




Emulating the estuarine morphology evolution using a deep convolutional neural network emulator based on hydrodynamic results of a numerical model

William Weber de Melo ^{a,*}, J. L. S. Pinho ^a and Isabel Iglesias ^b

^a Centre of Territory, Environment and Construction (CTAC), School of Engineering, University of Minho, Campus de Azurém, Guimarães 4800-058, Portugal

^b Interdisciplinary Centre of Marine and Environmental Research (CIIMAR/CIIMAR), University of Porto, Terminal de Cruzeiros do Porto de Leixões, Av. General Norton de Matos, Matosinhos 4450-208, Portugal

*Corresponding author. E-mail: id9257@alunos.uminho.pt

 WW, 0000-0002-3514-4229; JLS, 0000-0003-2070-8009; II, 0000-0002-4900-135X

ABSTRACT

Coastal and estuarine areas present remarkable environmental values, being key zones for the development of many human activities such as tourism, industry, fishing, and other ecosystem services. To promote the sustainable use of these services, effectively managing these areas and their water and sediment resources for present and future conditions is of utmost importance to implement operational forecast platforms using real-time data and numerical models. These platforms are commonly based on numerical modelling suites, which can simulate hydro-morphodynamic patterns with considerable accuracy. However, in many cases, considering the high spatial resolution models that are necessary to develop operational forecast platforms, a high computing capacity is also required, namely for data processing and storage. This work proposes the use of artificial intelligence (AI) models to emulate morphodynamic numerical model results, allowing us to optimize the use of computational resources. A convolutional neural network was implemented, demonstrating its capacity in reproducing the erosion and sedimentation patterns, resembling the numerical model results. The obtained root mean squared error was 0.59 cm, and 74.5 years of morphological evolution was emulated in less than 5 s. The viability of surrogating numerical models by AI techniques to forecast the morphological evolution of estuarine regions was clearly demonstrated.

Key words: artificial intelligence, convolutional neural networks, Delft3D, hydro-morphodynamics, numerical model emulator, TensorFlow

HIGHLIGHTS

- The application of convolutional neural network (CNN) for the development of an estuarine morphodynamic emulator is still rare.
- Delft3D hydrodynamic results processed in MATLAB for generating AI model inputs datasets.
- Python framework for hybrid use of Delft3D and TensorFlow is selected for hydro-morphodynamic models.
- The assessment of CNN hyperparameter for a morphodynamic problem is an area of focus for future research.
- A comparison between an emulator and a numerical model can be observed in sedimentation and erosion results.

INTRODUCTION

Understanding the morphological evolution of coastal and estuarine regions is essential to promote the sustainable use of its natural resources and protect populations and ecosystems. Particularly, estuarine areas present a high economic, social, and environmental strategic importance, with valuable ecosystem services such as food resources, leisure, tourism, energy, water, and raw materials. They also present advantages for the development of navigation activities, namely conditions for harbour structure location and a direct connection with oceanic and inland waterways, promoting recreational navigation, ship transport, or the fishing industry, among others (Lewis *et al.* 2021; Pearson *et al.* 2021; Taylor & Suthers 2021). However, the anthropogenic pressure in these areas is increasing, as well as the effects of climate change, threatening the availability and quality of these natural system services (Elliott *et al.* 2019; Pörtner *et al.* 2022; Xie *et al.* 2022).

However, it must be noticed that the morphology of alluvial estuarine channels and the hydrodynamic behaviour and sediment transport along them are linked in a feedback loop, resulting in a complex interaction. The morphology is shaped by these two processes and, at the same time, it defines the main sediment transportation patterns (Church & Ferguson

This is an Open Access article distributed under the terms of the Creative Commons Attribution Licence (CC BY 4.0), which permits copying, adaptation and redistribution, provided the original work is properly cited (<http://creativecommons.org/licenses/by/4.0/>).

2015; Poelhekke *et al.* 2016). High current velocities increase the bottom shear stress in the channels, and consequently the sediment transport, which strongly depends on the sediment size. The sediments are transported downstream and deposited in zones with lower current velocities, creating sediment reservoirs and changing the morphology of the channel. When the hydrodynamic conditions are modified, these deposits can be eroded, serving as a sediment source for other estuarine areas.

These dynamics reinforce the strong necessity to fully understand and forecast the morphology of coastal and estuarine areas to promote navigation security and implement adequate measures to protect the populations and the habitats. For instance, forecasting the sedimentation and erosion patterns of coastal and estuarine regions is necessary to understand the effects of accretion areas during flood events, helping, for example, to optimize dredging operations to maintain the navigation channel depth. The knowledge of sediment transport trends and coastal and fluvial geomorphology is essential to understand the possible impacts of floods, implement measures to avoid the effects of extreme events, and regulate the navigation and use of the sediment resources that are abundant in these ecosystems (Kandolf & Piégay 2003). In addition, there is an increase in the pressure and the vulnerability in the coastal regions due to the intensification of human settlements and commercial activities, which reinforce the need for efficient tools that help to understand the complexity of sediment supply and transport (Church & Ferguson 2015).

In this context, the development of tools to monitor and assess morphological changes is of utmost importance. A common methodology to study the morphological evolution of coastal and estuarine regions is based on the implementation of numerical models, which have already reached the capacity to generate highly accurate solutions (Guerrero *et al.* 2013; Iglesias *et al.* 2019; Pinho *et al.* 2020; Zhou *et al.* 2021). Sediment transport models, coupled with hydrodynamic numerical models, allow the simulation of erosion and accretion patterns of rivers, estuaries, and coastal areas. However, the natural process that takes place in these environments can take several centuries to reach an equilibrium condition. The complexity of morphological numerical simulations is also increased due to hydrodynamic changes that could take place at different time and spatial scales. The hydrodynamics is affected by tide variations, for example, presenting hourly to daily variations that can vary from centimetres to several metres, with a direct effect in the current velocity and, therefore, in the bottom shear stresses. Longer time-scale processes like sea level rise (SLR) due to climate change conditions also have a strong effect in the hydrodynamic conditions. However, it must be taken into account that these short- and long-term hydrodynamic variations can result in barely noticed variations in the morphological conditions, depending, among other factors, on the bottom slope and the sediment characteristics (Roering *et al.* 2001).

Additionally, numerical modelling-based methodologies are usually limited by computational resources, such as memory capacity and processing power. This is even more limited when long-term simulations with high spatial resolution models are required, which is a common practice when the focus of the numerical models is the morphodynamic evolution. The higher the resolution of the model and the amount of data that it generates, the higher the computational resource requirements (processing power and storage capacity), demanding, sometimes, prohibitive computational infrastructures and making the application of this type of platform unfeasible. Nevertheless, long-term solutions of high-resolution models are essential to assess the hydro-morphodynamics on local scales.

A possible solution to overcome that difficulty is to use data-based models, also known as surrogate models (Poelhekke *et al.* 2016; Chen *et al.* 2018; Parker *et al.* 2019). These models are implemented using artificial intelligence (AI) algorithms that can learn relationships between the input and the output data instead of solving the differential equations that describe the fluid and sediment transport. Therefore, the focus of this work was to implement a deep convolutional neural network (CNN) to emulate erosion and sedimentation patterns using the hydrodynamic results of a numerical model as input. The architecture of the network used different branches to generate feature maps of the input variables with different spatial resolutions for posterior concatenation in the last layer of the network (Ronneberger *et al.* 2015; Santhanam *et al.* 2016; Li *et al.* 2021). The depth-averaged current velocity and the bottom shear stress variables were used as inputs, and the erosion and sedimentation results were obtained as outputs. The numerical model was implemented in the Delft3D (D3D, <https://oss.deltares.nl/web/delft3d>) software by Elmilady *et al.* (2022), which studied the SLR impact on the long-term morphological evolution of intertidal sandy shoals in a constrained channel-shoal system.

The use of AI in water resources is growing with the increase of data availability and computational processing capacity. For example, genetic programming was used for prediction of the sediment settling velocity (Goldstein & Coco 2014), a deep learning model was applied to forecast the water level in lakes (Barzegar *et al.* 2021), a long short-term memory (LSTM) model was used to forecast the surface water temperature of large-deep reservoirs (Wang *et al.* 2022), and a multilayer perceptron model was applied to relate the water discharge with the sediment rate in the Mississippi river (Jain 2008). Most of

these problems were simplified to a single dimension, making feasible the application of the mentioned techniques. Since the aim of this study was to emulate the results obtained with a two-dimensional (2DH) numerical model, CNNs were selected due to their capacity to process multidimensional data, allowing the use of 2DH model results as input and output.

