Predictions of Reynolds and Nusselt numbers in turbulent convection using machine-learning models

Shashwat Bhattacharya,1, a) Mahendra K. Verma,2 and Arnab Bhattacharya3

1) Institut für Thermo-und Fluidodynamik, Technische Universität Ilmenau, Postfach 100565, D-98684 Ilmenau, Germany
2) Department of Physics, Indian Institute of Technology Kanpur, Kanpur 208016, India
3) Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur 208016, India

(Dated: 21 January 2022)

In this paper, we develop a multivariate regression model and a neural network model to predict the Reynolds number (Re) and Nusselt number in turbulent thermal convection. We compare their predictions with those of earlier models of convection: Grossmann-Lohse [Phys. Rev. Lett. 86, 3316 (2001)], revised Grossmann-Lohse [Phys. Fluids 33, 015113 (2021)], and Pandey-Verma [Phys. Rev. E 94, 053106 (2016)] models. We observe that although the predictions of all the models are quite close to each other, the machine learning models developed in this work provide the best match with the experimental and numerical results.

I. INTRODUCTION

Thermal convection is encountered in natural phenomena, such as stars and planets, as well as in various engineering applications. A simplified version of thermal convection is Rayleigh-Bénard convection (RBC), which is a model of a fluid enclosed between two horizontal plates with the bottom plate kept at a higher temperature than the top one. RBC is governed primarily by two nondimensional parameters: the Rayleigh number \( Ra \), which is the ratio of the buoyancy and the dissipative force, and the Prandtl number \( Pr \), which is the ratio of kinematic viscosity and thermal diffusivity of the fluid. Two important response parameters of RBC are the Nusselt number \( Nu \) and the Reynolds number \( Re \), which are respective measures of large-scale heat transport and velocity in turbulent RBC. The Reynolds number is defined as \( Re = Ud/\nu \), where \( U \) is the large-scale velocity, \( \nu \) is the kinematic viscosity, and \( d \) is the distance between the thermal plates. The Nusselt number is given by \( Nu = 1 + \langle u_2 T \rangle/\kappa \Delta d^{-1} \), where \( u_2 \) is the vertical component of velocity, \( T \) is the temperature field, \( \kappa \) is the thermal diffusivity, \( \Delta \) is the temperature difference between the thermal plates, and \( \langle \cdot \rangle \) denotes the volume average.

Researchers have attempted to model \( Re \) and \( Nu \) on the governing parameters, \( Ra \) and \( Pr \). Early theoretical, numerical, and experimental studies of RBC reveal a power-law scaling of \( Nu \) and \( Re \), i.e., \( Nu \sim Ra^{\alpha}Pr^\beta \) and \( Re \sim Ra^{\gamma}Pr^\delta \). The exponents \( \alpha, \beta, \gamma, \) and \( \delta \) vary for different regimes of \( Pr \) and \( Ra \). For the scaling of \( Nu \), the exponent \( \alpha \) ranges from 1/4 for \( Pr \ll 1 \) to approximately 1/3 for \( Pr \gtrsim 1 \). However, \( Nu \) has a weak dependence on \( Pr \), with the exponent \( \beta \) ranging from approximately zero for \( Pr \gtrsim 1 \) to 0.14 for small \( Pr \). Regarding the scaling of \( Re \), the exponent \( \gamma \) was observed to be approximately 2/5 for \( Pr \ll 1 \), 1/2 for \( Pr \sim 1 \), and 3/5 for \( Pr \gg 1 \). \( \delta \) has been observed to range from -0.7 for \( Pr \lesssim 1 \) to -0.95 for \( Pr \gtrsim 1 \). In the limit of very large \( Ra \) (the ultimate regime), Kraichnan argued that \( Nu \sim \sqrt{RaPr} \) and \( Re \sim \sqrt{Ra/Pr} \) for \( Pr \leq 0.15 \), and \( Nu \sim \sqrt{RaPr^{1/2}} \) and \( Re \sim \sqrt{RaPr^{3/2}} \) for \( 0.15 < Pr \leq 1 \), with logarithmic corrections. However, the existence of the aforementioned regime in RBC is still under debate.

Since the scaling of \( Re \) and \( Nu \) depends crucially on the regime of \( Ra \) and \( Pr \), researchers felt a need for a unified model that encompasses all the regimes. Grossmann and Lohse (henceforth referred to as GL) derived the first set of formulae for \( Re \) and \( Nu \). Recently, Bhattacharya, Verma, and Samtaney revised the GL model by incorporating the observed suppression in the dissipation rates in turbulent convection. Pandey and Verma also derived formulae for \( Nu \) and \( Re \). All the above models have been largely successful in predicting \( Nu \) and \( Re \) over a range of \( Ra \) and \( Pr \).

In this paper, we employ machine-learning (ML) tools to construct models for predicting \( Re \) and \( Nu \). This approach, which is increasingly gaining popularity in fluid mechanics, involves building and improving prediction algorithms by “learning” from the existing data. Hence, in the present work, we do not delve much into the physics behind \( Re \) and \( Nu \) relations; instead, we use the data from previous works to develop the prediction models. We construct a multivariate regression (MR) model and a neural network (NN) model for the predictions of \( Nu \) and \( Re \). We compare the predictions of the above ML models with those of the earlier models. We observe that the predictions of all the models are close to each other, but the machine learning models developed in this study provide better match with the experimental and numerical results.

The outline of the paper is as follows. In Sec. II, we briefly describe GL and Pandey-Verma models. In
Sec. III, we discuss the MR and NN models developed in the present study. We compare the predictions of the above models with the existing ones in Sec. IV. We conclude in Sec. V.

II. A SUMMARY OF EXISTING MODELS

In this section, we summarize GL and Pandey-Verma models briefly.

A. Grossmann-Lohse (GL) model

Grossmann and Lohse\textsuperscript{41,42} provided the first unifying model for thermal convection. Grossmann and Lohse\textsuperscript{41,42} derived this model by splitting the total viscous and thermal dissipation rates ($\tilde{D}_u$ and $\tilde{D}_T$ respectively) into bulk and the boundary layer (BL) contributions as

$$\tilde{D}_u = \tilde{D}_{u,\text{bulk}} + \tilde{D}_{u,\text{BL}}, \quad (1)$$

$$\tilde{D}_T = \tilde{D}_{T,\text{bulk}} + \tilde{D}_{T,\text{BL}}. \quad (2)$$

Using the Prandtl-Blasius theory of boundary layers, and assuming homogeneous and isotropic turbulence-like properties of dissipation rates in the bulk, Grossmann and Lohse\textsuperscript{41,42} expressed the bulk and boundary layer contributions in terms of Re, Nu, and Pr. These contributions were then substituted in the exact relations relating the total dissipation rates with the Nusselt number$^8$:

$$\tilde{D}_u = \frac{\nu^3}{d^2}(\text{Nu} - 1)\frac{Ra}{Pr^2}, \quad (3)$$

$$\tilde{D}_T = \frac{\kappa^2}{d^2}\text{Nu}. \quad (4)$$

Following the above procedure, Grossmann and Lohse arrived at general expressions for Nu and Re valid for all Pr and Ra. The formulae for Re and Nu were later updated by Stevens \textit{et al.}\textsuperscript{54} using more up-to-date data of RBC. The GL model provides good estimates for the Nusselt and Reynolds numbers for a wide range of Ra and Pr.

