

Sharifi Ghazijahani, Mohammad; Heyder, Florian; Schumacher, Jörg;
Cierpka, Christian





On the benefits and limitations of Echo State Networks for turbulent flow prediction

Original published in: Measurement science and technology. - Bristol : IOP Publ.. - 34 (2022),
1, art. 14002, 18 pp.
Original published: 2022-10-20
ISSN: 1361-6501
DOI: [10.1088/1361-6501/ac93a4](https://doi.org/10.1088/1361-6501/ac93a4)
[Visited: 2023-05-08]



This work is licensed under a [Creative Commons Attribution 4.0 International license](https://creativecommons.org/licenses/by/4.0/). To view a copy of this license, visit <https://creativecommons.org/licenses/by/4.0/>

On the benefits and limitations of Echo State Networks for turbulent flow prediction

Mohammad Sharifi Ghazijahani^{1,*} , Florian Heyder¹ , Jörg Schumacher^{1,2} 
and Christian Cierpka¹ 

¹ Institute of Thermodynamics and Fluid Mechanics, Technische Universität Ilmenau, Ilmenau 98684, Germany

² Tandon School of Engineering, New York University, New York, NY 11201, United States of America

E-mail: mohammad.sharifi-ghazijahani@tu-ilmenau.de

Received 8 April 2022, revised 5 September 2022

Accepted for publication 21 September 2022

Published 20 October 2022



CrossMark

Abstract

The prediction of turbulent flow by the application of machine learning (ML) algorithms to big data is a concept currently in its infancy which requires further development. It is of special importance if the aim is a prediction that is good in a statistical sense or if the vector fields should be predicted as good as possible. For this purpose, the statistical and deterministic prediction of the unsteady but periodic flow of the von Kármán Vortex Street (KVS) was examined using an Echo State Network (ESN) which is well suited for learning from time series due to its recurrent connections. The experimental data of the velocity field of the KVS were collected by Particle Image Velocimetry (PIV). Then, the data were reduced by Proper Orthogonal Decomposition (POD) and the flow was reconstructed by the first hundred most energetic modes. An ESN with 3000 neurons was optimized with respect to its three main hyperparameters to predict the time coefficients of the POD modes. For the deterministic prediction, the aim was to maximize the correct direction of the vertical velocities. The results indicate that the ESN can mimic the periodicity and the unsteadiness of the flow. It is also able to predict the sequence of the upward and downward directed velocities for longer time spans. For the statistical prediction, the similarity of the probability density functions of the vertical velocity fields between the predicted and actual flow was achieved. The leaking rate of the ESN played a key role in the transition from deterministic to statistical predictions.

Keywords: machine learning, turbulence, vortex shedding, von Kármán Vortex Street

(Some figures may appear in colour only in the online journal)

* Author to whom any correspondence should be addressed.



Original Content from this work may be used under the terms of the [Creative Commons Attribution 4.0 licence](https://creativecommons.org/licenses/by/4.0/). Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

1. Introduction

Fluid mechanics is known for its abundant data, either from numerical simulations or from experimental measurements. Machine learning (ML) methods on the other hand, are known for their exceptional capabilities to perform tasks of regression, classification and prediction, given enough training data is provided for the learning algorithm. In the past years, the application of ML models to turbulent fluid flows has gained a lot of attention, as data-driven methods are able to model flow physics without solving the underlying complex and often non-linear equations of motion [1–5]. This is why in the past years, such algorithms have gained popularity in the fluid mechanics community. Recent ML applications in this field cover turbulence modeling [6–10], indirect determination of fluid properties from flow data [11], flow control [12–14], production of super-resolved small-scale features of complex flows [15, 16], as well as turbulent flow prediction [17]. The latter, poses a challenge, as turbulent flows are inherently chaotic, so that predictions, starting from a certain flow state will quickly diverge from the true trajectory over time. Nevertheless, several attempts have been made for ML-applied turbulent flow prediction, such as prediction of shear flow [18], laminar flow over an airfoil [19], generation of turbulent channel flow [20], heated cavity flow [21], and two-dimensional turbulent Rayleigh–Bénard convection [22–25]. Still, further investigations are needed, in order to understand the capability and limitations of these algorithms when applied to flows of differing complexity. This sets the stage for the present work.

In this study, we focus on the vortex shedding behind a cylinder, known as von Kármán Vortex Street (KVS). The wake flow behind a cylinder is a phenomenon which ranges from applications in engineering to occurrences in nature. The KVS offers features of increasing complexity from laminar 2D flow to fully turbulent 3D flow with increasing Reynolds number. This makes the KVS a suitable candidate for testing the prediction performance of a ML algorithm on flows of differing stages of steadiness and complexity.

In the last years, many attempts have been made on utilizing ML methods on KVS such as flow control [26–36], shape optimization [37, 38], reduced order modeling [39–41], detection of flow features [42–47], data reconstruction [48], and super resolution [49]. Studies of KVS flow prediction cover the prediction of the velocity field by a convolutional neural network [50], trajectories of bubbles inside the vortex street using a long short-term memory recurrent neural network (RNN) [51] and the prediction of vorticity and pressure fields by means of Proper Orthogonal Decomposition (POD) and Sparse Dynamic Mode Decomposition [52]. Nevertheless, none of these studies targeted the independent prediction of the flow field in KVS for a reasonable time period without feeding the network with the information of the flow field during the prediction.

The aim of this study differs therefore from previous studies, in several points. Firstly, we run a RNN in a closed loop fashion, i.e. where the last prediction serves as basis for the

next prediction step and no additional information about the flow is given to the network. Secondly, the network architecture differs from those of previous studies as we apply a reservoir computing algorithm, called Echo State Network (ESN). ESNs are RNNs which include a reservoir of neurons with sparse recurrent connections [53, 54] where merely the output layer is trained. The ESN possesses an internal memory of past inputs [55] which makes them suitable for application to prediction of periodic turbulent flows. In recent years, ESNs have been used to predict and control extreme events in a chaotic shear flow [56], estimate the life time of fuel cells [57], predict spatiotemporal chaos [58], infer missing variables of the Lorenz '63 system [59], as well as to model two dimensional turbulent convection [22, 23, 60]. While these results demonstrate the potential of this network architecture for the purpose of predicting nonlinear tasks, it is unclear how such a network performs on experimental measurements of unsteady periodic flows like the KVS.

This study aims to investigate the predictive potential of an ESN for KVS as an archetype for typical wake flows. We obtain experimental data of the KVS velocity fields at $Re = 100$ and 1000 by means of Particle Image Velocimetry (PIV). The data from the flow measured at the two Reynolds numbers allows us to evaluate the quality of the predictions of unsteady and steady vortex shedding respectively. We distinguish between deterministic and statistical prediction of the flow, and highlight the challenges of defining suitable criteria for the evaluation of predictions and the tuning of the hyperparameter. As the ESN is not capable of processing all data points of the velocity fields, we introduce a preprocessing step to reduce the amount of input data. We perform POD and optimize an ESN with 3000 neurons to obtain the best deterministic and statistical prediction of the first 100 Time Coefficients of the POD Modes (TCPM). The complexity of the TCPMs increases with the mode number, so that we are exposing the ESN to an input with a wide range of time scales. We find that an ESN can both predict the time trajectories (named deterministic prediction) and probability density function (PDF) (named statistical prediction) of the flow, when it is optimized accordingly.

The outline of this manuscript is as follows. First the experimental and numerical methods of data generation, processing and prediction are discussed in section 2. Section 3 presents the results of the closed-loop ESN prediction of the KVS and discusses the hyperparameter optimization of the ML method. Finally, we summarize our results and give a brief outlook in section 4.

2. Methods

In this section, first the experimental method for conducting experiments and collecting flow data is explained. This is followed by an explanation of how to reduce data for ML and a discussion of the results of the data reduction. Finally, the algorithm and method for applying ML to the reduced data of the flow are discussed.

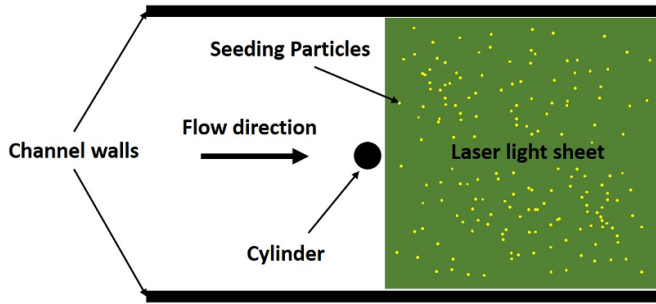


Figure 1. Schematic sketch of the experimental setup for the PIV measurements.

2.1. Experimental setup

A schematic sketch of the experimental setup for the PIV measurements can be seen in figure 1. The cylinder was placed inside a water channel with a cross-section of $50 \times 50 \text{ mm}^2$. To increase the Reynolds number range two different diameters, namely 8 mm (C8) and 1 mm (C1) for the cylinder were used. The flow was seeded by Polyamide particles of $20 \mu\text{m}$ and $5 \mu\text{m}$ in diameter for C8 and C1, respectively. A continuous-wave laser (Laserworld Green-200 532) illuminated the vertical mid-plane of the channel. A light sheet was formed by optical elements and had a thickness of 1 mm. PIV images of the illuminated particles were then captured using a high-speed camera (HS 4M by LaVision GmbH) perpendicular to the laser sheet outside the channel. For C8, the images were calibrated with respect to the channel walls at the top and bottom of the images, and for C1, due to the proximity of the camera to the flow and the lack of wall boundaries in the images, they were calibrated against a calibration target. The target was two-dimensional and consisted of white dots in a black background. The Reynolds number (Re) is calculated based on the diameter of the cylinders (D) as $Re = V_\infty D / \nu$. Here, V_∞ stands for free stream velocity (133 mm s^{-1}) and ν is the kinematic viscosity. This results in $Re = 1000$ and 100 for C8 and C1, respectively. In $Re = 1000$, the flow is 3D, whereas in $Re = 100$, the flow is 2D. This results in unsteady and steady vortex shedding for $Re = 1000$ and 100 , respectively. The data were collected at recording frequencies (f) of 50 Hz for C8, and 400 Hz for C1, with measurement durations of 100 and 12 s , respectively. The oscillatory behavior of the KVS is usually described by Strouhal number (St) which is often calculated as $St = fD / V_\infty$, where f is the frequency of the vortex shedding. The typical St for the KVS in $Re = 100$ and 1000 are around 0.16 and 0.22 , respectively. This results in 280 and 330 vortex shedding events with a temporal resolution, $t/t_{\text{characteristic}}$, of 18 or 15 time steps per vortex shedding event, respectively. For the current study the PIV analysis was done via DaVis Software provided by LaVision GmbH. Detailed information about recent PIV algorithms including the results of the currently used algorithm for benchmarks (among the best) can be found in [61]. An advanced cross-correlation evaluation was applied for PIV processing for an initial rectangular interrogation window size of 64×64 pixels

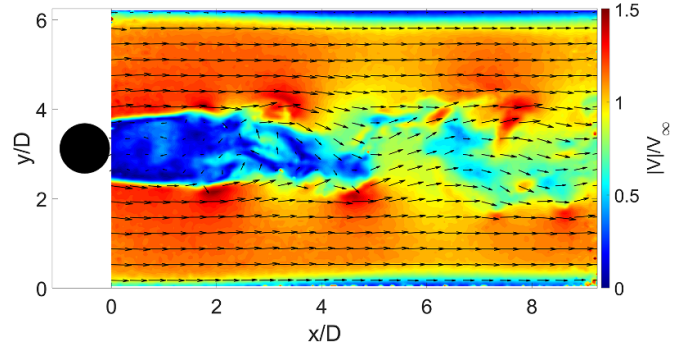


Figure 2. Instantaneous sample vector field of PIV measurement for $Re = 1000$. Color indicates velocity magnitude. Every 8th vector shown.

for both the C8 and C1 with 50% overlap, and a final circular Gaussian window weighting of 16×16 pixels for C8 and 32×32 pixels for C1. The number of passes of the initial and final interrogation windows were one and three for C8, and two and three for C1, respectively. For C8, the final spatial resolution was $0.35 \times 0.35 \text{ mm}^2$ in physical coordinates with 141×211 vectors, and for C1, $0.13 \times 0.13 \text{ mm}^2$ in physical coordinates with 56×89 vectors. For both $Re = 100$ and 1000 , in a normalized median test, the number of outlier vectors amount to below one percent [62]. For carefully adjusted experiments, the absolute error can be estimated to be around 0.1 pixels of displacements. This results in minimum relative uncertainties of 1.3% and 1.5% for $Re = 100$ and 1000 in the free stream. However the values will be higher for the areas with lower displacements in the flow. Figure 2 shows a sample vector field of the measured data.

