

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/340901286>

# A perspective on machine learning in turbulent flows

Article in *Journal of Turbulence* · April 2020

DOI: 10.1080/14685248.2020.1757685

---

CITATIONS  
24

READS  
970

---

3 authors, including:



Sandeep Pandey

25 PUBLICATIONS 243 CITATIONS

[SEE PROFILE](#)



Joerg Schumacher

Technische Universität Ilmenau

225 PUBLICATIONS 3,751 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Liquid Metal Duct Flow under the influence of a Magnetic point Dipole [View project](#)



Fundamental research and modelling of heat and fluid flow of supercritical fluids [View project](#)

## ARTICLE TEMPLATE

### A perspective on machine learning in turbulent flows

Sandeep Pandey<sup>a</sup>, Jörg Schumacher<sup>a,b</sup>, and Katepalli R. Sreenivasan<sup>b,c,d</sup>

<sup>a</sup>Institut für Thermo - und Fluiddynamik, Technische Universität, Ilmenau, Germany

<sup>b</sup>Tandon School of Engineering, New York University, New York, USA

<sup>c</sup>Courant Institute of Mathematical Sciences, New York University, New York, USA

<sup>d</sup>Department of Physics, New York University, New York, USA

#### ARTICLE HISTORY

Compiled April 16, 2020

#### ABSTRACT

The physical complexity and the large number of degrees of freedom that can be resolved today by direct numerical simulations of turbulent flows, and by the most sophisticated experimental techniques, require new strategies to reduce and analyze the data so generated, and to model the turbulent behavior. We discuss a few concrete examples for which the turbulence data have been analyzed by machine learning tools. We also comment on work in neighboring fields of physics, particularly astrophysical (and astronomical) work, where Big Data has been the paradigm for some time. We discuss unsupervised, semi-supervised and supervised machine learning methods to direct numerical simulations data of homogeneous isotropic turbulence, Rayleigh-Bénard convection, and the minimal flow unit of a turbulent channel flow; for the last case, we discuss in some detail the application of echo state networks, this being one implementation of reservoir computing. The paper also provides a brief perspective on machine learning applications more broadly.

#### KEYWORDS

Fully developed turbulence, Machine learning, Data-driven turbulence research

#### 1. Introduction

Machine learning algorithms, in particular those that apply deep (convolutional) neural networks [1–5] have changed our everyday life (e.g., crowd-counting, recommendations in WebPages, Netflix or Amazon). Much of this change has come from the meeting of Artificial Intelligence (AI) and Machine Learning (ML). While ML is sometimes regarded as a subset of AI, there are some differences in usage. AI mimics natural intelligence to solve complex problems and enables decision making; efficiency is not its main driver, and it is an intelligence capability which we want to build into all machines. Machine learning, on the other hand, is about improving and maximizing performance by means of self-learning algorithms. Both of them require large databases from which to learn: the more the high-quality data that becomes available, the better the results — hence the close connection of AI and ML to Big Data. This revolution

---

Email: sandeep.pandey@tu-ilmenau.de

Email: joerg.schumacher@tu-ilmenau.de

Email: katepalli.sreenivasan@nyu.edu

Contact: K. R. Sreenivasan

has been spreading to turbulence research and neighboring fields such as astrophysics, atmospheric physics and climate science. This article is a brief assessment via some examples with which the present authors have been involved. It is not a comprehensive account of possible directions or a full-fledged review of existing work but does attempt a few general remarks.

In turbulence, much of the research work is about detecting patterns and correlations. Our thinking of the physics of turbulence and how we teach it to students is dependent on these patterns and correlations, and their interactions with each other. The patterns may be geometrical (large-scale coherent structures, “worms” in the vorticity field, etc) or statistical (e.g., the non-zero value of the third-order structure function, the so-called 4/5ths law, Lagrangian tetrads, etc). From the 4/5ths law has been created a whole phenomenology of energy cascades and inter-scale interactions; the tetrads tell us something about the how vorticity vector is aligned with the rate of strain. Another example is the pressure/rate-of-strain correlation which contains an important core of ideas about the directional isotropization of turbulent energy.

The role of Big Data and its nexus with AI and ML in turbulence is somewhat different from what is happening in the Big Data world. Much of the latter is about organizing the data, processing it, classifying it, reducing it and marketing it. The focus is usually not on “physical understanding” — which remains to be one of the professed goals in turbulence; scientific insights cannot be obtained from massive amounts of data without digesting them through proper analysis. It is conceivable that we may at some future time be able teach a turbulence course rooted mostly in the data, but it is not clear that data can supplant concepts. In the world of complexity at large, within which turbulence belongs, our contention is that understanding is a premium entity; patterns and correlations help us get there. Modern computing and experimental systems possess the speed, power and flexibility needed to quickly access massive amounts of data, creating such issues as storage, retrieval and post-processing, etc; turbulence research shares these issues with all Big Data problems.

In turbulence research, the primary focus in the last few years has been the exploration of new routes to parametrize unresolved scales in complex flow configurations at high Reynolds numbers, which still remain inaccessible [6–11]. Some examples are stated here for concreteness.

1. As already mentioned, turbulence is replete with statistical laws and correlations (anomalous dissipation, spectral power laws, log-law, etc); most of the ones we know now are obtained by inspiration and belabored data analysis, whose solidification sometimes lingers on for decades. We have created concepts such as energy transfer across scales, triad interactions, cascades, the existence of an inertial range, multifractality, etc., but that phenomenology can be explored more accurately and more readily by ML methods. Those types of analysis can be automated, and the data can be queried more extensively and with less bias. Intense labor can be relegated to machines.

2. New statistical laws will almost certainly be discovered. Some examples of recent vintage are the role of shape fluctuations in describing anomaly for Kraichnan’s model [13]; the area rule for circulation and the role of minimal surfaces [14]; and the connection between ramp-cliff structures and incomplete mixing [15].

3. Turbulence requires closure because of its strong nonlinearity. This means that we are interested in the interrelationship between quantities. In the past, we have modeled them using our preconceived notions (as with pressure/rate-of-strain models) and examined averages, but we should almost certainly look for correlations at a deeper level. Data sizes of real turbulence problems are very large, though typically long time-series can be obtained only in experiments (with their own attendant limitations).

The new quality of ML methods is that one was earlier limited only to few variables while now higher-dimensional information can be extracted. The addition of physical invariance properties of the flow as in Ling et al. [6] leads to more accurate models (also probably to better physics). A comparison conducted by Fang et al. [16] showed that the model of Ling et al. [6] outperforms the classical linear and quadratic eddy viscosity model. Sotgiu et al. [17] demonstrated that feed-forward neural network can be employed in flow with heat transfer where the network is used to find model coefficients in algebraic scalar flux models. Maulik et al.[18] tackled the turbulence modelling hypothesis in an LES framework differently. They devised a data-driven strategy to dynamically switch between structural or functional models, from a priori experience for the closure-modelling.

4. Extreme events and their structure, especially the interrelationship between real space and spectral space, the relation between geometrical shapes and their functions. The important task of discovering the relation between geometry and statistics can be accomplished much better via ML methods.

