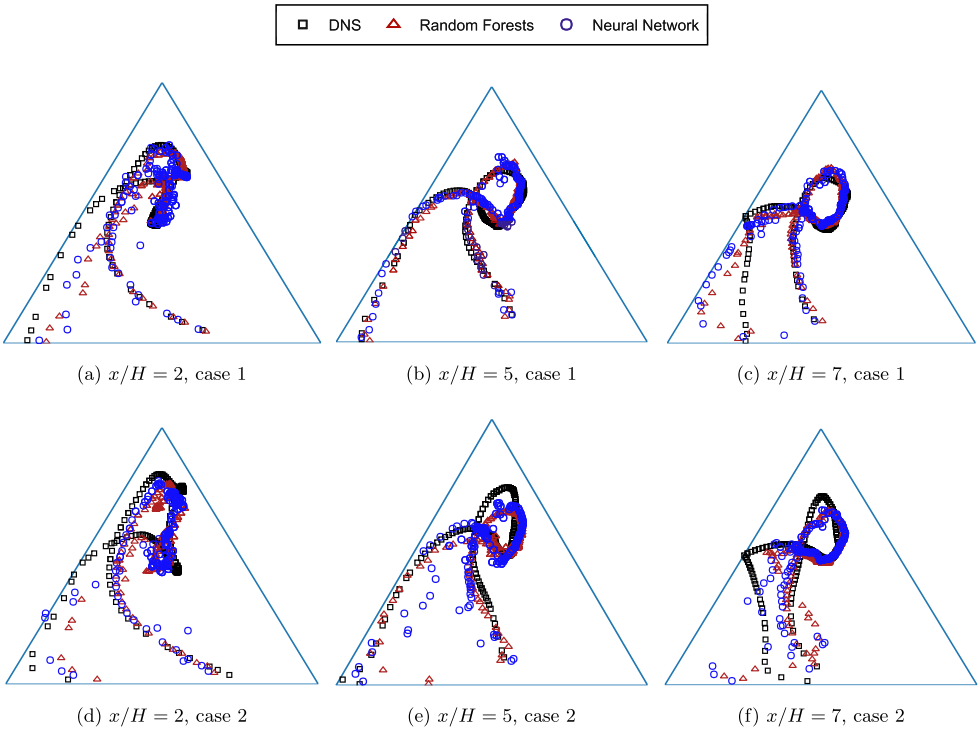


Fig. 10. Predicted Reynolds stress anisotropy for case 1 (interpolation; top panels a–c; the flow with baseline geometry $\alpha = 1$ is predicted) and case 2 (extrapolation; bottom panels d–f; the flow with geometry $\alpha = 1.5$ is predicted), displayed in Barycentric triangles. Predictions from random forests (RF) and neural networks (NN) are compared with the ground truth (DNS).

Table 4
Comparison of predictive performances in error percentage of (ξ, η) between random forests and neural networks on the interpolation and extrapolation cases.

	Random Forest	Neural Network
Case 1 (interpolation)	(10.4%, 4.8%)	(10.9%, 4.8%)
Case 2 (extrapolation)	(13.1%, 7.3%)	(14.5%, 6.8%)

random forest and neural network models are approximately 10% (for ξ) and 5% (for η) in the interpolation case (Case 1). These numbers are 14% and 7% in the extrapolation case (Case 2). Prediction percentage errors have been calculated using Frobenius norm, which for ξ can be defined as:

$$\epsilon_\xi = \frac{\|\xi_{\text{truth}} - \xi_{\text{predicted}}\|_F}{\|\xi_{\text{truth}}\|_F} \tag{5}$$