CNNs have already been successfully applied to predict the water level and water quality indicators in the Nakdong river basin (Baek *et al.* 2020), to predict the daily rainfall–runoff at two monitoring stations in Chau Doc and Can Tho (Van *et al.* 2020), for flood forecast in the Xixian basin (Chen *et al.* 2021), and to reproduce beach morphology from coastal video monitoring systems in France (Soloy *et al.* 2021). However, applications of CNN to assess morphodynamic evolution are still rare and mainly focused on sediment load modelling (Nagy *et al.* 2002; Gupta *et al.* 2021). It is expected that this study will contribute to fill this gap of knowledge in the application of CNNs to simulate the long-term morphology evolution of coastal systems, demonstrating their capacity and limitations to predict accretion and erosion patterns.

METHODS

The used methodology considers four main steps. The first step consists in the implementation of the hydro-morphodynamic numerical model to generate input data for the AI model. This task was based on the work developed by Elmilady *et al.* (2022). In their study, a tidal basin with an open seaward boundary and a landward boundary was implemented in the D3D software to investigate SLR impact on the long-term morphodynamics. A 200-year morphological evolution simulation was performed and the results were made publicly available. Using these results, the present study was mainly focused on the AI model development, and details about the hydro-morphodynamic numerical model implementation can be found in the previous referenced work.

Secondly, the results of the depth-averaged current velocity and the bottom shear stress, two hydrodynamic variables obtained with the numerical model, were exported as images to be used as input for the emulator. Additionally, the bathymetric variation results (cumulative erosion and sedimentation) were also exported to be used as an output. In this process, special care was taken to ensure that all the figures were uniform in terms of size, colour scale, and represented area. The difference in the values of the selected variables between consecutive time steps was computed to reduce the upper- and lower-pixel value limits. The limit values of the selected variables were -0.0025 and 0.0025 m/s for the depth-averaged current velocity, -0.04 and 0.04 Pa for the bottom shear stress, and -0.15 and 0.15 m for the variation in the bathymetry (negative values represent erosion and positive accretion).

In the third step, the numerical model result datasets were organized. The images were loaded and split into training and testing datasets, selecting 292 images for training and 73 for testing.

In the last step, the architecture of the model was set, the training optimizer RMSprop was selected, and the metrics to evaluate the model performance were defined. All these tasks are explained in detail in the following paper subsections.

Numerical model setup and data pre-processing

The D3D is a fully integrated software suite for coastal, riverine, and estuarine areas, simulating non-steady flow and transport phenomena (Deltares 2018). The D3D solves the three-dimensional (3D) time-averaged Navier–Stokes equations for incompressible fluids in the x , y , and z directions, assuming the shallow water equations and the Boussinesq assumptions (Agoshkov *et al.* 1993; Akan 2006; Deltares 2018). For the sediment transport, the software solves a 3D advection-diffusion equation for suspended sediments (Deltares 2018). It considers the Van Rijn approach for sand transport, including bed and suspended loads, and the formulation also considers the combined effects of currents and waves (Elmilady *et al.* 2022). The implementation of these kinds of models is based on the definition of a numerical grid where the transport equations are solved, depending on the boundary conditions. Finer grids produce more detailed solutions, although they increase the simulation computational costs. On the contrary, coarse grids are faster to process, although they can oversimplify the information, resulting in less accurate models.

The numerical model used in this paper was implemented by Elmilady *et al.* (2022), whose work investigated the SLR effects on the evolution of long-term morphodynamics of a sandy channel-shoal system dominated by intertidal sandy shoals, considering the effects of wind and waves in similar conditions to those that occur in the Wadden Sea. The numerical model had an outer coarse grid with 100×200 m of spatial resolution and an inner fine grid with 33×66 m resolution. Two sandy sediment fractions of 100 and 250 μm were considered to represent fine and coarse fractions, respectively. Two hydrodynamic years were simulated, including a morphological acceleration factor of 100, resulting in 200 years of morphological evolution.

The numerical model had one open sea boundary where a semidiurnal tidal component with an amplitude of 1.5 m was adopted. All the other boundaries were closed. Also, a spectral wave model implemented in Simulating Waves Nearshore (SWAN) was used to simulate wind-generated waves. Additional details of the model can be found in [Elmilady et al. \(2022\)](#).

The D3D model results from the high-resolution domain were selected as input and output to the AI model. Images of bed shear stress and square depth-averaged velocity magnitude were generated with a 6-month frequency, resulting in 365 images for each one of the variables. Images of erosion and accretion were also processed with the same frequency to be used by the CNN as output. Each image showed the variation in the cumulative erosion/sedimentation considering the current time step (t) and the previous time step ($t-1$). All images were processed, ensuring that the same colour scale, image dimensions, and spatial resolution (DPI) were adopted. The D3D output file was initially loaded using the D3D MATLAB interface, the numerical model results were read, and a correspondent image was created and exported to a proper image format file. This procedure is repeated for all the simulation time steps. The images were exported at 300 DPI, resulting in images with $1,360 \times 1,060$ pixels. A grey colour map was also considered.

The values of the bottom shear stress varied between 0 and 0.34 Pa, the depth-averaged velocity magnitude varied between 0 and 0.35 m/s, and the bathymetry varied between -11.08 and 13.59 m. These limits were determined by computing the maximum and minimum values observed in the domain, considering all the time steps of the numerical model results. It must be stressed that the input and output values of the images are the differences between two consecutive time steps but not the value of the variable at single time steps.

After processing the images, the TensorFlow function `image_dataset_from_directory` was used, creating a matrix containing each variable data ([Abadi et al. 2015](#)). 80% (292 images) of the data generated by the numerical model was used to train the model and define the values of the CNN parameters. The remaining 20% (73 images) of the data generated was used to assess the performance of the model. These division rates were selected considering the size of the available dataset. It is recommended to use 10–40% of the data for testing and the remaining for training ([Palani et al. 2008](#); [Barzegar et al. 2016](#); [Agrawal & Mittal 2020](#)). Furthermore, the training dataset must provide sufficient and varied samples that allow the model to learn the data pattern. The testing dataset must include values that are not necessarily considered in the training dataset to ensure accurate forecasting for broad conditions, preventing overfitted models.

Input variables

The number of input variables of the CNN model should be defined before generating the artificial dataset that includes the hydrodynamic results obtained by the numerical model. Four locations ([Figure 1](#)) were selected within the domain considering the following criteria: (i) far from the boundaries, (ii) over the axis of the main formed channels, and (iii) erosion/accretion phenomena prone (two locations for each) at the end of the numerical simulation. At these locations, the water level, water depth, bed shear stress, depth-averaged velocity, and square depth-averaged velocity were correlated with cumulative erosion/sedimentation using a Pearson correlation analysis to determine the most adequate variables to be used as input to the AI model ([Asuero et al. 2006](#)). It should be noted that the Pearson correlation coefficient indicates the relation

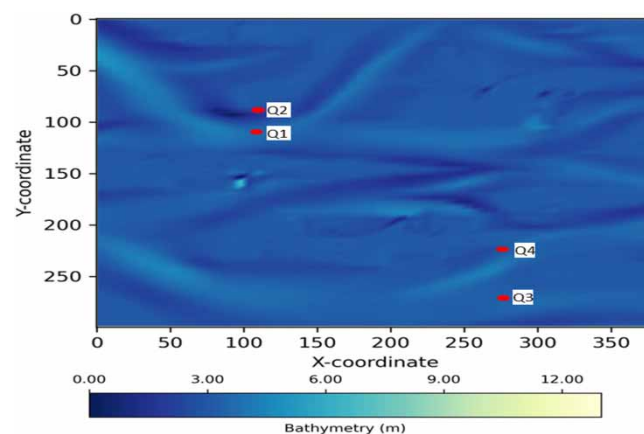


Figure 1 | Initial bathymetry of the estuary and location of selected points for erosion/sedimentation correlation analysis.

between two linear variables. Considering that the hydro-morphodynamic variables do not present linear behaviours, the correlation results were not used to exclude variables with no correlation, but to guide the selection of variables to be considered.

