



Hybrid LSTM+CNN architecture for unsteady flow prediction



Koldo Portal-Porras^a, Unai Fernandez-Gamiz^{a,*}, Ekaitz Zulueta^b, Oscar Irigaray^a, Roberto Garcia-Fernandez^{a,c}

^a Nuclear Engineering and Fluid Mechanics Department, University of the Basque Country, UPV/EHU, Nieves Cano 12, Vitoria-Gasteiz, 01006 Araba, Spain

^b System Engineering and Automation Control Department, University of the Basque Country, UPV/EHU, Nieves Cano 12, Vitoria-Gasteiz, 01006 Araba, Spain

^c Sunsundegui S.A., Polígono Ibarrea, s/n, 31800 Altsasu, Navarra, Spain

ARTICLE INFO

Keywords:
Unsteady flow
Neural networks
LSTM
CNN
Computational fluid dynamics

ABSTRACT

Introduction: Data-driven methods are increasingly used for modeling fluid dynamic systems, since traditional numerical methods, such as Computational Fluid Dynamics (CFD), have certain limitations, including the required computational resources and user influence. There are many Deep Learning based methods capable of providing very accurate results for stationary problems. However, the prediction of unsteady flows remains being a challenge, since with the addition of the time component, these methods lose reliability.

Objectives: This paper aims to design a hybrid neural network for unsteady flow prediction, which combines a Long-Short Term Memory (LSTM) and a Convolutional Neural Network (CNN).

Methods: Unsteady-state RANS-based CFD simulations are conducted to obtain data of flows around cylinders. In these simulations different inlet velocities and cylinder diameters are considered, to ensure diversity in the dataset. A hybrid neural network is designed, in which a LSTM predicts the Lift Coefficient for each time step and then, based on those predictions, a CNN predicts the velocity and pressure fields. For training and testing the proposed net the conducted CFD simulations are used.

Results: Even if there is a small mismatch between the ground-truth vortex shedding frequency and the predicted one, the proposed network is able to accurately predict the vortex shedding behind the cylinders, with very low errors throughout the whole studied range.

1. Introduction

Since their introduction, CFD tools have been the most common method for solving fluid dynamic problems, due to their simplicity and capability to provide detailed characteristics of the flow. However, CFD tools have certain limitations, which can be prohibitive in cases where the analyzed system is complex or an accurate modelling of the turbulence is required. The main limitation associated with CFD simulations is the required computational time and resources. In addition to this, there are other aspects that can negatively influence the accuracy of the obtained results, such as the influence of the user when generating the mesh, selecting the models and establishing the conditions of the simulation, and many others. These aspects, added to the exponential growth of Artificial Intelligence (AI) in the last few years and the irruption of Deep Learning (DL) techniques, have led more and more authors using data-driven methods for fluid dynamic problem solving.

Among the different neural network structures, the Convolutional

Neural Network (CNN) is the most popular for predicting flow phenomena, since it provides the possibility of easily working with multi-dimensional data; and as stated by Zhang et al [1], these networks provide the highest generalization capacity. The pioneering work of Guo et al [2] demonstrates that Convolutional Neural Networks (CNNs) are able to predict flow characteristics around simple geometries accurately and extremely fast, reducing computational time by four orders of magnitude in comparison with CFD methods. Based on this work, other authors focused their efforts on improving the predictions obtained using the CNN. For example, Ribeiro et al [3] evaluated the results obtained with different encoder and decoder variants. Portal-Porras et al [4] proposed the addition of a previous stage for pressure and vorticity field prediction, for subsequent velocity field prediction, showing that predicting additional flow features enhances the performance of the CNN. Abucide-Armas et al [5] proposed a data augmentation method for improving the accuracy of the CNN without requiring additional data. Since its initial implementations, CNNs have been proven to be successful for many different fluid dynamics applications, such as, car

* Corresponding author.

E-mail address: unai.fernandez@ehu.eus (U. Fernandez-Gamiz).

Nomenclature	
AI	Artificial Intelligence
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
ConvLSTM	Convolutional LSTM
DL	Deep Learning
LSTM	Long-Short Term Memory
RANS	Reynolds-Averaged Navier-Stokes
ReLU	Rectifier Linear Unit
RNN	Recurrent Neural Network
SDF	Signed Distance Function
SST	Shear Stress Transport
*	Dimensionless variable
'	Variable ranged between 0 and 1
D	Cylinder diameter
C_L	Lift Coefficient
f_s	Vortex shedding frequency
Φ	Normalized magnitude
p	Pressure
P	Order of accuracy (Richardson Extrapolation)
ρ	Density
R	Convergence condition (Richardson Extrapolation)
RE	Estimated value (Richardson Extrapolation)
Re	Reynolds number
St	Strouhal number
σ	Standard deviation
u	Velocity
u_∞	Freestream velocity
μ	Arithmetic mean
ν	Kinematic viscosity
x, y, z	Cartesian components
Z	Zero-level set

aerodynamics [6], airfoils [7], flow control device modelling [8], built environment [9], and many others.

Nevertheless, although CNN is very accurate for steady cases, as shown in the study conducted by Abucide-Armas et al [10], the CNN by itself is not able to accurately predict the temporal component in unsteady cases. In that study, it is shown that even if in the first instants the predictions are accurate, as the predicted time increases the error also increases. For this reason, authors have attempted to use other neural network architectures that are more suitable for predicting the temporal component. For example, Mohan and Gaitonde [11] proposed a Reduced Order Model (ROM) for turbulent flow prediction based on a Long-Short Term Memory (LSTM), which is a especial case of a Recurrent Neural Network (RNN) introduced by Hochreiter and Schmidhuber [12]. Other authors, such as Fan et al [13] and Hou et al [14], added the LSTM layer to the CNN architecture in order to directly predict the flow fields. This type of network has been used for several CFD-related areas, such as aerodynamics [15], chemistry [16] and air pollution [17].

However, in traditional LSTM layers the states are one-dimensional, which means that when working with higher dimensional data, such as velocity or pressure fields, these have to be reshaped, leading to the loss of some flow characteristics. For that reason, Shi et al [18] introduced the Convolutional LSTM (ConvLSTM), which allows working with higher dimensional data, and therefore, retaining spatial information. Mohan et al [19] proposed a convolutional autoencoder with a ConvLSTM to predict three-dimensional flows. Han et al [20] designed a encoder-decoder network with a ConvLSTM layer for unsteady flow prediction under laminar and turbulent regimes, showing good agreements between the results obtained with the neural network and the ground-truth CFD data. Following this study, Han et al [21] developed a similar network structure for unsteady flow prediction with moving geometries.

In the present paper, a hybrid neural network architecture is proposed for predicting the vortex shedding on the wake behind different cylinders at various Reynolds numbers. The proposed network combines a LSTM network, for predicting the Lift Coefficient (C_L) in each time step; and a CNN, for predicting the velocity and pressure fields based on the predictions of the LSTM. Hence, this combination allows performing predictions of unsteady-state flows by means of two simple neural network structures.

