



Research papers

Rapid spatio-temporal flood prediction and uncertainty quantification using a deep learning method

R. Hu^a, F. Fang^{a,*}, C.C. Pain^a, I.M. Navon^b^a Applied Modelling and Computation Group, Department of Earth Science and Engineering, Imperial College London, UK^b Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA

ARTICLE INFO

Keywords:

ROM
POD/SVD
Deep learning
LSTM
Flood prediction

ABSTRACT

Recently accrued attention has been given to machine learning approaches for flooding prediction. However, most of these studies focused mainly on time-series flooding prediction at specified sensors, rarely on spatio-temporal prediction of inundations. In this work, an integrated long short-term memory (LSTM) and reduced order model (ROM) framework has been developed. This integrated LSTM-ROM has the capability of representing the spatio-temporal distribution of floods since it takes advantage of both ROM and LSTM. To reduce the dimensional size of large spatial datasets in LSTM, the proper orthogonal decomposition (POD) and singular value decomposition (SVD) approaches are introduced. The LSTM training and prediction processes are carried out over the reduced space. This leads to an improvement of computational efficiency while maintaining the accuracy. The performance of the LSTM-ROM developed here has been evaluated using Okushiri tsunami as test cases. The results obtained from the LSTM-ROM have been compared with those from the full model (Fluidity). In predictive analytics, it is shown that the results from both the full model and LSTM-ROM are in a good agreement whilst the CPU cost using the LSTM-ROM is decreased by three orders of magnitude compared to full model simulations. Additionally, prescriptive analytics has been undertaken to estimate the uncertainty in flood induced conditions. Given the time series of the free surface height at a specified detector, the corresponding induced wave conditions along the coastline have then been provided using the LSTM network. Promising results indicate that the use of LSTM-ROM can provide the flood prediction in seconds, enabling us to provide real-time predictions and inform the public in a timely manner, reducing injuries and fatalities.

1. Introduction

Flooding causes considerable damage to people, infrastructure and economies in many countries of the world. During a flooding event, a rapid response management is critical to reducing damage resulting from a flood event. Emergency managers require timely and accurate information on the areas affected by floodwater to plan mitigation measures against damage. In recent years, the Early Warning Systems (EWS) have been widely used to better predict floods and reduce their impact on urbanized areas. Learning from the past flood events is vital to enhancing the capability of flood forecasting.

Flood modelling involves the complex and nonlinear flow and physical processes. Compared to a physically based numerical model, machine learning approaches can provide a powerful way for flooding prediction without explicitly knowing such nonlinear dynamic processes (Mosavi et al., 2018). In the past, great efforts have been made to flooding prediction using machine learning methods, for example,

autoencoder (Alves et al., 2010), genetic algorithm (GA) (Wu and Chau, 2006), artificial neural networks (ANN) (Mukerji et al., 2009), back propagation neural networks (BPNN) (Lin and Chen, 2008), radial basis function neural networks (RBFNN) (Lin and Chen, 2004) and recurrent neural networks (RNN) (Chang et al., 2014; Chen et al., 2013). Most of these studies focused mainly on time-series flooding prediction (e.g. water levels, streamflow and discharge) at specified sensors (Mosavi et al., 2018), but rarely addressed spatio-temporal prediction of inundations since it is difficult to handle large spatial data sets.

In this work, we propose the long short-term memory (LSTM) for flood prediction. LSTM was introduced by Sepp Hochreiter and Jurgen Schmidhuber (Hochreiter and Schmidhuber, 1997). LSTMs are able to learn non-linear functions of arbitrary-length input sequences and effectively capture long-term temporal dependencies. They have been widely used in various research areas, such as, language modelling (Zaremba et al., 2014), handwriting recognition (Bertolami et al., 2009; Doetsch et al., 2014; Pham et al., 2014), speech synthesis (Fan et al.,

* Corresponding author.

E-mail address: f.fang@imperial.ac.uk (F. Fang).<https://doi.org/10.1016/j.jhydrol.2019.05.087>

Received 19 March 2019; Received in revised form 23 May 2019; Accepted 28 May 2019

Available online 31 May 2019

0022-1694/ © 2019 Elsevier B.V. All rights reserved.

2014), audio analysis (Marchi et al., 2014), and video recognition (Donahue et al., 2015).

For efficient LSTM training and spatio-temporal prediction in flood events, ROM techniques are introduced to LSTM in this study. ROM techniques have been widely used in various research fields including fluid dynamics (Wang et al., 2016; Alotaibi et al., 2015), molecular dynamics (Hoang et al., 2016a), heat transfer (Peherstorfer and Willcox, 2016), data assimilation (Dimitriu et al., 2009; Dimitriu and Apreutesei, 2007), elasticity problems (Hoang et al., 2016b), shape optimization (Quilliec et al., 2015), and aeroplane components design (Manzoni et al., 2015). The integrated LSTM and ROM framework developed here takes advantage of both ROM and RNN-LSTM. This is the first time that LSTM network based on ROM has been considered in flood forecasting and used to estimate spatial aggregation of inundations. It should be noted that having the compatibility of LSTM and ROM will enable solving predictive problems efficiently and accurately.

The integrated LSTM-ROM framework developed here has the capability of predictive and prescriptive analytics. Predictive analysis is used to forecast the future by learning patterns from holistic/experience data. Prescriptive analytics is used for inverse problems. In prescriptive analytics the past/estimated data is used to explore dependencies among results. These previous learned dependencies are then used to estimate the cause factors given the prescribed results at a future time. This means that predictive analytics tells what will happen, while prescriptive analytics provides the cause factors and suggests what to do. Predictive and prescriptive analytics with big data are becoming more and more prevalent in industries (Soltanpoor and Sellis, 2016; Vahn, 2014). To the best of our knowledge, there has not been any particular study which incorporates these two types of analytics together to provide a framework to address flood issues. The goal of this work is to develop an advanced fast-running computational model to predict urban floods and guide effective response in the event of emergencies. It will lead to a step change in the speed of forecasting, with possible CPU reductions of several orders of magnitude compared to existing methods.

The layout of the paper is as follows. Section 2 introduces the methodology of LSTM. The integrated LSTM-ROM framework for flood prediction is provided in detail in Section 3 while LSTM for predictive and prescriptive analytics in flood prediction is described in Section 4. Section 5 demonstrates the performance of the LSTM-ROM using Okushiri tsunami as test cases. Finally in Section 6, conclusions and future work are presented.

2. Long-short term memory (LSTM)

Over existing recurrent neural networks, LSTM has an internal memory system to deal with temporal sequence inputs, which makes it a powerful tool for flooding predictions.

A LSTM module is composed of a memory cell and three gates: an input gate, a forgetting gate and an output gate. The input gate controls the information transmitting from the input activations into the cell, and the forget gate and output gate select the information needed to be transmitted into next cell. Fig. 1 shows how information is flowing through a LSTM cell via the cell state, which is considered as a conveyor belt in the network. The relationship between the input (x_s) and output (h_s) can be obtained as follows:

$$\begin{aligned} f_s &= \sigma(W_f \cdot [h_{s-1}, x_s] + b_f), i_s = \sigma(W_i \cdot [h_{s-1}, x_s] + b_i), \tilde{C}_s \\ &= \tanh(W_c \cdot [h_{s-1}, x_s] + b_c), C_s = f_s * C_{s-1} + i_s * \tilde{C}_s, o_s \\ &= \sigma(W_o \cdot [h_{s-1}, x_s] + b_o), h_s = o_s * \tanh(C_s), \end{aligned} \quad (1)$$

where i_s , o_s and f_s represent the input, output and forget gates respectively, σ is the gate activation function, W_i , W_o , W_f and W_c are the weights for each of the gates and cells, b represent bias weights, C denotes the cell state, \tilde{C} is the updated cell state, x_s and h_s are the cell input and output respectively where the subscripts s and $s - 1$ denote

the current and previous predictive instances respectively. With transmitting of the information, the cell states are propagated forward and weights are updated through time.

