A Deep Learning based Approach to Reduced Order Modelling for Turbulent Flow Control using LSTM Neural Networks

Authors:
Arvind T. Mohan Datta V. Gaitonde

Paper available at: https://arxiv.org/pdf/1804.09269.pdf

Presentation by Alix Bernard

March 2020

Content

Introduction

The modelling of turbulent flows in fluids is useful for many applications such as aeronautics, energy generation system, weather forecasting, etc.
The use of Computational Fluid Dynamics (CFD) can give us more detailed insight on turbulent flows, the Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS) techniques provide high fidelity data however they are very time intensive and they generate extremely high dimensional large datasets making it hard to efficiently handle and analyse.

Techniques for modelling turbulent flows with high fidelity while minimizing the computation and storage costs associated with LES/DNS are of active research. Such models in low dimensional space are referred to Reduced Order Models (ROM), they have two primary objectives:

a) The ability to model the key dynamics/coherent features of the turbulent flow
b) Provide an efficient mean of data compression for LES/DNS datasets.

ROM for engineering application has been a major research topic for a few decades. Its goal is to model key physic features of a flow-field without computing full Navier-Stokes (NS) equations; this is done by extracting a low dimensional subspace typically using Proper Orthogonal Decomposition (POD aka PCA).

Once the ROM is done, the next step is to use this reduced basis for modelling the flow at future time instants, a highly popular technique being the Galerkin Projection (GP) approach.

The GP method consist in the use of spatio-temporal dynamics captured by the reduced basis and then evolve them in time instead of computing the NS equations. Such approach ensures us cheaper computation.

However GP models may be unstable under different conditions, this problem leading to the focus on Galerkin-free alternatives.

This paper explores a non-Galerkin projection based approach to ROM through deep learning. The focus is on exploiting Neural Networks (NN) to “learn” the key dynamics of turbulent flows from high-fidelity simulation databases and use them to generates ROMs to model the flow field at future instants, for flow control applications.
For this, Recurrent NNs (RNN) will be used considering there efficiency for modelling sequential data and more precisely the popular Long Short-Term Memory (LSTM) variant that accounts for memory since data obtained from turbulent flows may exhibit memory effects which have to be accommodated for accurate predictive models.

Long Short-Term Memory Neural Networks

The goal of predicting the evolution of a flow-field through its POD temporal coefficients is a sequence modelling problem in machine learning that requires to preserve an order on the observations, hence the use of RNNs.

The LSTM variant overcomes some memory problems (like the Vanishing Gradient one), it also learn and harness temporal dependence from the data, and moreover it also utilize internal memory:

Predictions are conditional to the recent context in the input sequence, not just what has been presented as the current input in the network, e.g. can be shown one observation at a time and learn what observation it has seen previously are relevant and how they can be used to make a prediction.

The LSTM cell usually contains three gates: input, output, and forget; these gates allow the LSTM to control the flow of training information by respectively adding, letting through the next cell, or removing information.

These gates are denoted: i, o, f; the cell state is denoted C; and the cell input and output are denoted: x, h.

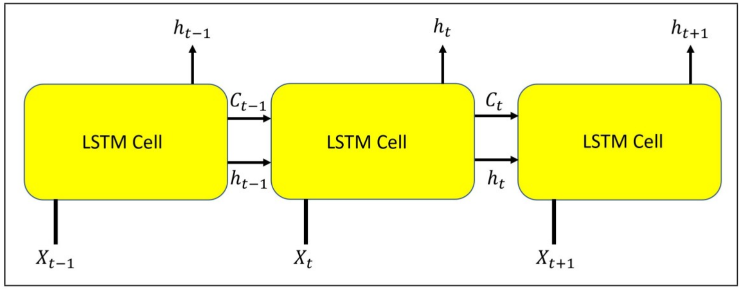

Figure 1. LSTM Layout with Cell Connections

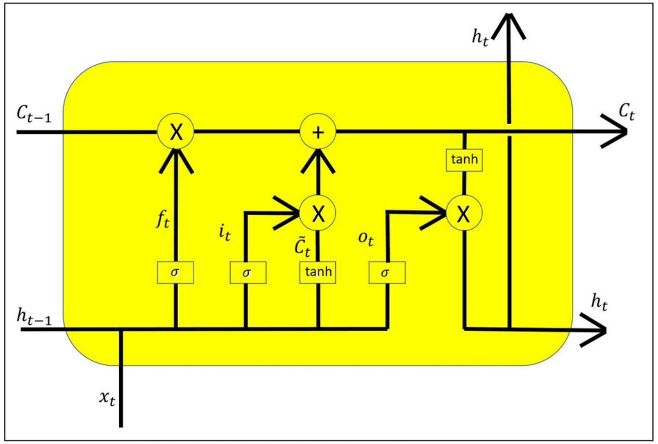

Figure 2. Architecture of a LSTM Cell with Various Gates

The equations to compute the gates and states of the cell are:

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \quad (1)$$

$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \quad (2)$$

$$\tilde{C}_t = tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \quad (3)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \quad (4)$$

$$o_t = \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \quad (5)$$

$$h_t = o_t * tanh\left(C_t\right) \quad (6)$$

W are the weights for each gates and C-tilde is the updated cell state. The states are propagated ahead through the network (cf. fig. 1) and weights are updates by back propagation through time.
The forget gates prevent over-fitting.

The paper explores two types of LSTMs:

a) The traditional LSTM algorithm by Hochreiter
b) The bidirectional LSTM (BiLSTM) by Graves et al.

In the classic LSTM each cell has an input dependent on the cell at the previous time instant.

On the other hand the BiLSTM has a two way flow of information, the sequence is therefore trained using by two LSTM networks, one in each direction.

These two networks are connected to the same output layer.