where the Frobenius norm is defined as $\|X\|_F = \sqrt{\sum_{i=1}^{n} |X_i|^2}$ for a vector $X \in \mathbb{R}^n$. Note that the values associated with all cells in the field are treated equally regardless of their volumes. That is, no volume averaging is performed when calculating the error norm. The percentage error for η is defined similarly. The reported error percentages would be approximately reduced by half if volume averaging is performed. When interpreting the errors rates in Table 4, the trend is more important than the exact numbers, because these numbers largely depend on the tuning parameters and the definition of error norm while the trend is valid in general. The similar predictive performances of random forests and neural networks as observed above are noteworthy, because neural networks are much more complex models than random forests. Neural networks have many more tuning parameters and are much more computationally expensive to training than random forests. A benefit that comes with such complexity is that they generally perform better in extrapolations. The counter-intuitive observation here seems to suggest that the data among different training flows does not have clear intrinsic trends that can be extrapolated. In such cases, a simpler model such as random forest is the preferred choice. In general cases where neural networks are preferred, random forests can still be a valuable tool to obtain first predictions which neural networks are expected to outperform.

Another clear trend as seen from Fig. 10 and Table 4 is that the predictive errors in the extrapolation case (Case 2; the flow at $\alpha = 1.5$ is predicted) is much larger than in the interpolation case (Case 1, where the flow at $\alpha = 1.0$ is predicted). This is expected because the case at $\alpha = 1.5$ has a much milder slope and much smaller separation bubble (see Fig. 2b) than any of the training flows. We have performed a similar extrapolation experiment by training on $\alpha = \{0.8, 1.0, 1.2, 1.5\}$ and testing on $\alpha = 0.5$. The predictive performance is comparable to Case 1 (interpolation) above for both random forests and neural networks. This is probably because the massive separation characterizing the flow at $\alpha = 0.5$ is already present in the training sets, particularly in flows with $\alpha = 0.8$ and 1.0.

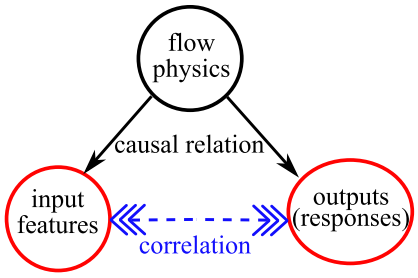
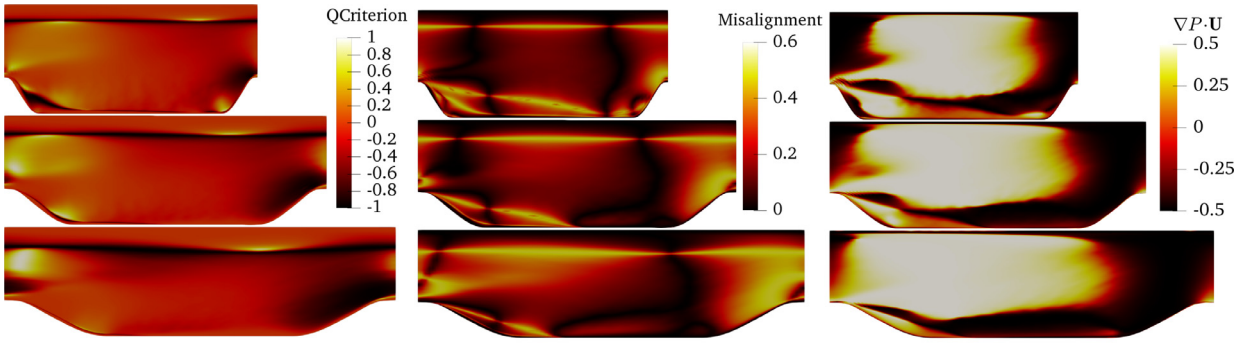


Fig. 11. Schematic illustration of why the machine learning based prediction works for flow problems. The input features and the output quantities stem from the same underlying physics. Machine learning algorithms discover such correlation from training data and are thus able to make physics-grounded prediction, even though the underlying physics are not explicitly identified.

3.4. Demystifying machine learning based flow predictions

Given the good agreement between the machine learning predicted Reynolds stress anisotropy and the DNS data, it is important to examine why and how it worked. State-of-the-art machine learning methods are still based on correlations rather than causal relations. This is fundamentally different from traditional, first-principle-based modeling workflow, which typically consists of (1) identifying principles from observations, (2) formulating equations to describe such principles, and (3) solving the equations to obtain predictions. In contrast, machine learning builds upon correlation between input features and output responses. This is not to say that machine learning does not capture physics. Rather, the input and the output are manifestation of the same underlying physics and thus are inevitably correlated. Such correlations are embedded in the training data and can be discovered by machine learning algorithms. The relations between input features, output responses, and the underlying flow physics are illustrated in Fig. 11. Machine learning algorithms discover input–output correlation from training data and are thus able to make physics-grounded prediction, even though the underlying physics are not explicitly identified.