Convolutional neural networks

CNNs were designed to work with grid-structured input, like images, which is an adequate characteristic for emulation of 2DH numerical models results datasets. Each layer of a CNN is a 3D grid with height, width, and depth (or channels) (Aggarwal 2018). Coloured images have three RGB channels, each one representing the intensity of each pixel in the red, green, and blue colour scale, while greyscale images have the same value for all three RGB channels. This way, when using greyscale images as a data source for AI models, it is common to reduce the number of channels to one, decreasing the number of elements in the dataset and optimizing the performance of the model.

Each convolution operation consists of sliding a filter in each position of the image and performing a dot product between the pixel's values and the filter. CNNs also present the property of equivariance to translation, which means that the computed pixel value will be processed in the same way regarding its spatial location (Aggarwal 2018).

The CNN model was implemented in a Spyder environment version 5.2.2 using Python version 3.9.7 and the machine learning system TensorFlow version 2.7.0 (Abadi *et al.* 2015). This framework was selected because it is an open-source platform that allows the implementation of different machine learning models, such as artificial neural networks and deep neural networks. The TensorFlow Keras Functional Application Programming Interface (API) was selected due to the requirement of sharing and combining layers in the network architecture, providing more flexibility during the network implementation. The CNN model used two sets of images as input and one set of images as output. This will require more flexibility during the implementation of the model, like, for example, the use of concatenating operations to merge the convolutional layer outputs for posterior processing.

The definition of the hyperparameters of the convolutional layer is essential to achieve satisfactory results with a CNN. Hyperparameters are model parameters of the CNN layers whose values are not estimated from data. For instance, the number of filters is one hyperparameter of each convolution layer that is defined before training the model, while the values of the weights of each filter are defined during the network training (Noriega 2005; Agrawal & Mittal 2020). Fine-tuning the hyperparameter values is necessary to optimize the network computational performance and improve the accuracy of the model. In this work, five hyperparameters were analysed:

- (i) *Filters*: Each filter captures different information. In each convolution operation, the filter passes along each position of the image and performs a dot product between the data and the filter value (Aggarwal 2018), resulting in a feature map.
- (ii) *Kernel size*: It is the size, in width and height dimensions, of each filter.
- (iii) *Strides*: they are used to control the number of pixel shifts over the input data and to reduce the dimensions of the output. If strides are equal to (2,2), the output size will be half of the input size in the width and height directions. This parameter can be replaced by the use of pooling layers, which also reduces the input dimensions.
- (iv) *Dilation rate*: It is used to increase the range of the filter preserving its size (Yu & Koltun 2015). It is the area of influence of each pixel.
- (v) *Activation function*: It is an activated neuron that can be retained and mapped by nonlinear functions. It is used to increase the performance of the neural network model (Wang *et al.* 2020). Different activation functions result in completely different outputs. Common activation functions are the hyperbolic tangent (tanh), sigmoid, or Rectified Linear Unit (ReLU). In the CNN model implemented in this study, the tanh and the ReLU were selected as activation functions.

The number of filters and the filter size are the most important hyperparameters of a CNN. Variations on these parameters can strongly affect the performance of the algorithm (Agrawal & Mittal 2020; Ahmed & Karim 2020). Particularly, the filter size effect on the result accuracy is strongly dependent on the characteristics of the data that forces the CNN. It should also be noticed that a higher number of filters will produce an increase in the model's accuracy but also an increase in the processing power requirements. However, a small number of filters can result in non-representative feature maps, decreasing the performance of the model.

The activation function is also one of the main decisions that must be taken in the architecture of a CNN. The selected function is responsible for the addition of nonlinearity to the network. This function must be carefully chosen since it affects the computational cost of the network. It is necessary to select functions that demand low computational resources (Aggarwal

2018). It is worth noting that changes in the activation function can completely modify the output of the model. Wang *et al.* (2020) studied different activation functions in a facial expression recognition model, demonstrating that the accuracy of the model can vary more than 30% according to the selected function.

AI techniques with convolutional layers were already selected to be used with numerical model solutions. Tu *et al.* (2021) used the weather research and forecast (WRF) model to generate inputs for a CNN that downscaled hybrid precipitation forecasts, and Kim & Song (2020) implemented a discharge estimator using a CNN with hydrological images as inputs. However, the application of these techniques to solve morphodynamic problems is still scarce and more related to classification problems, such as, for example, the work of Ellenson *et al.* (2020), who used a CNN to classify the nearshore morphological pattern of beaches using images from cameras. This type of network has the potential to develop complex models using images as inputs, making it possible to model the variables of interest in a whole domain, instead of restricting the results to specific points. Therefore, the implementation of a CNN model to emulate a numerical model becomes a promising choice.

The CNN model implemented in this work has five layers, including one input layer, three hidden layers (branch 1, branch 2, and branch 3), and one output layer (Figure 2). The architecture was based on the application of a recursively branched deconvolutional network with three branches, each one used to extract features at different scales (Santhanam *et al.* 2016). The activation function of the convolutional layers was the ReLU (Fukushima 1975).

The input layer read the data, converted it into images with 380×300 pixels, rescaled the values of the pixels between -1 and 1 , and performed three convolutional operations. Two of these operations are considered strides equal to two for down-sampling the data, resulting in images with 190×150 pixels in the first step and finally with 95×75 pixels. The image size reduction was necessary to prevent memory exhaustion during the network training and to amplify the features of the images. The use of strides allows one to rapidly increase the receptive field of each feature and reduce the dimensions of the entire layer (Aggarwal 2018).

The branch 1 and branch 2 layers applied convolution and up-sampling operations in the images with a quarter and a half of the resolution, respectively. The branch 1 inputs were the 95×75 pixel images. The branch 2 inputs were the 190×150 pixel images and the branch's 1 output. Lastly, the branch 3 input was the branch 2 output and the original images (380×300 pixels).

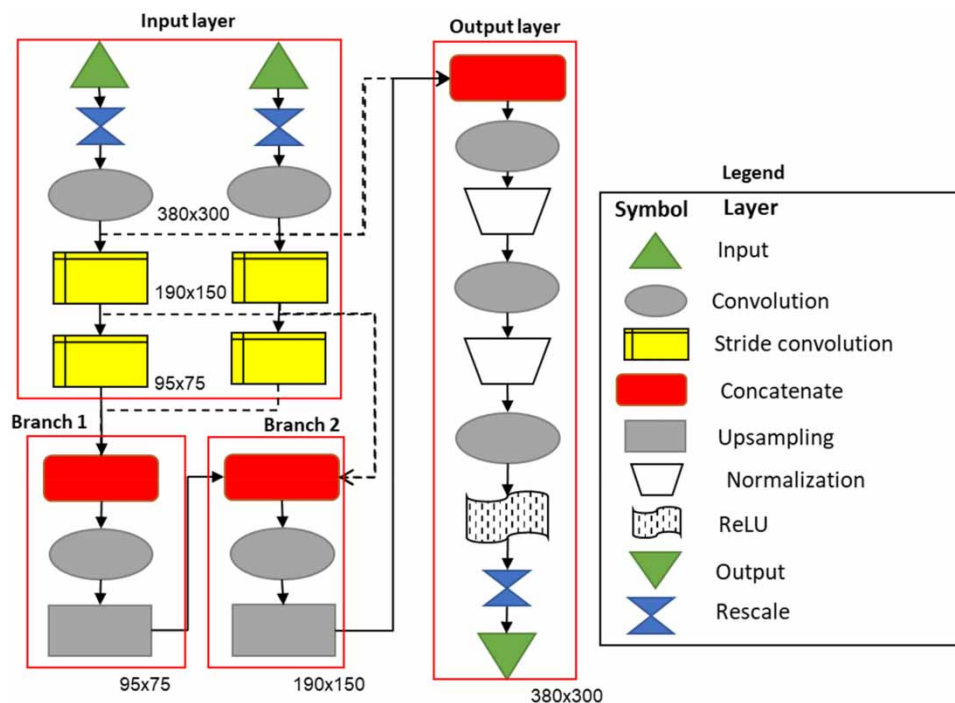


Figure 2 | Architecture of the CNN.