2.2. POD

In order to reduce the amount of data, which has to be fed to the reservoir, and to have better understanding of the flow features, snapshot POD [63] was applied to the data. The use of POD-based reduced order modeling of complex geometry flows has been widespread for decades [64]. Different data reduction methods have been adapted to suit various aspects of the flow. For example, balanced POD analysis captures the most controllable modes of the flow, and Dynamic Mode Decomposition extracts the growth rates and frequencies of dynamic modes from flow field data [65]. For this study, we chose POD due to its widespread implementation in the past for wakeflows [66] and the fact that it does not add any additional information to the ESN about the modes other than the fact that it captures the maximum kinetic energy. As a result, the discovery of the relationships between the different POD modes will rely solely on ML.

Figure 3 shows the share of each POD mode in the total kinetic energy. Two main modes with distinct higher energies are present for both Reynolds number, with higher values for $Re = 100$. This is an indication of more steady vortex shedding for this Re . Figures 4, 5, 6(a), and 7(a) show the vertical velocity fields and time coefficients of the first four

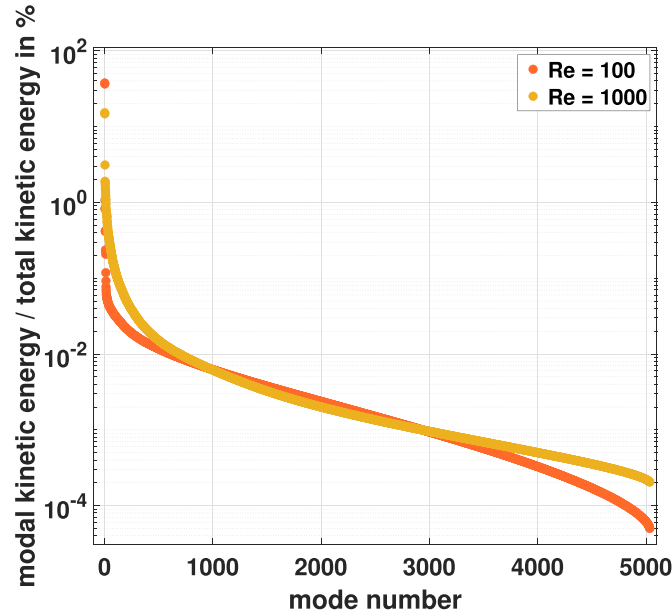


Figure 3. Energy of POD modes.

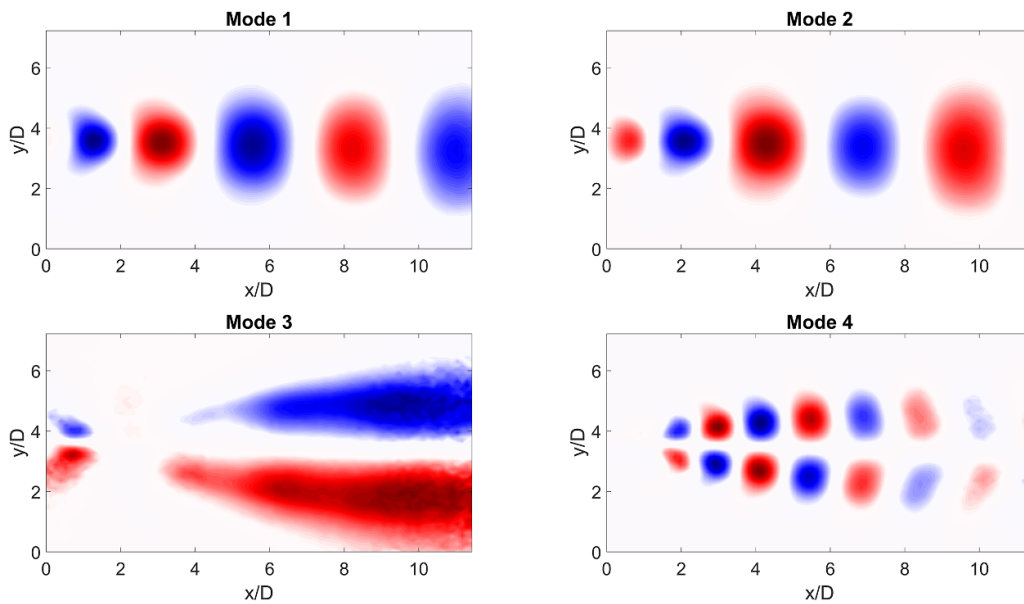


Figure 4. Spatial distribution of modes 1, 2, 3, and 4 for $Re = 100$.

modes for $Re = 100$ and 1000 respectively, where T is the time period for one vortex shedding event. The first two modes have a sinusoidal pattern with 90° phase shift from each other, which is an indication of the vortex shedding in the flow. However, for $Re = 1000$, mode three and four show more complex time coefficient distributions. The results of the POD analysis for the current study are in line with similar available data in the literature (see Feng *et al* [67] for similar POD results for $Re = 950$). Figures 6(b) and 7(b) show the time coefficient of the first mode for a longer period, where $Re = 100$ has fixed amplitude for this mode, whereas the variation in the amplitude of this mode is evident for $Re = 1000$. For the current

ML application, only the first hundred modes were considered. For the cumulative energy percentage of the POD modes the elbow method cuts the flow for first 228 and 404 modes in $Re = 100$ and 1000 respectively, which is way above first hundred selected modes for this study. As a result, it can be argued that mode number hundred in this flow is most likely not part of the noise. Additionally, in order to assess whether the ML algorithm is capable of deducing a meaning from the oscillations of a relatively large number of intrinsically connected modes, it is necessary to feed it with such a high number of POD modes. This is in line with other studies in literature which examined similar methods [23]. In addition the ESN

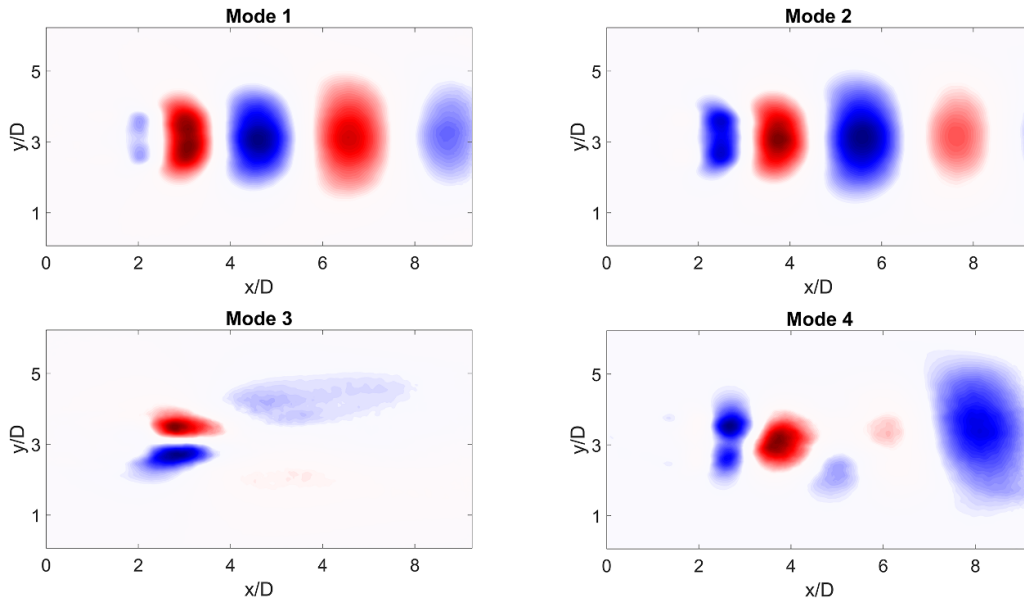
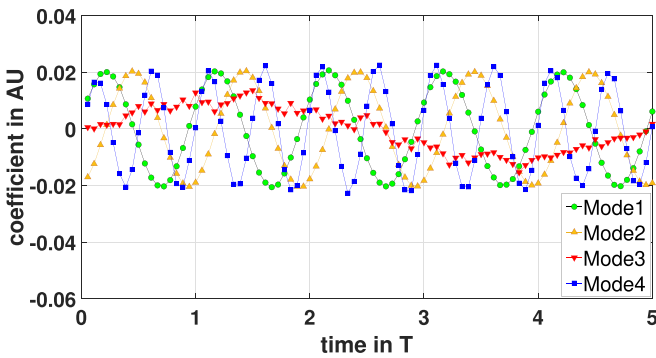
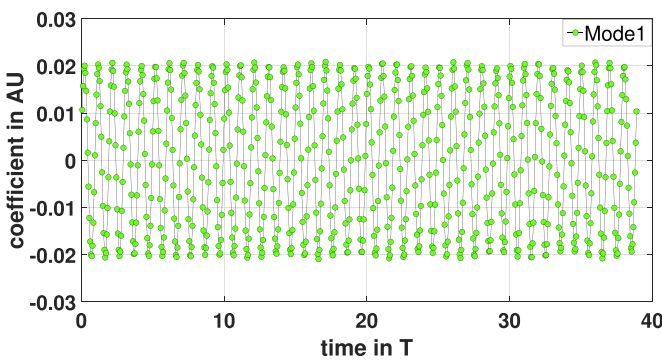


Figure 5. Spatial distribution of modes 1, 2, 3, and 4 for $Re = 1000$.

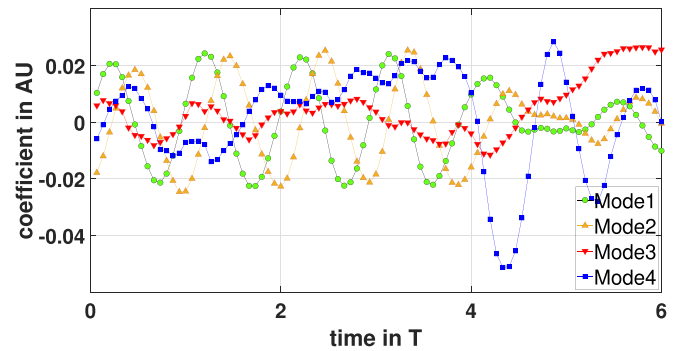


(a) Temporal coefficient values of modes 1, 2, 3, and 4.

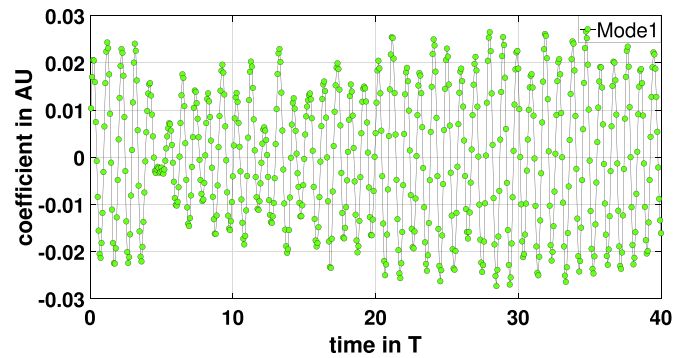


(b) Temporal coefficient values of mode one.

Figure 6. Temporal coefficient values of modes for $Re = 100$.



(a) Temporal coefficient values of modes 1, 2, 3, and 4.



(b) Temporal coefficient values of mode one.

Figure 7. Temporal coefficient values of modes for $Re = 1000$.

