Research

Author for correspondence:
L. Magri
e-mail: lm547@cam.ac.uk

Short- and long-term predictions of chaotic flows and extreme events: a physics-constrained reservoir computing approach

N. A. K. Doan[1,2,5], W. Polifke[2] and L. Magri[3,1,4]

[1]Institute for Advanced Study, Technical University of Munich, Lichtenbergstrasse 2a, 85748 Garching, Germany
[2]Department of Mechanical Engineering, Technical University of Munich, Boltzmannstrasse 15, 85748 Garching, Germany
[3]Department of Aeronautics, Imperial College London, South Kensington, London, UK
[4]The Alan Turing Institute, 96 Euston Road, London NW1 2DB, UK
[5]Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

LM, 0000-0002-0657-2611

We propose a physics-constrained machine learning method—based on reservoir computing—to time-accurately predict extreme events and long-term velocity statistics in a model of chaotic flow. The method leverages the strengths of two different approaches: empirical modelling based on reservoir computing, which *learns* the chaotic dynamics from data only, and physical modelling based on conservation laws. This enables the reservoir computing framework to output physical predictions when training data are unavailable. We show that the combination of the two approaches is able to accurately reproduce the velocity statistics, and to predict the occurrence and amplitude of extreme events in a model of self-sustaining process in turbulence. In this flow, the extreme events are abrupt transitions from turbulent to quasi-laminar states, which are deterministic phenomena that cannot be traditionally predicted because of chaos. Furthermore, the physics-constrained machine learning method is shown to be robust with respect to noise. This work opens up new possibilities for

THE ROYAL SOCIETY
PUBLISHING

1. Introduction

Many fluid dynamics systems exhibit extreme events, which are violent and sudden changes of a flow from the average evolution [1]. Examples of extreme events in fluids are oceanic rogue waves [2], extreme patterns in atmospheric and climate science [3,4], intermittency in turbulence [5] and thermoacoustic instabilities in aeroengines and rocket motors [6], to name only a few. Although governed by deterministic equations such as conservation laws, extreme events occur in a seemingly random way. The time-accurate detection and prediction of the dynamics of extreme events can be achieved for a short time scale, which is the predictability time [7]. This is a roadblock for the *time-accurate* prediction because, after the predictability time, a tiny difference between the initial conditions, such as floating-point errors, is exponentially amplified. This is popularly known as the butterfly effect in chaos theory [8]. Because of this, the time-accurate prediction of extreme events remains an open problem [1].

Most state-of-the-art predictive approaches rely on statistical methods. Extreme value theory [9] and large deviation theory [10] characterize the probability of an event and the heavy tail of the probability density function, which can be used to compute the initial conditions with the highest probability of transitioning towards extreme values. Statistical approaches successfully identified precursors of turbulent channel flow relaminarization [11] and of nonlinear rogue waves [12]. Another approach for the prediction of extreme events is data-driven. Machine learning has proved successful at predicting the dynamics of some chaotic flows with accurate short-term predictions and long-term statistics. Vlachas *et al*. [13] used a long short-term memory (LSTM) network, which is a type of recurrent neural network (RNN) [14], to predict the evolution of the Kuramoto–Sivashinsky equation and of a barotropic climate model. They showed that the proposed neural network had a good short-term accuracy, and converged towards the correct invariant measures. A similar architecture was employed in [15] to simulate the evolution of a model of shear turbulence showing that it was capable of reproducing the moments of the velocity statistics. Another type of RNN—the echo state network (ESN) [16,17]—based on reservoir computing, can successfully learn the chaotic dynamics beyond the predictability time for short-term predictions [18–20] and ergodic averages [21]. Data-driven methods have the capability of predicting chaotic dynamics, but it is still an open question whether they can be robustly used to predict extreme events. Wan *et al*. [22] tackled this problem with a hybrid approach that combined a reduced-order model with an LSTM. They predicted the evolution of dissipation events in the Kolmogorov flow and intermittent transitions between two flow regimes in a model of barotropic flow. Sapsis [23] combined large deviation theory with a data-driven method to efficiently characterize the heavy tail of the distribution in the Kolmogorov flow.

Statistical methods provide a robust framework to identify precursors and calculate the probability of extreme events, but they do not provide a robust way to time-accurately predict their occurrence and amplitude. On the other hand, recurrent neural networks, such as LSTMs and ESNs, can predict the dynamics of chaotic systems by learning temporal patterns in data only, but, because they are fully data driven, they provide solutions that may violate the governing physical laws, such as momentum conservation. In fluid mechanics, machine-learned solutions, however, should obey physical principles such as momentum and mass conservation. This calls for embedding physical and domain knowledge in data-driven methods [20,24–27].

To achieve this physical embedding, various methods have been proposed in the past [24]. On the one hand, several works have explored the possibility of directly using physical systems as computational basis for deep learning [28–30]. Specifically, in these works, the dynamics of the Karman vortex sheet, or of nonlinear waves, were leveraged as a computational reservoir, or RNN, which were then used to compute and predict the dynamics of other systems. This

demonstrated the strong computational capability of such physical machine learning approaches. This approach worked particularly well when the physical system used as a reservoir had similarities with the system to be studied, for example, in flows [30].

On the other hand, another approach consists of including physical knowledge within traditional machine learning architectures [20,24,25,31,32], or modifying the target that the machine learning architecture has to learn. This automatically ensures physical conservation, such as by learning the Lagrangian of the system [33]. In [25,32], a physics-informed neural network was developed to approximate the solution of PDEs [25], or infer unmeasured variables from measurable quantities in flows [32]. This approach relied on a feedforward neural network combined with a physics-based loss to estimate the quantities to infer that are not present in the training data. This enhancement of the loss function with physical knowledge was also explored in [31], where the optimization process to identify the dynamics of a reduced order flow model was constrained with energy-preserving constraints in the Sparse Identification of Nonlinear Dynamics (SINDy) framework. The models obtained with those energy-preserving constraints compared favourably with reduced order models obtained with traditional Galerkin projection methods.

Further combinations of machine learning tools with physics-based methods were explored in the combination of computational fluid dynamics with machine learning [34,35]. For example, in [34], neural networks were used to discover improved spatial discretizations of PDEs on a coarse grid, which allowed for the accuracy to be equivalent to the accuracy of a finer grid, thereby enabling the accurate resolution of PDEs on a coarse mesh. Following similar ideas, in [35], convolutional neural networks were used within a direct numerical simulation solver to learn the correction required for a coarse simulation to reach the accuracy of a finer simulation allowing for a speed-up of the calculation while improving the accuracy drastically.

The discussion above highlights that additional research is necessary to enable purely data-driven machine learning approaches to time-accurately predict the evolution of chaotic systems during extreme events, therefore requiring the integration of some physical knowledge. In this paper, we propose a specific method based on a physics-based modification of the loss function used during the training of the machine learning model.

Specifically, the objective of this paper is to propose a machine learning method that (i) produces physical solutions to time-accurately predict extreme events in a qualitative model of shear turbulence and (ii) reproduces the long-term statistics. To achieve this objective, we will start from a reservoir computing framework based on the ESN and show that, by embedding physical knowledge during the training, the proposed *physics-informed* machine learning framework can achieve short- and long-term time-accurate predictions. This paper studies for the first time the application of such a hybrid physics-informed reservoir computing framework to the extreme event predictions where the inclusion of such physics-based knowledge can produce a marked improvement compared to purely data-driven approaches given that extreme events are generally rare within any dataset. The flow configuration is presented in §2. The physics-informed echo state network (PI-ESN) is developed in §3. Results are shown in §4. The long-term statistical behaviour of the network is discussed in §4a. Short-term predictions of extreme events are analysed in §4b. The robustness of the overall architecture with respect to noise is presented in §4c. A final discussion with future directions concludes the paper.