Finally, the output layer performed three convolution operations, in which the two first convolutions were followed by normalization layers, and the last convolution activation function was the hyperbolic tangent. Also, a ReLU layer, with the maximum output value set to 1, was selected to limit the pixel values between 0 and 1. A rescale operation was performed to set the pixel values between 0 and 255, the same scale presented by the input images. These last two layers become necessary because, in each epoch of the training phase, the algorithm determines the error by comparing the output of the AI model with the images of the training datasets. Therefore, to limit errors, both figures must be on the same scale. The addition of the normalization layers standardized the inputs of the convolution layers, improving the metrics in the training dataset and reducing the error between the emulator and the numerical model.

Different image sizes were tested to avoid memory exhaustion and assess the impact of the input resolution in the performance of the model. Theoretically, larger images present more detailed results; however, they also demand more computational resources during the training phase. Since each cell of the numerical model represents a set of pixels, if the input size is reduced, this will not produce a reduction in the details of the images.

Training and testing

This task consists in defining the values of the weights of each layer through an iterative process. This is performed according to an optimization algorithm responsible for reducing the losses and increasing the accuracy of the output (Vijendra Babu *et al.* 2020). The optimizer used in this task was the RMSprop, which restricts the vertical oscillations during the search of the values of the weights, reducing the time to converge (Vijendra Babu *et al.* 2020). A learning rate of 0.9 was used. The mean squared error (MSE) was chosen as loss function. The root mean squared error (RMSE) was also computed to assess the performance of the CNN in the testing dataset. Also, it was necessary to convert the RMSE from the pixel scale to centimetres according to the following equation:

$$\frac{RMSE_{pix}}{scale_{pix}} = \frac{RMSE_{cm}}{scale_{cm}} \quad (1)$$

where the $RMSE_{pix}$ is the RMSE in the pixel scale; $scale_{pix}$ is the difference between the upper and lower limits of the pixel scale; $RMSE_{cm}$ is the RMSE in the centimetres scale, and $scale_{cm}$ is the difference between the upper and lower limits of the centimetres scale.

Each epoch during the training took approximately 25 s to be processed, and the maximum number of epochs in the training was 100. The training was developed using parallel computing and the CUDA toolkit v.11.2 (Harish & Narayanan 2007) in a computer with a graphics card Nvidia GeForce GTX 1060 with 6 Gb, a CPU Intel Core i7-8750H 2.20 GHz, 16 GB random access memory (RAM), and the windows operating system.

RESULTS AND DISCUSSION

Numerical model results

Figure 3 presents an example of the images of each variable used as input and output at different time steps of the simulation. The images were exported in greyscale to reduce the number of channels in the CNN and optimize its performance. Each image represents the difference of the considered variables between the current and the previous time step. In the squared depth-averaged velocity images, the information of possible erosion (black) and accretion (white) areas during that time instant can be identified as associated with higher and lower velocities, respectively. The darker colour in the shear stress represents the areas with the higher values, indicating also where the main erosion channels can be found. In the sedimentation and erosion results, the evolution of the channels according to the time step can be observed. Grey tones indicate neither erosion nor sedimentation in that time step, while black colour indicates sedimentation and white indicates erosion.

Input variable correlation

Table 1 presents the correlation results between the hydrodynamic variables and the cumulative erosion and sedimentation. These results indicate that a simple correlation is not sufficient to affirm the relevance of the input variables due to the non-linear relationship between them. While, at point Q1, the best correlation (−0.95) was obtained with water depth, the correlation at point Q2 with the same variable was 0.67. The correlation with the water level for the selected locations that presented accretion processes presented values of 0.57 at Q1 and of 0.96 at Q4. But this lack of a clear erosion/

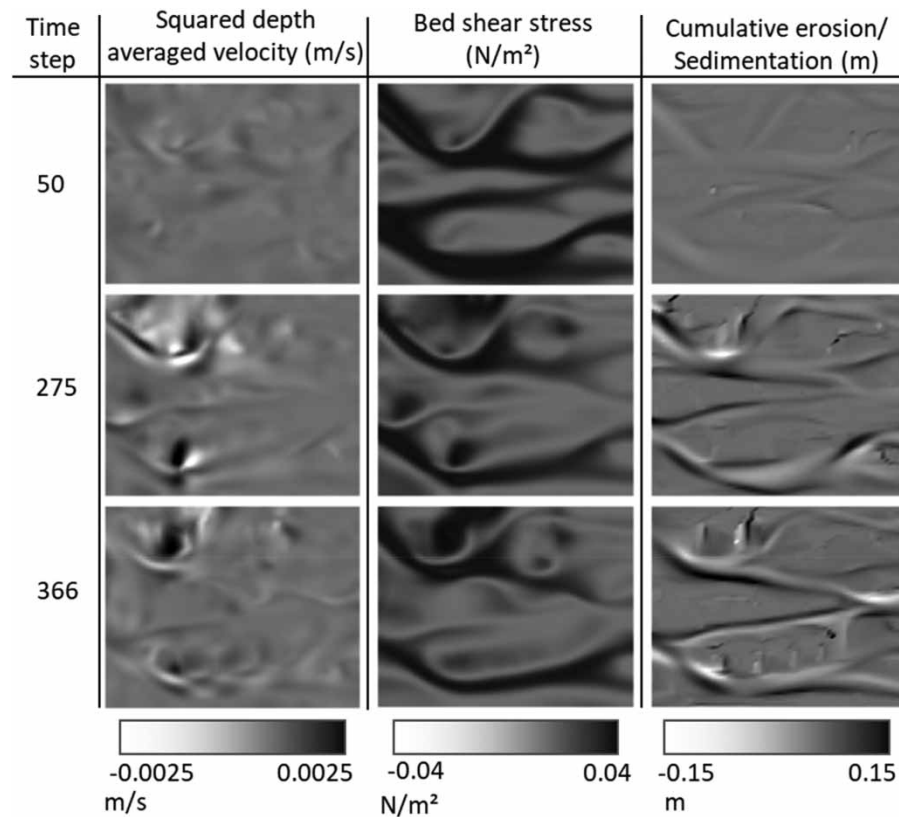


Figure 3 | Examples of the D3D results images used to train the CNN.

Table 1 | Hydrodynamic variables' correlation with cumulative erosion/sedimentation

Obs. point	Pattern	Water level	Water depth	Magnitude of bed shear stress	Magnitude of depth-averaged velocity	Squared depth-averaged velocity	Cumulative erosion/sedimentation
Q1	Sedimentation	0.57	-0.95	0.62	0.62	0.62	1
Q2	Erosion	-0.81	-0.99	0.41	0.43	0.41	1
Q3	Erosion	-0.93	0.71	0.72	0.71	0.71	1
Q4	Sedimentation	0.96	0.67	0.95	0.97	0.95	1

sedimentation pattern correlation with hydrodynamic variables can be observed not only for water depth or water level, but also for all the analysed variables.

Concerning the water depth, the proximity between the points resulted in a similar correlation, regardless of whether the point is eroding or accreting. The other variables presented a pattern with a higher similarity considering the same point. All the correlation values obtained were closer to the velocities and the bed shear stress. This behaviour is clearly explained because the squared depth-averaged velocity and the bottom shear stress are dependent on the depth-averaged velocity. The squared depth-averaged velocity is a linear transformation of the depth-averaged velocity, and the bottom shear stress is proportional to this variable.

After analysing the correlation results presented in Table 1 and considering the physical formulations that describe the sediment transport, the magnitude of bed shear stress and the squared depth-averaged velocity were selected to be tested as input variables.

Training results

To simplify the reference to the convolution layers used in the CNN, they were named according to the layer to which each convolution belongs. Conv n is referred to the input layer, Conv $b.n$ to the convolutions of the branches, and Conv

$4.n$ to the convolutions in the output layer, n being the number of the convolution in that layer and b the number of the branch.

The values of the hyperparameters used to implement the CNN were tested to find the combination that produces the best performance and accuracy. It must be stressed that the activation function of the layer Conv 4.3 was the hyperbolic tangent due to the requirement of its results to be in the range of -1 to 1 . This will simplify the output values and ensure that, after the ReLU and rescaling layers, all the pixel values are in the interval between 0 and 255 .