(number of neurons) is built to work with a certain size for input within a reasonable time. In our case it works well with 100 TCPM as input and in practice a user would then always use 100 input modes independent of the Reynolds number. In our case 100 modes correspond to 82% and 73% of total kinetic energy for $Re = 100$ and $Re = 1000$, respectively.

2.3. ESN

In this study, an ESN was used to predict the TCPM of the KVS. It is a RNN where the output weights are trained only. It was proposed by Jaeger [53] as an alternative to the gradient descent training scheme of RNNs which suffer from the

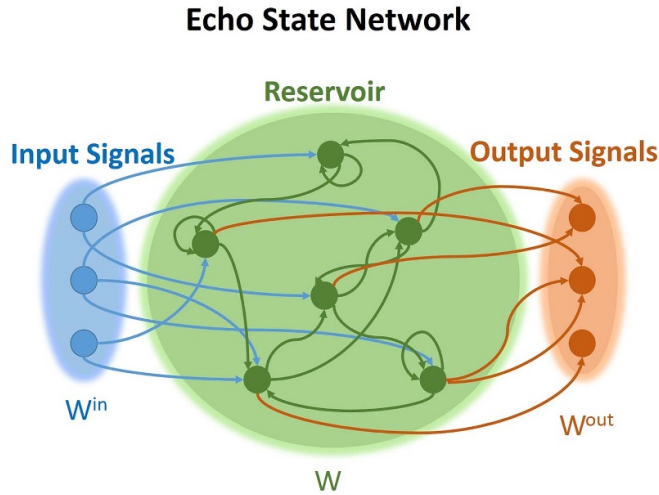


Figure 8. A schematic sketch of an Echo State Network (ESN).

exploding and vanishing gradient problems [68]. Independent of Jaeger, a similar approach, called liquid state machine, was developed by Maas *et al* [69]. Both are now summarized by the name reservoir computing which is a rapidly growing field [54]. A schematic of an ESN is illustrated in figure 8. An ESN is simply a reservoir of N neurons with the reservoir state (s) that receives N_{in} input signals (u) and produces N_{out} output signals (q), where u contains the predicted TCPMs from the previous time step and q their predictions. For our task $N_{in} = N_{out}$. There are three layers of connections within an ESN. First, input signals are connected to different reservoir neurons with different weights. $W^{in} \in \mathbb{R}^{N \times N_{in}}$ is the weight matrix for this connection, and it is randomly preselected before running the ESN. The weight values W_{ij}^{in} of this matrix are chosen according to $W_{ij}^{in} \sim \mathcal{U}[-INS/2, INS/2]$. The Input Scaling (INS) is the main variable for the control of the input weights in W^{in} , and it can regulate the amount of the nonlinearity of the reservoir plus the relative weight of current input against the history. INS can be set individually or uniformly across all input signals. In addition, the neurons of the reservoir are connected by random weights $W_{ij} \sim \mathcal{U}[-0.5, 0.5]$. These weights are components of the reservoir matrix $W \in \mathbb{R}^{N \times N}$. Despite W^{in} and W being preselected randomly, the random state of them can be fixed by a variable named Random Seed (RS) in order to be able to recreate the same ESN later.

An important control parameter is the maximum eigenvalue of W , referred to as the spectral radius (SR). It controls the contribution of the internal reservoir interactions to the nonlinear update rule of the reservoir. High values of SR can lead to more chaotic reservoir dynamics, while a small SR increases the influence of the external reservoir input u on the output q , where u represents the predicted TCPMs in the previous time step and q is their predicted values for the current time step. However, SR should stay below unity to ensure the *echo state property*. The *echo state property* is a necessary condition for an effective reservoir and states that a reservoir becomes independent of its past states and is therefore uniquely defined by the last inputs [53, 70]. Until now, no sufficient condition for an effective reservoir has been found, see Yildiz *et al* [71] for a

discussion. A further hyperparameter is the sparsity (or density D) of W which controls how many connections each neuron possesses. As shown in equation (1), each neuron updates $\tilde{s}(n)$ based on the signals it receives from inputs and other neurons.

$$\tilde{s}(n) = \tanh(W^{in}[1; u(n)] + Ws(n-1)) \quad (1)$$

$$s(n) = (1 - LR)s(n-1) + LR\tilde{s}(n) \quad (2)$$

$$q(n) = W^{out}[1; u(n); s(n)] \quad (3)$$

$$W^{out} = \operatorname{argmin} \frac{1}{N_{out}} \sum_{i=1}^{N_{out}} \left[\sum_{n=1}^T (q_i(n) - q_i(n)^{\text{target}})^2 + \beta \|w_i^{\text{out}}\|^2 \right] \quad (4)$$

Following this, the current state of the neurons $s(n)$ is calculated by combining the state of the neurons at the current time step with their state at the previous time step based on the leaking rate (LR), as shown in equation (2). LR is a measure of how fast the reservoir is updated. Consequently, the optimal LR is heavily influenced by the flow dynamics.

In the end, the output signals are compiled from neurons by a weight matrix called W^{out} , which is calculated during training as shown in equations (3) and (4). Here $\beta > 0$ is the ridge regression parameter which prevents the amplification of small differences in state dimensions by large rows of W^{out} . It also counters the effect of overfitting, where the algorithm learns the training data by heart. For this study, the ESN model was created in *Python* using the library *easyesn* [72] and *turbESN* [73] for $N = 3000$ neurons, where each neuron is sparsely connected to 20 percent of the others, i.e. $D = 0.2$. Then it is trained and tested for $T = 700$ time steps with different sets of hyperparameters to reach the maximum efficiency.

3. Results and discussion

3.1. Deterministic prediction

This section aims to evaluate the performance of the ESN for a deterministic prediction of the flow in each time step with the least amount of error possible. Therefore, the aim is to determine the actual flow field as measured in the testing phase by the prediction via ESN. In order to optimize the ESN for that purpose, a parameter for prediction evaluation is required to rank the predictions based on their proximity to the corresponding flow field. This depends on the flow feature that is preferred to be predicted, such as vertical velocity, vorticity, velocity magnitude, etc. The ideal case would be to consider the proximity of all flow features in the prediction. However, this is a quite complex task to accomplish. Since the entire flow information is preserved in TCPM and hence their correct estimation will result in the preservation of all flow features in the predictions, one can consider the absolute error $\left(\sum_{i=1}^{100} \sum_{t=1}^{700} |q_i(t)^{\text{prediction}} - q_i(t)^{\text{target}}| \right)$ of the time coefficients as a reasonable parameter for evaluating these predictions. Figure 9 shows the prediction set with the minimum

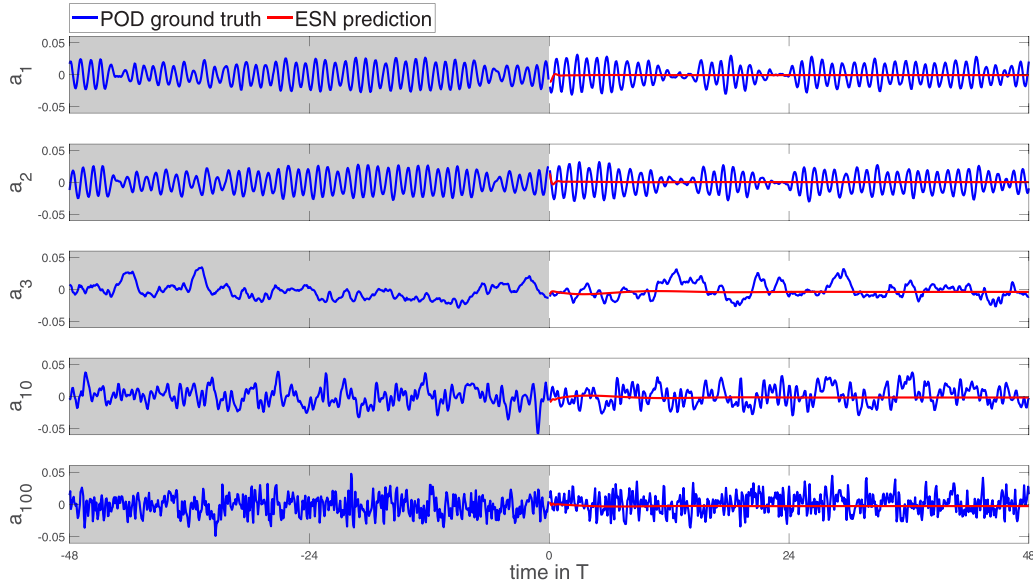


Figure 9. Actual and predicted values of the temporal coefficients of modes 1, 2, 3, 10, and 100 for $Re = 1000$. The lines marked with ESN indicate the predicted values of the network with the optimized set of hyperparameters with lowest absolute error with respect to actual values, while the lines marked with POD show the actual values of the temporal coefficients during training and testing.

absolute error of the POD time coefficients compared to reality. Here the outputs of the ESN are just constant lines over zero. It is clear that the prediction is incapable of reproducing any oscillations of time coefficients. This is quite foreseeable. Due to time shift of oscillatory predictions with reality, which will result in even larger absolute errors, the constant lines will have minimum absolute error compared to other predictions. Therefore, another parameter for prediction evaluation is needed.

The vertical velocity fluctuation is the primary footprint of the KVS. After all, the wake flow behind a cylinder is a series of regions with upward and downward flow and the momentum transfer is due to V_y inside the vortices. Accordingly, the aim can be the correct prediction of negative or positive directions of the vertical velocity field in $Re = 1000$ for $|V_y| > 17.6 \text{ mm s}^{-1}$. The value of 17.6 mm s^{-1} , which corresponds to $V_y/V_\infty = 0.132$, was arbitrarily chosen and corresponds to one pixel displacement in the PIV images. It is defined as a threshold to only predict the flow inside the wake region with the shedding vortices and neglect the free stream. This parameter is named Vertical Velocity Prediction of Direction (VVPD) and shows the percentage of $|V_y| > 17.6 \text{ mm s}^{-1}$ which have correct direction in terms of their positive or negative values in the predictions. It should be stressed, that this parameter does not deal with the prediction of vertical velocity magnitudes. Figure 10 shows a schematic of the division of the flow into three regions based on the vertical velocity values, where the white area is the part of the flow which is the free stream outside the regions of higher vertical momentum transfer, and blue and red regions represent the positive and negative $|V_y|$ that are aimed to be predicted. From figure 10, it is obvious that the pattern of the up and down flow is quite complex, as well as somewhat unpredictable due to the unsteadiness of the vortex shedding. To some degree

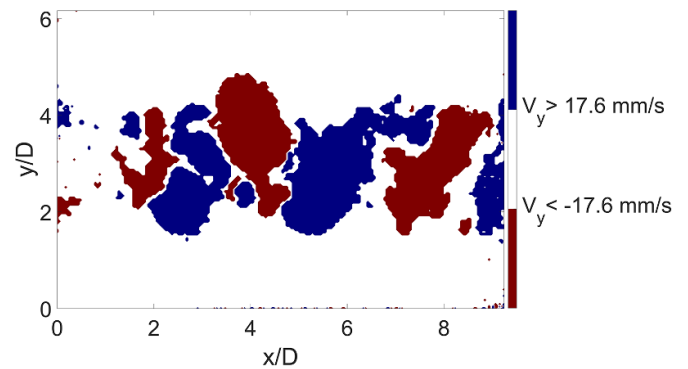


Figure 10. Vertical velocity field behind the cylinder. Blue and red stand for up- and downward directed flows respectively, and white regions are for $|V_y| < 17.6 \text{ mm s}^{-1}$ in the free-stream which corresponds to less than one pixel displacement in the PIV images.

this is due to the experimental nature of the data, which has a larger error than its numerical counterparts. Mainly it is due to the turbulent nature of the flow, i.e. unsteady vortex shedding in the case of $Re = 1000$, which shows up in the TCPMs of the first two modes.