B. Revised Grossmann-Lohse (RGL) model

Although the GL model has been quite successful in explaining past experimental and numerical results, it based on certain assumptions that are not very accurate for RBC. First, the model assumes that the viscous and thermal dissipation rates in the bulk scale as $U^3/d$ and $U\Delta^2/d$ (for $Pr \leq 1$) respectively, as in passive scalar turbulence with open boundaries\textsuperscript{55,56}. Recent studies, however, show that these rates get an additional correction of approximately $Ra^{-0.2}$ for $Pr \sim 1$\textsuperscript{10,26,29,30,44,45}. The above correction is because of the inhibition of nonlinear interactions due to the presence of walls and buoyancy\textsuperscript{6,29,30,57}. In addition, the above studies reveal that the thickness of the viscous boundary layer in RBC deviates from $Re^{-1/2}$, contrary to what is assumed in GL model\textsuperscript{44,58,59}.

Recently, Bhattacharya, Verma, and Samtaney\textsuperscript{43} incorporated the above corrections in the dissipation rates and the boundary layer thickness in the equations of the GL model. In the revised model (henceforth referred to as Revised Grossmann-Lohse or RGL model), the Reynolds and Nusselt numbers are obtained by solving a cubic polynomial equation consisting of four functions $[f_i(Ra, Pr)]$ that are prefactors for the dissipation rates in the bulk and boundary layers. The functional forms of the prefactors in the RGL model were determined using multivariate linear regression on 60 sets of simulation data of RBC. Bhattacharya, Verma, and Samtaney\textsuperscript{43} observed that the predictions of the RGL model were marginally better than the GL model, especially for extreme Prandtl numbers\textsuperscript{43}.

C. Pandey-Verma (PV) model

Pandey and Verma\textsuperscript{29,30} used a different approach from the aforementioned models. To compute the Péclet number ($Pe = RePr$) as a function of Ra and Pr, Pandey and Verma\textsuperscript{29,30} constructed a model (henceforth referred to as PV model) by estimating the relative strengths of the various terms of the momentum equation of RBC using data from their numerical simulations. Based on these estimates, they arrived at a quadratic formula for the Péclet number. The predictions of Re by the PV model were as accurate as those by the GL model\textsuperscript{16}. Pandey and Verma\textsuperscript{29,30} also derived a general formula for the Nusselt number using correlations. Note that the relation for Nu was obtained using simulation data for $Pr \geq 1$ and thus is not accurate for $Pr \ll 1$ regime as shown later in this paper.

The main aspects of GL, RGL, and PV models are summarized in Table I. In the next section, we describe the machine-learning models developed in the present study.

III. MACHINE-LEARNING MODELS

Having discussed three important models of the past, we now describe two machine-learning models to determine the Nusselt and Reynolds numbers for given Rayleigh and Prandtl numbers. The first one, referred to as Multivariate Regression model (MR), employs linear regression to the available data directly. The second approach, referred to as Neural Network (NN) model, employs artificial neural networks for the predictions.

To construct the ML models, we use 62 sets of simulation data as training datasets. Sixty of these datasets were generated by Bhattacharya, Verma, and Samtaney\textsuperscript{43} using high-resolution direct numerical simulations of RBC. The Rayleigh number for these datasets
ranges from $5 \times 10^5$ to $5 \times 10^8$, and the Prandtl number ranges from 0.02 to 100. All the above simulations were conducted using the finite-difference code SARAS\textsuperscript{60,61} on a cubical domain of unit dimension with no-slip walls on all the sides. Isothermal boundary conditions were imposed on the horizontal walls. The grid resolution varied between $257^3$ to $1025^3$, depending on the resolution requirements. For further details on the simulations and their validations, refer to Bhattacharya, Verma, and Samtaney\textsuperscript{43,62}.

Additionally, we conduct two simulations of RBC for $Pr = 0.02$ with $Ra = 10^9$ and $2 \times 10^9$; these simulations were performed on a $513^3$ grid. Here, we remark that the output parameters — Re and Nu — are global quantities that are based on velocity and heat fluxes that are averaged over the entire domain. Further, Nu and Re used for training the model are not computed for a specific timeframe; rather they are computed for multiple timeframes (ranging from 14 to 263; see Bhattacharya, Verma, and Samtaney\textsuperscript{43} for details) and then averaged over these timeframes. Thus, we take care of the spatial and temporal fluctuations of quantities that are typically encountered in turbulent flows. The parameters of all the data samples, along with their corresponding Re and Nu, are tabulated in Table II.

### A. Multivariate regression (MR) model

We construct multivariate regression (MR) model using machine-learning software WEKA\textsuperscript{64} and obtain the functional forms of Re and Nu. We look for a power-law relation for Re and Nu of the form $ARa^\alpha Pr^\beta$, take logarithms of this expression, and employ linear regression to obtain $A$, $\alpha$, and $\beta$. However, as discussed in Sec. I, the Pr and Ra dependences of Re and Nu are different at small, moderate, and large Prandtl number regimes.

Hence, the aforementioned power-law will not be accurate of all the regimes of Pr.

To overcome the above problem, we split our parameter space into three regimes — small Pr, moderate Pr, and large Pr — such that in each regime, the quantities $A$, $\alpha$, and $\beta$ do not vary significantly. We obtain the boundaries (or breakpoints) between the regimes as follows. For $Ra = 10^6$ and $10^7$, we plot the logarithms of Nu and Re versus the logarithm of Pr and fit a piecewise linear curve with three segments [see Figs. 1(a) and (b)]. We employ the “PiecewiseLinFit” function from the “pwlf” module of python\textsuperscript{65}. The above module determines the breakpoints of the piecewise linear curve using an optimization process that solves a least squares fit several times for different breakpoint locations. The two breakpoints of Pr versus Nu and Pr versus Re curves determined by the “pwlf” module for $Ra = 10^6$ and $10^7$ are tabulated in Table III. Finally, we average the breakpoints over the two Rayleigh numbers and using this average, we obtain the parameter regimes for Re as

$$\text{Small Pr : } Pr \leq 0.45,$$
$$\text{Moderate Pr : } 0.45 \leq Pr \leq 10,$$
$$\text{Large Pr : } Pr \geq 10,$$

and the parameter regimes for Nu as

$$\text{Small Pr : } Pr \leq 0.25,$$
$$\text{Moderate Pr : } 0.25 \leq Pr \leq 4.6,$$
$$\text{Large Pr : } Pr \geq 4.6.$$  

To ensure continuity across the parameter regimes, we employ six matching functions $H_i(Pr); i = 1$ to 6. The functions $H_1$, $H_2$, $H_3$, $H_4$, $H_5$, and $H_6$ are such that they become unity inside the regimes defined by $Pr < 0.45$, $0.45 < Pr < 10$, $Pr > 10$, $Pr < 0.25$, $0.25 < Pr < 4.6$,
TABLE II. Details of the training dataset for the Multivariate Regression (MR) and the Neural Network (NN) models. The input parameters are the Prandtl number (Pr) and the Rayleigh number (Ra), and the output parameters are the Nusselt number (Nu) and the Reynolds number (Re).