Furthermore, pooling layers were tested after the convolutional operations to decrease the size of the images in the branches of the network, improving the processing capacity of the model. However, the performance of the model decreased. As the CNN was defined to forecast both points of erosion and sedimentation, the use of a maximum number of pooling layers filtered the information of only one pattern. For this reason, the layers Conv 2 and Conv 3 used 2×2 strides to reduce the input footprint. Also, dropout layers were tested in the place of the normalization layers. However, the results were overestimated when compared to the performance of the selected architecture.

Different combinations of hyperparameter values were tested. The number of filters in each layer varied from 1 to 16 , and the kernel size from $(3,3)$ to $(7,7)$. The combination that resulted in the best model performance and accuracy is presented in Table 2.

Furthermore, it was necessary to reduce the resolution of the input images from $1,360 \times 1,060$ pixels to 380×300 pixels to prevent memory exhaustion during the training. It was observed that this reduction slightly decreased the accuracy of the model; meanwhile, higher resolution images than the one selected presented performance problems in the training or testing. The reduction from $1,360 \times 1,060$ pixels to 380×300 pixels represents 1.3276 million fewer pixels per image to be analysed by the network, being the highest resolution value that preserved the proportion between the width and the height of the images without exhausting the memory of the processor. It must also be stressed that each cell in the numerical model grid is represented by a set of pixels. Hence, the reduction in the number of pixels does not affect the resolution of the results when compared to the numerical model results.

Testing results

A visual comparison between the simulated and the emulated results shows that the main channels are similar in both images (Figure 4). The erosion is represented in red colours as positive values and the sedimentation in blue colours as negative values. However, the CNN tended to predict more sedimentation spots than the numerical model. Despite these differences, the MSE between the numerical model and the surrogate model in the testing dataset (73 samples) was 26.41 cm, and the RMSE was 5.01 cm considering the 0 – 255 pixel scale. Both metrics indicate a reasonable agreement between both models. Considering that a difference of one pixel represents an error of 0.12 cm, the obtained mean RMSE for all time steps is 0.59 cm.

The RMSE for each test sample is presented in Figure 5, and the mean RMSE is the mean value of the RMSE considering all the samples. The performance of the emulator is comparable to the results obtained by other authors that apply a similar approach to other water resource problems. Barzegar *et al.* (2021) achieved an RMSE of 0.04 m with their CNN-LSTM model to forecast water levels in the Michigan Lake, and Wang *et al.* (2022) achieved a 0.25 °C RMSE with a model that forecasted

Table 2 | Hyperparameters of the convolutional layers

Layer	Layer name	Filters	Kernel size	Strides	Dilation rate	Activation function
Input	Conv 1	10	(5,5)	(1,1)	(3,3)	ReLU
	Conv 2	10	(5,5)	(2,2)	(1,1)	ReLU
	Conv 3	16	(5,5)	(2,2)	(1,1)	ReLU
Branch 1	Conv 1.1	16	(5,5)	(1,1)	(3,3)	ReLU
Branch 2	Conv 2.1	16	(5,5)	(1,1)	(3,3)	ReLU
Branch 3	Conv 3.1	10	(5,5)	(1,1)	(3,3)	ReLU
Output	Conv 4.1	10	(5,5)	(1,1)	(3,3)	ReLU
	Conv 4.2	10	(5,5)	(1,1)	(3,3)	ReLU
	Conv 4.3	1	(5,5)	(1,1)	(3,3)	tanh

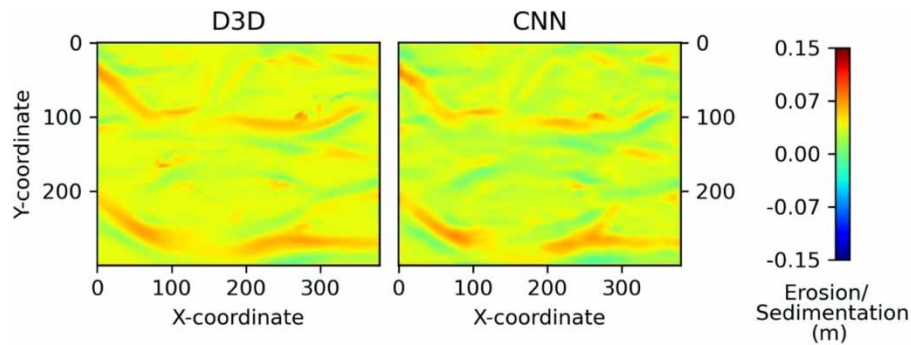


Figure 4 | Cumulative sedimentation and erosion results of the numerical model (D3D) and the data model (CNN).

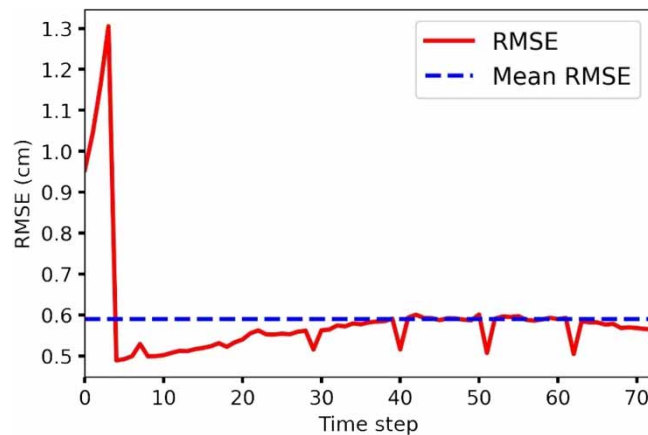


Figure 5 | RMSE variation in the testing dataset.

the surface water temperature of a reservoir. However, the output of both models is 1D, while, in this work, the outputs are 2D, being necessary to determine the mean RMSE to compare, in each time step, the differences between the numerical model and the emulator.

The CNN here presented was able to reproduce the erosion not only in the main channels but also in secondary channels, which demonstrates the capacity of the surrogate model in reproducing the erosion/accretion patterns of complex riverine, coastal, and estuarine areas. Though the numerical model output is smoother than the CNN, which could mean that the numerical model is able to better represent the sediment transport.

Regarding the error of the data model, the computation of the RMSE considers all the pixels of the figures. The maximum RMSE occurred in the third time step, reaching 1.3 cm. It was also observed that the error oscillated between 0.5 and 0.6 cm, demonstrating the robustness of the CNN model.

The comparison of the numerical model and the CNN was also performed at the observation points inside the domain during the simulation (Figure 6). Q1, Q2, and Q3 presented a good agreement between the numerical model and the CNN results. The higher oscillation in the initial time steps is associated with a high variation in the input values. In the following time steps, the oscillations are relatively small and the values are closer to the numerical model values. At Q4, the performance of the CNN is almost identical to the one obtained with the numerical model; however, a systematic error that tends to underestimate the numerical model results can be observed.

Regarding the behaviour of the AI model, in both Figures 4 and 6, predictions are much noisier than the results of the numerical model. The emulator forecasted more areas with no sediment transport than the numerical model, represented by the green colours in Figure 4. This indicates that some adjustments are necessary to improve its performance. This problem can probably be solved by increasing the training dataset size, which will allow the AI model to better-fit its weights and increase its accuracy, or even by adding more input variables.