The following of the manuscript is divided as follows: Section 2 provides a detailed explanation of the methodology followed to set up and conduct the CFD simulations, prepare the data for the neural network, design and train the network; Section 3 shows the obtained results, with an in-depth analysis of the main sources of error; and Section 4, summarizes the main findings of this study.

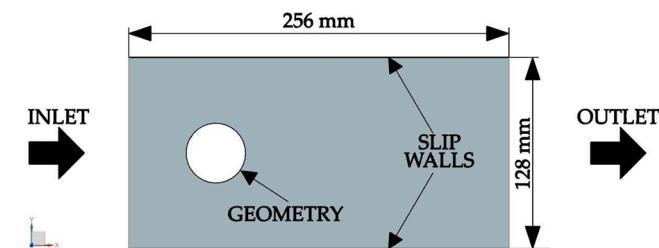


Fig. 1. Numerical domain (not to scale).

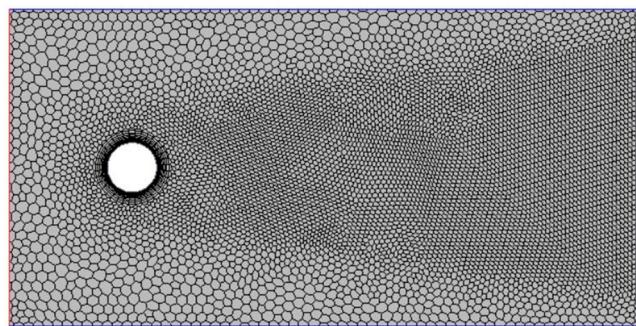


Fig. 2. Example of the mesh generated.

2. Methodology

2.1. CFD setup

With the objective of obtaining data for training and testing the proposed network, several CFD simulations were conducted. In those simulations, a channel is considered, with different geometries and input velocities. For running the simulations Star-CCM+ v2019.1 [22] commercial code was used.

The numerical domain consists of a two-dimensional $128 \text{ mm} \times 256 \text{ mm}$ plate. Left and right sides of the plate are set as inlet and outlet, respectively; and top and bottom sides as walls with slip conditions. On the horizontal symmetry axis, there is a cylinder located at 50 mm from the inlet. Cylinders of different diameters, between 10 mm and 20 mm, are considered. No-slip wall conditions are set on the cylinder. Fig. 1 provides a detailed view of the numerical domain

Table 1

Mesh verification and comparison with experimental data.

Case	u_∞	$D(\text{mm})$	Mesh Resolution			Richardson Extrapolation			Experimental
			Coarse	Medium	Fine	RE	p	R	
5	0.01	3200	0.813	0.871	0.902	0.952	2.178	0.534	0.94
10	0.02	12,800	1.062	1.131	1.158	1.17	1.805	0.391	1.18

and the boundary conditions.

With this domain, an unstructured polygonal mesh is generated, with around 25000 cells. This mesh contains a mesh refinement in the near-geometry region and on the wake behind the geometry, since these areas are considered the most conflictive for carrying out the simulations. Fig. 2 provides an example of the designed mesh.

Regarding the fluid, incompressible turbulent unsteady air is considered. The density (ρ) of the fluid is equal to 1.18415 kg/m^3 , and its dynamic viscosity (μ) is equal to $1.85508 \cdot 10^{-5} \text{ Pa}\cdot\text{s}$. These values are assumed to be constant. For turbulence modelling Menter's [23] RANS-based $k-\omega$ Shear Stress Transport (SST) turbulence model is selected. In many studies, such as in Rajani et al [24], and Rahman et al [25], this model is proven to be appropriate for simulations similar to the ones carried out in this study. Each case was simulated for 1 s, with a time step of 10^{-4} s , which is considered to be small enough for obtaining a good residual convergence and properly capturing the vortex shedding and the changes in the lift coefficient. Upwind scheme [26] was used to discretize the convective terms, ensuring the robustness of the solution; and a second-order temporal discretization was selected. After each time step the instantaneous data required for training and testing the network was extracted.

Different inlet velocities have been considered, between 5 m/s and 10 m/s; which means that the Reynolds number, according to Expression (1) is between 3200 and 12,800. On the contrary, in the second group the input velocity is set at 5 m/s in all cases, so the Reynolds number is 3200 in all cases.

$$\text{Re} = \frac{u_\infty D \cdot \rho}{\mu} \quad (1)$$

where D represents the diameter of the cylinder.

In order to verify sufficient mesh resolution of the meshes, the General Richardson Extrapolation Method [27] is applied to the mean Drag Coefficient (C_D). For this, two different cases are considered, with $\text{Re} = 3200$ and $\text{Re} = 12800$. This method consists of estimating the value of the analyzed parameter when the size of the cells tend to zero. For estimating this value, a minimum of three meshes is required. Therefore, in this case a coarse mesh (of around 15,000 cells), a medium mesh (of around 20,000 cells) and a fine mesh (the previously-explained mesh, around 25,000 cells) are considered. As in this case the mesh refinement is not equal to 2, the procedure detailed by Almohammadi et al. is followed [28]. As summarized in Table 1, the convergence condition (R), which should be between 0 and 1 to ensure a monotonic convergence, is fulfilled, and the estimated values (RE) are close to the ones obtained with the fine mesh. Therefore, the mesh is suitable for the conducted simulations. Additionally, the obtained results are compared with the experimental ones obtained by Roshko et al [29], showing very close values. The same procedure for mesh validation of a similar case is followed in Aramendia et al [30].

2.2. Neural networks

In this study a combination of two different networks is proposed, with the aim of predicting unsteady flow fields. On the one hand, a LSTM is designed to predict the C_L ; and on the other hand, a CNN is designed to predict the velocity and pressure fields, based on the C_L calculated by the LSTM. With both networks the same dataset is considered, where 60 % of the geometries is used for training, 30 % for validation and 10 % for

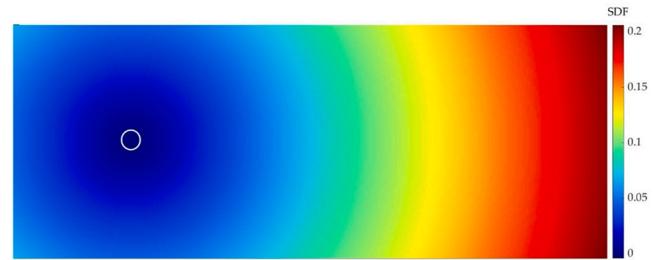


Fig. 3. Example of the SDF of a cylinder, the contour of the geometry is drawn in white.

testing the network. For training and testing the networks MATLAB 2022b [31] commercial code, with its Deep Learning Toolbox [32] was used.

2.2.1. Long short-term memory

Recurrent Neural Networks (RNN) are the most suitable option for sequential data handling. However, due to the simplicity of their structure, basic RNNs are not capable of predicting long-term dependencies, see Yu et al [33]. For this reason, Hochreiter and Schmidhuber [12] introduced the Long Short-Term Memory (LSTM). In this network, a gate is added to remember long-term significant data.