| S. No | Pr | Ra  | Nu  | Re  |
|-------|----|-----|-----|-----|
| 1     | 0.02 | $10^8$ | 3.01 | 1180 |
| 2     | 0.02 | $2 \times 10^5$ | 3.52 | 1640 |
| 3     | 0.02 | $5 \times 10^5$ | 4.48 | 2440 |
| 4     | 0.02 | $1 \times 10^6$ | 5.78 | 3200 |
| 5     | 0.02 | $2 \times 10^6$ | 6.91 | 4290 |
| 6     | 0.02 | $5 \times 10^6$ | 8.85 | 6650 |
| 7     | 0.02 | $1 \times 10^7$ | 10.3 | 9420 |
| 8     | 0.1  | $5 \times 10^5$ | 6.11 | 749  |
| 9     | 0.1  | $1 \times 10^6$ | 7.34 | 1030 |
| 10    | 0.1  | $2 \times 10^6$ | 8.85 | 1380 |
| 11    | 0.1  | $5 \times 10^6$ | 11.3 | 2090 |
| 12    | 0.1  | $1 \times 10^7$ | 13.9 | 2870 |
| 13    | 0.1  | $2 \times 10^7$ | 16.4 | 3870 |
| 14    | 0.1  | $5 \times 10^7$ | 20.8 | 6020 |
| 15    | 0.1  | $1 \times 10^8$ | 26.7 | 8140 |
| 16    | 0.5  | $1 \times 10^6$ | 8.38 | 285  |
| 17    | 0.5  | $3 \times 10^6$ | 11.4 | 482  |
| 18    | 0.5  | $1 \times 10^7$ | 15.9 | 874  |
| 19    | 0.5  | $3 \times 10^7$ | 21.6 | 1480 |
| 20    | 0.5  | $1 \times 10^8$ | 30.6 | 2610 |
| 21    | 1    | $1 \times 10^6$ | 8.18 | 147  |
| 22    | 1    | $2 \times 10^6$ | 10.1 | 213  |
| 23    | 1    | $5 \times 10^6$ | 13.3 | 340  |
| 24    | 1    | $1 \times 10^7$ | 16.3 | 491  |
| 25    | 1    | $2 \times 10^7$ | 19.8 | 702  |
| 26    | 1    | $5 \times 10^7$ | 26.0 | 1100 |
| 27    | 1    | $1 \times 10^8$ | 31.4 | 1530 |
| 28    | 1    | $2 \times 10^8$ | 38.6 | 2170 |
| 29    | 1    | $5 \times 10^8$ | 49.2 | 3330 |
| 30    | 1    | $1 \times 10^9$ | 61.2 | 4700 |
| 31    | 1    | $2 \times 10^9$ | 76.8 | 6580 |

| S. No | Pr | Ra  | Nu  | Re  |
|-------|----|-----|-----|-----|
| 32    | 6.8 | $1 \times 10^6$ | 7.90 | 24.9 |
| 33    | 6.8 | $2 \times 10^6$ | 9.46 | 35.6 |
| 34    | 6.8 | $5 \times 10^6$ | 12.99 | 59.7 |
| 35    | 6.8 | $1 \times 10^7$ | 15.99 | 82.2 |
| 36    | 6.8 | $2 \times 10^7$ | 19.59 | 128 |
| 37    | 6.8 | $5 \times 10^7$ | 26.19 | 217 |
| 38    | 6.8 | $1 \times 10^8$ | 31.69 | 314 |
| 39    | 6.8 | $2 \times 10^8$ | 38.59 | 452 |
| 40    | 6.8 | $5 \times 10^8$ | 50.59 | 729 |
| 41    | 6.8 | $1 \times 10^9$ | 65.7 | 1070 |
| 42    | 6.8 | $2 \times 10^9$ | 77.0 | 1520 |
| 43    | 6.8 | $5 \times 10^9$ | 101 | 2400 |
| 44    | 6.8 | $1 \times 10^{10}$ | 167 | 140 |
| 45    | 6.8 | $2 \times 10^{10}$ | 20.2 | 21.1 |
| 46    | 6.8 | $5 \times 10^{10}$ | 26.4 | 35.2 |
| 47    | 6.8 | $1 \times 10^{11}$ | 31.8 | 50.8 |
| 48    | 6.8 | $2 \times 10^{11}$ | 38.6 | 74.6 |
| 49    | 6.8 | $5 \times 10^{11}$ | 51.8 | 137 |
| 50    | 6.8 | $1 \times 10^{12}$ | 61.5 | 202 |
| 51    | 6.8 | $2 \times 10^{12}$ | 79.4 | 180 |
| 52    | 6.8 | $5 \times 10^{12}$ | 10.4 | 2.78 |
| 53    | 6.8 | $1 \times 10^{13}$ | 13.9 | 4.90 |
| 54    | 6.8 | $2 \times 10^{13}$ | 16.8 | 7.02 |
| 55    | 6.8 | $5 \times 10^{13}$ | 20.1 | 9.91 |
| 56    | 6.8 | $1 \times 10^{14}$ | 26.1 | 17.1 |
| 57    | 6.8 | $2 \times 10^{14}$ | 31.8 | 26.0 |
| 58    | 6.8 | $5 \times 10^{14}$ | 39.1 | 37.5 |
| 59    | 6.8 | $1 \times 10^{15}$ | 49.7 | 71.4 |

TABLE III. For Ra = $10^6$ and $10^7$: boundaries separating the small, moderate and large Pr regimes. Here, Pr$_{1,Re}$ = boundary between the small and moderate Pr regimes for the predictions of Re; Pr$_{u,Re}$ = boundary between the moderate and large Pr regimes for the predictions of Re; Pr$_{1, Nu}$ = boundary between the small and moderate Pr regimes for the predictions of Nu; Pr$_{u, Nu}$ = boundary between the moderate and large Pr regimes for the predictions of Nu.

| Ra     | Pr$_{1,Re}$ | Pr$_{u,Re}$ | Pr$_{1, Nu}$ | Pr$_{u, Nu}$ |
|--------|------------|------------|-------------|-------------|
| $10^6$ | 0.25       | 6.8        | 0.28        | 2.5         |
| $10^7$ | 0.65       | 14         | 0.22        | 6.8         |

Average 0.45 10 0.25 4.6

and Pr > 4.6, respectively, and become negligible outside their regimes. The value of these functions is 1/2 at the boundaries of their respective regimes. The following expressions of $H_i$ satisfy the aforementioned conditions:

$$H_1(Pr) = \frac{1}{1 + e^{-k_1(0.45 - Pr)}} \quad (5)$$

$$H_2(Pr) = \frac{1}{1 + e^{-k_1(Pr - 0.45)} - \frac{1}{1 + e^{-k_2(Pr - 10)}}} \quad (6)$$

$$H_3(Pr) = \frac{1}{1 + e^{-k_2(Pr - 10)}} \quad (7)$$

$$H_4(Pr) = \frac{1}{1 + e^{-k_1(0.25 - Pr)}} \quad (8)$$

$$H_5(Pr) = \frac{1}{1 + e^{-k_1(Pr - 0.25)} - \frac{1}{1 + e^{-k_2(Pr - 4.6)}}} \quad (9)$$

$$H_6(Pr) = \frac{1}{1 + e^{-k_2(Pr - 4.6)}} \quad (10)$$

where $k_1 = 50$ and $k_2 = 7$. The above values of $k_1$ and $k_2$ provide the optimal steepness for the $H_i$ curves at the boundary points. See Fig. 2 for an illustration of the behavior of the matching functions $H_1$, $H_2$, and $H_3$. We then determine the logarithms of the matching functions $A$ and the exponents $\alpha$ and $\beta$ for each regime using the linear
regression function of WEKA, and then combine these regimes using the matching functions $H_i$. We make use of cross-validation technique to get an insight on how the MR model will generalise to an independent dataset and thus to estimate how accurately it will perform in practice. We employ ten-fold cross-validation, which involves random partitioning of the complete dataset into 10 subdatasets of equal size. Of these 10 subdatasets, a single subdataset is retained as the validation data for testing the model, and the remaining 9 subdatasets are used as training data. This process is then repeated 10 times, with each of the 10 subdatasets used exactly once as the validation data. The mean of the absolute percentage error, given by

$$D = \frac{|\text{Predicted value} - \text{Actual value}|}{\text{Actual value}} \times 100,$$  \hspace{1cm} (11)

is computed for all the samples of each fold and then averaged over all the folds. The above average is observed to be 4% for the predictions of both Re and Nu, thus ensuring that the model is robust. After having done the cross-validation, we invoke the learning algorithm for a final (11th) time on the entire dataset and obtain the following expressions for $Re$ and $Nu$.