2. Extreme events in a model of chaotic flow

We regard the chaotic flow as an autonomous dynamical system

$$\dot{y} = \mathcal{N}(y), \quad y(0) = y_0 \tag{2.1}$$

where $\dot{(\,)}$ is the temporal derivative; and $\mathcal{N}$ is a deterministic nonlinear differential operator, which encapsulates the numerical discretization of the spatial derivatives and boundary conditions (if any).

The flow is governed by momentum and mass conservation laws, i.e. the Navier–Stokes equations, which were reduced in form by Moehlis, Faisst and Eckhart (MFE) [36]. This model, which was inspired by earlier works [37,38], provides the nonlinear operator $\mathcal{N}$. This is called the MFE model for brevity. The MFE model captures the qualitative features of the transition from turbulence to quasi-laminar states such as the exponential distribution of turbulent lifetimes. The velocity field in the model is decomposed as

$$v(x, t) = \sum_{i=1}^{9} a_i(t) v_i(x),\qquad(2.2)$$

where $v_i(x)$ are spatial Fourier modes (or combinations of them) [36]. The Navier–Stokes equations are projected onto $v_i(x)$ to yield nine ordinary differential equations for the modes' amplitudes, a_i, which are nonlinearly coupled. Consequently, the state vector is $y = \{a_i\}_1^9$. All the variables are non-dimensional [36]. Physically, v_1 is the laminar profile mode; v_2 is the streak mode; v_3 is the downstream vortex mode; v_4 and v_5 are the spanwise flow modes; v_6 and v_7 are the normal vortex modes; v_8 is the three-dimensional mode; and v_9 is the modification of the mean profile caused by turbulence. The flow has a fixed point $a_1 = 1$, $a_2 = \cdots = a_9 = 0$, which is a laminar state. The electronic supplementary material reports the expressions for v_i and the equations for a_i [36]. Detailed analysis of the MFE model was performed by Kim & Moehlis [39] and Joglekar et al. [40].

The flow under investigation is incompressible. The domain is a cuboid of size $L_x \times L_y \times L_z$ between two infinite parallel walls at $y = 0$ and $y = L_y$, which are periodic in the x- and z-directions. The domain size is $L_x = 1.75\pi$, $L_y = 2$ and $L_z = 1.2\pi$. The Reynolds number is 600. A sinusoidal volume force is applied in the y-direction. The initial condition (electronic supplementary material) is such that the turbulent flow displays chaotic bursts between the fully turbulent and quasi-laminar states. These are the extreme turbulent events we wish to predict. The governing equations are integrated in time with a 4th-order Runge–Kutta scheme with a time step $\Delta t = 0.25$ to provide the evolution of the nine modes, a_i, from $t = 0$ to $t = 80\,000$. The evolution of the kinetic energy, $k = 0.5 \sum_{i=1}^{9} a_i^2$, is shown in figure 1 for the first 30 000 timesteps. The time is normalized by the largest Lyapunov exponent, $\lambda_{\max} \approx 0.0244$, which is calculated as the average logarithmic error growth rate between two nearby trajectories [7]. The Lyapunov time scale, $\lambda_{\max}^{-1}$, provides an estimate of the predictability time, which is used to define the non-dimensional time

$$t^+ = \frac{t}{\lambda_{\max}^{-1}}.\qquad(2.3)$$

The kinetic energy, k, has sudden large bursts that arise from a chaotic oscillation with a small amplitude. Each burst is a quasi-relaminarization event, which occurs in three phases (figure 1): (i) the originally laminar velocity profile becomes unstable and breaks down into vortices due to the shear imposed by the volume force (panels 5–7); (ii) the vortices align to become streaks (panels 8–9 and 1–2); and (iii) the streaks break down leading to flow relaminarization (panels 3–5).

3. Physics-constrained reservoir computing

To learn the reduced-order dynamics of shear turbulence, we constrain the physical knowledge of the governing equations into a reservoir computing data-driven method based on the ESN [16,17]: the PI-ESN [20]. A schematic of the network is shown in figure 2.

We have training data with an input time series $u(n) \in \mathbb{R}^{N_u}$ and a target time series $y(n) \in \mathbb{R}^{N_y}$, where $n = 0, 1, 2, \ldots, N_t$ are the discrete time instants that span from 0 to $T = N_t \Delta t$. During prediction, the target at time n becomes the input at time $n + 1$, i.e. $u(n + 1) = y(n)$. The training of the PI-ESN is achieved by (i) minimizing the error between the prediction, $\widehat{y}(n)$, and the target data $y(n)$ when the PI-ESN is excited with the input, $u(n)$ (figure 2a), and (ii) enforcing that the prediction does not violate the physical constraints. To enforce (ii), we observe that a solution
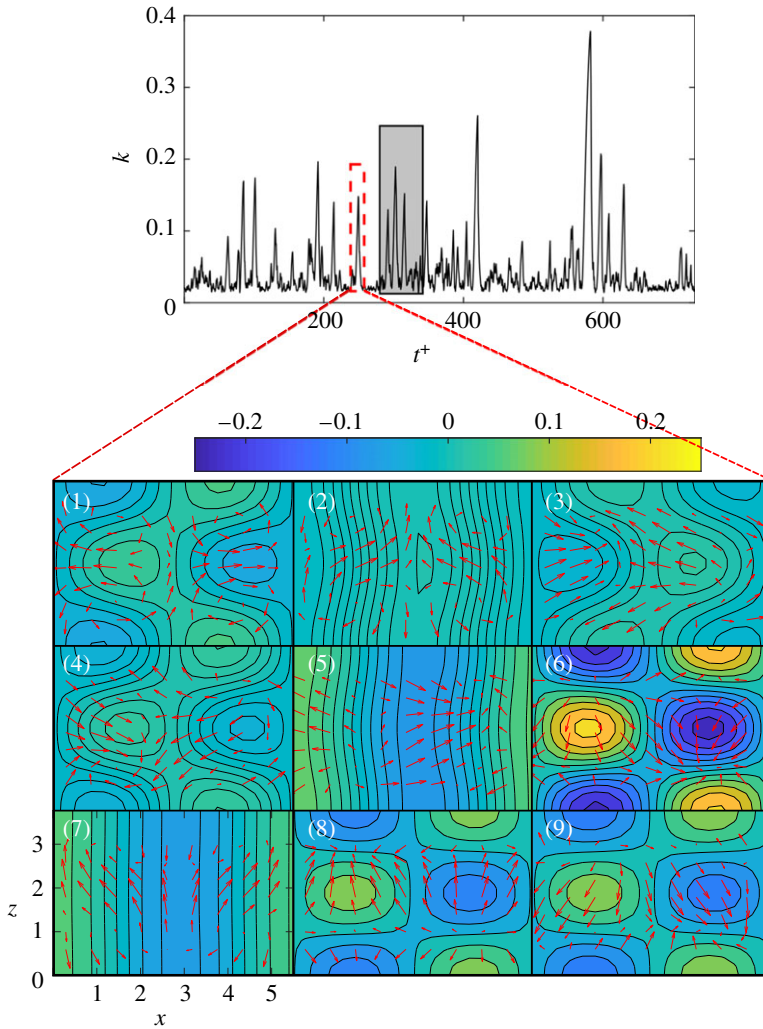
Figure 1. Top panel: kinetic energy, k. The grey box indicates the training window of the PI-ESN. Bottom panel: velocity field in the mid-y plane. The arrows indicate the in-plane velocity (x–z directions), and the colour maps indicate the out-of-plane velocity. t^+ is the time normalized by the largest Lyapunov exponent. (Online version in colour.)