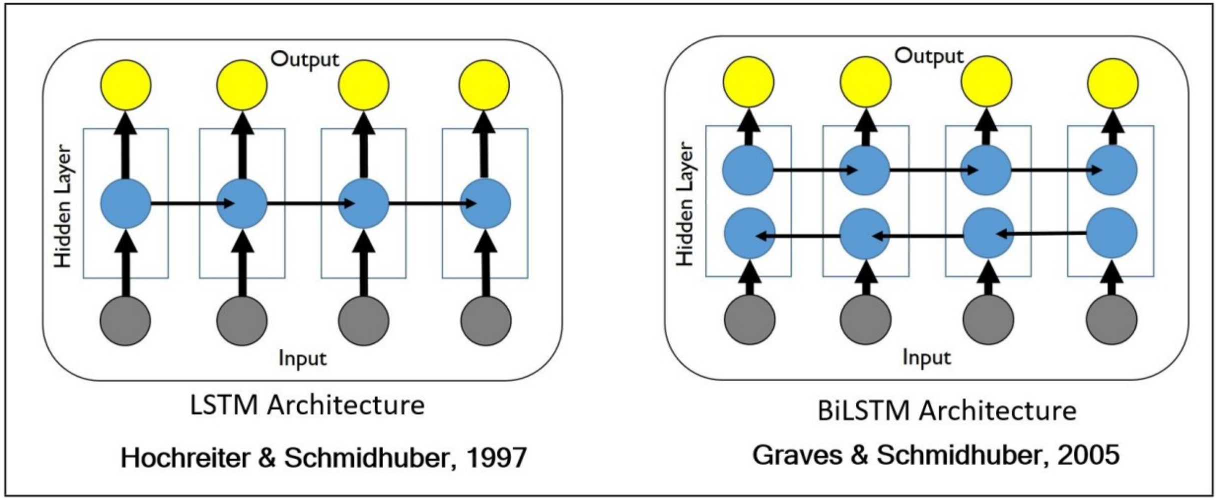

Figure 3. LSTM and BiLSTM Architectures

Methodology

For a NN to learn it needs data, therefore two DNS databases from Johns Hopkins turbulence database (JHTB) will be used since it is a canonical case:

a) Force Isotropic Turbulence dataset (ISO)
Sourced from 3-D DNS Navier-Stokes simulations solved using spectral method on a grid size of 1024^3 with 5023 time steps for 10 seconds of high-fidelity flow data (~ every 2 ms).

b) Magnetohydrodynamic Turbulence dataset (MHD)
Sourced from a 3-D DNS Navier-Stokes simulation solved on a grid size of 1024^3 with 1024 time steps for 2.056 seconds of high-fidelity flow data (~ every 2 ms).

Considering the high computational cost of a 3-D DNS,the JHTB provides these datasets for only one single Reynolds number, whereas a DL approach would preferably require multiple closely related datasets.

To palliate this problem, the 3-D ISO dataset is decomposed into 2-D planes of the same size, we then choose a subset of size N of these 2-D planes with their associated snapshots (k=5023 for ISO, k=1024 for MHD) or a smaller subset.

Such process enable to create a large number of unique training datasets – for more info refer to paper page 5.

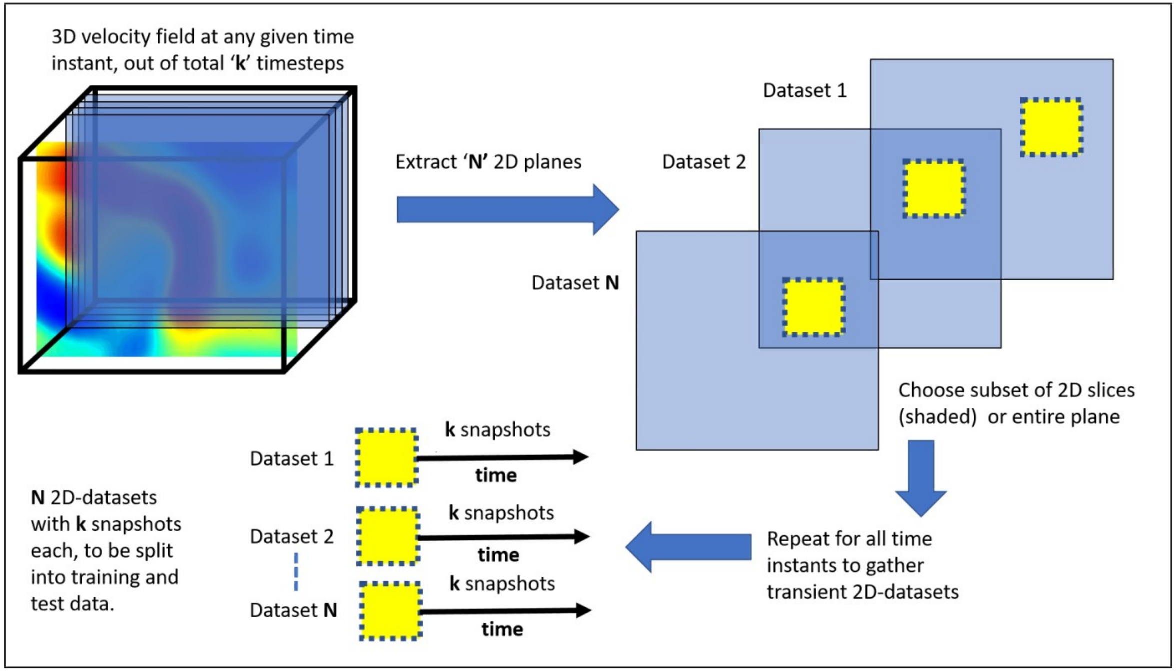

Figure 4. Extracting 2-D datasets from a 3-D flow field

An advantage of well designed NNs models is that they account for the dissimilarities in various training datasets extracting only the key features of the data which tend to be universal, it will be shown in later results. A classic strategy consists in using dominant POD modes to represent the key features of the flow since they usually capture most of the flow energy.