(a) Q criterion (b) Non-orthogonality (c) Streamline pressure gradient

Fig. 12. Color contours of selected input features, Q criterion (q_1), velocity/velocity-gradient non-orthogonality (q_7), and pressure gradient along the streamlines (q_4) for three cases: $\alpha = 0.5$ (top panels), 1.0 (middle panels), and 1.5 (bottom panels).

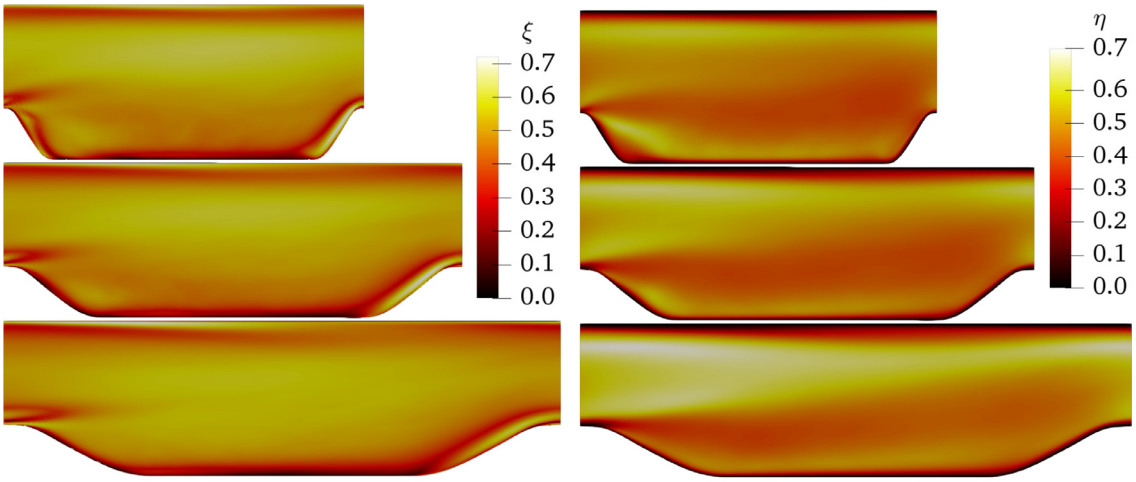


Fig. 13. Plots of **anisotropy** ξ and η for three cases: $\alpha = 0.5$ (top panels), 1.0 (middle panels), and 1.5 (bottom panels).

The relation between input features and the flow physics is illustrated in Fig. 12 by using three selected features: the Q-criterion (q_1), the non-orthogonality between velocity and its gradient (q_7), and pressure gradient along streamlines (q_4). Fig. 12a shows that all three flows ($\alpha = 0.5$, 1.0, and 1.5) show small values (darker regions) along the upper channel and the acceleration region (near the outlet). Larger values (brighter regions) are seen near the inlet. This observation suggests that the three flows do share similar flow patterns and the Q-criterion feature reflects such similarities. Fig. 12b shows that upper channels and the outer edge of the recirculation bubble are characterized by large velocity/velocity-gradient non-orthogonality, indicating their departure from parallel shear flow (channel-like flows). Again, such patterns are observed through all three flows ($\alpha = 0.5$, 1.0, and 1.5). Finally, Fig. 12c shows a similar trend for the pressure gradient along streamlines.