$$Re = 0.40H_1Ra^{0.45}Pr^{-0.72} + 0.11H_3Ra^{0.52}Pr^{-0.86} + 0.055H_3Ra^{0.58}Pr^{-0.99}, \hspace{1cm} (12)$$

$$Nu = 0.24H_1Ra^{0.27}Pr^{0.16} + 0.15H_3Ra^{0.29}Pr^{0.018} + 0.13H_6Ra^{0.30}Pr^{0.0084}. \hspace{1cm} (13)$$

Thus, interested researchers can use the above equations to obtain the Nusselt and Reynolds numbers for an arbitrary set of Rayleigh and Prandtl numbers.

The above training algorithm was run on a personal computer consisting of an 8-gigahertz Core i5 processor with 8 gigabytes of RAM. The time taken for executing the training algorithm is 0.3 second for both Re and Nu. It is important to note that the present model is an application of machine-learning model to the raw data itself, in contrast to RGL model where ML model was used for parameter estimation of the prefactors of the dissipation equations [Eqs. (1) and (2)].

### B. Neural Network (NN) model

Lastly, we discuss the neural network model. Neural network is known to be a more robust prediction algorithm than regression. This is due to the intrinsic nonlinearity of neural networks and also by the virtue of universal approximation theorems associated with these networks. A schematic diagram of the neural networks for predicting Re and Nu are exhibited in Fig. 3(a,b). We employ the Keras module of Python to construct the neural networks. The input features for both the networks are the logarithms of Rayleigh and Prandtl numbers. The networks consist of one hidden layer, with 6 nodes for Re predictions and 16 nodes for Nu predictions. We arrived at the above configuration after trial and error as it provided us the best results. It must be noted from our discussion in Sec. I that the range of the scaling

![Fig. 1](image1.png)

FIG. 1. (color online) For $Ra = 10^6$ (black) and $Ra = 10^7$ (red): (a) plots of Re versus Pr, and (b) plots of Nu versus Pr. Also shown are the piecewise linear fits for Re and Nu with the breakpoints (exhibited by vertical dashed lines) computed using the “pwlf” module of python.

![Fig. 2](image2.png)

FIG. 2. (color online) Plots of the matching functions $H_i(Pr)$ vs. Pr. $H_1$, $H_2$, and $H_3$ become unity in the regimes given by $Pr < 0.45$, $0.45 < Pr < 10$, and $Pr > 10$ respectively. They attain the value of 1/2 at the regime boundaries and become negligible outside their respective regimes.
exponent for Nu ($0.25 \leq \alpha \leq 0.33$) is shorter than that for Re ($0.40 \leq \gamma \leq 0.60$); hence, more care is needed for predicting Nu. Thus, more neurons are required in the neural network for Nu.

The hidden layers of both the networks are activated by the Exponential Linear Unit (ELU) function. The networks are set to minimize the mean-squared error between the actual values in the training set and the predictions. We employ the Adam algorithm to optimize our models. The networks are trained using the mini-batch gradient descent algorithm with a batch size of 8, that is, eight training examples are used to estimate the error gradient. The gradient descent algorithm is run for 1000 epochs. The training algorithms were run on the same personal computer on which the MR algorithm was executed. The computational time taken to train the model is 11 seconds for both Re and Nu; thus the neural networks take more time than the multivariate regression for training. However, the time required for training is still not very large and the necessary computations can easily be carried out on a laptop. The relevant segment and functions of the code describing the model (for Re) are given below.

```python
def baseline_model():
    model = Sequential()
    # Specify the number of input parameters (input_dim),
    # which in this case is 2. Add the hidden layer with
    # 6 nodes, activated using ELU function.''
    model.add(Dense(6, input_dim=2,
    kernel_initializer = 'normal', activation = 'elu'))
    # Add the output layer containing one output node.
    model.add(Dense(1, kernel_initializer = 'normal'))
    # Specify the loss function and the optimizer
    model.compile(loss='mean_squared_error',
    optimizer='adam')
    return model
```

The relevant code segment for describing the model for Nu is similar to that for Re, except that the number of nodes in the hidden layer is specified to be 16. The user can pass a two-dimensional array of arbitrary Rayleigh and Prandtl numbers as an argument to the “estimator.predict()” function; this function returns the corresponding array for predictions. Refer to the supplemental materials for the complete codes (“suppl1.txt” for predicting Re and “suppl2.txt” for predicting Nu) and the input dataset (“suppl3.csv”).

We employ 10-fold cross-validation for our above models (similar to that done for the MR model described ear-
TABLE IV. Effect of adding regularizing hyperparameters to the neural networks on the performance of the models on test data. Here, $D_{Re}$ and $D_{Nu}$ respectively represent the percentage deviation of the predictions from the actual values of the test data.

| Regularizer hyperparameter | $D_{Re}$ | $D_{Nu}$ |
|---------------------------|---------|---------|
| None                      | 19.2%   | 9.76%   |
| 0.001 (Hidden layer)      | 20.4%   | 14.0%   |
| 0.001 (Hidden layer)      | 20.5%   | 14.4%   |
| 0.001 (Output layer)      |         |         |
| 0.01 (Hidden layer)       | 21.7%   | 25.8%   |
| 0.01 (Hidden layer)       | 23.5%   | 26.1%   |
| 0.01 (Output layer)       |         |         |

Finally, we remark that we also tried to use less width (that is, 6 for Re and 16 for Nu). Interestingly, we observed that the performance of the networks decreases with the increase in depth. This is likely because the deeper neural networks have a tendency to overfit and hence their predictions are less accurate.

The important aspects of the MR and NN models are summarized in Table I. In the next section, we discuss and compare the predictions of these models with respect to the GL, RGL, and PV models.