of the chaotic flow, $y = \{a_i\}_1^9$, is such that the *physical error* (also known as the residual) is zero, i.e. $\mathcal{F}(y) \equiv \dot{y} - \mathcal{N}(y) = 0$. To estimate the physical error beyond the training data, after exciting the PI-ESN with the training data, the PI-ESN is then looped back to its input (figure 2b) to obtain the predictions $\{\widehat{y}(N_t + p)\}_{p=1}^{N_p}$ in the time window with no training data, $(T + \Delta t) \leq t \leq (T + N_p \Delta t)$. The number of collocation points, N_p, is user-defined. The physical error $\mathcal{F}(\widehat{y}(N_t + p))$ is evaluated to train the PI-ESN such that the sum of (i) the physical error between the prediction and the available data from $t = 0$ to $t = T$, E_d, and (ii) the physical error for $t > T$, E_p, is minimized. Mathematically, we wish to find $\widehat{y}(n)$ for $n = 0, 1, \ldots, N_t + N_p$ that minimizes

$$E_{\text{tot}}^P = \underbrace{\frac{1}{N_t + 1} \sum_{n=0}^{N_t} ||\widehat{y}(n) - y(n)||^2}_{E_d} + \underbrace{\frac{1}{N_p} \sum_{p=1}^{N_p} ||\mathcal{F}(\widehat{y}(N_t + p))||^2}_{E_p}, \tag{3.1}$$

where $||\cdot||$ is the Euclidean norm. The PI-ESN is straightforward to implement because it requires only cheap residual calculations at the collocation points, i.e. it does not require the exact solution.
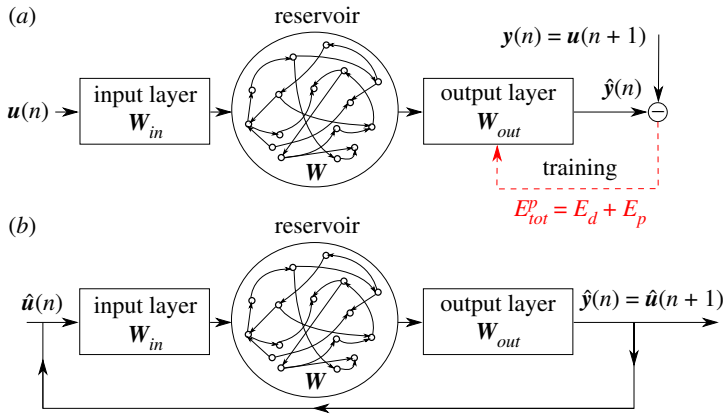
Figure 2. PI-ESN during (*a*) training and (*b*) prediction. (Online version in colour.)

Practically, the estimation of the time-derivative in $\mathcal{F}(\widehat{\boldsymbol{y}})$ is performed with finite difference and, for a specific time-step k, the physical error is estimated as $\mathcal{F}(\widehat{\boldsymbol{y}}) \approx (\widehat{\boldsymbol{y}}_{k+1} - \widehat{\boldsymbol{y}}_k)/\Delta t - \mathcal{N}(\widehat{\boldsymbol{y}}_k)$. Therefore, this loss function penalizes solutions that do not fulfil the governing equations (to a numerical tolerance), similarly to physics-informed feedforward neural networks to approximate the solution of PDEs [25] or infer unmeasured quantities [32]. Here, we apply this physics-based loss to an echo state network to obtain a time-accurate surrogate model capable of autonomously reproducing the dynamics of the system.

(a) Network architecture

The architecture of the PI-ESN follows that of the ESN, which consists of an input matrix $\boldsymbol{W}_{\text{in}} \in \mathbb{R}^{N_x \times N_u}$, which is a sparse matrix; a *reservoir* that contains N_x neurons that are connected by the recurrent weight matrix $\boldsymbol{W} \in \mathbb{R}^{N_x \times N_x}$, which is another sparse matrix; and the output matrix $\boldsymbol{W}_{\text{out}} \in \mathbb{R}^{N_y \times N_x}$. The input time series, $\boldsymbol{u}(n)$, is connected to the reservoir through $\boldsymbol{W}_{\text{in}}$ to excite the states of the neurons, $\boldsymbol{x}$, as

$$\boldsymbol{x}(n + 1) = \tanh(\boldsymbol{W}\boldsymbol{x}(n) + \boldsymbol{W}_{\text{in}}\boldsymbol{u}(n + 1)), \tag{3.2}$$

where $\tanh(\cdot)$ is the activation function. The output of the PI-ESN, $\widehat{\boldsymbol{y}}(n)$, is computed by linear combination of the reservoir states as $\widehat{\boldsymbol{y}}(n) = \boldsymbol{W}_{\text{out}}\boldsymbol{x}(n)$. The matrices $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}$ are randomly generated and fixed [41]. Only $\boldsymbol{W}_{\text{out}}$ is trained to minimize (3.1). Following [19], each row of $\boldsymbol{W}_{\text{in}}$ has only one non-zero element, which is drawn from a uniform distribution over $[-\sigma_{\text{in}}, \sigma_{\text{in}}]$; $\boldsymbol{W}$ has an average connectivity $\langle d \rangle$, whose non-zero elements are drawn from a uniform distribution over the interval $[-1, 1]$; and $\boldsymbol{W}$ is scaled such that its largest eigenvalue is $\Lambda \leq 1$, which ensures the echo state property [41]. The same quadratic transformation of the reservoir state before readout as in [19] is also applied.

The training of the PI-ESN is achieved in two steps. First, the network is initialized by an output matrix, $\boldsymbol{W}_{\text{out}}$, that minimizes a data-only cost functional $E_{\text{tot}}^{NP} = E_d + \gamma ||\boldsymbol{w}_{\text{out},i}||^2$, where γ is a Tikhonov regularization factor and $\boldsymbol{w}_{\text{out},i}$ denotes the ith row of $\boldsymbol{W}_{\text{out}}$. This is the output matrix of the conventional ESN [18]. Second, the physical error (3.1) is minimized with the L-BFGS method [42], which is a quasi-Newton optimization algorithm. On the one hand, the proposed PI-ESN architecture has similarities to an Elman-RNN architecture because the PI-ESN can be interpreted as a three-layer RNN trained with a gradient-based optimizer. On the other hand, the PI-ESN is different from an Elmann-RNN because it employs a physics-based loss function, whose additional training is limited to $\boldsymbol{W}_{\text{out}}$ to reduce the computational cost. In principle, other parts of the ESN, such as the input and recurrent weight matrices, $\boldsymbol{W}_{\text{in}}$ and $\boldsymbol{W}$, can be trained. This is not attempted here because the training of $\boldsymbol{W}_{\text{out}}$ is sufficient to show a substantial improvement in the prediction accuracy (§4).

4. Results

A grid search provides the hyperparameters $\Lambda = 0.9$, $\sigma_{in} = 1.0$, $\langle d \rangle = 3$, $\gamma = 10^{-6}$, which enable accurate predictions in the range of $N_x = [500, 3000]$ neurons (electronic supplementary material). These optimal hyperparameters are determined for a data-only ESN, and are re-used for the PI-ESN, unless mentioned otherwise, to provide a base assessment of the improvement that can be obtained by introducing the physics-based loss. Only $t = 2500$ time units (equivalent to $t^+ \approx 61$) in the window $t = [11\,500, 14\,000]$ (equivalent to $t^+ \approx [280, 341]$ in the grey box of figure 1) are used for training. The data beyond this time window are used for validation only. We use $N_p = 5000$ collocation points (equivalent to $t = 1250$ or $t^+ \approx 30.5$), which provide a sufficient number of predictions beyond the training data with a relatively low computational time.