3. LSTM-ROM for predictive modelling of the spatio-temporal distribution of floods

As discussed above, most of existing flooding studies based on machine learning techniques are limited to either the time-series flooding prediction at sensors or the simulations of the spatial distribution of floods. The main objective of this work is to develop a new LSTM tool for predicting the spatio-temporal distribution of floods. In spatio-temporal simulations, the input dataset in Eq. (1) is the spatial function and can be rewritten:

$$x_s = (h_{b,s}, \mu), \quad (2)$$

where $h_{b,s}$ is the flood induced condition (here, an incoming wave along the coastal boundary), and μ represents the uncertainty (drag coefficient, for example) in modelling, while the output in Eq. (1) is the spatio-temporal distribution during the specified flooding period $[0, t_{N_t}]$, that is,

$$h_s = (h_{s,0}, \dots, h_{s,t_{N_t}}, \dots, h_{s,t_{N_t}}), \quad (3)$$

where $h_{s,t_{N_t}}$ is the variable spatial solution at time level t_{N_t} (here, $n_t \in \{0, \dots, N_t\}$, $t_{N_t} \in [0, t_{N_t}]$ and N_t is the number of time instances). In a discretisation form, the spatial variables in Eqs. (2) and (3) can be rewritten:

$$h_{s,0} = (h_{s,0,1}, \dots, h_{s,0,n}, \dots, h_{s,0,N}), \quad h_{s,t_{N_t}} = (h_{s,t_{N_t},1}, \dots, h_{s,t_{N_t},n}, \dots, h_{s,t_{N_t},N}), \quad (4)$$

and the variables:

- $h_{s,0}, h_{s,t_{N_t}}, h_{s,t_{N_t}} \in \mathcal{R}^{\mathcal{N}}$, where \mathcal{N} is the number of nodes where data or solution is provided in the computational domain Ω ;
- $h_b \in \mathcal{R}^{N_b}$, where N_b is the number of parameters used for defining the flood induced conditions (for example, the peak values and period in an incoming wave);
- $\mu \in \mathcal{R}^{N_m}$, where N_m is the number of uncertainties in modelling;
- $x_s \in \mathcal{R}^{N_b+N_m}$;
- $h_s \in \mathcal{R}^{N_t \times \mathcal{N}}$.

One may note that for spatio-temporal prediction of floods within a large flooding area, huge spatial datasets have to be handled during the training and predictive processes. To reduce the size of data sets, a ROM based on proper orthogonal decomposition (POD) is introduced to LSTM in this work.

3.1. Reduction of a large spatial dataset using ROM techniques

3.1.1. Proper orthogonal decomposition (POD) and singular value decomposition (SVD)

Let H the snapshot matrix containing flooding variables at instances of the temporal sequence during the flooding period $[0, t_{N_t}]$, that is,

$$H = (h_{s,0}^T, \dots, h_{s,t_{N_t}}^T, \dots, h_{s,t_{N_t}}^T), \quad (5)$$

where $H \in \mathcal{R}^{\mathcal{N} \times N_t}$, $\mathcal{N} \gg N_t$, the flooding dataset $\{h_{s,t_{N_t}}\}$ is constructed from the output of Eq. (1) and the superscript T denotes the transpose.

The essence of ROM is to project the snapshot matrix onto a low-dimensional space $\mathcal{R}^{\mathcal{M} \times N_t}$ (here, $\mathcal{M} \ll \mathcal{N}$). By using POD/SVD, one finds a set of POD modes ϕ_m to construct the reduced space $\mathcal{R}_m = \text{span}\{\phi_1, \dots, \phi_m, \dots, \phi_{\mathcal{M}}\}$ in an optimal way such that the residual of the projection $\|H - \Phi^T H \Phi\|^2$ is minimized, that is equivalent to maximizing the variance of the projected data:

$$\max_{\phi_m} \text{var}(H) = \max_{\phi_m} \Phi^T \frac{H^T H}{N_t} \Phi, \quad (6)$$

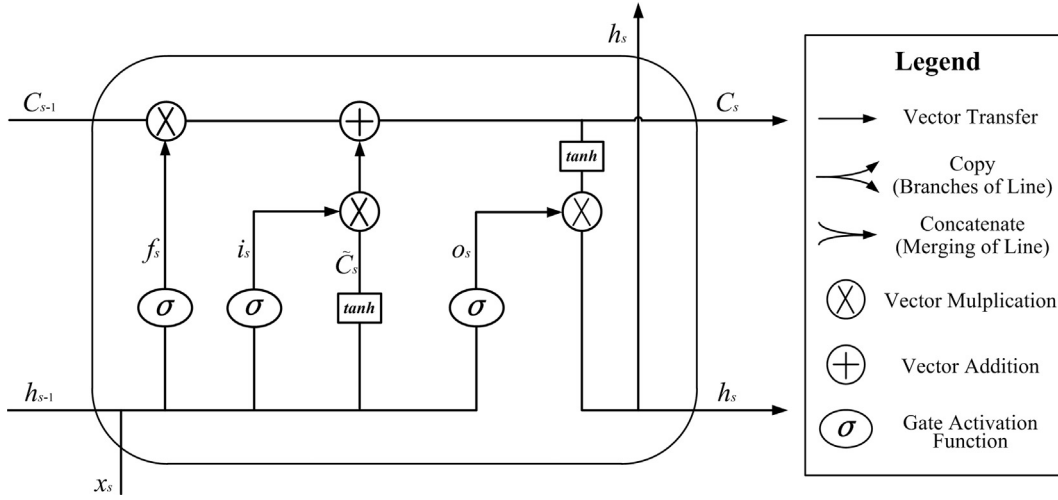


Fig. 1. One cell in the LSTM network.

where $\Phi = (\phi_1, \dots, \phi_m, \dots, \phi_{\mathcal{M}})$. Using Singular Value Decomposition (SVD), the rectangular matrix H can be written if it is invertible:

$$H = U \Sigma V^T, \quad (7)$$

where the columns of the unitary matrices $U = (U_1, \dots, U_{\mathcal{M}}) \in \mathbb{R}^{\mathcal{M} \times \mathcal{M}}$ and $V = (V_1, \dots, V_{N_t}) \in \mathbb{R}^{N_t \times N_t}$ are the orthonormal eigenvectors of HH^T and H^TH respectively, $\Sigma \in \mathbb{R}^{N_t \times N_t}$ is the diagonal matrix with entries σ_i , $i \in \{1, \dots, N_t\}$, which are the positive singular values arranged in descending order.