IV. PERFORMANCE OF THE ML MODELS

We test the performance of the multivariate regression and neural network models with the GL, RGL, and PV models. We compare their predictions for the numerical results of Bhattacharya, Verma, and Samtaney \(^\text{43}\) (Pr = 0.02, 0.1, 0.5, 1, 6.8, 50, and 100), Scheel and Schumacher \(^\text{10}\) (Pr = 0.005 and 0.02), along with the test dataset Wagner and Shishkina \(^\text{14}\) (Pr = 0.7), Emran and Schumacher \(^\text{26}\) (Pr = 0.7), Kaczorowski and Xia \(^\text{15}\) (Pr = 4.38), and Horn, Shishkina, and Wagner \(^\text{13}\) (Pr = 2547.9); and the experimental results of Cioni, Ciliberto, and Sommeria \(^\text{9}\) (Pr = 0.02), and Niemela \(\text{et al.}^{24}\) (Pr = 0.7). The simulations of Wagner and Shishkina \(^\text{14}\), Kaczorowski and Xia \(^\text{15}\), and Bhattacharya, Verma, and Samtaney \(^\text{43}\) involve a cubical cell, whereas the others employ a cylindrical cell. The aspect ratio of the RBC cell is unity in all the above works. We compute the absolute percentage error between the estimated and actual values as per Eq. (11). For every Pr, we average the aforementioned errors over all the Rayleigh numbers and list this average error in Table VI (for the predictions of Re) and Table VII (for the predictions of Nu). Note that for Pr = 0.02, we provide two entries in the above tables: one corresponding to the data of Bhattacharya, Verma, and Samtaney \(^\text{43}\) (which were used for training), and the other corresponding to those of Scheel and Schumacher \(^\text{10}\) and Cioni, Ciliberto, and Sommeria \(^\text{9}\). In Fig. 4(a,b), we plot the predictions of the normalized Re by the different models discussed in the paper along with the numerically computed values of Refs. 10, 13, 14, 26, 43. See Fig. 4(c, d) for the corresponding plots for Nu. Note that in the above figures, we normalize Nu with $Gr^{0.3}$, where $Gr = Ra/Pr$ is the Grashoff number. This is because Nu has only a weak dependence on Pr, due to which the Nu versus Ra curves for different Pr’s would have tended to overlap on each other. Further, we do not show the plots for Pr = 6.8 (only for Nu) and Pr = 0.5 in the above figures because their datapoints lie very close to those of Pr = 4.38 and 0.7 respectively.

A. Predictions of the Reynolds number

First, we analyse in detail the performance of the models in their predictions of the Reynolds number. From Ta-
TABLE V. Different trial configurations of neural networks and their corresponding performances. Here, HL = hidden layer, $D_{\text{Re}}$ = percentage deviation of the predictions from the actual Re, $D_{\text{Nu}}$ = percentage deviation of the predictions from the actual Nu.

| Configuration (Re) | $D_{\text{Re}}$ (training) | $D_{\text{Re}}$ (testing) | Configuration (Nu) | $D_{\text{Nu}}$ (training) | $D_{\text{Nu}}$ (testing) |
|-------------------|-----------------------------|---------------------------|-------------------|-----------------------------|---------------------------|
| 1 HL, 6 nodes per HL | 2.98% | 19.2% | 1 HL, 16 nodes per HL | 2.31% | 9.76% |
| 2 HL, 3 nodes per HL | 2.98% | 36.0% | 2 HL, 8 nodes per HL | 3.04% | 17.9% |
| 3 HL, 2 nodes per HL | 5.93% | 81.6% | 4 HL, 4 nodes per HL | 4.64% | 17.1% |

FIG. 4. (color online) Comparison between Grossmann and Lohse (GL, blue lines), Pandey and Verma (PV, orange lines), Revised Grossmann and Lohse (RGL, black lines), multivariate regression (MR, green lines), and the neural network (NN, red lines) models for the predictions of Re for (a) $Pr < 1$, and (b) $Pr \geq 1$; and of Nu for (c) $Pr < 1$, and (d) $Pr \geq 1$. The predictions are compared with the past numerical and experimental results for $Pr = 0.005$ (triangles), $Pr = 0.02$ (squares), $Pr = 0.1$ (pentagons), $Pr = 0.7$ (circles), $Pr = 1$ (left-pointing triangles), $Pr = 4.38$ (right-pointing triangles), $Pr = 6.8$ (hexagons), $Pr = 50$ (inverted triangles), and $Pr = 2547.9$ (diamonds). The filled markers correspond to the training dataset for the MR and NN models that includes the numerically computed values of Bhattacharya, Verma, and Samtaney.

Table VI and Figs. 4(a,b), we observe that the GL model’s predictions for Re are close to the actual values for moderate Prandtl numbers. Note, however, that Re is sensitive to the way it is computed (that is, based on root mean square velocity, maximum plume velocity, etc.).

The RGL model further improves the predictions of Re, especially for extreme Prandtl numbers. The PV model also marginally improves the predictions of the Reynolds number compared to the GL model but is not as accurate as the RGL model.
In Sec. IV A and Sec. IV B, we have shown that the ML models developed in this study exhibit better accuracy in predictions compared to GL and PV models. Further, although the ML models offer better predictions than the RGL model as well, the improvement is only marginal. In spite of the above, the ML models are more promising than the RGL model as they easier to construct provided we have sufficient training dataset. Fortunately, there is a large amount of datasets available in the literature (see, for example, Ahlers, Grossmann, and Lohse, Chinellà and Schumacher). Also, experiments and numerical simulations of the past provide a wide range of Ra and Pr for training the neural network. For example, Ahlers et al., He et al., and Urban et al. performed experiments for Ra as large as 10$^{15}$, while Niemela et al. derived their expression for Nu using the data from simulations with Pr $\geq 1$. As a result, the final expression for the Nusselt number did not contain any Pr dependence, which is applicable only for moderate and large Prandtl numbers. Hence, for small Prandtl numbers, the errors between the predictions and the actual Nu values are very high for the PV model. Thus, as far as the previously developed models are concerned, the RGL model exhibits the best performance.

The MR and NN models developed in this study exhibit very good accuracy in their predictions of Nu and are observed to mostly perform better than the GL and PV models. The predictions of the MR and NN models are even better, albeit marginally, than the RGL model. For small and moderate Pr’s (Pr $\lesssim 1$), both MR and NN models perform better than the RGL model, with the MR model marginally outperforming the NN model. For larger Pr’s, with the exception of Pr $< 2547.9$, the NN model performs better than the RGL model. The performances of the latter two models are comparable in the aforementioned regime. For Pr $= 2547.9$, the ML models, especially the NN model, do not perform as well as they do for the remaining Pr’s; this is possibly because of lack of data points for very large Pr’s.

### C. Significance of the ML models

Now, we will analyze the performances of the MR and NN models developed in the present study. Both the above models exhibit very good accuracy in their predictions of Re and provide significantly better predictions than the GL and the PV models. As exhibited in Table VI, the MR and NN models provide better predictions than even the RGL model for large Prandtl numbers. For small Pr’s, however, the MR model fares marginally worse than the NN and RGL models, with the accuracy of the latter two models being comparable. It must be noted, however, that the MR model performs better than the RGL model if we consider all the Prandtl numbers into account (as shown in Table VI).

### B. Predictions of the Nusselt number

Having analyzed the predictions of Re, we will now examine the performance of the models in predicting Nu. Figures 4(c,d) and Table VII show that the GL model predicts Nu with a good accuracy. The RGL model marginally improves the predictions of Nu for moderate Pr but significantly improves the predictions for extreme Pr. The PV model’s predictions of Nu are close to those by the GL model for moderate and large Pr. However, for small Prandtl numbers, Nu predicted by the PV model deviates significantly from the actual values. This is because Pandey and Verma derived their expression for Nu using the data from simulations with Pr $\geq 1$. As a result, the final expression for the Nusselt number did not contain any Pr dependence, which is applicable only for moderate and large Prandtl numbers. Hence, for small Prandtl numbers, the errors between the predictions and the actual Nu values are very high for the PV model. Thus, as far as the previously developed models are concerned, the RGL model exhibits the best performance.