Although the aim of this study is to predict the unsteady vortex shedding in the wake flow of a cylinder in $Re = 1000$, however, it is of particular interest to first see the predictions in $Re = 100$ where vortex shedding is fairly stable. In figure 11, the performance of the ESN is optimized for the main three hyperparameters for $Re = 100$ according to the VVPD values. It should be noted that for $Re = 100$, 14.8 mm s^{-1} corresponds to one pixel displacement in the PIV images and $V_y/V_\infty = 0.11$, and thus it is considered as the minimum threshold for calculating VVPD. Table 1 shows the values that were used in the grid search for hyperparameters. To the left of figure 11, SR is fixed at 0.95 and LR and INS are varied,

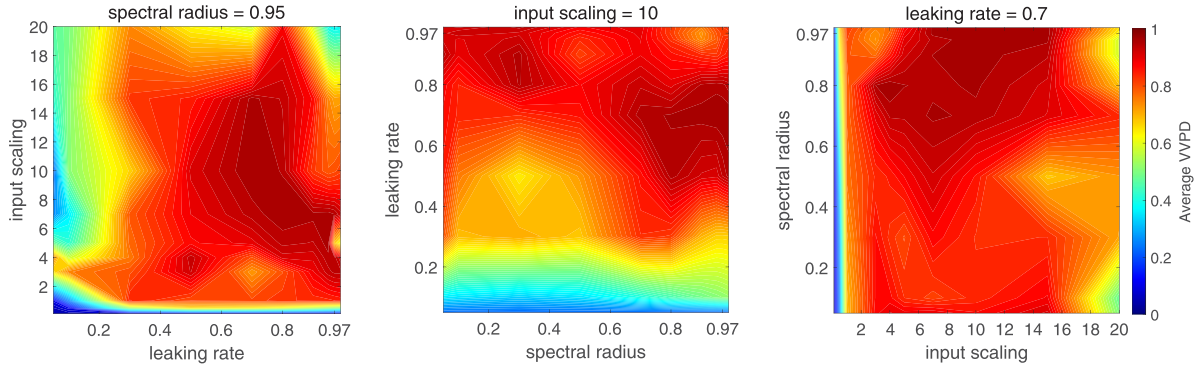


Figure 11. Vertical velocities prediction of direction (VVPD) values with respect to variations in input scaling and leaking rate (left), leaking rate and spectral radius (middle), and spectral radius and input scaling (right) for $Re = 100$.

Table 1. The range of the three main hyperparameters in which the optimizations were conducted.

INS	0.1	0.5	1	3	4	5	7	10	15	20
LR	0.05	0.1	0.3	0.5	0.7	0.8	0.9	0.95	0.97	0.99
SR	0.05	0.1	0.3	0.5	0.7	0.8	0.9	0.95	0.97	0.99

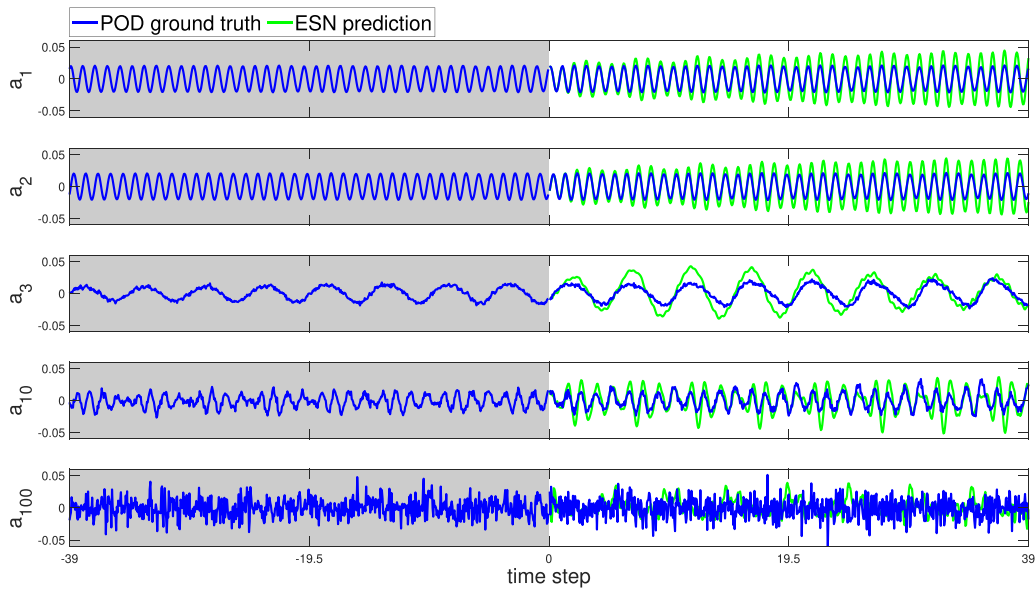


Figure 12. Actual and predicted values of the temporal coefficients of modes 1, 2, 3, 10, and 100 for $Re = 100$. The lines marked with ESN indicate the predicted values of the network with the optimized set of hyperparameters for the deterministic prediction, while the lines marked with POD show the actual values of the temporal coefficients during training and testing.

where $INS = 10$ and $LR = 0.7$ represent the highest VVPD value. Next, INS is fixed to 10 at the middle, and LR and SR are varied. The maximum VVPD is obtained when $LR = 0.7$ and $SR = 0.97$. Finally, at the left, LR is set to 0.7 and SR and INS are changed where $SR = 0.8$ and $INS = 4$ is the best choice. This resulted in an optimized ESN with $INS = 4$, $LR = 0.7$, and $SR = 0.8$. Here, it is also essential to take into account the more general behavior of the ESN with respect to its hyperparameters. Generally, the ESN performs similarly for a wide range of all three hyperparameters. The only exception is extremely low LR values, notably below 0.3, which always significantly reduce the VVPD of the predictions.

Figure 12 shows the TCPMs of the optimized case for $Re = 100$ for modes 1, 2, 3, 10, and 100. The blue lines stand for the ground truth (POD) and the green lines for the optimized ESN. First, from time step -700 to 0 the ESN is trained, and then the prediction phase starts for the next 700 time steps. The first two modes are predicted accurately in terms of phase. However, in the later time steps there are some enlargements of the amplitudes. This indicates that sometimes the errors are not related to predicting the oscillations in wrong time steps, instead they are simply the natural divergence of the ESN itself. However, the ESN is capable of keeping them under control and returning to an imitation of the oscillation. The

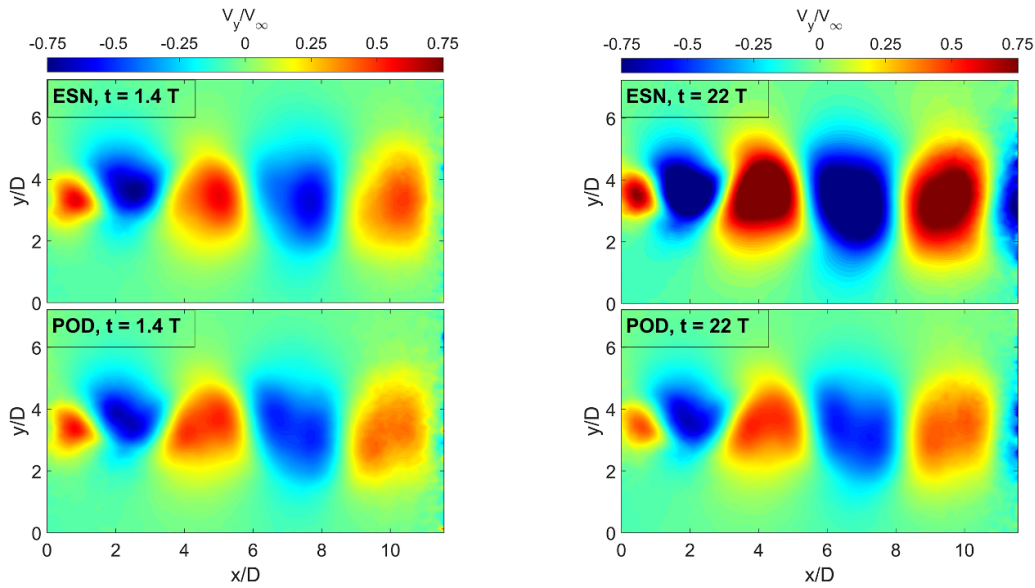


Figure 13. The vertical velocity fields of optimized deterministic prediction and reality for $t = 25$ (left) and 400 (right) for $Re = 100$.

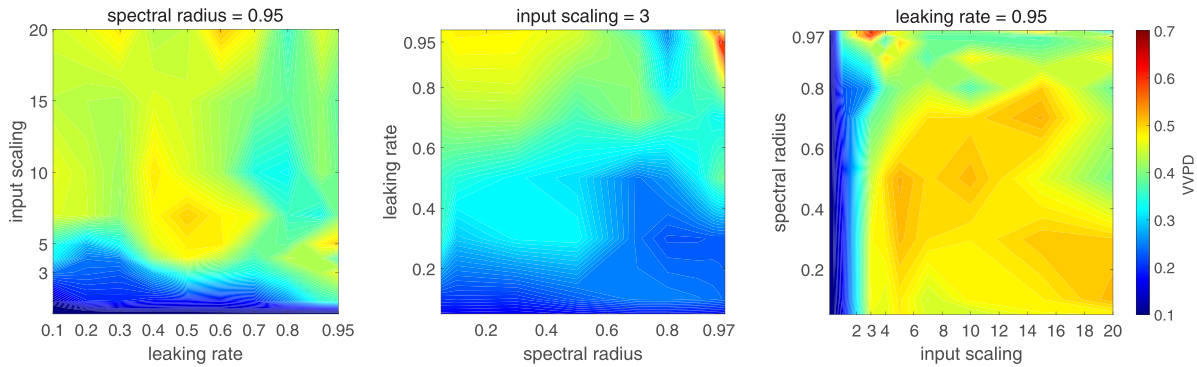


Figure 14. Vertical velocities prediction of direction (VVPD) values with respect to variations in input scaling and leaking rate (left), leaking rate and spectral radius (middle), and spectral radius and input scaling (right) for $Re = 1000$.

time coefficients of next modes are also predicted quite well, but they have more divergence due to their higher complexity. Figure 13 shows the vertical velocity fields predicted by ESN and the real ones for $t = 25$ and 400 . In both time steps, the location and shape of the upward and downward flows are predicted accurately. However, for the later time step the magnitudes are exaggerated.

Figure 14 shows the optimization of the ESN with respect to VVPD values for the main three hyperparameters for $Re = 1000$. Again, the values of the hyperparameters used in the grid search are shown in table 1. At the left of figure 14 SR is fixed to 0.95 and LR and INS are varied, where $INS = 3$ and $LR = 0.95$ represent the highest VVPD value. Therefore, next the INS is fixed to 3 for the mid panel, and LR and SR are varied. Here, $LR = 0.95$ and $SR = 0.97$ has the maximum VVPD. Finally, in the right panel of the figure, LR is set to 0.95 and SR and INS are changed where $SR = 0.97$ and $INS = 3$ is the best choice. Therefore, the optimized ESN had $INS = 3$, $LR = 0.95$, and $SR = 0.97$.