As the scientific community generates vast amounts of data with great effort and expense, every day by experiment and simulations, as well as observations, in turbulence, space missions, cosmology, particle physics, material science, many-body problems, etc., more and more exciting avenues will be open to us. One such example is the potential to discover habitable exoplanets and exomoons. Machine learning and AI tools allow us to interrogate the data more fully.

Machine learning can be classified into three big categories — unsupervised, semi-supervised and supervised machine learning [5,12]. Unsupervised machine learning extracts features in (high-dimensional) data sets without pre-labeled training data. These techniques are established already and known for decades. They comprise, for example, clustering such as the  $k$ -means or spectral clustering and dimensionality-reduction techniques, such as the well-known Proper Orthogonal Decomposition (POD) [19] or Dynamic Mode Decomposition (DMD) [20]. The new element is the enormous amount of data that can be handled now. In contrast, supervised machine learning requires labeled data for the training of the ML algorithm. Needless to say, semi-supervised learning combines aspects of supervised and unsupervised learning methods.

In the following, we will discuss some applications to turbulence which involve example cases from these three ML methods. We start with unsupervised and semi-supervised learning in the next section. Section 3 will be concerned with supervised learning cases — the focus of this work. The first example stands for a sophisticated pattern analysis in an extended turbulent convection flow, and the second example is a recurrent neural network that provides an equation-free model for the prediction of the dynamical behavior of the flow. Furthermore, we shed some light on difficulties in using standard ML methods for the detection of extreme events in turbulent flows. Our examples include examples of turbulent convection problems. Section 4 builds a bridge to ML in astrophysics and astronomy that have been facing the analysis of big data for many years. The work concludes with a brief summary and outlook.

## 2. Unsupervised and semi-supervised learning

Unsupervised algorithms are used to perform cluster analysis, dimensionality reduction, visualization, and so forth, as discussed above. Applications of semi-supervised ML to three-dimensional turbulence comprise the use of Generative Adversarial Networks (GANs) [5] to learn statistical properties of turbulent flows. GANs combine two

competing neural networks. The first one, termed (convolutional) generator network, produces synthetic turbulence samples which are compared by the (convolutional) discriminator network, the second network, with the training data coming from DNS (or experiment, in principle). This iterative interaction between both networks proceeds as long as it takes for the discriminator to become insensitive to the differences between the synthetic data and the real data. We highlight applications for each case here.

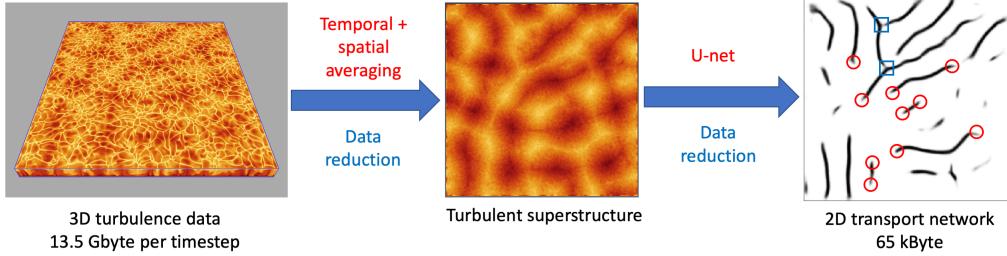
*Lagrangian trajectories in turbulent thermal convection:* In Schneide et al. [21], unsupervised machine learning techniques in the form of spectral clustering were applied to three-dimensional turbulent thermal convection in a large-aspect-ratio domain to analyze the time evolution of an ensemble of Lagrangian particles. To this end, the individual trajectories were composed into a network and their connection to the large-scale organization of the convection flow at hand — termed turbulent superstructures of convection [22] — was studied. The set of individually advected Lagrangian particles at an instant of time was considered as an undirected graph. The vertices of the graph were connected by edges. The dynamical histories of the Lagrangian particle ensemble up to a certain time were then encoded into the weights which are assigned to each edge. These weights were calculated as the inverse of a time-averaged distance of mutual tracer trajectories and set to zero if this distance exceeded a threshold [23]. The latter is the graph sparsification step. By solving a balanced cut problem via an equivalent generalized eigenvalue problem of the Laplacian matrix of the graph [24], the network was decomposed into  $k$  subgraphs or clusters. The clusters of the graph so obtained were then related to large-scale patterns of convective flow obtained in the Eulerian frame of reference.

*Standard turbulent statistics in homogeneous isotropic turbulence:* Convolutional GANs have been applied recently to homogeneous isotropic box turbulence by King et al. [27]. The authors fed into their scheme two-dimensional slices of three-dimensional data from the Johns Hopkins turbulence data base [28] and achieved reasonable agreement of the output in terms of energy spectra, probability density functions of velocity gradient tensor components, and the anomalous scaling exponents of velocity increment moments in the inertial range. Recall that this is a static method and ignores all dynamical evolution of the turbulent flow. Improvement can be made (as was discussed in [27,29]) by utilizing supervised ML algorithms that incorporate dynamical information; this is described in the next section.

In summary, unsupervised and semi-supervised learning algorithms can be used to extract new knowledge from existing datasets and facilitate new discoveries. It is clear that we have merely scratched their potential at this stage.

### 3. Supervised learning

Supervised ML is used to uncover relationships between a set of measurements and the target variables. It can describe very complex nonlinear relations between measurements and the target variables and is superior to traditional algorithms based on fitting of predefined models. Supervised ML makes use of the fact that it is often easier to train a system, such as a deep convolutional neural or an echo state network in the present case, with a number of labeled examples of an intended input-output behavior, than to develop a specific computer program to provide the correct answer for all possibilities. In the following section, we provide three examples of supervised ML algorithms for different tasks.



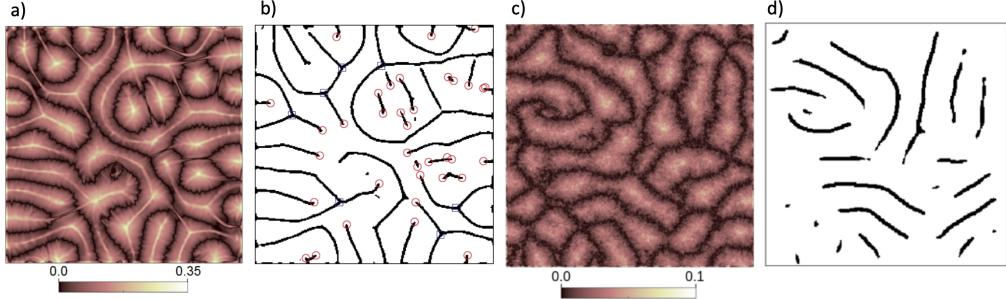
**Figure 1.** The workflow scheme illustrates data reduction for a three-dimensional convection flow snapshots to the planar network. Data compression is by more than a factor of 100,000 for the largest Rayleigh number considered here, from a sequence of fully resolved three-dimensional snapshots of the temperature field to a ridge pattern that stands for the regions with the on average largest convective heat flux across the layer. In the right panel, defect points in the network are indicated: open circles for wedge points which are the end points of the ridges, and open square for defect points marking the merger of two ridges.