The MR and NN models developed in this study exhibit very good accuracy in their predictions of Nu and are observed to mostly perform better than the GL and PV models. The predictions of the MR and NN models are even better, albeit marginally, than the RGL model. For small and moderate Pr’s (Pr $\lesssim 1$), both MR and NN models perform better than the RGL model, with the MR model marginally outperforming the NN model. For larger Pr’s, with the exception of Pr $< 2547.9$, the NN model performs better than the RGL model. The performances of the latter two models are comparable in the aforementioned regime. For Pr $= 2547.9$, the ML models, especially the NN model, do not perform as well as they do for the remaining Pr’s; this is possibly because of lack of data points for very large Pr’s.

### C. Significance of the ML models

Now, we will analyze the performances of the MR and NN models developed in the present study. Both the above models exhibit very good accuracy in their predictions of Re and provide significantly better predictions than the GL and the PV models. As exhibited in Table VI, the MR and NN models provide better predictions than even the RGL model for large Prandtl numbers. For small Pr’s, however, the MR model fares marginally worse than the NN and RGL models, with the accuracy of the latter two models being comparable. It must be noted, however, that the MR model performs better than the RGL model if we consider all the Prandtl numbers into account (as shown in Table VI).

### B. Predictions of the Nusselt number

Having analyzed the predictions of Re, we will now examine the performance of the models in predicting Nu. Figures 4(c,d) and Table VII show that the GL model predicts Nu with a good accuracy. The RGL model marginally improves the predictions of Nu for moderate Pr but significantly improves the predictions for extreme Pr. The PV model’s predictions of Nu are close to those by the GL model for moderate and large Pr. However, for small Prandtl numbers, Nu predicted by the PV model deviates significantly from the actual values. This is because Pandey and Verma derived their expression for Nu using the data from simulations with Pr $\geq 1$. As a result, the final expression for the Nusselt number did not contain any Pr dependence, which is applicable only for moderate and large Prandtl numbers. Hence, for small Prandtl numbers, the errors between the predictions and the actual Nu values are very high for the PV model. Thus, as far as the previously developed models are concerned, the RGL model exhibits the best performance.

The MR and NN models developed in this study exhibit very good accuracy in their predictions of Nu and are observed to mostly perform better than the GL and PV models. The predictions of the MR and NN models are even better, albeit marginally, than the RGL model. For small and moderate Pr’s (Pr $\lesssim 1$), both MR and NN models perform better than the RGL model, with the MR model marginally outperforming the NN model. For larger Pr’s, with the exception of Pr $< 2547.9$, the NN model performs better than the RGL model. The performances of the latter two models are comparable in the aforementioned regime. For Pr $= 2547.9$, the ML models, especially the NN model, do not perform as well as they do for the remaining Pr’s; this is possibly because of lack of data points for very large Pr’s.

### C. Significance of the ML models

Now, we will analyze the performances of the MR and NN models developed in the present study. Both the above models exhibit very good accuracy in their predictions of Re and provide significantly better predictions than the GL and the PV models. As exhibited in Table VI, the MR and NN models provide better predictions than even the RGL model for large Prandtl numbers. For small Pr’s, however, the MR model fares marginally worse than the NN and RGL models, with the accuracy of the latter two models being comparable. It must be noted, however, that the MR model performs better than the RGL model if we consider all the Prandtl numbers into account (as shown in Table VI).

### B. Predictions of the Nusselt number

Having analyzed the predictions of Re, we will now examine the performance of the models in predicting Nu. Figures 4(c,d) and Table VII show that the GL model predicts Nu with a good accuracy. The RGL model marginally improves the predictions of Nu for moderate Pr but significantly improves the predictions for extreme Pr. The PV model’s predictions of Nu are close to those by the GL model for moderate and large Pr. However, for small Prandtl numbers, Nu predicted by the PV model deviates significantly from the actual values. This is because Pandey and Verma derived their expression for Nu using the data from simulations with Pr $\geq 1$. As a result, the final expression for the Nusselt number did not contain any Pr dependence, which is applicable only for moderate and large Prandtl numbers. Hence, for small Prandtl numbers, the errors between the predictions and the actual Nu values are very high for the PV model. Thus, as far as the previously developed models are concerned, the RGL model exhibits the best performance.

The MR and NN models developed in this study exhibit very good accuracy in their predictions of Nu and are observed to mostly perform better than the GL and PV models. The predictions of the MR and NN models are even better, albeit marginally, than the RGL model. For small and moderate Pr’s (Pr $\lesssim 1$), both MR and NN models perform better than the RGL model, with the MR model marginally outperforming the NN model. For larger Pr’s, with the exception of Pr $< 2547.9$, the NN model performs better than the RGL model. The performances of the latter two models are comparable in the aforementioned regime. For Pr $= 2547.9$, the ML models, especially the NN model, do not perform as well as they do for the remaining Pr’s; this is possibly because of lack of data points for very large Pr’s.
reached Ra up to $10^{17}$. In addition, Zhu et al.\textsuperscript{68} and Iyer et al.\textsuperscript{69} recently carried out numerical simulations for Ra up to $10^{14}$ and $10^{15}$ respectively. Thus, the machine learning models developed in this study has a good scope for improvement with inclusion of these datasets.

On the other hand, the training dataset for the RGL model is more difficult to construct because it requires the prefactors for the viscous and thermal dissipation rates in the bulk and boundary layers as output parameters. The data on these prefactors is quite limited; in fact, to the best of our knowledge, only Bhattacharya, Verma, and Samtaney\textsuperscript{43} have computed the above quantities using numerical data of RBC. Further, there is no available data on $f_i$’s for extreme Rayleigh numbers ($\gg 10^{10}$). Note that most of the work on such large Rayleigh numbers are experimental because simulations of RBC in that regime require large computational resources. However, computation of prefactors for the dissipation rates requires multipoint measurements. However, due to these constraints of the RGL model, we believe that NN and MR models hold a lot of promise for a robust model for forecasting Re and Nu. We conclude in the next section.

V. SUMMARY AND CONCLUSIONS

In summary, we construct two ML models—multivariate regression (MR) model and neural network (NN) model—to predict Re and Nu for given Ra and Pr. We compared the models’ predictions with existing models. We observe that although all the above models predict Re and Nu with good accuracy, the ML models exhibit the best performance. In general, we observed that Re is more sensitive to modeling parameters compared to Nu; this is because Re depends more critically on Pr compared to Nu. We also discussed that the machine learning models developed in this study can be enhanced further by using data for a wider range of parameters.

The positive results from the machine learning models reinforce the importance of data-driven methods for modelling turbulence, a practice that is gaining popularity\textsuperscript{46–49}. An accurate estimation of the large-scale velocity and heat transport helps scientists and engineers to better understand thermally driven flows encountered in nature and in engineering applications. For example, our results will be useful in understanding atmospheric flows and thus can aid in more accurate climate modelling and weather predictions. On the other hand, our findings will help engineers to better design buildings, heat sinks, solar collectors, and chimneys. Further, since the improvement in the predictions by our models is more pronounced for extreme Prandtl numbers, we believe that our results will be useful in modelling natural flows and designing applications where such extreme Prandtl numbers are involved. Examples of above include the mantle flow (very large Prandtl number) and flows in liquid metal batteries (small Prandtl number). We hope that in future, more machine-learning tools will be constructed for thermal convection that will help model natural and engineering flows.