In the present study, a LSTM is used to predict the Lift Coefficient (C_L). In order to simplify and improve the LSTM training process, the data of the C_L has been standardized in accordance with Expression (2).

$$C_{Lstd} = \frac{C_L - \mu_{C_L}}{\sigma_{C_L}} \quad (2)$$

The proposed network consists of a single LSTM cell, with 128 hidden units. For training the network Adam optimizer [34] was selected, with a descending learning rate starting from 0.001. Training was conducted until a good training-loss convergence was achieved.

2.2.2. Convolutional neural network

A CNN is used to predict the velocity and pressure fields. This type of network has been proven to be suitable for solving different fluid dynamic problems in many studies, such as in [3,7,8,35,36].

Four different input layers are considered for this network: a Signed Distance Function (SDF) layer, to represent the geometry; an input velocity layer, only for the cylinders, since in the other geometry-set it is constant; and two layers for the C_L , one for the current instant and another one for the previous instant. The last three layers are constant on the whole domain.

The SDF layer provides the shortest distance between each cell and the contour of the geometry. As demonstrated by Guo et al [2], in comparison with binary representation, SDF layers improve significantly the performance of the CNN. Fig. 3 provides an example of the SDF layer.

To create SDF layers, firstly the contour of the geometry is defined as the zero-level set (Z), following Expression (1). In those cells the SDF value is equal to zero.

$$Z = \{(X, Y) \in R^2 / SDF(x, y) = 0\} \quad (3)$$

where (X, Y) are the coordinates of the cells of the geometry contour.

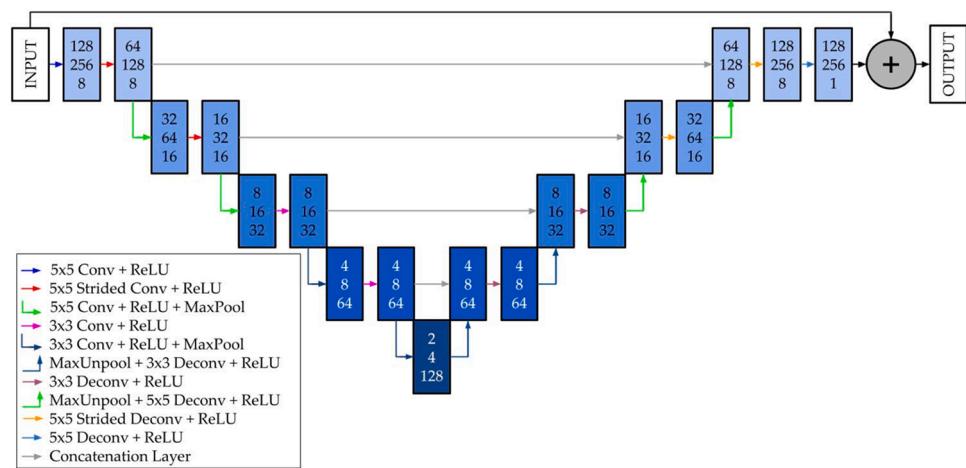


Fig. 4. CNN architecture.

Then, the sign of SDF is defined depending on the location of the cell. If the cell is on the geometry contour, $SDF(x,y) = 0$; if the cell is inside the geometry, $SDF(x,y) < 0$; and if the cell is outside the geometry, $SDF(x,y) > 0$.

Finally, after defining the sign, the SDF value of each cell is calculated by Expression (2).

$$SDF(x,y) = \min_{(X,Y) \in Z} |(x,y) - (X,Y)| \cdot \text{sign} \quad (4)$$

Three different output layers are considered, one for each magnitude to be predicted. To prepare these layers, the velocity and pressure fields are first interpolated to fit into a 128×256 mesh, in order to adapt the CFD data to the CNN. Subsequently, data is made dimensionless following Expressions (3), (4) and (5).

$$u_x^* = \frac{u_x}{u_\infty} \quad (5)$$

$$u_y^* = \frac{u_y}{u_\infty} \quad (6)$$

$$p^* = \frac{p}{\rho \cdot u_\infty^2} \quad (7)$$

where u_x^* , u_y^* and p^* are dimensionless magnitudes.

Finally, dimensionless magnitudes are normalized between 0 and 1 according to Expression (6). This last step is also applied in the input layers, with the purpose of speeding up and improving the training of the network.

$$\Phi_{norm} = \frac{\Phi - \Phi_{min}}{\Phi_{max} - \Phi_{min}} \quad (8)$$

where Φ represents each dimensionless magnitude.

Regarding the architecture, in this study a U-Net architecture [37] is proposed, which is a special case of an encoder-decoder network. In this kind of networks, the input layers are compressed by means of convolutions, obtaining a Latent Geometry Representation (LGR); and then, the LGR is expanded by means of deconvolutions, getting the desired output. This network has been successfully applied for flow field prediction by Portal-Porras et al [4,8].

The proposed network consists of four encoding/decoding blocks. Each encoding block contains two convolutional layers, the first one followed by a ReLU (Rectifier Linear Unit) activation layer, and the second one followed by a ReLU layer and a 2-by-2 Max Pooling layer. In the first two blocks, the kernel size in the convolutions is equal to 5. Additionally, strided convolutions are performed on those blocks, in order to reduce the handled data. Conversely, in the last two blocks the kernel size is equal to 3. The number of filters is set to 8 in the first block,

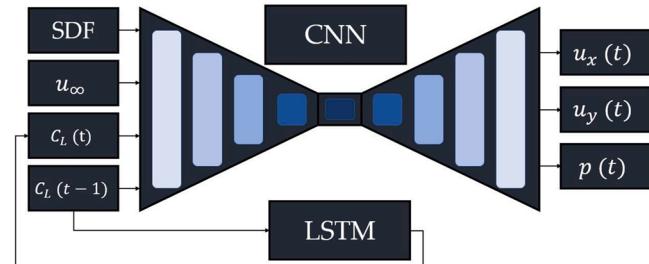


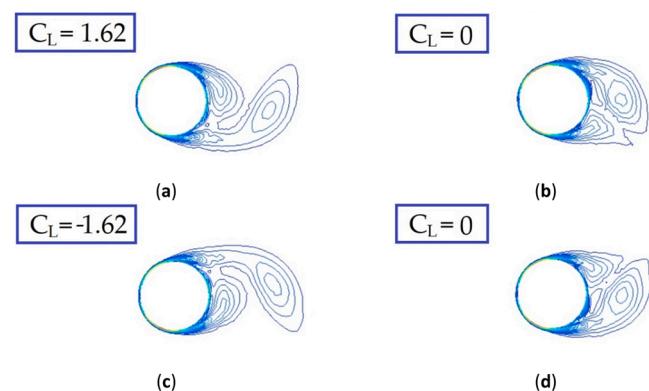
Fig. 5. Hybrid LSTM+CNN network.

and is doubled in each of the following blocks. The decoding blocks, by means of deconvolutions and Unpooling layers, perform the reverse process of their symmetrical blocks of the encoding phase. Encoding and decoding blocks are connected by concatenation layers. Fig. 4 provides a schematic view of the proposed CNN.