Figure 15 shows the temporal evolution of the TCPMs for $Re = 1000$ in the real flow (blue lines) and in the prediction

(green lines) of the optimized case for modes 1, 2, 3, 10, and 100. For the first two modes, the ESN is capable of learning the vortex shedding frequency and thus accurately predicts the phase of the oscillations. However, the amplitudes of oscillations diverge from the real flow after almost 150 time steps. Apparently, the significant variation of the amplitude in the real flow due to the unsteady vortex shedding is something that the ESN fails to predict or at least imitate. However, the predictions have a limit in their amplitude and they continue to oscillate without growing infinitely. For the next modes, the predictions are quite close to the general behavior of the real values, however, the exact prediction of the values are out of reach for the ESN. Therefore, the ESN has accomplished the task of mimicking the oscillations of the time coefficients of the modes in general. In order to better assess the quality of the predictions, figure 16 compares the vertical velocity fields of the predictions by ESN and reality at time step 25 and 400. In both time steps, the succession of the upward and downward flows are predicted acceptably. Also, in $t = 25$ the V_y magnitudes are well predicted. However, for $t = 400$, they are quite exaggerated by the ESN. Generally, the predictions

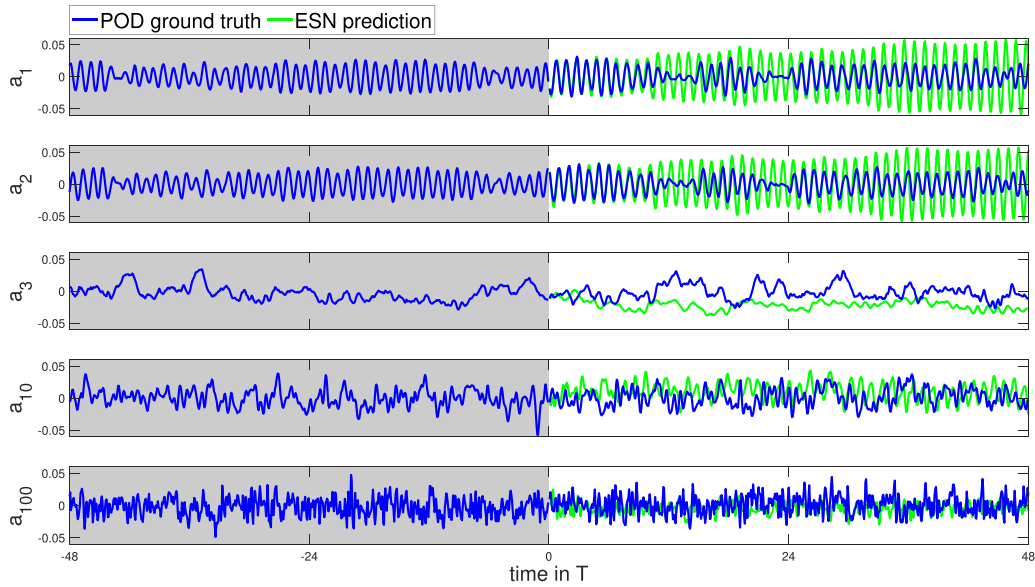


Figure 15. Actual and predicted values of the temporal coefficients of modes 1, 2, 3, 10, and 100 for $Re = 1000$. The lines marked with ESN indicate the predicted values of the network with the optimized set of hyperparameters for the deterministic prediction, while the lines marked with POD show the actual values of the temporal coefficients during training and testing.

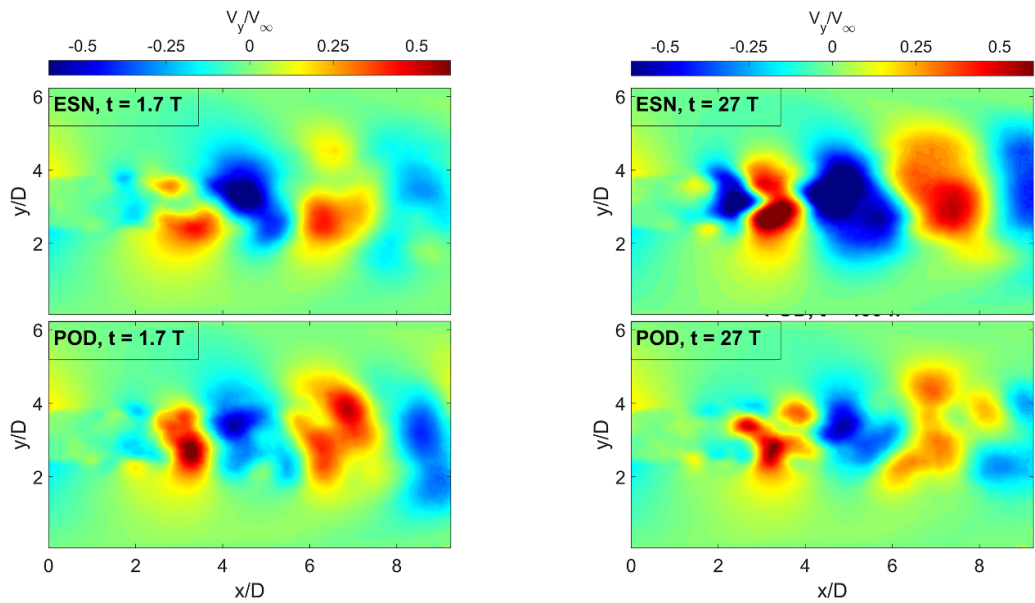


Figure 16. The vertical velocity fields of optimized deterministic prediction and reality for $t = 25$ (left) and 400 (right) for $Re = 1000$.

match reality well at the start, but accumulating a number of minute errors over a period of time causes the flow to become exaggerated in the later time steps after around ten vortex shedding events.

Although the ESN is optimized for the three main hyperparameters, however, the robustness of the results should also be evaluated with respect to the different random states of the network. RS is the variable that fixes the randomly preselected W^{in} and W , and by its variation new random values will be assigned. Since the deterministic prediction for $Re = 1000$ was comparatively more challenging, thus the effect of RS variation will be discussed for this case only. Figure 17 shows the degree in which the VVPD value of the

optimized deterministic prediction for $Re = 1000$ ($INS = 3$, $LR = 0.95$, and $SR = 0.97$) depends on RS. $RS = 0$ refers to the RS used during optimization. Therefore, it is predictable that it should have the highest VVPD. For the next seeds VVPD drops significantly and after around 20 RSs the cumulative average VVPD value approaches to 0.35. This is a clear indication of the RS effect in the performance of the ESN. Although one can argue that the dependence on RS should decrease with the increase of the reservoir size, the dependence on RS is inevitable for the current reservoir with 3000 neurons.

Therefore, the ESN is optimized for hundred different RS values and their respective maximum VVPD and

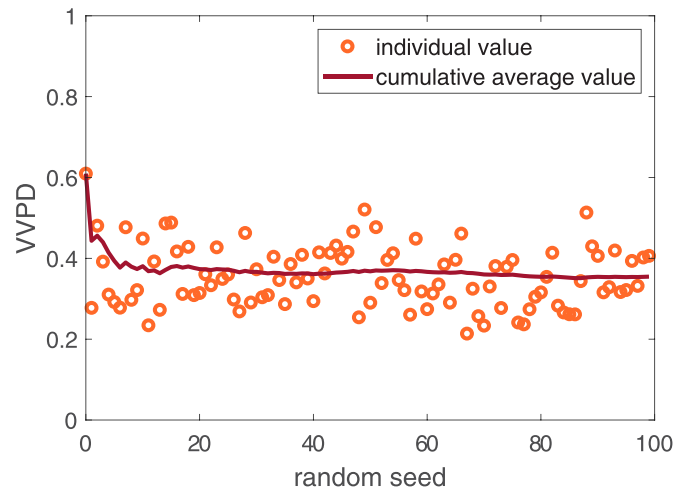


Figure 17. The variation of VVPD values using the optimized hyperparameters (INS = 3, LR = 0.95, and SR = 0.97) for random seed RS = 0 versus different random initializations (RS > 0) for Re = 1000.

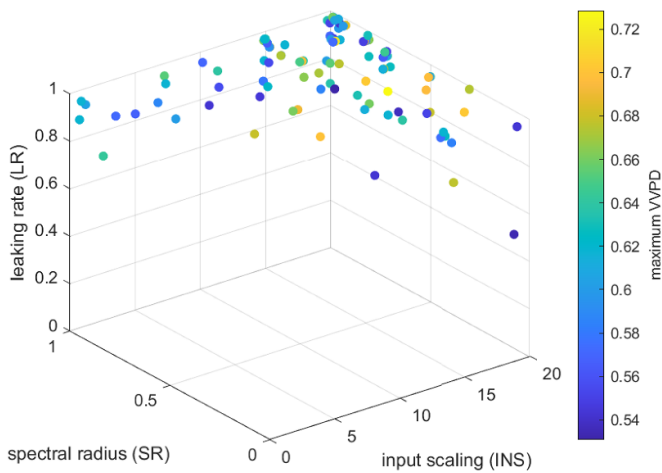


Figure 18. The optimized hyperparameters for maximum VVPD values for one hundred different random seeds for Re = 1000.

hyperparameter values are shown in figure 18. The maximum VVPD values are between 0.53 and 0.73, with 0.62 being their average. This indicates that the optimized set for each RS will have a comparable performance in terms of VVPD. For all RSs the optimized sets have large LR values close to one, and there is no single optimal set with $LR < 0.5$. SR is also predominantly high and close to one, although there are still some cases with a very low SR that have very high INS values. Finally, most of the cases have large INS as well, and the domain of LR and SR increases for high INS values. In general, sixty of the hundred different RSs have SR and LR ≥ 0.8 . Therefore, the initial conclusion about INS = 3, LR = 0.95, and SR = 0.97 being the optimized set of hyperparameters for deterministic prediction of the flow for Re = 1000 is generally valid. However, the INS value will be highly dependent in the RS.

Overall, the deterministic predictions of the flow show the potential of the ESN for the prediction of the KVS. However, as Reynolds number increases and flow becomes unsteady, predictions deviate increasingly from the ground truth. Rather

than predicting the oscillations of the TCPMs exactly, an ESN is by far more capable to reproduce their evolution qualitatively. Therefore, a clear boundary between prediction and imitation should be drawn in our understanding of ML application for flow prediction. While prediction results in exact overlap of prediction and ground truth, imitation results in general agreement of prediction and ground truth in terms of statistics, which is the topic of the next section.

3.2. Statistical prediction

The second objective of this study is to predict the statistics of the unsteady flow at Re = 1000 velocity, which means that the PDF of the predicted velocity should overlap the one of the real flow (or ground truth). This ability is important in the case of a coarse flow simulation and a subgrid modeling of the statistical properties of the turbulent flow. The goal is to produce a flow that is indistinguishable from the actual flow, regardless of how closely the individual time steps between the prediction and reality align. As figure 9 showed, the absolute error of TCPMs cannot be used as a prediction evaluation parameter. VVPD is not a suitable candidate due to its exaggerated velocity magnitudes, particularly at further time steps, once the ESN is optimized based on it. Therefore, a new prediction evaluation parameter needs to be defined. If the vertical velocity fields are regarded as images, then the Structural SIMilarity index (SSIM) [74] between the reconstructed velocity fields of the predictions and real flows can be defined as a prediction evaluation parameter. Then, the overall proximity of the prediction set and the real flow will be analyzed by calculating the maximum SSIM between the prediction set and the real flow through the entire 700 time steps. To do so, a matrix of SSIM values is created between the entire prediction and actual flow fields. The maximum SSIM value in the matrix is recorded, and its respective prediction and actual time steps are deleted. The remaining part of the matrix is then searched for the next maximum SSIM. In order to do so, this procedure is repeated until 700 prediction time steps

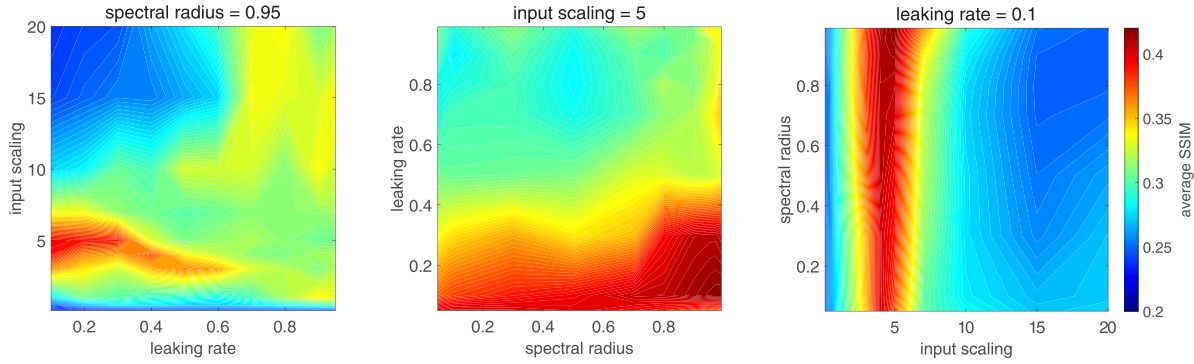


Figure 19. Average SSIM (structural similarity index) values with respect to variations in input scaling and leaking rate (left), leaking rate and spectral radius (middle), and spectral radius and input scaling (right) for $Re = 1000$.