The output quantities also show the similar patterns across three flows as is evident from Fig. 13. For example, Fig. 13b shows that small values (darker regions) are found in the near-wall regions. Along with the small ξ values in the same region found from Fig. 13a, this observation suggests that the turbulence states here are closer to two-component state (lower left corner of the Barycentric triangle; see Fig. 8). In contrast, the core region downstream of the center of the inlet shows larger values of ξ and η, suggesting more isotropic turbulence states. In summary, machine learning algorithms discover the correlation between the input features and the outputs that stem from the same underlying physics to make flow predictions. The predicted trajectories in the Barycentric triangle as shown in Fig. 10 are in good agreement with the DNS data. This is impressive since a typical linear eddy viscosity model, which belongs to the most widely used turbulence models, are completely incapable of predicting such anisotropy correctly. Rather, it predicts an isotropic state (top vertex, 3C, in Fig. 8) at the wall and moves towards the center of the Barycentric triangle as we trace the turbulence away from the wall.

4. Conclusion

Data-driven turbulence modeling has emerged as a promising field in light of the long stagnation of traditional turbulence model development. However, the development of data-driven models is hindered by the lack of datasets specifically tailored for such purposes, because existing databases are sparsely scattered in the flow parameter space and are not suitable for training and testing data-driven models. To alleviate this bottleneck, in this work we advocate the construction of benchmark datasets by systematically varying flow configurations. To this end, we present the design and generation of a dataset consisting of flow over periodic hills of various slopes. It is expected that such a dataset will be valuable for data-driven turbulence model developers to train and evaluate the predictive capabilities of their models on separated flows. We further illustrate example usage of such presented dataset in training a data-driven model that maps mean flow features to Reynolds stress anisotropy.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Heng Xiao: Conceptualization, Methodology, Visualization, Writing - original draft. **Jin-Long Wu:** Investigation, Validation, Writing - original draft. **Sylvain Laizet:** Investigation, Software, Writing - original draft. **Lian Duan:** Investigation, Validation.

Acknowledgment

The computing time used in this work was provided by the UK turbulence consortium (grant number EP/R029321/1), which is gratefully acknowledged.

Appendix A. Description of the Hill Geometry

The coordinates of the first hill geometry consists of six segments of third-order polynomials described by the following equations:

$$\hat{y} = \min(1; 1 + 2.42 \times 10^{-4}\hat{x}^2 - 7.588 \times 10^{-5}\hat{x}^3), \quad \hat{x} \in [0, 0.3214]$$
$$\hat{y} = 0.8955 + 3.484 \times 10^{-2}\hat{x} - 3.629 \times 10^{-3}\hat{x}^2 + 6.749 \times 10^{-5}\hat{x}^3, \quad \hat{x} \in (0.3214, 0.5]$$
$$\hat{y} = 0.9213 + 2.931 \times 10^{-2}\hat{x} - 3.234 \times 10^{-3}\hat{x}^2 + 5.809 \times 10^{-5}\hat{x}^3, \quad \hat{x} \in (0.5, 0.7143]$$
$$\hat{y} = 1.445 - 4.927 \times 10^{-2}\hat{x} + 6.95 \times 10^{-4}\hat{x}^2 - 7.394 \times 10^{-6}\hat{x}^3, \quad \hat{x} \in (0.7143, 1.071]$$
$$\hat{y} = 0.6401 + 3.123 \times 10^{-2}\hat{x} - 1.988 \times 10^{-3}\hat{x}^2 + 2.242 \times 10^{-5}\hat{x}^3, \quad \hat{x} \in (1.071, 1.429]$$
$$\hat{y} = \max(0; 2.0139 - 7.18 \times 10^{-2}\hat{x} + 5.875 \times 10^{-4}\hat{x}^2 + 9.553 \times 10^{-7}\hat{x}^3), \quad \hat{x} \in (1.429, 1.929]$$

where $\hat{x} = x/H$ and $\hat{y} = y/H$ are normalized horizontal and vertical coordinates, respectively.