(a) Long-term statistical behaviour

A long-term prediction of the modes' amplitudes from the trained ESN and PI-ESN is shown in figure 3 for a reservoir of 1500 units. This prediction is made for 10 000 time units, which corresponds to approximately 240 Lyapunov times. Out of that time-series, the first 5000 time units are shown in figure 3. Both the ESN and the PI-ESN qualitatively reproduce the dynamics of the MFE model. In particular, both the ESN and PI-ESN exhibit transition towards a quasi-laminar state with large growth of the first mode, a_1, and an associated increase in the kinetic energy, k.

A statistical assessment of the performance of the PI-ESN is analysed in terms of the probability density function (PDF) of k and a_1 (figure 4). The PDFs are computed from a time series of 240 Lyapunov times, which is longer than the time series of figure 3 to ensure statistical convergence. The PI-ESN captures more accurately the tail of the PDFs of k and a_1 as compared to the data-only ESN. This means that constraining the physics reproduces more accurately the occurrence of extreme events (associated with large values of a_1 and k) from a statistical point of view.

An additional quantitative comparison, with the statistics of the velocity field, is shown in figure 5 for ESNs and PI-ESNs of different reservoir sizes. The statistics, which are collected for 5000 time units, are computed by averaging the velocity along the two periodic directions, x and z, and time.

Figure 5 shows that, for small reservoirs (500 units), both the ESN and PI-ESNs do not accurately reproduce the mean streamwise velocity profile and the variance. This is because there are not enough neurons in the reservoir to correctly reproduce the intricate dynamics of the MFE model and its long-term evolution. However, as the reservoir size increases to 1500 units, the PI-ESN accurately captures the mean velocity profile, the variance and the Reynolds stress. Larger reservoirs, with up to 3000 units, have an accuracy that is similar to the reservoir with 1500 units. Therefore, the resulting velocity profiles are shown only for 1500 units in figure 5. On the other hand, the data-only ESN solution is less accurate. A larger reservoir of 2500 units is required for the data-only ESN to accurately reproduce the statistics of the MFE model (result not shown). This highlights how the physical constraint augments the information used during the training of the PI-ESN allowing it to better capture the long-term evolution compared to a data-only ESN. The PI-ESN approach needs a relatively small training dataset of 2500 time units to learn the long-term statistics of the system. Compared to previous studies with LSTM units, these training data are two to three orders of magnitude smaller [15].

(b) Short-term extreme events prediction

Figure 6 shows the evolution of three modes during the extreme event in the dashed red box of the top panel of figure 1. The PI-ESN solution (solid blue line) and the conventional ESN solution (dashed red line) are computed with a reservoir of $N_x = 3000$ units. Both solutions are compared
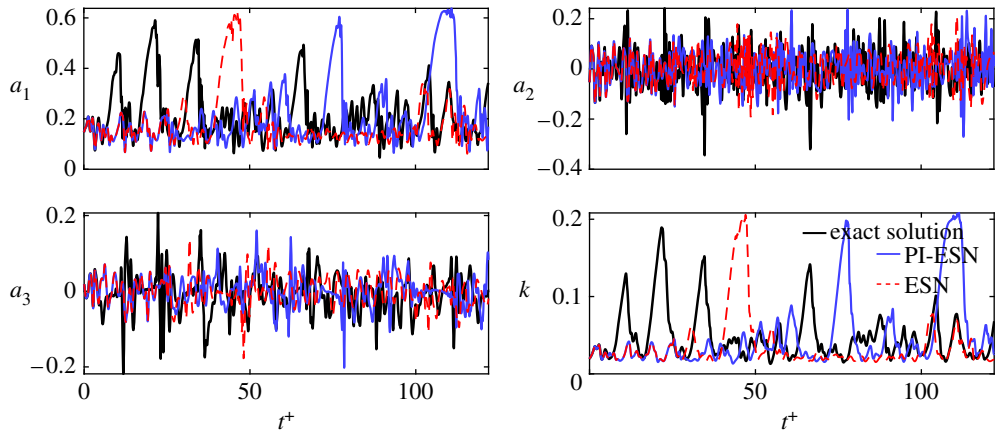
Figure 3. Long-term evolution of the modes' amplitudes: exact evolution (thick black lines), ESN solution (dashed red lines), PI-ESN solution (blue full lines) with reservoir of 1500 units. (Online version in colour.)
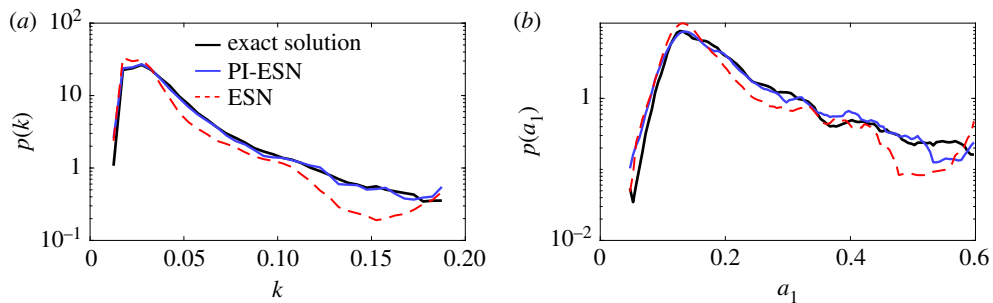

Figure 4. Probability density function of (*a*) turbulent kinetic energy, *k*, and (*b*) first modal coefficient, a_1, of the long-term evolution: exact evolution (thick black line), echo state network (dashed red lines) and physics-informed echo state network (blue full line). (Online version in colour.)

against the exact solution from numerical integration (solid black line). The normalized error between the exact evolution and the PI-ESN/ESN predictions is computed as

$$E(n) = \frac{||\boldsymbol{y}(n) - \widehat{\boldsymbol{y}}(n)||}{\sqrt{\frac{1}{N_t} \sum_{n=1}^{N_t} ||\boldsymbol{y}(n)||}}, \tag{4.1}$$