Moreover the evolution in time of the POD modes is given by their temporal coefficients that can be used to describe temporal evolution of the key features, denoted as f in the following equation:

$$f = \sum_{i=1}^{i=k} \phi_i \alpha(t)_i \qquad (7)$$

ϕ_i – a POD mode for the snapshot i

$\alpha(t)_i$ – the vector of time coefficient for ϕ_i

ROMs can then be built by predicting *α(t)*$_i$ at future time instant for dominant POD modes, using techniques such as GP.

This is extremely time efficient for DNS/LES data since it only models the time evolution of a few important flow structures.

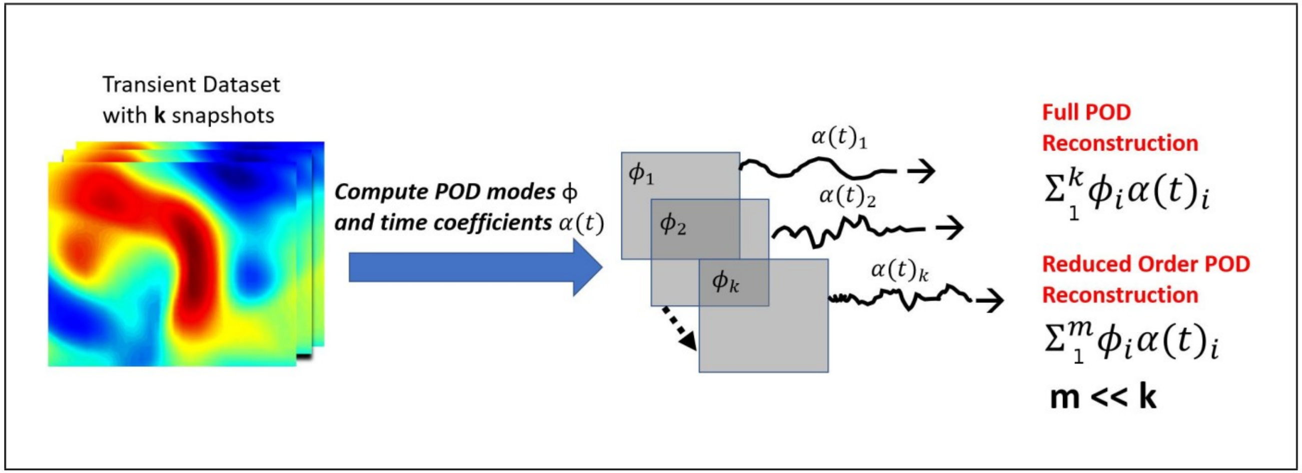

Figure 5. Proper Orthogonal Decomposition and reduced order reconstruction

In the following, the existing ROM framework will be retained but LSTM NNs will be explored to model $\alpha(t)_i$ instead of GP. The key steps in the LSTM-ROM methodology are:

1. Select the number of 2-D planes to be used as training datasets.
2. From the 2-D planes select a test dataset.
3. Extract dominant POD modes (usually 5 to 10 modes with highest eigenvalues) and their $\alpha(t)_i$ for each of the training datasets and test dataset; the $\alpha(t)_i$ of the test dataset POD modes will be used to validate the LSTM-ROM prediction.

4. Train LSTM/BiLSTM NN for the dominant POD modes chosen previously.

5. Validation: using a short history of the dataset POD mode $\alpha(t)_i$ as input, predict the next few time instants $\alpha(t+t')_i$ and compare the prediction with the true $\alpha(t+t')_i$ from the test dataset. Repeat this for all chosen dominant POD modes.

6. Using POD modes and the DL-ROM predicted temporal coefficients, compute predicted flow field using eq. 7.

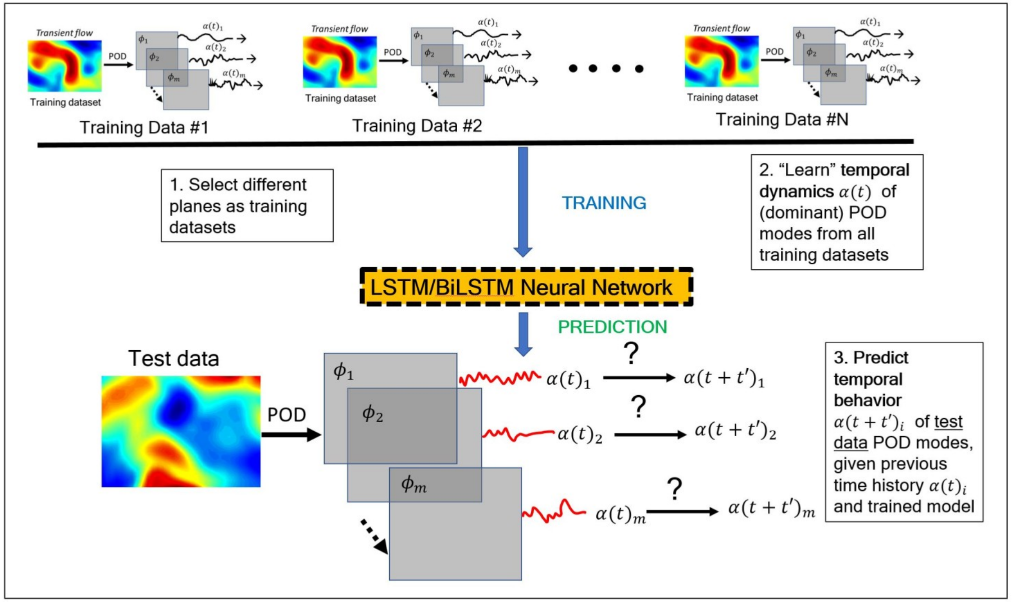

Figure 6. LSTM-ROM Methodology

For details on the LSTM implementation software framework and training parameters refer to the appendix in the paper.

In the paper, the POD modes and temporal coefficients from five 2-D planes which are equidistant from each other are used as the training data; and the test dataset consist of a single 2-D plane also equidistant, whose POD temporal coefficient are modelled using the LSTM NN.