With regards to the network training progress, Adam optimizer [34] is selected, with a batch size of 64, a weight decay of 0.005 and a learning rate that starts at 0.001 and decreases as training progresses. The validation is performed after each epoch. For training the network 200 time-steps of each case are considered, after the flow is fully developed. It is estimated that this number of time-steps is enough to capture several vortex shedding periods.

2.2.3. Hybrid network

In the present study a hybrid network is proposed, which combines

Fig. 6. Vorticity contour lines in four different instants for $Re = 12800$. (a) $t = 0.25T$; (b) $t = 0.5T$; (c) $t = 0.75T$; (d) $t = T$.

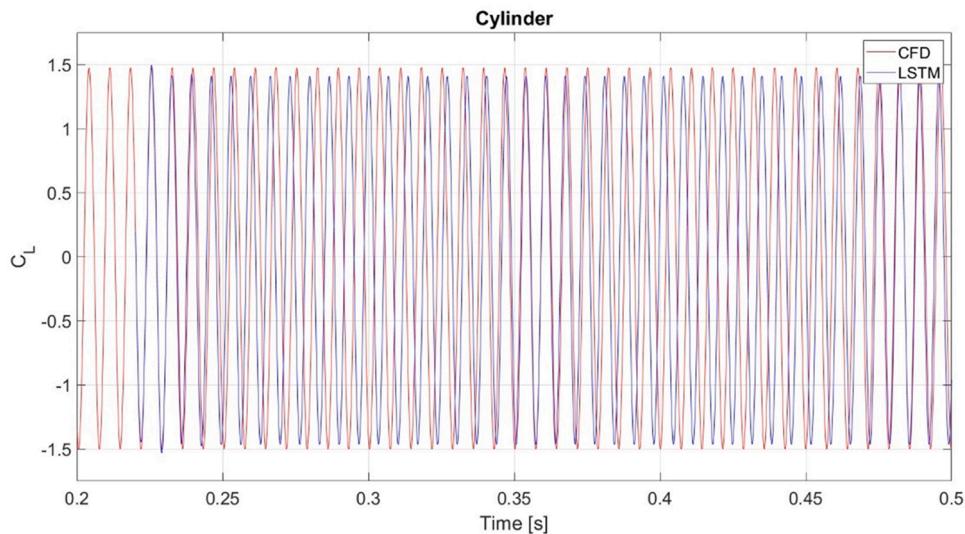


Fig. 7. Comparison of the C_L obtained with both CFD and LSTM for a sample of the test-set.

both the LSTM and CNN networks. The LSTM predicts the lift coefficient in each time step. Based on the predictions of the LSTM, the CNN predicts the velocity and pressure fields. As demonstrated by Portal-Porras et al [4], including a previous stage for predicting characteristics of the flow improves the final predictions. Hence, the predictions of the velocity and pressure fields should improve taking the C_L as input of the CNN. Fig. 5 provides a schematic view of the proposed hybrid network.

3. Results

3.1. Vortex shedding

When fluid flows across a bluff body, vortices are generated on the back of the body, and detach periodically from one side of the body to the other, leading to the von Karman vortex street. When analyzing vortex shedding the Lift Coefficient (C_L) is a very relevant parameter, since, due to the behavior of the vortices, the C_L quantifies the vortex shedding frequency (f_s). Fig. 6 shows the vorticity contour and instantaneous C_L at four different moments of a single vortex shedding period for the case $Re = 12800$. In this Figure the generation, convection and diffusion of the vortexes can be clearly observed.

For oscillating flow analysis, such as vortex shedding, the Strouhal number (St) is a commonly used parameter. The Strouhal number is a dimensionless number, which represents the ratio between the inertial forces caused by the unsteadiness of the flow or the local acceleration, and the inertial forces caused by the changes in velocity from one point of the flow field to another. St is calculated according to Expression (9).

$$St = \frac{f_s \cdot D}{u_\infty} \quad (9)$$

3.2. Lift coefficient prediction

As mentioned, the recurrent part of the proposed network consists of

predicting the instantaneous C_L based on the previous values. As the predictions of the CNN are based on the predictions of the LSTM, a correct C_L prediction is essential. Fig. 7 shows a comparison of the ground-truth C_L obtained by means of CFD and the ones obtained with the LSTM.

The results show that the LSTM is able to correctly predict the C_L amplitude. However, a small difference between vortex-shedding frequencies obtained by CFD and LSTM is noticeable.

In order to obtain a quantitative view of the obtained predictions, the vortex shedding frequency (f_s) C_L amplitude and St is evaluated. For this, the cases corresponding to the test-set are considered. These values are provided in Table 2.

The results show that the relative errors are low for all the analyzed cases. The LSTM performs uniform predictions for amplitude among cases, underpredicting cases with high amplitude and overpredicting those with low amplitude. Regarding the vortex shedding frequency and the Strouhal number, which are directly related, the LSTM predicts lower frequencies for low-Re cases, and higher frequencies for cases with higher Re . Nonetheless, the errors can be attributed to the fact of considering different input velocities, since the diversity among the samples can make the training process more complicated.

3.3. Velocity and pressure field prediction

Velocity and pressure fields obtained with the CNN are compared with the ground-truth values obtained by means of CFD. For this comparison, four different instants are considered, with the vortex moving upward, the vortex at the top, the vortex moving downward and the vortex at the bottom. Fig. 8 provides the mentioned fields at each moment.

The results show that the CNN is able to accurately predict the velocity and pressure fields, modeling the vortex shedding behind the geometries. In all the cases, the larger errors appear on the contour of the

Table 2

Comparison of the C_L and St obtained with both CFD and LSTM.

Case	CFD			LSTM			Relative error			
	Re	u_∞ [m/s]	D [mm]	C_L amplitude	f_s [Hz]	St	C_L amplitude	f_s [Hz]	St	C_L amplitude
4400	5	14	1.6659	77.2	0.216	1.4952	72.9	0.204	10.2 %	5.5 %
4400	7	10	1.3676	150.5	0.215	1.4108	139.3	0.199	3.16 %	7.4 %
6200	8	12	1.3685	137.8	0.207	1.41	144.2	0.216	3.03 %	4.3 %
9200	9	16	1.4767	113.1	0.201	1.4105	119.7	0.213	4.48 %	5.9 %