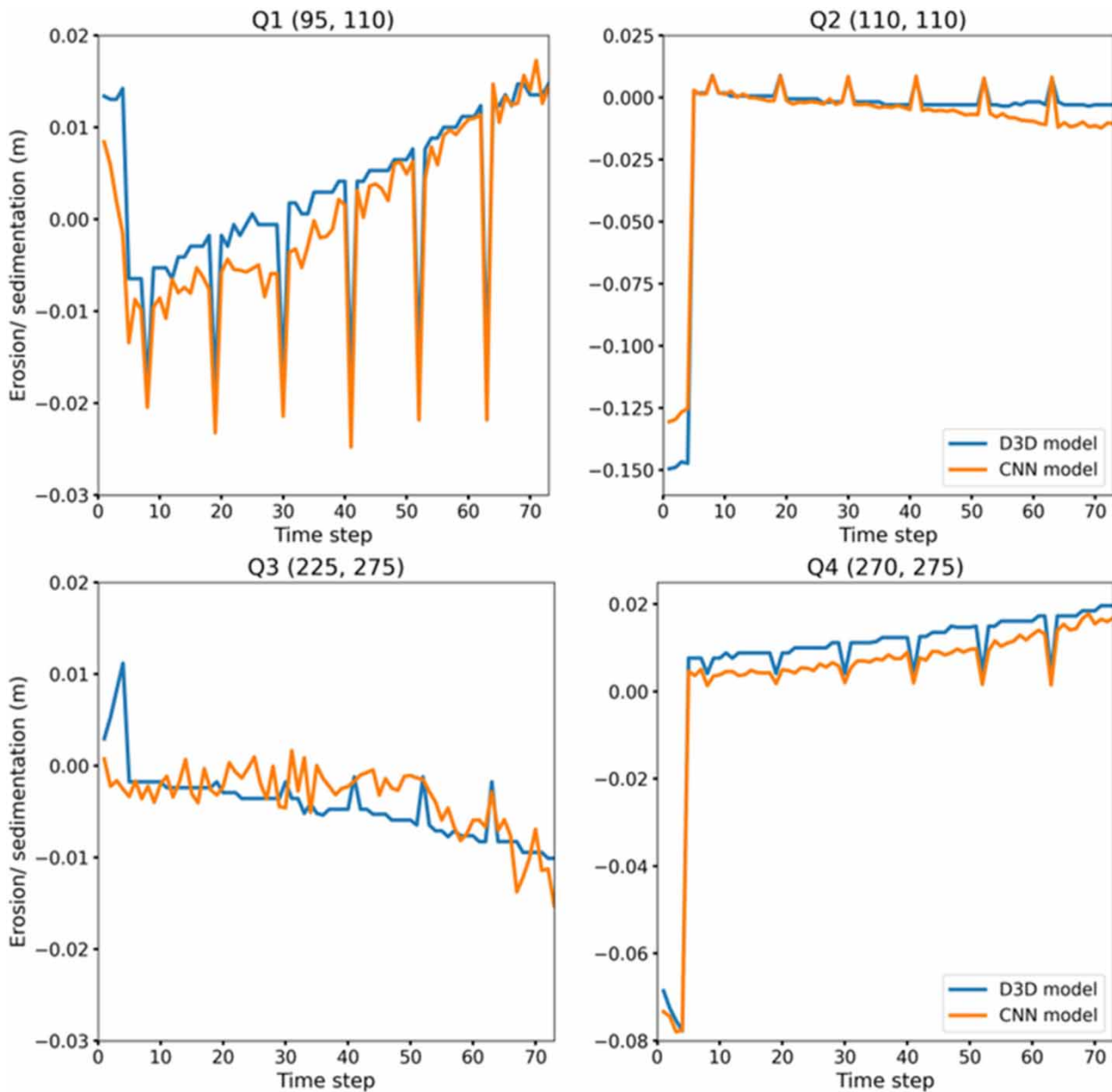


Figure 6 | Differences in the erosion/sedimentation prediction between models at selected points.

The results of the numerical model presented a noisier behaviour than the CNN results, which represents one of the challenges in the implementation of the emulator. The data used for training and testing presented an oscillation between time steps resulting from the hydro-morphodynamic variables. Hence, it is already expected that the results will not be totally linear. At Q2, it is demonstrated that, when the oscillation is smaller, the agreement between the models is better. But despite these differences, it is important to highlight that, on average, the results of both models are close.

Despite the good results obtained as revealed by the computed metrics, indicating a satisfactory performance of the emulator, it is important to note that there are certain aspects that must be improved in future work to optimize the application of CNNs to morphodynamic studies.

Firstly, the D3D exported images were represented using the same grey colour scale for all time steps. The scale was defined by using the highest and lowest values for each considered variable along the whole numerical simulation, ensuring that the values closer to the limits will be considered in the CNN. However, this large colour scale could be hiding small oscillations. Among the strong changes that the sedimentation patterns have during extreme events, the erosion/accretion processes take

place in longer timescales with lower velocities and weaker sediment transport. The consideration of larger colour scales can underestimate the sediment transport during long-term simulations. Therefore, it is possible that a change in the limits of the colour bar of the input images, without changing any other variable, can improve the network results. However, it must be highlighted that if the colour scale varies for each time step, the AI model will have difficulties in interpreting what an oscillation means in the values of the pixels.

Secondly, the size of the images in the input layer must be carefully considered. If the resolution of the input images is low, the time needed by the network to process all the data is shorter, but the details of the images will also be simplified. Hence, it is important to make a balance between the resolution and the computational performance during the implementation of the CNN. The theoretically minimum resolution would be restricted to one pixel per numerical model cell, considering a square image with all the cells aligned. However, this reduction would oversimplify the model, not bringing any advantage in the processing capacity. On the other hand, increasing the resolution of the images can result in memory exhaustion because it will increase the computational cost of the model.

Regarding the performance of the emulator, it was able to simulate 74.5 years in less than 5 s (test dataset), while for a numerical model it took approximately 15 h to simulate the same period with the same computer. So, it is possible to increase the complexity of the AI model to improve its accuracy without being concerned about its computational performance. It was demonstrated that the use of AI algorithms can be an interesting choice to surrogate numerical models when the processing capacity is limited.

Numerical models versus data models

Before comparing the D3D and CNN model results, it is of utmost importance to understand how each model works. Numerical models solve the conservation and transport equations that describe the motion of the water and the transport of sediments for a specific domain. They present the advantage of considering the hydrodynamic and morphodynamic physical-based formulations and, when the models are properly calibrated/validated, they can forecast events for a broader range of boundary conditions. However, the calibration is completely dependent on observational field data to ensure the model's accuracy. At the same time, the calibration must be performed for similar conditions than the ones to be forecasted. For example, if the focus is to forecast the effect of floods, the model must be calibrated with data from previous floods. Otherwise, the uncertainties of the numerical model solutions will be higher. Among the data needed for calibration processes, the numerical model also demands information about the boundary conditions to force the model, namely tides, waves, and river flow. In addition, up-to-date bathymetry is essential to build the numerical grid to accurately represent the hydro-morphodynamic patterns and accurately forecast future conditions.

The considerable time reduction obtained in the simulations performed by the emulator can be explained by how each model computes the variables in each time step. The numerical models integrate a set of 3D transport equations both in time and space (grid cells), allowing the interaction between the different hydro-morphodynamic variables and the adjacent cells. The emulator does not solve these equations. It multiplies the input matrix by the trained weights of each layer. Due to the complexity of the transport equations, solving these differential equations will require more computational resources than multiplying matrixes. However, the most time-consuming activity to implement an AI model is the AI training because the determination of the hyperparameters and the necessary epochs to achieve a stable result is a laborious iterative process.

Combining the use of both models can be an interesting choice to diminish the computational costs while maintaining the accuracy of the solutions. The use of data models is computationally more efficient than the use of numerical models for most of the conditions, and once the model has learned how to interpret new input data, its accuracy will improve, being possible to make more robust data models over time. The numerical models can be useful by generating artificial data to feed the data models with events without measured field data, such as extreme floods, storms, or droughts. Surrogate models fed with numerical model data can bring some insights into the unknown conditions, reducing its error when tested in unprecedented data. This idea becomes even more interesting when the data models can learn from images, as remote sensing data and its training dataset are complemented by numerical models' outputs, increasing the diversity of applications of numerical and data models.

Future discussions

While AI algorithms are a powerful tool to study complex phenomena such as long-term morphodynamics, understanding the problems of AI-based models and the limits in their applications is not an easy task. Future work to improve the CNN

performance must consider the utilization of a higher number of input images in the training process, the addition of more variables to the input datasets, the addition of more filters, the inclusion of changes in other hyperparameters values, or even the modification of the proposed network architecture, including the selection of long- / short-term memory layers, which are ideal for models based on time-series datasets.

Particularly, the assessment of the CNN performance considering different input and output resolutions is an interesting focus for future research. The emulator was much faster than the numerical model, and the main variable that affects the emulator's computational time is the resolution of the images. Hence, to assess the potential limits of the proposed methodology, the sensitivity of the technique to the image resolution should be assessed.