It is often assumed that in ROMs, including Galerkin-based ones, the dominant POD modes for the training and test datasets are qualitatively similar:

e.g. flows with a narrow range of Reynolds number can show qualitative (but not quantitative) similar behavior which are encoded in their dominant POD modes.
For the ISO training and test 2-D planes used here.

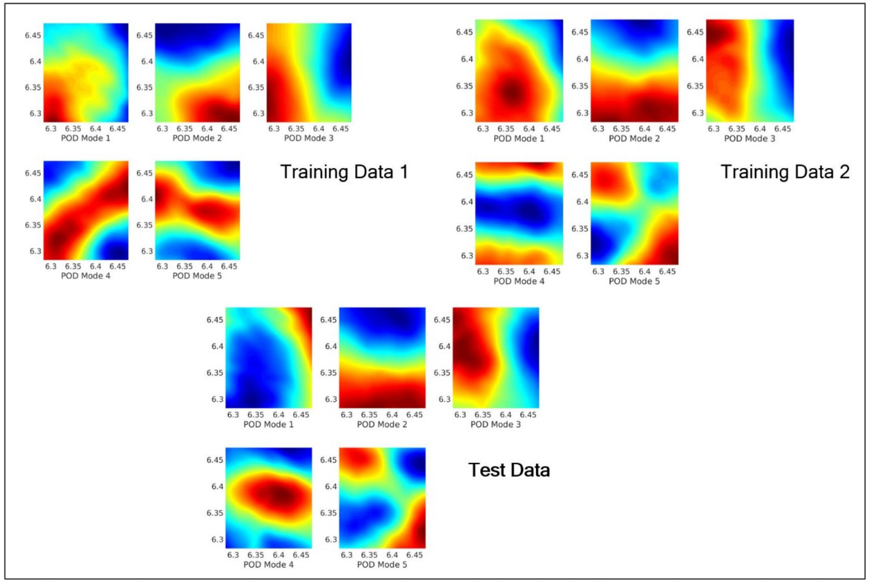

Figure 7. Examples of Dominant POD mode U-velocity fields from training and test datasets

We can observe significant qualitative similarity due to the data-driven nature of the technique used and use them to model the behavior of an unseen dataset of the same family.

The training strategy consists in decomposing the training $\alpha(t)_i$ signals from all the datasets into several short samples by a moving window. These samples are divided into an input part and its corresponding output part.

The LSTM NN is then trained using a set of input and output signal, the training problem consist now in learning signatures in 1-D signals of the training datasets and predict a 1-D signal from a test dataset.

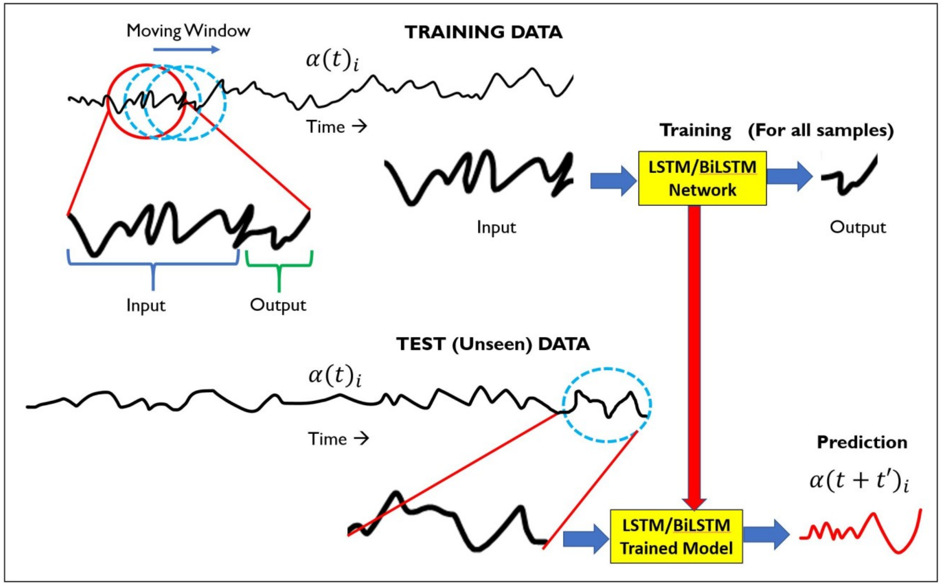

Figure 8. LSTM training strategy with input/output framework

Results

Time window/prediction horizon choices have to be made at the NN design level before training as they can significantly affect the accuracy of the model; to do so, some trial and error based on the physics of the flow are made.

The presented results have been generated with a time window and predictive horizon of 10 time steps each. Performance with longer windows/horizons will be studied as well.

In a first part the results on ISO are presented, then in a second part the results on MHD.

Isotropic Turbulence

The LSTM and BiLSTM NNs are trained on each of the 5 dominant POD temporal coefficients therefore obtaining 5 trained models – meaning if the ROM is built using dominant m modes, m models would be required.