References

[1] Moin P, Kim J. Tackling turbulence with supercomputers. Sci Am 1997;276(1):46–52.
[2] Xiao H, Cinnella P. Quantification of model uncertainty in RANS simulations: a review. Prog Aerosp Sci 2019;108:1–31.
[3] Launder BE, Sharma BI. Application of the energy-dissipation model of turbulence to the calculation of flow near a spinning disc. Lett Heat Mass Transfer 1974;1(2):131–7.
[4] Menter FR. Two-equation eddy-viscosity turbulence models for engineering applications. AIAA J 1994;32(8):1598–605.
[5] Spalart PR, Allmaras SR. A one equation turbulence model for aerodynamic flows. AIAA Paper 1992. doi:10.2514/6.1992-439.
[6] Adler M, Wright M, Campbell C, Clark I, Engelund W, Rivellini T. Entry, descent, and landing roadmap. NASA TA09, April.
[7] National. Research council, NASA space technology roadmaps and priorities: restoring NASA's technological edge and paving the way for a new era in space. National Academies Press; 2012.
[8] Slotnick J, Khodadoust A, Alonso J, Darmofal D, Gropp W, Lurie E, Mavriplis D. CFD vision (2030 study: a path to revolutionary computational aerosciences, tech. rep. National Aeronautics and Space Administration, Langley Research Center. Virginia: Hampton; 2014. 23681-2199
[9] Singh AP, Duraisamy K. Using field inversion to quantify functional errors in turbulence closures. Phys Fluids 2016;28:45110.
[10] Singh AP, Medida S, Duraisamy K. Machine learning-augmented predictive modeling of turbulent separated flows over airfoils. AIAA J 2017;55(7):2215–27.
[11] Wang J-X, Wu J-L, Xiao H. Physics-informed machine learning approach for reconstructing Reynolds stress modeling discrepancies based on DNS data. Phys Rev Fluids 2017;2(3):34603.
[12] Wu J-L, Xiao H, Paterson EG. Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework. Phys Rev Fluids 2018;3:74602.
[13] Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modelling using deep neural networks with embedded invariance. J Fluid Mech 2016;807:155–66.
[14] Weatheritt J, Sandberg R. A novel evolutionary algorithm applied to algebraic modifications of the rans stress–strain relationship. J Comput Phys 2016;325:22–37.
[15] Weatheritt J, Sandberg R. The development of algebraic stress models using a novel evolutionary algorithm. Int J Heat Fluid Flow 2017;68:298–318.
[16] Schmelzer M, Dwight RP, Cinnella P. Machine learning of algebraic stress models using deterministic symbolic regression. Flow Turbul Combust.
[17] Raissi M, Perdikaris P, Karniadakis G. Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J Comput Phys 2019;378:686–707.
[18] Raissi M, Wang Z, Triantafyllou MS, Karniadakis GE. Deep learning of vortex-induced vibrations. J Fluid Mech 2019;861:119–37.
[19] Zeng Y, Wu J-L, Xiao H. Enforcing physical constraints on generative adversarial networks with application to fluid flows. Submitted to Journal of Computational Physics. Also available as arxiv:1905.06671.
[20] Wu J-L, Kashinath K, Albert A, Chirila D, Prabhat M, Xiao H. J Comput Phys 2019. doi:10.1016/j.jcp.2019.109209.
[21] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: in: Advances in Neural Information Processing Systems; 2012. p. 1097–105.
[22] Deng J, Dong W, Socher R, Li L-J, Li K, Fei-Fei L. Imagenet: a large-scale hierarchical image database. 2009. In: CVPR09.
[23] Kim J, Moin P, Moser R. Turbulence statistics in fully developed channel flow at low Reynolds number. J Fluid Mech 1987;177:133–66.
[24] Lee M, Moser RD. Direct numerical simulation of turbulent channel flow up to Re=5200. J Fluid Mech 2015;774:395–415.
[25] AGARD. A selection of test cases for the validation of Large-Eddy Simulations of turbulent flows. AGARD advisory report no. 345, data: http://torroja.dmt.upm.es/turbdata/agard/.
[26] Rumsey C.L.. NASA Langley turbulence modeling portal. 2018. https://turbmodels.larc.nasa.gov.
[27] ERCOFTAC. ERCOFTAC classic collection database. http://cfd.mace.manchester.ac.uk/ercoftac/.