where the denominator of the error cannot be zero because the fixed point of the MFE model has $||\boldsymbol{y}|| = 1$ and the system has unsteady chaotic oscillations. Although the same training data are used for both the PI-ESN and the conventional ESN, the PI-ESN has a significantly higher capability in accurately predicting autonomously the real evolution than the conventional ESN. To compare the performance, we define the predictability horizon as the time required for E to exceed the threshold 0.2 from the same initial condition. The predictability horizon of the PI-ESN is ∼2 Lyapunov times longer than the predictability horizon of the conventional ESN. This improvement is achieved by enforcing the prior physical knowledge of the flow, whose evolution must fulfil the momentum and mass conservation laws. As shown in figure 6, until $t^+ \approx 2.14$, both ESN and PI-ESN accurately predict the flow evolution. The predicted solution from the conventional ESN starts diverging from the exact evolution at $t^+ \approx 3.21$, which leads to a completely different solution during the extreme event. On the other hand, the PI-ESN is able to time-accurately predict the occurrence and the amplitude of the extreme event. After the
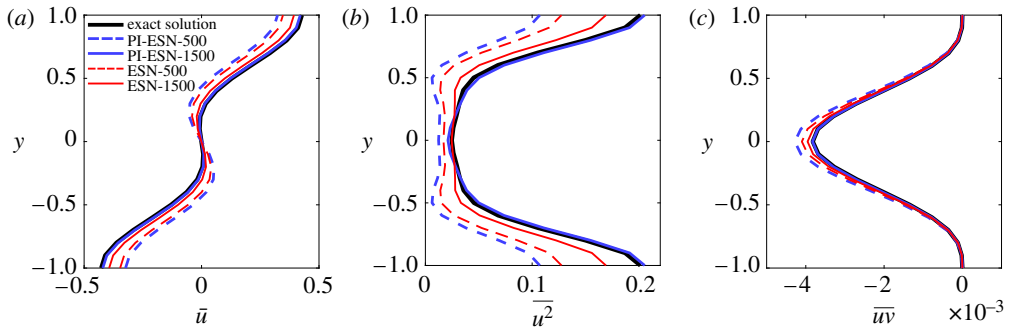
Figure 5. Profile of (*a*) mean streamwise velocity, (*b*) variance and (*c*) mean Reynolds stress. (Online version in colour.)
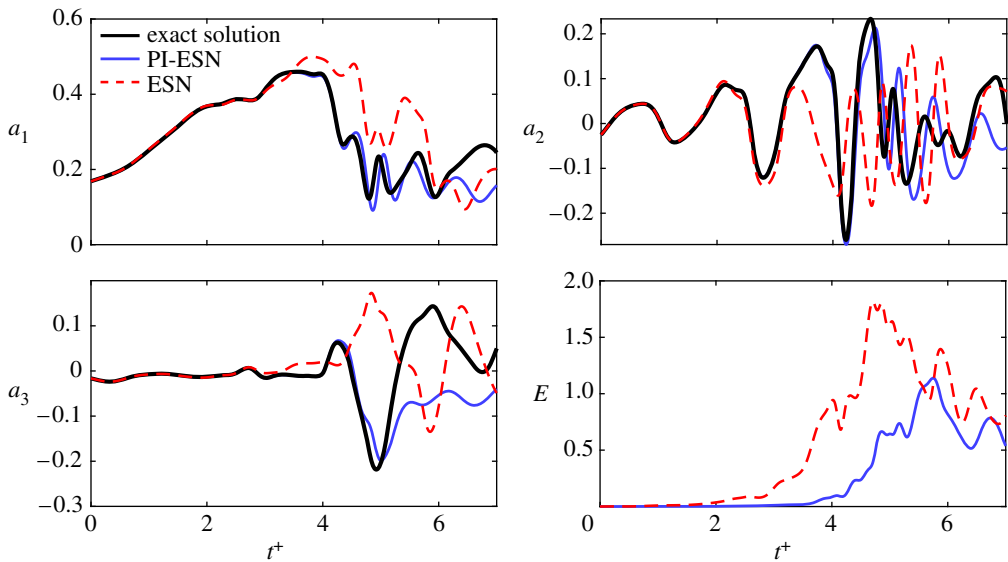

Figure 6. Evolution of modes a_1, a_2, a_3 during the extreme event of figure 1: exact evolution in solid black line; prediction with PI-ESN in solid blue line; and prediction with fully data-driven ESN in dashed red line. The reservoir has $N_x = 3000$ neurons. E is the error (equation (4.1)). (Online version in colour.)

event has occurred, the solution diverges because the butterfly effect is significant. The velocity fields predicted by the conventional ESN and PI-ESN are shown in figure 7*a*,*b*, respectively. The solutions are shown at the same times as the exact solution in panels (3–5) of figure 1. The bottom rows of figure 7*a*,*b* show the normalized absolute error between the predicted velocity field and the exact velocity field. The discrepancy in the velocity field is mainly due to the error on the prediction of the downstream vortex mode, a_3 (figure 6). On the one hand, because no physical knowledge is constrained in the conventional ESN, the sign and amplitude of a_3 are incorrectly predicted, which means that the out-of-plane velocity evolves in the opposite direction of the exact solution. On the other hand, the PI-ESN is able to predict satisfactorily both the in-plane velocity and the out-of-plane velocity during the extreme event.

To quantitatively assess the robustness of these results, we compute the average predictability horizon of the machines with no further training. We follow the following steps: (i) by inspection of our dataset (a subset of which is shown in figure 1), we define events as extreme when their kinetic energy is $k \geq 0.1$; (ii) we identify the times at which all the extreme events start in our
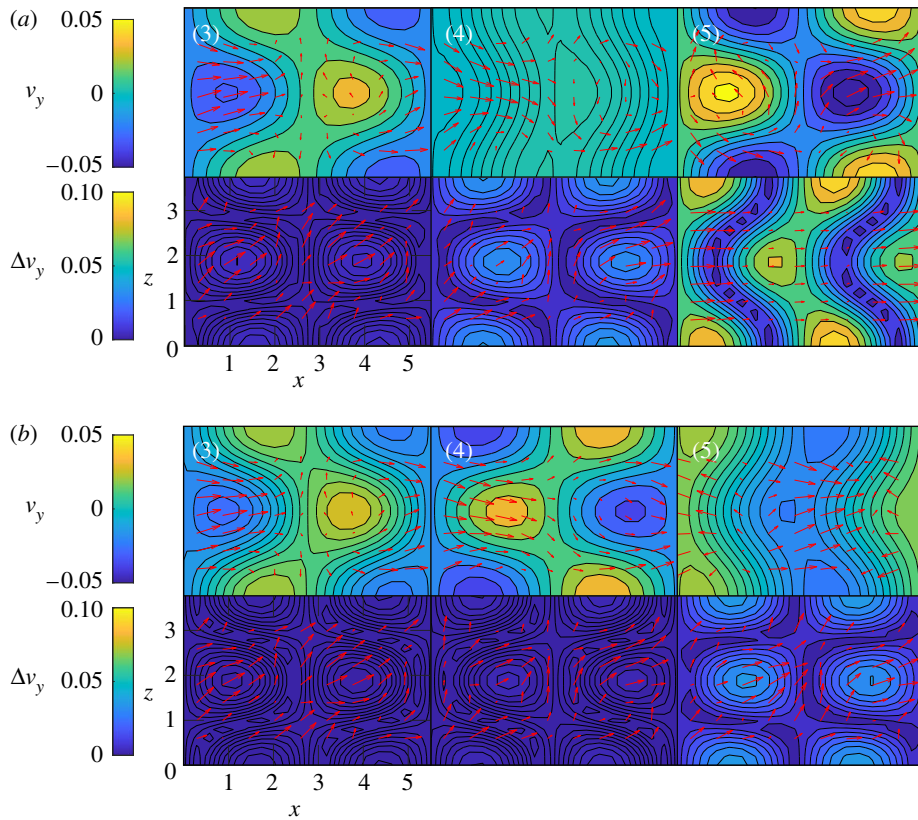
Figure 7. Evolution of the velocity field (top rows) and the normalized error on the out-of-plane velocity (bottom rows) in the velocity field in the mid-y plane at the same time instants as panels (3)–(5) of figure 1. Predictions from (*a*) the conventional ESN and (*b*) the PI-ESN. The arrows indicate the in-plane velocity (x–z directions) and the coloured contour indicates the out-of-plane velocity. The panels correspond to $t^+ \approx [2.14, 3.21, 4.27]$ of figure 6. (Online version in colour.)

dataset (50 extreme events are identified); (iii) for each time, the exact initial condition at $t^+ \approx 0.61$ just before the time instant in which the extreme events starts is inputted in the PI-ESN and ESN; (iv) the machines are time evolved to provide the prediction; and (v) the predictability time is computed by averaging over all the extreme events in the dataset. The mean predictability time and the standard deviation are computed with validation data containing 50 extreme events. The results are parametrized with the size of the reservoir, N_x (figure 8). On the one hand, for small reservoirs ($N_x \lesssim 2000$), the performances of the ESN and PI-ESN are comparable. This means that the performance is more sensitive to the data cost functional, E_d, than the physical error, E_p. On the other hand, for larger reservoirs ($N_x \gtrsim 2000$), the physical knowledge is fully exploited by the PI-ESN. This means that the performance becomes markedly sensitive to the physical error, E_p. This results in an improvement in the average predictability of up to ≈ 1.5 Lyapunov times. Because an extreme event takes ≈ 3 Lyapunov time on average, the improved predictability time of the PI-ESN is key to the time-accurate prediction of the occurrence and amplitude of the extreme events. To further assess the accuracy of the PI-ESN with respect to the data-only ESN, we analyse data-only ESNs trained with different Tikhonov regularization factors ($\gamma = 10^{-5}$ and $\gamma = 10^{-7}$) in figure 8. The PI-ESN outperforms the data-only ESN.