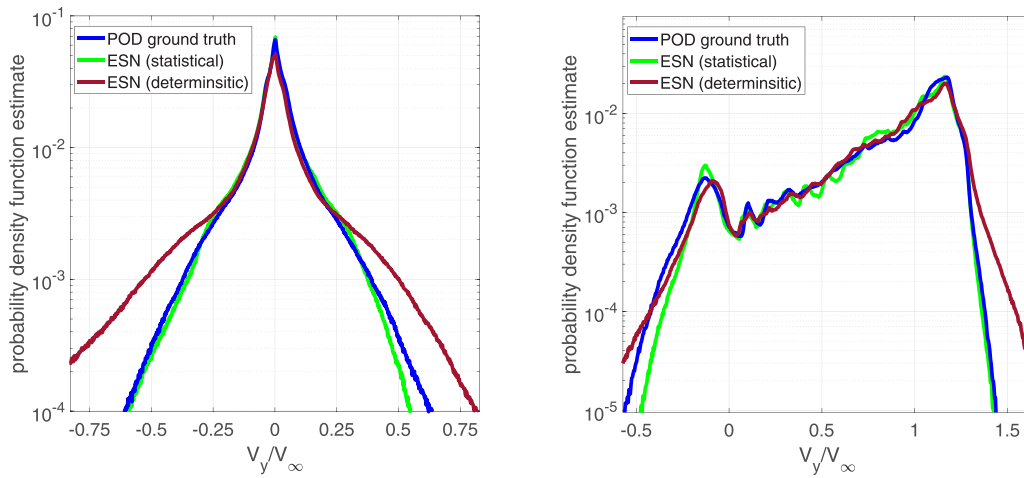


Figure 20. Probability density function of the actual flow and the optimized statistical prediction of it for $Re = 1000$.

have been matched with their corresponding actual flow fields, irrespective of their temporal occurrence. Then, the average of the maximum SSIMs is reported. Figure 19 shows the result of optimizing the ESN according to the SSIM. Table 1 presents the hyperparameter values once again. The chart on the left shows that when $SR = 0.95$, the $INS = 5$ and its surroundings have markedly higher SSIM values. In the middle of the figure, $LR = 0.1$ shows the maximum SSIM value for $INS = 5$. Finally, at $LR = 0.1$, $SR = 0.99$ is representing the highest average maximum SSIM, however the SSIM is quite insensitive to the SR values. The optimized set of hyperparameter according to SSIM, for the current setup is $INS = 5$, $LR = 0.1$, $SR = 0.99$.

Figure 20 shows the PDFs of the V_y and V_x of the optimal statistical (green lines) and deterministic (red lines) predictions of ESN in comparison to POD or the actual flow (blue lines) in $Re = 1000$. For plotting the PDFs, the entire 700 time steps of the prediction phase are considered. Both statistically optimized and actual PDFs overlap very well for V_y and V_x , however, at the far tails of the positive V_y and negative V_x range, there are some differences. Meanwhile, the PDF of deterministically optimized prediction diverges from actual flow for both positive and negative ranges of V_y and V_x . In figure 21, the TCPMs of the actual flow and the statistically

optimized prediction are shown. There is no variation in the amplitude of the predictions of the first two modes. This is unlike the optimized deterministic prediction, which aims to predict the amplitude variation of these modes as well (see figure 15). The reason for this is that the statistical prediction of a flow needs to reflect the average variation of the flow, not its exact perturbations. Accordingly, predictions of the other modes with lower energies have minimal oscillations when compared to their actual variations. Figure 22 shows the V_y fields of the predictions and corresponding real flow fields. In addition to the sequence of positive and negative V_y , the magnitudes of the velocities are also within the range of the actual flow. Nonetheless, in terms of their shape and pattern, the predicted fields do not correspond well to the actual flow in the further time steps ahead. It can be concluded that the ESN is perfectly able to reconstruct a prediction with similar velocity statistics to the actual flow by mimicking the two main modes and trying not to overestimate their amplitudes in the actual flow. However, the deterministic prediction exhibits more realistic oscillations of the modes when compared. Also, it is interesting to note that both deterministic and statistical predictions have comparable INS and SR values, but the LR values differ significantly. This might suggest that the LR plays a specific role in determining how much to deviate from average

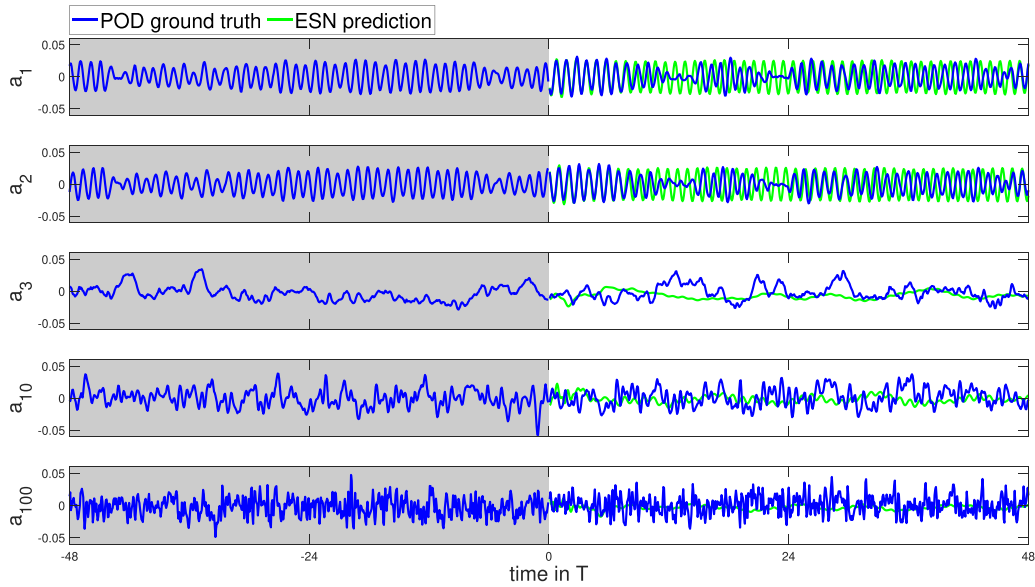


Figure 21. Actual and predicted values of the temporal coefficients of modes 1, 2, 3, 10, and 100 for $Re = 1000$. The lines marked with ESN indicate the predicted values of the network with the optimized set of hyperparameters for statistical prediction, while the lines marked with POD show the actual values of the temporal coefficients during training and testing.

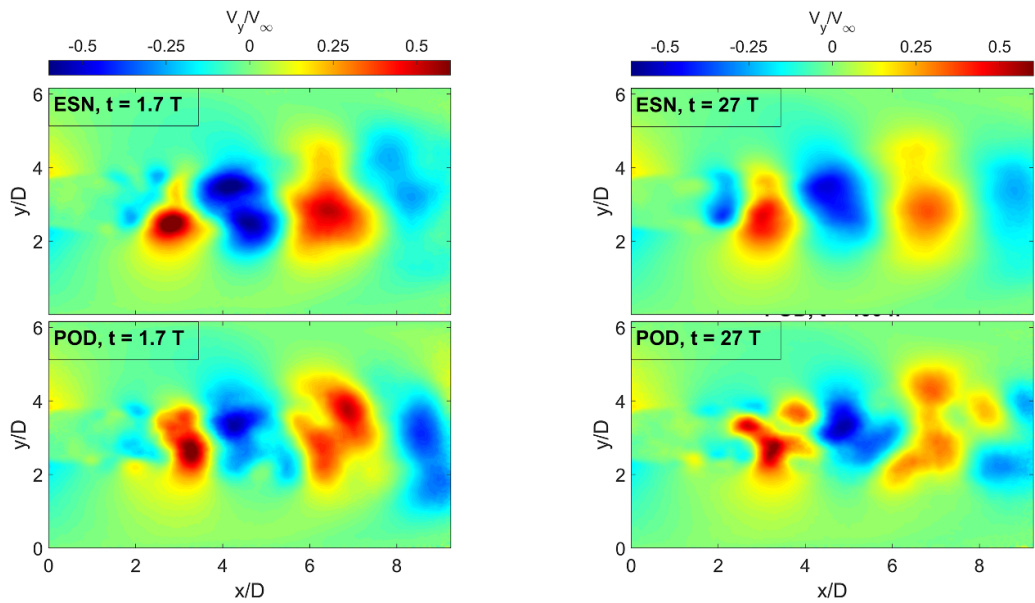


Figure 22. The vertical velocity fields of optimized statistical prediction and reality for $t = 25$ (left) and 400 (right) for $Re = 1000$.

oscillations in order to capture more complex perturbations. For the purpose of comparison, the statistical optimization and corresponding PDFs of V_y and V_x for $Re = 100$ are added to the [appendix](#).

4. Conclusions

In this study, an ESN was used to elaborate on its ability to predict a turbulent flow deterministically and statistically. For this purpose, the wake flow behind a cylinder commonly known as the KVS was used at different Reynolds numbers to increase

the complexity of the flow. PIV measurements were conducted to collect the experimental data of the KVS in $Re = 100$ and 1000 . Data reduction of the velocity field was achieved by POD. Subsequently, only the first 100 most energetic POD modes were considered, assuming that higher modes contribute little to the variance of the flow and reflect small-scale fluctuations. TCPM were used to train and test an ESN with 3000 neurons for 700 time steps, which corresponded to 39 steady vortex shedding events for $Re = 100$ and 47 unsteady vortex shedding events for $Re = 1000$. To reach the best deterministic and statistical predictions of the flow, the ESN was optimized according to its three main hyperparameters, INS, Leaking

Rate (LR), and Spectral Radius (SR). Other parameters were left constant, e.g. the reservoir density.

The aim of deterministic prediction was the accurate reconstruction of the flow direction for V_y at $Re = 1000$ and 100 for values above 17.6 and 14.8 mm s^{-1} , respectively. This was termed as Vertical Velocity Prediction of Direction (VVPD). For the case of steady vortex shedding in $Re = 100$, the optimization resulted in the highest VVPD value of 97% for the case of $INS = 4$, $LR = 0.7$, and $SR = 0.8$. In this case both TCPMs and velocity fields matched well with the actual flow. For unsteady vortex shedding in $Re = 1000$, the optimization resulted in the highest VVPD value of 60% for the case of $INS = 3$, $LR = 0.95$, and $SR = 0.97$. The ESN modeled the TCPMs well, specially for the lower, more energetic modes. For the first two modes the phase of the oscillations was predicted well, however, the amplitudes were exaggerated after some time steps. Therefore, the reconstructed V_y fields by the ESN matched the actual flow in the early time steps, but later they had larger magnitudes of V_y , even though they were successful in predicting the position of both up and downward V_y in later time steps after 40 vortex shedding events. For this case, the random state of the ESN was also varied by variation of Random Seed (RS) for a hundred different values. Different seeds had different set of hyperparameters for their best VVPD. However, LR and SR values were predominantly close to one, and the INS was changing with the change of RS. Moreover, maximum VVPD varied between 0.53 and 0.73 with an average of 0.62, indicating that the RS variation will not have a significant impact on the prediction quality.

Statistical predictions aimed at reconstructing a flow that has the best fit with the actual flow in terms of its PDF of V_y and V_x for $Re = 1000$. This was accomplished by searching for a set of hyperparameters with predictions closest to actual flow in terms of the Structural SIMilarity index (SSIM) of the V_y fields. The optimization resulted in $INS = 5$, $LR = 0.1$, and $SR = 0.99$ for hyperparameters. The PDF of V_y and V_x of the optimized prediction was very similar to the actual flow. Here, the predictions matched well with the actual flow in terms of both TCPMs and velocity fields.