### 3.1. Pattern analysis in turbulent Rayleigh-Bénard convection flow

In Fonda *et al.* [30] we presented an application of a deep convolutional neural network (DCNN) to extract large-scale patterns in a turbulent and horizontally extended Rayleigh-Bénard convection (RBC) system. Specifically, our goal was to quantify the importance of turbulent superstructures in the heat transport. Turbulent superstructure describe a large-scale organization of convection into patterns with a horizontal extension larger than the layer height. They evolve only gradually with time. It has been thought that they form the backbone for the turbulent heat transport [22]. These physical insights into the turbulent transport mechanisms in the convection layer were used to pre-process the data that enter the neural network. For example, it is well-known that local maxima of convective heat flux coincide with local temperature minima and maxima — a property that remains robust across the whole height.

We analyzed data records obtained in a Cartesian domain of aspect ratio  $25H : 25H : H$ , where  $H$  is the convection layer height. The Prandtl number of the fluid was  $\text{Pr} = 7$  and the Rayleigh numbers were  $\text{Ra} = 10^5, 10^6$  and  $10^7$ . We trained a DCNN to reduce the three-dimensional temperature field to a temporal planar network in the mid-plane of the convection layer, as illustrated in figure 1. The resulting planar network is used to quantify the turbulent heat transport carried by the superstructure. In ref. [30] we also showed that the network performs better as a standard edge detection method. Defect points (indicated in the figure) disappear and form new connections with time; they are the “hot spots”, or the points of locally enhanced heat flux. It was shown that the fraction of heat carried by the superstructure decreases as the Rayleigh number increases; and that an increasing amount of heat is carried by the small-scale background turbulence rather than by the coherent large-scale superstructures of convection. Since the area fraction of the ridge network also decreases with increasing Rayleigh number, the network as a contributor to the turbulent heat transfer remains intact if normalized by the area it occupies. Figure 2 demonstrates the extraction of ridge patterns by the deep neural network (see panels b,d), even though the data were quite noisy for the highest Rayleigh number (see panels c,d).

The DCNN applied here is the U-net [31] that combines a contraction path of a standard convolutional neural network [5] with a subsequent expansion path of concatenations and up-convolutions that finally create a detailed segmentation map. This specific architecture is essential for the present application of a large-aspect-ratio RBC because it requires small sets of manually annotated data for training purposes. The



**Figure 2.** Symmetrized temperature  $\Theta(x, y)$  as input (a,c) and ridge network as output (b,d) of the U-net, for Rayleigh numbers  $\text{Ra} = 10^5$  in (a,b) and  $10^7$  in (c,d). All data are for a Prandtl number  $\text{Pr} = 7$ . For the lowest Rayleigh number, we highlighted the defect points; open circles show wedge points at the end of a ridge and squares show trisector points. Input data to the U-net is the symmetrized temperature field in the midplane, given by  $\Theta(x, y) = |T(x, y, z = 1/2) - 1/2|$ , as shown in panels (a) and (c).

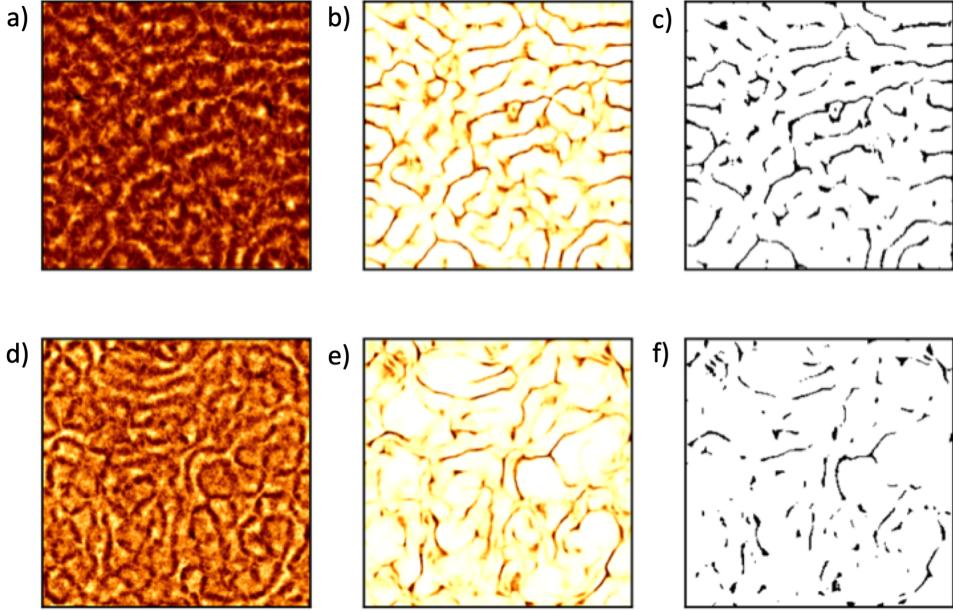
slow evolution of the superstructures in the large-aspect-ratio domain would otherwise require extremely long simulations of the full RBC flow, over a few tens of thousands of convective time units, to obtain an appropriate amount of independent training data, as discussed by Pandey et al. [22].

Figure 3 demonstrates the ability of the DCNN to extract ridge patterns at different Prandtl numbers. We display cases at  $\text{Ra} = 10^5$  and  $\text{Pr} = 0.7$  (top row) and  $0.021$  (bottom row). This generalization can be obtained when additional augmentation methods of the training data are used, such as the addition of noise. We have also combined data from different simulations in the training [32]. It can be seen in panels (c) and (f) of the figure that the ridge patterns become more fragmented and reveal many details, particularly for the lower Prandtl number data in the bottom row. Although surprisingly robust, our data-reduction method approaching a simple dynamic network will reach its limits, especially for convection data which are characterized by strong fluid inertia.

A possible continuation of this analysis is to build a network-based reduced dynamical model that provides time series for the heat flux and its fluctuations across an extended convection layer.

### 3.2. Statistical prediction in simple turbulent shear flow

To examine the potential of neural networks for the prediction of turbulent dynamics of shear flows without a detailed use of the underlying dynamical equations, we have extended recent investigations of Srinivasan et al. [33] in a simple shear flow consisting of a 9-mode Galerkin model of Moehlis et al. [34]. The velocity field is given by  $\mathbf{u}(\mathbf{x}, t) = \sum_{k=1}^9 a_k(t) \mathbf{u}_k(\mathbf{x})$ . The modes  $\mathbf{u}_k(\mathbf{x})$  prescribe the spatial dependence of the important building blocks of the near-wall flow and the 9 real expansion coefficients  $a_k(t)$  govern their dynamics. The model, which is also known as the minimal flow unit of a turbulent shear flow, stands for a cycle of coherent structure formation as supposed to be present in every shear flow: the non-normal amplification and transient growth of perturbations in the mean profile in the form of streamwise vortices, the formation of streamwise streaks and their instability and breakdown, and the reformation of streamwise vortices by nonlinear interactions [35,36]. With these 9 modes, instantaneous velocity fields can be obtained by Galerkin projection in a domain  $L_x : L_y : L_z = 4\pi H : 2H : 2\pi H$  with periodic boundary conditions in the



**Figure 3.** Ridge extraction for data at different Prandtl numbers. (a)  $\text{Pr} = 0.7$  and (d)  $\text{Pr} = 0.021$ . Panels (b) and (e) show the direct U-net output, and panels (c) and (f) the conversion into binary files. Both data records were obtained for  $\text{Ra} = 10^5$  and aspect ratio of 25:25:1.

streamwise and spanwise directions,  $x$  and  $z$ , respectively (see also figure 4). Free-slip boundary conditions are applied in the wall-normal direction,  $y$ .