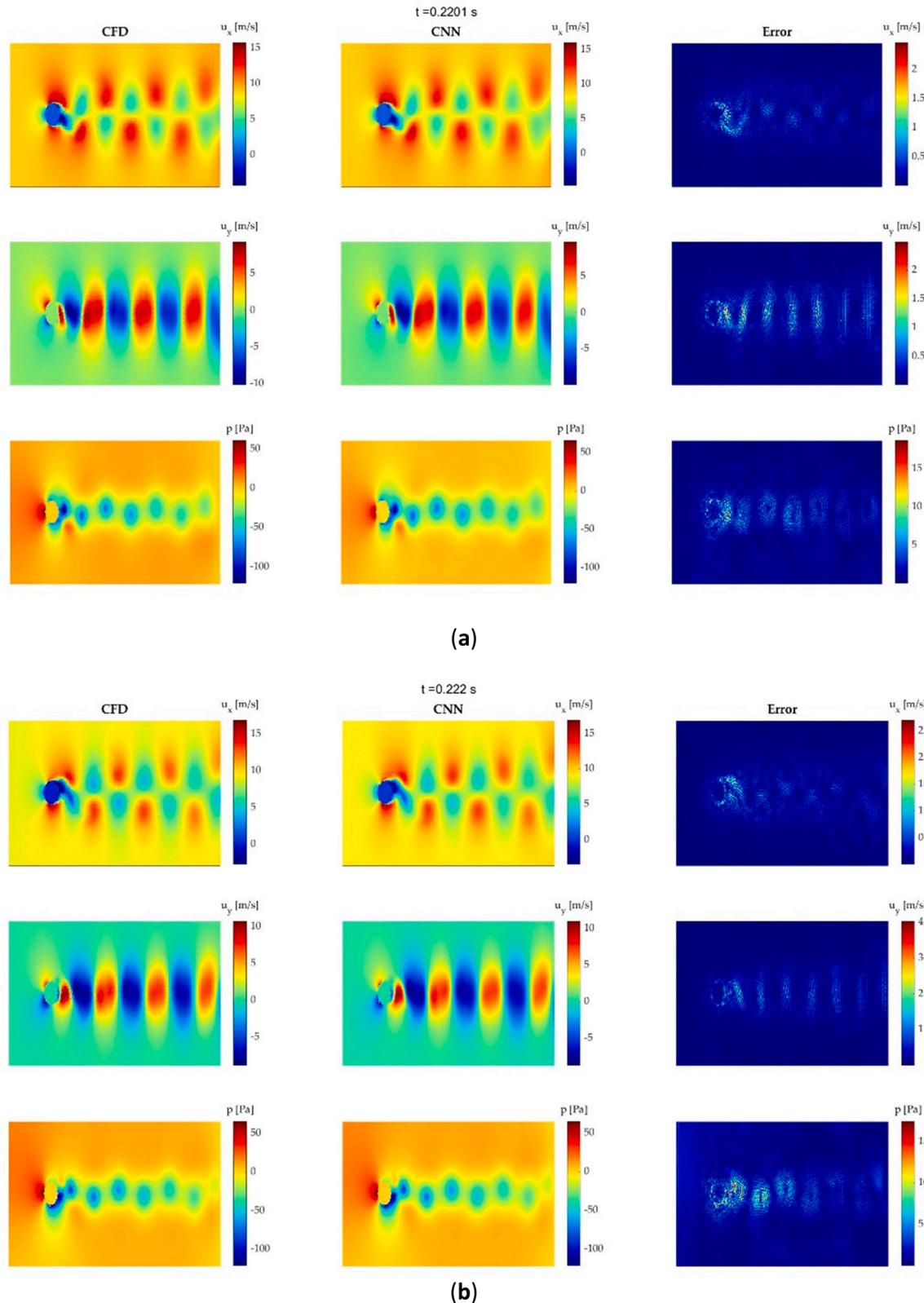


Fig. 8. Comparison of the velocity and pressure fields obtained with both CFD and CNN+LSTM in four different instants around a cylinder. (a) Upward vortex; (b) Vortex at top; (c) Downward vortex; (d) Vortex at bottom.,,

geometry and on the wake behind the cylinder. These areas are assumed to be the most troubleshooting ones. The boundary layer is on the contour of the geometry. Because of the large velocity changes the flow

experiences in that area, the network struggles to make accurate predictions. With respect to the wake behind the cylinder, this zone is the one in which most differences appear between the cases analyzed. For

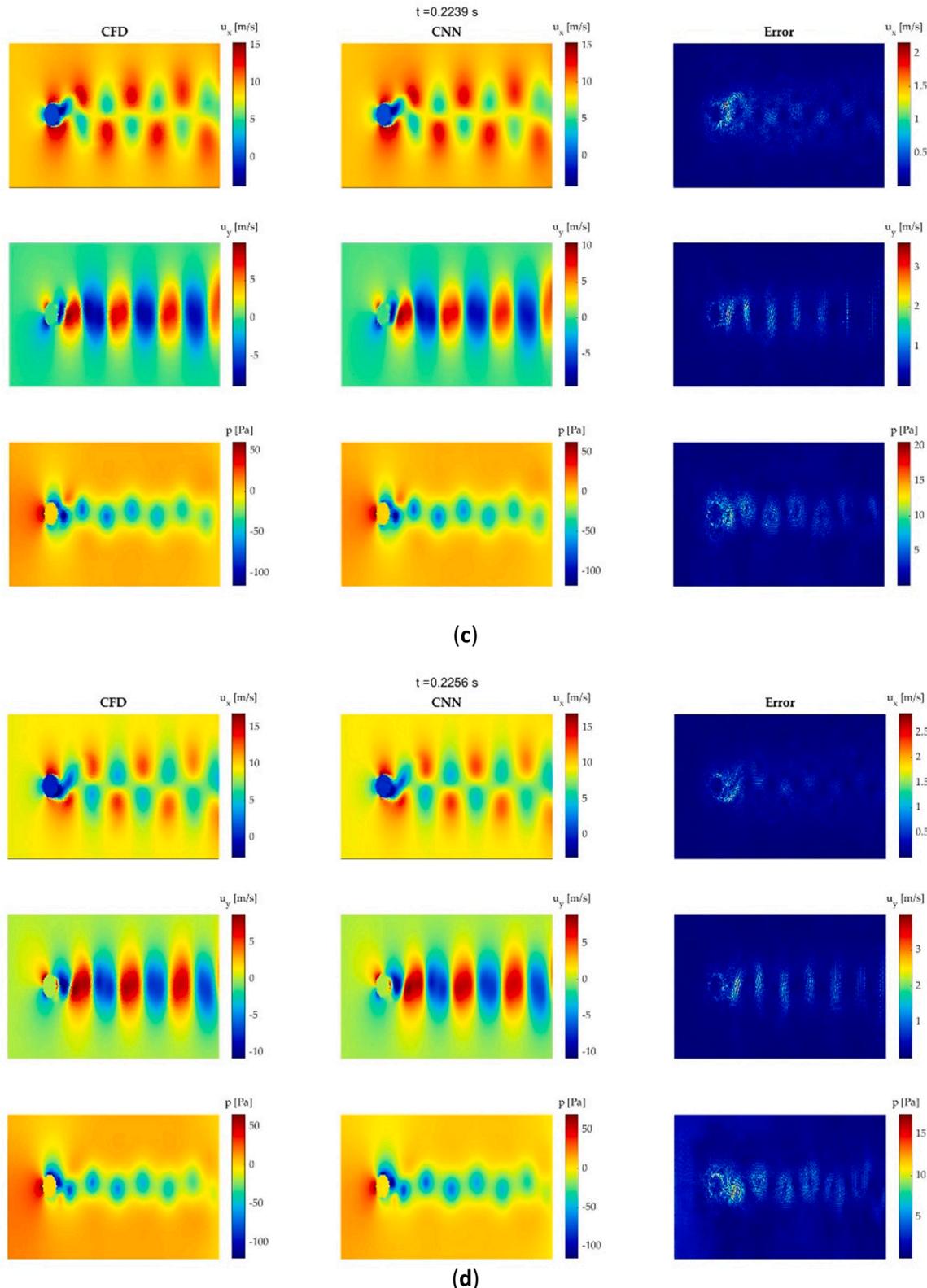


Fig. 8. (continued).

this reason, it is the area in which the network has more difficulties to properly learn all the characteristics of the flow. However, these errors are very low in all the analyzed range.