[28] University J.H. Johns Hopkins turbulence database (JHTDB) site. http://turbulence.pha.jhu.edu.
[29] Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, et al. A public turbulence database cluster and applications to study lagrangian evolution of velocity increments in turbulence. J Turbul 2008;9:N31.
[30] Pinelli A, Uhlmann M, Sekimoto A, Kawahara G. Reynolds number dependence of mean flow structure in square duct turbulence. J Fluid Mech 2010;644:107–22.
[31] Uhlmann M. Fully-developed, pressure-driven flow of an incompressible, isothermal fluid through a straight duct with square cross section: data from DNS. http://www.ifh.kit.edu/dns_data/duct/.
[32] Maaß C, Schumann U. Direct numerical simulation of separated turbulent flow over a wavy boundary. In: in: Flow Simulation with High-Performance Computers II. Springer; 1996. p. 227–41.
[33] Bentaleb Y, Lardeau S, Leschziner MA. Large-eddy simulation of turbulent boundary layer separation from a rounded step. Journal of Turbulence 2012;13. N4
[34] Laval J-P, Marquillie M. Direct numerical simulations of converging–diverging channel flow. In: in: Progress in Wall Turbulence: Understanding and Modeling. Springer; 2011. p. 203–9.
[35] Breuer M, Peller N, Rapp C, Manhart M. Flow over periodic hills: numerical and experimental study in a wide range of Reynolds numbers. Comput Fluids 2009;38(2):433–57.
[36] Kutz JN. Deep learning in fluid dynamics. J Fluid Mech 2017;814:1–4.
[37] R M, P D. Large-eddy simulation of turbulent flow over a parametric set of bumps. J Fluid Mech 2019;866:503–25.
[38] Xiao H, Jenny P. A consistent dual-mesh framework for hybrid LES/RANS modeling. J Comput Phys 2012;231(4):1848–65.
[39] Xia Z, Shi Y, Hong R, Xiao Z, Chen S. Constrained large-eddy simulation of separated flow in a channel with streamwise-periodic constrictions. J Turbul 2013;14(1):1–21.
[40] Fröhlich J, Mellen CP, Rodi W, Temmerman L, Leschziner MA. Highly resolved large-eddy simulation of separated flow in a channel with streamwise periodic constrictions. J Fluid Mech 2005;526:19–66.
[41] Rapp C, Manhart M. Flow over periodic hills: an experimental study. Exp Fluids 2011;51(1):247–69.
[42] Gloerfelt X, Cinnella P. Large eddy simulation requirements for the flow over periodic hills, flow. Turbul Combust 2019;103(1):55–91.
[43] Gloerfelt X, Cinnella P. Benchmark database: 2D periodic hill flow. https://www.researchgate.net/publication/315413324_Benchmark_database_2D_periodic_hill_flow.
[44] Wu J-L, Sun R, Laizet S, Xiao H. Representation of stress tensor perturbations with application in machine-learning-assisted turbulence modeling. Comput Methods Appl Mech Eng 2019;346:707–26.
[45] Wu J-L, Xiao H, Sun R, Wang Q. Reynolds-averaged Navier–Stokes equations with explicit data-driven Reynolds stress closure can be ill-conditioned. J Fluid Mech 2019;869:553–86.
[46] Kravchenko AG, Moin P. On the effect of numerical errors in large eddy simulation of turbulent flows. J Comp Phys 1997;131:310–22.
[47] Laizet S. Incompressible Navier-Stokes equations solver with multiple scalar transport equations. https://github.com/xcompact3d/Incompact3d.
[48] Gautier R, Laizet S, Lamballais E. A DNS study of jet control with microjets using an immersed boundary method. Int J of Comput Fluid Dyn 2014;28(6–10):393–410.
[49] Laizet S, Lamballais E. High-order compact schemes for incompressible flows: a simple and efficient method with the quasi-spectral accuracy. J Comp Phys 2009;228:5989–6015.
[50] Laizet S, Li N. Incompact3d: a powerfull tool to tackle turbulence problems with up to $O(10^5)$ computational cores. Int J Heat and Fluid Flow 2011;67:1735–57.
[51] Espath L, Pinto L, Laizet S, Silvestrini J. Two-and three-dimensional direct numerical simulation of particle-laden gravity currents. ComputGeosci 2014;63:9–16.
[52] Schuch FN, Pinto LC, Silvestrini JH, Laizet S. Three-dimensional turbulence-resolving simulations of the plunge phenomenon in a tilted channel. J Geophys Research 2018;123(7):4820–32.