Rather than integrating the nonlinear Galerkin model so obtained, written for a given Reynolds number  $\text{Re}$  as

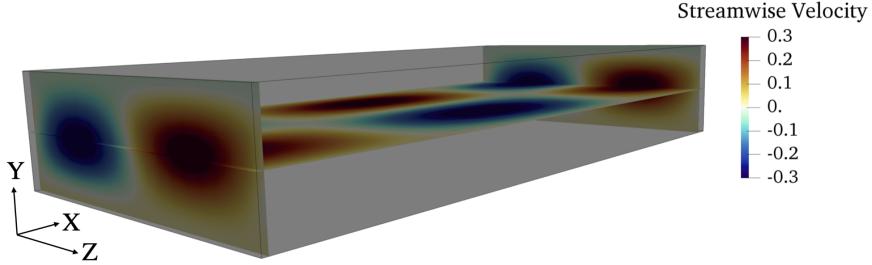
$$\frac{da_k}{dt} = F_k(a_m, t; \text{Re}) \quad \text{with} \quad \text{Re} = \frac{U_0 d}{2\nu} \quad \text{and} \quad k, m = 1, \dots, 9, \quad (1)$$

the objective is now to substitute (1) by a neural network such that one can bypass the solution of the Galerkin model. For this particular study, 100 time series with 4000 time steps in each series result in total of 400,000 unique data points that were used to train three different neural networks. We tested subsequently the networks so trained on a completely new and unseen time-series to assess the generality and robustness of the models. In addition to a multilayer perceptron (MLP) — a standard DNN — and a long short-term memory (LSTM) network [37] — a recurrent neural network (RNN) — both of which have been examined already by Srinivasan et al. [33], we focused on an implementation of reservoir computing in the form of an echo state network (ESN) [38–41].

Similar to the other RNNs, the ESN has an input and an output layer, in our case the 9-dimensional state vectors  $\mathbf{a}_{\text{in}}(t)$  and  $\mathbf{a}_{\text{out}}(t)$ , respectively. Instead of several hidden layers, the input layer is connected to a state vector in the reservoir by  $\mathbf{r}(t) = \mathbf{W}_{\text{in}} \mathbf{a}_{\text{in}}(t)$  with the randomly initialized  $N \times 9$  weight matrix  $\mathbf{W}_{\text{in}}$  and  $N \gg 9$ . A typical reservoir dynamics in the training phase is given by

$$\mathbf{r}(t + \Delta t) = (1 - \alpha)\mathbf{r}(t) + \alpha \tanh(\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\mathbf{a}_{\text{in}}(t) + \xi\mathbf{1}). \quad (2)$$

Here,  $\mathbf{A}$  is the reservoir — a randomly initialized  $N \times N$  sparse matrix,  $\alpha$  the leakage



**Figure 4.** Snapshot with iso-contours of the streamwise velocity fluctuations highlighting the low- and high-speed streaks, which are the characteristic coherent structures in this flow. The flow Reynolds number is  $\text{Re} = U_0 d / (2\nu) = 400$ , where  $d = 2H$  is the distance between the walls and  $U_0$  is the amplitude of the sinusoidal laminar flow.

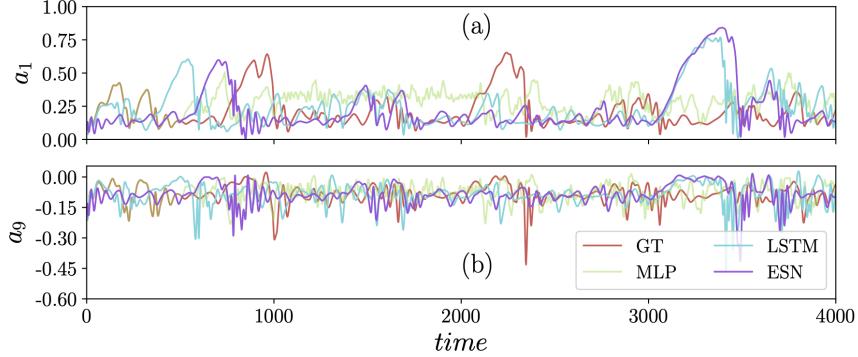
rate, and  $\xi\mathbf{1}$  a bias vector containing a chosen constant  $\xi$  in each component. While matrices  $\mathbf{W}_{\text{in}}$  and  $\mathbf{A}$  remain fixed in the training procedure, the output  $9 \times N$  weight matrix, which is defined via  $\mathbf{a}_{\text{out}}(t) = \mathbf{W}_{\text{out}}\mathbf{r}(t) + \mathbf{c}$  with bias  $\mathbf{c}$ , is updated by a standard optimization procedure leading eventually to  $\mathbf{W}_{\text{out}}^*$  and  $\mathbf{c}^*$ . In addition to the training without expensive back-propagation in a multi-layered convolutional network, reservoir computing methods thus provide a high-dimensional dynamical system that is driven close to instability with a spectral radius  $\rho(\mathbf{A}) \leq 1$ . Once the training at the output layer is completed, the dynamics is given by

$$\mathbf{r}(t + \Delta t) = (1 - \alpha)\mathbf{r}(t) + \alpha \tanh [\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{\text{in}}(\mathbf{W}_{\text{out}}^*\mathbf{r}(t) + \mathbf{c}^*) + \xi\mathbf{1}] . \quad (3)$$

Table 1 shows the ESN parameters used for our subsequent analysis, compared to the other two configurations.

All three neural networks (MLP, LSTM and ESN) are trained locally with 100 time series, each of them containing 4000 time steps (overall 400,000 data points for each mode). Depending on the specific network, the input dimension can differ. Table 2 summarizes the optimized models in terms of training and validation loss in the training phase. Here, the term loss refers to the mean-square error between the actual and predicted values. All three models achieve an acceptable level of training and validation loss, which suggests that the models are well-trained and do not suffer from overfitting. Computationally, MLP is the most expensive method for this problem due to large input dimension combined with large number of hidden layers. LSTM achieves the lowest loss for both training and test datasets. Also, the training time is less compared to MLP. However, ESN outperforms both MLP and LSTM in terms of the training time. The training of the ESN takes a fraction of a second on normal computer while LSTM training needs a couple of minutes with our current training size. We recall that the present problem is small with respect to the training data size compared to a typical turbulence problem. We therefore expect that the ESN will perform better for a system with much larger number of degrees of freedom than the 9 chosen for the present case.