Nevertheless, as the results obtained with the LSTM are used as input in the CNN and the LSTM predicts the Strouhal number with a small error, as shown in Fig. 7, there are some instances where there is a

mismatch between the CFD and CNN results, especially as the predictions become longer in time. Fig. 9 shows an instant where the C_L obtained with CFD and the one obtained with the neural network mismatch, causing the vortices to be generated in opposite directions. In the analyzed case, the ground-truth CFD C_L is in its maximum value, while the C_L predicted by the LSTM is in its minimum value. However,

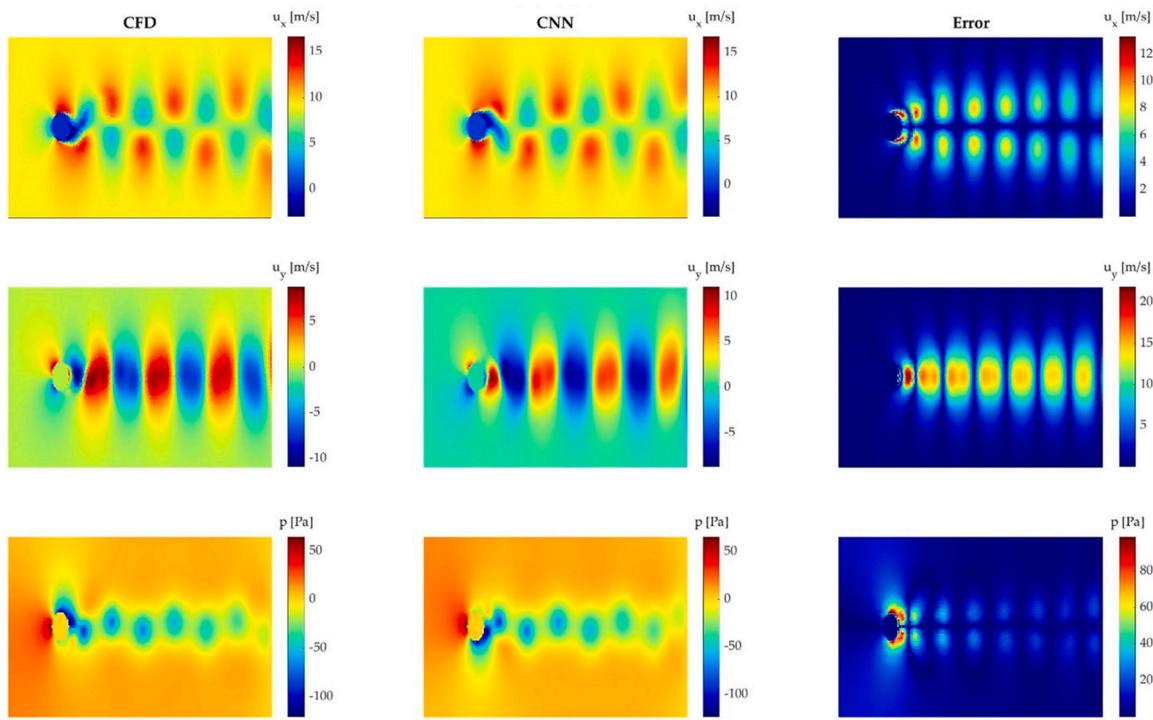


Fig. 9. Velocity and pressure fields around a cylinder at an instant with phase mismatch between CFD and CNN+LSTM results.

despite the difference in frequency, the evolution of the vortices is the same throughout the whole analyzed time.

3.4. Error analysis

In order to obtain a detailed view of the error along a vortex shedding period, the error in a period where the C_L obtained with CFD and LSTM match is analyzed for all the test-set cases. Fig. 10 provides the mean error evolution of all the analyzed magnitudes throughout a vortex shedding period for all the cases of the test-set.

The mean error evolution of velocity fields show that the larger errors appear when the vortex is located at the top or at the bottom. In the case of pressure, the results are more diverse. In this case the error does not follow a clear trend with the position of the vortex, so it depends on the sample. In all the analyzed cases, the mean errors show a direct correlation with the error in C_L amplitude explained in Section 3.2. As CNN predictions are made taking the LSTM predictions as input, larger errors in C_L lead to larger errors in the predicted velocity and pressure fields.

3.5. Performance comparison

The main advantage of data-driven methods over traditional numerical methods is the speed with which results can be obtained. For that reason, the computational time required by each method is compared. Table 3 shows the computational time required by each method to perform a single simulation using a single Intel Xeon Gold 5120 CPU core.

As shown in the comparison, CFD methods require an average of around 121 h to perform a single simulation of 1 s, while the proposed LSTM+CNN network only requires about 37 min to predict all the time-steps of the same simulation. Therefore, the proposed network is 192.4 times faster than CFD. Around 30 min were required to train the LSTM, and around 12 h to train the CNN.

4. Conclusions

Introduction: Traditional numerical methods for fluid dynamics system analysis have some limitations, such as the required computational resources and the influence of the user for defining the case. For that reason, data-driven methods, especially Deep Learning-based methods, are becoming increasingly popular. This kind of methods have been proven to be very effective and accurate for steady-state problems. However, the prediction of unsteady flows remains being a challenge, since with the addition of the time component, these methods lose reliability.

Objectives: The objective of this study is to propose a combination of neural networks for the prediction of unsteady flows. The proposed network combines a LSTM network, for predicting the C_L in each time step; and a CNN, for predicting the velocity and pressure fields based on the predictions of the LSTM.

Methods: In order to collect data of unsteady flows, different vortex shedding cases are simulated. Unsteady-state RANS-based CFD simulations of different flows around cylinders are conducted. In those simulations, various inlet velocities and cylinder diameters are considered to ensure a diverse dataset. The collected CFD simulation data is used for training and testing the proposed neural network.

Results: The LSTM is able to accurately predict the C_L throughout the whole analyzed range of time. The CNN, which takes as input the C_L predicted by the LSTM, is also able to accurately predict the velocity and pressure fields. Nevertheless, the LSTM predicts the C_L with a slight error on the frequency. This leads to a mismatch between the position of the vortices predicted by the CNN and the ones obtained by CFD at advanced time-steps. However, despite the error on the vortex shedding frequency, the CNN is able to correctly predict the evolution of the vortices.

The error analysis shows that the most troubleshooting areas are the contour of the cylinder and the wake behind the cylinder, where the vortices appear. For velocity fields the larger mean errors appear when the vortex is located at the top or at the bottom, while for pressure no clear trend can be observed. The error in the CNN predictions is directly related to the error in the C_L predictions of the LSTM.