For completeness, we re-tune the hyperparameter of the PI-ESN by performing a grid search, which provides the optimal parameters $\Lambda = 0.7$, $\sigma_{\text{in}} = 0.9$, $\langle d \rangle = 3$. The extreme event prediction horizon for the PI-ESN with these new optimal hyperparameters is shown in figure 8 (in green). A higher prediction horizon can be achieved with the hyperparameter re-tuning.
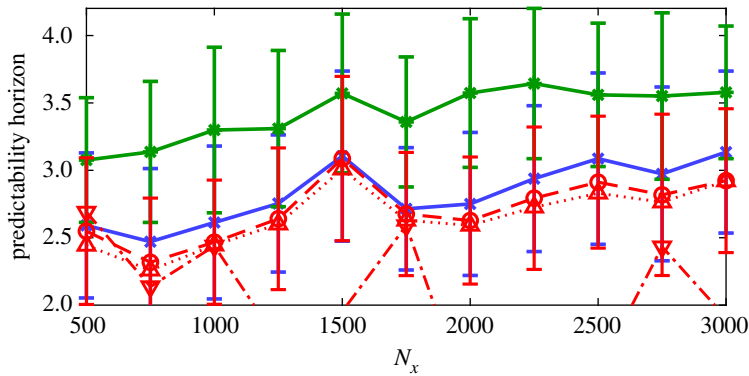
Figure 8. Comparison of the average predictability horizons of extreme events from the PI-ESN with original hyperparameters (full blue line with crosses), the PI-ESN with optimized hyperparameters (full green line with stars), ESN with $\gamma = 10^{-6}$ (dashed red line with circles), $\gamma = 10^{-5}$ (dotted red line with upward triangles) and $\gamma = 10^{-7}$ (dash-dotted line with downward triangles) for the prediction of 50 extreme events. The error bars show half of the standard deviation. N_x is the number of neurons. (Online version in colour.)

(c) Robustness to noisy data

Because the accuracy of the ESN and PI-ESN—and more generally, of any data-driven method—depends on the quality of the training data, the effect of additive noise is analysed. Noise is added to the training dataset presented in figure 1 with signal-to-noise ratios (SNR) of 30 dB and 40 dB, which are representative of noise in experiments [43]. The ESN and PI-ESN are trained on the noisy data, and the accuracy of the long- and short-term prediction is analysed. The dataset length and hyperparameters are the same as those of §§4a and 4b. Here, because the training dataset is small (2500 time units, which correspond to 60 Lyapunov times), noise has a marked (detrimental) impact on the training of the ESN for learning the physical dynamics. (By contrast, when the training dataset is large—i.e. the data contain sufficient physical dynamics for the machine to be trained—a small addition of noise can have a positive effect on the robustness of the ESN [41].)

Figure 9 shows the evolution of three modes, the normalized error and the noisy solution. Similarly to the noise-free case (figure 6), the PI-ESN is quantitatively more predictive than the data-only ESN. For example, the PI-ESN predicts the growth of the a_1 mode, which is physically the quasi-laminar profile forming during the extreme event. Although after 4 Lyapunov times the PI-ESN diverges due to the high sensitivity of chaotic flows to noise, the overall statistics are well predicted, as discussed next. One possible strategy to improve the performance is to increase the number of collocation points in the training of the PI-ESN.

The statistical assessment is shown in figure 10. Compared to the noise-free case of §4b, the predictability time of extreme events becomes smaller as the noise level becomes larger. This loss in accuracy is due to the noise in the training data. Both the ESN and PI-ESN minimize the error on the noisy data, E_d, which causes the machines to try to fit the noisy data, resulting in the reduction of predictive capability. This loss becomes more significant as the noise increases. If the number of neurons in the reservoir is further increased, the ESN and, to a lesser extent, the PI-ESN will start overfitting the noise. The PI-ESN mitigates the loss of accuracy through the physical loss, E_p, which acts as a regularization term that filters out the noise. This allows the PI-ESN to maintain a higher accuracy in the prediction of the extreme event (figure 6). As a result, the PI-ESN provides a longer accurate prediction of the extreme events, by up to 0.4 Lyapunov time, compared to the data-only ESN. This effect becomes more significant as the number of neurons in the reservoir increases. In other words, the physical constraint in the PI-ESN acts as a denoizer. In the presence of noise, the PI-ESN does not attempt to fit the noisy data, but it balances the fit with
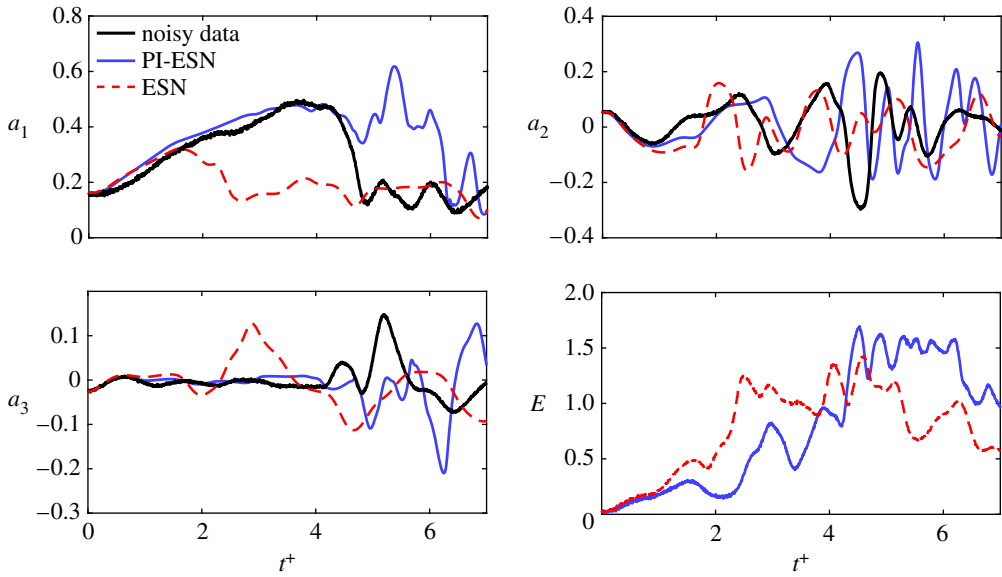
12

royalsocietypublishing.org/journal/rspa *Proc. R. Soc. A* **477**: 20210135

Figure 9. Evolution of a_1, a_2, a_3 during the extreme event of figure 1: noisy evolution in solid black line; prediction with PI-ESN in solid blue line; and prediction with fully data-driven ESN in dashed red line. The reservoir has $N_x = 3000$ neurons. E is the error (equation (4.1)). The noisy data correspond to the case with SNR = 30 dB. (Online version in colour.)
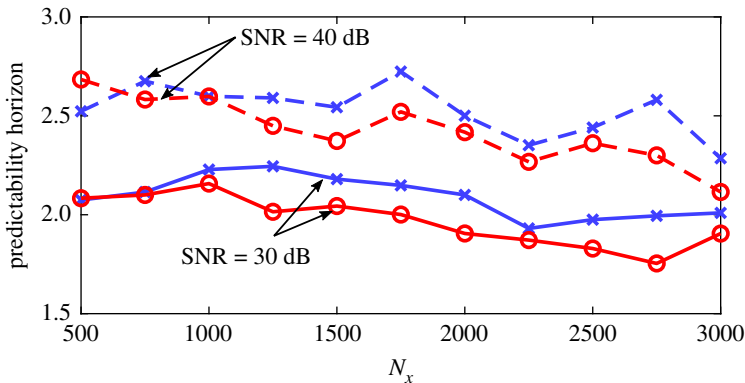