The eigenvalue problem of the correlation matrix $H^TH = V \Sigma U^T U \Sigma V^T = V \Sigma^2 V^T \in \mathbb{R}^{N_t \times N_t}$ is solved (Sirovich, 1987):

$$H^TH V_i = \lambda_i V_i, \quad i \in \{1, \dots, N_t\}, \quad (8)$$

where $\lambda_i = \sigma_i^2$ is the eigenvalue of H^TH . The left eigenvector of H can thus be obtained:

$$U_j = \frac{1}{\sqrt{\lambda_j}} H V_j, \quad j \in \{1, \dots, N_t\}, \quad (9)$$

where the eigenvalues are ordered $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_n \geq 0$ while the eigenvectors are orthogonal. The \mathcal{M} leading left eigenvectors are chosen to be the POD modes for constructing the reduced space $\Phi = (\phi_1, \dots, \phi_{\mathcal{M}}) = (U_1, \dots, U_{\mathcal{M}})$. The eigenvector associated with the largest eigenvalue is the direction along which the data have the most variance of the projection. The variance of the projection can be calculated by $\sum_{m=1}^{\mathcal{M}} \lambda_m$ while the percentage of the total energy captured by the POD modes is:

$$E(\mathcal{M}) = \frac{\sum_{i=1}^{\mathcal{M}} \lambda_i}{\sum_{i=1}^{N_t} \lambda_i}, \quad (10)$$

The number of POD modes \mathcal{M} is chosen such that

$$\mathcal{M} = \operatorname{argmin}\{E(k): E(k) \geq \gamma\}, \quad (11)$$

where γ ($0 \leq \gamma \leq 1$) is the tolerance percentage chosen to enable most of the energy of the full system to be captured.

3.1.2. Re-construction of solution datasets

Once the POD modes are obtained, the solution snapshot at each instance of the data sequence can be re-constructed as the linear combination of the POD modes:

$$H \approx \sum_{m=1}^{\mathcal{M}} \sqrt{\lambda_m} \phi_m V_m^T, \quad (12)$$

where the truncated error (the residual of the projection) is $\sum_{m=\mathcal{M}+1}^{N_t} \lambda_m$.

For any input $x_s \in \mathcal{S} \setminus \mathcal{S}_{tr}$ in Eq. (2) (where \mathcal{S} and \mathcal{S}_{tr} are the whole input dataset and the training input dataset respectively), the predictive solution at each time instance n_t can be expressed as a linear combination of POD modes:

$$h_{s,n_t}^T = \Phi a_{s,n_t}^T = \sum_{m=1}^{\mathcal{M}} a_{s,n_t,m} \phi_m, \quad (13)$$

where a_{s,n_t} is the time-dependent coefficient associated with the corresponding POD mode ϕ_m . The coefficients $a_{s,n_t} = (a_{s,n_t,1}, \dots, a_{s,n_t,m}, \dots, a_{s,n_t,\mathcal{M}})$ can be calculated using the LSTM described in Eq. (1). The output $h_s = (h_{s,0}, \dots, h_{s,n_t}, \dots, h_{s,N_t})$ in Eq. (3) is projected onto the low-dimensional space spanned by $\{\phi_1, \dots, \phi_m, \dots, \phi_{\mathcal{M}}\}$:

$$a_s = (a_{s,1}, \dots, a_{s,n_t}, \dots, a_{s,N_t}) = (h_{s,0} \Phi, \dots, h_{s,n_t} \Phi, \dots, h_{s,N_t} \Phi). \quad (14)$$

Note that the dimensional size of the output dataset in Eq. (3) is reduced from $\mathcal{N} \times N_t$ to $\mathcal{M} \times N_t$ ($\mathcal{M} \ll \mathcal{N}$). The coefficient set a_s can replace h_s in Eq. (1) and be obtained using LSTM.

4. LSTM in predictive and prescriptive analytics of floods

In this section, taking an example of flooding caused by incoming waves (but not limited to), we demonstrate the way how to use LSTM for flood prediction and uncertainty quantification.

4.1. Predictive analytics: the spatio-temporal model based on LSTM-ROM

Predictive analytics \Rightarrow Given $\{x_s\}$, find $y_s = f(x_s, t)$. Given a set of inputs $\{x_s \in \mathcal{S}_{tr}\}$ (where \mathcal{S}_{tr} is the training input dataset), using machine learning techniques, one can find a mathematical predictive model, $f(x_s, t)$, which is used for providing the variable values at a future time t . In flooding simulations, for a set of given training input-output pairs $(x_s, y_s) \in (\mathcal{S}_{tr}, \mathcal{C}_{tr})$, one first uses the past flooding data and machine learning techniques to construct the predictive model. For a new flood induced condition $x_s \in \mathcal{S} \setminus \mathcal{S}_{tr}$, one can forecast the future floods. To tackle the challenge of large spatio-temporal datasets, the ROM/POD described above can be used to reduce the dimensional size of data. Both the training and predictive processes are then undertaken in a reduced space, thus reducing significantly the computational cost.

The input dataset is the incoming wave boundary condition:

$$x_s = (h_{b,s}^{cr}, h_{b,s}^{dr}, T_{b,s}), \quad (15)$$

where, $h_{b,s}^{cr}$ and $h_{b,s}^{dr}$ are the crest and trough wave heights respectively, and $T_{b,s}$ is the wave period. Three input variables $h_{b,s}^{cr}$, $h_{b,s}^{dr}$ and $T_{b,s}$ are used to define the incoming wave (Arhan and Plaisted, 1981).

Given S_w as the number of incoming wave series used for training purpose, the input data x_{s_w} in the training input dataset \mathcal{S}_{tr} are:

$$x_{s_w} = (h_{b,s_w}^{cr}, h_{b,s_w}^{dr}, T_{b,s_w}), \quad s_w \in \{1, \dots, S_w\}. \quad (16)$$

The corresponding output data h_{s_w} in the training dataset \mathcal{O}_{tr} are:

$$h_{s_w} = (h_{s_w,0}^T, \dots, h_{s_w,t_{N_t}}^T, \dots, h_{s_w,t_{N_t}}^T), \quad s_w \in \{1, \dots, S_w\}. \quad (17)$$

To reduce the dimensional size of $h_{s_w} \in \mathbb{R}^{\mathcal{N} \times N_t}$, using POD/SVD a set of POD modes is obtained from the snapshots $(h_1, \dots, h_{s_w}, \dots, h_{S_w})$ and used for constructing the reduced space $\mathcal{R}_m = \text{span}\{\phi_1, \dots, \phi_{\mathcal{M}}\}$. Projecting the training output data h_{s_w} onto the reduced space yields:

$$a_{s_w} = (a_{s_w,1}, \dots, a_{s_w,t_{N_t}}, \dots, a_{s_w,t_{N_t}}) = (h_{s_w,0}\Phi, \dots, h_{s_w,t_{N_t}}\Phi, \dots, h_{s_w,t_{N_t}}\Phi), \quad (18)$$

where $a_{s_w} \in \mathbb{R}^{\mathcal{M} \times N_t}$. The input-output dataset in Eq. (1) is (x_{s_w}, a_{s_w}) , $s_w \in \{1, \dots, S_w\}$. The LSTM training process is thus undertaken in the reduced space. In LSTM, most of input-output pairs $(x_{s_w}, a_{s_w}) \in (\mathcal{S}_{tr}, \mathcal{O}_{tr})$ are used for training while the rest for validation.

Once the LSTM training and validation processes are completed, one can find the relationship between the input-output pair (x_s, a_s) in the reduced space. By using LSTM based on ROM, the predictive solutions during the flooding period $[0, t_{N_t}]$ for any input data $x_s \in \mathcal{S} \setminus \mathcal{S}_{tr}$ can be obtained in Eq. (13).

The framework of LSTM-ROM methodology is demonstrated in Fig. 2.

4.2. Prescriptive analytics for uncertainty quantification in flooding modelling

Prescriptive/inverse analytics: \Rightarrow Given y_s , find $x_s = f^{-1}(y_s, t)$. Given a prescribed outcome y_s at time level t , using the previously learned dependencies, one can find an inverse function, $f^{-1}(y_s, t)$, which is used for optimising the inputs (or uncertainties in modelling). Flood modelling is a complex and parametric problem. The flood induced condition is the main factor of uncertainties in flood risk management. In this work, LSTM is used to conduct the prescriptive analytics to estimate the uncertainty in flood induced conditions (incoming waves) for given historical/estimated flood results.