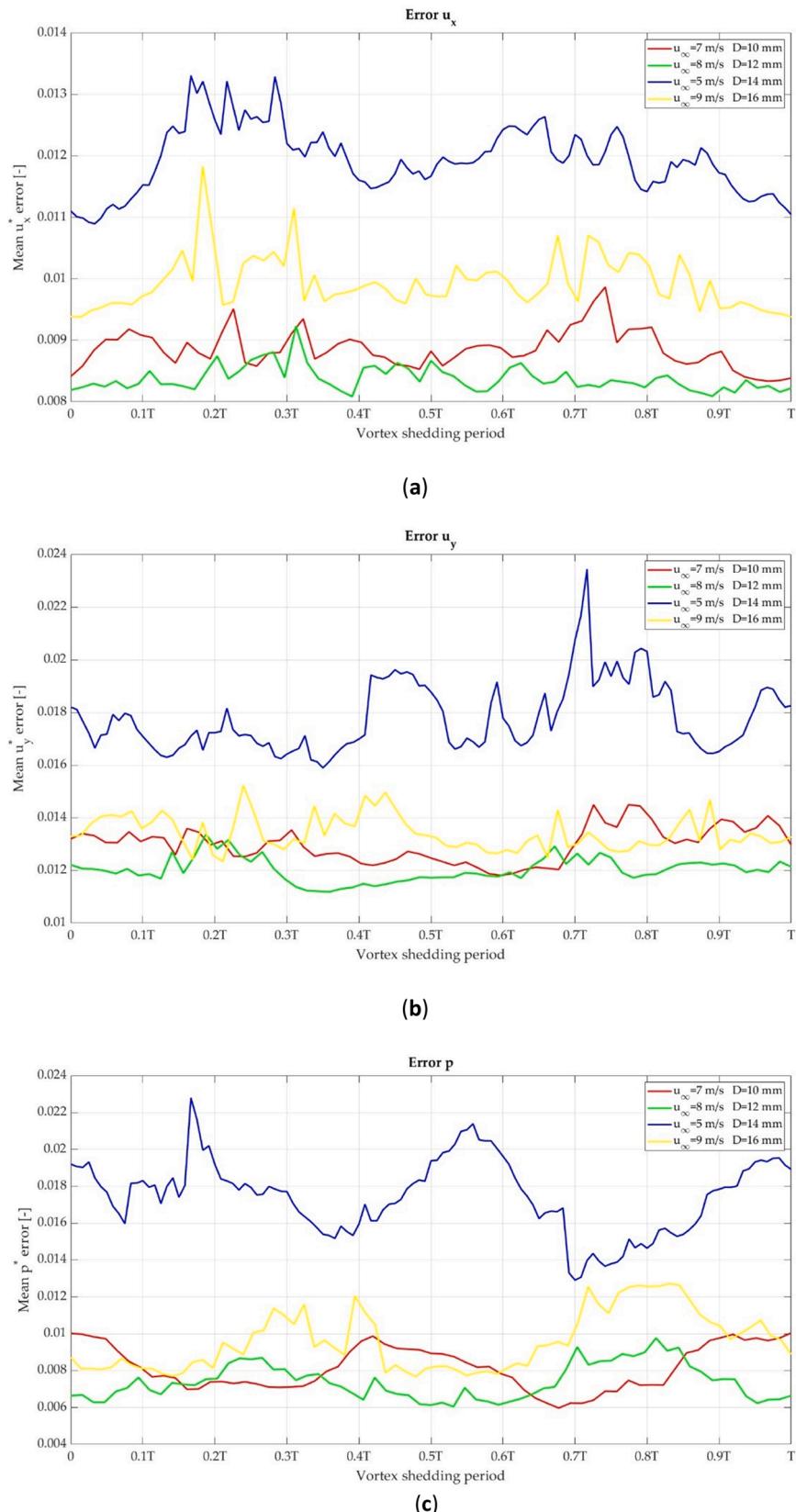


Fig. 10. Mean error of the dimensionless magnitudes during a vortex shedding period. (a) u_x^* error; (b) u_y^* error; (c) p^* error.

Table 3

Computational time required by each method to obtain the predictions of the lift coefficient, velocity and pressure fields.

Method	Time [s]	Speedup
CFD	435,543.74 s	-
LSTM+CNN	2263.96 s	192.4

With regards to the computational resources, the proposed model clearly outperforms the CFD simulations, reducing 192.4 times the computational time required.

Funding

The authors are thankful to the Government of the Basque Country for the financial support of ELKARTEK21/10 KK-2021/00014 and ELKARTEK20/78 KK-2020/00114 research programs, respectively.

CRediT authorship contribution statement

Conceptualization, K.P.-P. and U.F.-G.; Methodology, E.Z.; Software, E.Z. and R.G-F.; Validation, K.P.-P. and U.F.G.; Formal analysis, U.F-G. and O.I.; Investigation, K.P.-P.; Resources, U.F.-G.; Data curation, K.P.-P.; Writing – original draft, K.P.-P.; Writing – review & editing, E.Z. and O.I.; Visualization, E.Z. and R.G-F.; Supervision, U.F.-G.; Project administration, E.Z. and R.G-F.; Funding acquisition, U.F.-G. All authors have read and agreed to the published version of the manuscript.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

The authors are grateful for the support provided by the SGiker of UPV/EHU.