Figure 10. Comparison of the average predictability horizons of extreme events from the PI-ESN (blue full and dashed lines with crosses) and ESN (red full and dashed lines with circles) for all the extreme events in the dataset when trained with noisy database. Full lines: SNR = 40 dB, dashed lines: SNR = 30 dB. N_x is the number of neurons. (Online version in colour.)

the physics. The physical loss allows the PI-ESN to determine an appropriate physical prediction, which deviates from the noisy data to approach the physical dynamics. This denoizing property of the PI-ESN is consistent with previous studies [44,45]. This overcomes the lack of robustness of the data-only ESN, which cannot discriminate the noise from the actual flow dynamics.

Finally, figure 11 compares the statistics of the velocity obtained from a long-term prediction of 5000 time units for a reservoir size of 1500 units (similarly to the noise-free case of figure 5). Consistently with the short-term performance, the accuracy on the statistics decreases as the noise level increases in the training data. For both noise levels, the PI-ESN outperforms the data-only ESN.
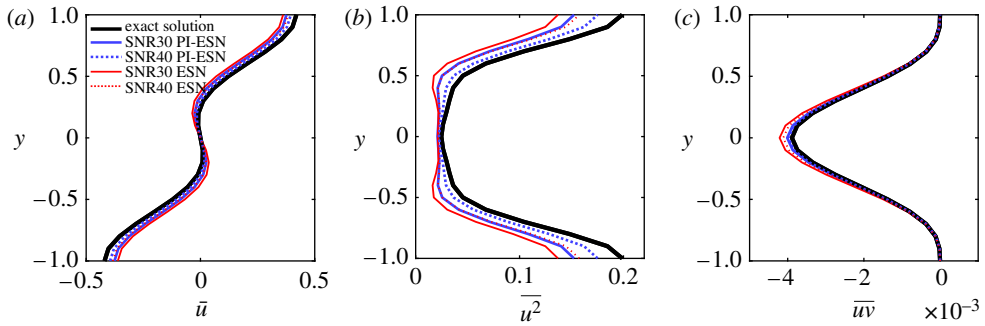
Figure 11. Profile of (*a*) mean streamwise velocity, (*b*) variance and (*c*) mean Reynolds stress. (Online version in colour.)

5. Final discussion and future directions

We propose a PI-ESN, which combines empirical modelling, based on reservoir computing, with physical modelling, based on conservation laws, to time-accurately predict extreme events (short-term dynamics) and the statistics (long-term dynamics) of a chaotic flow. We compare the performance of the PI-ESN with a fully-data driven ESN. The former is a physics-constrained machine, whereas the latter is a physics-unaware machine because it is trained with data only.

In the PI-ESN, the physical error from the conservation laws is minimized beyond the training data. This brings in crucial information, which is exploited in two ways: (i) with the same amount of available data, the PI-ESN solution is accurate for a longer time than the conventional ESN solution, i.e. the information from physical knowledge enhances the accuracy of the autonomous prediction of the PI-ESN, and (ii) fewer data are required to obtain the same accuracy as the conventional ESN. Here, we take advantage of property (i) for the prediction of extreme events in a model of turbulence, and the statistics of velocity. Extreme events may be generally rare, which makes it difficult for physics-unaware data-driven methods to be trained. By contrast, constraining the physics enables the PI-ESN to predict chaotic dynamics that cannot be inferred from data only. Additionally, the PI-ESN well captures the long-term flow statistics with relatively small training data. Finally, the approach also shows robustness with respect to noise.

It should be noted that the framework proposed in this paper relies on knowing the governing equations of the system studied, which is an ideal case. Cases when this knowledge is not perfect, for example, when only approximate equations are available, will constitute the scope for future work. In addition, the possibility of applying the proposed method to larger dimensional flow systems, such as the full Navier–Stokes equations, will also be explored. This can be achieved by using autoencoders to learn the spatial correlations, as shown in [46]. These avenues will be investigated in future research.

References

1. Farazmand M, Sapsis TP. 2019 Extreme events: mechanisms and prediction. *Appl. Mech. Rev.* **71**, 050801. (doi:10.1115/1.4042065)

2. Dysthe K, Krogstad HE, Müller P. 2008 Oceanic rogue waves. *Annu. Rev. Fluid Mech.* **40**, 287–310. (doi:10.1146/annurev.fluid.40.111406.102203)

3. Easterling DR *et al.* 2000 Observed variability and trends in extreme climate events: a brief review. *Bull. Am. Meteorol. Soc.* **81**, 417–425. (doi:10.1175/1520-0477(2000)081<0417:OVATIE>2.3.CO;2)

4. Majda AJ. 2012 Challenges in climate science and contemporary applied mathematics. *Commun. Pure Appl. Math.* **65**, 920–948. (doi:10.1002/cpa.21401)

5. Platt N, Sirovich L, Fitzmaurice N. 1991 An investigation of chaotic Kolmogorov flows. *Phys. Fluids A* **3**, 681–696. (doi:10.1063/1.858074)

6. Lieuwen TC, Yang V. 2006 *Combustion instabilities in gas turbine engines*. Reston, VA: American Institute of Aeronautics and Astronautics.

7. Boffetta G, Cencini M, Falcioni M, Vulpiani A. 2002 Predictability: a way to characterize complexity. *Phys. Rep.* **356**, 367–474. (doi:10.1016/S0370-1573(01)00025-4)

8. Lorenz EN. 1963 Deterministic nonperiodic flow. *J. Atmos. Sci.* **20**, 130–141. (doi:10.1175/1520-0469(1963)020<0130:DNF>2.0.CO;2)

9. Nicodemi M. 2015 *Extreme value statistics*. New York, NY: Springer.

10. Varadhan SRS. 2008 Large deviations. *Ann. Probab.* **36**, 397–419.

11. Blonigan PJ, Farazmand M, Sapsis TP. 2019 Are extreme dissipation events predictable in turbulent fluid flows? *Phys. Rev. Fluids* **4**, 044606. (doi:10.1103/PhysRevFluids.4.044606)

12. Dematteis G, Grafke T, Vanden-Eijnden E. 2018 Rogue waves and large deviations in deep sea. *Proc. Natl Acad. Sci. USA* **115**, 855–860. (doi:10.1073/pnas.1710670115)

13. Vlachas PR, Byeon W, Wan ZY, Sapsis TP, Koumoutsakos P. 2018 Data-driven forecasting of high-dimensional chaotic systems with long short-term memory networks. *Proc. R. Soc. A* **474**, 20170844. (doi:10.1098/rspa.2017.0844)

14. Goodfellow I, Bengio Y, Courville A. 2016 *Deep learning*. Cambridge, MA: MIT Press.

15. Srinivasan PA, Guastoni L, Azizpour H, Schlatter P, Vinuesa R. 2019 Predictions of turbulent shear flows using deep neural networks. *Phys. Rev. Fluids* **4**, 054603. (doi:10.1103/PhysRevFluids.4.054603)

16. Jaeger H. 2007 Discovering multiscale dynamical features with hierarchical Echo State Networks. *Jacobs Univ. Tech. Rep.* **10**, 1–30.