To construct the LSTM network in the prescriptive analytics, the training input is the temporal variation of one flood characteristic (e.g. free surface height, water depth, quantity of flow, or velocity) at specified sensor locations, while the corresponding flood induced condition (e.g. waves or rainfall) is the output. Subsequently, the LSTM network is built through learning with the input-output pairs. Finally, given a new estimated temporal variation of the flood characteristic, the corresponding flood induced condition will be provided using the LSTM network.

As an example, for a series of induced waves, the input dataset is the prescribed free surface height at specified sensors:

$$h_{s_w}(\mathbf{x}_d) = (h_{s_w,0}\mathbf{x}_d, \dots, h_{s_w,t_{N_t}}\mathbf{x}_d, \dots, h_{s_w,t_{N_t}}\mathbf{x}_d), \quad s_w \in \{1, \dots, S_w\}, \quad d \in \{1, \dots, \mathcal{N}_d\}, \quad (19)$$

where \mathcal{N}_d is the number of sensors, while the corresponding induced wave is uncertainty to be determined, that is, the output data is the parameters used for defining the induced wave $x_s = h_{b,s} = (h_{b,s}^{cr}, h_{b,s}^{dr}, T_s)$. Using LSTM described in Eq. (1), the relationship $x_s = h_{b,s}(\mathbf{x}_d) = f^{-1}(h_{s_w}(\mathbf{x}_d))$ can be found and used to estimate the induced wave condition for given observational data at specified sensors.

5. Numerical examples

The example used for validation of the new LSTM-ROM is the Okushiri tsunami test case. The event is the Okushiri tsunami in 1993 caused by the Hokkaido Nansei-Oki earthquake offshore of south-western Hokkaido Island, Japan. The resulting tsunami hit a sparsely

populated part of the Okushiri Island, Japan with a runup height of up to 30 m. To investigate the danger of such extreme events, the Research Institute for Electric Power Industry (CRIEPI) in Abiko, Japan constructed a scaled-down laboratory model of the area around the island (Liu et al., 2008). The inundated area measures $5.448 \text{ m} \times 3.402 \text{ m}$ with walls on each side except the left where the water level is enforced (see Fig. 3). Fig. 3 shows the bathymetry and the three gauge stations, as well as the actual induced wave data used for the example case.

The tsunami problem is simulated using the 3D unstructured mesh finite element fluid model (Fluidity) (Pain et al., 2005). The equations are solved with the $P_1 - P_1$ finite element pair and a backward Euler time discretisation. The threshold value of wetting and drying (d_0) is set to be 0.5 mm in dry area to prevent non-physical flows in numerical simulations. A Manning-Strickler drag is used at the bottom with $n = 0.002 \text{ s m}^{-1/3}$. The acceleration of gravity magnitude is 9.81 ms^{-2} . In this case, the simulation period is $[0, 70] \text{ s}$, and a time step size is 0.1 s is used. Thus, the training data were obtained by running the full model (Fluidity), with an unstructured mesh of 6894 nodes.

5.1. Predictive analytics in flooding modelling

Calculation of snapshots via running the full model: As seen in Fig. 2, for training purpose, given a series of incoming waves (here, $s_w = 1, \dots, 10$), the corresponding solution snapshots $(h_{s_w} \in \mathcal{O}_{tr})$ are obtained from the full model simulations. By scaling the actual induced wave (in Fig. 2) with a factor of 0.7 – 1.25, 10 series of incoming waves (in Fig. 4) are given as inputs. Thus, for each incoming wave x_{s_w} , the collection of 100 snapshots during the period $[25, 35] \text{ s}$ for the free surface height is the output training dataset h_{s_w} . The dimensional size of the output dataset is large, i.e. $\mathcal{N} \times N_t = 6894 \times 100 = 689,400$. Reduction of the size of the training dataset is necessary for LSTM prediction.

Construction of LSTM-ROM: With the datasets of snapshots obtained above, a LSTM-ROM model is generated using POD/SVD techniques. Fig. 5 shows that the singular values decreases dramatically by two orders of magnitude between the first 3 leading POD modes and 99.5% of the total energy is captured by 20 POD modes. Accordingly, the first 20 leading POD modes ($\mathcal{M} = 20$) were chosen for constructing the reduced space, thus ensuring the predictive accuracy. The training and prediction processes were then undertaken over the reduced space. The training input dataset is the parameters used to define the incoming waves, $x_{s_w} = (h_{b,s_w}^{cr}, h_{b,s_w}^{dr}, T_{b,s_w})$, while the corresponding output dataset $a_{s_w} \in \mathbb{R}^{\mathcal{M} \times N_t}$ (where $\mathcal{M} = 20$ and $N_t = 100$) is obtained by projecting $h_{s_w} \in \mathbb{R}^{\mathcal{N} \times N_t}$ (where $\mathcal{N} = 6894$) onto the reduced space. By using ROM, the dimensional size of the output dataset is reduced from 689,400 to 2000. In LSTM model, the epochs is a measure of the number of times passing through the entire training dataset. The number of epochs should be large enough to allow the learning model fit optimally to the data. Here, the number of epochs is set to 1000 and RMSprop optimizer (Tieleman and Hinton, 2012) is used here.

Prediction of floods using LSTM-ROM: To evaluate the predicability of the LSTM-ROM, for the given actual input wave $(x_s \in \mathcal{S} \setminus \mathcal{S}_{tr})$ shown in Fig. 2, a comparison between the LSTM-ROM and full modelling results is carried out (see Fig. 6). It is worth noting that results from the full model and LSTM-ROM are in very good agreement with each other. As shown in Fig. 7, the differences of the free surface height between the full model and LSTM-ROM are quite small over the whole domain, which suggests that the accuracy of LSTM-ROM is maintained whilst the CPU cost is reduced by several orders of magnitude in comparison with the full model.

To further estimate the performance of the LSTM model, the root mean square error (RMSE) is calculated for each time step by,