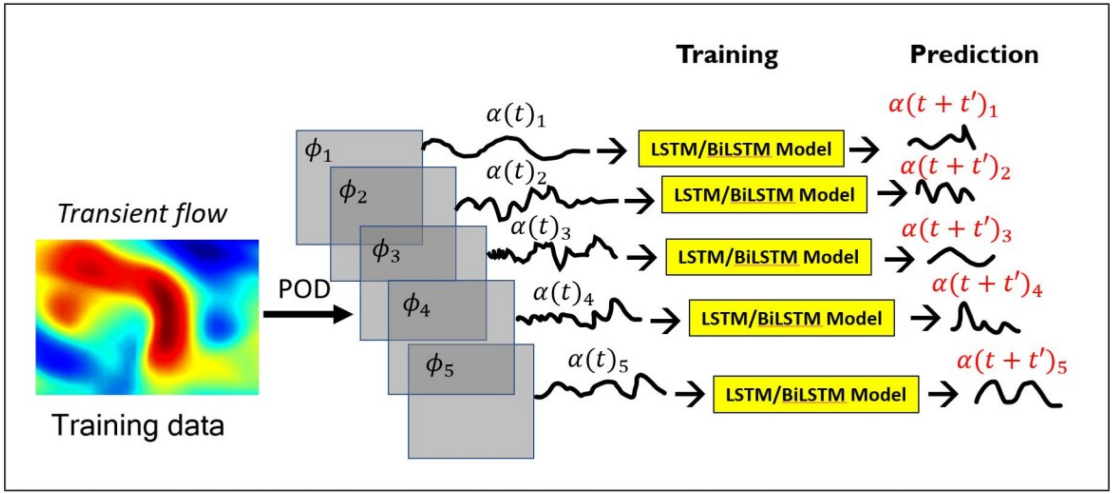

Figure 9. Training of a Multiple NN models for all POD dominant modes

To evaluate the prediction, an input of any sample from the temporal coefficients of the relevant mode from the test dataset is provided, the NN model then predict the sample which follows immediately after it. Having thousands of samples to test for a given mode the findings are statistically significant.

Figure 10 shows the prediction results at a randomly chosen sample, we notice that surprisingly the BiLSTM architecture is less accurate than the LSTM one.

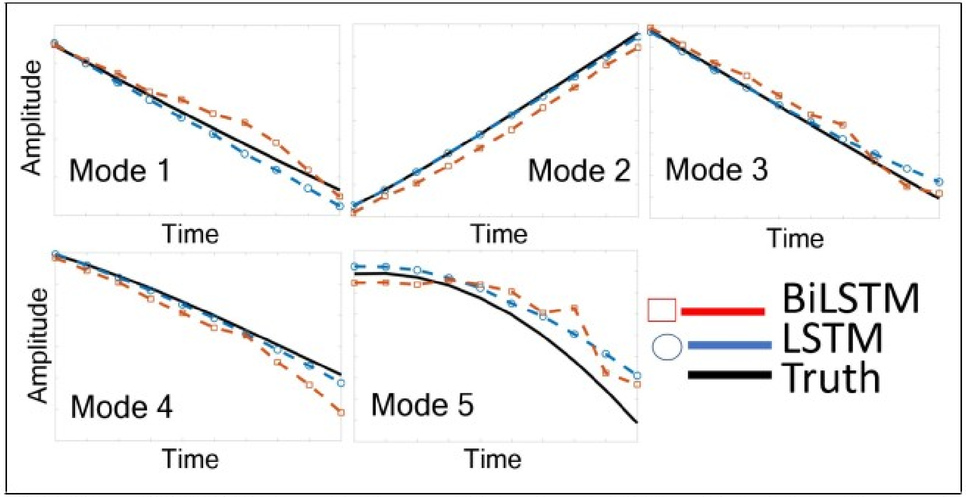

Figure 10. LSTM and BiLSTM predictions of Dominant POD $\alpha(t+t')$ for Isotropic turbulence test data

To evaluate the accuracy of these NN architectures the Mean Absolute Scaled Error (MASE) metric is chosen to quantify the deviation of prediction trend and calculated over a large number of sample. In figure 11 and 12 the average of MASE for each mode is displayed for LSTM and BiLSTM respectively.

We can see that the MASE is generally low and its average for each mode is higher for BiLSTM.

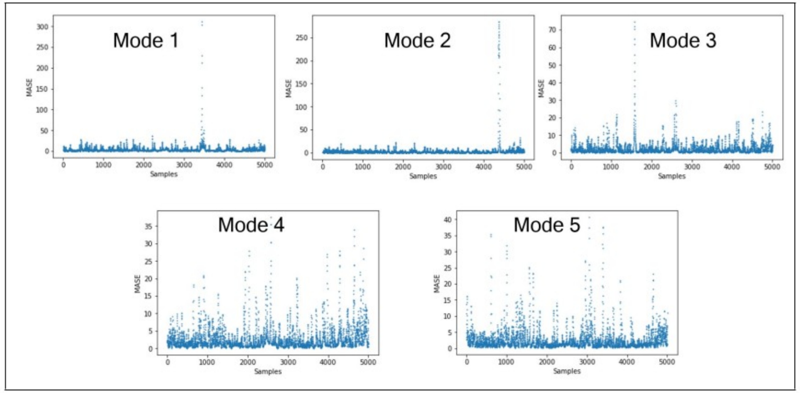

Figure 12. Mean Absolute Scaled Error (MASE) for BiLSTM predictions on all test samples in ISO dataset

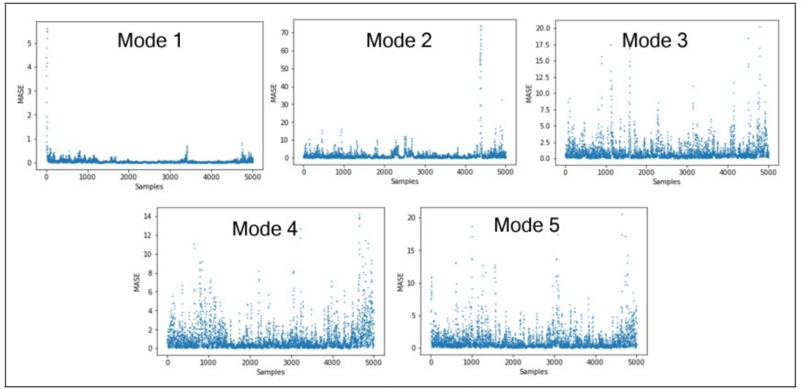

Figure 11. Mean Absolute Scaled Error (MASE) for LSTM predictions on all test samples in ISO dataset

This counter intuitive results may be explained by the fact that BiLSTM has been shown to improve performance for long range statistical correlation, however that is unlikely present (even though it may happen) for signals generated from highly chaotic, non-linear dynamical systems like turbulence.

BiLSTM NNs are likely to over-fit the data.

Using the predicted temporal coefficients, the future time evolution of the flow is computed through equation 7.

The comparison of the expected flow field and the predicted one is shown below.