After the training procedure, we performed predictions by utilizing a new time series that is not used during the training phase. These new series (ground truth) then fed to the network to start the prediction from trained models. For instance, we need at least 500 time steps data in each series to start a prediction in an MLP. This number is smaller in the LSTM case with only 10 time steps due to its inherent internal memory and feedback. In contrast, ESN needs only  $a_k(0)$  to start its prediction. A comparison



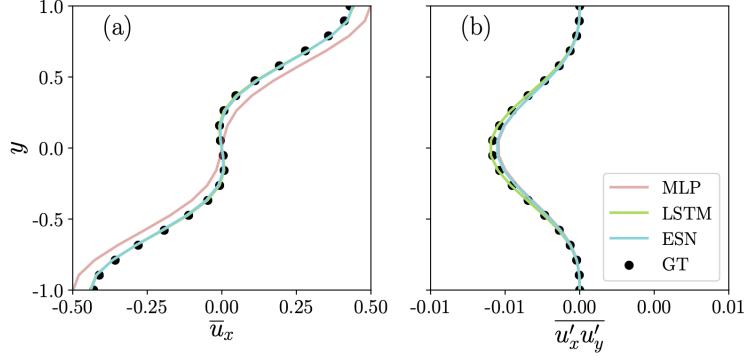
**Figure 5.** Comparison of predicted time series for the expansion coefficients obtained from the three neural networks, MLP, LSTM and ESN, with ground truth (GT). (a)  $a_1(t)$  versus time  $t$  (b)  $a_9$  versus  $t$ . The model was trained with 400,000 data points and results are from a blind test for one of the time series with 4000 steps.

of the predictions of  $a_1$  and  $a_9$  is shown in figure 5. A visual inspection suggests that trained models are not able to recreate the exact trend of modes, and the mean-square error is higher than the training loss. However, the overall result is very close to the actual data, i.e., all three networks give a nonlinear dynamics that is close to the ground truth. The reason behind the remaining disparity is the highly chaotic nature of the model, which implies a sensitivity with respect to small round-off errors. Since our aim is to recreate the statistical properties of this dynamical system, we compare the mean and fluctuation profile of velocity components to quantify the error. Therefore, after the prediction, the velocity fields were reconstructed by means of the Galerkin model. We recreated 400,000 data points for the velocity vector in the physical domain of  $20 \times 20 \times 20$ . This domain and the field are further used for obtaining the mean velocity and Reynolds shear stress. The normalized average relative error in the ground truth (GT) and the predicted value (PR) is calculated following [33]. For example, for the mean streamwise velocity profile, this error is given by

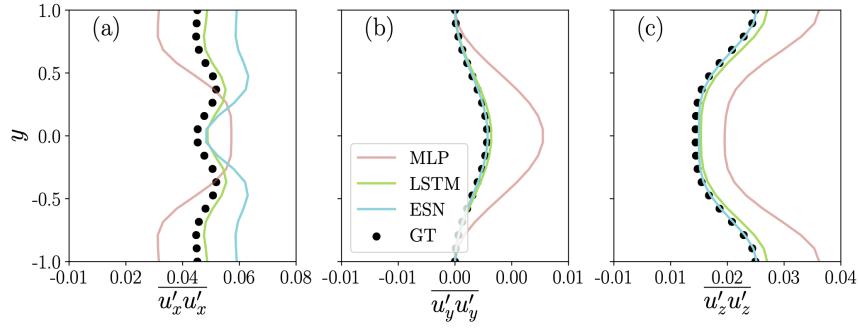
$$E[\bar{u}_x(y)] = \frac{1}{2 \max_{y \in [-1,1]}(|\bar{u}_x^{\text{GT}}(y)|)} \int_{-1}^1 |\bar{u}_x^{\text{GT}}(y) - \bar{u}_x^{\text{PR}}(y)| dy. \quad (4)$$

Figure 6 illustrates the mean streamwise velocity profile and Reynolds shear stress profile in the wall-normal direction along with the profile for streamwise and wall-normal fluctuations. One can clearly notice an excellent prediction capability of LSTM and ESN on the blind data set in the first and second order statistics. However, none of the three models accurately predicts the streamwise fluctuation while the agreement for wall-normal and spanwise fluctuation is excellent, as depicted in figure 7. The error, quantified by eq. (4), is shown in Table 2. LSTM achieves the lowest error in the prediction of mean velocity and Reynolds shear stress followed by ESN. The reduced training time, the dynamical-systems nature of the reservoir, and the higher accuracy and the robust framework favor however to our view ESN among the three methods. We believe that the error can be further reduced, e.g. with a deep ESN network.

The dynamics of the reservoir of the ESN is determined by several parameters, the spectral radius ( $\rho$ ), the leakage rate ( $\alpha$ ), number of reservoir nodes ( $N$ ) and density of active nodes in the reservoir ( $D$ ). Therefore, we also made an attempt to unveil

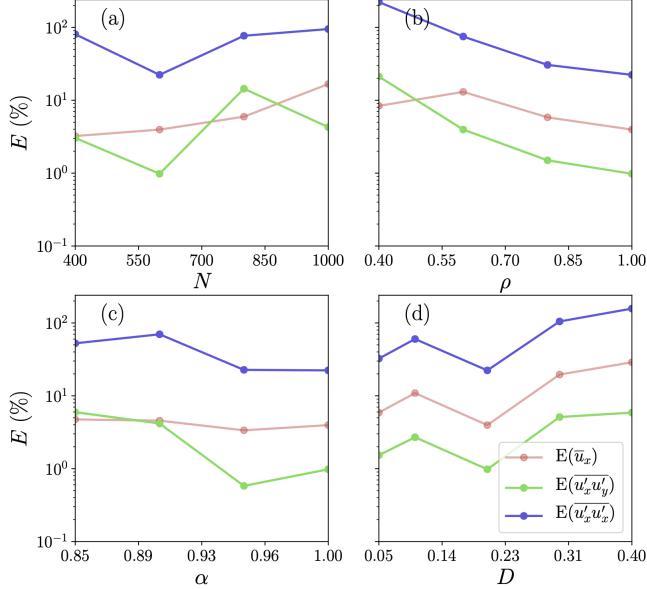


**Figure 6.** Comparison of predicted values from MLP, LSTM and ESN with the ground truth (GT) for (a) mean streamwise velocity component profile and (b) Reynolds shear stress profile. Data are obtained from an average in streamwise and spanwise directions in combination with time.



**Figure 7.** Comparison of predicted values from MLP, LSTM and ESN with the ground truth (GT) for (a) streamwise velocity fluctuations, (b) wall-normal velocity fluctuations, and (c) spanwise velocity fluctuations. Data are obtained from an average in streamwise and spanwise directions in combination with time.

their effect on this particular problem. Figure 8 shows the variation of the normalized relative error as a function of the reservoir parameters. An increase in the number of reservoir nodes or the density of active reservoir nodes may bring down the training loss, but the normalized relative error can increase after a minimum, indicating the possibility of overfitting (see figure 8a and 8d). Similarly, very low leakage rates or a high spectral radius can lead to overfitting. In this work, the dynamical reservoir was initialized with an internal weight matrix randomly, and stayed untrained. Different initialization strategies can be employed such as Sparse and Orthogonal Reservoir Matrices (SORM), CyclicSORM, Ring of Neurons and Chain of Neurons [43] to improve the dynamics of reservoir. To summarize, we have demonstrated for this specific example that the reservoir computing approach, implemented here as an echo state network, is capable of predicting the low-order statistics in a simple turbulent flow without expensive training. It is thought that this advantage of the ESN becomes even more pronounced for examples that obey significantly more degrees of freedom. Detailed studies of more complex turbulent flows are planned for the future and will be reported elsewhere.