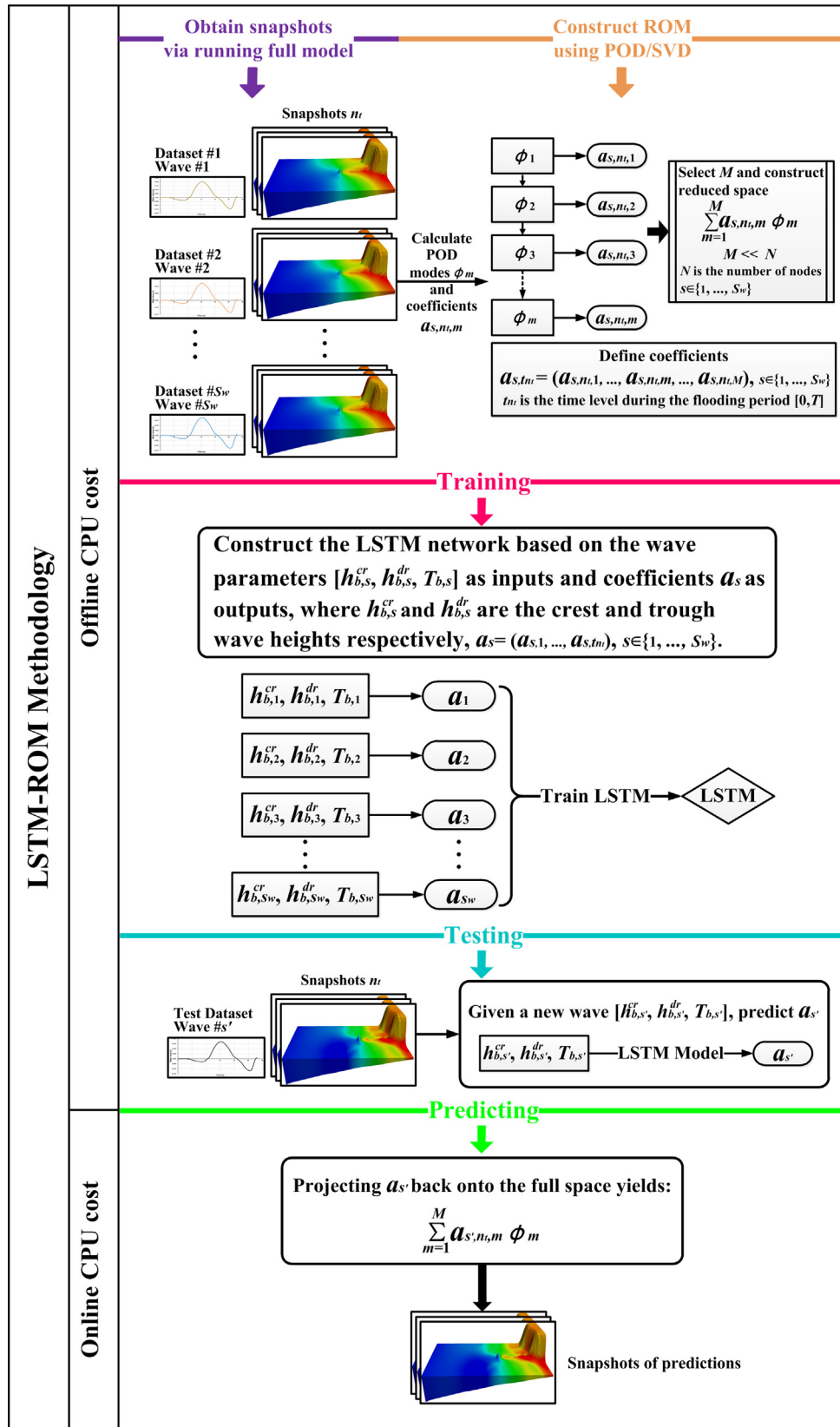


Fig. 2. Framework of LSTM-ROM.

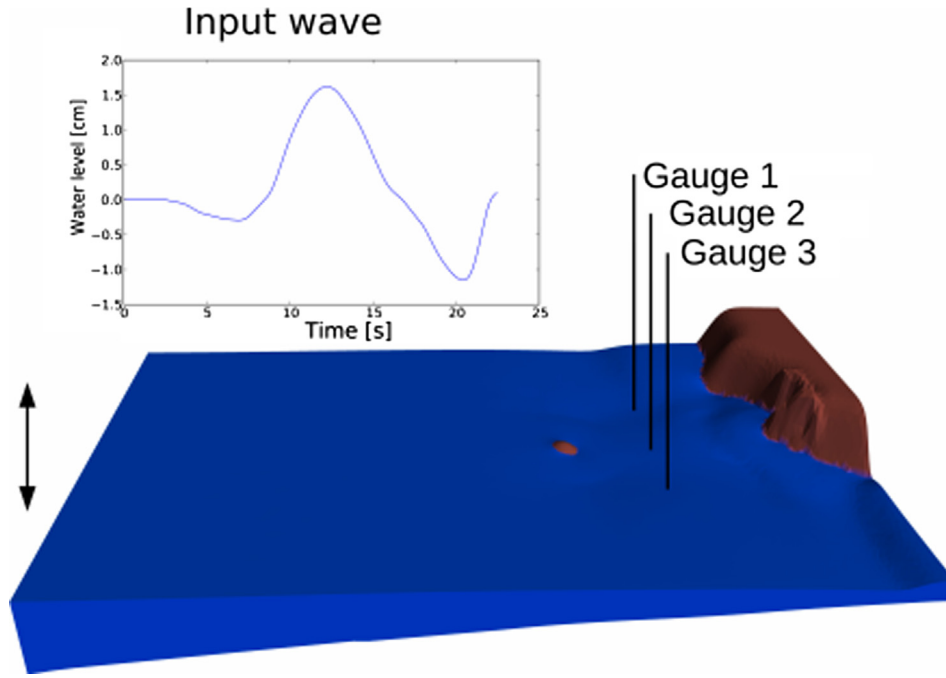


Fig. 3. The bathymetry and the three gauge stations used for the Hokkaido-Nansei-Oki tsunami example (Fluidity, manual 2015).

$$RMSE_j = \sqrt{\frac{\sum_{i=1}^N (S_{full}^{i,j} - S_{LSTM-ROM}^{i,j})^2}{N}}, \quad i = 1, 2, \dots, \mathcal{N}, j = 1, 2, \dots, N_t. \quad (20)$$

where $\mathcal{N} = 6894$ denotes the number of nodes on the full mesh; $N_t = 20$ is the number of time step; $S_{full}^{i,j}$ and $S_{LSTM-ROM}^{i,j}$ represent the solution on node i at time step j from the original full model and LSTM-ROM, respectively. As shown in Fig. 8, the errors are quite small, demonstrating that LSTM-ROM performs very well and the predicted solutions are in good agreement with the true solutions.

The LSTM prediction includes both offline and online procedures. The offline CPU cost is defined as the time for pre-computing, that is, the calculation of snapshots and POD modes and LSTM training. The online CPU cost is the time required for prediction of floods using LSTM-ROM for any given flood induced condition. In this study, the simulations were performed on a computer with 24 cores (Intel Xeon(R) CPU@ 2.90 GHz) and 16 GB RAM. Only one of the cores was used when

running the model. It is worth mentioning that the online CPU cost for running LSTM-ROM during one time-step is only 0.02 s, while the full model is 30.8 s. Although the offline CPU cost of training the LSTM network is somewhat expensive (331 s), it is much less than the simulation time of the full model (6160 s). Results obtained from this test case show that the novel LSTM-ROM can provide reasonable accuracy in flood prediction whilst the CPU time is reduced by three orders of magnitude.

In summary, for any given input $x_s \in \mathcal{S} \setminus \mathcal{S}_{tr}$, the LSTM-ROM can provide a better prediction of floods at a future time $t \in [0, t_{N_t}]$. However, it is also noted that the predictive accuracy of LSTM-ROM decays when the predictive lead-time is extended to beyond the period $[0, t_{N_t}]$. The longer the predictive lead-time, the larger the RMSE of LSTM results (see Table 1, similarly in the work of Babovic et al., 2001 and Sun et al., 2010). This indicates that capturing dependencies of features between successive time period in the model becomes more intractable as the prediction horizon extends in time. In this case, it is suggested that the use of data assimilation methods will help the

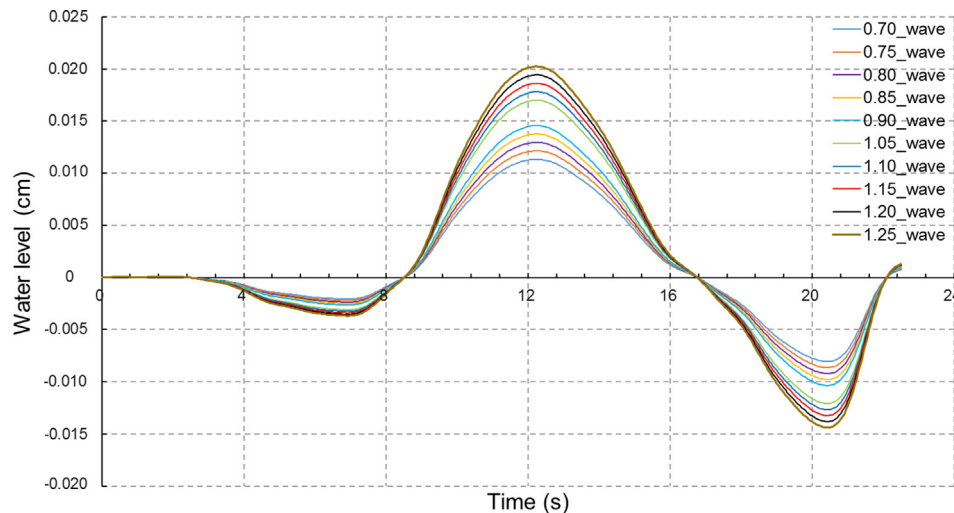


Fig. 4. The series of flood induced waves for training purpose.