Dominant modes comprise a significant amount of flow energy, therefore prediction errors in lower modes (such as 4 and 5) tend to less negatively impact the flow field accuracy – than prediction errors in modes1 and 2.

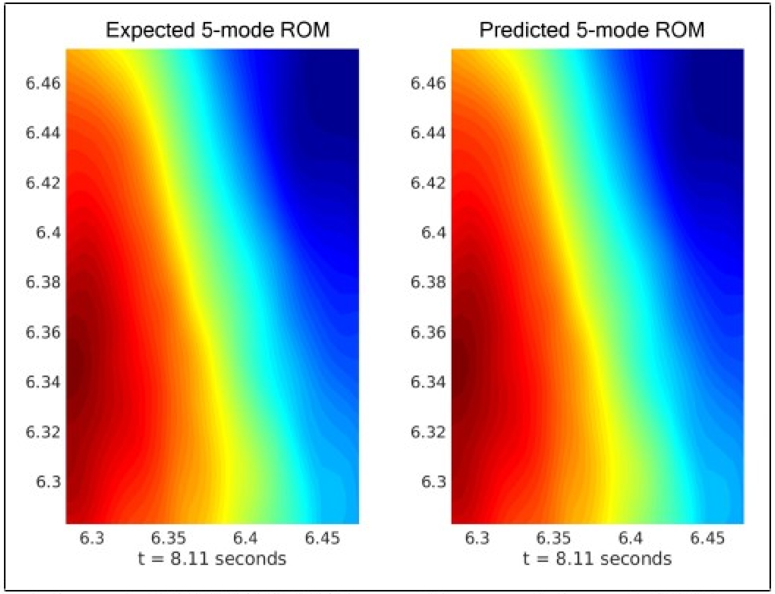

Figure 13. Predicted 5-mode LSTM-ROM flow with actual 5-mode POD reconstructed flow at $t = 8.10$

Magnetohydrodynamic Turbulence

The previous section required m models to be trained (one for each POD mode), a multiple model approach, this requires more memories for embedding NN models on the onboard electronic, being therefore impractical. Another issue is that a POD is a linear combination of eigenvectors and eigenvalues, which do not explicitly account for inter-modal, non-linear interaction seen in turbulence.

Here an alternative strategy will be proposed to account for these issues, the unified model approach. This approach consist in training one NN with samples from all m chosen modes, from all training datasets.

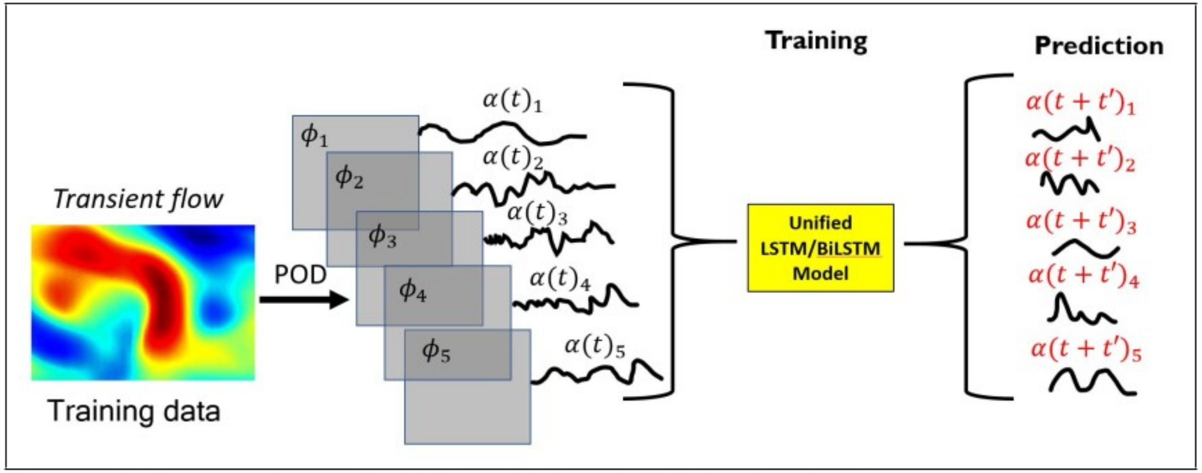

Figure 15. Training of a unified NN model for all POD dominant modes

The LSTM and BiLSTM NNs are trained as before but using the unified model. The results at a randomly chosen sample are shown below. We can see a generally good performance, the unified model may learn statistics common among different POD modes, thus boosting performance.

Moreover, we can once again see that the BiLSTM under-performs the LSTM predictions.

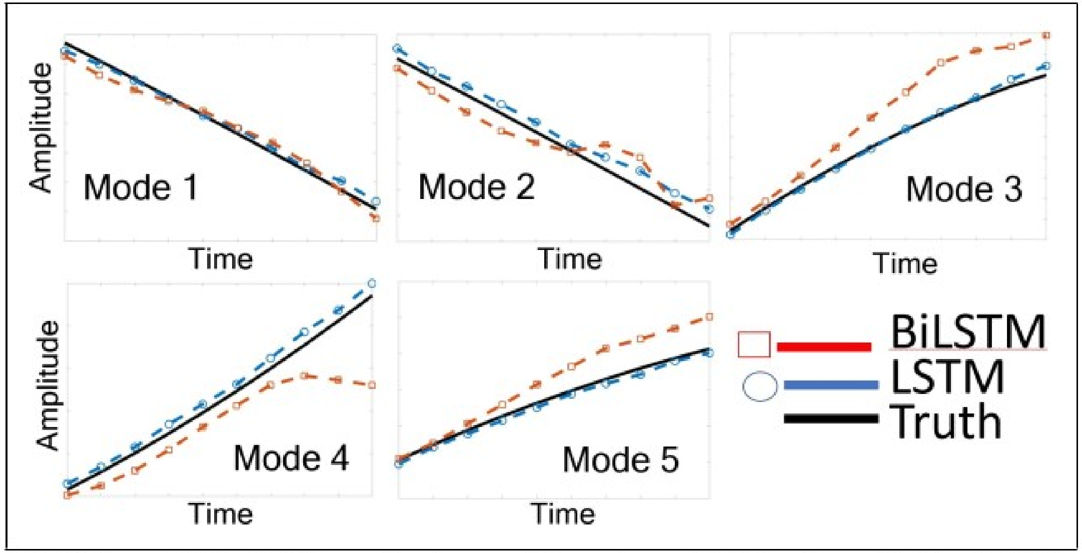

Figure 16. LSTM and BiLSTM predictions of Dominant POD $\alpha(t+t')$ for MHD test data

The MASE is plotted as well.