AI models' performance can present extremely high accuracy if the expected forecast results are in the range of the data for which the model was trained. However, the reliability of the AI model is reduced when it is extrapolated for other environmental conditions not included in the training phase. In these situations, the data models do not know which output must be associated with the input, since they were not trained for that set of conditions. Despite these limitations, data models are a promising choice for operational forecast platforms because they can be updated as more data from extreme events become available in the forecast system.

CONCLUSIONS

In this work, a deep CNN data model was successfully implemented to emulate the morphology evolution of an estuary. The results of a hydro-morphodynamic numerical model were selected as the input of the CNN, effectively combining the capacity of real-time processing of data models with the extrapolation capacity of the numerical models to improve the computational efficiency of the morphology evolution forecast. The CNN was implemented to emulate the bathymetric evolution simulated by the D3D numerical model. The hydrodynamic results of squared averaged-depth velocity and bottom shear stress obtained with the numerical model were used as input for the CNN, as well as the erosion and sedimentation variations for output. The images to train and test the surrogate model were carefully generated to maintain the same pattern in size and colour scale.

Obtained results demonstrated the emulator capacity for reproducing the accretion/erosion patterns in the estuarine channels. However, the emulator was not able to represent all the sediment transport patterns reproduced by the numerical model. When the performance of the data model was analysed at four specific locations, it can be observed that the data model forecast presented a noisier behaviour than the numerical model. However, the mean RMSE was 0.59 cm, which demonstrates an acceptable performance of the CNN.

The performance of the emulator was widely superior to the numerical model. While the numerical model needed 15 h to simulate 74.5 years, the CNN needed only 5 s. This demonstrates that the use of AI models will allow us to optimize computational resources, reducing the operation cost of forecasting platforms. Data models are a possible solution to surrogate high-resolution numerical models, strongly reducing the computational resources needed to process and store the data, making it a possible solution for real-time monitoring systems and operational forecasting.

ACKNOWLEDGEMENTS

This research was supported by the Doctoral Grant SFRH/BD/151383/2021 financed by the Portuguese Foundation for Science and Technology (FCT), and with funds from the Ministry of Science, Technology and Higher Education, under the MIT Portugal Program.

DATA AVAILABILITY STATEMENT

All relevant data are included in the paper or its Supplementary Information.

CONFLICT OF INTEREST

The authors declare there is no conflict.

REFERENCES

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D.,

- Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y. & Zheng, X. 2015 *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. Available from: <https://www.tensorflow.org/>.
- Aggarwal, C. C. 2018 Neural networks and deep learning. In: *Artificial Intelligence*. Springer International Publishing. <https://doi.org/10.1007/978-3-319-94463-0>.
- Agoshkov, V. I., Ambrosi, D., Pennati, V., Quarteroni, A. & Saleri, F. 1993 **Mathematical and numerical modelling of shallow water flow**. *Computational Mechanics* **11** (5–6), 280–299. <https://doi.org/10.1007/BF00350088>.
- Agrawal, A. & Mittal, N. 2020 **Using CNN for facial expression recognition: a study of the effects of kernel size and number of filters on accuracy**. *The Visual Computer* **36** (2), 405–412. <https://doi.org/10.1007/s00371-019-01630-9>.
- Ahmed, W. S. & Karim, A. A. A. 2020 The impact of filter size and number of filters on classification accuracy in CNN. In: *2020 International Conference on Computer Science and Software Engineering (CSASE)*, pp. 88–93. <https://doi.org/10.1109/CSASE48920.2020.9142089>.
- Akan, A. O. 2006 Introduction to unsteady open-channel flow. In: *Open Channel Hydraulics*, Vol. 1. Elsevier, pp. 315–359. <https://doi.org/10.1016/B978-075066857-6/50009-6>.
- Asuero, A. G., Sayago, A. & González, A. G. 2006 **The correlation coefficient: an overview**. *Critical Reviews in Analytical Chemistry* **36** (1), 41–59. <https://doi.org/10.1080/10408340500526766>.
- Baek, S.-S., Pyo, J. & Chun, J. A. 2020 **Prediction of water level and water quality using a CNN-LSTM combined deep learning approach**. *Water* **12** (12), 3399. <https://doi.org/10.3390/w12123399>.
- Barzegar, R., Moghaddam, A. A. & Baghban, H. 2016 **A supervised committee machine artificial intelligent for improving DRASTIC method to assess groundwater contamination risk: a case study from Tabriz plain aquifer, Iran**. *Stochastic Environmental Research and Risk Assessment* **30** (3), 883–899. <https://doi.org/10.1007/s00477-015-1088-3>.
- Barzegar, R., Aalami, M. T. & Adamowski, J. 2021 **Coupling a hybrid CNN-LSTM deep learning model with a boundary corrected maximal overlap discrete wavelet transform for multiscale lake water level forecasting**. *Journal of Hydrology* **598**, 126196. <https://doi.org/10.1016/j.jhydrol.2021.126196>.
- Chen, R., & Hutton, S. B. & Hutton, P. H. 2018 **Emulation of a process-based estuarine hydrodynamic model**. *Hydrological Sciences Journal* **63** (5), 783–802. <https://doi.org/10.1080/02626667.2018.1447112>.
- Chen, C., Hui, Q., Xie, W., Wan, S., Zhou, Y. & Pei, Q. 2021 **Convolutional neural networks for forecasting flood process in internet-of-things enabled smart city**. *Computer Networks* **186**, 107744. <https://doi.org/10.1016/j.comnet.2020.107744>.
- Church, M. & Ferguson, R. I. 2015 **Morphodynamics: rivers beyond steady state**. *Water Resources Research* **51** (4), 1885–1897. <https://doi.org/10.1002/2014WR016862>.
- Deltas 2018 *Delft3D-Flow: Simulation of Multi-Dimensional Hydrodynamic Flows and Transport Phenomena, Including Sediments – User Manual*.
- Ellenson, A. N., Simmons, J. A., Wilson, G. W., Hesser, T. J. & Splinter, K. D. 2020 **Beach state recognition using argus imagery and convolutional neural networks**. *Remote Sensing* **12** (23), 1–20.
- Elliott, M., Day, J. W., Ramachandran, R. & Wolanski, E. 2019 **A synthesis: what is the future for coasts, estuaries, deltas and other transitional habitats in 2050 and beyond?** In: *Coasts and Estuaries*. Elsevier, pp. 1–28. <https://doi.org/10.1016/B978-0-12-814003-1.00001-0>.
- Elmilady, H., van der Wegen, M., Roelvink, D. & van der Spek, A. 2022 **Modeling the morphodynamic response of estuarine intertidal shoals to sea-level rise**. *Journal of Geophysical Research: Earth Surface* **127** (1), 1–26. <https://doi.org/10.1029/2021JF006152>.
- Fukushima, K. 1975 **Cognitron: a self-organizing multilayered neural network**. *Biological Cybernetics* **20** (3–4), 121–136. <https://doi.org/10.1007/BF00342633>.
- Goldstein, E. B. & Coco, G. 2014 **A machine learning approach for the prediction of settling velocity**. *Water Resources Research* **50** (4), 3595–3601. <https://doi.org/10.1002/2013WR015116>.
- Guerrero, M., Di Federico, V. & Lamberti, A. 2013 **Calibration of a 2-D morphodynamic model using water–sediment flux maps derived from an ADCP recording**. *Journal of Hydroinformatics* **15** (3), 813–828. <https://doi.org/10.2166/hydro.2012.126>.
- Gupta, D., Hazarika, B. B., Berlin, M., Sharma, U. M. & Mishra, K. 2021 **Artificial intelligence for suspended sediment load prediction: a review**. *Environmental Earth Sciences* **80** (9), 346. <https://doi.org/10.1007/s12665-021-09625-3>.
- Harish, P. & Narayanan, P. J. 2007 **Accelerating large graph algorithms on the GPU using CUDA**. In: *High Performance Computing – HiPC 2007: Vol. 4873 LNCS*. Springer, Berlin, Heidelberg, pp. 197–208. https://doi.org/10.1007/978-3-540-77220-0_21.
- Iglesias, I., Venâncio, S., Pinho, J. L., Avilez-Valente, P. & Vieira, J. M. P. 2019 **Two models solutions for the Douro estuary: flood risk assessment and breakwater effects**. *Estuaries and Coasts* **42** (2), 348–364. <https://doi.org/10.1007/s12237-018-0477-5>.
- Jain, S. K. 2008 **Development of integrated discharge and sediment rating relation using a compound neural network**. *Journal of Hydrologic Engineering* **13** (3), 124–131. [https://doi.org/10.1061/\(ASCE\)1084-0699\(2008\)13:3\(124\)](https://doi.org/10.1061/(ASCE)1084-0699(2008)13:3(124)).
- Kandolf, G. M., Piégay, H., 2003 In: *Tools in Fluvial Geomorphology*, 2nd edn (Kandolf, G. M. & Piégay, H., eds). Wiley. <https://doi.org/10.1002/0470868333>.
- Kim, D. Y. & Song, C. M. 2020 **Developing a discharge estimation model for ungauged watershed using CNN and hydrological image**. *Water* **12** (12), 3534. <https://doi.org/10.3390/w12123534>.
- Lewis, N. S., Young, D. R., Folger, C. L. & DeWitt, T. H. 2021 **Assessing the relative importance of estuarine nursery habitats – a Dungeness Crab (*Cancer magister*) case study**. *Estuaries and Coasts* **44** (4), 1062–1073. <https://doi.org/10.1007/s12237-020-00821-1>.