References

- Zhang, Y.; Sung, W.-J.; Mavris, D., Application of convolutional neural network to predict airfoil lift coefficient, ArXiv171210082 Cs Stat, 2018.
- Guo, X.; Li, W.; Iorio, F., Convolutional neural networks for steady flow approximation, in: Proceedings of the Twenty Second ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, San Francisco California USA, August 13–17, 2016, 481–490.
- Ribeiro, M.D.; Rehman, A.; Ahmed, S.; Dengel, A., DeepCFD: efficient steady-state laminar flow approximation with deep convolutional neural networks, ArXiv200408826 Phys., 2020.
- K. Portal-Porras, U. Fernandez-Gamiz, A. Ugarte-Anero, E. Zulueta, A. Zulueta, Alternative artificial neural network structures for turbulent flow velocity field prediction, Mathematics 9 (2021) 1939, <https://doi.org/10.3390/math9161939>.
- A. Abucide-Armas, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, A. Teso-Fz-Betöño, A data augmentation-based technique for deep learning applied to CFD simulations, Mathematics 9 (2021) 1843, <https://doi.org/10.3390/math9161843>.
- S.J. Jacob, M. Mrosek, C. Othmer, H. Köstler, Deep learning for real-time aerodynamic evaluations of arbitrary vehicle shapes, SAE Int. J. Passeng. Veh. Syst. 15 (2022) 15–15-02-0006, <https://doi.org/10.4271/15-15-02-0006>.
- N. Thuerey, K. Weißenow, L. Prantl, X. Hu, Deep learning methods for reynolds-averaged navier–stokes simulations of airfoil flows, AIAA J. 58 (2020) 25–36, <https://doi.org/10.2514/1.J058291>.
- K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, A. Ballesteros-Coll, A. Zulueta, CNN-based flow control device modelling on aerodynamic airfoils, Sci. Rep. 12 (2022) 8205, <https://doi.org/10.1038/s41598-022-12157-w>.
- G. Calzolari, W. Liu, Deep learning to replace, improve, or aid CFD analysis in built environment applications: a review, Build. Environ. 206 (2021), 108315, <https://doi.org/10.1016/j.buildenv.2021.108315>.
- A. Abucide-Armas, K. Portal-Porras, U. Fernandez-Gamiz, E. Zulueta, A. Teso-Fz-Betöño, Convolutional neural network predictions for unsteady Reynolds-averaged navier–stokes-based numerical simulations, J. Mar. Sci. Eng. 11 (2023) 239, <https://doi.org/10.3390/jmse11020239>.
- Mohan, A.T.; Gaitonde, D.V., A deep learning based approach to reduced order modeling for turbulent flow control using LSTM neural networks, ArXiv180409269 Phys., 2018.
- S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (1997) 1735–1780, <https://doi.org/10.1162/neco.1997.9.8.1735>.
- Fan, S.; Fei, J.; Guo, X.-W.; Yang, C.; Revell, A., CNN+LSTM accelerated turbulent flow simulation with link-wise artificial compressibility method, in: Proceedings of the Fiftyifth International Conference on Parallel Processing, ACM, Lemont IL USA, August 9, 2021, 2021, 1–10.
- Y. Hou, H. Li, H. Chen, W. Wei, J. Wang, Y. Huang, A novel deep U-Net-LSTM framework for time-sequenced hydrodynamics prediction of the SUBOFF AFF-8, Eng. Appl. Comput. Fluid Mech. 16 (2022) 630–645, <https://doi.org/10.1080/19942060.2022.2030802>.
- T. Li, T. Wu, Z. Liu, Nonlinear unsteady bridge aerodynamics: reduced-order modeling based on deep LSTM networks, J. Wind Eng. Ind. Aerodyn. 198 (2020), 104116, <https://doi.org/10.1016/j.jweia.2020.104116>.
- H. Kim, M. Park, C.W. Kim, D. Shin, Source localization for hazardous material release in an outdoor chemical plant via a combination of LSTM-RNN and CFD simulation, Comput. Chem. Eng. 125 (2019) 476–489, <https://doi.org/10.1016/j.compchemeng.2019.03.012>.
- Quilodrán-Casas, C.; Arcucci, R.; Mottet, L.; Guo, Y., C. Pain, Adversarial Autoencoders and Adversarial LSTM for Improved Forecasts of Urban Air Pollution Simulations, 2021.
- Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.; WOO, W., Convolutional LSTM network: a machine learning approach for precipitation nowcasting, in: Proceedings of the Advances in Neural Information Processing Systems, Curran Associates, Inc., 28, 2015.
- Mohan, A.; Daniel, D.; Chertkov, M.; Livescu, D., Compressed convolutional LSTM: an efficient deep learning framework to model high fidelity 3D turbulence, ArXiv190300033 Nlin Physicsphysics, 2019.
- R. Han, Y. Wang, Y. Zhang, G. Chen, A new prediction method of unsteady wake flow by the hybrid deep neural network, Phys. Fluids 31 (2019), 127101, <https://doi.org/10.1063/1.5127247>.
- R.-K. Han, Z. Zhang, Y.-X. Wang, Z.-Y. Liu, Y. Zhang, G. Chen, Hybrid deep neural network based prediction method for unsteady flows with moving boundary, Acta Mech. Sin. 37 (2021) 1557–1566, <https://doi.org/10.1007/s10409-021-01129-4>.
- STAR-CCM+ V2019.1. <https://www.plm.automation.siemens.com/>. (Accessed 2 June 2020).
- Menter, F., Zonal two equation K-w turbulence models for aerodynamic flows, in: Proceedings of the Twenty Third Fluid Dynamics, Plasmadynamics, and Lasers Conference; American Institute of Aeronautics and Astronautics, Orlando, FL, U.S.A., July 6, 1993, 1993.
- B.N. Rajani, A. Kandasamy, S. Majumdar, Numerical simulation of laminar flow past a circular cylinder, Appl. Math. Model. 33 (2009) 1228–1247, <https://doi.org/10.1016/j.apm.2008.01.017>.
- M.M. Rahman, Md.M. Karim, M.A. Alim, Numerical investigation of unsteady flow past a circular cylinder using 2-D finite volume method, J. Nav. Archit. Mar. Eng. 4 (1970) 27–42, <https://doi.org/10.3329/jname.v4i1.914>.
- S. Osher, S. Chakravarthy, Upwind schemes and boundary conditions with applications to Euler equations in general geometries, J. Comput. Phys. 50 (1983) 447–481, [https://doi.org/10.1016/0021-9991\(83\)90106-7](https://doi.org/10.1016/0021-9991(83)90106-7).
- Richardson, L.F.; Gaunt, J.A., The deferred approach to the limit, Philos. Trans. R. Soc. Lond. Ser. Contain. Pap. Math. Phys. Charact., 1927, 226, 299–361. doi: 10.1098/rsta.1927.0008.
- K.M. Almomhamdi, D.B. Ingham, L. Ma, M. Pourkashan, Computational fluid dynamics (CFD) mesh independency techniques for a straight blade vertical axis wind turbine, Energy 58 (2013) 483–493, <https://doi.org/10.1016/j.energy.2013.06.012>.
- Roshko, A., Vortex shedding from circular cylinder at low Reynolds number, in: Proceedings of the Cambridge University Press, Cambridge, UK, 1954.
- I. Aramendia, U. Fernandez-Gamiz, E. Zulueta Guerrero, J. Lopez-Gude, J. Sancho, Power control optimization of an underwater piezoelectric energy harvester, Appl. Sci. 8 (2018) 389, <https://doi.org/10.3390/app8030389>.
- MATLAB .<https://es.mathworks.com/products/matlab.html>. (Accessed 9 June 2021).
- Deep Learning Toolbox. <https://es.mathworks.com/products/deep-learning.html>. (Accessed 3 July 2021).
- Y. Yu, X. Si, C. Hu, J. Zhang, A review of recurrent neural networks: LSTM cells and network architectures, Neural Comput. 31 (2019) 1235–1270, https://doi.org/10.1162/neco_a_01199.
- Kingma, D.P.; Ba, J., Adam: a method for stochastic optimization, ArXiv14126980 Cs, 2017.
- L. Liu, A.K. Padthe, P.P. Friedmann, Computational study of microflaps with application to vibration reduction in helicopter rotors, AIAA J. 49 (2011) 1450–1465, <https://doi.org/10.2514/1.J050829>.
- L. Deng, Y. Wang, Y. Liu, F. Wang, S. Li, J. Liu, A CNN-based vortex identification method, J. Vis. 22 (2019) 65–78, <https://doi.org/10.1007/s12650-018-0523-1>.
- Ronneberger, O.; Fischer, P.; Brox, T. U-Net: convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W.M. Wells, A.F.

Frangi (Eds.), Proceedings of the Medical Image Computing and Computer-Assisted

Intervention – MICCAI, 2015, 2015, Springer International Publishing, Cham,
234–241.