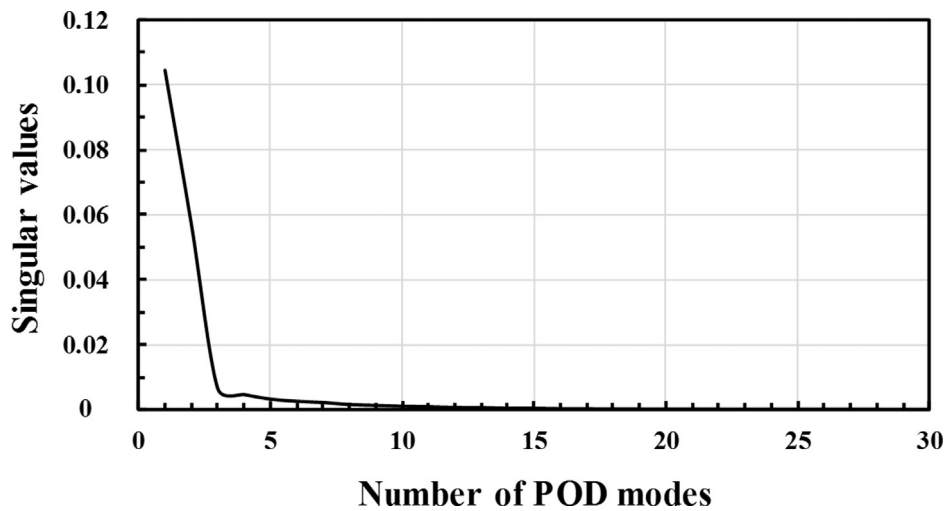


Fig. 5. Singular values of free surface heights to construct the reduced space.

improvement of model performance (Babovic et al., 2001). Using data assimilation techniques, only the reliable information from observations will be extracted to modify/correct the input datasets during the prediction process. It helps extending the prediction horizon and enables LSTM-ROM predictions in a real-time mode, which will be the focus of our future work.

5.2. Prescriptive analytics for uncertainty quantification in flooding modelling

The uncertainty quantification is critical in flood prediction. Can one estimate the incoming wave through LSTM prescriptive analytics if the temporal measurements of free surface heights at specified locations? In this study, ten flood scenarios are set up with the induced waves shown in Fig. 4. The corresponding pseudo-data at Gauge 2 are

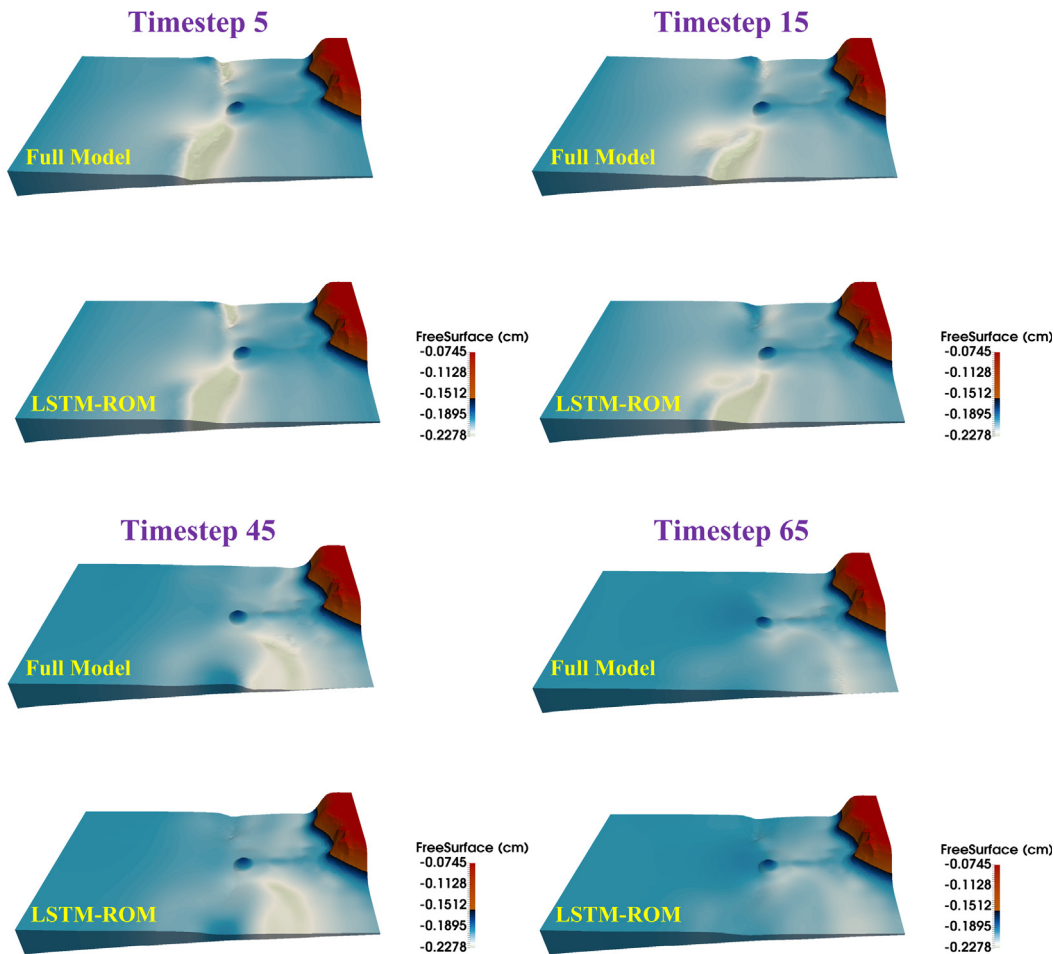


Fig. 6. The comparison of free surface heights between the full model and LSTM-ROM.

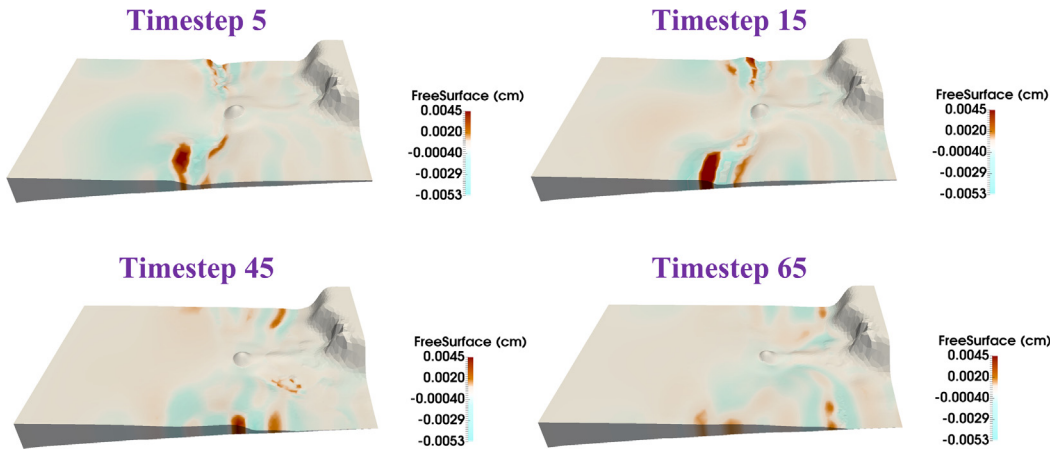


Fig. 7. Differences of free surface heights between the full model and LSTM-ROM during the whole period.