**Figure 8.** Performance of the echo state network while varying (a) the number of reservoir nodes  $N$ , (b) the spectral radius  $\rho$ , (c) the leakage rate  $\alpha$ , and (d) the reservoir density  $D$ . The results are from the test dataset and quantified in terms of normalized relative errors of the streamwise velocity ( $E(\bar{u}_x)$ ), the Reynolds shear stress ( $E(u'_x u'_y)$ ), and the mean-square streamwise velocity fluctuation ( $E(u'_x u'_x)$ ). During these parametric tests, only one parameter is varied while the others are held fixed with default values of  $N = 600$ ,  $\rho = 1$ ,  $\alpha = 1$ , and  $D = 0.20$ .

### 3.3. Extreme events in homogeneous isotropic box turbulence

Extreme events in nature such as tornadoes, large floods and strong earthquakes are rare but have disproportionate consequences. The predictability of these events is very limited at present. If we translate the question to turbulence, we can ask (see ref. [25]) if there are precursor patterns for extreme events (e.g., in the turbulent energy dissipation). As discussed already, convolutional neural networks (CNN) are very good at recognizing patterns in images, but are they also good in predicting extreme events in turbulence? In particular, is the predictability of these events, limited at present because they occur on small spatial support and are highly intermittent, any better using ML methods than by conventional methods, as described by Donzis & Sreenivasan [25]? Fonda et al. [26] used a DNS database of homogeneous and isotropic turbulence in a box with periodic boundary conditions in all three space directions with Taylor microscale Reynolds numbers spanning a range from  $140 \leq R_\lambda \leq 1300$ , computed with different numerical grid resolutions varying from  $256^3$  to  $8192^3$  points, and started with an aggressive data augmentation to virtually increase the number of these rare events by two orders of magnitude to train a CNN, with the purpose of predicting their occurrence in an independent data set. The predictability of these events improves only marginally with the use of ML methods. It might be the case that ML algorithms are not too suited for detecting and classifying extreme and rare events unless some kind of self-similarity is exploited. At present, such laws of self-similarity are not understood well enough; neither do we know how to implement such laws, even if known, in effective ML methods.

#### 4. Machine learning in astronomy and astrophysical turbulence

In turbulence research, each group traditionally generated its own data and analyzed them with its own resources. Data were cheap. The tendency in turbulence as a field, especially in recent years, has been to push the parameter range constantly, in both simulations and experiment, and some of them are one of a kind. The situation is beginning to resemble that in astrophysics and astronomy. There is an enormous amount to be learned by the experience of those communities. Our goal in this short section is to highlight this point, which we believe is forward-looking. Although the types of concerns in these fields, and in neighboring areas such as cosmology, many-body physics and particle physics are somewhat different from those in turbulence (see, e.g., Carelo et al. [44]), the fact that frontier turbulence research is being driven by a few large data sets, much as in astronomy and astrophysics, is a good reason to describe a few examples where supervised and unsupervised learning has been used effectively in these fields. The examples chosen are closer to the authors' interests (dynamical phenomena), and are by no means extensive.

Astronomy and astrophysics are experiencing major growths in both the size and sophistication of data because of new telescopes and satellites. Indeed, astronomy has become a paradigm for Big Data science because of continuing developments of ground and space based observatories, including Sloan Digital Sky Survey (SDS), Large Synoptic Survey Telescope (LSST), the Square Kilometer Array (SKA), Solar Dynamics Observatory (SDO), Laser Interferometer Gravitational-wave Observatory (LIGO) and its European counterpart, Galactic Surveyor (GAIA), Chandra X-ray Observatory, Kepler, Solar Orbiter, Plato, etc. This change naturally calls for new automatic data-mining tools not only to extract features faster and better, but also to acquire new information and insight. ML techniques result in huge data compression and accomplish the assigned tasks with great rapidity and efficiency. ML has been used extensively in astronomy and asteroseismology for automatically analyzing large data sets accurately, e.g. by multilayered neural networks and convolutional random forest regressors. In effect, ML is used in these areas of research as a denoising algorithm that reconstructs representative underlying spectra and subsequently classifies them in some convenient way, more easily than manual methods, so that further physical characteristics to be extracted.

Space observatories like Kepler, stellar and planetary systems beyond our own are now being characterized en masse for the first time. These characterizations are pivotal for endeavors such as searching for Earth-like planets and solar twins, understanding the mechanisms that govern stellar evolution, and tracing the dynamics of our Galaxy. The volume of data that is becoming available, however, brings with it the need to process this information accurately and rapidly. While existing methods can constrain fundamental stellar parameters such as ages, masses, and radii from these observations, they require substantial computational effort to do so. Machine learning methods can rapidly estimate fundamental parameters of main-sequence solar-like stars from classical and asteroseismic observations.

- The magnetic field of the Sun is very important for determining the space weather with possible consequences to the Earth. One can map the magnetic field of the Sun on only one side (frontside), but a proper job of assessing the space weather requires the magnetic field on the farside as well. Kim et al. [45] construct the farside field by using ML techniques that learn from part of the data on the frontside field, and whose accuracy is tested by correlating with the remainder

of the data. Taking the whole data together, the authors successfully monitor the temporal evolution of magnetic fields of the Sun.

- Crater counting on the Earth’s moon (and other astronomical bodies in the solar system) is of interest for constructing the dynamical history of the solar system. The counting has traditionally been done by visual inspection of images, thus limiting the scope, efficiency, and accuracy of results. In Silburt et al. (2018) [46], the authors demonstrate the use of convolutional neural networks to determine the positions and sizes of craters from the lunar digital elevation maps; this process almost doubled the total number of crater detections.
- The Kepler mission alone has photographed about 15,000 stars every 30 minutes for four or so years. Significant work has been carried out on detecting the planets of these stars (the exoplanets) by monitoring large number of stars. Manual interpretation of potential exoplanet candidates is labor intensive and subject to human error, the results of which are difficult to quantify. Deep learning has been used for detecting exoplanets (about 4000 such planets have been discovered at this time, many of them compatible with habitability of life as we know it, although no life has been discovered so far). One such study using ML for imaging exo-planets is reported by [47], another by Pearson et al. [48]. It is estimated that there are some 200 billion stars in Milky Way alone (and some  $10^{23}$  in the Universe), so the importance of the machine learning tools is obvious for the future. It is also of interest to note that Alshehhi et al. [49] have explored the use of convolutional neural networks to detect exomoons (moons around exoplanets).
- Helioseismology techniques which monitor the acoustic field on the Sun’s surface can be used by inverse methods to detect the motion of the interior solar plasma. One of the inverse techniques is the so-called ring-diagram analysis. Alshehhi et al. [50] have utilized neural networks as a supervised learning method for predicting surface flows, especially in the context of recently discovered Rossby waves (see Löptien et al. [51]).
- Verma et al. [52] have demonstrated that artificial neural networks are successful in determining the evolutionary parameters of stars (the mass, initial helium abundance, initial metallicity, mixing length — assumed to be constant over time — and the age to which the star must be evolved). They compare the observations for a few stars with results from other techniques, and declare satisfactory agreement, except that ML methods are computationally cheaper.