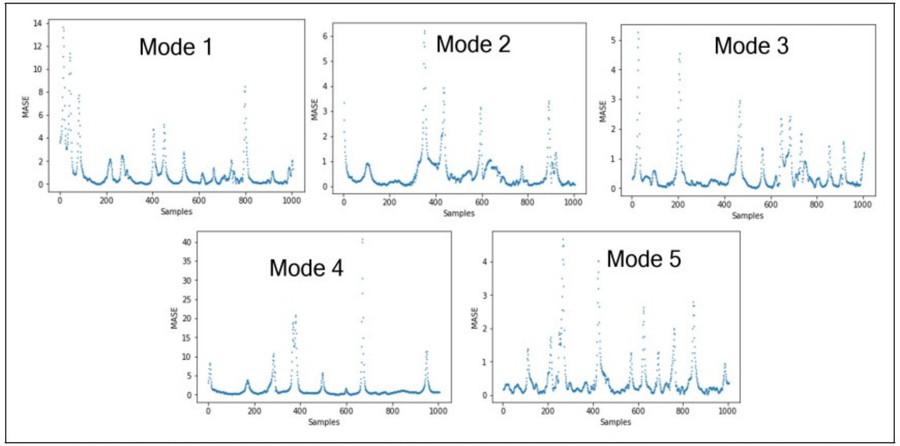

Figure 17. Mean Absolute Scaled Error (MASE) for LSTM predictions on all test samples in MHD dataset with Unified Model

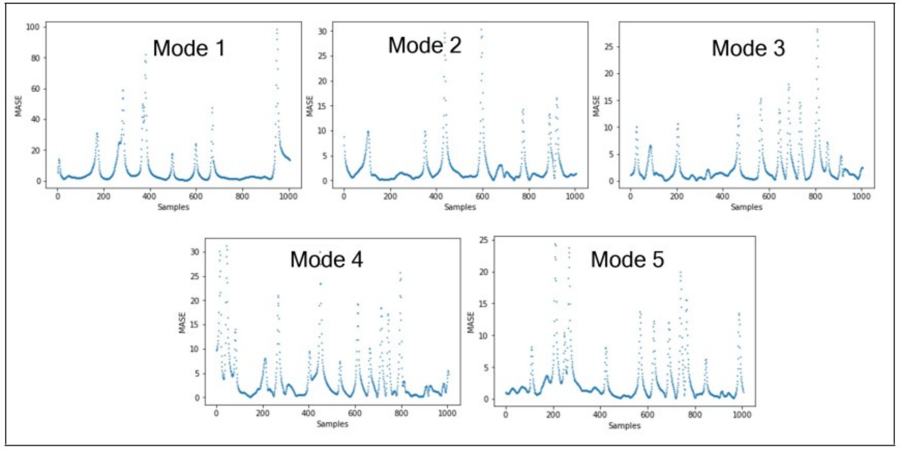

Figure 18. Mean Absolute Scaled Error (MASE) for BiLSTM predictions on all test samples in MHD dataset with Unified Model

Memory Effects and LSTM Model Accuracy

The focus of LSTM is to model sequential data by extracting the correlations between subsequent realizations – LSTM assumes that there is memory in the sequence. However, signals from chaotic dynamical systems tend to have very short correlated events and memory does not persist over a long time.

Now, is it possible to evaluate the suitability of LSTM by quantifying “memory” in a sequential dataset?

A prominent solution is the Hurst exponent, it is a quantitative estimate of the presence or absence of long-term trends in a sequential one-dimensional signal such as time series; it has been used extensively in hydrology, finance, climate sciences, etc. It is derived from rescaled-range analysis which is measure of how variable a time series is for different lengths, using the ratio of its range and the standard deviation.

The Hurst exponent is expressed as k tend to infinity by:

$$\mathbb{E}\left[\frac{R(k)}{S(k)}\right] = Ck^H \tag{8}$$

R(k) – the range of the first k values in the series.

S(k) – the corresponding standard deviation.

E – the expected value the ratio with k being the number of data points in the series being currently Processed.

C – a constant.

H – the Hurst exponent, it lies between 0 and 1.

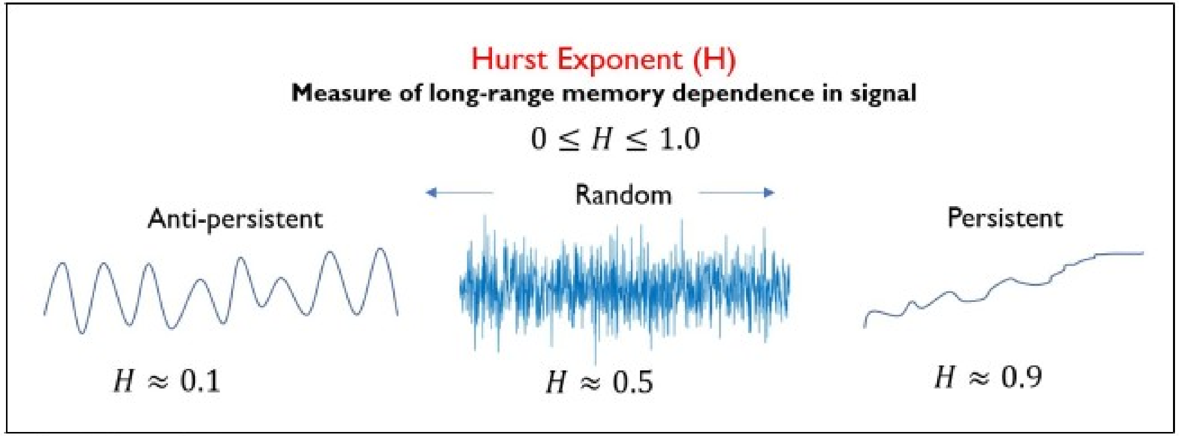

Figure 19. The Hurst Exponent

If H tend to 0, then an increase will most likely be followed by a decrease.