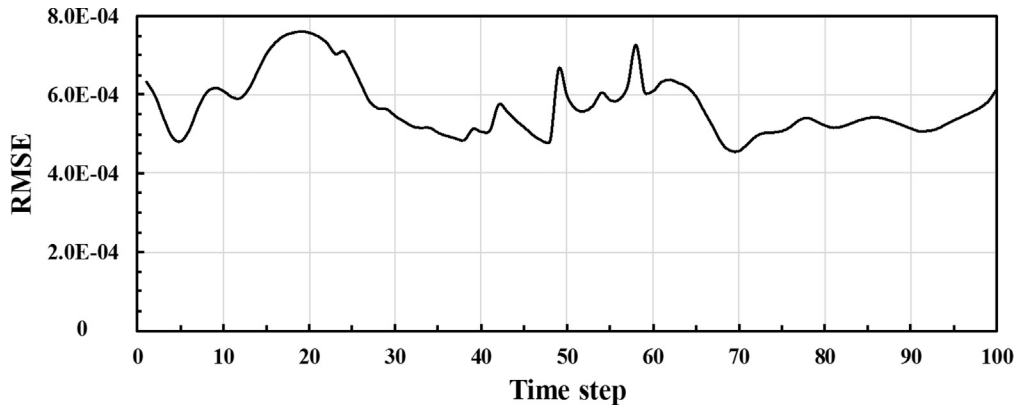


Fig. 8. The RMSE errors of free surface heights between the original full model and LSTM-ROM.

Table 1
The RMSE errors of free surface heights between the original full model and LSTM-ROM during predicted period [67.5, 70] s.

Time (s)	67.5	68	68.5	69	69.5	70
RMSE ($\times 10^{-3}$)	2.16	2.67	4.03	5.99	8.39	11.13

shown in Fig. 9 which are obtained by running the full model. Here 12 key values are selected from each incoming wave (values marked with red circle in Fig. 9) as the input data for training, while choosing 3 key values from the corresponding wave time series as the output data for training. The LSTM network would thus be constructed through the training process and used to estimate the flood induced condition (here, the incoming wave condition) for any given time series of free surface height at Gauge 2.

For a given measurement of free surface heights at Gauge 2 shown in Fig. 10 (left), the corresponding flood induced condition was estimated using the established LSTM network (shown in the right of Fig. 10). It is seen that the percent relative errors (%) between the estimated and actual wave peak values (marked with green rectangles) are (0.01, −1.56), (0.12, 0.31), (0.05, −0.23), respectively. It is clear that in the prescriptive analytics, the corresponding flood induced condition can be accurately provided using the LSTM network.

6. Conclusions

In this work, a RNN architecture LSTM in combination with POD/ROM has been proposed for flood prediction, which simultaneously takes advantage of both LSTM and ROM. This is the first time that the

ROM based LSTM network has been considered in flood forecasting and used to estimate spatial aggregation of inundations. This novel integrated LSTM-ROM framework which consists of both predictive and prescriptive analytics leads to a step change in the speed of forecasting, thus providing effective response management in emergencies.

The performance of the new method has been illustrated using Okushiri tsunami test cases. To estimate the accuracy of the LSTM-ROM, a comparison of results obtained from the LSTM-ROM and full model has been undertaken. It is shown that the accuracy of solutions from the LSTM-ROM is maintained while the CPU cost is reduced by three orders of magnitude. An error analysis has also been performed for the validation and accuracy of the LSTM-ROM through RMSE. Comparing LSTM-ROM results with those from the full model, the LSTM-ROM exhibits a good agreement with the full model. Subsequently in the prescriptive analytics, the corresponding flood induced conditions have been accurately provided using the LSTM network. Having the compatibility of LSTM and ROM, the integrated framework will enable predictive problems to be solved rapidly and accurately.

In our future work, further evaluation of LSTM-ROMs in flood forecast will be carried out with different forecasting configurations including optimized training parameters, the predictive lead-time and the type of inputs. For example, the selection of training parameters in the networks (e.g. values for time delays and embedding dimension) is critical for LSTM model predictive performance optimization. With optimal architectures, the underlying structure of nonlinear time series can be better viewed (Sun et al., 2010). To further improve predictive accuracy, data assimilation methods can be adopted. The data assimilation techniques can improve the accuracy of model prediction by updating output variables based on the observed variables during a

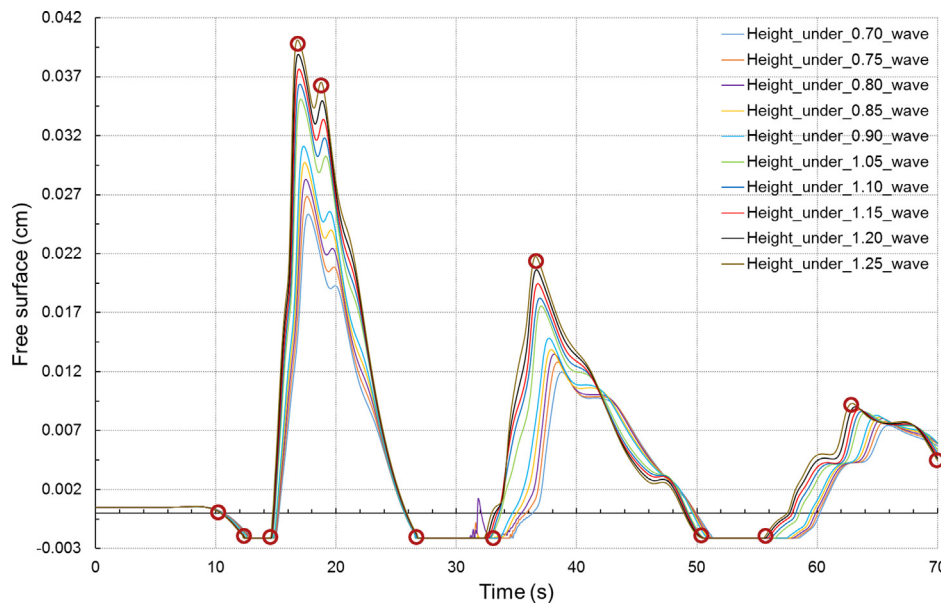


Fig. 9. (a) The free surface height at Gauge 2.

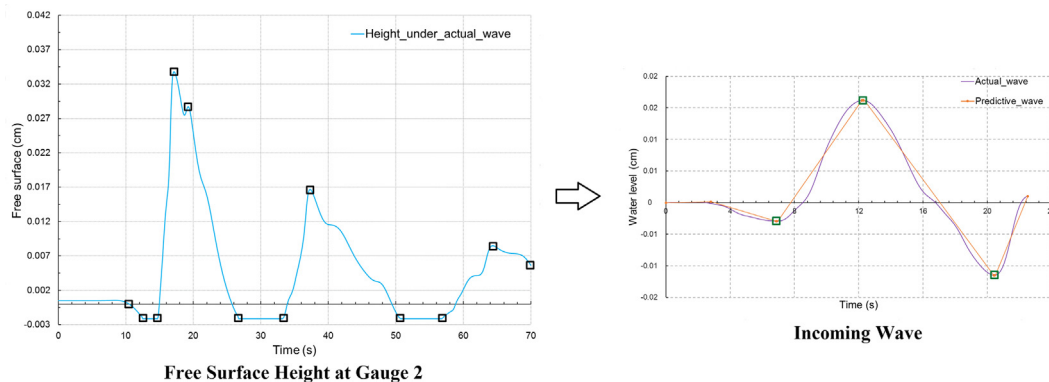


Fig. 10. The input (left) and output (right) in prescriptive analytics.

forecast period. For instance, given historical/simulated datasets of snapshots from floods, the target value (e.g. water depth) of the whole domain for the next few time steps in the future will be predicted. Usually, the performance of the model decays as the forecasting horizon extends beyond these few time steps, while data assimilation may be the most suitable approach to address this issue.