A review of some specific applications in astronomy and related areas can be found in Baron [53].

## 5. Conclusions and outlook

The range of machine learning methods for turbulent flows is broad, from unsupervised clustering algorithms via semi-supervised methods to supervised learning. Deep convolutional neural or recurrent neural networks can predict low-order statistics in simple turbulent flows without the knowledge of the underlying mathematical equations of fluid motion, as illustrated by a few examples. The algorithms suggest efficient strategies to reduce the three-dimensional data records as in cases of the spectral clustering of Lagrangian trajectories or the reduction of the three-dimensional turbulent superstructures to a planar transport network with slow dynamics.

Recurrent networks, specifically echo state networks, are found to be a promising

tool to model dynamical processes in flows without the knowledge of the underlying Navier-Stokes or Boussinesq equations and with smaller training effort than other recurrent networks such as LSTM. Echo state networks (also known as reservoir computing models) can thus serve as simple mesoscale models which describe and predict the sub-grid scale dynamics in larger-scale or global convection models of the atmosphere or inside stars. These networks form an interesting application for a further reason. It is known that the mathematical foundations of DNNs are still poorly developed, although the practical progress made is impressive [54]. Echo state networks are dynamical systems and open door to mathematical analysis and deeper understanding. A similar aspect was recently discussed by E [55]. For the simple shear flow, the ESN demonstrated a tremendous potential in terms of reduced training time with high accuracy. The performance can be improved further by the application of deep ESN. This route and the transformation of these concepts to more complex flows with a larger number of degrees of freedom are currently under investigation and will be reported elsewhere.

Machine learning analysis in turbulence can benefit significantly from applications of AI in neighboring research fields such as astronomy and astrophysics where the stream of new research data coming from satellites, telescopes, and big sky surveys is significantly bigger than in fluid mechanics currently.

## Acknowledgements

SP is supported by the Deutsche Forschungsgemeinschaft with grant SCHU 1410/30-1. We thank Enrico Fonda for his initial work on predicting extreme events, whose outcome has been reported here, Ambrish Pandey and Chris Hanson for several discussions, and Robert Kräuter for his help with the analysis of the performance of the U-net for different Prandtl numbers. The research was also supported by supercomputing resources which were provided by the project grant HIL12 of the John von Neumann Institute for Computing (NIC) at the Jülich Supercomputing Centre and by the Large Scale Project pr62se of the Gauss Centre for Supercomputing (GCS) at the Leibniz Rechenzentrum Garching.

## References

- [1] Hinton G, Deng L, Yu D, Dahl GE, et al. Deep neural networks for acoustic modeling of speech recognition. *IEEE Signal Processing Mag.* 2012; 29(11):82–97.
- [2] Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science* 2015; 349:255–260.
- [3] LeCun Y, Bengio Y, Hinton G. Deep Learning. *Nature* 2015; 521:436–444.
- [4] Schmidhuber J. Deep learning in neural networks: An overview. *Neural Networks* 2015; 61:85–117.
- [5] Goodfellow I, Bengio Y, Courville A. Deep Learning. The MIT Press, Cambridge (MA), 2017.
- [6] Ling J, Kurzawski A, Templeton J. Reynolds averaged turbulence modeling using deep neural networks with embedded invariance. *J. Fluid Mech.* 2016; 807:155–166.
- [7] Kutz NJ. Deep learning in fluid dynamics. *J. Fluid Mech.* 2017; 814:1–4.
- [8] 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; 2:034603.

- [9] Wu J-L, Xiao H, Paterson E. Physics-informed machine learning approach for augmenting turbulence models: A comprehensive framework. *Phys. Rev. Fluids* 2018; 3:074602.
- [10] Duraisamy K, Iaccarino G, Xiao H. Turbulence modeling in the age of data. *Annu. Rev. Fluid Mech.* 2019; 51:357–377.
- [11] Brenner MP, Eldredge JD, Freund JB. Perspective on machine learning for advancing fluid mechanics. *Phys. Rev. Fluids* 2019; 4:100501.
- [12] Brunton SL, Noack BR, Koumoutsakos P. Machine Learning for Fluid Mechanics. *Annu. Rev. Fluid Mech.* 2020; 52:477–508.
- [13] Falkovich G, Sreenivasan KR. Lessons from hydrodynamic turbulence. *Phys. Today* 2006; 59:43–49.
- [14] Iyer KP, Sreenivasan KR, Yeung PK. Circulation in High Reynolds Number Isotropic Turbulence is a Bifractal. *Phys. Rev. X* 2019; 9:041006 (13 pages).
- [15] Iyer KP, Schumacher J, Yeung PK, Sreenivasan KR. Fractal iso-level sets in high-Reynolds-number scalar turbulence. *Phys. Rev. Fluids* 2019; submitted.
- [16] Fang R, Sondak D, Protopapas P, Succi S. Neural network models for the anisotropic Reynolds stress tensor in turbulent channel flow. *J. Turbul.* 2019; 1–19.
- [17] Sotgiu C, Weigand B, Semmler K. A turbulent heat flux prediction framework based on tensor representation theory and machine learning. *Int. Commun. Heat Mass.* 2018; 95:74–79.
- [18] Maulik R, San O, Jacob JD, Crick C. Sub-grid scale model classification and blending through deep learning. *J. Fluid Mech.* 2019; 870:784–812.
- [19] Berkooz G, Holmes P, Lumley JL. The proper orthogonal decomposition in the analysis of turbulent flows. *Annu. Rev. Fluid Mech.* 1993; 25:539–575.
- [20] Williams MO, Kevrekidis IO, Rowley CW. A data-driven approximation of the Koopman operator: Extending Dynamic Mode Decomposition. *J. Nonlinear Sci.* 2015; 25:1307–1346.
- [21] Schneide C, Pandey A, Padberg-Gehle K, Schumacher J. Probing turbulent superstructures in Rayleigh-Bénard convection by Lagrangian trajectory clusters. *Phys. Rev. Fluids* 2018; 3:113501 (15 pages).
- [22] Pandey A, Scheel JD, Schumacher J. Turbulent superstructures in Rayleigh-Bénard convection. *Nat. Commun.* 2018; 9:2118 (11 pages).
- [23] Hadjighasem A, Karrasch D, Teramoto H, Haller G. Spectral-clustering approach to Lagrangian vortex detection. *Phys. Rev. E* 2016; 93:063107 (17 pages).
- [24] Shi J, Malik J. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* 2000; 22:888–905.
- [25] Donzis DA, Sreenivasan KR. Short-term forecasts and scaling of intense events in turbulence. *J. Fluid Mech.* 2010; 647:13–26.
- [26] Fonda E, Iyer KP, Sreenivasan KR. Characterization and prediction of extreme events in turbulence. *APS Division of Fluid Dynamics Meeting Abstracts* 2017; E28.003:
- [27] King R, Hennigh O, Mohan A, Chertkov M. From deep to physics-informed learning of turbulence: Diagnostics. Preprint 2018; arXiv:1810.07785 (8 pages).
- [28] Li Y, Perlman E, Wan M, Yang Y, Meneveau C, Burns R, Chen S, Szalay A, Eyink G. A public turbulence database cluster and applications to study Lagrangian evolution of velocity increments in turbulence. *J. Turb.* 2008; 9:N31.
- [29] Mohan A, Daniel D, Chertkov M, Livescu D. Compressed Convolutional LSTM: An efficient deep learning framework to model high fidelity 3D turbulence. Preprint 2019; arXiv:1903.00033 (27 pages).
- [30] Fonda E, Pandey A, Schumacher J, Sreenivasan KR. Deep learning in turbulent convection networks. *Proc. Natl. Acad. Sci. USA* 2019; 116:8667–8672.
- [31] Ronneberger O, Fischer P, Brix T. U-Net: Convolutional Networks for Biomedical Image Segmentation. *Lecture Notes in Computational Science* 2015; 9351:234–241.
- [32] Kräuter R, Krasnov D, Pandey A, et al. Machine learning applications to convective turbulence. *Proceedings of NIC Symposium* 2020; 337–346.
- [33] Srinivasan PA, Guastoni L, Azizpour H, Schlatter P, Vinuesa R. Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* 2019; 4:054603 (15 pages).