If H tend to 1, then an increase will most likely be followed by an another increase.

If H = 0.5, then it indicates a purely random behavior, and if H ≈ 0.5 it means that there is a lack of memory.

Since the LSTM architecture has the assumption of memory intrinsically built into its algorithm, the LSTM-ROM accuracy is greatly influenced by the interplay between persistence and horizon – respectively the behavior of a sequential series described by the Hurst exponent, and the number of steps ahead we want the model to predict.

To study the influence of the persistence on the accuracy of the prediction horizon, the ISO and MHD datasets are used once again with the LSTM-ROM methodology.

1. H is estimated for the *α(t)* of all the POD modes with non-negligible eigenvalue from the dataset.

2. Modes are chosen accordingly to different H regimes, i.e. persistent, random, and anti-persistent.

3. LSTM models are developed for each of these modes, for a given horizon length.

In the following tables are displayed the accuracy of the predicted model through the mean MASE with additionally the percentage change in the mean MASE. First table is for the ISO dataset and the second one is for the MHD dataset.

For the ISO dataset we notice that for low POD mode rank – and relatively high Hurst exponent – the accuracy is low for short horizon lengths but jumps rapidly for longer horizon lengths.

However for high POD mode rank – and medium Hurst exponent – the accuracy is quite stable with greater horizon lengths.

POD mode rank	Horizon Length (L)					Hurst Exponent *H*
	10	25	50	75	100	
7	1.0767	6.8621 (537 %)	15.0335 (1296 %)	19.4249 (1704 %)	21.4012 (1887 %)	**0.7809**
15	1.3670	6.8924 (404 %)	12.6044 (822 %)	13.9043 (917 %)	15.1012 (1004 %)	**0.7665**
50	1.9457	5.8721 (201 %)	6.8984 (254 %)	7.0691 (263 %)	7.3354 (733 %)	**0.6519**
100	2.3605	4.9828 (111 %)	5.4702 (131 %)	6.2263 (163 %)	5.8782 (149 %)	**0.6526**
400	2.0315	2.3555 (16 %)	2.3861 (17 %)	2.4338 (20 %)	2.4147 (19 %)	**0.5635**
800	2.0037	2.3453 (17 %)	2.3918 (19 %)	2.5052 (25 %)	2.4146 (20 %)	**0.54**

Figure 22. Impact of Persistence and Horizon on LSTM-ROM model accuracy in ISO dataset

For the MHD dataset we observe similar results for low POD mode rank but for the high ones, contrary to the ISO dataset, the prediction is extremely poor regardless of the length of horizon – this being most likely linked to the low Hurst exponent and therefore highly anti-persistent model.

POD mode rank	Horizon Length (L)					Hurst Exponent *H*
	10	25	50	75	100	
7	0.8610	1.8579 (115 %)	10.1191 (1075 %)	20.6342 (2296 %)	30.9824 (3498 %)	**0.8941**
15	0.7816	2.8978 (270 %)	11.1429 (1325 %)	16.0197 (1949 %)	18.0512 (2209 %)	**0.8581**
50	0.6060	2.5394 (319 %)	4.9804 (721 %)	7.9789 (1216 %)	8.2481 (1261 %)	**0.5923**
100	0.9531	1.7591 (84.56 %)	2.7140 (184 %)	2.8359 (197 %)	2.9914 (213 %)	**0.4063**
400	533.207	527.468 (1 %)	757.10 (41%)	642.5658 (20%)	714.1748 (34 %)	**0.17**
800	7397.673	7931.605 (7 %)	10232.75 (38 %)	9902.367 (34 %)	10745.377 (45 %)	**0.1522**

Figure 23. Impact of Persistence and Horizon on LSTM-ROM model accuracy in MHD dataset

Out of these results a few observations are made about the LSTM behavior:

a) Strongly persistent behavior -- shown by low rank modes – tend to be modelled accurately for short horizon.
b) Strongly anti-persistent behavior tend to be modelled poorly regardless of horizon.
c) For any given mode, increase in horizon tends to reduce accuracy -- being less pronounced for strongly anti-persistent modes.
d) For weakly persistent modes, i.e. $H \approx 0.5$, there may be some improvement in accuracy with increase in horizon.

We notice that highly persistent modes tend to be low rank, containing more energy; likewise, weakly-persistent and anti-persistent modes tend to have low energy. Therefore for short horizons accurate modelling of the high energy modes leads to accurate ROMs, even though at longer horizon their prediction are poor.

Conclusion

ROMs for turbulent flows using the LSTM NN have shown great potential to model complex sequential data in multiple domain. From the paper is drawn the conclusion that using the BiLSTM consistently performed worse than the LSTM despite its theoretical formulation intending otherwise, this being likely due to over-fitting data by assuming long range memory.
Better tuning of the hyper parameter could lead to further improvement however it is thought that they would be marginal and the qualitative trends would hold.

The implicit assumption that has been made is that the dominant POD spatial modes are consistent within the same regime. This generally holds for simplified flow fields and geometries, at relatively close related Reynolds numbers, but can be extremely restrictive otherwise.

The effort made in this paper was to demonstrate the capability of LSTMs in modelling non-stationary signals from high-fidelity turbulence. Such LSTM-ROM would be suitable for flow control applications with a narrow class of regimes and a 'physics' insight into the flow is not a necessity, moreover the low computational cost of the trained LSTM-ROMs for inference is convenient for resource-scarce on-board control hardware.