In general, the ESN was able to predict the flow both deterministically and statistically. However, precise predictions of the unsteady vortices in $Re = 1000$ were not possible. The question if turbulence flow can be predicted at all has to be answered in further studies among the community. However, it was clearly shown that the time horizon for a good representation of the turbulent wake flow was decreasing with increasing turbulence level. LR values for the deterministically and statistically optimized cases differed significantly, which

illustrates how the reservoir update speed influences the difference between statistical and deterministic predictions. The second notable difference was the independence of the statistical predictions from SR, which was in contrast to the deterministic predictions, the latter of which were very sensitive to variations in SR.

As a preliminary step towards understanding the benefits and limitations of using ESNs to predict fluid flow, this study can be extended in several directions. As a first step, the ESNs can be used in more complex flows with different types of periodicity. In addition, different methods of flow reduction other than POD might be employed to increase the efficiency and quality of data reduction. Lastly, it might be advantageous to combine ESNs with other ML algorithms, so that their points of strength can be taken advantage of while also covering their downsides.

Data availability statement

The data that support the findings of this study are available upon reasonable request from the authors.

Acknowledgments

This work is supported by the Project No. P 2018-02-001 ‘DeepTurb—Deep Learning in and of Turbulence’, which is funded by the Carl Zeiss Foundation. We thank Alexander Thieme and Helmut Hoppe for their valuable support in conducting the experiments.

Appendix

For the purpose of comparison, the statistical optimization and corresponding PDFs of V_y and V_x for $Re = 100$ are added in this section. Figure A.1 shows the optimization of the ESN with respect to its main three hyperparameters (INS, LR, SR) based on the average SSIM values of the predictions. The hyperparameter values for the grid search are again represented in table 1. The optimized ESN for this case has $INS = 1$, $LR = 0.9$, and $SR = 0.9$. Figure A.2 shows the PDFs of the V_y and V_x of the optimal statistical (green lines) and deterministic (red lines) predictions of ESN in comparison to POD or the actual flow (blue lines) in $Re = 100$. Figure A.3 shows the temporal evolution of the TCPMs for $Re = 100$ in the real flow (blue lines) and in the prediction (green lines) of the optimized case for modes 1, 2, 3, 10, and 100. Finally, the V_y fields of the predictions and corresponding real flow fields for $t = 25$ and 400 are shown in figure A.4.

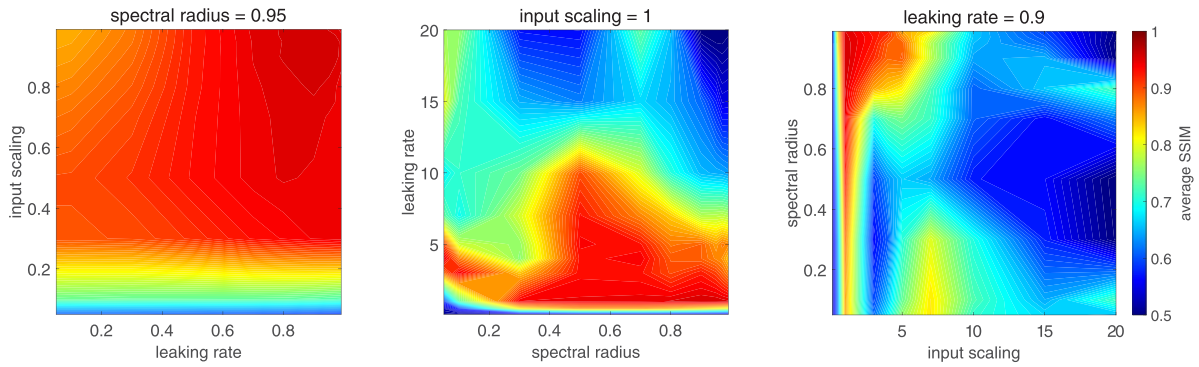


Figure A.1. Average SSIM (structural similarity index) values with respect to variations in input scaling and leaking rate (left), leaking rate and spectral radius (middle), and spectral radius and input scaling (right) for $Re = 100$.

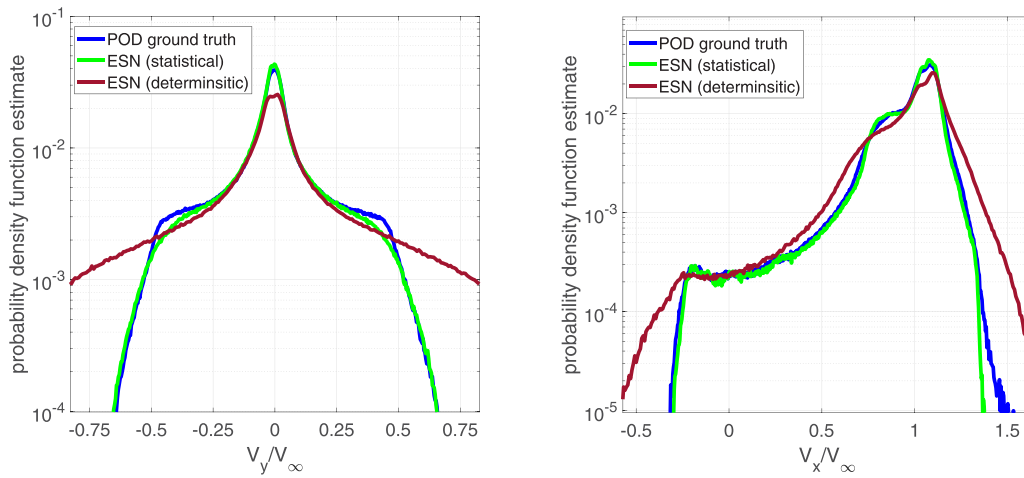


Figure A.2. Probability density function of the actual flow and the optimized statistical prediction of it for $Re = 100$.

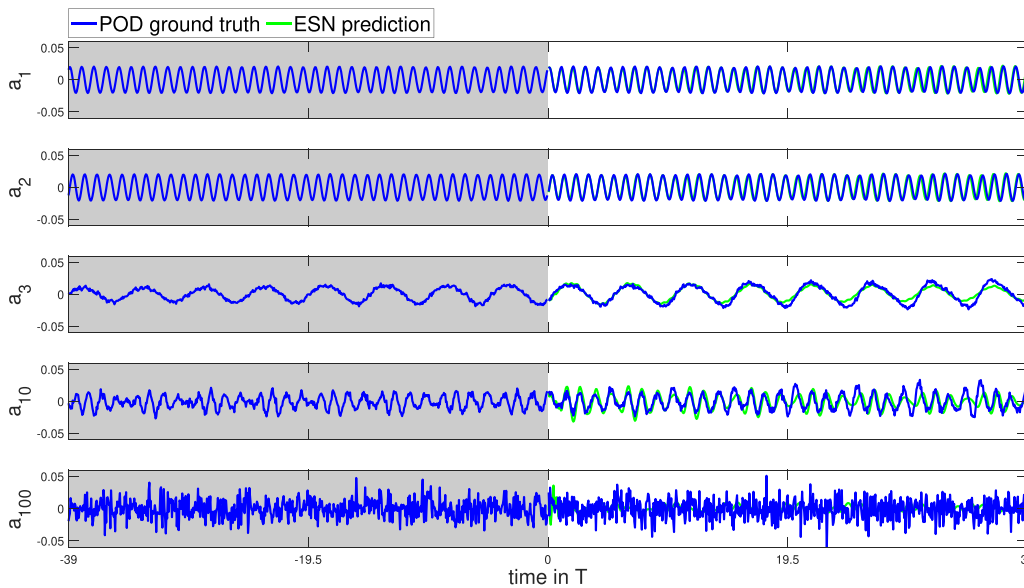


Figure A.3. Actual and predicted values of the temporal coefficients of modes 1, 2, 3, 10, and 100 for $Re = 100$. The lines marked with ESN indicate the predicted values of the network with the optimized set of hyperparameters for statistical prediction, while the lines marked with POD show the actual values of the temporal coefficients during training and testing.

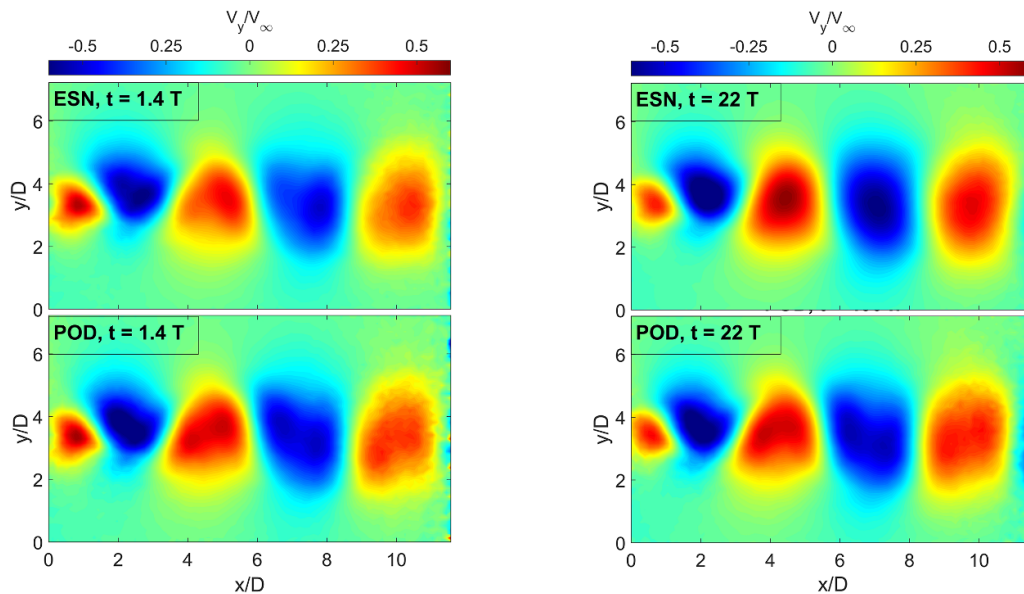



Figure A.4. The vertical velocity fields of optimized statistical prediction and reality for $t = 25$ (left) and 400 (right) for $Re = 100$.