ACKNOWLEDGEMENTS

We thank J. Schumacher, K. R. Sreenivasan, A. Pandey, M. Sharma, R. Samuel, and S. Alam for useful discussions. The present work was mostly conducted at Indian Institute of Technology Kgpur, India, and was funded by the research grant number SPO/STC/PHY/2018037 from Indian Space Research Organization, India. The simulations of convection were performed on Shaheen II of King Abdullah University of Science and Technology, Saudi Arabia, under the project k1416. Shashwat Bhattacharya is currently funded by a postdoctoral fellowship of Alexander von Humboldt Foundation (Germany).

CONFLICTS OF INTEREST

The authors have no conflicts to disclose.

SUPPLEMENTARY MATERIAL

See supplementary material for the python codes for predicting Re (suppl1.txt) and Nu (suppl2.txt) using neural network, as well as the training dataset (suppl3.csv).

REFERENCES

\textsuperscript{1}S. Chandrasekhar, \textit{Hydrodynamic and Hydromagnetic Stability} (Dover publications, Oxford, 1981).
\textsuperscript{2}G. Ahlers, S. Grossmann, and D. Lohse, “Heat transfer and large scale dynamics in turbulent Rayleigh-Bénard convection,” Rev. Mod. Phys. \textbf{81}, 503–537 (2009).
\textsuperscript{3}F. Chillà and J. Schumacher, “New perspectives in turbulent Rayleigh-Bénard convection,” Eur. Phys. J. E \textbf{35}, 58 (2012).
\textsuperscript{4}E. D. Siggia, “High Rayleigh number convection,” Annu. Rev. Fluid Mech. \textbf{26}, 137–168 (1994).
\textsuperscript{5}K.-Q. Xia, “Current trends and future directions in turbulent thermal convection,” Theor. Appl. Mech. Lett. \textbf{3}, 052001 (2013).
\textsuperscript{6}M. K. Verma, \textit{Physics of Buoyant Flows: From Instabilities to Turbulence} (World Scientific, Singapore, 2018).
\textsuperscript{7}W. V. R. Malkus, “The Heat Transport and Spectrum of Thermal Turbulence,” Proceedings of the Royal Society of London. Series A \textbf{225}, 196–212 (1954).
\textsuperscript{8}B. I. Shraiman and E. D. Siggia, “Heat transport in high-Rayleigh-number convection,” Phys. Rev. A \textbf{42}, 3650–3653 (1990).
\textsuperscript{9}S. Cioni, S. Ciliberto, and J. Sommeria, “Strongly turbulent Rayleigh–Bénard convection in mercury: comparison with results at moderate Prandtl number,” J. Fluid Mech. \textbf{335}, 111–140 (1997).
\textsuperscript{10}J. D. Scheel and J. Schumacher, “Predicting transition ranges to fully turbulent viscous boundary layers in low Prandtl number convection flows,” Phys. Rev. Fluids \textbf{2}, 123501 (2017).
11B. Castaing, G. Gunaratne, Kadannof, L. P., A. Libchaber, and F. Heclot, “Scaling of hard thermal turbulence in Rayleigh-Bénard convection,” J. Fluid Mech. 204, 1–30 (1989).

12X. Chavanne, F. Chilla, B. Castaing, B. Hebral, B. Chabaud, and J. Chaussy, “Observation of the ultimate regime in Rayleigh-Bénard convection,” Phys. Rev. Lett. 79, 3648–3651 (1997).

13S. Horn, O. Shishkina, and C. Wagner, “On non-Oberbeck-Boussinesq effects in three-dimensional Rayleigh-Bénard convection in glycerol,” J. Fluid Mech. 724, 175–202 (2013).

14S. Wagner and O. Shishkina, “Aspect-ratio dependency of Rayleigh-Bénard convection in box-shaped containers,” Phys. Fluids 25, 085110 (2013).

15M. Kaczorowski and K.-Q. Xia, “Turbulent flow in the bulk of Rayleigh-Bénard convection: small-scale properties in a cubic cell,” J. Fluid Mech. 722, 596–617 (2013).

16J. N. Niemela and K. R. Sreenivasan, “Confined turbulent convection,” J. Fluid Mech. 481, 355–384 (2003).

17D. Funfschilling, E. Brown, A. Nikolaenko, and G. Ahlers, “Heat transport in turbulent Rayleigh-Bénard convection in cylindrical samples with aspect ratio one and larger,” J. Fluid Mech. 536, 145–154 (2005).

18R. J. A. M. Stevens, R. Verzicco, and D. Lohse, “Radial boundary layer structure and Nusselt number in Rayleigh-Bénard convection,” J. Fluid Mech. 643, 495–507 (2010).

19D.-L. Dong, B.-F. Wang, Y.-H. Dong, Y.-X. Huang, N. Jiang, Y.-L. Liu, Z.-M. Lu, X. Qiu, Z.-Q. Tang, and Q. Zhou, “Influence of spatial arrangements of roughness elements on turbulent rayleigh-bénard convection,” Phys. Fluids 32, 045114 (2020).

20U. Madanan and R. J. Goldstein, “High-Rayleigh-number thermal convection of compressed gases in inclined rectangular enclosures,” Phys. Fluids 32, 017103 (2020).

21M. Vial and R. H. Hernández, “Feedback control and heat transfer measurements in a Rayleigh-Bénard convection cell,” Phys. Fluids 29, 074103 (2017).

22K.-Q. Xia, S. Lam, and S.-Q. Zhou, “Heat-flux measurement in high-Prandtl-number turbulent Rayleigh-Bénard convection,” Phys. Rev. Lett. 88, 064501 (2002).

23R. Verzicco and R. Camussi, “Prandtl number effects in convecive turbulence,” J. Fluid Mech. 383, 55–73 (1999).

24J. N. Niemela, L. Skrbek, K. R. Sreenivasan, and R. J. Donnelly, “The wind in confined thermal convection,” J. Fluid Mech. 449, 169–178 (2001).

25S. Lam, X.-D. Shang, S.-Q. Zhou, and K.-Q. Xia, “Prandtl number dependence of the viscous boundary layer and the Reynolds numbers in Rayleigh-Bénard convection,” Phys. Rev. E 65, 066306 (2002).

26M. S. Emran and J. Schumacher, “Fine-scale statistics of temperature and its derivatives in convecitive turbulence,” J. Fluid Mech. 611, 13–34 (2008).

27G. Silano, K. R. Sreenivasan, and R. Verzicco, “Numerical simulations of Rayleigh–Bénard convection for Prandtl numbers between $10^{-4}$ and $10^{4}$ and Rayleigh numbers between $10^{5}$ and $10^{9}$,” J. Fluid Mech. 662, 409–446 (2010).

28M. K. Verma, P. K. Mishra, A. Pandey, and S. Paul, “Scalings of field correlations and heat transport in turbulent convection,” Phys. Rev. E 85, 016310 (2012).

29A. Pandey and M. K. Verma, “Scaling of large-scale quantities in Rayleigh-Bénard convection,” Phys. Fluids 28, 095105 (2016).

30A. Pandey, A. Kumar, A. G. Chatterjee, and M. K. Verma, “Dynamics of large-scale quantities in Rayleigh-Bénard convection,” Phys. Rev. E 94, 053106 (2016).

31E. Brown, D. Funfschilling, and G. Ahlers, “Anomalous Reynolds-number scaling in turbulent Rayleigh-Bénard convection,” J. Stat. Mech. Theor. Exp. 2007, P10005 (2007).