Declaration of Competing Interest

None.

Acknowledgments

The Authors acknowledge the support of: the NSFC grant 11502241 and funding from the EPSRC (MAGIC) (EP/N010221/1) in the UK.

References

- Alotaibi, M., Calo, V.M., Efendiev, Y., Galvis, J., Ghomm, M., 2015. Global-local nonlinear model reduction for flows in heterogeneous porous media. *Comput. Methods Appl. Mech. Eng.* 292, 122–137.
- Alves, J.M.M., et al., 2010. Autoencoders: application to forecasting (Master thesis). University of Porto, Porto.
- Arhan, M., Plaisted, R., 1981. Non-linear deformation of sea-wave profiles in intermediate and shallow water. *Oceanol. Acta* 4 (2), 107–115.
- Babovic, V., Cañizares, R., Jensen, H.R., Kliting, A., 2001. Neural networks as routine for error updating of numerical models. *J. Hydraul. Eng.* 127 (3), 181–193.
- Bertolami, R., Bunke, H., Fernandez, S., Graves, A., Liwicki, M., Schmidhuber, J., 2009. A novel connectionist system for improved unconstrained handwriting recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* 31 (5).
- Chang, F.-J., Chen, P.-A., Lu, Y.-R., Huang, E., Chang, K.-Y., 2014. Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. *J. Hydrol.* 517, 836–846.
- Chen, P.-A., Chang, L.-C., Chang, F.-J., 2013. Reinforced recurrent neural networks for multi-step-ahead flood forecasting. *J. Hydrol.* 497, 71–79.
- Dimitriu, G., Apreutesei, N., 2007. Comparative study with data assimilation experiments using proper orthogonal decomposition method. In: *International Conference on Large-Scale Scientific Computing*. Springer, pp. 393–400.
- Dimitriu, G., Apreutesei, N., Ștefănescu, R., 2009. Numerical simulations with data assimilation using an adaptive POD procedure. In: *International Conference on Large-Scale Scientific Computing*. Springer, pp. 165–172.
- Doetsch, P., Kozielski, M., Ney, H., 2014. Fast and robust training of recurrent neural networks for offline handwriting recognition. In: *Frontiers in Handwriting Recognition (ICFHR)*, 2014 14th International Conference on. IEEE, pp. 279–284.
- Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T., 2015. Long-term recurrent convolutional networks for visual recognition and description. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2625–2634.
- Fan, Y., Qian, Y., Xie, F.-L., Soong, F.K., 2014. TTS synthesis with bidirectional LSTM based recurrent neural networks. In: *Fifteenth Annual Conference of the International Speech Communication Association*.
- Fluidity, manual 2015. Fluidity manual v4.1.12, Applied Modelling and Computation Group, Imperial College. URL: https://figshare.com/articles/Fluidity_Manual/1387713.
- Hoang, K., Fu, Y., Song, J., 2016a. An hp-proper orthogonal decomposition-moving least squares approach for molecular dynamics simulation. *Comput. Methods Appl. Mech. Eng.* 298, 548–575.
- Hoang, K.C., Kerfriden, P., Bordas, S.P.A., 2016b. A fast, certified and “tuning free” two-field reduced basis method for the metamodeling of affinely-parametrised elasticity problems. *Comput. Methods Appl. Mech. Eng.* 298, 121–158.

- Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. *Neural Comput.* 9 (8), 1735–1780.
- Le Quilliec, G., Raghavan, B., Breikopf, P., 2015. A manifold learning-based reduced order model for springback shape characterization and optimization in sheet metal forming. *Comput. Methods Appl. Mech. Eng.* 285, 621–638.
- Lin, G.-F., Chen, G.-R., 2008. A systematic approach to the input determination for neural network rainfall-runoff models. *Hydrol. Processes: Int. J.* 22 (14), 2524–2530.
- Lin, G.-F., Chen, L.-H., 2004. A non-linear rainfall-runoff model using radial basis function network. *J. Hydrol.* 289 (1–4), 1–8.
- Liu, P.L.F., Yeh, H., Synolakis, C., 2008. Advanced numerical models for simulating tsunami waves and runup. *Adv. Coast. Ocean Eng.*
- Manzoni, A., Salmoiraghi, F., Heltai, L., 2015. Reduced basis Isogeometric Methods (RB-IGA) for the real-time simulation of potential flows about parametrized NACA airfoils. *Comput. Methods Appl. Mech. Eng.* 284, 1147–1180.
- Marchi, E., Ferroni, G., Eyben, F., Gabrielli, L., Squartini, S., Schuller, B., 2014. Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. In: *Acoustics, Speech and Signal Processing (ICASSP)*, 2014 IEEE International Conference on. IEEE, pp. 2164–2168.
- Mosavi, A., Ozturk, P., Chau, K.-W., 2018. Flood prediction using machine learning models: literature review. *Water* 10 (11), 1536.
- Mukerji, A., Chatterjee, C., Raghuvanshi, N.S., 2009. Flood forecasting using ANN, neuro-fuzzy, and neuro-GA models. *J. Hydrol. Eng.* 14 (6), 647–652.
- Pain, C., Piggott, M., Goddard, A., Fang, F., Gorman, G., Marshall, D., Eaton, M., Power, P., De Oliveira, C., 2005. Three-dimensional unstructured mesh ocean modelling. *Ocean Model.* 10 (1–2), 5–33.
- Peherstorfer, B., Willcox, K., 2016. Data-driven operator inference for nonintrusive projection-based model reduction. *Comput. Methods Appl. Mech. Eng.* 306, 196–215.
- Pham, V., Bluche, T., Kermorvant, C., Louradour, J., 2014. Dropout improves recurrent neural networks for handwriting recognition. In: *Frontiers in Handwriting Recognition (ICFHR)*, 2014 14th International Conference on. IEEE, pp. 285–290.
- Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. I. Coherent structures. *Q. Appl. Math.* 45 (3), 561–571.
- Soltanpoor, R., Sellis, T., 2016. Prescriptive analytics for big data. In: *Australasian Database Conference*. Springer, pp. 245–256.
- Sun, Y., Babovic, V., Chan, E.S., 2010. Multi-step-ahead model error prediction using time-delay neural networks combined with chaos theory. *J. Hydrol.* 395 (1–2), 109–116.
- Tieleman, T., Hinton, G., 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Netw. Mach. Learn.* 4 (2), 26–31.
- Vahn, G.-Y., 2014. Business analytics in the age of Big Data. *Business Strategy Rev.* 25 (3), 8–9.
- Wang, Y., Navon, I.M., Wang, X., Cheng, Y., 2016. 2D Burgers equation with large Reynolds number using POD/DEIM and calibration. *Int. J. Numer. Meth. Fluids* 82 (12), 909–931.
- Wu, C., Chau, K., 2006. A flood forecasting neural network model with genetic algorithm. *Int. J. Environ. Pollut.*
- Zaremba, W., Sutskever, I., Vinyals, O., 2014. Recurrent neural network regularization. [arXiv:1409.2329](https://arxiv.org/abs/1409.2329).