[53] Lucchese LV, Monteiro LR, Schettini EBC, Silvestrini JH. Direct numerical simulations of turbidity currents with evolutive deposit method. Considering topography updates during the simulation. Computers & Geosciences 2019;133. 104306
[54] Mahfoze O, Laizet S. Skin-friction drag reduction in a channel flow with streamwise-aligned plasma actuators. Int J Heat Fluid Flow 2017;66:83–94.
[55] Yao J, Chen X, Hussain F. Drag control in wall-bounded turbulent flows via spanwise opposed wall-jet forcing. J Fluid Mech 2018;852:678–709.
[56] Mahfoze O, Moody A, Wynn A, Whalley R, Laizet S. Reducing the skin-friction drag of a turbulent boundary-layer flow with low-amplitude wall-normal blowing within a bayesian optimization framework. Phys Rev Fluids 2019;4(9):94601.
[57] Dairay T, Fortuné V, Lamballais E, Brizzi L. Les of a turbulent jet impinging on a heated wall using high-order numerical schemes. Int J Heat Fluid Flow 2014;50:177–87.
[58] Dairay T, Fortuné V, Lamballais E, Brizzi LE. Direct numerical simulation of a turbulent jet impinging on a heated wall. J Fluid Mech 2015;764:362–94.
[59] Deskos G, Laizet S, Piggott MD. Development and validation of the higher-order finite-difference wind farm simulator, winc3d. in: 3rd international conference on renewable energies offshore (renew2018). Portugal: Lisbon; 2018.
[60] Deskos G, Laizet S, Piggott MD. Turbulence-resolving simulations of wind turbine wakes. Renew Energy 2019;134:989–1002.
[61] Dairay T, Obligado M, Vassilicos JC. Non-equilibrium scaling laws in axisymmetric turbulent wakes. J Fluid Mech 2015;781:166–95.
[62] Obligado M, Dairay T, Vassilicos JC. Nonequilibrium scalings of turbulent wakes. Phys Rev Fluids 2016;1(4):44409.
[63] Zhou Y, Vassilicos J. Related self-similar statistics of the turbulent/non-turbulent interface and the turbulence dissipation. J Fluid Mech 2017;821:440–57.
[64] Xiao H, Wu J, Laizet S, Duan L. Flow over periodic hills of parameterized geometries: Example code and dataset for data-driven turbulence modeling, see also. 2019. https://github.com/xiaoh/para-database-for-PIML.
[65] Ling J, Templeton J. Evaluation of machine learning algorithms for prediction of regions of high Reynolds-averaged Navier–Stokes uncertainty. Physics of Fluids 2015;27(8):085103. (1994-present)
[66] Wang J-X, Huang J, Duan L, Xiao H. Prediction of Reynolds stresses in high-Mach-number turbulent boundary layers using physics-informed machine learning. Theor Comput Fluid Dyn 2019;33(1):1–19.
[67] Zhu L, Zhang W, Kou J, Liu Y. Machine learning methods for turbulence modeling in subsonic flows around airfoils. Phys Fluids 2019;31(1):15105.
[68] Emory M, Pecnik R, Iaccarino G. Modeling structural uncertainties in Reynolds-averaged computations of shock/boundary layer interactions. AIAA Paper 2011:479.
[69] Banerjee S, Krahl R, Durst F, Zenger C. Presentation of anisotropy properties of turbulence, invariants versus eigenvalue approaches. J Turbul 2007;8(32):589–610.
[70] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[71] Hornik K, Stinchcombe M, White H. Multilayer feedforward networks are universal approximators. Neural Netw 1989;2(5):359–66.
[72] Friedman J, Hastie T, Tibshirani R. The elements of statistical learning. Berlin: Springer; 2001.