32R. H. Kraichnan, “Turbulent thermal convection at arbitrary prandtl number,” Phys. Fluids 5, 1374–1389 (1962).

33D. Lohse and F. Toschi, “Ultimate state of thermal convection,” Phys. Rev. Lett. 90, 034502 (2003).

34S. Pawar and J. H. Arakeri, “Two regimes of flux scaling in axially homogeneous turbulent convection in vertical tube,” Phys. Rev. Fluids 1, 042401(R) (2016).

35E. E. Schmidt, E. Calzavarini, D. Lohse, and F. Toschi, “Axially homogeneous Rayleigh-Bénard convection in a cylindrical cell,” J. Fluid Mech. 691, 52–68 (2012).

36P.-E. Roche, B. Castaing, B. Chabaud, and B. Hebral, “Observation of the 1/2 power law in Rayleigh-Bénard convection,” Phys. Rev. E 63, 045303(R) (2001).

37G. Ahlers, X. He, D. Funfschilling, and E. Bodenschatz, “Heat transport by turbulent Rayleigh-Bénard convection for $Pr \approx 0.8$ and $3 \times 10^{12} \lesssim Ra \lesssim 10^{15}$: aspect ratio $T = 0.50$,” New J. Phys. 12, 103012 (2012).

38X. He, D. Funfschilling, H. Nobach, E. Bodenschatz, and G. Ahlers, “Transition to the Ultimate State of Turbulent Rayleigh-Bénard Convection,” Phys. Rev. Lett. 108, 024502 (2012).

39J. N. Niemela, L. Skrbek, K. R. Sreenivasan, and R. J. Donnelly, “Turbulent convection at very high Rayleigh numbers,” Nature 404, 837–840 (2000).

40P. Urban, P. Hanzelka, T. Krälik, V. Musilová, A. Srnka, and L. Skrbek, “Effect of Boundary Layers Asymmetry on Heat Transfer Efficiency in Turbulent Rayleigh-Bernard Convection at Very High Rayleigh Numbers,” Phys. Rev. Lett. 109, 154301 (2012).

41S. Grossmann and D. Lohse, “Scaling in thermal convection: a unifying theory,” J. Fluid Mech. 407, 27–56 (2000).

42S. Grossmann and D. Lohse, “Thermal convection for large Prandtl numbers,” Phys. Rev. Lett. 86, 3316–3319 (2001).

43S. Bhattacharya, M. K. Verma, and R. Samtaney, “Revisiting Reynolds and Nusselt numbers in turbulent thermal convection,” Phys. Fluids 33, 015113 (2021).

44S. Bhattacharya, A. Pandey, A. Kumar, and M. K. Verma, “Complexity of viscous dissipation in turbulent thermal convection,” Phys. Fluids 30, 031702 (2018).

45S. Bhattacharya, R. Samtaney, and M. K. Verma, “Scaling and spatial intermittency of thermal dissipation in turbulent convection,” Phys. Fluids 31, 075104 (2019).

46E. J. Parish and K. Duraisamy, “A paradigm for data-driven predictive modeling using field inversion and machine learning,” J. Comput. Phys. 305, 758–774 (2016).

47E. Fonda, A. Pandey, J. Schumacher, and K. R. Sreenivasan, “Deep learning in turbulent convection networks,” Proc. Natl. Acad. Sci. U.S.A. 116, 8667–8672 (2019).

48S. Pandey and J. Schumacher, “Reservoir computing model of two-dimensional turbulent convection,” Phys. Rev. Fluids 5, 113506 (2020).

49S. Pandey, J. Schumacher, and K. R. Sreenivasan, “A perspective on machine learning in turbulent flows,” J. of Turbulence 21, 567–584 (2020).

50S. L. Brunton, B. R. Noack, and P. Koumoutsakos, “Machine Learning for Fluid Mechanics,” Annu. Rev. Fluid Mech. 52, 477–508 (2020).

51Goodfellow, Y. Bengio, and A. Courville, Deep Learning (The MIT Press, Cambridge, Massachusetts, 2016).

52T. Hastie, R. Tibshirani, and J. Friedman, The Elements of Statistical Learning (Springer, New York, 2009).

53A. Burkov, The Hundred-Page Machine Learning Book (Andriy Burkov, Québec, 2019).

54R. J. A. M. Stevens, E. P. van der Poel, S. Grossmann, and D. Lohse, “The unifying theory of scaling in thermal convection: the updated prefactors,” J. Fluid Mech. 730, 295–308 (2013).

55M. Lesieur, Turbulence in Fluids (Springer-Verlag, Dordrecht, 2008).

56M. K. Verma, Energy transfers in Fluid Flows: Multiscale and Spectral Perspectives (Cambridge University Press, Cambridge, 2011).

57S. Bhattacharya, S. Sadhukhan, A. Guha, and M. K. Verma, “Similarities between the structure functions of thermal convection and hydrodynamic turbulence,” Phys. Fluids 31, 111507
58. J. D. Scheel, E. Kim, and K. R. White, “Thermal and viscous boundary layers in turbulent Rayleigh–Bénard convection,” J. Fluid Mech. 711, 281–305 (2012).

59. N. Shi, M. S. Emran, and J. Schumacher, “Boundary layer structure in turbulent Rayleigh–Bénard convection,” J. Fluid Mech. 706, 5–33 (2012).

60. R. J. Samuel, S. Bhattacharya, A. Asad, S. Chatterjee, M. K. Verma, R. Samtaney, and S. F. Anwer, “SARAS: A general-purpose PDE solver for fluid dynamics,” J. Open Source Softw. 6, 2095 (2021).

61. M. K. Verma, R. J. Samuel, S. Chatterjee, S. Bhattacharya, and A. Asad, “Challenges in fluid flow simulations using exascale computing,” S.N. Comput. Sci. 1, 178 (2020).

62. S. Bhattacharya, M. K. Verma, and R. Samtaney, “Prandtl number dependence of the small-scale properties in turbulent Rayleigh-Bénard convection,” Phys. Rev. Fluids 6, 063501 (2021).

63. C. F. Jekel and G. Venter, “Pwlf: a python library for fitting 1d continuous piecewise linear functions,” URL: https://github.com/cjekel/piecewise_linear_fit.py (2019).

64. E. Frank, M. Hall, G. Holmes, R. Kirkby, B. Pfahringer, I. H. Witten, and L. Trigg, “Weka-a machine learning workbench for data mining,” in Data mining and knowledge discovery handbook (Springer, 2009) pp. 1269–1277.

65. K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” Neural Netw. 2, 359–366 (1989).

66. A. Gulli and S. Pal, Deep Learning with Keras (Packt Publishing, Birmingham, 2017).

67. C. M. Bishop, Pattern Recognition and Machine Learning (Springer, Singapore, 2006).

68. X. Zhu, V. Mathai, R. J. A. M. Stevens, R. Verzicco, and D. Lohse, “Transition to the Ultimate Regime in Two-Dimensional Rayleigh-Bénard Convection,” Phys. Rev. Lett. 120, 144502 (2018).

69. K. P. Iyer, J. D. Scheel, J. Schumacher, and K. R. Sreenivasan, “Classical 1/3 scaling of convection holds up to Ra = 10^{15},” Proc. Natl. Acad. Sci. U.S.A. 117, 7594–7598 (2020).