17. Lukoševičius M, Jaeger H. 2009 Reservoir computing approaches to recurrent neural network training. *Comput. Sci. Rev.* **3**, 127–149. (doi:10.1016/j.cosrev.2009.03.005)

18. Pathak J, Hunt B, Girvan M, Lu Z, Ott E. 2018 Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. *Phys. Rev. Lett.* **120**, 24102. (doi:10.1103/PhysRevLett.120.024102)

19. Pathak J, Wikner A, Fussell R, Chandra S, Hunt BR, Girvan M, Ott E. 2018 Hybrid forecasting of chaotic processes: using machine learning in conjunction with a knowledge-based model. *Chaos* **28**, 041101. (doi:10.1063/1.5028373)

20. Doan NAK, Polifke W, Magri L. 2019 Physics-informed echo state networks for chaotic systems forecasting. In *Computational science—ICCS2019* (eds JMF Rodrigues, PJS Cardoso, J Monteiro, R Lam, VV Krzhizhanovskaya, MH Lees, JJ Dongarra, PM Sloot). Lecture Notes in Computer Science, vol. 11539, pp. 192–198. Cham, Switzerland: Springer. (doi:10.1007/978-3-030-22747-0_15)

21. Huhn F, Magri L. 2020 Learning ergodic averages in chaotic systems. In *Computational science—ICCS2020* (eds V Krzhizhanovskaya *et al.*). Lecture Notes in Computer Science, vol. 12142, pp. 124–132. Cham, Switzerland: Springer. (doi:10.1007/978-3-030-50433-5_10)

22. Wan ZY, Vlachas P, Koumoutsakos P, Sapsis TP. 2018 Data-assisted reduced-order modeling of extreme events in complex dynamical systems. *PLoS ONE* **13**, e0197704. (doi:10.1371/journal.pone.0197704)

23. Sapsis TP. 2018 New perspectives for the prediction and statistical quantification of extreme events in high-dimensional dynamical systems. *Phil. Trans. R. Soc. A* **376**, 21270133. (doi:10.1098/rsta.2017.0133)

24. Karpatne A, Atluri G, Faghmous JH, Steinbach M, Banerjee A, Ganguly A, Shekhar S, Samatova N, Kumar V. 2017 Theory-guided data science: a new paradigm for scientific discovery from data. *IEEE Trans. Knowl. Data Eng.* **29**, 2318–2331. (doi:10.1109/TKDE.2017.2720168)

25. Raissi M, Perdikaris P, Karniadakis GE. 2019 Physics-informed neural networks: a deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **378**, 686–707. (doi:10.1016/j.jcp.2018.10.045)

26. Baker N *et al.* 2019 Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence. Technical report.

27. Brunton SL, Noack BR, Koumoutsakos P. 2020 Machine learning for fluid mechanics. *Annu. Rev. Fluid Mech.* **52**, 477–508. (doi:10.1146/annurev-fluid-010719-060214)

28. Hughes TW, Williamson IA, Minkov M, Fan S. 2019 Wave physics as an analog recurrent neural network. *Sci. Adv.* **5**, eaay6946. (doi:10.1126/sciadv.aay6946)

29. Marcucci G, Pierangeli D, Conti C. 2020 Theory of neuromorphic computing by waves: machine learning by rogue waves, dispersive shocks, and solitons. *Phys. Rev. Lett.* **125**, 93901. (doi:10.1103/PhysRevLett.125.093901)

30. Goto K, Nakajima K, Notsu H. 2020 Computing with vortices: bridging fluid dynamics and its information-processing capability. (http://arxiv.org/abs/2001.08502)

31. Loiseau JC, Brunton SL. 2018 Constrained sparse Galerkin regression. *J. Fluid Mech.* **838**, 42–67. (doi:10.1017/jfm.2017.823)

32. Raissi M, Yazdani A, Karniadakis GE. 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations. *Science* **367**, 1026–1030. (doi:10.1126/science.aaw4741)

33. Cranmer M, Greydanus S, Hoyer S, Battaglia P, Spergel D, Ho S. 2020 Lagrangian neural networks. (https://arxiv.org/abs/2003.04630)

34. Bar-Sinai Y, Hoyer S, Hickey J, Brenner MP. 2019 Learning data-driven discretizations for partial differential equations. *Proc. Natl Acad. Sci. USA* **116**, 15 344–15 349. (doi:10.1073/pnas.1814058116)

35. Kochkov D, Smith JA, Alieva A, Wang Q, Brenner MP, Hoyer S. 2021 Machine learning: accelerated computational fluid dynamics. *Proc. Natl Acad. Sci. USA* **118**, e2101784118. (doi:10.1073/pnas.2101784118)

36. Moehlis J, Faisst H, Eckhardt B. 2004 A low-dimensional model for turbulent shear flows. *New J. Phys.* **6**, 56. (doi:10.1088/1367-2630/6/1/056)

37. Waleffe F. 1995 Transition in shear flows. Nonlinear normality versus non-normal linearity. *Phys. Fluids* **7**, 3060–3066. (doi:10.1063/1.868682)

38. Waleffe F. 1997 On a self-sustaining process in shear flows. *Phys. Fluids* **9**, 883–900. (doi:10.1063/1.869185)

39. Kim L, Moehlis J. 2008 Characterizing the edge of chaos for a shear flow model. *Phys. Rev. E* **78**, 036315. (doi:10.1103/PhysRevE.78.036315)

40. Joglekar M, Feudel U, Yorke JA. 2015 Geometry of the edge of chaos in a low-dimensional turbulent shear flow model. *Phys. Rev. E* **91**, 052903. (doi:10.1103/PhysRevE.91.052903)

41. Lukoševičius M. 2012 A practical guide to applying echo state networks. In *Neural networks: tricks of the trade* (eds G Montavon, GB Orr, KR Muller). Lecture Notes in Computer Science, vol. 7700, pp. 659–686. Berlin, Germany: Springer. (doi:10.1007/978-3-642-35289-8_36)

42. Byrd RH, Lu P, Nocedal J, Zhu C. 1995 A limited memory algorithm for bound constrained optimization. *J. Sci. Comput.* **16**, 1190–1208. (doi:10.1137/0916069)

43. Ouellette NT, Xu H, Bodenschatz E. 2006 A quantitative study of three-dimensional Lagrangian particle tracking algorithms. *Exp. Fluids* **40**, 301–313. (doi:10.1007/s00348-005-0068-7)

44. Doan NAK, Polifke W, Magri L. 2020 Learning hidden states in a chaotic system: a physics-informed echo state network approach. In *Computational science—ICCS2020* (eds V Krzhizhanovskaya *et al.*). Lecture Notes in Computer Science, vol. 12142, pp. 117–123. Cham, Switzerland: Springer. (doi:10.1007/978-3-030-50433-5_9)

45. Doan NAK, Polifke W, Magri L. 2020 Physics-informed echo state networks. *J. Comput. Sci.* **47**, 101237. (doi:10.1016/j.jocs.2020.101237)

46. Doan NAK, Polifke W, Magri L. 2021 Auto-encoded reservoir computing for turbulence learning. In *Computational science—ICCS2021* (eds M Paszynski, D Kranzlmüller, V Krzhizhanovskaya, JJ Dongarra, PM Sloot). Lecture Notes in Computer Science, vol. 12746, pp. 344–351. Cham, Switzerland: Springer. (doi:10.1007/978-3-030-77977-1_27)