ORCID iDs

Mohammad Sharifi Ghazijahani  <https://orcid.org/0000-0001-5885-4138>

Florian Heyder  <https://orcid.org/0000-0002-5415-4510>

Jörg Schumacher  <https://orcid.org/0000-0002-1359-4536>

Christian Cierpka  <https://orcid.org/0000-0002-8464-5513>

References

- [1] Brunton S L, Noack B R and Koumoutsakos P 2020 Machine learning for fluid mechanics *Annu. Rev. Fluid Mech.* **52** 477–508
- [2] Pandey S, Schumacher J and Sreenivasan K R 2020 A perspective on machine learning in turbulent flows *J. Turbul.* **21** 567–84
- [3] Brunton S L, Hemati M S and Taira K 2020 Special issue on machine learning and data-driven methods in fluid dynamics *Theor. Comput. Fluid Dyn.* **34** 333–7
- [4] Beck A and Kurz M 2021 A perspective on machine learning methods in turbulence modeling *GAMM-Mitt.* **44** 202100002
- [5] Vinuesa R and Brunton S L 2022 Enhancing computational fluid dynamics with machine learning *Nat. Comput. Sci.* **2** 358–66
- [6] Duraisamy K, Iaccarino G and Xiao H 2019 Turbulence modeling in the age of data *Annu. Rev. Fluid Mech.* **51** 357–77
- [7] Wu J-L, Xiao H and Paterson E 2018 Physics-informed machine learning approach for augmenting turbulence models: a comprehensive framework *Phys. Rev. Fluids* **3** 074602
- [8] Bin Y, Chen L, Huang G and Yang X I A 2022 Progressive, extrapolative machine learning for near-wall turbulence modeling *Phys. Rev. Fluids* **7** 084610
- [9] Yan C, Zhang Y and Chen H 2022 Data augmented turbulence modeling for three-dimensional separation flows *Phys. Fluids* **34** 075101
- [10] Fang L, Bao T, Xu W, Zhou Z, Du J and Jin Y 2022 Data driven turbulence modeling in turbomachinery—an applicability study *Comput. Fluids* **238** 105354
- [11] Raissi M, Yazdani A and Karniadakis G E 2020 Hidden fluid mechanics: learning velocity and pressure fields from flow visualizations *Science* **367** 1026–30
- [12] Ren F, Hu H and Tang H 2020 Active flow control using machine learning: a brief review *J. Hydrodynamics* **32** 247–53
- [13] Gautier N, Aider J-L, Duriez T, Noack B, Segond M and Abel M 2015 Closed-loop separation control using machine learning *J. Fluid Mech.* **770** 442–57
- [14] Park J and Choi H 2020 Machine-learning-based feedback control for drag reduction in a turbulent channel flow *J. Fluid Mech.* **904** A24
- [15] Zhuang J, Kochkov D, Bar-Sinai Y, Brenner M P and Hoyer S 2021 Learned discretizations for passive scalar advection in a two-dimensional turbulent flow *Phys. Rev. Fluids* **6** 064605
- [16] Cheng Y, Giometto M G, Kauffmann P, Lin L, Cao C, Zupnick C, Li H, Li Q, Huang Y, Abernathy R et al 2022 Deep learning for subgrid-scale turbulence modeling in large-Eddy simulations of the convective atmospheric boundary layer *J. Adv. Model. Earth Syst.* **14** e2021MS002847
- [17] Brenner M, Eldredge J and Freund J 2019 Perspective on machine learning for advancing fluid mechanics *Phys. Rev. Fluids* **4** 100501
- [18] Srinivasan P A, Guastoni L, Azizpour H, Schlatter P and Vinuesa R 2019 Predictions of turbulent shear flows using deep neural networks *Phys. Rev. Fluids* **4** 054603
- [19] Sekar V, Jiang Q, Shu C and Khoo B C 2019 Fast flow field prediction over airfoils using deep learning approach *Phys. Fluids* **31** 057103
- [20] Fukami K, Nabae Y, Kawai K and Fukagata K 2019 Synthetic turbulent inflow generator using machine learning *Phys. Rev. Fluids* **4** 064603
- [21] Pawar S, Rahman S, Vaddirreddy H, San O, Rasheed A and Vedula P 2019 A deep learning enabler for nonintrusive reduced order modeling of fluid flows *Phys. Fluids* **31** 085101

- [22] Pandey S and Schumacher J 2020 Reservoir computing model of two-dimensional turbulent convection *Phys. Rev. Fluids* **5** 113506
- [23] Heyder F and Schumacher J 2021 Echo state network for two-dimensional turbulent moist Rayleigh–Bénard convection *Phys. Rev. E* **103** 053107
- [24] Heyder F, Mellado J P and Schumacher J 2022 Two-dimensional convective boundary layer: numerical analysis and echo state network model *Phys. Rev. E* submitted (arXiv:2108.06195)
- [25] Pandey S, Teutsch P, Mäder P and Schumacher J 2022 Direct data-driven forecast of local turbulent heat flux in Rayleigh–Bénard convection *Phys. Fluids* **34** 045106
- [26] Ren F, Wang C and Tang H 2019 Active control of vortex-induced vibration of a circular cylinder using machine learning *Phys. Fluids* **31** 093601
- [27] Mei Y-F, Zheng C, Aubry N, Li M-G, Wu W-T and Liu X 2021 Active control for enhancing vortex induced vibration of a circular cylinder based on deep reinforcement learning *Phys. Fluids* **33** 103604
- [28] Rabault J, Kuchta M, Jensen A, Réglade U and Cerardi N 2019 Artificial neural networks trained through deep reinforcement learning discover control strategies for active flow control *J. Fluid Mech.* **865** 281–302
- [29] Tang H, Rabault J, Kuhnle A, Wang Y and Wang T 2020 Robust active flow control over a range of reynolds numbers using an artificial neural network trained through deep reinforcement learning *Phys. Fluids* **32** 053605
- [30] Ren F, Rabault J and Tang H 2021 Applying deep reinforcement learning to active flow control in weakly turbulent conditions *Phys. Fluids* **33** 037121
- [31] Raibaudo C, Zhong P, Noack B R and Martinuzzi R J 2020 Machine learning strategies applied to the control of a fluidic pinball *Phys. Fluids* **32** 015108
- [32] Tokarev M, Palkin E and Mullyadzhyanov R 2020 Deep reinforcement learning control of cylinder flow using rotary oscillations at low reynolds number *Energies* **13** 5920
- [33] Bieker K, Peitz S, Brunton S L, Kutz J N and Dellnitz M 2020 Deep model predictive flow control with limited sensor data and online learning *Theor. Comput. Fluid Dyn.* **34** 577–91
- [34] Xu H, Zhang W, Deng J and Rabault J 2020 Active flow control with rotating cylinders by an artificial neural network trained by deep reinforcement learning *J. Hydrodynamics* **32** 254–8
- [35] Han B-Z, Huang W-X and Xu C-X 2022 Deep reinforcement learning for active control of flow over a circular cylinder with rotational oscillations *Int. J. Heat Fluid Flow* **96** 109008
- [36] Castellanos R, Cornejo Maceda G, de la Fuente I, Noack B, Ianiro A and Discetti S 2022 Machine-learning flow control with few sensor feedback and measurement noise *Phys. Fluids* **34** 047118
- [37] Rabault J, Ren F, Zhang W, Tang H and Xu H 2020 Deep reinforcement learning in fluid mechanics: a promising method for both active flow control and shape optimization *J. Hydrodynamics* **32** 234–46
- [38] Viquerat J, Rabault J, Kuhnle A, Ghraieb H, Larcher A and Hachem E 2021 Direct shape optimization through deep reinforcement learning *J. Comput. Phys.* **428** 110080
- [39] Eivazi H, Veisi H, Naderi M H and Esfahanian V 2020 Deep neural networks for nonlinear model order reduction of unsteady flows *Phys. Fluids* **32** 105104
- [40] Hasegawa K, Fukami K, Murata T and Fukagata K 2020 CNN-LSTM based reduced order modeling of two-dimensional unsteady flows around a circular cylinder at different reynolds numbers *Fluid Dyn. Res.* **52** 065501
- [41] Hasegawa K, Fukami K, Murata T and Fukagata K 2020 Machine-learning-based reduced-order modeling for unsteady flows around bluff bodies of various shapes *Theor. Comput. Fluid Dyn.* **34** 367–83
- [42] Fukami K, Fukagata K and Taira K 2020 Assessment of supervised machine learning methods for fluid flows *Theor. Comput. Fluid Dyn.* **34** 497–519
- [43] Bright I, Lin G and Kutz J N 2013 Compressive sensing based machine learning strategy for characterizing the flow around a cylinder with limited pressure measurements *Phys. Fluids* **25** 127102
- [44] Patil A, Viquerat J, Larcher A, El Haber G and Hachem E 2021 Robust deep learning for emulating turbulent viscosities *Phys. Fluids* **33** 105118
- [45] Li B, Yang Z, Zhang X, He G, Deng B-Q and Shen L 2020 Using machine learning to detect the turbulent region in flow past a circular cylinder *J. Fluid Mech.* **905** A10
- [46] Ye S, Zhang Z, Song X, Wang Y, Chen Y and Huang C 2020 A flow feature detection method for modeling pressure distribution around a cylinder in non-uniform flows by using a convolutional neural network *Sci. Rep.* **10** 1–10
- [47] Morast A 2019 Learning from the von Kármán Vortex Street *PhD Thesis* South Dakota School of Mines and Technology, Rapid City
- [48] Morimoto M, Fukami K and Fukagata K 2021 Experimental velocity data estimation for imperfect particle images using machine learning *Phys. Fluids* **33** 087121
- [49] Fukami K, Fukagata K and Taira K 2021 Machine-learning-based spatio-temporal super resolution reconstruction of turbulent flows *J. Fluid Mech.* **909** A9
- [50] Jin X, Cheng P, Chen W-L and Li H 2018 Prediction model of velocity field around circular cylinder over various reynolds numbers by fusion convolutional neural networks based on pressure on the cylinder *Phys. Fluids* **30** 047105
- [51] Wan Z Y and Sapsis T P 2018 Machine learning the kinematics of spherical particles in fluid flows *J. Fluid Mech.* **857** R2
- [52] Arntzen S 2019 Prediction of flow-fields by combining high-fidelity CFD data and machine learning algorithms *Master Thesis* TU Delft (available at: <https://repository.tudelft.nl/islandora/object/uuid:9092d9df-e827-4ade-9009-81bb8e854ae7>)
- [53] Jaeger H 2001 The “echo state” approach to analysing and training recurrent neural networks—with an erratum note *Technical Report* (GMD-Forschungszentrum Informationstechnik) vol 148
- [54] Lukoševičius M and Jaeger H 2009 Reservoir computing approaches to recurrent neural network training *Comput. Sci. Rev.* **3** 127
- [55] Hermans M and Schrauwen B 2010 Memory in reservoirs for high dimensional input *IJCNN* p 1
- [56] Racca A and Magri L 2022 Data-driven prediction and control of extreme events in a chaotic flow (arXiv:2204.11682)
- [57] Morando S, Jemei S, Gouriveau R, Zerhouni N and Hissel D 2013 Fuel cells prognostics using echo state network *IECON 2013—39th Annual Conf. IEEE Industrial Electronics Society* pp 1632–7
- [58] Pathak J, Hunt B R, Girvan M, Lu Z and Ott E 2018 Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach *Phys. Rev. Lett.* **120** 024102
- [59] Lu Z, Pathak J, Hunt B R, Girvan M, Brockett R and Ott E 2017 Reservoir observers: model-free inference of unmeasured variables in chaotic systems *Chaos* **27** 041102
- [60] Pfeffer P, Heyder F and Schumacher J 2022 Hybrid quantum-classical reservoir computing of thermal convection flow *Phys. Rev. Res.* **4** 033176
- [61] Kähler C J, Astarita T, Vlachos P P, Sakakibara J, Hain R, Discetti S, La Foy R and Cierpka C 2016 Main results of the 4th international PIV challenge *Exp. Fluids* **57** 1–71
- [62] Westerweel J and Scarano F 2005 Universal outlier detection for PIV data *Exp. Fluids* **39** 1096–100

- [63] Sirovich L 1987 Turbulence and the dynamics of coherent structures. I. Coherent structures *Q. Appl. Math.* **45** 561–71
- [64] Deane A, Kevrekidis I, Karniadakis G E and Orszag S 1991 Low-dimensional models for complex geometry flows: application to grooved channels and circular cylinders *Phys. Fluids A* **3** 2337–54
- [65] Taira K, Brunton S L, Dawson S T M, Rowley C W, Colonius T, McKeon B J, Schmidt O T, Gordeyev S, Theofilis V, Ukeiley L S 2017 Modal analysis of fluid flows: an overview *AIAA J.* **55** 4013–41
- [66] Cierpka C, Weier T and Gerbeth G 2010 Synchronized force and particle image velocimetry measurements on a NACA 0015 in poststall under control of time periodic electromagnetic forcing *Phys. Fluids* **22** 075109
- [67] Feng L-H, Wang J-J and Pan C 2011 Proper orthogonal decomposition analysis of vortex dynamics of a circular cylinder under synthetic jet control *Phys. Fluids* **23** 014106
- [68] Doya K 1992 Bifurcations in the learning of recurrent neural networks [*Proc.*] 1992 *IEEE Int. Symp. on Circuits and Systems* vol 6 pp 2777–80
- [69] Maass W, Natschläger T and Markram H 2002 Real-time computing without stable states: a new framework for neural computation based on perturbations *Neural Comput.* **14** 2531
- [70] Lukoševičius M 2012 A practical guide to applying echo state networks *Neural Networks: Tricks of the Trade* (Berlin: Springer) pp 659–86
- [71] Yildiz I B, Jaeger H and Kiebel S J 2012 Re-visiting the echo state property *Neural Netw.* **35** 1
- [72] easyesn library v0.1.6.1 (available at: <https://github.com/kalekiu/easyesn>)
- [73] turbESN library (v0.1.8.3) (available at: <https://github.com/flohey/turbESN>)
- [74] Wang Z, Bovik A C, Sheikh H R and Simoncelli E P 2004 Image quality assessment: from error visibility to structural similarity *IEEE Trans. Image Process.* **13** 600–12