- Li, J., Bare, B., Zhou, S., Yan, B. & Li, K. 2021 Organ-branched CNN for robust face super-resolution. In: *2021 IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6. <https://doi.org/10.1109/ICME51207.2021.9428152>.
- Nagy, H. M., Watanabe, K. & Hirano, M. 2002 Prediction of sediment load concentration in rivers using artificial neural network model. *Journal of Hydraulic Engineering* **128** (6), 588–595. [https://doi.org/10.1061/\(ASCE\)0733-9429\(2002\)128:6\(588\)](https://doi.org/10.1061/(ASCE)0733-9429(2002)128:6(588)).
- Noriega, L. 2005 *Multilayer Perceptron Tutorial*. School of Computing, Staffordshire University, pp. 1–12. Available from: <http://www.amno.moph.go.th/research/uploadfile/1365058846mlp.pdf>.
- Palani, S., Liong, S. Y. & Tkalich, P. 2008 An ANN application for water quality forecasting. *Marine Pollution Bulletin* **56** (9), 1586–1597. <https://doi.org/10.1016/j.marpolbul.2008.05.021>.
- Parker, K., Ruggiero, P., Serafin, K. A. & Hill, D. F. 2019 Emulation as an approach for rapid estuarine modeling. *Coastal Engineering* **150**, 79–93. <https://doi.org/10.1016/j.coastaleng.2019.03.004>.
- Pearson, R. G., Connolly, N. M., Davis, A. M. & Brodie, J. E. 2021 Fresh waters and estuaries of the Great Barrier Reef catchment: effects and management of anthropogenic disturbance on biodiversity, ecology and connectivity. *Marine Pollution Bulletin* **166**, 112194. <https://doi.org/10.1016/j.marpolbul.2021.112194>.
- Pinho, J. L. S., Vieira, L., Vieira, J. M. P., Venâncio, S., Simões, N. E., Sá Marques, J. A. & Santos, F. S. 2020 Assessing causes and associated water levels for an urban flood using hydroinformatic tools. *Journal of Hydroinformatics* **22** (1), 61–76. <https://doi.org/10.2166/hydro.2019.019>.
- Poelhekke, L., Jäger, W. S., van Dongeren, A., Plomaritis, T. A., McCall, R. & Ferreira, Ó. 2016 Predicting coastal hazards for sandy coasts with a Bayesian network. *Coastal Engineering* **118**, 21–34. <https://doi.org/10.1016/j.coastaleng.2016.08.011>.
- Pörtner, H.-O., Roberts, D. C., Tignor, M., Poloczanska, E. S., Mintenbeck, K., Alegría, A., Craig, M., Langsdorf, S., Löschke, S., Möller, V., Okem, A. & Rama, B. 2022 *Climate Change 2022: Impacts, Adaptation, and Vulnerability. Contribution of Working Group II to the Sixth Assessment Report of the Intergovernmental Panel on Climate Change*. Cambridge University Press, Cambridge, UK and New York, NY.
- Roering, J. J., Kirchner, J. W. & Dietrich, W. E. 2001 Hillslope evolution by nonlinear, slope-dependent transport: steady state morphology and equilibrium adjustment timescales. *Journal of Geophysical Research: Solid Earth* **106** (B8), 16499–16513. <https://doi.org/10.1029/2001JB000323>.
- Ronneberger, O., Fischer, P., Brox, T., 2015 U-Net: convolutional networks for biomedical image segmentation. In: *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (Navab, N., Hornegger, J., Wells, W. M. & Frangi, A. F., eds). Springer International Publishing, Vol. 9351, pp. 234–241. https://doi.org/10.1007/978-3-319-24574-4_28.
- Santhanam, V., Morariu, V. I. & Davis, L. S. 2016 Generalized deep image to image regression. In: *Proceedings – 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017 January, pp. 5395–5405. <https://doi.org/10.1109/CVPR.2017.573>.
- Soloy, A., Turki, I., Lecoq, N., Gutiérrez Barceló, Á. D., Costa, S., Laignel, B., Bazin, B., Soufflet, Y., Le Louargant, L. & Maquaire, O. 2021 A fully automated method for monitoring the intertidal topography using video monitoring systems. *Coastal Engineering* **167**. <https://doi.org/10.1016/j.coastaleng.2021.103894>.
- Taylor, M. D. & Suthers, I. M. 2021 The socio-ecological system of urban fisheries in estuaries. *Estuaries and Coasts* **44** (7), 1744–1751. <https://doi.org/10.1007/s12237-021-00916-3>.
- Tu, T., Ishida, K., Ercan, A., Kiyama, M., Amagasaki, M. & Zhao, T. 2021 Hybrid precipitation downscaling over coastal watersheds in Japan using WRF and CNN. *Journal of Hydrology: Regional Studies* **37**, 100921.
- Van, S. P., Le, H. M., Thanh, D. V., Dang, T. D., Loc, H. H. & Anh, D. T. 2020 Deep learning convolutional neural network in rainfall-runoff modelling. *Journal of Hydroinformatics* **22** (3), 541–561. <https://doi.org/10.2166/hydro.2020.095>.
- Vijendra Babu, D., Karthikeyan, C., Shreya & Kumar, A. 2020 Performance analysis of cost and accuracy for Whale Swarm and RMSprop optimizer. In: *IOP Conference Series: Materials Science and Engineering*, Vol. 993, No. 1, p. 012080. <https://doi.org/10.1088/1757-899X/993/1/012080>.
- Wang, Y., Li, Y., Song, Y. & Rong, X. 2020 The influence of the activation function in a convolution neural network model of facial expression recognition. *Applied Sciences* **10** (5), 1897. <https://doi.org/10.3390/app10051897>.
- Wang, L., Xu, B., Zhang, C., Fu, G., Chen, X., Zheng, Y. & Zhang, J. 2022 Surface water temperature prediction in large-deep reservoirs using a long short-term memory model. *Ecological Indicators* **134**, 108491. <https://doi.org/10.1016/j.ecolind.2021.108491>.
- Xie, D., Schwarz, C., Kleinhans, M. G., Zhou, Z. & Maanen, B. 2022 Implications of coastal conditions and sea-level rise on mangrove vulnerability: a bio-morphodynamic modeling study. *Journal of Geophysical Research: Earth Surface* **127** (3), 1–28. <https://doi.org/10.1029/2021jfo006301>.
- Yu, F. & Koltun, V. 2015 Multi-scale context aggregation by dilated convolutions. In: *4th International Conference on Learning Representations, ICLR 2016 – Conference Track Proceedings*. Available from: <http://arxiv.org/abs/1511.07122>.
- Zhou, X., Ma, W., Echizenya, W. & Yamazaki, D. 2021 The uncertainty of flood frequency analyses in hydrodynamic model simulations. *Natural Hazards and Earth System Sciences* **21** (3), 1071–1085. <https://doi.org/10.5194/nhess-21-1071-2021>.

First received 14 May 2022; accepted in revised form 9 September 2022. Available online 20 September 2022