- [34] Moehlis J, Faisst H, Eckhardt B. A low-dimensional model for turbulent shear flows. *New. J. Phys.* 2004; 6:56 (17 pages).
- [35] Waleffe F. On a self-sustaining process in shear flows. *Phys. Fluids* 1997; 9:883–900.
- [36] Schumacher J, Eckhardt B. Evolution of turbulent spots in a parallel shear flow. *Phys. Rev. E* 2001; 63:046307 (9 pages).
- [37] Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput.* 1997; 9:1735–1780.
- [38] Jaeger H. A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach. *GMD Report* 2002, 159:48 (46 pages).
- [39] Jaeger H, Haas H. Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communication. *Science* 2004; 304:78–80.
- [40] Lukoševičius M, Jaeger H. Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* 2009, 3:3, 127–149.
- [41] Lu Z, Pathak J, Hunt B, Girvan M, Brockett R, Ott E. Reservoir observers: Model-free inference of unmeasured variables in chaotic systems. *Chaos* 2017; 27:041102 (8 pages).
- [42] Kingma DP, Ba JL. Adam: A method for stochastic optimization. In Proceedings of the International Conference for Learning Representations 2015; arXiv:1412.6980v9 (15 Pages).
- [43] Strauss T, Wustlich W, and Labahn R. Design Strategies for Weight Matrices of Echo State Networks. *Neural Comput.* 2012; 24:12 (30 pages).
- [44] Carelo G, Cirac I, Cranmer K, Daudet L, Schuld M, Tishby N, Vogt-Maranto L, Zdeborova L. Machine learning and the physical sciences. *Rev. Mod. Phys.* 2019; 91:045002 (39 pages).
- [45] Kim T, Park E, Lee H, Moon Y-J, Bae S-H, Lim D, Jang S, Kim L, Cho I-H, Choi M, Cho K-S. Solar farside magnetograms from deep learning analysis of STEREO/EUVI data. *Nat. Astron.* 2019; 3:397–400.
- [46] Silburt, A., Ali-Diba, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., Menou, K. Lunar Crater Identification via Deep Learning Preprint 2018; arXiv:1803.02192v3 (51 pages).
- [47] Bellinger EP, Angelou GC, Hekker S, Basu S, Ball WH, Guggenberger E. Fundamental parameter of main-sequence stars in an instant with machine learning. *Astrophys. J.* 2016; 830:31 (20 pages).
- [48] Pearson, KA, Palafox, L Griffith, CA. Searching for exoplanets using artificial intelligence. *MNRAS* 2018; 474, 478–491
- [49] Alshehhi R, Rodenbeck K, Gizon L, Sreenivasan KR. Detection of exomoons in simulated light curves with a regularized convolutional neural network. Preprint.
- [50] Alshehhi R, Hanson CS, Gizon L, Hanasoge S. Supervised neural networks for helioseismic ring-diagram inversions. *A & A* 2019; 622: A124 (10 pages).
- [51] Löptien B, Gizon L, Birch AC et al. Global-scale equatorial Rossby waves as an essential component of solar internal dynamics. *Nat. Astron.* 2018; 2:568–573.
- [52] Verma K, Hanasoge S, Bhattacharya J, Antia HM, Krishnamurthi G. Asteroseismic determination of fundamental parameters of Sun-like stars using multilayered neural networks. *MNRAS* 2016; 461:4206–4214.
- [53] Baron D Machine learning in astronomy: a practical overview. arXiv 2019; 1904.9248
- [54] Vidal R, Bruna J, Giryes R, Soatto S. Mathematics of Deep Learning. In Proceedings of the 56th IEEE Conference on Decision and Control 2017; arXiv:1712.04741 (10 pages).
- [55] E, W. A Proposal on Machine Learning via Dynamical Systems. *Commun. Math. Stat.* 2017; 5:1–11.

**Table 1.** Details on the optimized multilayer perceptron (MLP), long short-time memory (LSTM), and echo state (ESN) architectures for the 9-mode model by Moehlis et al. [34]. In case of the ESN, we took a spectral radius  $\rho(\mathbf{A}) = 1.0$  and a density of active nodes in the reservoir of  $D = 0.2$ . Ridge regression involves a regularization term that is added to the cost functional. In the present case it has a prefactor of  $5 \times 10^{-2}$ .

Parameter	MLP	LSTM	ESN
Input dimension	4500	90	9
Output dimension	9	9	9
Hidden layers	4	1	
Neurons in hidden layers	90	90	–
Reservoir nodes	–	–	600
Activation function	Hyperbolic tangent	Hyperbolic tangent	Hyperbolic tangent
Optimization Algorithm	Adam [42]	Adam [42]	Ref. [41]
Counter overfitting strategy	Early stopping	Early stopping	Ridge regression

**Table 2.** Summary of training and testing results of the three compared neural networks.

Network	Training loss	Validation loss	Training time (minutes)	$E(\bar{u}_x) (\%)$	$E(\bar{u}'_x \bar{u}'_x) (\%)$	$E(\bar{u}'_x \bar{u}'_y) (\%)$
MLP	$2.0 \times 10^{-5}$	$3.1 \times 10^{-5}$	$O(10^2)$	12.24	19.95	2.94
LSTM	$2.8 \times 10^{-8}$	$3.6 \times 10^{-8}$	$O(10^1)$	1.01	6.64	0.90
ESN	$2.0 \times 10^{-7}$	$2.4 \times 10^{-7}$	$O(10^0)$	3.96	22.33	0.98